PEMODELAN DAN SIMULASI KEJADIAN DISKRIT PADA SISTEM ANTREAN KLINIK KESEHATAN MENTAL MENGGUNAKAN ALGORITMA GAUSSIAN NAÏVE BAYES DAN DESCRETE EVENT SIMULATION

Naufal Prmaudya Ananda [1)], Septian Dwi Saputra[1)], Nanang Alfian Rizki[1)], Afif Dhiyaulhaq[1)] , Herliana Rosika[2)]

Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: naufalananda79@gmail.com

Abstrak

Tingginya kesadaran akan kesehatan mental menyebabkan peningkatan kunjungan pasien di fasilitas kesehatan, yang berpotensi memicu penumpukan antrean dan memperburuk kondisi psikologis pasien akibat waktu tunggu yang lama. Penelitian ini bertujuan untuk memodelkan dinamika sistem antrean faskes dan menentukan alokasi jumlah psikolog yang paling optimal guna meminimalkan waktu tunggu tanpa mengabaikan efisiensi biaya operasional. Metode yang digunakan adalah Simulasi Kejadian Diskrit (*Discrete Event Simulation*) yang terintegrasi dengan algoritma Gaussian Naïve Bayes. Klasifikasi Naïve Bayes digunakan pada tahap triase awal untuk memprediksi probabilitas jenis gangguan jiwa seperti Depresi atau Bipolar berdasarkan data kuesioner pasien. Hasil probabilitas tersebut kemudian dikonversi menjadi durasi waktu pelayanan (*service time*) dinamis di dalam ruang konsultasi, sementara laju kedatangan pasien dimodelkan menggunakan distribusi eksponensial. Hasil simulasi menunjukkan bahwa operasional klinik dengan satu psikolog menghasilkan efek bola salju (*snowball effect*) dengan waktu tunggu pasien mencapai 113 menit, yang menandakan sistem tidak layak secara medis. Skenario penambahan kapasitas menjadi dua psikolog terbukti sebagai titik keseimbangan (*trade-off*) yang paling optimal, mampu mereduksi penumpukan secara signifikan dan menstabilkan rata-rata waktu tunggu di bawah 20 menit.
Kata kunci: Gaussian Naïve Bayes; Kesehatan Mental; Simulasi Kejadian Diskrit; Teori Antrean; Waktu Tunggu

Abstract

The increasing awareness of mental health has led to a surge in patient visits at healthcare facilities, potentially causing queue congestion and worsening patients' psychological conditions due to prolonged waiting times. This study aims to model the dynamics of the healthcare queueing system and determine the optimal allocation of psychologists to minimize waiting times without neglecting operational cost efficiency. The method used is Discrete Event Simulation integrated with the Gaussian Naïve Bayes algorithm. Naïve Bayes classification is applied at the initial triage stage to predict the probability of mental disorders, such as Depression or Bipolar, based on patient questionnaire data. The resulting probabilities are then translated into dynamic service times within the consultation room, while the patient arrival rate is modeled using an exponential distribution. Simulation results indicate that operating the clinic with a single psychologist creates a snowball effect, with maximum patient waiting times reaching 113 minutes, rendering the system medically unfeasible. The scenario of expanding capacity to two psychologists proved to be the most optimal trade-off point, significantly reducing congestion and stabilizing the average waiting time under 20 minutes.
Keyword: *Discrete Event Simulation; Gaussian Naïve Bayes; Mental Health; Queueing Theory; Waiting Time*

I. PENDAHULUAN

Kesehatan mental merupakan salah satu aspek kesehatan yang semakin mendapat perhatian serius dalam beberapa tahun terakhir. Meningkatnya kesadaran masyarakat terhadap pentingnya kesehatan psikologis menyebabkan lebih banyak individu datang ke fasilitas kesehatan mental untuk memperoleh pemeriksaan, konseling, maupun terapi. Fenomena ini merupakan perkembangan yang positif karena menunjukkan bahwa masyarakat mulai memahami pentingnya penanganan kesehatan jiwa secara profesional. Namun, peningkatan jumlah pasien juga memunculkan masalah baru, yaitu ketidakseimbangan antara permintaan layanan dan ketersediaan sumber daya manusia, terutama psikolog dan psikiater [1].

Dalam praktik layanan kesehatan mental, waktu tunggu merupakan faktor yang sangat

sensitif. Pasien dengan gejala depresi, kecemasan, bipolar, atau gangguan mental lainnya sangat mungkin mengalami tekanan psikologis tambahan ketika harus menunggu terlalu lama sebelum diperiksa. Berbagai studi menunjukkan bahwa waiting time yang panjang pada layanan kesehatan mental berkaitan dengan peningkatan distress psikologis dan persepsi pasien bahwa kondisinya memburuk selama masa tunggu,. Pada situasi tertentu, pasien juga dapat kehilangan motivasi untuk melanjutkan layanan karena merasa tidak mendapatkan penanganan yang cepat dan responsif. Dengan demikian, pengelolaan antrean tidak hanya berkaitan dengan efisiensi operasional, tetapi juga dengan keamanan dan kenyamanan pasien [2].

Permasalahan tersebut mendorong perlunya sistem yang mampu mengintegrasikan prediksi kondisi pasien dengan pemodelan aliran layanan secara dinamis. Dalam penelitian ini, dibangun suatu purwarupa sistem antrean cerdas dan digital twin untuk klinik kesehatan mental. Sistem ini tidak hanya mensimulasikan antrean secara konvensional, tetapi juga memanfaatkan data kuesioner pasien sebagai dasar triase awal, sehingga durasi konsultasi dapat disesuaikan dengan tingkat keparahan gangguan yang diprediksi. Pendekatan ini diharapkan mampu membantu klinik dalam mengambil keputusan terkait alokasi psikolog, penjadwalan layanan, dan pengendalian antrean [3].

Untuk merealisasikan tujuan tersebut, penelitian ini menggunakan Gaussian Naïve Bayes sebagai algoritma klasifikasi probabilistik dan Simulasi Kejadian Diskrit sebagai metode pemodelan sistem pelayanan. Gaussian Naïve Bayes dipilih karena mampu menangani fitur numerik kontinu secara efisien dan menghasilkan prediksi probabilistik yang dapat diinterpretasikan untuk kebutuhan triase medis,. Sementara itu, Simulasi Kejadian Diskrit berbasis SimPy digunakan untuk merepresentasikan interaksi pasien, psikolog, dan ruang pelayanan dalam ruang waktu kronologis yang realistis,. Dengan mengombinasikan kedua pendekatan tersebut, penelitian ini bertujuan membangun model keputusan yang dapat digunakan untuk mengevaluasi skenario jumlah psikolog serta dampaknya terhadap waktu tunggu dan stabilitas antrean [4].

Selain meminimalkan waktu tunggu pasien, penelitian ini juga mengintegrasikan analisis efektivitas biaya operasional (cost-effectiveness) untuk membantu manajemen fasilitas kesehatan dalam mengambil keputusan alokasi sumber daya manusia yang paling efisien, sehingga tercipta keseimbangan antara standar keselamatan medis dan efisiensi finansial

II. TINJAUAN PUSTAKA

Kesehatan mental mencakup kondisi kesejahteraan psikologis yang memungkinkan seseorang berpikir, merasa, dan bertindak secara adaptif dalam kehidupan sehari-hari. Dalam konteks pelayanan kesehatan, gangguan mental yang tidak ditangani secara tepat dapat memperburuk kualitas hidup pasien dan menambah beban sistem kesehatan. Sejumlah penelitian menunjukkan bahwa akses layanan kesehatan mental masih menjadi tantangan di banyak negara karena keterbatasan tenaga profesional, tingginya permintaan layanan, serta waktu tunggu yang panjang,. Pada level pelayanan, antrean yang terlalu lama dapat mengurangi efektivitas intervensi dini dan meningkatkan risiko memburuknya kondisi pasien [5].

Teori antrean digunakan untuk memahami sistem pelayanan yang memiliki aliran kedatangan, proses menunggu, dan pelayanan. Dalam sistem antrean, pasien dapat dipandang sebagai entitas yang datang secara stokastik, menunggu di antrian bila server sibuk, dan meninggalkan sistem setelah dilayani. Parameter utama yang diamati dalam teori antrean mencakup laju kedatangan, laju pelayanan, jumlah server, waktu tunggu, dan panjang antrean. Jika laju kedatangan lebih besar daripada kapasitas pelayanan, maka antrean akan tumbuh dan sistem cenderung tidak stabil,. Oleh karena itu, teori antrean sangat relevan digunakan untuk menganalisis kapasitas layanan pada klinik kesehatan mental [6].

Simulasi Kejadian Diskrit merupakan pendekatan yang sesuai untuk sistem antrean karena perubahan status sistem hanya terjadi pada momen tertentu, seperti saat pasien datang, mulai dilayani, atau selesai berkonsultasi. Dengan simulasi, peneliti dapat membangun replika digital dari sistem nyata tanpa mengganggu operasional klinik. SimPy sebagai pustaka Python

banyak digunakan untuk membangun model DES karena menyediakan mekanisme event scheduling, resource contention, dan process interaction yang mudah diimplementasikan,. Dalam konteks penelitian ini, SimPy digunakan untuk menyusun digital twin yang merepresentasikan alur layanan klinik kesehatan mental secara kronologis [7].

Gaussian Naïve Bayes adalah salah satu algoritma klasifikasi yang didasarkan pada Teorema Bayes dengan asumsi independensi antarfitur. Pada versi Gaussian, distribusi fitur numerik diasumsikan mengikuti distribusi normal. Algoritma ini cocok untuk data gejala klinis yang bersifat numerik kontinu, seperti skor kuesioner kesehatan mental. Beberapa studi menunjukkan bahwa Naïve Bayes dapat memberikan performa yang baik pada klasifikasi kesehatan mental, terutama ketika dipakai untuk data yang tidak terlalu besar namun memiliki fitur-fitur yang relevan,. Dalam penelitian ini, Gaussian Naïve Bayes digunakan sebagai agen triase untuk mengklasifikasikan pasien ke dalam kelas Normal, *Depression*, dan Bipolar Type-2 [8].

Penelitian sebelumnya mengenai waktu tunggu layanan kesehatan mental menunjukkan bahwa durasi tunggu yang panjang sering dianggap terlalu lama oleh pasien dan berhubungan dengan peningkatan distress psikologis,. Selain itu, penelitian pada sistem klasifikasi kesehatan mental juga menunjukkan bahwa pendekatan Naïve Bayes dapat digunakan sebagai metode prediksi yang ringan namun efektif,. Berdasarkan temuan tersebut, integrasi antara klasifikasi probabilistik dan simulasi kejadian diskrit menjadi pendekatan yang logis untuk membangun sistem pendukung keputusan pada klinik kesehatan mental [9].

III. METODOLOGI PENELITIAN

Penelitian ini dilaksanakan melalui serangkaian tahap sistematis yang mengintegrasikan pendekatan klasifikasi berbasis probabilitas dengan simulasi kejadian diskrit. Diagram alir metodologi penelitian disajikan pada Gambar 2, sedangkan uraian masing-masing tahap dijelaskan sebagai berikut.

Tahap 1: Studi Literatur.
Peneliti melakukan kajian mendalam mengenai teori antrean dan model M/M/c, konsep Simulasi Kejadian Diskrit (DES), serta algoritma Gaussian Naïve Bayes. Kajian juga mencakup penelitian terdahulu yang relevan mengenai penggunaan DES dalam sistem layanan kesehatan mental

Tahap 2: Pengumpulan Data.
Data penelitian diperoleh dari basis data sekunder (Kaggle) yang berisi kuesioner kesehatan mental, yang mencakup respons pasien terhadap berbagai gejala psikologis seperti sadness, euphoric, concentration, dan overthinking sebagai input klasifikasi. Selain data kuesioner, parameter operasional klinik meliputi laju kedatangan pasien, jumlah psikolog yang tersedia, dan estimasi durasi konsultasi diperoleh dari data pendukung fasilitas kesehatan yang digunakan sebagai dasar pemodelan simulasi.

Tahap 3: Preprocessing Data.
Data kuesioner diproses melalui pembersihan data (*data cleaning*), penanganan nilai yang hilang, dan normalisasi fitur agar seluruh atribut memiliki skala yang sebanding sebelum diolah oleh algoritma

Tahap 4: Klasifikasi Gaussian Naïve Bayes.
Algoritma Gaussian Naïve Bayes diterapkan pada data kuesioner yang telah diproses untuk memprediksi probabilitas jenis gangguan jiwa setiap pasien, yaitu Depresi atau Bipolar. Algoritma ini mengestimasi parameter mean (μ) dan variansi (σ^2) dari data latih, kemudian menghitung probabilitas posterior untuk masing-masing kelas berdasarkan Teorema Bayes dengan asumsi independensi kondisional antar fitur.

Tahap 5: Konversi Probabilitas ke Service Time.
Output probabilitas dari GNB dikonversi menjadi durasi waktu pelayanan (service time) yang bersifat dinamis. Pasien dengan probabilitas Bipolar yang lebih tinggi diasumsikan membutuhkan durasi konsultasi yang lebih panjang dibandingkan pasien dengan dominansi Depresi, sehingga setiap pasien memiliki service time yang unik sesuai kondisi psikologisnya.

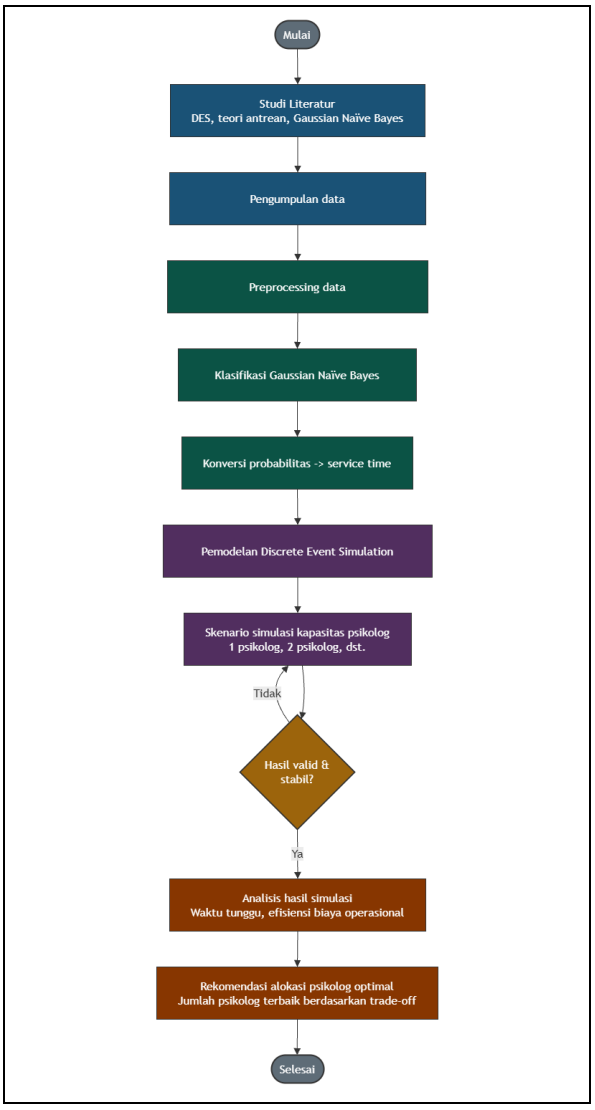
Tahap 6: Pemodelan Discrete Event Simulation.
Model DES dibangun dengan mengadopsi struktur M/M/c di mana laju kedatangan pasien mengikuti distribusi Poisson dengan parameter λ dan waktu pelayanan bersifat dinamis berdasarkan output GNB. Entitas dalam model merepresentasikan pasien, sumber daya merepresentasikan psikolog, dan antrean

terbentuk ketika seluruh psikolog sedang dalam kondisi sibuk melayani pasien.

Tahap 7: Simulasi Skenario Kapasitas.
Simulasi dijalankan pada tiga skenario kapasitas yang berbeda, yaitu klinik dengan satu psikolog, dua psikolog, dan tiga psikolog. Setiap skenario dijalankan dalam rentang waktu operasional satu hari kerja penuh guna menangkap dinamika penumpukan antrean secara komprehensif. Apabila hasil simulasi belum menunjukkan keluaran yang valid dan stabil, parameter model dievaluasi ulang dan simulasi diulang.

Tahap 8: Analisis Hasil Simulasi.
Hasil dari setiap skenario simulasi dianalisis berdasarkan tiga metrik utama, yaitu rata-rata waktu tunggu pasien dalam antrean, tingkat utilisasi psikolog, dan estimasi biaya operasional. Perbandingan antar skenario dilakukan untuk mengidentifikasi titik keseimbangan (trade-off) antara kualitas layanan dan efisiensi biaya.

Tahap 9: Rekomendasi Alokasi Psikolog Optimal.
Berdasarkan hasil analisis komparatif ketiga skenario, ditarik rekomendasi jumlah psikolog yang paling optimal untuk dioperasikan di klinik kesehatan mental. Rekomendasi ini mempertimbangkan standar kelayakan medis atas waktu tunggu pasien sekaligus efisiensi dari sisi biaya operasional klinik.



Gambar 1. Diagram Alir Metodologi Penelitian

IV. HASIL DAN PEMBAHASAN

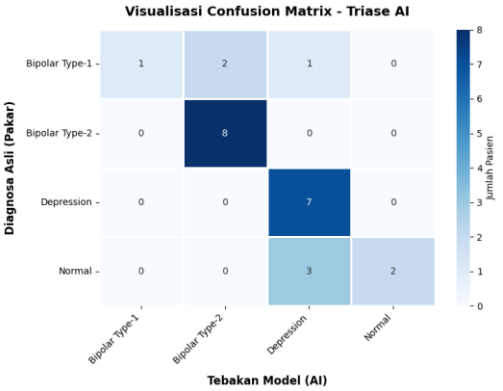
Model Gaussian Naïve Bayes dievaluasi dan mencapai tingkat akurasi 75,00%, dengan deteksi sempurna (100%) pada kasus depresi dan Bipolar tipe-2. Hasil prediksi probabilitas ini sukses dikonversi menjadi waktu pelayanan dinamis antara 15 hingga 60 menit. Simulasi dieksekusi selama durasi 4 jam operasional dengan membandingkan dua skenario kapasitas sumber daya.

Pada penelitian ini, pengujian sistem dilakukan dalam dua tahap utama: evaluasi kinerja model kecerdasan buatan sebagai agen

triase, dan analisis dinamika sistem antrean menggunakan Simulasi Kejadian Diskrit.

A. Evaluasi Kinerja Model Gaussian Naïve Bayes

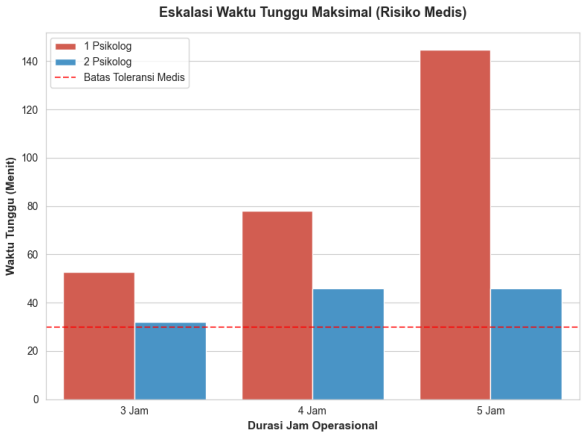
Model Gaussian Naïve Bayes dilatih menggunakan 96 data latih dan dievaluasi menggunakan 24 data uji (rasio 80:20). Hasil pengujian menunjukkan bahwa model mencapai akurasi keseluruhan sebesar 75,00% (18 tebakan benar dari 24 pasien uji). Rincian kinerja klasifikasi dapat dilihat melalui matriks kebingungan (confusion matrix) pada Gambar 3.



Gambar 2. Hasil Prediksi Model

Visualisasi heatmap pada Gambar 2 menunjukkan kinerja model Gaussian Naïve Bayes dengan keberhasilan (recall) 100% pada deteksi Bipolar Type-2 dan Depression. Meskipun memunculkan anomali False Positive (pasien Normal diprediksi Depression), sifat over-sensitive ini justru sangat ideal untuk keamanan triase medis. Karakteristik ini mencegah bahaya False Negative (mengabaikan pasien berisiko), sehingga menjamin setiap pasien mendapatkan alokasi waktu pelayanan yang memadai dalam simulasi antrean.

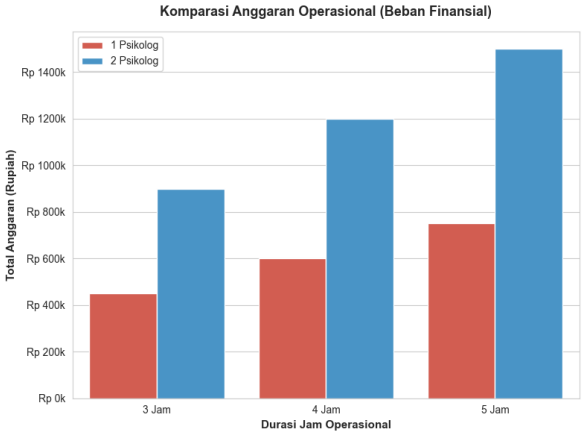
B. Perbandingan Estimasi Waktu Tunggu Untuk Simulasi Skenario



Gambar 3. Hasil Prediksi Waktu Tunggu Maksimal

Visualisasi eskalasi waktu tunggu membuktikan secara empiris terjadinya kegagalan operasional fatal pada skenario 1 psikolog (batang merah), di mana waktu tunggu memuncak secara progresif seiring bertambahnya jam operasional hingga menyentuh angka lebih dari 140 menit, jauh melampaui batas toleransi medis 30 menit (garis merah putus-putus). Sebaliknya, skenario 2 psikolog (batang biru) mendemonstrasikan ketahanan sistem yang tinggi (steady-state); meskipun klinik beroperasi hingga 5 jam berturut-turut, mekanisme keseimbangan beban (load balancing) sukses menahan efek bola salju sehingga fluktuasi waktu tunggu tetap landai dan terkendali di dekat batas aman.

C. Beban Anggaran Operasional Berdasarkan Jumalh Psikolog

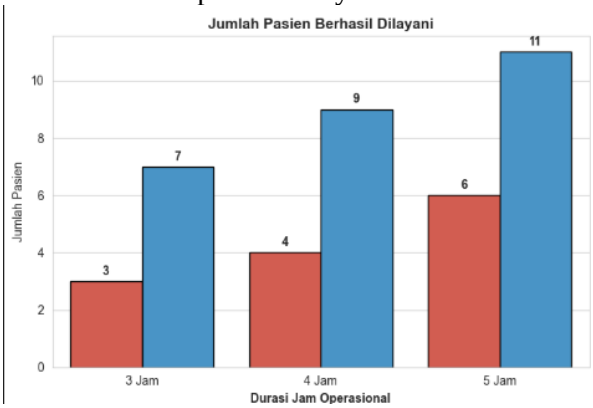


Gambar 4. Komparasi Anggaran Operasional

Grafik komparasi anggaran menunjukkan bahwa alokasi 2 psikolog secara logis menuntut

beban finansial dua kali lipat lebih besar dibandingkan 1 psikolog. Meskipun tampak sebagai pembengkakan biaya, pengeluaran ini merupakan titik ekuilibrium (optimal *trade-off*) yang rasional dan esensial untuk mencegah kolapsnya sistem antrean sekaligus menjamin keselamatan psikologis pasien. Analisis efektivitas biaya membuktikan bahwa konfigurasi dua psikolog selama tiga jam operasional merupakan strategi alokasi terbaik bagi manajemen fasilitas kesehatan. Skenario ini tidak hanya mampu memangkas waktu tunggu pasien secara drastis dari 140 menit menjadi di bawah 30 menit, tetapi durasinya yang singkat dengan kapasitas tinggi juga efektif menghindari pemborosan anggaran akibat idle time tenaga medis, sehingga tercipta keseimbangan yang ideal antara standar pelayanan klinis yang manusiawi dan efisiensi finansial rumah sakit.

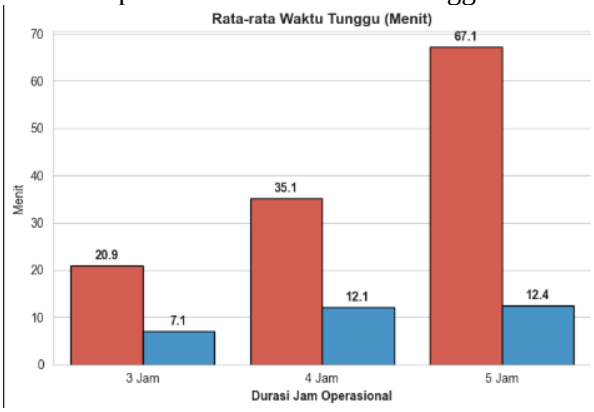
D. Analisis Kapasitas Pelayanan



Gambar 5. Komparasi Kapasitas Pelayanan

Berdasarkan visualisasi pada Gambar 5, penambahan jumlah sumber daya psikolog berbanding lurus dengan peningkatan jumlah pasien yang berhasil dilayani (tingkat throughput). Pada skenario operasional dengan 1 psikolog, kapasitas maksimal yang dapat ditangani dalam rentang 5 jam operasional hanyalah 6 pasien. Hal ini disebabkan oleh durasi konsultasi kesehatan mental yang bervariasi dan cenderung memakan waktu lama. Sebaliknya, dengan mengalokasikan 2 psikolog pada durasi operasional yang sama, sistem antrean mampu menyerap dan melayani hingga 11 pasien. Fakta ini membuktikan bahwa penambahan satu unit sumber daya medis tidak hanya mempercepat aliran antrean, tetapi juga meningkatkan kapasitas penyerapan fasilitas kesehatan hingga hampir dua kali lipat (peningkatan sekitar 83%).

E. Komparasi Rata-Rata waktu Tunggu



Gambar 6. Komparasi Rata-rata Waktu Tunggu

Dampak positif dari distribusi beban kerja antar-psikolog tersebut terefleksi secara langsung pada metrik kenyamanan pasien, yakni rata-rata waktu tunggu. Pada sistem dengan 1 psikolog, rata-rata waktu tunggu pasien terus mengalami eskalasi seiring bertambahnya jam operasional; merambat naik dari 20,9 menit (pada operasional 3 jam) dan memuncak hingga 67,1 menit (pada operasional 5 jam). Tren kenaikan yang konstan ini merupakan indikator utama dari ketidakstabilan sistem antrean.

Namun, ketika intervensi skenario 2 psikolog diterapkan, rata-rata waktu tunggu berhasil ditekan secara drastis dan menunjukkan kondisi ekuilibrium (steady-state). Meskipun durasi operasional diperpanjang hingga 4 dan 5 jam, rata-rata waktu tunggu pasien cenderung stabil pada kisaran angka 12,1 hingga 12,4 menit. Kestabilan ini sangat krusial secara klinis, karena waktu tunggu yang singkat dan konsisten terbukti efektif dalam menjaga kenyamanan, mereduksi rasa cemas (anxiety), serta meminimalisasi potensi distres psikologis tambahan yang sering dialami pasien saat menunggu giliran konsultasi.

F. Rekapitulasi Hasil Analisis *Cost-Benefit* Skenario Simulasi

	Skenario	Waktu Operasional	Pasien Dilayani	Waktu Tunggu Maksimal	Rata-rata Tunggu	Anggaran Operasional
0	1 Psikolog	3 Jam	3	52.8 Menit	20.9 Menit	Rp 450.000
1	1 Psikolog	4 Jam	4	78.0 Menit	35.1 Menit	Rp 600.000
2	1 Psikolog	5 Jam	6	144.7 Menit	67.1 Menit	Rp 750.000
3	2 Psikolog	3 Jam	7	31.9 Menit	7.1 Menit	Rp 900.000
4	2 Psikolog	4 Jam	9	45.9 Menit	12.1 Menit	Rp 1.200.000
5	2 Psikolog	5 Jam	11	45.9 Menit	12.4 Menit	Rp 1.500.000

Gambar 7. Rekapitulasi Hasil Analisis Cost-Benefit Skenario Simulasi

Berdasarkan data rekapitulasi pada Gambar 7, dapat diamati hasil komparasi metrik kinerja dari enam skenario operasional klinik yang telah disimulasikan. Pada skenario dengan alokasi 1 psikolog, terlihat jelas adanya fenomena penumpukan antrean (*bottleneck*). Meskipun durasi operasional diperpanjang hingga 5 jam, kapasitas sistem hanya mampu melayani 6 pasien. Hal yang paling mengkhawatirkan adalah terjadinya lonjakan eksponensial pada metrik Waktu Tunggu Maksimal yang menembus angka 144,7 menit. Angka ini mengindikasikan risiko medis yang fatal karena telah jauh melampaui batas toleransi kedaruratan psikiatri (30 menit), sehingga berpotensi memicu snowball effect (efek bola salju) dan memperburuk kondisi mental pasien yang sedang mengantre.

Sebaliknya, intervensi dengan mengalokasikan 2 psikolog terbukti memberikan dampak positif yang sangat signifikan terhadap kualitas layanan (benefit). Pada durasi operasional 5 jam, kapasitas pelayanan sistem berhasil menyerap hampir dua kali lipat lebih banyak, yakni 11 pasien. Lebih lanjut, eskalasi Waktu Tunggu Maksimal berhasil dipangkas secara drastis dan distabilkan pada angka 45,9 menit, dengan rata-rata waktu tunggu yang sangat singkat, yaitu 12,4 menit. Penurunan ini membuktikan bahwa pembagian beban kerja (*workload distribution*) antar-psikolog berjalan optimal.

Dari perspektif finansial (cost), sistem menunjukkan bahwa skenario 2 psikolog secara matematis menuntut Anggaran Operasional tepat dua kali lipat lebih besar dibandingkan skenario 1 psikolog pada setiap rentang waktu (sebagai contoh, Rp 1.500.000 berbanding Rp 750.000 untuk operasional 5 jam). Namun demikian, pembengkakan biaya ini dinilai sangat terjustifikasi (justified). Nilai tambah yang didapatkan berupa pencegahan risiko kedaruratan psikologis (pemangkasan waktu tunggu sebesar nyaris 100 menit) dan peningkatan jumlah pasien (throughput) menjadikan skenario 2 psikolog sebagai titik keseimbangan terbaik antara efisiensi anggaran dan standar keselamatan medis klinik.

V. KESIMPULAN DAN SARAN

Penerapan algoritma Gaussian Naïve Bayes terbukti sukses menerjemahkan gejala klinis pasien menjadi parameter waktu pelayanan dinamis bagi mesin Simulasi Kejadian Diskrit dengan tingkat akurasi model sebesar 75,00%. Melalui analisis metrik performa secara komprehensif, penelitian ini menyimpulkan bahwa konfigurasi operasional dengan satu psikolog terbukti memicu penumpukan antrean kritis yang berisiko fatal pada keselamatan mental pasien. Sebaliknya, intervensi dengan dua psikolog tidak hanya mampu meningkatkan kapasitas penyerapan pasien (*throughput*) secara signifikan, tetapi juga sukses menstabilkan rata-rata waktu tunggu sistem pada ekuilibrium yang ideal. Secara spesifik, konfigurasi dua psikolog selama tiga jam operasional teridentifikasi sebagai titik keseimbangan (optimal *trade-off*) yang paling efektif. Solusi strategis ini berhasil menahan eskalasi waktu tunggu maksimal tetap di bawah batas toleransi kedaruratan medis (30 menit), sekaligus mencegah pemborosan finansial klinik akibat idle time tenaga medis. Sebagai saran pengembangan, model simulasi ini sangat potensial untuk diintegrasikan ke dalam arsitektur Sistem Informasi Rumah Sakit (SIRS) berbasis web yang dilengkapi fitur penjadwalan dinamis berdasarkan traffic pasien secara real-time guna meningkatkan pengalaman pengguna dan efektivitas manajemen fasilitas kesehatan di masa depan.

VI. UCAPAN TERIMAKASIH

Penulis menyampaikan penghargaan dan terima kasih yang sebesar-besarnya kepada Ibu Herliana Rosika selaku dosen pembimbing atas arahan, dukungan moral, serta ilmu yang telah diberikan selama proses penelitian hingga penyelesaian makalah ini. Ucapan terima kasih juga ditujukan kepada Program Studi Teknik Informatika Universitas Mataram atas penyediaan fasilitas yang sangat mendukung kelancaran kegiatan simulasi dan komputasi. Penulis juga berterima kasih kepada rekan-rekan sejawat yang

telah memberikan masukan berharga dalam proses diskusi dan penyempurnaan karya tulis ini.

DAFTAR PUSTAKA

[1] A. Nama, “Keterbatasan akses layanan kesehatan mental di Indonesia,” Jurnal Kesehatan Masyarakat, 2025.

[2] A. Kartiani, “The urgent call for clinical psychologists in Indonesia,” Jurnal Psikologi Klinis Indonesia, 2024.

[3] M. A. Smith et al., “While they wait: A cross-sectional survey on wait times for mental health treatment for anxiety and depression for adolescents in Australia,” BMJ Open, vol. 15, no. 3, e087342, 2025.

[4] R. R. Putra dan D. H. Pratama, “Analysis of waiting time and patient anxiety in healthcare services,” Jurnal Kesehatan, 2023.

[5] R. S. Lestari dan M. F. Hakim, “Mental health classification using Naïve Bayes and random forest,” JAIC, vol. 9, no. 4, pp. 1740–1750, 2025.

[6] N. Pakvilai, “Plant macronutrient analysis of enzyme ionic plasma and organic fertilizer from biodegradable waste,” dalam The 2nd Annual South East Asian International Seminar (ASAIS), 2013, hlm. 1–6.

[7] D. G. R. R. dan M. H. S., “Discrete Event Simulation: It’s Easy with SimPy!,” arXiv preprint arXiv:2405.01562, 2024.

[8] A. Putri, “Comparison of Naive Bayes classifier and support vector machine for mental bias classification,” dalam Seminasi/Teknologi Informasi, 2023.

[9] P. Alvarado, “Getting started with SimPy for patient flow modeling,” Bits of Analytics, 2017.

SIMULASI SISTEM PENGAMBIL KEPUTUSAN EVALUASI RISIKO OTOMASI PEKERJAAN MENGGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING*

Rafly Ridho' S.[1)], M. Khalid Al R.[1)], Zainul Majdi.[1)], Yurian Fathur F.[1)], Herliana Rosika.[1)]
[1]Program Studi Teknik Informatika Universitas Mataram
e-mail: www.inya2830@gmail.com

Abstrak

Perkembangan kecerdasan buatan (Artificial Intelligence/AI) telah meningkatkan risiko otomatisasi pada berbagai jenis pekerjaan. Penelitian ini bertujuan mengevaluasi tingkat kerentanan profesi terhadap otomasi AI menggunakan metode Simple Additive Weighting (SAW) sebagai Sistem Pendukung Keputusan (SPK). Penelitian memanfaatkan dua dataset yang terdiri atas 702 profesi dan 500 data pasar kerja dari 10 kategori pekerjaan. Penilaian dilakukan berdasarkan lima kriteria, yaitu probabilitas otomasi (30%), rata-rata gaji tahunan (25%), tingkat pendidikan (20%), jumlah tenaga kerja terpapar (15%), dan tingkat risiko otomasi lintas dataset (10%). Hasil menunjukkan bahwa profesi dengan tingkat kerentanan tertinggi adalah Cashiers dengan nilai preferensi 0,9799, diikuti Combined Food Preparation and Serving Workers, Including Fast Food (0,9073) dan Laborers and Freight, Stock and Material Movers, Hand (0,8120). Hasil penelitian menunjukkan bahwa pekerjaan yang bersifat rutin dengan tingkat pendidikan dan pendapatan relatif rendah lebih rentan terhadap otomasi AI. Metode SAW mampu menghasilkan peringkat profesi secara objektif dan sistematis sebagai dasar perencanaan karier dan kebijakan ketenagakerjaan.
Kata kunci: kecerdasan buatan; otomasi pekerjaan; pasar kerja; sistem pendukung keputusan; *simple additive weighting.*

Abstract

The advancement of Artificial Intelligence (AI) has increased the risk of automation across various occupations. This study aims to evaluate occupational vulnerability to AI-driven automation using the Simple Additive Weighting (SAW) method as a Decision Support System (DSS). Two datasets containing 702 occupations and 500 labor market records from 10 job categories were utilized. Five criteria were considered: automation probability (30%), average annual wage (25%), education level (20%), number of exposed workers (15%), and cross-dataset automation risk level (10%). The results show that Cashiers have the highest vulnerability score (0.9799), followed by Combined Food Preparation and Serving Workers, Including Fast Food (0.9073) and Laborers and Freight, Stock and Material Movers, Hand (0.8120). Occupations characterized by repetitive tasks, relatively low educational requirements, and lower income are found to be more vulnerable to AI-driven automation. The SAW method provides an objective and systematic ranking that can support career planning and labor policy development.
Keywords: *artificial intelligence; job automation; labor market; decision support system; simple additive weighting.*

I. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI) telah menjadi salah satu pendorong transformasi industri dan perekonomian global [1], [2]. Integrasi AI dalam berbagai proses bisnis meningkatkan efisiensi dan produktivitas, sekaligus mengubah struktur pasar tenaga kerja melalui otomatisasi berbagai tugas yang sebelumnya dilakukan manusia [4], [6]. Kondisi ini memengaruhi berbagai sektor industri dan menuntut tenaga kerja untuk beradaptasi terhadap perubahan yang terjadi [7].

Di satu sisi, penerapan AI memberikan manfaat berupa peningkatan produktivitas dan efisiensi operasional. Namun, perkembangan tersebut juga meningkatkan risiko otomasi pada sejumlah profesi, terutama pekerjaan yang

bersifat rutin dan repetitif [4], [5]. Oleh karena itu, identifikasi profesi yang rentan terhadap otomasi menjadi penting sebagai dasar penyusunan kebijakan ketenagakerjaan dan program peningkatan keterampilan (upskilling) [7], [10], [11].

Evaluasi risiko otomasi melibatkan berbagai faktor, seperti probabilitas otomasi, tingkat pendapatan, dan karakteristik pekerjaan, sehingga diperlukan pendekatan yang mampu mempertimbangkan banyak kriteria secara bersamaan [2], [8], [9]. Sistem Pendukung Keputusan (SPK) menjadi salah satu pendekatan yang dapat digunakan untuk melakukan penilaian secara lebih objektif.

Berdasarkan kondisi tersebut, penelitian ini menerapkan metode Simple Additive Weighting (SAW) sebagai Sistem Pendukung Keputusan untuk mengevaluasi tingkat kerentanan profesi terhadap otomasi AI. Metode SAW dipilih karena mampu melakukan penilaian multikriteria secara sederhana dan menghasilkan peringkat alternatif secara objektif [13], [14], [16]. Hasil penelitian diharapkan dapat memberikan gambaran mengenai profesi yang paling rentan terhadap otomasi serta mendukung pengambilan keputusan di era transformasi digital.

II. TINJAUAN PUSTAKA

1. Dinamika Pasar Kerja dan Disrupsi Kecerdasan Buatan (AI)

Perkembangan kecerdasan buatan (Artificial Intelligence/AI) telah mengubah struktur pasar kerja global melalui otomatisasi berbagai tugas yang sebelumnya dilakukan manusia sehingga meningkatkan efisiensi dan produktivitas di berbagai sektor industri [1], [2]. Dampak AI terhadap tenaga kerja dapat dijelaskan melalui efek substitusi, yaitu penggantian pekerjaan yang bersifat rutin dan terstruktur, serta efek komplementaritas, yaitu pemanfaatan AI sebagai alat yang mendukung dan meningkatkan kemampuan manusia [6], [8]. Selain itu, perkembangan AI juga mendorong munculnya kebutuhan keterampilan baru sehingga tenaga kerja dituntut untuk terus meningkatkan kompetensinya agar dapat beradaptasi dengan perubahan kebutuhan pasar kerja di masa depan [7], [10], [11].

2. Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan (SPK) merupakan sistem berbasis komputer yang dirancang untuk membantu proses pengambilan keputusan pada permasalahan semi-terstruktur maupun tidak terstruktur dengan mengintegrasikan data, model, dan metode analisis sehingga menghasilkan keputusan yang lebih objektif, sistematis, dan efektif [12]. Dalam permasalahan yang melibatkan banyak kriteria, SPK berperan penting dalam mengevaluasi berbagai alternatif berdasarkan tingkat kepentingan masing-masing kriteria sehingga keputusan yang dihasilkan menjadi lebih transparan dan dapat dipertanggungjawabkan [13].

3. *Simple Additive Weighting* (SAW)

Metode Simple Additive Weighting (SAW) atau metode penjumlahan terbobot merupakan salah satu metode Multi Attribute Decision Making (MADM) yang digunakan untuk menentukan peringkat alternatif berdasarkan penjumlahan nilai kinerja setiap alternatif terhadap seluruh kriteria yang telah diberi bobot [14], [16]. Metode ini banyak diterapkan pada Sistem Pendukung Keputusan karena proses perhitungannya sederhana, mudah diimplementasikan, serta mampu menghasilkan peringkat alternatif secara objektif [13], [16]. Tahapan metode SAW meliputi penentuan alternatif dan kriteria, pemberian bobot preferensi, pembentukan matriks keputusan, normalisasi nilai berdasarkan atribut benefit dan cost, serta perhitungan nilai preferensi akhir yang digunakan untuk menentukan alternatif terbaik [14], [15], [16]. Dalam penelitian ini, nilai preferensi yang lebih tinggi menunjukkan tingkat kerentanan profesi yang semakin besar terhadap risiko otomasi AI.

III. METODOLOGI PENELITIAN

1. Tahapan Pemodelan dan Alur Simulasi

Penelitian ini menerapkan pendekatan kuantitatif berbasis komputasi untuk mensimulasikan sistem pendukung keputusan dalam mengevaluasi risiko otomasi pekerjaan. Model yang digunakan dibangun menggunakan metode Simple Additive Weighting (SAW) untuk melakukan pembobotan dan pemeringkatan terhadap berbagai profesi berdasarkan tingkat kerentanan mereka terhadap integrasi teknologi Artificial Intelligence (AI).

Secara umum, alur simulasi komputasi yang diimplementasikan dalam penelitian ini ditunjukkan melalui tahapan-tahapan berikut:

a. Data Ingestion & Preprocessing: Menggabungkan dan membersihkan data dari file dataset_cleaned.csv dan job-automation-probability.csv untuk membentuk satu kesatuan data profesi yang valid.
b. Pembentukan Matriks Keputusan (X): Mengekstraksi nilai dari kriteria-kriteria terpilih untuk setiap alternatif (profesi).
c. Normalisasi Matriks Keputusan (R): Mentransformasikan nilai setiap kriteria ke dalam skala [0, 1] berdasarkan sifat kriteria (benefit atau cost).
d. Perkalian dengan Vektor Bobot (W): Menerapkan bobot preferensi yang telah ditentukan untuk masing-masing kriteria.
e. Perhitungan Nilai Preferensi Akhir (V_i): Menghitung skor akhir setiap profesi untuk menghasilkan peringkat kerentanan dari yang tertinggi hingga terendah.

2. Sumber Data dan Variabel Penelitian

Dataset yang digunakan dalam simulasi ini terdiri dari dua sumber utama yang saling melengkapi, mencakup data karakteristik pasar kerja global serta probabilitas otomasi sektoral. Eksplorasi data melibatkan penyesuaian fitur-fitur berikut sebagai basis penentuan keputusan:

a. Alternatif (A_i) Entri profesi/pekerjaan yang dievaluasi (sebanyak m alternatif).
b. Kriteria (C_j): Indikator yang menjadi dasar penilaian tingkat kerentanan otomasi pekerjaan.

3. Penentuan Kriteria, Sifat, dan Pembobotan

Dalam menentukan Indeks Kerentanan Otomasi, setiap kriteria harus dikategorikan berdasarkan sifatnya, yaitu apakah memberikan kontribusi positif terhadap risiko (*benefit* bagi model risiko) atau justru menekan risiko tersebut (*cost* bagi model risiko).

Berdasarkan struktur data pada program simulasi, ditetapkan 4 kriteria utama beserta bobot preferensinya (W) sebagai berikut:

a. C_1: Probabilitas Otomasi
   i. *Sifat:* Benefit (Semakin tinggi nilai probabilitas otomasi suatu profesi, maka semakin tinggi pula tingkat risiko kerentanannya).
   ii. *Bobot (w_1):* 0.40 (40%) — Diberikan bobot terbesar karena merupakan indikator empiris langsung dari paparan teknologi.
b. C_2: Tingkat Gaji Tahunan
   i. *Sifat:* Cost (Secara umum, pekerjaan dengan rata-rata pendapatan lebih rendah memiliki kecenderungan tugas yang repetitif dan lebih rentan digantikan oleh sistem otomatis, sehingga nilai gaji yang tinggi memperkecil tingkat risiko).
   ii. *Bobot (w_2):* 0.20 (20%).
c. C_3: Tingkat Adopsi AI
   i. *Sifat:* Benefit (Semakin masif tingkat adopsi AI pada sektor industri terkait, semakin cepat proses pergeseran tenaga kerja manusia terjadi).
   ii. *Bobot (w_3):* 0.20 (20%).
d. C_4: Proyeksi Pertumbuhan Pekerjaan
   i. *Sifat:* Cost (Profesi yang memproyeksikan pertumbuhan

lapangan kerja yang positif menunjukkan resiliensi yang lebih baik, sehingga mengurangi indeks risiko kerentanan keseluruhan).

ii. *Bobot (w_4):* 0.20 (20%).

Matriks pembobotan yang terbentuk disimbolkan sebagai vektor W:

$$W = [0.40, 0.20, 0.20, 0.20]$$

4. Formulasi Matematis Metode Simple Additive Weighting (SAW)

Metode SAW atau yang dikenal sebagai metode penjumlahan terbobot bekerja dengan mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif untuk semua kriteria. Langkah-langkah penyelesaian matematisnya adalah:

a. Pembentukan Matriks Keputusan (X)

Matriks X berukuran $m \times n$ dibentuk dari m jumlah profesi dan n jumlah kriteria:

$$X = [x_{11}\ x_{12}\ \cdots\ x_{1n}\ x_{21}\ x_{22}\ \cdots\ x_{2n}\ \vdots\ \vdots\ \ddots\ \vdots\ x_{m1}\ x_{m2}\ \cdots\ x_{mn}\ ]$$

b. Normalisasi Matriks Keputusan (R)

Untuk menghindari bias akibat perbedaan satuan antar-kriteria (misalnya presentase probabilitas vs nilai mata uang USD), dilakukan normalisasi elemen matriks x_{ij} menjadi r_{ij} menggunakan persamaan berikut:

i. Untuk Kriteria Keuntungan (*Benefit*):

$$r_{ij} = \frac{x_{ij}}{(x_{ij})}$$

ii. Untuk Kriteria Biaya (*Cost*):

$$r_{ij} = \frac{(x_{ij})}{x_{ij}}$$

Di mana:

i. r_{ij} = nilai rating kinerja yang telah ternormalisasi.

ii. (x_{ij}) = nilai maksimum dari setiap kolom kriteria.

iii. (x_{ij}) = nilai minimum dari setiap kolom kriteria.

c. Perhitungan Nilai Preferensi Akhir (V_i)

Nilai preferensi akhir untuk setiap alternatif (V_i) diperoleh melalui perkalian elemen matriks yang telah ternormalisasi (R) dengan vektor bobot preferensi (W), menggunakan formula:

$$V_i = \sum_{j=1}^{n} w_j \cdot r_{ij}$$

Nilai V_i yang lebih besar menunjukkan bahwa alternatif A_i memiliki nilai preferensi tertinggi, yang dalam konteks simulasi ini diinterpretasikan sebagai profesi dengan tingkat kerentanan tertinggi terhadap otomasi AI.

IV. HASIL DAN PEMBAHASAN

1. Hasil Integrasi dan Prapemrosesan Data

Sebelum proses perhitungan metode Simple Additive Weighting (SAW) dilakukan, dataset probabilitas otomasi (job-automation-probability.csv) yang berisi 702 profesi diintegrasikan dengan dataset pasar kerja (ai_job_market_insights.csv) yang terdiri dari 500 entri pada 10 kategori profesi. Proses integrasi dilakukan menggunakan teknik fuzzy matching dengan bantuan pustaka fuzzywuzzy.

Proses pencocokan dilakukan menggunakan ambang kecocokan (threshold) sebesar 50 agar cakupan pemetaan antarprofesi lebih luas. Dari proses tersebut diperoleh sebanyak 390 profesi yang berhasil dipasangkan dari total 702 profesi pada dataset probabilitas otomasi. Seluruh data hasil integrasi memiliki nilai lengkap tanpa missing value (NaN = 0), sehingga dapat langsung digunakan pada proses perhitungan metode SAW.

Tabel 1 menyajikan ringkasan hasil integrasi data.

Tabel 1. Hasil Integrasi Dataset

Keterangan	Jumlah
Profesi pada dataset probabilitas otomasi	702
Entri pada dataset pasar kerja	500
Ambang batas fuzzy matching	50
Profesi berhasil dicocokkan	390
Nilai NaN setelah integrasi	0

Berdasarkan hasil tersebut, proses integrasi data dapat dikatakan berhasil karena seluruh alternatif yang digunakan pada tahap perhitungan SAW telah memiliki data lengkap tanpa nilai kosong (missing value).

2. Hasil Pembentukan Matriks Keputusan

Setelah proses integrasi selesai, dibentuk matriks keputusan berukuran 390 × 5 yang terdiri atas lima kriteria utama, yaitu:
C1 = Probabilitas Otomasi
C2 = Rata-rata Gaji Tahunan
C3 = Tingkat Pendidikan
C4 = Jumlah Tenaga Kerja Terpapar
C5 = Tingkat Risiko Otomasi Lintas Dataset

Seluruh kriteria kemudian dinormalisasi ke dalam rentang nilai 0–1 agar dapat dibandingkan secara proporsional meskipun memiliki satuan yang berbeda.

Bobot preferensi yang digunakan pada penelitian ini ditunjukkan pada Tabel 2.

Tabel 2. Kriteria, Bobot, dan Sifat

Kriteria	Bobot	Sifat
Probabilitas Otomasi	0.30	Benefit
Gaji Tahunan	0.25	Cost
Tingkat Pendidikan	0.20	Cost
Jumlah Tenaga Kerja	0.15	Benefit
Automation Risk	0.10	Benefit

Probabilitas otomasi (C1) diberikan bobot terbesar karena secara langsung menunjukkan kemungkinan suatu profesi terdampak oleh otomatisasi. Sementara itu, gaji tahunan dan tingkat pendidikan diperlakukan sebagai kriteria cost karena nilai yang lebih tinggi diasumsikan menunjukkan tingkat keamanan profesi yang lebih baik terhadap otomatisasi.

3. Hasil Perhitungan Metode SAW

Nilai preferensi akhir (Vi) diperoleh dari hasil penjumlahan terbobot antara matriks ternormalisasi dengan bobot setiap kriteria (W = [0,30; 0,25; 0,20; 0,15; 0,10]). Hasil perhitungan terhadap 390 alternatif menghasilkan nilai Vi pada rentang 0,1115 hingga 0,9799, dengan rata-rata sebesar 0,4287. Sebanyak 18 profesi (4,6% dari total 390) memperoleh nilai Vi di atas 0,7, yang dalam penelitian ini dikategorikan sebagai kelompok dengan kerentanan paling kritis terhadap otomasi AI.

Nilai preferensi tertinggi diperoleh oleh profesi Cashiers dengan skor 0,9799, diikuti oleh Combined Food Preparation and Serving Workers, Including Fast Food (0,9073) dan Laborers and Freight, Stock and Material Movers, Hand (0,8120).

Sebagian hasil perhitungan (10 peringkat teratas) ditunjukkan pada Tabel 3.

Tabel 3. Hasil Perhitungan SAW (Top 10)

Urut	Profesi	Nilai
1	Cashiers	0.9799
2	Combined Food Preparation and Serving Workers, Including Fast Food	0.9073
3	Laborers and Freight, Stock and Material Movers, Hand	0.8120
4	Dishwashers	0.7954
5	Janitors and Cleaners, Except Maids and Housekeeping Cleaners	0.7808
6	Taxi Drivers and Chauffeurs	0.7686
7	Bakers	0.7660
8	Door-to-Door Sales Workers, News and Street Vendors and Related Workers	0.7638
9	Helpers–Painters, Paperhangers, Plasterers and Stucco Masons	0.7632
10	Food Preparation Workers	0.7562

4. Analisis Ranking Profesi Rentan Terhadap Otomasi AI

Hasil perankingan menunjukkan bahwa Cashiers memperoleh nilai preferensi tertinggi sebesar 0,9799, diikuti oleh Combined Food Preparation and Serving Workers, Including Fast Food (0,9073), Laborers and Freight, Stock and Material Movers, Hand (0,8120), Dishwashers (0,7954), dan Janitors and Cleaners, Except Maids and Housekeeping Cleaners (0,7808). Profesi-profesi tersebut umumnya memiliki tingkat pendidikan yang relatif rendah, probabilitas otomasi yang tinggi, serta jumlah tenaga kerja yang besar sehingga menghasilkan nilai preferensi yang tinggi pada metode SAW.

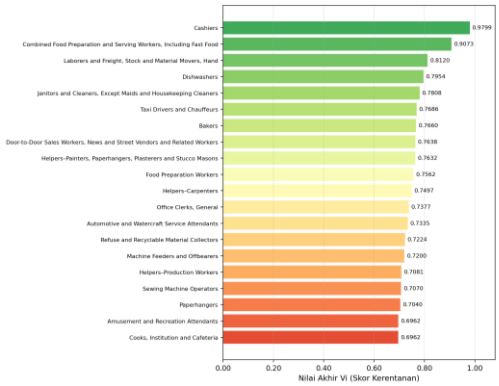
Selain itu, hasil penelitian menunjukkan bahwa label risiko kategorikal pada dataset pasar kerja tidak selalu sejalan dengan peringkat akhir yang diperoleh. Hal ini menunjukkan bahwa pendekatan multikriteria menggunakan metode SAW mampu memberikan hasil yang lebih komprehensif dibandingkan hanya menggunakan satu indikator risiko. Dengan demikian, tingkat

kerentanan profesi terhadap otomasi AI dipengaruhi oleh kombinasi berbagai kriteria yang digunakan dalam proses pengambilan keputusan.

Secara keseluruhan, hasil analisis menunjukkan bahwa pekerjaan yang bersifat rutin, membutuhkan keterampilan dasar, dan melibatkan jumlah tenaga kerja yang besar cenderung memiliki tingkat kerentanan yang lebih tinggi terhadap perkembangan teknologi AI.

5. Visualisasi Hasil Ranking

Hasil pemeringkatan kemudian divisualisasikan dalam bentuk diagram batang untuk memperlihatkan perbandingan nilai preferensi antar profesi.



Gambar 1. Top 20 Profesi

Berdasarkan Gambar 1, profesi Cashiers memiliki skor kerentanan tertinggi sebesar 0,9799, diikuti oleh Combined Food Preparation and Serving Workers, Including Fast Food dengan nilai 0,9073 dan Laborers and Freight, Stock and Material Movers, Hand sebesar 0,8120. Secara umum, profesi yang berada pada peringkat atas didominasi oleh pekerjaan di sektor layanan dan pekerjaan manual yang memiliki karakteristik tugas yang berulang serta tingkat pendidikan yang relatif rendah.

6. Pembahasan

Hasil penelitian menunjukkan bahwa metode Simple Additive Weighting (SAW) dapat digunakan sebagai sistem pendukung keputusan untuk mengevaluasi tingkat kerentanan profesi terhadap risiko otomasi AI. Melalui proses normalisasi dan pembobotan lima kriteria, yaitu probabilitas otomasi, rata-rata gaji tahunan, tingkat pendidikan, jumlah tenaga kerja terpapar, serta tingkat risiko otomasi, metode SAW mampu menghasilkan nilai preferensi yang dapat digunakan untuk menentukan peringkat profesi secara objektif.

Hasil simulasi menunjukkan bahwa profesi dengan probabilitas otomasi tinggi, tingkat pendidikan yang relatif rendah, dan jumlah tenaga kerja yang besar cenderung memperoleh nilai preferensi yang lebih tinggi. Profesi seperti Cashiers, Combined Food Preparation and Serving Workers, Including Fast Food, serta Laborers and Freight, Stock and Material Movers, Hand secara konsisten menempati peringkat teratas karena memiliki karakteristik pekerjaan yang bersifat rutin dan lebih mudah diotomatisasi.

Selain itu, besarnya jumlah tenaga kerja pada profesi-profesi tersebut menunjukkan bahwa dampak otomasi AI tidak hanya berpengaruh terhadap jenis pekerjaan tertentu, tetapi juga berpotensi memengaruhi populasi pekerja dalam jumlah yang besar. Dengan demikian, informasi yang dihasilkan dari penelitian ini dapat dimanfaatkan sebagai bahan pertimbangan dalam perencanaan karier, pengembangan keterampilan (upskilling), serta penyusunan kebijakan ketenagakerjaan dalam menghadapi transformasi digital berbasis kecerdasan buatan.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah Berdasarkan hasil penelitian yang telah dilakukan, metode Simple Additive Weighting (SAW) berhasil diterapkan sebagai Sistem Pendukung Keputusan untuk mengevaluasi tingkat kerentanan profesi terhadap risiko otomasi berbasis Artificial Intelligence (AI). Dengan mempertimbangkan lima kriteria, yaitu probabilitas otomasi, rata-rata gaji tahunan, tingkat pendidikan, jumlah tenaga kerja, dan tingkat risiko otomasi, metode SAW mampu menghasilkan peringkat profesi secara objektif. Hasil penelitian menunjukkan bahwa profesi Cashiers memiliki tingkat kerentanan tertinggi dengan nilai preferensi sebesar 0,9799, diikuti oleh Combined Food Preparation and Serving Workers, Including Fast Food sebesar 0,9073,

serta Laborers and Freight, Stock and Material Movers, Hand sebesar 0,8120. Temuan ini menunjukkan bahwa pekerjaan dengan karakteristik tugas yang repetitif, tingkat pendidikan rendah, serta jumlah tenaga kerja yang besar memiliki risiko lebih tinggi terhadap otomasi AI. Untuk penelitian selanjutnya, disarankan penggunaan dataset yang lebih luas dan mutakhir serta perbandingan dengan metode pengambilan keputusan multikriteria lainnya guna meningkatkan kualitas hasil analisis.

DAFTAR PUSTAKA

[1] H. Kausar, et al, "AI-Powered Job Market Insights to Predict Future Demand for Skills Across Various Job Markets," International Journal of Engineering Research and Science & Technology, vol. 21, no. 2, 2025.

[2] L. B. Solo, dkk, "AI-Powered Job Market Insights: How AI Adoption Influences Salary and Job Growth Projections," North American Academic Research, vol. 8, no. 2, pp. 224–241, 2025.

[3] A. VeeraBabu, dkk, "Skillcast AI: Predictive Model for Forecasting Future Job Market Demands," International Journal of Engineering Research and Science & Technology, vol. 21, no. 3, 2025.

[4] E. Brynjolfsson, dkk, "Generative AI at Work," National Bureau of Economic Research (NBER) Working Paper, 2023.

[5] E. W. Felten, dkk, "Occupational, Industry, and Geographic Exposure to Artificial Intelligence: A Novel Dataset and Its Potential Uses," Strategic Management Journal, vol. 42, no. 12, pp. 2195–2217, 2021.

[6] D. Acemoglu dkk, "Automation and New Tasks: How Technology Displaces and Reinstates Labor," Journal of Economic Perspectives, vol. 33, no. 2, pp. 3–30, 2019.

[7] World Economic Forum, Future of Jobs Report 2025. Geneva, Switzerland: World Economic Forum, 2025.

[8] D. H. Autor, "Why Are There Still So Many Jobs? The History and Future of Workplace Automation," Journal of Economic Perspectives, vol. 29, no. 3, pp. 3–30, 2015.

[9] D. M. Popa, dkk, "Generative-AI and the Transformation of Workforce: A Job Postings-Driven Analysis," arXiv preprint arXiv:2605.00843, 2026.

[10] F. Stephany, dkk, "AI Skills Improve Job Prospects: Causal Evidence from a Hiring Experiment," arXiv preprint arXiv:2601.13286, 2026.

[11] H. Chen, et al, "Artificial Intelligence and Skills: Evidence from Contrastive Learning in Online Job Vacancies," arXiv preprint arXiv:2601.03558, 2026.

[12] E. Turban, dkk, Decision Support and Business Intelligence Systems, 10th ed. Boston, MA, USA: Pearson, 2017.

[13] C. I. Agustyaningrum, dkk, "Decision Support System Application with Simple Additive Weighting," in Proc. Int. Conf. Industrial Engineering and Operations Management, Monterrey, Mexico, 2021.

[14] U. I. Arsyah, dkk, "Analysis of the Simple Additive Weighting Method in Educational Aid Decision Making," Turkish Journal of Computer and Mathematics Education, vol. 12, no. 14, pp. 2389–2396, 2021.

[15] L. Pasaribu dkk, "Decision Support System to Determine the Amount of Performance Allowances Using the Simple Additive Weighting Method," Journal of Intelligent Decision Support System, vol. 3, no. 3, 2020.

[16] H. Taherdoost, "Analysis of Simple Additive Weighting Method (SAW) as a Multi-Attribute Decision-Making Technique: A Step-by-Step Guide," Journal of Management Science & Engineering Research, vol. 6, no. 1, 2023.

[17] Andrianus dkk, "Boarding House Decision Support System with Simple Additive Weighting," Journal of System and Management Sciences, vol. 14, no. 2, pp. 90–99, 2024.

[18] A. Sulistyohati dkk, "Automation of Credit Approval Eligibility Using the Simple Additive Weighting Method at Financial Institutions," SAINTEKBU: Journal of Science and Technology, vol. 17, no. 2, 2025.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

PARALLEL GENETIC ALGORITHM-BASED HYPERPARAMETER OPTIMIZATION FOR INDOBERT-LORA IN INDONESIAN STOCK NEWS SENTIMENT ANALYSIS

I Kadek Mahesa Permana Putra[1)]
Department of Informatics Engineering, University of Mataram
Corresponding author e-mail: mahesamp19@gmail.com

Abstrak

Analisis sentimen berita saham Indonesia memerlukan pipeline pemrosesan bahasa alami yang akurat dan efisien, serta disesuaikan dengan domain keuangan. Kontribusi utama penelitian ini adalah pengusulan kerangka Parallel Genetic Algorithm (PGA) berbasis PySpark untuk mengoptimasi hyperparameter IndoBERT-LoRA, yang dievaluasi pada 9.819 sampel judul berita dari CNBC Indonesia dengan tiga kelas sentimen: positif, negatif, dan netral. Ruang pencarian mencakup 6 dimensi hyperparameter, meliputi parameter pelatihan standar dan parameter spesifik LoRA, sehingga menghasilkan 972 kemungkinan konfigurasi. Untuk mengurangi biaya evaluasi fitness, strategi evaluasi proxy fidelitas rendah diterapkan, di mana setiap kandidat dievaluasi pada subset data pelatihan 30% yang diambil secara stratifikasi selama satu epoch, sementara hanya tiga kandidat terbaik yang dilatih ulang secara penuh pada keseluruhan data. Eksperimen membandingkan tiga konfigurasi: sequential GA, parallel GA, dan surrogate-assisted parallel GA. Hasil menunjukkan bahwa optimasi berbasis GA meningkatkan weighted F1-score dari 0,6831 menjadi 0,8254 (+20,8%) dibandingkan dengan baseline tanpa optimasi. Namun, paralelisasi PySpark saja meningkatkan waktu eksekusi sebesar 19,2% pada lingkungan single-GPU akibat overhead penjadwalan. Pipeline surrogate-assisted berhasil mengurangi total waktu eksekusi sebesar 6,4% dibandingkan sequential GA, sambil tetap mempertahankan performa model terbaik sebesar 95,4%. Temuan ini menegaskan bahwa evaluasi berbasis surrogate merupakan kunci efisiensi dalam kerangka ini dan bahwa lingkungan multi-device mungkin diperlukan agar paralelisasi PySpark dapat memberikan percepatan yang berarti pada workload fine-tuning model Transformer.

Kata kunci: algoritma genetika; analisis sentimen; IndoBERT-LoRA; optimasi hyperparameter; PySpark

Abstract

Performing sentiment analysis of Indonesian stock news requires an accurate, efficient natural language processing pipeline designed for the financial sector. This study describes a PySpark-based Parallel Genetic Algorithm (PGA) framework created for hyperparameter optimization of IndoBERT-LoRA. The framework is evaluated using 9,819 news headlines from CNBC Indonesia, categorized into three sentiment classes: positive, negative, and neutral. The search space spans six hyperparameter dimensions, including standard training and LoRA-specific parameters, yielding 972 possible configurations. To reduce fitness evaluation costs, a low-fidelity proxy evaluation strategy is used. Each candidate is evaluated on a stratified 30% subset of the training data for one epoch, and only the top three candidates are fully retrained on the full dataset. Experiments compare three configurations: sequential GA, parallel GA, and surrogate-assisted parallel GA. Results show that GA-based optimization improves the weighted F1-score from 0.6831 to 0.8254 (+20.8%) relative to the unoptimized baseline. However, PySpark parallelization alone increases execution time by 19.2% in a single-GPU environment because of scheduling overhead. The surrogate-assisted pipeline reduces total execution time by 6.4% relative to sequential GA while retaining 95.4% of the best model's performance. These outcomes indicate that surrogate-based evaluation is the primary driver of efficiency in this system, and that a multi-device environment may be necessary for PySpark parallelization to yield meaningful speedups in transformer model fine-tuning workloads.

Keyword: *genetic algorithm; hyperparameter optimization; IndoBERT-LoRA; PySpark; sentiment analysis*

I. INTRODUCTION

Stock market patterns in Indonesia are strongly influenced by financial news flow. Trader sentiment, which is determined by news coverage, has been shown to affect stock returns in the Indonesian market [2]. At a wider scale, large-scale analysis of Indonesian financial news has demonstrated that sentiment extracted from news articles correlates with stock price movements across sectors and thematic clusters [1]. This points to the need for automated sentiment analysis systems that are accurate and specifically adapted to the Indonesian financial domain.

Natural Language Processing (NLP) has enabled scalable approaches to

sentiment analysis, but Indonesian brings several challenges. The language displays rich morphological variation, extensive affixation, domain-specific financial terminology, and high lexical diversity across news sources [3], [15]. These characteristics make general-purpose multilingual models less optimal for Indonesian financial text, motivating the development of language-specific pre-trained models. The release of IndoNLU [3] and the IndoBERT model (indobenchmark/indobert-base-p1) offered a solid base for Indonesian NLP research. However, fine-tuning IndoBERT for downstream tasks such as sentiment classification requires updating many parameters, which can be computationally costly in resource-limited settings.

Low-Rank Adaptation (LoRA) [4] offers an effective alternative by introducing trainable low-rank matrices into frozen transformer layers. This approach greatly decreases the number of trainable parameters while preserving competitive performance [9]. Although LoRA improves efficiency, its performance remains highly sensitive to hyperparameter configuration, including rank, scaling factor, dropout rate, learning rate, and batch size [9]. The resulting search space becomes large and non-convex, making exhaustive methods such as Grid Search impractical and Random Search insufficient for consistently identifying high-performing configurations [5], [12].

Genetic Algorithm (GA) provides a structured evolutionary search strategy capable of navigating high-dimensional hyperparameter spaces through selection, crossover, and mutation [5], [6]. However, the main computational bottleneck lies in fitness evaluation, where each candidate configuration requires full model training. Sequential evaluation becomes prohibitively expensive when applied to large transformer-based models such as IndoBERT-LoRA [8]. To overcome this limitation, fitness evaluation can be parallelized using a distributed computing framework such as Apache Spark, facilitating concurrent evaluation of multiple candidate solutions during each generation [7], [8].

Despite growing interest in parallel evolutionary optimization and hyperparameter tuning for deep learning models, no prior work has combined Parallel Genetic Algorithms with LoRA-based fine-tuning for Indonesian-language models in financial sentiment analysis.

This study proposes a PySpark-based Parallel Genetic Algorithm (PGA) framework for hyperparameter optimization of IndoBERT-LoRA for Indonesian stock news sentiment classification. To cut computational cost during the search process, we employ a low-fidelity proxy evaluation strategy: each candidate is evaluated on a 30% training subset for one epoch; only the top-3 candidates from the final generation are fully retrained on the complete dataset for three epochs [10], [11].

The main contributions of this work are as follows: (1) a reproducible PySpark-based parallel genetic algorithm framework for IndoBERT-LoRA hyperparameter optimization in Indonesian financial sentiment analysis; (2) the integration of a low-fidelity proxy evaluation strategy to cut computational cost during GA-based hyperparameter search; and (3) an empirical comparison of sequential and parallel genetic algorithms in terms of model effectiveness and runtime efficiency in a single-GPU, resource-constrained environment.

II. RELATED WORK

Research regarding sentiment analysis of Indonesian financial texts has been relatively limited, though recent studies show the potential of AI-based approaches in this domain. Alamsyah et al. [1] showed that news sentiment correlates with stock price movements across Indonesian industries, while Zainul et al. [2] confirmed the influence of trader sentiment on stock returns in the Indonesian stock market. On the NLP side,

Wilie et al. [3] introduced IndoNLU and IndoBERT as benchmark resources for Indonesian language understanding, which were later complemented by Koto et al. [15] through IndoLEM and an independently trained IndoBERT variant. Together, these works establish the foundation for Indonesian-language fine-tuning research.

Parameter-efficient fine-tuning methods, particularly Low-Rank Adaptation (LoRA) [4], have become widely adopted for adapting large language models under computing limitations. Biderman et al. [9] conducted a systematic analysis of LoRA hyperparameter sensitivity and showed that key parameters, such as rank, scaling factor (α), and learning rate, interact in complicated ways, with suboptimal configurations markedly impairing performance. These findings show the importance of structured hyperparameter optimization strategies rather than manual adjustment.

The Genetic Algorithm (GA) has been widely used for hyperparameter optimization in deep learning models due to its ability to explore multi-dimensional, non-convex search spaces. El-Hassani et al. [5] proposed a real-coded GA for multilayer perceptron optimization, while Lee et al. [6] demonstrated a GA-based architecture and hyperparameter search for neural networks. In the context of sentiment analysis, Darmawan et al. [17] applied GA to optimize SVM parameters for Indonesian social media sentiment classification, providing an early example of evolutionary optimization in Indonesian NLP tasks.

The scalability limitations of sequential GA have motivated research on parallel and distributed implementations. Lu et al. [7] proposed a parallel GA framework using Hadoop and Spark, showing improved execution speed on large-scale optimization problems. Wu et al. [8] developed a parallel GA-based AutoML framework for deep learning hyperparameter optimization, reporting significant reductions in training time while preserving competitive performance. Muhammadiyeva et al. [16] further showed the feasibility of implementing a parallel GA in Apache Spark MLlib for classification tasks, particularly via distributed fitness evaluation.

Despite these advances, existing studies have not investigated the combination of Parallel Genetic Algorithms with LoRA-based fine-tuning for Indonesian financial sentiment analysis. This gap motivates the proposed study.

III. METHODOLOGY

A. Dataset

This study uses the CNBC Indonesia Stock News Sentiment dataset, obtained from Kaggle, which consists of 9,819 labeled headline samples collected from January 2024 to Q1 2025. The dataset contains three sentiment classes: neutral (4,356 samples), positive (2,887 samples), and negative (2,576 samples). The dataset was split into training (7,855), validation (982), and test (982) sets using a stratified 80/10/10 ratio to preserve class distribution across splits.

B. Text Processing

Prior to tokenization, text cleaning was performed by removing URLs, special characters, numbers, and redundant whitespace. The cleaned headlines were then tokenized using the IndoBERT tokenizer with a maximum sequence length of 128 tokens. Inputs shorter than the maximum length were padded, while longer sequences were truncated.

C. Model Architecture

The base model used in this study is *indobenchmark/indobert-base-p1* [3], a BERT-based pre-trained language model for Indonesian. A classification head with three output classes was appended to the [CLS] token representation. Fine-tuning was performed using LoRA [4] via the PEFT library, applied to the query and value projection matrices of the self-attention layers while keeping remaining parameters frozen. The rank, alpha, and dropout parameters were included in the hyperparameter optimization process. All experiments were conducted on Google Colab with an NVIDIA Tesla T4 GPU.

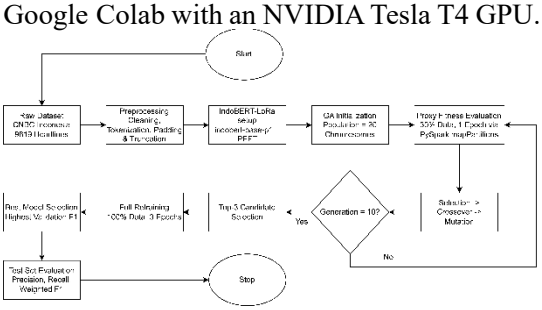


Figure 1. Model Architecture

D. Hyperparameter Search Space

The search space spans six dimensions, covering both standard training

and LoRA-specific parameters, and is defined based on recommendations from the BERT and LoRA literature [4], [5], [6], as listed in **Table 1**. The total number of possible configurations is $4 \times 3^5 = 972$. Evaluating all configurations exhaustively would require considerable computational resources, motivating the use of GA-based search.

Table 1. Hyperparameter Search Space

Hyperparameter	Candidate Values
Learning Rate	1e−5, 2e−5, 3e−5, 5e−5
Batch Size	8, 16, 32
LoRa Rank	4, 8, 16
LoRa Alpha	8, 16, 32
LoRa Dropout	0.05, 0.1, 0.2
Weight Decay	0.0, 0.01, 0.05

E. Genetic Algorithm Configuration

Each individual in the GA population represents a candidate hyperparameter configuration encoded as a fixed-length chromosome, with each gene corresponding to a hyperparameter dimension. The GA search proceeds through tournament selection, uniform crossover, and random-replacement mutation throughout generations. For the sequential GA, candidates are evaluated serially within a single process. For the parallel GA, candidate configurations were converted into Spark RDDs and distributed across worker partitions, with fitness evaluation performed via mapPartitions(), allowing evaluation tasks to be distributed across Spark workers while sharing the same computational environment. The GA configuration is summarized in **Table 2**.

Table 2. Genetic Algorithm Configuration

Parameter	Value
Population Size	20
Number of Generations	10
Selection Method	Tournament
Crossover Type	Uniform
Mutation Type	Random replacement
Mutation Probability	0.20
Fitness Function	Weighted F1-score

F. Proxy Fitness Evaluation Strategy

To cut computational cost during the GA search phase, a low-fidelity proxy evaluation strategy is adopted [10], [11]. During the search phase, all candidate configurations were trained for one epoch on a 30% stratified subset of the training data, and the weighted F1-score on the validation set was used as the fitness value. This approach is motivated by findings that single-epoch performance strongly predicts final model ranking [10], [11], enabling the early elimination of low-performing candidates without full training. After the search process, the top-3 candidates were retrained on the full training set for 3 epochs to obtain the final model, and the best-performing candidate on the validation set was selected as the final configuration. Iskoko et al. [12] adopted a comparable evaluation protocol as a reference baseline.

G. Evaluation Metrics

Model performance is evaluated using weighted precision, recall, and F1-score on the held-out test set. Runtime efficiency is compared between sequential and parallel GA using speedup S and time reduction R:

$$S = \frac{T_{\text{seq}}}{T_{\text{par}}}, \quad R(\%) = \frac{T_{\text{seq}} - T_{\text{par}}}{T_{\text{seq}}} \times 100$$

Parallel efficiency is additionally reported as $E = S/N_{\text{workers}}$. Both experiments were executed in separate notebook environments on identical hardware to ensure a fair comparison.

IV. RESULTS AND DISCUSSION

A. Baseline Performance

The baseline IndoBERT-LoRA model, trained with default hyperparameters and without optimization, achieved a weighted F1-score of 0.6831. This result indicates that, while the model is functional, the default settings are insufficient for reliable sentiment classification in Indonesian financial news, and structured hyperparameter optimization is warranted.

B. Hyperparameter Optimization Results

Table 3 summarizes the performance of all optimization methods relative to the baseline. The best configuration, identified by the sequential GA, achieved a weighted F1-score of 0.8254, representing an absolute improvement of 0.1423 and a relative improvement of approximately 20.8% over the baseline. All three optimization methods substantially outperform the baseline, confirming that GA-based hyperparameter search yields meaningful gains in fine-tuning IndoBERT-LoRA on this domain.

Table 3. Methods Performance

Method	Weighted F1	Δ F1	Relative Improvement
Baseline	0.6831	-	-
Sequential GA	0.8254	+0.1423	+20.8%
Parallel GA	0.8179	+0.1348	+19.7%
Parallel GA + Surrogate	0.7874	+0.1043	+15.3%

C. Best Parameter Configuration

Table 4 presents the best hyperparameter configuration identified by the sequential GA. The selected configuration favors a high LoRA rank (r = 16) with a correspondingly high alpha (α = 32), consistent with the recommendation that α = 2r tends to produce strong performance [9]. A low dropout rate (0.05) and zero weight decay were also selected, suggesting the model benefits from less regularization constraint on this relatively small financial news dataset.

Table 4. Best Parameter Configuration

Hyperparameter	Best Values
Learning Rate	5e−5
Batch Size	8
LoRa Rank	16
LoRa Alpha	32
LoRa Dropout	0.05
Weight Decay	0.0

D. Sequential vs. Parallel GA Runtime Comparison

Table 5 compares the total runtime and performance of sequential and parallel GA under identical experimental conditions.

Table 5. Runtime Comparison

Metric	Sequential GA	Parallel GA
Total Runtime (s)	1309.14	1559.89
Weighted F1	0.8254	0.8179
Speedup (S)	1.00	0.84

The parallel GA did not achieve a speedup over the sequential baseline, yielding $S = 0.84$, and its runtime increased by 19.2% relative to the sequential GA. This is attributable to the single-GPU execution environment on Google Colab, where all IndoBERT training processes share the same Tesla T4 GPU regardless of Spark worker configuration. PySpark's task scheduling and inter-process communication introduced overhead that exceeded any parallelization benefit under this constraint, consistent with observations in the distributed GA literature that GPU-bound workloads represent a fundamental bottleneck for Spark-based parallelization in single-device environments [13], [14].

Although the parallel implementation did not outperform the sequential baseline in this environment, the experiment provides practical evidence that Spark-based parallelization alone is insufficient to accelerate transformer fine-tuning workloads when GPU resources remain the dominant bottleneck. This observation motivates the integration of surrogate-assisted evaluation, which proved substantially more effective for reducing overall optimization cost.

E. Effect of Surrogate-Assisted Evaluation

To isolate the contribution of proxy fitness evaluation, Table 6 reports runtime and performance across all three methods.

Table 6. Surrogate-Assisted Evaluation

Method	Total Runtime (s)	Weighted F1	Speed up vs Seq.	Runtime Change
Sequential GA	1309.14	0.8254	1.00x	-
Parallel GA	1559.89	0.8179	0.84x	+19.2% longer
Parallel GA + Surrogate	1225.19	0.7874	1.07x	−6.4%

The surrogate-assisted parallel GA completed in 1225.19s, compared to 1309.14s for the sequential baseline, yielding a speedup of $S = 1.07\times$ and a runtime reduction of 6.4%. These gains came at an absolute F1 cost of 0.0380 (0.8254 → 0.7874), retaining 95.4% of the best weighted F1-score achieved by the sequential GA.

Although the overall speedup is modest, the result shows that surrogate-assisted evaluation is the primary driver of computational effectiveness in this pipeline. Without proxy evaluation, the parallel GA had

a runtime 19.2% longer than the sequential baseline due to Spark scheduling overhead. The introduction of proxy fitness evaluation partially mitigated this overhead, enabling the surrogate-assisted pipeline to achieve a modest runtime improvement over the sequential baseline. This is consistent with findings from the early-discarding literature [10], [11], which establish that single-epoch proxy evaluations can reliably identify high-performing candidate regions while substantially reducing total evaluation cost.

Regarding proxy ranking quality, the sequential GA experiment showed complete agreement between proxy and full-training rankings. Although minor ranking shifts were observed in the parallel experiment, the surrogate model still identified competitive candidate regions and substantially reduced the number of expensive GPU evaluations. Taken together, these results suggest that the main contribution of this work lies in combining distributed task management and surrogate-assisted evaluation as a practical strategy for efficient hyperparameter optimization under resource-constrained conditions, rather than in PySpark parallelization alone.

V. CONCLUSION AND SUGGESTION

This study introduced a PySpark-based parallel genetic algorithm framework for hyperparameter optimization of IndoBERT-LoRA, which is used to classify Indonesian stock news sentiment. Three key findings emerge from the experiments. First, GA-based hyperparameter optimization substantially improved model performance. The sequential GA identified a configuration that achieved a weighted F1-score of 0.8254, representing an absolute improvement of 0.1423 (+20.8%) over the unoptimized baseline of 0.6831. This confirms that structured hyperparameter search is effective for improving the performance of LoRA-adapted language models on domain-specific Indonesian NLP tasks. Second, PySpark parallelization alone did not yield runtime improvements under a single-GPU environment. The parallel GA incurred a 19.2% longer runtime than the sequential baseline (S = 0.84), attributable to Spark scheduling overhead that competes for the same GPU resource. This result suggests that distributed task management frameworks may require multi-device environments to deliver meaningful speedup for transformer fine-tuning workloads. Third, surrogate-assisted evaluation proved to be the more effective strategy for reducing optimization cost. The combination of proxy fitness evaluation and parallel GA reduced total runtime by 6.4% (S = 1.07×) relative to sequential GA while retaining 95.4% of the best weighted F1-score. Although the runtime improvement is modest, the results show that low-fidelity evaluation strategies can effectively reduce computational cost while preserving competitive classification performance under resource-constrained conditions. Overall, the results indicate that the main contribution of this work is demonstrating the practicality of surrogate-assisted hyperparameter optimization for IndoBERT-LoRA, rather than the direct acceleration achieved through PySpark parallelization alone. Future work should explore multi-GPU or multi-node environments where PySpark parallelization can run without GPU contention. Additionally, more sophisticated surrogate models, such as Gaussian Process Regression or neural surrogate estimators, may further improve the trade-off between optimization cost and final model performance.

REFERENCES

[1] A. Alamsyah, D. P. Ramadhani, F. T. Kristanti, and A. H. Nasution, "Deciphering news sentiment and stock price relationships in Indonesian companies: An AI-based exploration of industry affiliation and news co-occurrence," *Discover Artif. Intell.*, vol. 5, p. 87, 2025. doi: 10.1007/s44163-025-00350-5.

[2] Z. R. Zainul, K. A. Fachrudin, N. Irawati, and Syahyunan, "The influence of investor sentiment on returns in the Indonesian stock market," in *Proc. 8th Int. Research Conf. Economic and Business (IRCEB 2024)*, Atlantis Press, 2025, pp. 167–175. doi: 10.2991/978-94-6463-722-9_15.

[3] B. Wilie et al., "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," in *Proc. 1st Conf. Asia-Pacific Chapter ACL and 10th Int. Joint Conf. NLP (AACL-IJCNLP 2020)*, Suzhou, China, Dec. 2020, pp. 843–857. doi: 10.18653/v1/2020.aacl-main.85.

[4] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. 10th Int. Conf. Learning Representations (ICLR 2022)*, 2022. doi: 10.48550/arXiv.2106.09685.

[5] F. Z. El-Hassani, M. Amri, N.-E. Joudar, and K. Haddouch, "A new optimization model for MLP hyperparameter tuning: Modeling and resolution by real-coded genetic algorithm," *Neural Process. Lett.*, vol. 56, no. 2, p. 105, Mar. 2024. doi: 10.1007/s11063-024-11578-0.

[6] S. Lee, J. Kim, H. Kang, D.-Y. Kang, and J. Park, "Genetic algorithm-based deep learning neural network structure and hyperparameter optimization," *Appl. Sci.*, vol. 11, no. 2, p. 744, Jan. 2021. doi: 10.3390/app11020744.

[7] H.-C. Lu, F. J. Hwang, and Y.-H. Huang, "Parallel and distributed architecture of genetic algorithm on Apache Hadoop and Spark," *Appl. Soft Comput.*, vol. 95, p. 106497, Oct. 2020. doi: 10.1016/j.asoc.2020.106497.

[8] D. Wu, Q. Guan, Z. Fan, H. Deng, and T. Wu, "AutoML with parallel genetic algorithm for fast hyperparameters optimization in efficient IoT time series prediction," *IEEE Trans. Ind. Inform.*, vol. 19, no. 9, pp. 9555–9564, Sep. 2023. doi: 10.1109/TII.2022.3232794.

[9] D. Biderman et al., "LoRA learns less and forgets less," *Trans. Mach. Learn. Res.*, 2024. doi: 10.48550/arXiv.2405.09673.

[10] R. Egele, F. Mohr, T. Viering, and P. Balaprakash, "The unreasonable effectiveness of early discarding after one epoch in neural network hyperparameter optimization," *Neurocomputing*, vol. 597, p. 127964, 2024. doi: 10.1016/j.neucom.2024.127964.

[11] R. Egele, I. Guyon, Y. Sun, and P. Balaprakash, "Is one epoch all you need for multi-fidelity hyperparameter optimization?" in *Proc. ESANN 2023*, Bruges, Belgium, Oct. 2023. doi: 10.48550/arXiv.2307.15422.

[12] A. Iskoko, I. Tahyudin, and P. Purwadi, "Hyperparameter optimization of IndoBERT using grid search, random search, and Bayesian optimization in sentiment analysis of e-government application reviews," *J. Tek. Inform. (JUTIF)*, vol. 6, no. 5, pp. 3430–3444, Oct. 2025. doi: 10.52436/1.jutif.2025.6.5.4897.

[13] L. Al-Terkawi and M. Migliavacca, "An automated parallel genetic algorithm with parametric adaptation for distributed data analysis," *Sci. Rep.*, vol. 15, p. 10836, Mar. 2025. doi: 10.1038/s41598-025-93943-0.

[14] L. Al-Terkawi, "Survey on distributed parallel genetic algorithms for large-scale data analysis," *Computing*, 2025. doi: 10.1007/s00607-025-01596-8.

[15] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP," in *Proc. 28th Int. Conf. Computational Linguistics (COLING 2020)*, Barcelona, Spain, Dec. 2020, pp. 757–770. doi: 10.18653/v1/2020.coling-main.66.

[16] N. Muhammadiyeva, A. Xakimov, and J. Yan, "Parallel genetic algorithm–enhanced hyperparameter and feature optimization for large-scale classification in Apache Spark MLlib," *Univ. J. Acad. Multidiscip. Res.*, vol. 3, no. 30, pp. 103–106, 2025.

[17] I. P. D. W. Darmawan, G. A. Pradnyana, and I. B. N. Pascima, "Optimasi parameter support vector machine dengan algoritma genetika untuk analisis sentimen pada media sosial Instagram," *SINTECH J.*, vol. 6, no. 1, pp. 58–67, Apr. 2023. doi: 10.31598/sintechjournal.v6i1.1245.

PERAMALAN DAN ANALISIS FREKUENSI, SEVERITAS, DAN TOTAL KLAIM ASURANSI KESEHATAN AXA MELALUI PENDEKATAN TIME SERIES

Fadlullah Hasan [1)], I Nyoman Widiyasa Jayananda [1)], Lalu Akhadi Rizki. [1)]
1) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: f1d02310008@student.unram.ac.id

Abstrak

Peramalan klaim asuransi kesehatan berperan penting dalam pengelolaan risiko, perencanaan keuangan, dan penetapan strategi bisnis perusahaan asuransi. Namun, ketidakpastian pada frekuensi dan besaran biaya klaim menuntut model prediksi yang mampu menangkap pola historis secara akurat. Penelitian ini bertujuan menganalisis dan memprediksi frekuensi klaim, severitas klaim, dan total biaya klaim menggunakan pendekatan peramalan deret waktu. Data klaim dan polis diolah melalui tahapan preprocessing, penanganan missing values, feature engineering, dan agregasi menjadi deret waktu bulanan. Untuk meningkatkan kemampuan prediksi, penelitian ini mengintegrasikan variabel eksternal berupa Indeks Harga Konsumen sektor kesehatan, tingkat inflasi, pengeluaran kesehatan per kapita, dan jumlah hari libur nasional yang merepresentasikan perubahan biaya layanan kesehatan, kondisi ekonomi, serta pola pemanfaatan layanan kesehatan masyarakat. Tiga metode peramalan, yaitu Naive Seasonal, AutoETS, dan Exponential Smoothing, dibandingkan menggunakan rolling window cross-validation dengan metrik evaluasi Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), dan Symmetric Mean Absolute Percentage Error (sMAPE). Hasil penelitian menunjukkan bahwa AutoETS memberikan performa terbaik dalam memprediksi frekuensi dan severitas klaim dengan nilai MAPE masing-masing sebesar 8,76% dan 6,54%, sedangkan Exponential Smoothing menghasilkan prediksi total klaim yang lebih stabil dengan MAPE sebesar 3,36%. Berdasarkan evaluasi komposit, AutoETS dipilih sebagai model terbaik secara keseluruhan. Temuan ini menunjukkan bahwa integrasi feature engineering, variabel eksternal, dan pendekatan time series forecasting dapat menghasilkan prediksi klaim yang akurat serta berpotensi mendukung pengambilan keputusan dalam manajemen risiko, estimasi cadangan klaim, dan perencanaan keuangan pada industri asuransi kesehatan.
Kata kunci: Asuransi kesehatan; AutoETS; frekuensi klaim; peramalan deret waktu; severitas klaim.

Abstract

Health insurance claim forecasting plays a crucial role in risk management, financial planning, and strategic decision-making within the insurance industry. However, the uncertainty associated with claim frequency and claim costs necessitates predictive models capable of accurately capturing historical patterns. This study aims to analyze and forecast claim frequency, claim severity, and total claim costs using a time series forecasting approach. Claim and policy data were processed through data preprocessing, missing value handling, feature engineering, and aggregation into monthly time series. To enhance predictive performance, this study incorporates external variables, including the Health Consumer Price Index (CPI), inflation rate, health expenditure per capita, and the number of national holidays, which represent changes in healthcare costs, economic conditions, and healthcare utilization patterns. Three forecasting methods, namely Naive Seasonal, AutoETS, and Exponential Smoothing, were compared using rolling window cross-validation and evaluated based on Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and Symmetric Mean Absolute Percentage Error (sMAPE). The results indicate that AutoETS achieved the best performance in forecasting claim frequency and claim severity, with MAPE values of 8.76% and 6.54%, respectively, while Exponential Smoothing provided more stable predictions for total claim costs, achieving a MAPE of 3.36%. Based on a composite evaluation, AutoETS was selected as the overall best-performing model. These findings suggest that the integration of feature engineering, external variables, and time series forecasting techniques can produce accurate claim predictions and has the potential to support decision-making in risk management, claim reserve estimation, and financial planning in the health insurance industry.
Keyword: AutoETS; claim frequency; claim severity; health insurance; time series forecasting.

I. PENDAHULUAN

Industri asuransi kesehatan memiliki peran strategis dalam menjaga ketahanan finansial masyarakat terhadap risiko kesehatan yang tidak terduga. Seiring dengan meningkatnya biaya layanan kesehatan dan pertumbuhan jumlah peserta asuransi, perusahaan asuransi menghadapi

tantangan yang semakin kompleks dalam menjaga stabilitas keuangan serta merumuskan strategi pengelolaan risiko yang efektif. Ketidakpastian mengenai jumlah klaim yang akan diajukan maupun besaran biaya yang harus dibayarkan berpotensi mengganggu penetapan premi, estimasi cadangan klaim, serta perencanaan keuangan jangka panjang perusahaan. Oleh karena itu, kemampuan untuk memprediksi tren klaim secara akurat menjadi faktor kritis dalam mendukung pengambilan keputusan berbasis data pada industri asuransi [1].

Dalam praktik aktuaria, total biaya klaim pada suatu periode dipengaruhi oleh dua komponen utama, yaitu frekuensi klaim (*claim frequency*) yang mencerminkan jumlah kejadian klaim, dan tingkat keparahan klaim (*claim severity*) yang mencerminkan rata-rata nilai klaim yang diajukan. Pendekatan *time series forecasting* merupakan salah satu metode yang umum digunakan untuk memodelkan kedua komponen tersebut secara terpisah sebelum menghasilkan estimasi total biaya klaim. Metode ini memanfaatkan pola historis data, termasuk tren jangka panjang dan pola musiman, untuk memperkirakan nilai pada periode mendatang [2]. Data klaim asuransi umumnya memiliki karakteristik pola tren dan musiman yang khas sehingga memerlukan metode peramalan yang mampu menangkap dinamika tersebut secara efektif [3].

Meskipun berbagai penelitian sebelumnya telah mengeksplorasi penerapan metode statistik dan pembelajaran mesin untuk peramalan klaim asuransi, sebagian besar pendekatan tersebut belum secara komprehensif mengintegrasikan variabel eksternal yang merepresentasikan kondisi makroekonomi dan pola pemanfaatan layanan kesehatan. Padahal, faktor-faktor seperti perubahan harga layanan kesehatan, tingkat inflasi, serta pola hari libur nasional dapat mempengaruhi frekuensi dan severitas klaim secara signifikan. Kesenjangan ini mendorong perlunya pendekatan yang lebih holistik dalam pemodelan klaim asuransi kesehatan.

Penelitian ini bertujuan menganalisis dan meramalkan frekuensi klaim, severitas klaim, serta total biaya klaim asuransi kesehatan AXA dengan pendekatan *time series forecasting* yang diperkaya oleh variabel eksternal. Proses analisis mencakup tahapan data *preprocessing*, penanganan *missing value*, *feature engineering*, dan agregasi data klaim serta polis menjadi *time series* bulanan. Hasil penelitian ini diharapkan dapat memberikan manfaat praktis bagi perusahaan asuransi dalam menyusun estimasi cadangan klaim yang lebih akurat serta mendukung perencanaan keuangan yang lebih responsif terhadap perubahan kondisi ekonomi dan pola pemanfaatan layanan kesehatan masyarakat.

II. TINJAUAN PUSTAKA

Penelitian terdahulu mengenai peramalan klaim asuransi telah banyak mengeksplorasi penerapan metode statistik klasik maupun pembelajaran mesin, termasuk pendekatan berbasis *deep learning* yang menunjukkan potensi dalam menangkap pola non-linear pada data klaim [1], namun sebagian besar pendekatan tersebut berfokus pada data historis internal tanpa mengintegrasikan variabel eksternal secara komprehensif. Di sisi lain, kajian mengenai *multivariate time series forecasting* menunjukkan bahwa penggabungan beberapa variabel yang saling berkaitan dapat meningkatkan akurasi model dalam menangkap fenomena yang dipengaruhi oleh banyak faktor [4], sementara metode penanganan *missing value* melalui pendekatan imputasi statistik telah terbukti mampu menjaga karakteristik distribusi data ketika proporsi data yang hilang relatif kecil [5]. Konsep *shrinkage* dalam pemodelan statistik, yang menggabungkan hasil estimasi model dengan nilai referensi historis, juga telah banyak digunakan untuk meningkatkan stabilitas estimasi pada berbagai konteks pemodelan ekonometrika [7]. Berdasarkan tinjauan tersebut, penelitian ini memposisikan diri sebagai upaya untuk mengisi kesenjangan dengan mengintegrasikan variabel eksternal makroekonomi ke dalam kerangka peramalan *time series* klasik untuk kasus klaim asuransi kesehatan.

III. METODOLOGI PENELITIAN

A. Dataset dan Preprocessing

Data yang digunakan dalam penelitian ini bersumber dari dataset klaim dan dataset polis asuransi kesehatan AXA. Kedua dataset tersebut terlebih dahulu digabungkan (*join*) berdasarkan identifikasi peserta untuk memperoleh informasi yang lebih lengkap mengenai karakteristik peserta sekaligus klaim yang diajukan. Setelah proses

penggabungan, data transaksi diagregasi ke dalam *time series* bulanan sehingga setiap periode memiliki informasi mengenai jumlah klaim, rata-rata nilai klaim, serta total biaya klaim yang tercatat.

Sebelum dilakukan *feature engineering*, nilai yang hilang (*missing value*) dalam dataset ditangani menggunakan pendekatan *statistical imputation*, yaitu mengisi nilai yang tidak tersedia dengan nilai rata-rata (*mean*) atau nilai representatif dari distribusi variabel yang bersangkutan. Pendekatan ini dipilih karena mampu mempertahankan karakteristik distribusi data serta relatif stabil ketika proporsi *missing value* tidak terlalu besar [5].

B. Feature Engineering

Feature engineering bertujuan menghasilkan variabel baru yang lebih informatif guna meningkatkan kemampuan model dalam menangkap pola pada data *time series* [2]. Fitur internal diturunkan dari dataset klaim dan polis, meliputi rata-rata usia peserta (*avg_age*), rata-rata lama rawat inap (*avg_los*), persentase klaim rawat inap (*pct_inpatient*), dan persentase klaim penyakit kronis (*pct_chronic*). Fitur waktu (*time-based features*) ditambahkan untuk membantu model mengenali pola musiman, mencakup transformasi siklik *month_sin* dan *month_cos*, serta fitur *lag* untuk memanfaatkan informasi dari periode sebelumnya.

Variabel eksternal diintegrasikan untuk memberikan konteks makroekonomi yang relevan, meliputi Indeks Harga Konsumen (IHK) sektor kesehatan dan tingkat inflasi bulanan dari Badan Pusat Statistik (BPS), *health expenditure per capita* dari basis data World Bank, serta jumlah hari libur nasional dari *application programming interface* (API) Calendarific. Penggunaan variabel eksogen ini sejalan dengan konsep *multivariate time series forecasting* yang memungkinkan beberapa variabel dimodelkan secara bersama-sama untuk menangkap fenomena yang kompleks [4].

C. Model Peramalan

Penelitian ini menggunakan tiga model peramalan. Naive Seasonal digunakan sebagai *baseline* model dengan memprediksi nilai suatu periode berdasarkan nilai pada periode yang sama dalam siklus sebelumnya, menggunakan parameter $K = 12$ yang merepresentasikan siklus musiman tahunan.

AutoETS (Automatic Error – Trend – Seasonal) memodelkan komponen level, tren, dan musiman secara otomatis, dikonfigurasi dengan spesifikasi *model = "MAM"* (*multiplicative error*, *additive trend*, *multiplicative seasonality*) serta *season_length* = 4. Untuk meningkatkan stabilitas prediksi, hasil keluaran AutoETS melalui proses shrinkage dengan parameter $\lambda = 0{,}1$, yang dinyatakan sebagai:

$$y_t = (1 - \lambda)\, \hat{y}_t + \lambda \bar{y} \qquad (1)$$

di mana $\hat{y}_t$ adalah nilai prediksi model pada periode t, $\bar{y}$ adalah rata-rata historis data, dan λ adalah parameter *shrinkage*. Konsep ini banyak digunakan untuk meningkatkan stabilitas estimasi dengan menggabungkan informasi dari model dan nilai referensi tertentu [6]. Selain itu, diterapkan pula mekanisme *rolling floor constraint* dan *historical minimum clipping* untuk menjaga nilai prediksi tetap berada dalam rentang yang realistis.

Exponential Smoothing memberikan bobot lebih besar pada data terbaru sehingga lebih adaptif terhadap perubahan pola [2], dikonfigurasi dengan *additive trend* dan *additive seasonal* serta *seasonal_periods* = 6.

Frekuensi klaim dan severitas klaim diprediksi secara terpisah. Nilai total biaya klaim kemudian dihitung melalui hubungan aktuaria:

$$T_t = F_t \times S_t \qquad (2)$$

di mana T_t adalah total biaya klaim, F_t adalah frekuensi klaim, dan S_t adalah severitas klaim pada periode t.

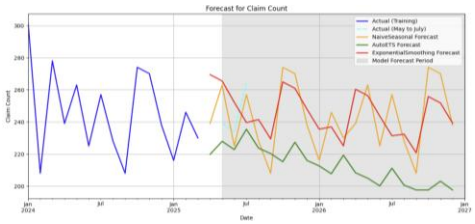
D. Evaluasi Model

Evaluasi model dilakukan menggunakan metode *rolling window cross-validation*, di mana model dilatih secara berulang dengan menggeser jendela waktu pelatihan sehingga performa model dapat diuji pada beberapa periode tanpa melanggar urutan temporal data [3]. Kinerja model diukur menggunakan tiga metrik, yaitu *Mean Absolute Percentage Error* (MAPE), *Root Mean Squared Error* (RMSE), dan *Symmetric Mean Absolute Percentage Error* (sMAPE), yang secara bersama-sama memberikan gambaran komprehensif mengenai akurasi dan stabilitas prediksi.

IV. HASIL DAN PEMBAHASAN

A. Analisis Eksploratif Data

Exploratory Data Analysis (EDA) dilakukan untuk memahami karakteristik data klaim asuransi kesehatan sebelum proses pemodelan, dengan fokus pada tiga variabel utama: frekuensi klaim (*claim_count*), rata-rata nilai klaim (*avg_severity*), dan total biaya klaim (*total_paid*). Visualisasi deret waktu ketiga variabel menunjukkan adanya pola fluktuasi dari waktu ke waktu yang mengindikasikan dinamika pada jumlah dan nilai klaim yang diajukan peserta asuransi, dipengaruhi oleh karakteristik demografis peserta, jenis layanan kesehatan yang dimanfaatkan, serta kondisi makroekonomi.



Gambar 1. Visualisasi pola *time series* dan prediksi terhadap frekuensi klaim

B. Pelatihan dan Evaluasi Model

Tiga model peramalan dilatih pada data historis yang telah melalui *preprocessing* dan *feature engineering*. Evaluasi *out-of-sample* dilakukan menggunakan *rolling window cross-validation* untuk mengukur kemampuan generalisasi masing-masing model.

Out-of-Sample Evaluation Results:

	Model	Target	MAPE	RMSE	sMAPE
0	NaiveSeasonal (Out-of-Sample)	claim_count	9.055297	2.805946e+01	9.206980
1	AutoETS (Out-of-Sample)	claim_count	8.761561	2.651692e+01	9.327099
2	ExponentialSmoothing (Out-of-Sample)	claim_count	10.218779	2.617586e+01	9.716740
3	NaiveSeasonal (Out-of-Sample)	avg_severity	9.677732	6.575019e+06	10.371526
4	AutoETS (Out-of-Sample)	avg_severity	6.541261	4.294321e+06	6.386183
5	ExponentialSmoothing (Out-of-Sample)	avg_severity	9.606977	5.570466e+06	10.164318
6	NaiveSeasonal (Out-of-Sample)	total_paid	8.867654	1.217167e+09	9.303688
7	AutoETS (Out-of-Sample)	total_paid	7.344783	9.817544e+08	7.639016
8	ExponentialSmoothing (Out-of-Sample)	total_paid	3.362888	7.082502e+08	3.502779

Gambar 2. Hasil evaluasi *out-of-sample* ketiga model pada variabel frekuensi klaim, severitas klaim, dan total biaya klaim

Hasil evaluasi menunjukkan bahwa AutoETS memberikan performa terbaik dalam memprediksi frekuensi klaim dan severitas klaim, dengan nilai MAPE sebesar 8,76% dan 6,54%. Keunggulan ini dapat dikaitkan dengan kemampuan AutoETS memodelkan komponen *error*, tren, dan musiman secara simultan dengan spesifikasi yang optimal, didukung oleh mekanisme *shrinkage* yang meredam volatilitas prediksi. Untuk variabel total biaya klaim, Exponential Smoothing menghasilkan prediksi yang lebih stabil dengan MAPE sebesar 3,36%, menunjukkan kemampuannya menangkap pola total klaim dengan lebih konsisten karena sifatnya yang lebih adaptif terhadap perubahan lokal dalam data.

Tabel 1. Perbandingan performa model berdasarkan metrik MAPE *out-of-sample*

Model	MAPE Frekuensi Klaim	MAPE Severitas Klaim	MAPE Total Klaim
Naive Seasonal	9,05%	9,67%	8,87%
AutoETS	8,76%	6,54%	7,34%
Exponential Smoothing	10,21%	9,61%	3,36%

Secara komposit, AutoETS dipilih sebagai model terbaik karena menghasilkan rata-rata kesalahan prediksi terendah di antara ketiga model pada sebagian besar indikator klaim yang dianalisis.

C. Hasil Prediksi

Menggunakan model AutoETS, prediksi frekuensi dan severitas klaim bulanan dihasilkan untuk periode 2026 dengan total biaya klaim dihitung secara aktuaria melalui Persamaan (2).

	id	value
0	2026_01_Claim_Frequency	2.125854e+02
1	2026_01_Claim_Severity	5.228496e+07
2	2026_01_Total_Claim	1.111502e+10
3	2026_02_Claim_Frequency	2.076759e+02
4	2026_02_Claim_Severity	5.714065e+07
5	2026_02_Total_Claim	1.186673e+10
6	2026_03_Claim_Frequency	2.192400e+02
7	2026_03_Claim_Severity	5.428362e+07
8	2026_03_Total_Claim	1.190114e+10

Gambar 3. Hasil prediksi model AutoETS untuk frekuensi klaim, severitas klaim, dan total biaya klaim pada tahun 2026

Prediksi yang dihasilkan mencerminkan tren yang realistis dan konsisten dengan pola historis data, sebagian berkat mekanisme *rolling floor constraint* dan *historical minimum clipping* yang diterapkan untuk membatasi nilai prediksi dalam rentang yang masuk akal. Kombinasi antara model AutoETS dan transformasi stabilisasi ini menghasilkan prediksi yang tidak hanya akurat secara statistik, tetapi juga dapat diinterpretasikan secara praktis oleh para pemangku kepentingan di industri asuransi.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil membangun dan membandingkan tiga model peramalan *time series* untuk memprediksi frekuensi klaim, severitas klaim, dan total biaya klaim asuransi kesehatan AXA. Melalui tahapan preprocessing yang terstruktur, *feature engineering* berbasis data internal dan eksternal, serta evaluasi menggunakan *rolling window cross-validation*, penelitian ini menemukan bahwa model AutoETS memberikan performa terbaik secara keseluruhan, dengan MAPE sebesar 8,76% untuk frekuensi klaim dan 6,54% untuk severitas klaim, sementara Exponential Smoothing unggul dalam memprediksi total biaya klaim dengan MAPE sebesar 3,36%. Temuan ini menegaskan bahwa integrasi *feature engineering* yang komprehensif, mencakup fitur internal, fitur waktu, dan variabel eksternal makroekonomi, bersama pendekatan *time series forecasting* yang tepat dapat menghasilkan prediksi klaim yang akurat, sehingga berpotensi mendukung perusahaan asuransi dalam merencanakan cadangan klaim yang lebih presisi, mengoptimalkan penetapan premi, serta merancang strategi manajemen risiko yang lebih proaktif. Penelitian ini masih memiliki keterbatasan, terutama dalam hal cakupan variabel eksternal yang digunakan dan kompleksitas model yang diterapkan, sehingga untuk pengembangan selanjutnya disarankan agar penelitian mengintegrasikan indikator ekonomi dan kesehatan yang lebih beragam, mengeksplorasi pendekatan *hybrid model* yang menggabungkan metode statistik dengan teknik pembelajaran mesin, serta memperluas cakupan data ke periode yang lebih panjang guna meningkatkan keandalan generalisasi model.

DAFTAR PUSTAKA

[1] K. Navatha, N. C. Babu, dan V. V. H. Gopal, "Forecasting insurance claims using deep learning approach," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, 2023.

[2] R. J. Hyndman dan G. Athanasopoulos, Forecasting: Principles and Practice. Melbourne, Australia: OTexts, 2014.

[3] R. M. Kunst, Econometric Forecasting. Vienna, Austria: University of Vienna, 2012.

[4] Y. Zhang, et al., "Multivariate time-series forecasting: A review of deep learning methods in Internet of Things application to smart cities," *IEEE Internet of Things Journal*, 2023.

[5] A. D. Shah, J. W. Bartlett, J. Carpenter, O. Nicholas, dan H. Hemingway, "Comparison of random forest and parametric imputation models for imputing missing data using MICE: A CALIBER study," American Journal of Epidemiology, vol. 179, no. 6, hlm. 764–774, 2014.

[6] G. Koop dan D. Korobilis, "Bayesian multivariate time series methods for empirical macroeconomics," Foundations and Trends in Econometrics, vol. 3, no. 4, hlm. 267–358, 2010.

[7] Badan Pusat Statistik. (2023). Indeks Harga Konsumen (2018=100) Menurut Kelompok dan Sub Kelompok 05 Kesehatan. [Online]. Tersedia: https://www.bps.go.id

[8] Badan Pusat Statistik. (2026). Inflasi Bulanan (Month-to-Month). [Online]. Tersedia: https://www.bps.go.id

[9] Calendarific. (2025). Indonesia Public Holidays in 2025. [Online]. Tersedia: https://calendarific.com/holidays/2025/ID

[10] World Bank. (2024). Health Expenditure per Capita (Current US$). [Online]. Tersedia: https://data.worldbank.org/indicator/SH.XPD.CHEX.PC.CD

PERBANDINGAN KINERJA DBSCAN CLUSTERING DENGAN DAN TANPA REDUKSI DIMENSI UMAP PADA DATA AKADEMIK MAHASISWA

Baiq Mutia Dewi Edelweis [1)], Naufal Pramudya Ananda [1)], Dodi Wijaya [1)]
1) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: f1d02310107@student.unram.ac.id

Abstrak

Curse of dimensionality menjadi tantangan utama DBSCAN karena jarak antar titik di ruang berdimensi tinggi kehilangan makna, sehingga menurunkan kualitas clustering. Penelitian ini membandingkan kinerja DBSCAN pada data akademik mahasiswa (N=4.421) dalam dua kondisi: tanpa reduksi dimensi (14 fitur) dan dengan reduksi dimensi menggunakan UMAP (5 dimensi). Penelitian ini menerapkan feature engineering, seleksi fitur menggunakan Random Forest, scaling dengan RobustScaler, pembersihan outlier melalui IsolationForest, serta grid search parameter DBSCAN berdasarkan composite scoring. Hasil menunjukkan kedua metode berkinerja baik, namun DBSCAN+UMAP unggul di 6 dari 8 metrik evaluasi. DBSCAN+UMAP menghasilkan silhouette score 0,838 (tanpa reduksi: 0,696), Calinski-Harabasz 17.156 (tanpa reduksi: 5.711), Davies-Bouldin 0,194 (tanpa reduksi: 0,403), dan noise ratio 0% (tanpa reduksi: 0,4%). UMAP juga unggul dalam NMI (0,1712 vs 0,1711) dan completeness (0,247 vs 0,240). Sementara itu, DBSCAN tanpa reduksi unggul tipis pada ARI (0,164 vs 0,160) dan homogeneity (0,133 vs 0,131). Kedua metode menghasilkan 3 cluster yang sesuai dengan label Dropout, Enrolled, dan Graduate. Kesimpulannya, meskipun DBSCAN tanpa reduksi sudah bekerja baik (silhouette >0,69), penerapan UMAP tetap memberikan peningkatan performa yang signifikan, terutama dalam kekompakan cluster (silhouette +20,5%) dan pengurangan noise. Reduksi dimensi UMAP direkomendasikan untuk segmentasi profil mahasiswa yang lebih presisi.
Kata kunci: Clustering; DBSCAN; Reduksi Dimensi; Segmentasi Mahasiswa; UMAP

Abstract

The curse of dimensionality is a major challenge for DBSCAN because distances between points in high-dimensional space lose their meaning, thereby reducing the quality of clustering. This study compares the performance of DBSCAN on student academic data (N=4,421) under two conditions: without dimensionality reduction (14 features) and with dimensionality reduction using UMAP (5 dimensions). This study employs feature engineering, feature selection using Random Forest, scaling with RobustScaler, outlier removal via IsolationForest, and a grid search for DBSCAN parameters based on composite scoring. The results show that both methods perform well, but DBSCAN+UMAP outperforms the other method on 6 out of 8 evaluation metrics. DBSCAN+UMAP yields a silhouette score of 0.838 (without reduction: 0.696), a Calinski-Harabasz score of 17.156 (without reduction: 5.711), a Davies-Bouldin score of 0.194 (without reduction: 0.403), and a noise ratio of 0% (without reduction: 0.4%). UMAP also outperformed DBSCAN in NMI (0.1712 vs. 0.1711) and completeness (0.247 vs. 0.240). Meanwhile, DBSCAN without reduction had a slight edge in ARI (0.164 vs. 0.160) and homogeneity (0.133 vs. 0.131). Both methods produced 3 clusters that matched the labels "Dropout," "Enrolled," and "Graduate." In conclusion, although DBSCAN without dimension reduction already performs well (silhouette > 0.69), the application of UMAP still provides a significant performance improvement, particularly in cluster compactness (silhouette +20.5%) and noise reduction. UMAP-based dimension reduction is recommended for more precise student profile segmentation.
Keyword: *Clustering; DBSCAN; Dimension Reduction; Student Segmentation; UMAP*

I. PENDAHULUAN

Fenomena *student dropout* di tingkat perguruan tinggi menciptakan tantangan akademik dan sosial-ekonomi bagi institusi maupun mahasiswa. Kerugian individual seperti hilangnya pendapatan dan biaya studi, serta kerugian sosial akibat berkurangnya potensi tenaga kerja dan kekurangan tenaga ahli terampil merupakan dampak dari fenomena ini [1]. Untuk mengatasi permasalahan terkait, model prediksi dini yang efektif dapat digunakan untuk mengidentifikasi mahasiswa berisiko dan menerapkan intervensi dini, meningkatkan efektivitas program dukungan sekaligus memungkinkan institusi membedakan mahasiswa berisiko tinggi dan rendah untuk alokasi sumber daya yang optimal [2]. Adapun pemahaman terhadap faktor-faktor (atribut) yang memengaruhi performa dan retensi mahasiswa merupakan salah satu poin penting dalam

pengimplementasian solusi dengan model prediksi. Dalam konteks ini, pendekatan unsupervised learning dapat dimanfaatkan untuk mengelompokkan mahasiswa berdasarkan kemiripan karakteristiknya.

Penggunaan machine learning dalam prediksi dini untuk kasus dropout telah banyak dikembangkan, dengan metode-metode seperti Random Forest, Decision Tree, Logistic Regression, dan Naive Bayes sebagai pendekatan yang paling umum digunakan [3]. Adapun DBSCAN (Density-Based Spatial Clustering of Applications with Noise) juga menjadi salah satu metode yang dapat digunakan, yaitu algoritma clustering berbasis kepadatan yang mampu mengidentifikasi cluster dengan bentuk tidak beraturan berdasarkan jarak dengan objek-objek di bawah nilai threshold tertentu dan menangani data yang mengandung noise atau outlier [4]. Keadaan ini diperparah oleh apa yang dikenal sebagai curse of dimensionality. DBSCAN bekerja dengan secara berulang memperluas setiap kluster ke area kepadatan tinggi di sekitarnya, sehingga kluster tersebut sepenuhnya dikelilingi oleh area kepadatan rendah [5]. DBSCAN harus melakukan pencarian rentang yang intensif untuk menentukan apakah titik tersebut titik inti atau bukan, yang memerlukan komputasi tinggi dan membatasi penerapannya dalam data berdimensi tinggi [6].

Curse of dimensionality merupakan fenomena di mana peningkatan dimensi fitur menyebabkan volume ruang berkembang secara eksponensial, sehingga data menjadi semakin jarang (sparse) dan jarak antar titik kehilangan maknanya [4]. Akibatnya, metrik jarak tradisional seperti Euclidean distance menjadi tidak efektif, algoritma menjadi lebih sensitif terhadap noise, dan performa clustering menurun secara substansial. K-Means, misalnya, menunjukkan kecenderungan jarak Euclidean untuk konvergen di ruang berdimensi tinggi, menyebabkan semua sampel terdistribusi seragam dalam ruang jarak dan strategi clustering berbasis jarak kehilangan daya diskriminatifnya. Untuk mengatasi masalah ini, reduksi dimensi menjadi langkah preprocessing yang penting.

Reduksi dimensi adalah teknik preprocessing yang bertujuan mengurangi dimensi data berdimensi tinggi dengan memetakannya ke ruang berdimensi lebih rendah sambil mempertahankan struktur dan karakteristik distribusi aslinya semaksimal mungkin [4]. Salah satu metode reduksi dimensi non-linear yang paling efektif adalah UMAP (Uniform Manifold Approximation and Projection), yang dibangun berdasarkan prinsip Riemannian geometry dan algebraic topology, serta dirancang untuk mempertahankan struktur lokal dan global data secara bersamaan dengan efisiensi komputasi yang tinggi [7]. Dengan kemampuan ini, UMAP menjadi solusi yang efektif untuk mengatasi curse of dimensionality sebelum clustering.

Penelitian terkait menunjukkan bahwa preprocessing dengan UMAP secara konsisten meningkatkan kualitas clustering pada berbagai algoritma, dengan peningkatan pada DBSCAN mampu melewati performa metode clustering kompleks seperti SPECTACL [8]. Selain itu, kombinasi UMAP dan DBSCAN terbukti efektif dalam mengidentifikasi subpopulasi yang terpisah secara visual pada data kompleks [8]. Namun, meskipun UMAP dan DBSCAN telah digunakan secara terpisah dalam analisis data pendidikan, studi yang secara sistematis membandingkan performa DBSCAN dengan dan tanpa reduksi dimensi UMAP pada dataset akademik mahasiswa masih terbatas. Penelitian ini bertujuan menjawab pertanyaan tersebut dengan membandingkan kinerja DBSCAN pada data akademik mahasiswa dalam dua kondisi: tanpa reduksi dimensi dan dengan reduksi dime nsi menggunakan UMAP, serta menganalisis pengaruh UMAP terhadap kualitas clustering yang dihasilkan. Hasil penelitian ini diharapkan memberikan manfaat bagi institusi pendidikan dalam memilih pendekatan preprocessing yang tepat untuk segmentasi mahasiswa, serta menjadi referensi bagi peneliti lain dalam menerapkan kombinasi UMAP dan DBSCAN pada data akademik berdimensi tinggi.

II. TINJAUAN PUSTAKA

Tinjauan pustaka ini membahas konsep-konsep yang menjadi dasar penelitian, yaitu curse of dimensionality, clustering berbasis densitas, UMAP sebagai teknik reduksi dimensi, seleksi fitur dan penanganan outlier, evaluasi kinerja klastering, serta penelitian terkait pada data akademik mahasiswa.

2.1 Curse of Dimensionality dan Clustering Berbasis Densitas

Curse of dimensionality adalah fenomena ketika peningkatan jumlah dimensi data menyebabkan jarak antar titik data menjadi kurang informatif, distribusi data menjadi makin jarang, dan kinerja algoritma pembelajaran mesin menurun. Dalam ruang berdimensi tinggi, banyak ukuran jarak klasik seperti Euclidean atau Manhattan dapat kehilangan makna karena perbedaan antar titik menjadi semakin sulit dibedakan. Kondisi ini sering menjadi tantangan dalam clustering karena proses pengelompokan sangat bergantung pada struktur jarak dan kepadatan data [9].

Clustering berbasis densitas dikembangkan untuk mengatasi keterbatasan metode berbasis pusat seperti K-Means, terutama ketika data memiliki bentuk klaster yang tidak beraturan atau mengandung noise. Pendekatan ini mengelompokkan data berdasarkan area dengan kepadatan tinggi yang dipisahkan oleh area berkepadatan rendah, sehingga lebih sesuai untuk data kompleks dan tidak homogen. Dalam penelitian dengan data berdimensi tinggi, pendekatan ini sering dipadukan dengan reduksi dimensi agar struktur kepadatan data lebih mudah diidentifikasi [9][10].

2.2 UMAP sebagai Teknik Reduksi Dimensi

UMAP (Uniform Manifold Approximation and Projection) adalah teknik reduksi dimensi berbasis manifold learning yang dirancang untuk menjaga struktur lokal data sekaligus tetap mempertahankan pola global secara efisien. Metode ini banyak digunakan karena skalabilitasnya yang baik dan kemampuannya menghasilkan representasi rendah dimensi yang cocok untuk visualisasi maupun pra-pemrosesan sebelum clustering. Dibandingkan beberapa metode lain, UMAP sering dipilih karena mampu memberikan kualitas visualisasi yang baik dengan waktu komputasi yang relatif lebih cepat [11].

Dalam konteks penelitian berbasis data kompleks, UMAP dapat membantu memproyeksikan data berdimensi tinggi ke ruang yang lebih sederhana sehingga pola kedekatan antar data lebih mudah diamati. Pendekatan ini bermanfaat ketika data mengandung struktur tersembunyi yang sulit terlihat pada ruang asli, termasuk ketika digunakan untuk mendeteksi outlier atau memisahkan kelompok data yang berdekatan. Oleh karena itu, UMAP relevan sebagai tahap awal sebelum proses clustering berbasis densitas dilakukan [12].

2.3 Seleksi Fitur dan Penanganan Outlier

Seleksi fitur bertujuan memilih atribut yang paling relevan agar model lebih sederhana, lebih cepat diproses, dan lebih sedikit dipengaruhi oleh fitur yang tidak informatif [9]. Pada data berdimensi tinggi, seleksi fitur menjadi penting karena banyak atribut justru dapat memperbesar efek curse of dimensionality dan menurunkan kualitas pengelompokan. Dengan memilih fitur yang paling bermakna, struktur data cenderung menjadi lebih jelas dan hasil clustering lebih stabil [10].

Penanganan outlier diperlukan karena data ekstrem dapat mengganggu distribusi, mengaburkan pola utama, dan menurunkan kualitas hasil analisis. Salah satu pendekatan yang sering digunakan adalah metode IQR untuk mendeteksi nilai yang berada jauh di luar sebaran normal data. Studi lain juga menunjukkan bahwa pembersihan outlier, penyaringan fitur, dan penanganan ketidakseimbangan data dapat meningkatkan akurasi model secara signifikan. Dalam penelitian yang menggabungkan clustering dan reduksi dimensi, langkah ini membantu memastikan bahwa pola yang terbentuk benar-benar mencerminkan struktur data, bukan gangguan dari noise atau nilai ekstrem [12].

2.4 Evaluasi Kinerja Klastering

Evaluasi kinerja klastering diperlukan untuk menilai kualitas hasil pengelompokan, baik dari sisi kesesuaian internal antar anggota klaster maupun dari kesesuaiannya terhadap acuan eksternal bila label sebenarnya tersedia. Amigó, Enrique, et al. menekankan bahwa evaluasi klastering perlu dipahami secara hati-hati karena setiap metrik menangkap aspek kualitas yang berbeda, sehingga pemilihan ukuran evaluasi harus disesuaikan dengan tujuan analisis. Dalam praktiknya, metrik yang sering digunakan untuk evaluasi internal antara lain Silhouette Score, Davies-Bouldin Index, dan Calinski-Harabasz Index. Silhouette Score digunakan untuk melihat seberapa baik suatu data berada di dalam klasternya dibandingkan dengan klaster lain, Davies-Bouldin Index menilai perbandingan antara kemiripan dalam klaster dan pemisahan antar klaster, sedangkan Calinski-Harabasz Index

mengukur rasio antara dispersi antar klaster dan dispersi di dalam klaster [13].

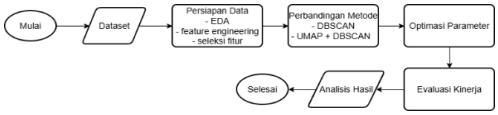
Selain evaluasi internal, evaluasi ekstrinsik juga dapat dilakukan dengan membandingkan hasil klaster terhadap label acuan atau ground truth apabila tersedia. Pendekatan ini relevan ketika penelitian ingin menilai apakah struktur klaster yang terbentuk benar-benar mencerminkan pola sebenarnya pada data. Dengan demikian, pemilihan metrik evaluasi tidak hanya ditentukan oleh karakteristik data, tetapi juga oleh tujuan penelitian, apakah untuk analisis deskriptif, segmentasi kelompok, atau penilaian kesesuaian hasil klastering secara objektif [13].

2.5 Penelitian Terkait pada Data Akademik Mahasiswa

Penelitian pada data akademik mahasiswa telah memanfaatkan clustering untuk mengelompokkan mahasiswa berdasarkan performa akademik dan karakteristik tertentu. Berbagai penelitian menunjukkan bahwa kualitas clustering dipengaruhi oleh dimensi data, fitur yang tidak relevan, serta keberadaan noise dan outlier [14]. Oleh karena itu, kombinasi reduksi dimensi, seleksi fitur, penanganan outlier, dan clustering berbasis densitas menjadi pendekatan yang relevan untuk meningkatkan kualitas segmentasi mahasiswa.

III. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode eksperimen kuantitatif dengan pendekatan komparatif untuk membandingkan kinerja algoritma Density-Based Spatial Clustering of Applications with Noise (DBSCAN) pada data akademik mahasiswa dengan dan tanpa reduksi dimensi menggunakan Uniform Manifold Approximation and Projection (UMAP). Tujuan penelitian adalah menganalisis pengaruh reduksi dimensi terhadap kualitas clustering yang dihasilkan oleh DBSCAN pada data berdimensi tinggi. Alur penelitian yang dilakukan ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

3.1 Dataset

Tahap pertama penelitian adalah pengumpulan data menggunakan dataset *Predict Students' Dropout and Academic Success* [15]. Dataset ini terdiri atas 4.421 data mahasiswa dengan berbagai atribut akademik, sosial, dan ekonomi yang digunakan untuk mengidentifikasi pola dan karakteristik mahasiswa melalui proses clustering.

Dataset memiliki karakteristik berupa banyak atribut, perbedaan skala antar fitur, dan keberadaan outlier yang berpotensi memengaruhi kualitas clustering. Kondisi ini dapat memicu fenomena curse of dimensionality pada algoritma berbasis kepadatan.

3.2 Persiapan Data

Setelah dataset diperoleh, dilakukan tahap persiapan data yang bertujuan meningkatkan kualitas data sebelum proses clustering. Tahap ini diawali dengan *Exploratory Data Analysis* (EDA) untuk memahami karakteristik data, distribusi fitur, serta mengidentifikasi potensi permasalahan seperti data ekstrem dan ketidakseimbangan distribusi.

Selanjutnya dilakukan *feature engineering* untuk menghasilkan fitur-fitur baru yang lebih representatif terhadap kondisi akademik mahasiswa. Setelah itu dilakukan seleksi fitur menggunakan Random Forest berdasarkan nilai *feature importance* sehingga hanya fitur yang paling relevan yang digunakan dalam penelitian[16].

Data hasil seleksi fitur kemudian melalui tahap preprocessing yang meliputi normalisasi menggunakan RobustScaler untuk menyamakan skala antar fitur dan mengurangi pengaruh data ekstrem [17]. Selain itu, dilakukan deteksi serta penghapusan outlier menggunakan Isolation Forest untuk meningkatkan kualitas data yang akan digunakan dalam proses clustering.

3.3 Perbandingan Metode Clustering

Pada tahap ini dilakukan perbandingan dua metode clustering yang menjadi fokus penelitian. Metode pertama adalah DBSCAN tanpa reduksi dimensi, di mana algoritma DBSCAN diterapkan langsung pada data hasil preprocessing yang memiliki 14 fitur. Metode kedua adalah kombinasi UMAP dan DBSCAN, yaitu data terlebih dahulu direduksi dimensinya menggunakan UMAP menjadi 5 dimensi sebelum dilakukan clustering menggunakan DBSCAN [18].

Perbandingan kedua metode ini dilakukan untuk mengetahui apakah reduksi dimensi menggunakan UMAP mampu meningkatkan

performa DBSCAN dalam mengelompokkan data akademik mahasiswa.

3.4 Optimasi Parameter

Untuk memperoleh konfigurasi terbaik, parameter DBSCAN dioptimasi menggunakan metode *grid search*. Proses ini dilakukan dengan menguji berbagai kombinasi parameter, seperti nilai *eps* dan *min_samples*, kemudian memilih kombinasi terbaik berdasarkan nilai *composite scoring*. Tahap optimasi dilakukan pada kedua metode agar perbandingan yang dihasilkan lebih objektif.

3.5 Evaluasi Kinerja

Hasil clustering dari masing-masing metode dievaluasi menggunakan metrik internal dan eksternal. Metrik internal terdiri atas Silhouette Score, Calinski-Harabasz Index, Davies-Bouldin Index, dan Noise Ratio yang digunakan untuk mengukur kualitas cluster berdasarkan struktur data. Sementara itu, metrik eksternal terdiri atas Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Homogeneity, dan Completeness yang digunakan untuk mengevaluasi kesesuaian hasil clustering terhadap karakteristik data yang tersedia [13].

3.6 Analisis Hasil

Tahap terakhir adalah analisis hasil evaluasi untuk membandingkan performa DBSCAN tanpa reduksi dimensi dan DBSCAN dengan reduksi dimensi UMAP. Analisis dilakukan berdasarkan seluruh metrik evaluasi yang digunakan sehingga dapat diketahui pengaruh reduksi dimensi terhadap kualitas clustering. Hasil analisis kemudian digunakan untuk menentukan metode yang memberikan performa terbaik dalam segmentasi profil mahasiswa.

IV. HASIL DAN PEMBAHASAN

Penelitian ini membandingkan kinerja algoritma DBSCAN pada data akademik mahasiswa dalam dua skenario, yaitu tanpa reduksi dimensi dan dengan reduksi dimensi menggunakan UMAP. Setelah melalui tahap persiapan data yang meliputi feature engineering, seleksi fitur menggunakan Random Forest, normalisasi menggunakan RobustScaler, serta penghapusan outlier menggunakan Isolation Forest, diperoleh data akhir yang siap digunakan dalam proses clustering.

Hasil seleksi fitur menghasilkan 14 fitur terbaik yang digunakan sebagai masukan pada skenario pertama. Pada skenario kedua, 14 fitur tersebut direduksi menggunakan UMAP menjadi 5 dimensi sebelum dilakukan proses clustering. Selanjutnya dilakukan optimasi parameter DBSCAN menggunakan metode grid search untuk memperoleh konfigurasi parameter terbaik pada masing-masing skenario.

Berdasarkan hasil clustering, kedua metode berhasil menghasilkan tiga cluster utama. Jumlah cluster yang terbentuk menunjukkan bahwa baik DBSCAN tanpa reduksi dimensi maupun DBSCAN dengan UMAP menghasilkan tiga cluster utama pada data akademik mahasiswa. Selain itu, kedua metode menghasilkan pola clustering yang relatif konsisten dengan karakteristik data akademik mahasiswa.

Tabel 1 menunjukkan hasil evaluasi kinerja clustering pada kedua metode.

Tabel 1. Perbandingan hasil evaluasi DBSCAN tanpa reduksi dimensi dan DBSCAN+UMAP

Metrik	DBSCAN Tanpa Reduksi	DBSCAN + UMAP
Dimensi Input	14	5
Jumlah Cluster	3	3
Noise Ratio	0,004	0,0
Silhouette Score	0,6956	0,8382
Calinski-Harabasz Index	5.710,95	17.156,43
Davies-Bouldin Index	0,4031	0,1941
ARI	0,1638	0,1599
NMI	0,1711	0,1712
Homogeneity	0,1330	0,1309
Completeness	0,2398	0,2472

Berdasarkan Tabel 1, metode DBSCAN+UMAP menunjukkan performa yang lebih baik pada sebagian besar metrik evaluasi. Nilai Silhouette Score meningkat dari 0,6956 menjadi 0,8382 atau sekitar 20,5%. Peningkatan ini menunjukkan bahwa cluster yang dihasilkan menjadi lebih kompak dan memiliki pemisahan yang lebih baik dibandingkan metode tanpa reduksi dimensi.

Peningkatan performa juga terlihat pada Calinski-Harabasz Index yang meningkat dari 5.710,95 menjadi 17.156,43. Nilai yang lebih tinggi menunjukkan bahwa variasi antar cluster

semakin besar dibandingkan variasi di dalam cluster, sehingga kualitas clustering menjadi lebih baik. Selain itu, Davies-Bouldin Index mengalami penurunan dari 0,4031 menjadi 0,1941. Karena nilai Davies-Bouldin yang lebih rendah menunjukkan kualitas cluster yang lebih baik, hasil ini mengindikasikan bahwa cluster yang dihasilkan DBSCAN+UMAP memiliki tingkat overlap yang lebih rendah.

Pada metrik evaluasi eksternal, DBSCAN+UMAP memperoleh nilai NMI sebesar 0,1712 dan Completeness sebesar 0,2472 yang sedikit lebih tinggi dibandingkan metode tanpa reduksi dimensi. Sebaliknya, DBSCAN tanpa reduksi dimensi masih menunjukkan keunggulan tipis pada metrik ARI sebesar 0,1638 dan Homogeneity sebesar 0,1330. Namun, selisih kedua metrik tersebut relatif kecil dibandingkan peningkatan yang diperoleh pada metrik internal.

Dari sisi deteksi noise, DBSCAN+UMAP berhasil menghilangkan seluruh data noise sehingga menghasilkan noise ratio sebesar 0%, sedangkan DBSCAN tanpa reduksi dimensi masih menghasilkan noise sebesar 0,4%. Hasil ini menunjukkan bahwa reduksi dimensi menggunakan UMAP mampu menghasilkan representasi data yang lebih baik sehingga memudahkan DBSCAN dalam mengidentifikasi struktur kepadatan data.

Secara keseluruhan, hasil penelitian menunjukkan bahwa penggunaan UMAP sebelum proses clustering memberikan peningkatan kualitas cluster yang signifikan. Temuan ini mendukung teori curse of dimensionality yang menyatakan bahwa data berdimensi tinggi dapat menyebabkan jarak antar titik menjadi kurang representatif. Kondisi tersebut menyebabkan algoritma berbasis jarak dan kepadatan, seperti DBSCAN, mengalami kesulitan dalam membedakan area cluster dan noise secara optimal. Dengan mereduksi dimensi dari 14 fitur menjadi 5 dimensi, UMAP mampu mempertahankan struktur penting data sehingga DBSCAN dapat membentuk cluster yang lebih kompak, lebih terpisah, dan memiliki tingkat noise yang lebih rendah dibandingkan metode tanpa reduksi dimensi.

Peningkatan performa DBSCAN+UMAP menunjukkan bahwa proses reduksi dimensi berhasil menghasilkan representasi data yang lebih sesuai untuk clustering berbasis kepadatan. Pada skenario tanpa reduksi dimensi, DBSCAN bekerja pada ruang fitur berdimensi 14 sehingga hubungan jarak antar data menjadi kurang representatif akibat fenomena *curse of dimensionality*. Kondisi ini menyebabkan proses identifikasi area berkepadatan tinggi menjadi kurang optimal. Sebaliknya, UMAP mereduksi data menjadi 5 dimensi dengan tetap mempertahankan struktur lokal data yang penting. Representasi yang lebih ringkas ini membuat pola kepadatan data menjadi lebih jelas sehingga DBSCAN dapat membentuk cluster yang lebih kompak dan lebih terpisah. Hal tersebut tercermin dari peningkatan Silhouette Score, peningkatan Calinski-Harabasz Index, penurunan Davies-Bouldin Index, serta berkurangnya noise ratio menjadi 0%.

V. KESIMPULAN DAN SARAN

Penelitian ini membandingkan kinerja algoritma DBSCAN pada data akademik mahasiswa dengan dan tanpa reduksi dimensi menggunakan UMAP. Hasil penelitian menunjukkan bahwa kedua metode mampu menghasilkan tiga cluster utama, namun DBSCAN+UMAP memberikan performa yang lebih baik pada enam dari delapan metrik evaluasi yang digunakan. Peningkatan paling signifikan terlihat pada Silhouette Score yang meningkat dari 0,6956 menjadi 0,8382, Calinski-Harabasz Index yang meningkat dari 5.710,95 menjadi 17.156,43, serta Davies-Bouldin Index yang menurun dari 0,4031 menjadi 0,1941. Selain itu, DBSCAN+UMAP berhasil menghilangkan noise sehingga menghasilkan noise ratio sebesar 0%.

Hasil penelitian menunjukkan bahwa reduksi dimensi menggunakan UMAP mampu mengurangi dampak curse of dimensionality dan meningkatkan kualitas clustering pada data akademik mahasiswa. Oleh karena itu, kombinasi UMAP dan DBSCAN direkomendasikan untuk segmentasi profil mahasiswa pada data berdimensi tinggi. Penelitian selanjutnya dapat membandingkan UMAP dengan metode reduksi dimensi lain seperti PCA atau t-SNE serta menguji performanya pada algoritma clustering yang berbeda.

DAFTAR PUSTAKA

[1] E. Alzabidi, Ö. Yakut, S. Saad, and O. Fındık, "SOD-FE: A supervised outlier detection and

feature engineering approach for student dropout prediction," *Sci. Rep.*, vol. 16, Art. no. 14616, 2026, doi: 10.1038/s41598-026-56591-6.

[2] O. Ovtšarenko, "Using AI to forecast student dropout risk in technical education using a learning analytics approach," *Sci. Rep.*, vol. 16, Art. no. 14616, 2026, doi: 10.1038/s41598-026-44919-1.

[3] R. D. Delena, N. J. Dia, R. R. Sacayan, J. C. Sieras, S. A. Khalid, A. H. T. Macatotong, and S. B. Gulam, "Predicting student retention: A comparative study of machine learning approach utilizing sociodemographic and academic factors," *Syst. Soft Comput.*, vol. 7, Art. no. 200352, 2025, doi: 10.1016/j.sasc.2025.200352.

[4] I. Assent, "Clustering high dimensional data," *WIREs Data Min. Knowl. Discov.*, vol. 2, no. 4, pp. 340–350, Jul./Aug. 2012, doi: 10.1002/widm.1062.

[5] X. Xu, W. Chen, and Y. Sun, "Over-sampling algorithm for imbalanced data classification," *J. Syst. Eng. Electron.*, vol. 30, no. 6, pp. 1182–1191, Dec. 2019, doi: 10.21629/JSEE.2019.06.12.

[6] Y. Wang and D. Z. Wang, "Learned accelerator framework for angular-distance-based high-dimensional DBSCAN," *arXiv preprint arXiv:2302.03136*, 2023.

[7] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[8] M. Herrmann, D. Kazempour, F. Scheipl, and P. Kröger, "Enhancing cluster analysis via topological manifold learning," *Data Min. Knowl. Discov.*, vol. 38, no. 3, pp. 840–887, May 2024, doi: 10.1007/s10618-023-00980-2. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2207.00510

[9] Peng, Dehua, Zhipeng Gui, and Huayi Wu. "Interpreting the curse of dimensionality from distance concentration and manifold effect." *arXiv preprint arXiv:2401.00422* (2023).

[10] Liu, Peng, et al. "Outcome-guided disease subtyping for high-dimensional omics data." *arXiv preprint arXiv:2007.11123* (2020).

[11] McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." *arXiv preprint arXiv:1802.03426* (2018).

[12] Islam, Mohammad Tariqul, and Jason W. Fleischer. "Outlier detection in large radiological datasets using umap." *International Workshop on Topology-and Graph-Informed Imaging Informatics*. Cham: Springer Nature Switzerland, 2024.

[13] Amigó, Enrique, et al. "A comparison of extrinsic clustering evaluation metrics based on formal constraints." *Information retrieval* 12.4 (2009): 461-486.

[14] Wafa, Moh. *Evaluasi kinerja akademik mahasiswa menggunakan algoritma clustering*. Diss. Universitas Islam Negeri Maulana Malik Ibrahim, 2013.

[15] M. Martins et al., "Predict Students' Dropout and Academic Success Dataset," UCI Machine Learning Repository, 2021.

[16] [16] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," Proc. ICDM, 2008.

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," Proc. KDD, 1996.

KLASIFIKASI TINGKAT KESUBURAN TANAH MENGGUNAKAN ARTIFICIAL NEURAL NETWORK (ANN) DENGAN ARSITEKTUR MLP BERBASIS DATA KANDUNGAN UNSUR HARA DAN FAKTOR LINGKUNGAN

Pudael Zikri [1)], Christian Hadi Candra [2)] Muhammad Ihdal Fahroni [3)]
1, 2, 3) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: christahadi@gmail.com

Abstrak

Kesuburan tanah merupakan faktor krusial yang menentukan produktivitas pertanian, namun penilaiannya secara konvensional melalui uji laboratorium membutuhkan waktu lama dan biaya tinggi. Penelitian ini bertujuan merancang dan mengimplementasikan model *Artificial Neural Network* (ANN) dengan arsitektur *Multilayer Perceptron* (MLP) berbasis *framework* PyTorch untuk mengklasifikasikan tingkat kesuburan tanah ke dalam dua kategori, yaitu *High* dan *Moderate*. *Dataset* yang digunakan adalah *Soil Nutrients* sebanyak 15.400 sampel dengan 16 fitur *input* yang mencakup kandungan unsur hara makro (Nitrogen, Fosfor, Kalium) dan faktor lingkungan (suhu, curah hujan, kelembaban, intensitas cahaya, dan pH). Mengingat distribusi kelas pada *dataset* bersifat tidak seimbang (rasio 1,75:1), penelitian ini mengevaluasi empat skenario penanganan data tidak seimbang, yaitu *Baseline*, *Undersampling*, *Oversampling*, dan *Synthetic Minority Over-sampling Technique* (SMOTE). Arsitektur MLP yang dirancang terdiri atas satu *hidden layer* berisi 64 *neuron* dengan fungsi aktivasi ReLU, dioptimasi menggunakan St*ochastic Gradient Descent* (SGD) dan fungsi kerugian *Cross-Entropy Loss* selama 100 *epoch*. Hasil pengujian menunjukkan model *Baseline* mencapai akurasi tertinggi sebesar 99,22%, sementara ketiga model dengan teknik penyeimbangan kelas (*Undersampling*, *Oversampling*, dan SMOTE) menghasilkan akurasi pada kisaran 98,34%–98,47% dengan *recall* kelas *High* mencapai 100%, yang berarti seluruh lahan berkesuburan tinggi berhasil terdeteksi tanpa ada yang terlewat. Temuan ini menunjukkan bahwa model berbasis penyeimbangan kelas lebih sesuai diterapkan pada sistem rekomendasi pertanian presisi yang mengutamakan sensitivitas deteksi lahan subur, sedangkan model *Baseline* lebih efisien secara komputasional ketika distribusi data mencerminkan kondisi nyata di lapangan.

Kata kunci: ANN; *imbalanced* data; kesuburan tanah; MLP; PyTorch; SMOTE

Abstract

Soil fertility is a crucial factor determining agricultural productivity, yet conventional laboratory-based assessment is time-consuming and costly. This study aims to design and implement an Artificial Neural Network (ANN) model with a Multilayer Perceptron (MLP) architecture built on the PyTorch framework to classify soil fertility level into two categories, namely High and Moderate. The Soil Nutrients dataset used consists of 15,400 samples with 16 input features covering macro nutrient content (Nitrogen, Phosphorus, Potassium) and environmental factors (temperature, rainfall, humidity, light intensity, and pH). Since the class distribution in the dataset is imbalanced (ratio of 1.75:1), this study evaluates four scenarios for handling imbalanced data, namely Baseline, Undersampling, Oversampling, and the Synthetic Minority Over-sampling Technique (SMOTE). The MLP architecture designed consists of a single hidden layer with 64 neurons using the ReLU activation function, optimized with Stochastic Gradient Descent (SGD) and Cross-Entropy Loss over 100 epochs. The test results show that the Baseline model achieved the highest accuracy of 99.22%, while the three class-balancing models (Undersampling, Oversampling, and SMOTE) achieved accuracies ranging from 98.34% to 98.47% with a recall of 100% for the High class, meaning that all highly fertile land was successfully detected without any being missed. These findings indicate that class-balancing-based models are more suitable for precision agriculture recommendation systems that prioritize the sensitivity of detecting fertile land, whereas the Baseline model is more computationally efficient when the data distribution reflects real-world field conditions.

Keywords*: ANN; imbalanced data; MLP; PyTorch; SMOTE; soil fertility*

I. PENDAHULUAN

Kesuburan tanah merupakan faktor penentu utama produktivitas sektor pertanian, baik pada skala lokal maupun nasional. Kemampuan tanah dalam menyediakan unsur hara esensial seperti Nitrogen (N), Fosfor (P), dan Kalium (K), yang dikombinasikan dengan faktor lingkungan seperti pH, suhu, curah hujan, kelembaban, dan intensitas cahaya, secara kolektif menentukan kapasitas tanah dalam menopang pertumbuhan tanaman secara optimal. Dalam era pertanian modern, optimalisasi pengelolaan lahan pertanian mulai banyak mengintegrasikan teknologi

kecerdasan buatan untuk memantau kondisi lingkungan serta memprediksi kualitas vegetasi secara lebih terukur [1]. Khususnya untuk unsur hara makro seperti Nitrogen, identifikasi ketersediaannya secara dini sangat krusial karena keterbatasan unsur ini secara langsung dapat menurunkan kualitas pertumbuhan vegetatif tanaman secara signifikan [2].

Penilaian tingkat kesuburan tanah secara konvensional umumnya masih bergantung pada analisis laboratorium kimia. Meskipun metode ini memiliki akurasi tinggi, pendekatan konvensional membutuhkan prosedur pengujian yang memakan waktu lama, biaya operasional yang tinggi, serta ketergantungan yang besar pada sumber daya manusia yang terlatih. Keterbatasan logistik dan waktu ini menjadi kendala signifikan bagi para petani serta institusi agrikultur dalam mengambil keputusan pemupukan dan pengelolaan lahan secara cepat dan tepat. Oleh karena itu, diperlukan transformasi digital melalui metode komputasi otomatis yang mampu mengklasifikasikan tingkat kesuburan tanah secara cepat dan akurat berdasarkan parameter hara yang tersedia. Penggunaan komputasi cerdas ini sejalan dengan tren pengembangan sistem klasifikasi karakteristik lahan guna mendukung efisiensi sistem rekomendasi pertanian modern [3].

Dalam ranah kecerdasan buatan, *Artificial Neural Network* (ANN) dengan arsitektur *Multilayer* Perceptron (MLP) menjadi salah satu algoritma *machine learning* yang terbukti efektif untuk menyelesaikan permasalahan klasifikasi dan analisis multivariat. Kemampuan MLP dalam mempelajari pola non-linear dari data berdimensi tinggi menjadikannya pendekatan yang ideal untuk mengolah karakteristik data tanah yang kompleks, saling berkorelasi, dan multidimensi. Eksperimen terdahulu menunjukkan bahwa model berbasis *neural network* dan MLP memiliki kinerja serta adaptibilitas yang sangat baik dalam memprediksi hasil panen maupun mengidentifikasi dinamika kadar nutrisi spesifik pada karakteristik lahan jika dibandingkan dengan metode statistik tradisional [4], [2].

Penelitian ini bertujuan merancang dan mengimplementasikan model ANN-MLP berbasis *framework* PyTorch untuk mengklasifikasikan tingkat kesuburan tanah menjadi dua kategori target, yaitu *High* (Tinggi) dan *Moderate* (Sedang). Eksperimen dilakukan menggunakan dataset *Soil Nutrients* yang mencakup 15.400 sampel data dengan melibatkan 16 fitur input lingkungan dan hara setelah melalui tahapan pra-pemrosesan data. Mengingat distribusi data di lapangan sering kali menunjukkan ketidakseimbangan kelas (*imbalanced data*) yang dapat membiaskan hasil prediksi model terhadap kelas mayoritas, penelitian ini secara khusus mengevaluasi pengaruh empat skenario penanganan ketidakseimbangan data, yaitu *Baseline* (tanpa penyeimbangan), *Undersampling*, *Oversampling*, dan *Synthetic Minority Over-sampling Technique* (SMOTE). Melalui evaluasi komprehensif ini, diharapkan diperoleh model klasifikasi terbaik yang tidak hanya memiliki akurasi tinggi secara keseluruhan, melainkan juga sensitivitas yang seimbang dalam mengenali setiap tingkat kesuburan lahan pertanian.

II. TINJAUAN PUSTAKA

2.1 Kajian Teori

1. Kesuburan Tanah dan Faktor Mikroklimat Lahan

Kesuburan tanah merupakan indikator kesiapan landasan ekologis dalam menyediakan unsur hara esensial secara berimbang untuk mendukung siklus hidup vegetasi secara optimal. Parameter kimia tanah yang paling kritis disangga oleh unsur hara makro primer, yaitu Nitrogen (N), Fosfor (P), dan Kalium (K). Nitrogen bertindak sebagai stimulan utama dalam sintesis protein dan klorofil pada fase vegetatif tanaman. Fosfor memainkan peran sentral dalam transfer energi seluler (ATP) serta merangsang perkembangan sistem perakaran. Sementara itu, Kalium berfungsi sebagai regulator osmotik yang mengatur pembukaan stomata dan efisiensi penyerapan air [2].

Aktivitas penyerapan unsur hara tersebut sangat dipengaruhi oleh faktor lingkungan atau mikroklimat sekitar lahan. Derajat keasaman (pH) tanah menentukan kelarutan unsur hara; nilai pH yang terlalu asam atau basa akan mengikat senyawa hara sehingga mengunci kapasitas serap akar. Komponen lingkungan lain seperti suhu tanah, curah hujan (*rainfall*), intensitas cahaya, durasi penyinaran, dan kelembaban nisbi (*Relative Humidity*) secara kolektif mengontrol laju transpirasi tanaman serta mineralisasi bahan organik oleh mikroorganisme tanah. Oleh karena itu, pengintegrasian parameter hara makro dan

variabel mikroklimat menghasilkan kluster data multivariat yang sangat representatif untuk pemodelan prediksi kesuburan lahan secara presisi [1], [3].

2. Arsitektur *Artificial Neural Network* (ANN) dan *Multilayer Perceptron* (MLP)

Artificial Neural Network (ANN) merupakan paradigma komputasi yang mensimulasikan struktur jaringan saraf biologis untuk memetakan hubungan non-linear yang kompleks antara ruang fitur masukan dan target luaran [2]. *Multilayer Perceptron* (MLP) adalah salah satu varian arsitektur ANN *feedforward* yang paling andal dalam menyelesaikan permasalahan klasifikasi berbasis data tabular berdimensi tinggi. Struktur dasar MLP terdiri dari tiga komponen utama: satu lapisan masukan (input layer), satu atau lebih lapisan tersembunyi (hidden layer), dan satu lapisan luaran (*output* layer) [4].

Mekanisme komputasi pada setiap neuron melibatkan operasi perkalian matriks antara vektor fitur masukan (x) dengan bobot korelasi (w), yang kemudian diakumulasikan dengan parameter bias (b). Transformasi linear ini dirumuskan pada Persamaan (1):

$$z = \sum_{i=1}^{n} w_i \, x_i + b$$

Nilai skalar z tersebut selanjutnya ditransmisikan menuju fungsi aktivasi non-linear untuk menghasilkan sinyal luaran *neuron* yang akan diteruskan ke lapisan berikutnya. Proses pembelajaran model dilakukan melalui kombinasi mekanisme *forward propagation* untuk menghitung nilai prediksi dan *backpropagation* untuk mengalirkan nilai kesalahan (*error*) secara mundur guna memperbarui bobot secara iteratif [2], [6].

3. Fungsi Aktivasi ReLU dan Algoritma *Optimasi Stochastic Gradient Descent* (SGD)

Fungsi aktivasi memegang peranan krusial dalam menyuntikkan sifat non-linearitas ke dalam jaringan saraf, sehingga MLP mampu mempelajari pola-pola rumit yang tidak dapat diselesaikan oleh pembatas linear sederhana. Pada arsitektur modern, *Rectified Linear Unit* (ReLU) menjadi fungsi aktivasi standar yang diterapkan pada lapisan tersembunyi. Secara matematis, ReLU didefinisikan pada Persamaan (2):

$$f(z) = \max(0, z)$$

Karakteristik ReLU yang mengabaikan nilai negatif dan mempertahankan nilai positif secara linear terbukti efektif dalam memitigasi fenomena *vanishing gradient*, yaitu kondisi di mana nilai gradien mengecil secara ekstrem saat proses *backpropagation* yang menghambat pembaruan bobot pada lapisan awal jaringan [6].

Untuk meminimalkan nilai fungsi kerugian (*loss function*), model membutuhkan algoritma optimasi. *Stochastic Gradient Descent* (SGD) merupakan optimizer fundamental yang bekerja dengan cara menghitung gradien dari fungsi kerugian untuk setiap sampel atau *mini-batch* data secara acak. Pembaruan bobot menggunakan SGD dilakukan secara bertahap searah dengan arah negatif dari gradien, sehingga memungkinkan model keluar dari jebakan minimum lokal (*local minima*) dan bergerak menuju titik optimal global (*global optimum*) secara konvergen.

4. Fenomena Ketidakseimbangan Kelas (*Imbalanced Data*) dan Metode Sampling

Ketidakseimbangan kelas (*imbalanced data*) merupakan suatu kondisi di mana representasi jumlah sampel antar kelas target di dalam dataset memiliki disparitas yang sangat timpang. Pada kasus pemetaan karakteristik lahan pertanian, data lapangan sering kali didominasi oleh kelas kondisi umum (kelas mayoritas), sedangkan kondisi lahan yang sangat subur atau sangat kritis berjumlah sangat terbatas (kelas minoritas). Ketidakseimbangan ini menciptakan bias komputasi, di mana model *machine learning* cenderung mengabaikan karakteristik kelas minoritas demi mengoptimalkan nilai akurasi global secara semu [7]. Untuk menanggulangi kendala tersebut, terdapat tiga strategi pendekatan pada level data (*data-level approach*):

a. *Undersampling*: Dilakukan dengan mereduksi jumlah sampel pada kelas mayoritas secara acak hingga setara dengan volume kelas minoritas. Pendekatan ini mempercepat durasi pelatihan model, namun berisiko menghilangkan informasi esensial dari data yang dihapus.
b. *Oversampling*: Bekerja dengan menduplikasi sampel kelas minoritas secara acak agar kuantitasnya seimbang dengan kelas mayoritas. Walaupun informasi data terjaga, replikasi data

identik ini rentan menyebabkan model mengalami *overfitting*.

c. *Synthetic Minority Over-sampling Technique* (SMOTE): Merupakan algoritma penyeimbangan data canggih yang bekerja dengan cara mensintesis sampel baru buatan secara matematis di ruang fitur kelas minoritas [7]. SMOTE beroperasi dengan mencari tetangga terdekat (*k-nearest neighbors*) dari sebuah sampel minoritas asli, kemudian menghasilkan data baru di sepanjang garis lurus yang menghubungkan sampel acak tersebut dengan tetangganya. Metode ini memperluas batas keputusan (*decision boundary*) model secara umum sehingga meningkatkan kemampuan generalisasi jaringan saraf tanpa memicu *overfitting* yang drastis.

5. Metrik Evaluasi Performa Klasifikasi

Penilaian performa model klasifikasi yang diuji pada dataset yang tidak seimbang tidak dapat hanya bersandar pada metrik akurasi murni. Akurasi dapat memberikan estimasi yang menyesatkan apabila model memprediksi seluruh data sebagai kelas mayoritas namun gagal total pada kelas minoritas. Oleh karena itu, evaluasi yang komprehensif memerlukan representasi matriks konfusi (*confusion matrix*) yang memetakan hasil prediksi menjadi empat kuadrat: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [8]. Berdasarkan nilai komponen matriks tersebut, metrik evaluasi yang diturunkan meliputi:

a. Akurasi (*Accuracy*): Mengukur rasio prediksi benar (positif dan negatif) terhadap keseluruhan total data, dirumuskan pada Persamaan (3):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

b. Presisi (Precision): Menunjukkan tingkat ketepatan model dalam mengenali sampel positif di antara seluruh sampel yang diprediksi sebagai positif, dirumuskan pada Persamaan (4):

$$\text{Precision} = \frac{TP}{TP + FP}$$

c. Recall / Sensitivitas: Mengukur kemampuan model dalam menjaring kembali seluruh sampel yang sejatinya termasuk dalam kategori kelas positif, dirumuskan pada Persamaan (5):

$$\text{Recall} = \frac{TP}{TP + FN}$$

d. F1-*Score*: Nilai rata-rata harmonik yang menyeimbangkan metrik presisi dan *recall*, menjadi parameter mutlak untuk menilai ketangguhan model pada kondisi *imbalanced data*, dirumuskan pada Persamaan (6):

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.2 Literatur (Penelitian Terdahulu)

Penelitian mengenai penerapan *Artificial Neural Network* (ANN), *Multilayer Perceptron* (MLP), dan metode *Deep Learning* untuk klasifikasi serta pemodelan prediksi telah banyak dilakukan sebelumnya. Kajian literatur ini memetakan posisi penelitian yang sedang dilakukan di antara studi-studi terdahulu.

Pertama, penelitian oleh Khasanah [1] serta Triyogi, Magdalena, dan Hidayat [5] mengkaji tentang pemanfaatan *Artificial Intelligence* (AI) dalam optimalisasi penyimpanan pasca-panen hasil pertanian berbasis monitoring lingkungan serta deteksi tingkat kematangan buah kelapa sawit menggunakan *Convolutional Neural Network* (CNN). Hasil penelitian tersebut menunjukkan bahwa integrasi algoritma cerdas mampu meningkatkan efisiensi operasional dan mereduksi potensi kerusakan komoditas agrikultur secara terukur. Persamaan dengan penelitian ini terletak pada pemanfaatan komputasi cerdas untuk meningkatkan produktivitas dan optimalisasi di sektor pertanian. Sementara perbedaannya adalah penelitian terdahulu berfokus pada monitoring fisik objek pasca-panen dan fitur citra buah, sedangkan penelitian ini menerapkan pendekatan jaringan saraf pada fase pra-tanam untuk memetakan kesuburan lahan.

Kedua, penelitian oleh Wahyuni [3] serta Fitri dan Nugraha [4] berfokus pada klasifikasi jenis tanah untuk rekomendasi tanaman menggunakan CNN serta optimasi kinerja MLP untuk prediksi hasil panen tanaman pangan. Hasil penelitian membuktikan bahwa arsitektur jaringan saraf tiruan mampu merekam fluktuasi variabel lingkungan dengan tingkat galat (eror) prediktif yang jauh lebih rendah dibandingkan metode regresi tradisional. Persamaan dengan penelitian ini terletak pada penggunaan arsitektur jaringan

saraf (CNN/MLP) untuk mengolah variabel karakteristik lahan pertanian. Perbedaannya adalah penelitian terdahulu berfokus pada pemrosesan citra jenis tanah dan prediksi regresi nilai kontinu hasil panen, sedangkan penelitian ini secara khusus mengkaji klasifikasi biner tingkat kesuburan tanah (*High* dan *Moderate*) dengan menitikberatkan pada penanganan ketidakseimbangan kelas (*imbalanced data*).

Ketiga, penelitian oleh Anggraini [2] mengaplikasikan metode Jaringan Saraf Tiruan dengan algoritma *Backpropagation* untuk mengidentifikasi kadar Nitrogen (N-total) pada lahan marginal. Hasil penelitian tersebut menunjukkan bahwa JST efektif memetakan hubungan non-linear kondisi lingkungan tanpa harus bergantung sepenuhnya pada pengujian laboratorium konvensional yang memakan waktu lama. Persamaan dengan penelitian ini terletak pada penggunaan rumpun JST untuk memetakan ketersediaan unsur hara makro pada tanah. Perbedaannya adalah penelitian terdahulu berfokus pada estimasi kuantitatif variabel hara tunggal (Nitrogen) pada lahan marginal, sedangkan penelitian ini mengklasifikasikan kluster hara lengkap (N, P, K) beserta 5 faktor lingkungan secara simultan menggunakan MLP berbasis PyTorch, serta mengevaluasi efisiensi empat teknik rekayasa sampling.

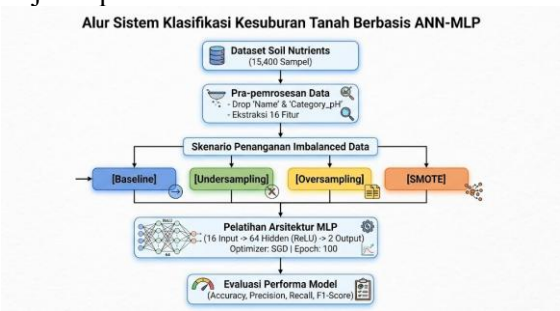
Keempat, penelitian oleh Sukmawati dan Pujiyanta [9], Mardianto dan Pratiwi [10], serta Liawrungrueang dkk. [15] mengimplementasikan model ANN dan algoritma *Backpropagation* dalam mendeteksi serta mengklasifikasikan penyakit kelainan tulang seperti *osteoporosis* dan fraktur kompresi vertebra. Hasil penelitian membuktikan ketangguhan algoritma jaringan saraf tiruan dalam mengekstrak ruang fitur yang kompleks dan mengenali pola asimetris pada objek yang diuji. Persamaan dengan penelitian ini adalah pemanfaatan struktur ANN untuk menyelesaikan masalah klasifikasi multivariat dengan tingkat akurasi tinggi. Perbedaannya terletak pada domain lokus penelitian; penelitian terdahulu mengolah data biomedis berbasis ekstraksi fitur citra rontgen (X-*ray*), sementara penelitian ini mentransfer kapabilitas komputasi tersebut ke domain agrikultur untuk klasifikasi data tabular unsur hara.

Kelima, penelitian oleh Wong dkk. [11], Pangestu dan Purnama [12], Nguyen dkk. [13], serta Xu dkk. [14] mengembangkan model pembelajaran mendalam (*deep learning*) seperti DenseNet dan CNN untuk pengukuran otomatis sudut kelengkungan spinal (skoliosis, kifosis, lordosis) serta klasifikasi derajat fraktur tulang belakang. Hasil penelitian menunjukkan bahwa digitalisasi pengukuran otomatis mampu mereduksi subjektivitas analisis manual serta memangkas latensi waktu tunggu diagnosis secara signifikan. Persamaan dengan penelitian ini adalah penggunaan arsitektur cerdas berlapis untuk mengenali deviasi kriteria objek yang memiliki variansi data yang tinggi. Perbedaannya, penelitian terdahulu memproses data spasial dari citra medis 2D, sedangkan penelitian ini berfokus pada optimasi performa model pada data tabular numerik tanah melalui penanggulangan bias sebaran sampel lapangan menggunakan algoritma SMOTE, *Oversampling*, dan *Undersampling*.

III. METODOLOGI PENELITIAN

3.1 Alur Penelitian

Metodologi yang digunakan dalam penelitian ini dirancang secara sistematis untuk menjamin validitas dan reproduksibilitas model klasifikasi tingkat kesuburan tanah. Secara garis besar, tahapan eksperimen dibagi menjadi lima fase utama, yaitu: (1) Akuisisi dataset *Soil Nutrients*; (2) Pra-pemrosesan data (*data preprocessing*) yang meliputi eliminasi fitur bias dan penanganan data; (3) Rekayasa penyeimbangan kelas (*class balancing*) menggunakan empat skenario berbeda; (4) Pelatihan model *Artificial Neural Network* (ANN) dengan arsitektur *Multilayer Perceptron* (MLP) menggunakan *framework* PyTorch; dan (5) Evaluasi serta komparasi performa model melalui matriks konfusi. Alur komprehensif penelitian ini disajikan pada Gambar 1.



Gambar 1. Diagram Alur Sistem Klasifikasi Kesuburan Tanah Berbasis ANN-MLP.

3.2 *Dataset* dan Pra-pemrosesan Data

Penelitian ini memanfaatkan *dataset Soil Nutrients* yang mencakup total 15.400 sampel data tabular. Pada kondisi awal, dataset ini memiliki 19 atribut yang mengombinasikan parameter kimia tanah, faktor lingkungan, serta identitas administratif. Guna meningkatkan performa komputasi dan menghindari bias model terhadap fitur-fitur non-kontributif, dilakukan tahapan seleksi fitur (*feature selection*) berupa eliminasi kolom "`Name`" dan "`Category_pH`." Kolom "`Name`" dieliminasi karena hanya bersifat pengidentifikasi tekstual acak yang tidak merepresentasikan karakteristik fisik tanah, sedangkan "`Category_pH`" dihapus untuk mencegah redundansi informasi (*multicollinearity*) karena representasi nilai keasaman tanah telah diakomodasi secara presisi oleh fitur numerik "`pH`."

Setelah fase eliminasi, diperoleh 16 fitur input utama yang terdiri dari parameter kandungan unsur hara makro (Nitrogen, Phosphorus, Potassium), indikator lingkungan mikroklimat (*Temperature*, *Rainfall*, *Light_Hours*, *Light_Intensity*, *Relative Humidity*), serta faktor agronomis pelengkap lainnya. Seluruh fitur numerik tersebut kemudian divalidasi dari keberadaan data kosong (*missing values*) dan dikonversi ke dalam bentuk tensor agar dapat diproses secara optimal oleh arsitektur *deep learning*.

3.3 Skenario Penanganan Ketidakseimbangan Kelas

Eksplorasi data menunjukkan adanya ketimpangan distribusi yang signifikan pada kelas target "`Fertility`", di mana sampel diklasifikasikan ke dalam dua kategori biner: *High* (Tinggi) dan *Moderate* (Sedang). Untuk menguji ketangguhan model dalam mengenali kelas minoritas tanpa mengorbankan akurasi global, penelitian ini merancang empat skenario penyeimbangan data pada level sampel:

1. Skenario I (*Baseline*): Model dilatih langsung menggunakan dataset asli tanpa adanya perlakuan penyeimbangan distribusi kelas target. Skenario ini berfungsi sebagai kontrol pembanding untuk melihat dampak laten dari data yang timpang terhadap kecenderungan bias model.
2. Skenario II (*Undersampling*): Melakukan reduksi jumlah sampel pada kelas mayoritas secara acak (*Random Under Sampler*) hingga volume datanya setara dengan kuantitas sampel pada kelas minoritas.
3. Skenario III (*Oversampling*): Menggunakan teknik *Random Over Sampler* untuk mereplikasi dan menduplikasi sampel pada kelas minoritas secara acak hingga mencapai keseimbangan kuantitas dengan kelas mayoritas.
4. Skenario IV (SMOTE): Menerapkan algoritma *Synthetic Minority Over-sampling Technique* untuk mensintesis data buatan baru di ruang fitur kelas minoritas berdasarkan pendekatan jarak *k-nearest neighbors*, sehingga memperluas batas keputusan model secara objektif [7].

3.4 Perancangan Arsitektur MLP berbasis PyTorch

Pemodelan klasifikasi dibangun menggunakan jaringan saraf tiruan bertipe *Multilayer Perceptron* (MLP) yang diimplementasikan melalui pustaka *open-source* PyTorch [16]. Arsitektur jaringan yang dirancang terdiri dari struktur tiga lapis terintegrasi dengan spesifikasi teknis sebagai berikut:

1. Lapisan Masukan (*Input Layer*): Terdiri dari 16 *neuron* yang merepresentasikan keseluruhan dimensi fitur parameter tanah dan lingkungan hasil pra-pemrosesan.
2. Lapisan Tersembunyi (*Hidden Layer*): Menggunakan satu lapisan tersembunyi tunggal (*single hidden layer*) yang padat (*fully connected*) dengan kapasitas 64 *neuron*. Pemilihan 64 *neuron* didasarkan pada trade-off optimal antara kapasitas ekspresi model dalam menangkap pola non-linear dan efisiensi waktu komputasi. Lapisan ini dilengkapi dengan fungsi aktivasi *Rectified Linear Unit* (ReLU) untuk mempercepat konvergensi gradien [6].
3. Lapisan Luaran (*Output Layer*): Memiliki 2 *neuron* yang merepresentasikan probabilitas kelas target, yaitu *High* dan *Moderate*.

Proses pembelajaran (*training*) pada seluruh skenario model dikendalikan oleh fungsi

kerugian *Cross-Entropy Loss* yang dikombinasikan dengan algoritma optimasi *Stochastic Gradient Descent* (SGD). Pembelajaran dilakukan secara berulang sepanjang 100 *epoch* untuk memastikan seluruh bobot dan bias pada jaringan saraf telah mengalami pembaruan (*backpropagation*) secara konvergen dan mencapai titik minimum global (*global minimum*).

3.5 Instrumen Evaluasi Performa

Kinerja dari keempat skenario model ANN-MLP diukur secara kuantitatif menggunakan metrik evaluasi yang diturunkan dari matriks konfusi (*confusion matrix*). Komponen pengujian terdiri dari *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), di mana kelas *High* ditetapkan sebagai representasi target positif. Metrik utama yang digunakan untuk menganalisis dan membandingkan performa model meliputi:

1. Akurasi Global (*Accuracy*): Menilai persentase ketepatan prediksi total model terhadap keseluruhan data tanah [Persamaan 3].
2. Presisi (*Precision*): Mengukur ketepatan model dalam memprediksi status kesuburan *High* di antara keseluruhan sampel yang diidentifikasi *High* oleh sistem [Persamaan 4].
3. Recall (Sensitivitas): Menilai efisiensi model dalam menjaring kembali seluruh lahan yang pada kenyataannya memiliki tingkat kesuburan *High* [Persamaan 5].
4. F1-*Score*: Rata-rata harmonik antara presisi dan *recall* yang menjadi parameter mutlak dalam menentukan skenario penyeimbangan data terbaik untuk pengelolaan tanah pertanian [Persamaan 6].

IV. HASIL DAN PEMBAHASAN

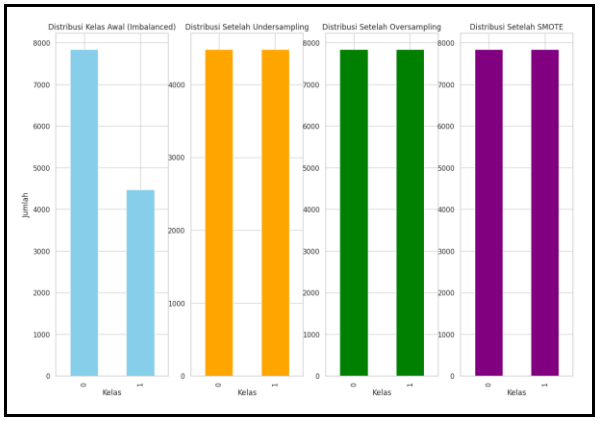
Bagian ini menguraikan hasil eksperimen secara komprehensif, mulai dari karakteristik dataset, hasil penanganan *outlier*, proses training model ANN-MLP pada empat skenario, hingga analisis perbandingan performa antar model berdasarkan metrik evaluasi yang telah ditetapkan.

4.1 Karakteristik *Dataset*

Dataset *Soil Nutrients* terdiri dari 15.400 sampel, 19 kolom original, tanpa missing values maupun duplikat. Setelah *preprocessing*, total fitur *input* menjadi 20 setelah *One-Hot Encoding*. Distribusi kelas awal menunjukkan ketidakseimbangan: kelas *Moderate* berjumlah 9.800 sampel dan *High* berjumlah 5.600 sampel, menghasilkan rasio *imbalance* ~1,75:1 pada data training. Tabel 1 merangkum distribusi kelas pada setiap skenario penyeimbangan.

Tabel 1. Distribusi Kelas Training per Skenario

Skenario	*Moderate*	*High*	Total	Rasio
Baseline	7.840	4.480	12.320	1,75:1
Undersampling	4.480	4.480	8.960	1:1
Oversampling	7.840	7.840	15.680	1:1
SMOTE	7.840	7.840	15.680	1:1



Gambar 2. Visualisasi Distribusi Kelas Target pada Setiap Skenario Penyeimbangan

4.2 Hasil Penanganan *Outlier*

Proses IQR *Capping* mengidentifikasi dan menangani *outlier* pada delapan fitur: Nitrogen (1.904 data), Yield (1.110), pH (1.047), Phosphorus (783), Potassium (731), *Temperature* (713), Light_Intensity (700), dan *Rainfall* (680). Fitur Rh dan Light_Hours tidak memiliki *outlier* signifikan. Penerapan batas fisiologis tambahan memastikan nilai tetap dalam rentang yang secara biologis masuk akal bagi kondisi tanah pertanian.

4.3 Hasil *Training* Model

Seluruh model menunjukkan konvergensi yang konsisten selama 100 *epoch*. Model *Baseline* memulai dengan *training loss* 0,7220 dan akurasi 56,10% pada *epoch* pertama, kemudian konvergen hingga loss 0,0998 dan akurasi 99,22% pada *epoch*

ke-100. Pola konvergensi yang pesat ini didorong oleh distribusi kelas yang tidak seimbang yang menguntungkan kelas mayoritas (*Moderate*).

Model dengan teknik penyeimbangan kelas menunjukkan konvergensi yang lebih lambat di awal karena distribusi kelas yang seimbang memaksa model belajar secara merata dari kedua kelas. Namun pada epoch ke-100, ketiga model *balanced* berhasil mencapai akurasi test di atas 98%, menunjukkan kemampuan generalisasi yang baik. Tabel 2 merangkum perbandingan performa keempat model.

Tabel 2. Perbandingan Performa Keempat Model

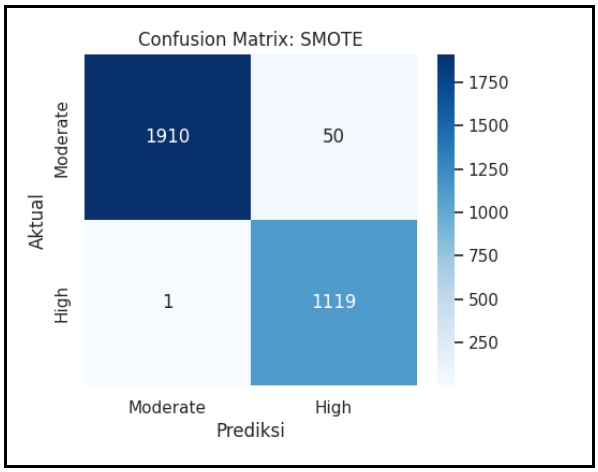
Model	Akurasi(%)	*Precision*(M/H)	*Recall*(M/H)	F1(M/H)	Epoch Konvergen
Baseline	99,22	0,99/0,99	0,99/0,99	0,99/0,99	≤70
Undersampling	98,47	1,00/0,96	0,98/1,00	0,99/0,98	≤80
Oversampling	98,41	1,00/0,96	0,98/1,00	0,99/0,98	≤85
SMOTE	98,34	1,00/0,96	0,97/1,00	0,99/0,98	≤90

**M = Moderate, H = High*

4.4 Pengujian Skenario dan Pembahasan

Analisis Model *Baseline* mencapai akurasi tertinggi (99,22%) karena distribusi *imbalanced* menguntungkan kelas mayoritas. Meskipun demikian, *classification report* menunjukkan *Baseline* tetap mampu mengklasifikasikan kelas *High* dengan precision dan *recall* 0,99. *Training loss* yang turun dari 0,7220 ke 0,0998 mengkonfirmasi optimasi SGD berjalan tanpa indikasi *overfitting* signifikan.

Analisis model *balanced* dengan model *undersampling*, *oversampling*, dan SMOTE menghasilkan *recall* kelas *High* = 1,00 (100%), artinya seluruh sampel tanah kesuburan *High* pada data uji terdeteksi tanpa ada yang terlewat (*False Negative* = 0). Hal ini sangat krusial dalam konteks pertanian presisi, di mana melewatkan identifikasi lahan subur dapat mengakibatkan alokasi sumber daya pemupukan yang tidak optimal.



Gambar 3. Matriks Konfusi (*Confusion Matrix*) pada Model dengan Skenario SMOTE

Perbandingan Teknik *Balancing*, yaitu *Undersampling* memberikan akurasi sedikit lebih tinggi (98,47%) dibandingkan *Oversampling* (98,41%) dan SMOTE (98,34%), namun perbedaannya tidak signifikan secara praktis. *Undersampling* yang mengurangi data training dari 12.320 menjadi 8.960 sampel masih mempertahankan performa kompetitif, mengindikasikan redundansi data yang tinggi dalam *dataset* ini.

Keunggulan SMOTE, Meskipun menghasilkan akurasi sedikit lebih rendah, SMOTE memiliki keunggulan konseptual karena sampel sintetis yang dihasilkan melalui interpolasi K-NN menghasilkan keragaman data lebih representatif. Pada dataset yang lebih kompleks atau berisik, SMOTE umumnya memberikan generalisasi lebih baik karena *decision boundary* model menjadi lebih umum dan tidak bergantung pada titik data tunggal.

Implikasi Praktis untuk sistem rekomendasi pemupukan yang mengutamakan tidak terlewatnya lahan kesuburan *High*, model dengan teknik penyeimbangan kelas (terutama SMOTE atau *Undersampling*) lebih direkomendasikan. Sebaliknya, jika prioritas adalah akurasi keseluruhan pada distribusi data nyata yang *imbalanced*, model *Baiseline* lebih efisien secara komputasional.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan model ANN-MLP

menggunakan PyTorch untuk mengklasifikasikan tingkat kesuburan tanah. Arsitektur sederhana dengan satu *hidden layer* (64 *neuron*) dan fungsi aktivasi ReLU terbukti mampu mencapai performa tinggi. Model *Baseline* mencapai akurasi tertinggi 99,22%, sementara model dengan teknik penyeimbangan kelas mencapai 98,34–98,47% dengan *recall* kelas *High* sempurna (100%). Pemilihan teknik penanganan *imbalance* sebaiknya disesuaikan kebutuhan aplikasi; untuk diagnosis pertanian yang mengutamakan tidak terlewatnya tanah subur, model *balanced* lebih direkomendasikan. Penelitian selanjutnya dapat mengeksplorasi arsitektur *deep* MLP, *optimizer* Adam, teknik regularisasi (*Dropout*, *Batch Normalization*), serta validasi menggunakan data lapangan dari berbagai wilayah geografis di Indonesia.

DAFTAR PUSTAKA

[1] R. N. Khasanah, "*Systematic Literature Review: Pemanfaatan Artificial Intelligence Dalam Optimalisasi Penyimpanan Pasca Panen Hasil Pertanian Berbasis Monitoring Lingkungan dan Prediksi Kerusakan Produk*," *Jurnal Media Akademik (JMA)*, vol. 4, no. 4, 2026.

[2] R. Anggraini, "*Identifikasi Kadar Nitrogen pada Lahan Marginal Menggunakan Metode Jaringan Saraf Tiruan (Backpropagation),"* Tesis, Program Pascasarjana Magister Teknik Elektro, Universitas Lampung, Bandar Lampung, 2025.

[3] Y. S. Wahyuni, "*Implementasi Convolutional Neural Network untuk Klasifikasi Jenis Tanah dalam Sistem Rekomendasi Jenis Tanaman Berbasis Website*," Skripsi, Program Studi Teknik Informatika, Universitas Nusa Putra, Sukabumi, 2025.

[4] E. Fitri dan S. N. Nugraha, "*Optimasi Kinerja Linear Regression, Random Forest Regression dan Multilayer Perceptron pada Prediksi Hasil Panen*," *INTI NUSA MANDIRI*, vol. 18, no. 2, 2024.

[5] R. Triyogi, R. Magdalena, dan B. Hidayat, "*Mendeteksi Kematangan Buah Kelapa Sawit Menggunakan Convolution Neural Network Deep Learning*," *Jurnal Nasional Sains dan Teknik*, vol. 1, no. 1, hlm. 22-27, 2023.

[6] Y. LeCun, Y. Bengio, dan G. Hinton, "*Deep learning*," *Nature*, vol. 521, no. 7553, hlm. 436-444, 2015.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, dan W. P. Kegelmeyer, "*SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research*, vol. 16, hlm. 321-357, 2002.

[8] I. H. Witten, E. Frank, M. A. Hall, dan C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Cambridge, MA: Morgan Kaufmann, 2016.

[9] K. Sukmawati dan A. Pujiyanta, "*Deteksi Penyakit Tulang Menggunakan Jaringan Syaraf Tiruan dengan Metode Backpropagation*," *Jurnal Sarjana Teknik Informatika*, vol. 2, no. 2, hlm. 1313-1320, 2014.

[10] I. Mardianto dan D. Pratiwi, "*Sistem Deteksi Penyakit Pengeroposan Tulang dengan Metode Jaringan Syaraf Tiruan Backpropagation dan Representasi Ciri dalam Ruang Eigen*," *Jurnal Teknologi Industri*, hlm. 71-79, 2013.

[11] J. Wong, M. Reformat, E. Parent, dan E. Lou, "*Using machine learning to automatically measure kyphotic and lordotic angle measurements on radiographs for children with adolescent idiopathic scoliosis*," *Medical Engineering and Physics*, vol. 130, hlm. 104202, 2024.

[12] S. D. Pangestu dan S. I. Purnama, "*Pendeteksi Sudut Kemiringan Tulang Pada Penderita Skoliosis Menggunakan Image Processing*," *e-Proceeding of Engineering*, vol. 12, no. 4, hlm. 5466-5473, 2025.

[13] T. P. Nguyen, J.-H. Kim, S.-H. Kim, J. Yoon, dan S.-H. Choi, "*Machine Learning-Based Measurement of Regional and Global Spinal Parameters Using the Concept of Incidence Angle of Inflection Points*," *Bioengineering*, vol. 10, no. 10, hlm. 1236, 2023.

[14] F. Xu et al., "*Deep learning-based artificial intelligence model for classification of vertebral compression fractures: A multicenter diagnostic study*," *Frontiers in Endocrinology*, vol. 14, hlm. 1025749, 2023.

[15] W. Liawrungrueang et al., "*Osteoporotic vertebral compression fracture (OVCF) detection using artificial neural networks model based on the AO spine-DGOU osteoporotic fracture classification system*," *North American Spine Society Journal (NASSJ)*, vol. 19, hlm. 100515, 2024.

[16] A. Paszke et al., "*PyTorch: An imperative style, high-performance deep learning library*," dalam *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, hlm. 8026-8037, 2019. *(Ditambahkan khusus sebagai referensi resmi implementasi PyTorch pada sub-bab D)*

EVALUASI KINERJA *FAILOVER CLUSTERING* BERBASIS HAPROXY UNTUK MENINGKATKAN *HIGH AVAILABILITY WEB SERVER* PADA LINGKUNGAN *CLOUD*

Nur Ayu Widianingsih [1)], Jafaruddin Gusti Amri Ginting [2)], Eko Fajar Cahyadi [3)]
[1,2,3)] Program Studi Teknik Telekomunikasi, Universitas Telkom
Corresponding author e-mail: ekofajarcahyadi@telkomuniversity.ac.id

Abstrak

Pemeliharaan *web server* bertujuan untuk mencegah *downtime* dan menjaga ketersediaan layanan. Penelitian ini menerapkan *failover clustering* menggunakan HAProxy untuk mengalihkan layanan dari *server* utama ke *server* cadangan ketika terjadi kegagalan. Pengujian dilakukan pada lima skenario *availability* serta parameter *Quality of Service* (QoS) dengan variasi 500, 1.000, 1.500, 2.000, dan 2.500 *request* pada laju 100 *request* per detik. Hasil pengujian menunjukkan nilai *availability* sebesar 99,91%, 98,53%, 99,86%, 99,25%, dan 100% pada masing-masing skenario. Pada kondisi *server* utama aktif, *throughput* terbaik diperoleh sebesar 7.508 Kbps pada 2.000 *request*. *Packet loss* hanya terjadi pada 500 *request* sebesar 15,72% dengan kategori "Sedang", sedangkan variasi request lainnya memperoleh *packet loss* 0% atau kategori "Sangat Baik". Nilai *delay* berada pada kategori "Sangat Baik" hingga "Baik", sementara *jitter* berada pada kategori "Baik" berdasarkan standar TIPHON. CPU *usage* meningkat seiring bertambahnya jumlah *request*, dengan rata-rata sebesar 74,22%. Hasil ini menunjukkan bahwa HAProxy mampu mendukung *failover clustering* untuk meningkatkan *availability* dan menjaga performansi *web server*.
Kata kunci: *failover, high availability, HAProxy, web server*

Abstract

Web server maintenance aims to prevent downtime and maintain service availability. This study implemented failover clustering using HAProxy to redirect services from the primary server to the backup server when a failure occurs. Testing was conducted on five availability scenarios and Quality of Service (QoS) parameters with variations of 500, 1,000, 1,500, 2,000, and 2,500 requests at a rate of 100 requests per second. The test results showed availability values of 99.91%, 98.53%, 99.86%, 99.25%, and 100% in each scenario. Under active primary server conditions, the best throughput was obtained at 7508 Kbps at 2,000 requests. Packet loss only occurred in 500 requests at 15.72% with the "moderate" category, while other request variations obtained 0% packet loss or the "very good" category. Delay values ranged from "very good" to "good," while jitter was in the "good" category based on TIPHON standards. CPU usage increased with the number of requests, averaging 74.22%. These results demonstrate that HAProxy is capable of supporting failover clustering to improve availability and maintain web server performance.
Keyword: *failover, high availability, HAProxy, web server*

I. PENDAHULUAN

Perkembangan internet sebagai media penyebaran informasi dan komunikasi telah meningkatkan trafik jaringan secara signifikan [1], termasuk meningkatnya jumlah *request* yang diterima oleh *web server*. Peningkatan *request* secara terus-menerus dapat menambah beban kerja *server*, menurunkan kualitas layanan, bahkan menyebabkan *overload* atau *downtime*. Padahal, *web server* sebagai penyedia layanan dituntut untuk tetap tersedia dan mampu melayani *client* secara berkelanjutan [2]. Selain itu, pemanfaatan teknologi internet yang semakin luas dalam penyampaian informasi menuntut infrastruktur layanan yang andal dan mudah diakses [3].

Salah satu pendekatan untuk menjaga ketersediaan layanan adalah penerapan *high availability* melalui mekanisme *failover clustering*. Mekanisme ini menyediakan *server* cadangan yang berada dalam satu *cluster* dengan *server* utama, sehingga ketika *server* utama mengalami gangguan, layanan dapat dialihkan ke *server* cadangan [3]. Dalam penelitian ini, *failover clustering* diterapkan menggunakan HAProxy dengan model *active-passive failover*. HAProxy berperan sebagai penghubung antara *client* dan *web server*, serta mengalihkan trafik ke *server* cadangan apabila *server* utama mengalami kegagalan.

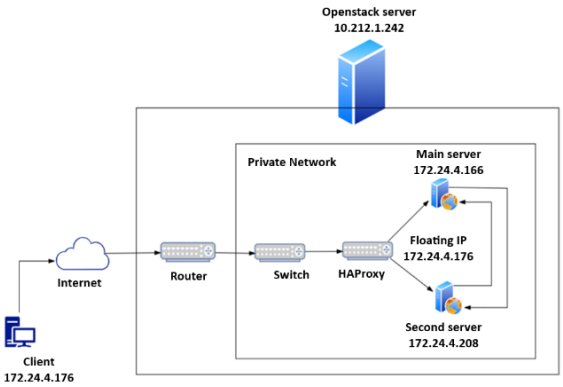
Penelitian sebelumnya oleh Pribadi dkk. [4] menunjukkan bahwa *failover clustering* mampu mendukung pencapaian *high availability* pada *web server*. Namun, penelitian tersebut masih menggunakan *server* fisik, sehingga proses konfigurasi dan peningkatan sumber daya relatif kurang fleksibel. Penggunaan lingkungan *cloud* menawarkan keunggulan karena sumber daya komputasi, penyimpanan, dan jaringan dapat dikelola secara lebih mudah dan efisien.

Berdasarkan permasalahan tersebut, penelitian ini mengevaluasi kinerja *failover clustering* berbasis HAProxy untuk meningkatkan *high availability web server* pada lingkungan *cloud*. Evaluasi dilakukan berdasarkan parameter *availability*, *Quality of Service* (QoS) yang meliputi *throughput*, *packet loss*, *delay*, dan *jitter*, serta CPU *usage*. Hasil penelitian diharapkan dapat menunjukkan efektivitas HAProxy dalam menjaga ketersediaan dan performansi layanan *web server* ketika terjadi kegagalan pada *server* utama.

II. TINJAUAN PUSTAKA

Pada tahun 2021, Pratama [3] membandingkan teknologi *failover* berbasis klaster menggunakan Heartbeat dan berbasis jaringan menggunakan Keepalived pada layanan *web server*. Hasilnya menunjukkan bahwa kedua pendekatan mampu menghasilkan waktu *downtime* dan *failback* yang rendah, sehingga mendukung peningkatan ketersediaan layanan. Kemudian, Pribadi dkk. [4] menganalisis *failover clustering* untuk mencapai *high availability* pada *web server* dengan parameter *availability*, *workload*, dan QoS, serta memperoleh nilai *availability* sebesar 99,90%. Hasil pengujian QoS menunjukkan bahwa kedua sistem memperoleh kategori memuaskan.

Sementara itu, Putra dkk. [5] mengimplementasikan *high availability cluster web server* menggunakan virtualisasi Docker dan HAProxy sebagai *load balancer* dengan algoritma least connection dan round robin. Hasil pengujian menunjukkan bahwa round robin memberikan performansi lebih tinggi dibandingkan least connection. Secara umum, penelitian-penelitian tersebut menunjukkan bahwa penerapan *failover*, *clustering*, dan HAProxy dapat meningkatkan ketersediaan serta performansi layanan *web server*.



Gambar 1. Topologi jaringan

III. METODOLOGI PENELITIAN

Pada bagian ini dibahas mengenai spesifikasi perangkat keras dan lunak yang digunakan, topologi jaringan, pengaturan simulasi, dan skenario pengujian.

Spesifikasi Perangkat Keras dan Lunak yang Digunakan

Perangkat keras yang diperlukan pada penelitian ini berupa laptop dan PC. Laptop digunakan sebagai alat perantara untuk *remote* PC dengan spesifikasi yang terdapat pada Tabel 1. PC digunakan untuk menjalankan virtualisasi dengan detil tercantum pada Tabel 2.

Beberapa mesin virtual yang dijalankan pada penelitian ini yaitu satu *client*, satu OpenStack *server*, satu HAProxy *server*, dan dua *web server*. Spesifikasi perangkat virtual yang digunakan terdapat pada Tabel 3.

Software tool dan aplikasi yang digunakan untuk melakukan simulasi *failover clustering* pada *web server* ditunjukkan pada Tabel 4.

Tabel 1. Spesifikasi laptop

Perangkat	Spesifikasi
Sistem Operasi	Ms. Windows 10
Processor	Intel (R) Core (TM) i3-6006U CPU@ 2.00GHz
RAM	4 GB

Tabel 2. Spesifikasi PC

Perangkat	Spesifikasi
Sistem Operasi	Ms. Windows 10
Processor	Intel (R) Core (TM) i7-9700KF CPU @ 3.60GHz
RAM	64 GB

Tabel 3. Perangkat virtual

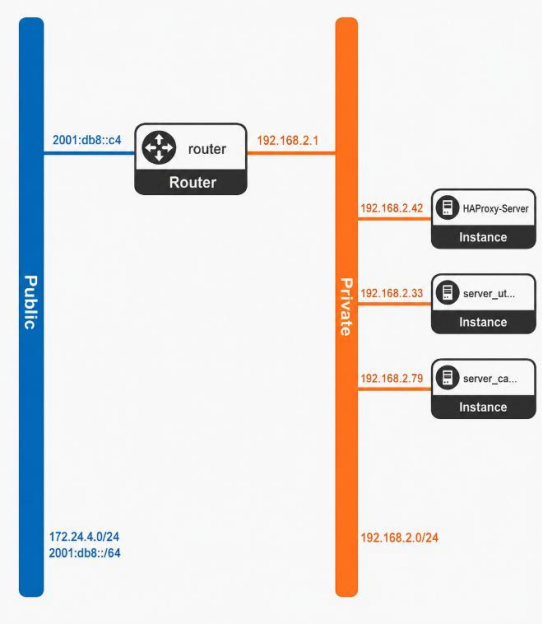
Peruntukan	Perangkat	Spesifikasi
CLIENT	Sistem Operasi	Ubuntu 18.04
	RAM	2 GB
	Harddisk	20 GB
	Alamat IP	172.24.4.195
OPENSTACK *SERVER*	Sistem Operasi	Ubuntu 20.04
	RAM	20 GB
	Harddisk	200 GB
	Alamat IP	192.168.1.242
***WEB SERVER* UTAMA**	Sistem Operasi	Ubuntu 18.04
	RAM	2 GB
	Harddisk	20 GB
	Alamat IP	172.24.4.166
***WEB SERVER* CADANGAN**	Sistem Operasi	Ubuntu 18.04
	RAM	2 GB
	Harddisk	20 GB
	Alamat IP	172.24.4.208
HAPROXY *SERVER*	Sistem Operasi	Ubuntu 18.04
	RAM	2 GB
	Harddisk	20 GB
	Alamat IP	172.24.4.176

Tabel 4. *Tools* dan aplikasi

No.	Software yang digunakan	Peruntukan
1.	VirtualBox	Virtualisasi *server*
2.	Apache	*Web server*
3.	Openstack	*Cloud computing*
4.	Apache benchmark	Mengukur QoS *web server*
5.	HTTPerf	Mengukur parameter CPU

Topologi Jaringan

Topologi jaringan pada Gambar 1 terdiri atas satu *client*, satu *server* OpenStack, *router*, *switch*, HAProxy, serta dua *web server*. *Client* digunakan untuk melakukan pengujian, sedangkan *server* OpenStack berperan sebagai layanan IaaS berbasis *cloud computing* untuk mengelola sumber daya komputasi, penyimpanan, dan jaringan virtual.



Gambar 2. Topologi jaringan OpenStack

Server OpenStack terhubung ke jaringan publik dan privat. Jaringan publik digunakan agar *client* dapat mengakses lingkungan OpenStack melalui internet, sedangkan jaringan privat digunakan untuk menghubungkan *router*, *switch*, HAProxy, dan *web server*. *Router* virtual pada OpenStack berfungsi menghubungkan jaringan privat dengan internet menggunakan alamat jaringan 172.24.4.0/24.

Switch digunakan untuk menghubungkan HAProxy dengan *web server* utama dan *web server* cadangan dalam satu jaringan lokal. Dengan konfigurasi ini, kedua *web server* dapat berada dalam satu *cluster* dan saling berkomunikasi. HAProxy berperan dalam mengatur serta mengalihkan trafik dari *web server* utama ke *web server* cadangan apabila diperlukan. *Floating* IP digunakan agar *instance* yang memiliki IP internal tetap dapat diakses dari jaringan publik.

Skenario Pengujian

Pengujian pada penelitian ini meliputi *availability*, QoS yang terdiri atas *throughput*, *packet loss*, *delay*, dan *jitter*, serta CPU *usage*. Perancangan sistem ditunjukkan pada Gambar 2. Gambar 2 menunjukkan topologi *logic instance* yang digunakan oleh layanan Neutron OpenStack.

Tahap awal penelitian dilakukan dengan mengonfigurasi OpenStack *server*. Di dalam OpenStack dibuat beberapa *instance*, yaitu *web server* utama, *web server* cadangan, dan HAProxy sebagai komponen *failover clustering*. *Client* mengakses layanan *web server* melalui *floating* IP, sehingga ketika *server* utama mengalami kegagalan, layanan dapat langsung dialihkan ke *server* cadangan. Dengan mekanisme ini, *client* tetap dapat mengakses *web server* tanpa mengetahui adanya gangguan pada *server* utama. Seluruh konfigurasi *instance* dilakukan melalui *remote* dari PC Host.

Konfigurasi dan Uji Coba *Failover Clustering* pada *Web Server* Menggunakan HAProxy

Implementasi *failover clustering* dilakukan menggunakan HAProxy untuk menjaga ketersediaan layanan *web server*. HAProxy berfungsi mengalihkan trafik dari *server* utama ke *server* cadangan ketika terjadi kegagalan, sehingga *client* tetap dapat mengakses layanan.

Sebelum pengujian, dilakukan konfigurasi OpenStack, IP *address* pada *web server*, instalasi Apache, serta konfigurasi HAProxy. *Client* mengakses *web server* melalui *floating* IP HAProxy, bukan langsung ke *server*. Jika *server* utama mati dan layanan berhasil dialihkan ke *server* cadangan, maka konfigurasi *failover clustering* dinyatakan berhasil.

Pengujian *Availability* dan QoS Web Server Berdasarkan Jumlah *Request*

Pengujian dilakukan pada sistem OpenStack *server* yang menjadi tempat implementasi *failover clustering* menggunakan HAProxy. *Client* mengakses *web server* melalui *floating* IP. *Failover* diuji untuk memastikan layanan *web server* tetap berjalan meskipun *server* utama mengalami kegagalan. *Availability* diuji berdasarkan lima skenario:

1. Kedua *web server* hidup, lalu *server* utama dimatikan;
2. Kedua *web server* mati, lalu *server* cadangan dihidupkan;
3. Kedua *web server* hidup, lalu *server* cadangan dimatikan;
4. Kedua *web server* mati, lalu *server* utama dihidupkan;
5. *Server* utama direstart.
6. *Data availability* diperoleh dari waktu *uptime* dan *downtime*.

Pengujian QoS dan CPU *usage* dilakukan dengan variasi jumlah permintaan 500, 1.000, 1.500, 2.000, dan 2.500 *request*, denga merujuk kepada standar TIPHON yang dikeluarkan oleh *European Telecommunications Standards Institute* (ETSI) [6].

IV. HASIL DAN PEMBAHASAN

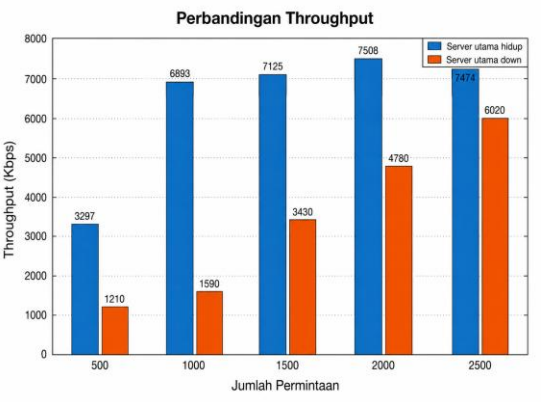
Bagian ini membahas hasil luaran *throughput*, *packet loss*, *delay*, dan *jitter*, serta CPU *usage*.

Throughput

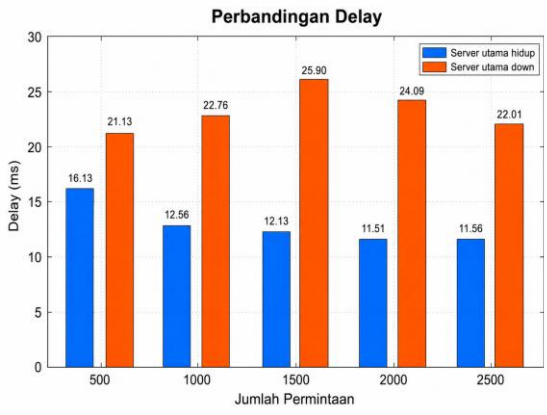
Saat *server* utama dalam kondisi hidup, *throughput* meningkat dari 3.297 Kbps pada 500 *request* menjadi 6.893 Kbps pada 1.000 *request*, 7.125 Kbps pada 1.500 *request*, dan mencapai nilai tertinggi sebesar 7.508 Kbps pada 2.000 *request*, sedikit menurun menjadi 7.474 Kbps pada 2.500 *request* (lihat Gambar 3). Sementara itu, saat *server* utama mengalami *down*, nilai *throughput* turun. Hasil ini menunjukkan bahwa kondisi *server* utama hidup menghasilkan *throughput* yang lebih baik dibandingkan saat terjadi *failover* ke *server* cadangan. Namun, ketika *server* utama *down*, sistem tetap mampu melayani *request* dengan peningkatan *throughput* seiring bertambahnya jumlah *request*.

Packet Loss

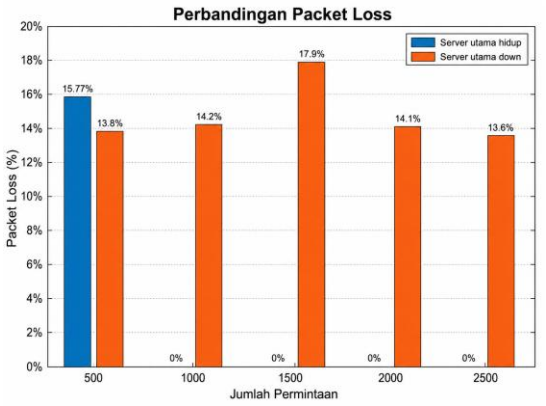
Saat *server* utama dalam kondisi hidup, *packet loss* hanya terjadi pada 500 *request* sebesar 15,72% dan termasuk kategori "Sedang", sedangkan pada 1.000 hingga 2.500 *request*, *packet loss* bernilai 0%, sehingga masuk kategori "Sangat Baik" (lihat Gambar 4). Sementara itu, ketika *server* utama mengalami *down* dan layanan dialihkan ke *server* cadangan, *packet loss* berada pada kisaran 13,6%–17,9%, dengan nilai tertinggi pada 1500 *request* sebesar 17,9%. Hasil ini menunjukkan bahwa kondisi *failover* masih mampu menjaga layanan tetap berjalan, tetapi kualitas pengiriman paket menurun dibandingkan kondisi *server* utama hidup. *Packet loss* dipengaruhi oleh kepadatan trafik, antrean paket, dan proses pengalihan layanan dari *server* utama ke *server* cadangan saat *failover* terjadi.



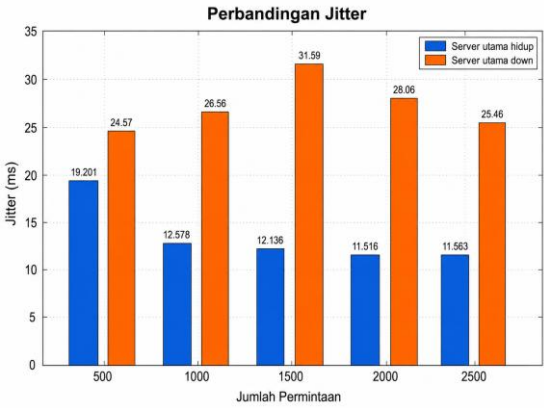
Gambar 3. Grafik *throughput server* utama hidup dan *down*



Gambar 5. Grafik *delay server* utama hidup dan *down*



Gambar 4. Grafik *packet loss server* utama hidup dan *down*



Gambar 6. Grafik *jitter server* utama hidup dan *down*

Delay

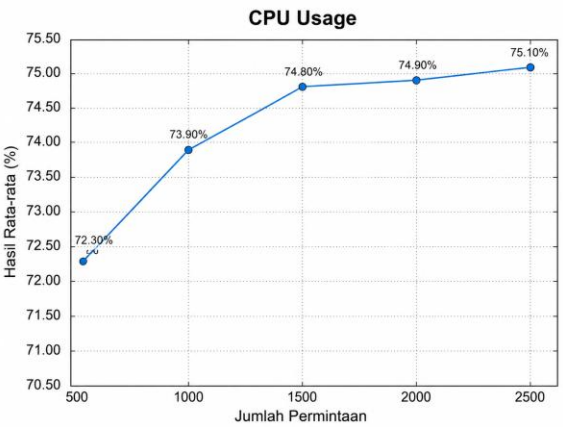
Saat *server* utama hidup, rata-rata *delay* berturut-turut sebesar 16,13 ms, 12,56 ms, 12,13 ms, 11,51 ms, dan 11,56 ms, sehingga seluruhnya masuk indeks 4 atau kategori "Sangat Baik" (lihat Gambar 5). Meskipun total *delay* meningkat seiring bertambahnya jumlah *request*, rata-rata *delay* justru cenderung menurun karena total *delay* dibagi dengan jumlah *request* yang lebih besar; nilai tertinggi pada 500 *request* juga dipengaruhi oleh adanya *packet loss* sebesar 15,72%. Ketika *server* utama mengalami *down*, rata-rata *delay* meningkat menjadi 21,13 ms, 22,76 ms, 25,90 ms, 24,09 ms, dan 22,01 ms karena setiap variasi *request* mengalami *packet loss*.

Jitter

Pengujian *jitter* dilakukan menggunakan Apache Benchmark untuk mengukur variasi waktu kedatangan paket data. Pada Gambar 6, saat *server* utama hidup, nilai rata-rata *jitter* berturut-turut sebesar 19,201 ms, 12,57 ms, 12,13 ms, 11,51 ms, dan 11,56 ms. Sementara itu, saat *server* utama mengalami *down*, nilai *jitter* sebesar 24,57 ms, 26,57 ms, 31,59 ms, 28,06 ms, dan 25,46 ms. Seluruh hasil pengujian berada pada indeks 3 berdasarkan standar TIPHON, sehingga termasuk kategori "Baik" karena nilainya masih di bawah 75 ms. Nilai *jitter* dipengaruhi oleh antrean paket, kepadatan trafik, dan waktu pemrosesan data.

CPU *Usage*

Hasil pengujian pada Gambar 7 menunjukkan bahwa CPU usage meningkat seiring bertambahnya jumlah *request*, yaitu 72,30% pada 500 *request*, 73,90% pada 1000 *request*, 74,80% pada 1.500 *request*, 74,90% pada 2.000 *request*, dan 75,10% pada 2.500 *request*.



Gambar 7. Grafik CPU *usage*

Peningkatan ini terjadi karena CPU harus memproses semakin banyak *request* dari *client*, sehingga beban kerja *web server* juga bertambah. Meskipun demikian, nilai CPU *usage* masih berada pada kisaran yang relatif stabil, dengan rata-rata keseluruhan sebesar 74,22%.

V. KESIMPULAN

Berdasarkan hasil pengujian, penerapan *failover clustering* menggunakan HAProxy pada lingkungan OpenStack terbukti mampu menjaga ketersediaan layanan *web server* ketika terjadi kegagalan pada *server* utama, dengan nilai *availability* pada lima skenario sebesar 99,91%, 98,53%, 99,86%, 99,25%, dan 100%. Hasil QoS menunjukkan bahwa performansi terbaik diperoleh saat *server* utama aktif, terutama pada *throughput* tertinggi sebesar 7.508 Kbps pada 2.000 *request*, sedangkan saat *failover* ke *server* cadangan terjadi penurunan *throughput* dan peningkatan *packet loss*. Meskipun demikian, *delay* dan *jitter* masih berada dalam kategori "Baik" berdasarkan standar TIPHON, sehingga sistem tetap mampu menjaga kualitas layanan. Penggunaan CPU juga meningkat seiring bertambahnya jumlah *request*, tetapi masih relatif stabil dengan rata-rata 74,22%. Secara keseluruhan, HAProxy efektif digunakan sebagai solusi *failover clustering* untuk meningkatkan *high availability web server*.

DAFTAR PUSTAKA

[1] R. Gunawan, S. Aulia, H. Supeno, A. Wijanarko, J. P. Uwiringiyimana, dan D. Mahayana, "Adiksi media sosial dan gadget bagi pengguna Internet di Indonesia," *Techno-Socio Ekon.*, 14(1), hlm. 1-14, 2021.

[2] Y. B. Ginting dan U. Ristian, "Implementasi metode failover sebagai backup server pada arsitektur load balancer," *Coding J. Komput. dan Apl.*, 9(2), hlm. 198-210, 2021.

[3] D. Pratama, "Perbandingan kinerja teknologi failover berbasis klaster (heartbeat) dengan teknologi failover berbasis jaringan (keepalived)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(12), hlm. 5572-5581, 2021.

[4] Y. Pribadi, A. B. P. Negara, dan M. A. Irwansyah, "Analisis penggunaan metode failover clustering untuk mencapai high availability pada web server (studi kasus: gedung jurusan informatika)," *J. Sist. dan Teknol. Inf.*, 8(2), hlm. 218, 2020.

[5] M. A. A. Putra, I. Fitri, dan A. Iskandar, "Implementasi high availability cluster web server menggunakan virtualisasi container docker," *J. Media Inform. Budidarma*, 4(1), hlm. 9-13, 2020.

[6] ETSI, "Telecommunications and internet protocol harmonization over networks (TIPHON); general aspects of quality of service (QoS)."

SIMULASI SISTEM ANTRIAN LOKET TIKET WISATA RUARASA MENGGUNAKAN METODE DISCRETE EVENT SIMULATION (DES)

Ayu Liza Putri Wiwaha[1)], Baiq Adelia Dwi Savitri[1)], Nayla Anugerah Nisa[1)], Alya Dwi Pangesti[1)], Herliana Rosika.[1)]
1) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: f1d02310020@student.unram.ac.id

Abstrak

Wisata RuaRasa merupakan destinasi wisata di Kota Mataram yang beroperasi sejak Januari 2026. Seiring meningkatnya jumlah pengunjung, sistem pelayanan loket tiket perlu dievaluasi agar tidak menimbulkan penumpukan antrian yang mengganggu kenyamanan pengunjung. Penelitian ini bertujuan untuk memodelkan dan mensimulasikan sistem antrian loket tiket RuaRasa menggunakan metode Discrete Event Simulation (DES) berbasis SimPy. Dataset yang digunakan adalah rekap transaksi nyata periode Januari–Mei 2026 sebanyak 1.485 transaksi dari 117 hari operasional. Hasil uji Kolmogorov-Smirnov menunjukkan bahwa Inter-Arrival Time (IAT) pengunjung mengikuti distribusi Gamma (k=0,6271, θ=36,49, p=0,4643), menolak asumsi distribusi Eksponensial (p=0,0000). Simulasi dilakukan dalam 4 skenario kombinasi jumlah kasir (1 dan 2) dengan kondisi kedatangan (normal dan jam puncak), masing-masing diulang sebanyak 30 replikasi. Seluruh skenario memiliki intensitas trafik (ρ) di bawah 1, menandakan sistem stabil. Skenario 1 kasir kondisi normal menghasilkan waktu tunggu rata-rata 0,313 menit, sedangkan skenario jam puncak dengan 2 kasir berhasil menurunkan waktu tunggu menjadi 0,001 menit. Penelitian ini merekomendasikan penerapan 1 kasir aktif untuk hari biasa dengan 1 kasir siaga pada jam puncak pukul 10.00 WITA.
Kata kunci: Discrete Event Simulation; Distribusi Gamma; Optimasi Kasir; RuaRasa; Sistem Antrian

Abstract

RuaRasa is a tourist destination in Mataram City that has been operating since January 2026. As visitor numbers grow, the ticket counter service system needs evaluation to prevent queue build-up that disrupts visitor comfort. This study aims to model and simulate the RuaRasa ticket counter queuing system using the Discrete Event Simulation (DES) method based on SimPy. The dataset consists of 1,485 actual transaction records from 117 operational days during January–May 2026. The Kolmogorov-Smirnov test showed that visitor Inter-Arrival Time follows a Gamma distribution (k=0.6271, θ=36.49, p=0.4643), rejecting the Exponential assumption (p=0.0000). Four scenarios combining cashier count and arrival conditions were each replicated 30 times. All scenarios showed traffic intensity (ρ) below 1, indicating a stable system. The 1-cashier normal scenario yielded 0.313 minutes average waiting time, while the 2-cashier peak scenario reduced it to 0.001 minutes. This study recommends 1 active cashier on regular days with 1 standby during peak hours at 10:00 WITA.
Keyword: *Cashier Optimization; Discrete Event Simulation; Gamma Distribution; Queuing System; RuaRasa*

I. PENDAHULUAN

Perkembangan sektor pariwisata di Indonesia mendorong berbagai destinasi wisata untuk terus meningkatkan kualitas layanan kepada pengunjung. Salah satu aspek layanan yang sering menjadi kendala adalah sistem antrian, khususnya pada loket pembelian tiket masuk. Antrian yang panjang dan waktu tunggu yang lama dapat menurunkan kepuasan pengunjung, mengurangi pengalaman wisata, dan berdampak buruk pada reputasi destinasi wisata tersebut. Pengelolaan sistem antrian yang efisien menjadi krusial, terutama bagi destinasi wisata yang baru beroperasi dan tengah membangun citra layanannya di mata pengunjung.

RuaRasa merupakan destinasi wisata berbasis edukasi dan hiburan yang berlokasi di Kota Mataram, Nusa Tenggara Barat. Sejak beroperasi pada Januari 2026, RuaRasa beroperasi setiap hari mulai pukul 10.00 hingga 21.00 WITA dengan sistem loket tiket yang dilayani oleh 2 kasir yang dibagi dalam 2 shift per hari, meskipun dalam praktiknya sering kali hanya 1 kasir yang aktif pada satu waktu. RuaRasa melayani pengunjung dengan empat jenis tiket, yaitu REGULER (69,6%), PELAJAR (19,1%), EVENT (8,6%), dan OUTING (2,6%). Berdasarkan data transaksi periode Januari–Mei 2026, rata-rata pengunjung per hari mencapai 46 orang dengan rata-rata 12,96 transaksi per hari. Distribusi kedatangan pengunjung sepanjang jam operasional menunjukkan pola yang bervariasi: jam puncak terjadi pada pukul 10.00 WITA dengan 191 transaksi (12,3% dari total), diikuti pukul 16.00 WITA dengan 173 transaksi (11,2%),

dan pukul 11.00 WITA dengan 140 transaksi (9,0%). Aktivitas pengunjung mulai mereda memasuki pukul 19.00–21.00 WITA dengan masing-masing 154 dan 43 transaksi. Seluruh segmen jam didominasi oleh pengunjung bertipe REGULER dengan proporsi 59–79% per jam, menandakan mayoritas pengunjung datang secara mandiri dengan pola kedatangan yang cenderung tidak terprediksi.

Permasalahan sistem antrian merupakan salah satu topik yang banyak dikaji menggunakan pendekatan simulasi, termasuk pada fasilitas wisata [1] dan layanan publik lainnya [2][3]. Metode Discrete Event Simulation (DES) merupakan pendekatan yang tepat untuk memodelkan sistem antrian karena mampu merepresentasikan kejadian diskrit yang terjadi secara sekuensial dalam suatu sistem [4]. DES memungkinkan pengujian berbagai skenario tanpa harus mengintervensi sistem nyata yang sedang berjalan, sehingga sangat efisien dari segi biaya dan waktu [5]. Penelitian ini menggunakan library SimPy berbasis Python untuk membangun model DES sistem antrian loket tiket RuaRasa berbasis data transaksi nyata, dan diharapkan dapat memberikan rekomendasi optimal terkait jumlah kasir aktif yang diperlukan, mulai dari kondisi operasional normal hingga jam puncak pukul 10.00 WITA, sebagai bahan pertimbangan manajemen RuaRasa dalam mengalokasikan sumber daya kasir secara efisien.

II. TINJAUAN PUSTAKA

a. Teori Antrian

Teori antrian (*queueing theory*) adalah cabang ilmu matematika yang mempelajari fenomena menunggu dalam suatu sistem layanan. Sistem antrian terdiri atas tiga komponen utama: entitas yang datang (*arrival*), fasilitas pelayanan (*server*), dan mekanisme antrian [6]. Karakteristik sistem antrian dinyatakan dalam notasi Kendall A/B/c. Intensitas trafik (ρ) merupakan parameter kunci yang menyatakan rasio antara laju kedatangan (λ) dan laju pelayanan (μ) dikalikan jumlah server. Sistem dikatakan stabil apabila $\rho < 1$ [7].

b. Discrete Event Simulation (DES)

Discrete Event Simulation (DES) adalah metode simulasi di mana perubahan status sistem hanya terjadi pada titik-titik waktu diskrit yang disebut *event*. DES sangat cocok untuk memodelkan sistem antrian karena setiap kedatangan dan selesainya pelayanan merupakan event diskrit yang dapat direkam dan dianalisis [6]. Penerapan DES pada sistem antrian loket kasir juga telah dilakukan oleh Mazri et al. [8] yang membuktikan efektivitas simulasi dalam mengidentifikasi alokasi sumber daya yang optimal. SimPy adalah *framework* DES berbasis Python yang menggunakan pendekatan berbasis proses sehingga interaksi antara entitas dan server dapat dimodelkan secara alami dan intuitif [9].

c. Distribusi Gamma

Distribusi Gamma merupakan distribusi probabilitas kontinu yang sering digunakan untuk memodelkan waktu antar kejadian dalam sistem yang tidak memenuhi asumsi Markovian (*memoriless*). Distribusi Gamma memiliki dua parameter: shape (k) dan scale (θ). Nilai $k<1$ mengindikasikan pola kedatangan yang lebih bersifat *bursty* (berkelompok), sementara $k>1$ mencerminkan kedatangan yang lebih teratur [10].

III. METODOLOGI PENELITIAN

a. Dataset

Dataset yang digunakan adalah Rekap Transaksi RuaRasa periode Januari–Mei 2026 yang diperoleh langsung dari sistem kasir RuaRasa, memuat 1.633 baris data. Proses preprocessing meliputi: (1) filter transaksi bertipe IN dengan variant tiket REGULER, PELAJAR, OUTING, atau EVENT; (2) konversi timestamp dari UTC ke WITA (UTC+8); (3) filter jam operasional 10.00–21.00 WITA; (4) konversi kolom Qty dan Real Total ke tipe numerik. Setelah preprocessing, diperoleh 1.485 transaksi tiket dari 117 hari operasional.

Tabel 1. Distribusi Jenis Pengunjung RuaRasa

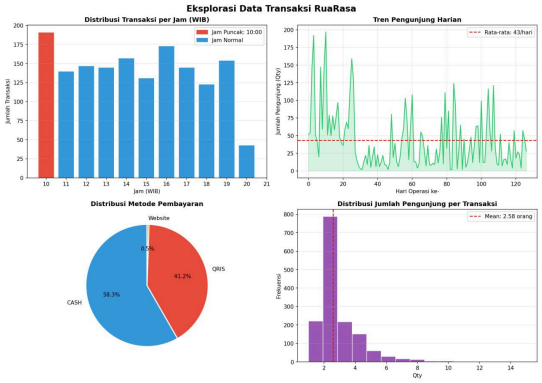
Jenis Tiket	Jumlah Transaksi
REGULER	1.040
PELAJAR	270
EVENT	139
OUTING	36
Total	1.485

Tabel 1 menunjukkan mayoritas pengunjung adalah tipe REGULER (70,0%), mengindikasikan sebagian besar pengunjung datang secara mandiri

sehingga pola kedatangan cenderung acak dan tidak terprediksi.

b. Eksplorasi Data (EDA)

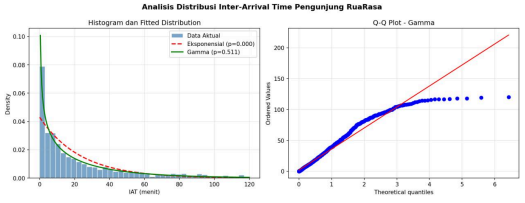
Eksplorasi data dilakukan untuk memahami pola transaksi, tren pengunjung harian, distribusi metode pembayaran, dan distribusi jumlah pengunjung per transaksi sebagaimana divisualisasikan pada Gambar 1. Jam puncak terjadi pada pukul 10.00 WITA dengan 178 transaksi selama periode pengamatan. Pembayaran tunai (CASH) mendominasi sebesar 58,3%, diikuti QRIS 41,2%, mengimplikasikan proses transaksi mungkin memerlukan waktu lebih lama dibanding pembayaran digital. Rata-rata Qty per transaksi adalah 2,60 orang dengan distribusi right-skewed.



Gambar 1. Eksplorasi Data Transaksi RuaRasa

c. Analisis Distribusi *Inter-Arrival Time*

Inter-Arrival Time (IAT) dihitung sebagai selisih waktu antar transaksi berurutan dalam satu hari operasional. IAT di luar rentang 0–120 menit dikecualikan, menghasilkan 1.254 sampel. Uji Kolmogorov-Smirnov (KS Test) digunakan untuk menentukan distribusi yang paling sesuai antara distribusi Eksponensial (asumsi standar M/M/c) dan distribusi Gamma, dengan parameter Gamma diestimasi menggunakan Maximum Likelihood Estimation (MLE) dengan constraint floc=0.



Gambar 2. Analisis Distribusi *Inter-Arrival Time* Pengunjung RuaRasa

Tabel 2. Hasil Uji Kolmogorov-Smirnov

Distribusi	KS Statistik	p-value	Kesimpul-an
Eksponensial	0,1162	0,0000	Ditolak
Gamma (k=0,6271, θ=36,49)	0,0239	0,4643	Diterima

Distribusi Eksponensial ditolak (p=0,0000), sedangkan distribusi Gamma diterima (p=0,4643). Nilai k=0,6271 < 1 mengonfirmasi pola kedatangan bersifat *over-dispersed* atau *bursty*, yaitu pengunjung cenderung datang dalam kelompok yang dipisahkan oleh jeda waktu panjang.

d. Parameter Model DES

Tabel 3. Parameter Model DES

Parameter	Nilai	Satuan
Arrival rate normal (λ)	0,0192	trx/menit
Arrival rate puncak (λ)	0,0256	trx/menit
Jam puncak	10:00 WITA	-
Service mean (1/μ)	3,0	menit/trx
Service rate (μ)	0,3333	trx/menit
Durasi simulasi	660	menit
Distribusi IAT terpilih	Gamma	-
Shape Gamma (k)	0,6271	-
Scale Gamma (θ)	36,4860	menit
Jumlah hari data	117	hari

Arrival rate normal diperoleh dari rata-rata transaksi per jam dibagi 60, sedangkan arrival rate puncak diperoleh dari jam dengan transaksi terbanyak yaitu pukul 10.00 WITA. Service mean ditetapkan 3 menit per transaksi berdasarkan estimasi proses *input* tiket, verifikasi, dan pencetakan tiket. *Service time* diasumsikan mengikuti distribusi Eksponensial, sehingga model terklasifikasi sebagai G/M/c. Penggunaan 30 replikasi didasarkan pada Central Limit Theorem yang menjamin distribusi

rata-rata sampel mendekati normal untuk n ≥ 30.

e. Skenario Simulasi

Tabel 4. Skenario Simulasi

Skenario	Jumlah Kasir	Kondisi Kedatangan	λ (trx/menit)
S1	1 Kasir	Normal	0,0192
S2	2 Kasir	Normal	0,0192
S3	1 Kasir	Jam Puncak	0,0256
S4	2 Kasir	Jam Puncak	0,0256

S1 merepresentasikan kondisi baseline (1 kasir, kedatangan rata-rata). S2 menguji dampak penambahan kasir pada kondisi normal. S3 menguji ketahanan sistem 1 kasir saat jam puncak. S4 merupakan skenario terbaik dengan 2 kasir pada jam puncak.

f. Validasi Model

Validasi model dilakukan dengan membandingkan rata-rata jumlah transaksi per hari hasil simulasi terhadap data nyata. Data historis menunjukkan rata-rata 12,69 transaksi per hari (1.485 transaksi dibagi 117 hari operasional), sedangkan simulasi skenario S1 menghasilkan rata-rata 12 transaksi per hari. Selisih sebesar 5,43% berada di bawah ambang batas toleransi 10%, sehingga model dinyatakan valid dan cukup representatif terhadap kondisi operasional nyata RuaRasa.

IV. HASIL DAN PEMBAHASAN

a. Intensitas Trafik (ρ)

Tabel 5. Intensitas Trafik per Skenario

Skenario	λ (trx/mnt)	c × μ	ρ	Status
S1 (1 kasir, normal)	0,0192	0,3333	0,058	Stabil
S2 (2 kasir, normal)	0,0192	0,6667	0,029	Stabil
S3 (1 kasir, puncak)	0,0256	0,3333	0,077	Stabil
S4 (2 kasir, puncak)	0,0256	0,6667	0,038	Stabil

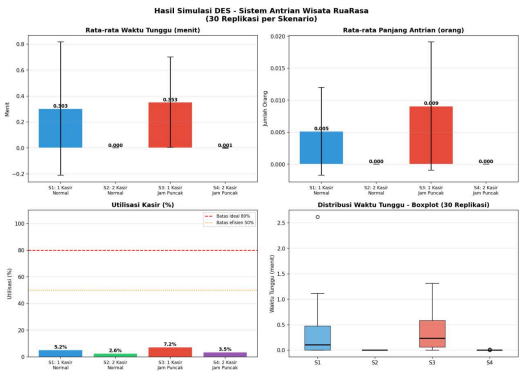
Seluruh skenario menghasilkan ρ jauh di bawah 1, dengan tertinggi pada S3 (ρ=0,077) dan terendah pada S2 (ρ=0,029). Nilai ρ yang sangat rendah menunjukkan kapasitas pelayanan kasir RuaRasa jauh melebihi laju kedatangan pengunjung. Karena IAT aktual mengikuti distribusi Gamma, hasil simulasi DES diperlukan untuk gambaran yang lebih akurat.

b. Hasil Simulasi DES

Tabel 6. Rekap Hasil Simulasi DES (30 Replikasi)

Skenario	Waktu Tunggu (mnt)	Panjang Antrian (org)	Utilisasi Kasir (%)	Trx
S1: 1 Kasir, Normal	0,313 ± 0,517	0,005	5,5%	12
S2: 2 Kasir, Normal	0,000 ± 0,000	0,000	2,7%	12
S3: 1 Kasir, Jam Puncak	0,350 ± 0,338	0,010	7,4%	17
S4: 2 Kasir, Jam Puncak	0,001 ± 0,003	0,000	3,7%	17

Penambahan 1 kasir hampir mengeliminasi waktu tunggu sepenuhnya: S2 (0,000 menit) vs S1 (0,313 menit), dan S4 (0,001 menit) vs S3 (0,350 menit). Standar deviasi yang besar pada S1 (±0,517) dan S3 (±0,338) mencerminkan variabilitas tinggi akibat sifat bursty distribusi Gamma (k<1).



Gambar 3. Hasil Simulasi DES – Sistem Antrian Wisata RuaRasa (30 Replikasi per Skenario)

Gambar 3 memvisualisasikan hasil simulasi dalam empat panel. Panel kiri atas menampilkan rata-rata waktu tunggu dengan error bar standar deviasi. Terlihat jelas kontras antara S1/S3 (batang biru dan

merah yang cukup tinggi) dengan S2/S4 (hampir menyentuh nol). Error bar yang panjang pada S1 dan S3 mengindikasikan bahwa pada beberapa replikasi terjadi antrian cukup signifikan, sementara pada replikasi lain tidak ada antrian sama sekali. Kondisi tersebut merupakan konsekuensi langsung dari pola kedatangan bursty akibat nilai shape distribusi Gamma yang kurang dari satu.

Panel kanan atas menampilkan rata-rata panjang antrian. S3 (0,010 orang) memiliki panjang antrian dua kali lipat S1 (0,005 orang), menunjukkan dampak nyata peningkatan λ pada jam puncak. Sementara itu, S2 dan S4 menghasilkan panjang antrian mendekati nol, membuktikan efektivitas 2 kasir dalam mengeliminasi penumpukan pengunjung.

Panel kiri bawah menampilkan utilisasi kasir. Seluruh skenario berada jauh di bawah batas ideal 80% maupun batas efisien 50%. Utilisasi tertinggi hanya 7,4% pada S3 dan terendah 2,7% pada S2, mengindikasikan kapasitas layanan saat ini sudah lebih dari cukup untuk melayani volume pengunjung rata-rata.

Panel kanan bawah menampilkan boxplot distribusi waktu tunggu dari 30 replikasi. Boxplot S1 dan S3 memiliki rentang interkuartil yang lebar dan whisker yang panjang, bahkan terdapat satu outlier di atas 2,5 menit pada S1. Sebaliknya, boxplot S2 dan S4 sangat tipis dan mendekati nol, menunjukkan konsistensi tinggi dari skenario dua kasir pada seluruh replikasi.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil memodelkan dan mensimulasikan sistem antrian loket tiket wisata RuaRasa menggunakan metode DES berbasis SimPy dengan memanfaatkan dataset transaksi nyata periode Januari–Mei 2026. Distribusi IAT pengunjung mengikuti distribusi Gamma ($k=0{,}6271$, $\theta=36{,}49$, $p=0{,}4643$), tidak mengikuti distribusi Eksponensial. Nilai $k<1$ mengindikasikan pola kedatangan bersifat bursty yang menyebabkan variabilitas cukup tinggi pada skenario satu kasir. Validasi model menunjukkan selisih 5,43% antara *output* simulasi dan data nyata, sehingga model dinyatakan valid. Seluruh skenario memiliki ρ kurang dari 0,08, sehingga sistem antrian dinyatakan stabil dan kapasitas layanan saat ini memadai untuk melayani rata-rata 45 pengunjung per hari. Penambahan kasir dari satu menjadi dua terbukti menurunkan waktu tunggu hingga 99,7–100%, namun tingkat utilisasi yang sangat rendah menunjukkan bahwa penambahan kasir permanen kurang efisien secara operasional dan ekonomi. Rekomendasi yang diberikan adalah menerapkan satu kasir aktif pada jam operasional normal dengan satu kasir siaga yang diaktifkan pada jam puncak pukul 10.00–11.00 WITA atau saat terdapat event khusus. Penelitian ini memiliki keterbatasan pada penetapan *service time* yang bersifat estimasi tanpa pengukuran dan pengujian distribusi statistik secara langsung, sehingga penelitian lanjutan disarankan untuk mengukur *service time* aktual per jenis tiket serta mempertimbangkan pengaruh metode pembayaran terhadap durasi pelayanan.

VI. UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Ibu Herliana Rosika, S.Kom., M.Kom. selaku dosen pengampu Mata Kuliah Pemodelan dan Simulasi atas bimbingan dan arahan yang diberikan selama proses pembelajaran. Terima kasih juga kepada pihak RuaRasa Mataram yang telah memberikan akses data transaksi sehingga penelitian ini dapat terlaksana, serta kepada seluruh rekan mahasiswa atas dukungan dan kerja samanya.

DAFTAR PUSTAKA

[1] I. W. Ananta, “Simulasi Sistem Pelayanan Antrian di Destinasi Wisata Jimbaran,” *J. Inform. dan Sist. Inf.*, vol. 5, no. 1, pp. 36–51, 2024.

[2] B. Soekarno-Hatta, “Analisis Simulasi Antrian Penumpang di Check-In Counter Bandara Menggunakan Discrete Event Simulation: Studi Kasus Citilink di Bandara Soekarno-Hatta,” *J. Tek. Ind.*, vol. 4, pp. 3449–3460, 2022.

[3] S. Shella, M. Wara, M. Nasrudin, A. F. Adziima, and A. R. Pratama, “Optimasi Sistem Antrian pada Medical Center ITS dengan Simulasi Discrete Event dan Response Surface Methodology,” *J. Tek. Ind.*, 2025.

[4] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 6th ed. Pearson, 2022.

[5] A. M. Law, *Simulation Modeling and Analysis*, 6th ed. McGraw-Hill Education, 2019.

[6] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 11th ed. McGraw-Hill Education, 2021.

[7] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of Queueing Theory*, 5th ed. Wiley, 2018.

[8] A. F. Mazri, J. K. Cheng, and Y. Liu, "Application of Discrete Event Simulation for Enhancing Queuing System at Lotus Alor Setar," *Int. J. Ind. Manag.*, vol. 19, no. 4, pp. 197–209, 2025, doi: 10.15282/ijim.19.4.2025.11959.

[9] N. Matloff, *Introduction to Discrete-Event Simulation and SimPy*. University of California, Davis, 2023.

[10] G. Casella and R. L. Berger, *Statistical Inference*, 3rd ed. Cengage Learning, 2021.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

Simulasi Penilaian Kelayakan Kredit Nasabah Menggunakan Metode Simple Additive Weighting (SAW) dan Analisis Sensitivitas pada Berbagai Variasi Bobot Kriteria

Lalu Rifqi Ramadhan [1)], I Putu Ananta Sugiartha [2)], Ahmad Madani [3)], Danang Adiwijaya [4)], Herliana Rosika[5)]
Program Studi Teknik Informatika Universitas Mataram
f1d02310071@student.unram.ac.id

Abstrak

Penilaian kelayakan kredit merupakan salah satu proses penting dalam perbankan untuk mengurangi risiko terjadinya kredit bermasalah. Salah satu metode yang sering digunakan dalam sistem pendukung keputusan adalah Simple Additive Weighting (SAW) karena memiliki proses perhitungan yang relatif sederhana dan efisien. Namun, sebagian besar penelitian sebelumnya menggunakan bobot kriteria yang bersifat tetap, sehingga belum menggambarkan kondisi nyata ketika kebijakan penilaian kredit dapat berubah sesuai kebutuhan manajemen risiko. Penelitian ini bertujuan untuk menganalisis pengaruh perubahan bobot kriteria terhadap hasil penilaian kelayakan kredit menggunakan metode SAW melalui pendekatan analisis sensitivitas. Pengujian dilakukan menggunakan German Credit Dataset yang terdiri atas 1.000 data nasabah. Beberapa skenario pembobotan diterapkan, yaitu skenario fokus finansial, bobot merata, dan fokus riwayat kredit. Hasil penelitian menunjukkan bahwa perubahan bobot kriteria mempengaruhi tingkat akurasi model. Pada pembobotan dasar, model menghasilkan akurasi sebesar 49,20%. Setelah dilakukan analisis sensitivitas, akurasi pada skenario fokus finansial sebesar 45,80%, skenario bobot merata sebesar 54,00%, dan skenario fokus riwayat kredit sebesar 47,00%. Hasil perbandingan antar skenario menunjukkan bahwa kriteria checking balance dan credit history memiliki pengaruh yang cukup besar terhadap perubahan hasil penilaian kelayakan kredit. Perubahan bobot pada kedua kriteria tersebut dapat menyebabkan perubahan peringkat maupun status kelayakan nasabah. Perlu diperhatikan bahwa nilai akurasi yang dihasilkan mencerminkan keterbatasan inheren metode SAW sebagai alat perangkingan (ranking tool), bukan sebagai sistem klasifikasi biner, sehingga fokus utama penelitian ini adalah pada perubahan relatif antar skenario, bukan pada nilai akurasi absolut. Selain itu, hasil penelitian dapat menjadi bahan pertimbangan bagi pihak perbankan dalam mengevaluasi dampak perubahan kebijakan kredit sebelum diterapkan pada sistem yang digunakan.

Kata kunci: Analisis Sensitivitas, Kelayakan Kredit, *Simple Additive Weighting*, Simulasi, Variasi Bobot.

Abstract

Creditworthiness assessment is a crucial process in banking to reduce the risk of non-performing loans. One method frequently used in decision support systems is Simple Additive Weighting (SAW) due to its relatively simple and efficient calculation process. However, most previous studies used fixed criteria weights, thus failing to reflect the real-world conditions under which credit assessment policies can change according to risk management needs. This study aims to analyze the effect of changes in criteria weights on creditworthiness assessment results using the SAW method through a sensitivity analysis approach. Testing was conducted using the German Credit Dataset, consisting of 1,000 customer data. Several weighting scenarios were applied: financial focus, even weighting, and credit history focus. The results showed that changes in criteria weights affected the model's accuracy. With the baseline weighting, the model achieved an accuracy of 49.20%. After sensitivity analysis, the accuracy in the financial focus scenario was 45.80%, the even weighting scenario 54.00%, and the credit history focus scenario 47.00%. Comparisons between scenarios indicated that the checking balance and credit history criteria significantly influenced changes in creditworthiness assessment results. Changes in the weighting of these two criteria can lead to changes in a customer's rating and eligibility status. It should be noted that the accuracy values reflect the inherent limitation of SAW as a ranking tool rather than a binary classifier; the primary contribution of this study lies in the relative changes across weighting scenarios rather than in the absolute accuracy figures. Furthermore, the research findings can be used by banks to evaluate the impact of changes in credit policy before implementing them in their systems.

Keyword: *Credit Eligibility, Sensitivity Analysis, Simple Additive Weighting, Simulation, Weight Variations.*

I. PENDAHULUAN

Penilaian kelayakan kredit merupakan salah satu proses paling krusial bagi institusi perbankan dan lembaga keuangan untuk menjaga stabilitas bisnis dan meminimalkan risiko terjadinya kredit macet (*non-performing loans*). Dalam perkembangannya, berbagai metode *Multi-Criteria Decision Making* (MCDM) telah banyak diterapkan untuk membantu memetakan keputusan ini secara terukur [1], [2]. Di antara berbagai metode tersebut, *Simple Additive Weighting* (SAW) menjadi salah satu algoritma yang paling populer dan sering diimplementasikan dalam sistem pendukung keputusan di berbagai domain [3], [7]. Keunggulan utama metode SAW terletak pada kesederhanaan formulanya dan efisiensi komputasinya yang tinggi saat menangani banyak alternatif.

Meskipun metode SAW terbukti efektif dalam melakukan perangkingan alternatif, penerapannya pada kasus nyata seperti kelayakan kredit nasabah masih memiliki keterbatasan yang signifikan. Mayoritas penelitian terdahulu menggunakan pendekatan pembobotan kriteria yang bersifat statis atau mutlak. Skema pembobotan statis ini diperoleh baik secara subjektif langsung dari pengambil keputusan, maupun secara objektif menggunakan bantuan metode lain seperti *Analytic Hierarchy Process* (AHP) [8] atau *Rank Order Centroid* (ROC) [9]. Kelemahan mendasar dari model pembobotan satu arah (statis) ini adalah ketidakmampuannya dalam mengakomodasi dinamika fluktuasi kebijakan manajemen risiko perbankan di dunia nyata, yang sangat bergantung pada kondisi ekonomi global maupun internal lembaga [10], [11]. Sebagai contoh, dalam kondisi pengetatan risiko, manajemen bank mungkin akan menaikkan bobot kriteria riwayat kredit secara signifikan, sedangkan pada kondisi pelonggaran pasar, fokus bobot dapat bergeser ke arah tingkat pendapatan nasabah.

Kondisi tersebut menunjukkan adanya *research gap* (celah penelitian) yang nyata di dalam literatur. Di satu sisi, penelitian terapan MCDM untuk penilaian kredit cenderung kaku dan mengabaikan dinamika perubahan bobot parameter [12]. Di sisi lain, kajian teoritis mengenai analisis sensitivitas bobot pada rumpun ilmu pengambilan keputusan sebagian besar masih bersifat matematis abstrak dan belum banyak diintegrasikan dengan data kasus praktis di dunia perbankan [13]. Tanpa adanya pengujian ketahanan model, manajemen risiko perbankan tidak dapat memprediksi dampak pergeseran kebijakan internal mereka terhadap stabilitas hasil penilaian kelayakan nasabah.

Oleh karena itu, penelitian ini dirancang untuk mengembangkan sebuah pendekatan simulasi eksperimen yang menguji tingkat ketahanan model SAW saat dihadapkan pada perubahan parameter bobot secara sistematis. Fokus utama dari kerja penelitian ini diarahkan pada pemodelan skenario berbasis kode program untuk mengamati bagaimana fluktuasi kebijakan pembobotan memengaruhi kestabilan peringkat alternatif nasabah secara dinamis. Melalui eksperimen ini, pergeseran bobot akan dijalankan dalam beberapa skema guna mendeteksi secara presisi batas perubahan atau titik kritis (*tipping point*) yang dapat mengubah status keputusan kelayakan nasabah dari "Layak" menjadi "Tidak Layak". Seluruh visualisasi perilaku model ini akan disajikan secara grafis, sehingga karakteristik sensitivitas tiap kriteria penilaian kredit dapat dipetakan dengan jelas. Melalui implementasi simulasi ini, sistem yang dibangun diharapkan mampu memberikan gambaran taktis dan instan bagi manajemen perbankan dalam memprediksi dampak perubahan kebijakan risiko kredit sebelum diterapkan sepenuhnya di lingkungan operasional yang nyata.

II. TINJAUAN PUSTAKA

Pengambilan keputusan untuk proses penyaluran kredit merupakan aktivitas penting sekaligus kompleks bagi institusi keuangan. Kesalahan dalam menilai karakter dan kapasitas finansial calon debitur berpotensi meningkatkan rasio kredit bermasalah (*Non-Performing Loan* / NPL) yang secara langsung mengancam stabilitas likuiditas bank. Menurut Yoshino dan Taghizadeh-Hesary, penilaian risiko kredit yang komprehensif sangat diperlukan karena performa stabilitas lembaga keuangan sangat dipengaruhi oleh kondisi makroekonomi global maupun fluktuasi internal lembaga itu sendiri [11]. Oleh karena itu, diperlukan sebuah

mekanisme penilaian risiko yang objektif, terukur, serta mampu meminimalkan unsur subjektivitas pemutus kredit yang memiliki banyak kriteria sebagai parameter penilaian.

Untuk mengatasi kompleksitas dalam menentukan keputusan yang dinamis dan kurang terstruktur, integrasi antara konsep multi-kriteria dengan Sistem Pendukung Keputusan (SPK) terbukti sangat efektif. Berbagai metode MCDM telah banyak diimplementasikan untuk melakukan pemeringkatan maupun klasifikasi data objek keputusan. Sebagai contoh, algoritma MCDM seperti *Technique for Order of Preference by Similarity to Ideal Solution* (TOPSIS) telah sukses diterapkan dalam mengoptimalkan penempatan posisi karyawan berdasarkan kesesuaian kompetensi [1]. Di samping itu, teknik pengambilan keputusan multi-kriteria lainnya seperti metode Bayes, MPE, CPI, dan *Analytic Hierarchy Process* (AHP) juga banyak dikombinasikan guna menyelesaikan permasalahan manajerial yang kompleks dan dinamis di tingkat perusahaan [2].

Sebagai alternatif yang sangat populer dalam ruang lingkup MCDM, metode *Simple Additive Weighting* (SAW) menawarkan pendekatan yang lebih baik melalui konsep penjumlahan terbobot. Karakteristik utama dari metode SAW adalah adanya proses normalisasi matriks keputusan ke suatu skala yang dapat diperbandingkan dengan seluruh rating alternatif yang ada. Fleksibilitas dan kesederhanaan formula matematis metode ini membuatnya sangat luas diterapkan pada berbagai domain SPK dengan efisiensi komputasi yang tinggi. Di antaranya adalah penerapan SAW untuk pemilihan dosen berprestasi demi menjaga objektivitas penilaian [3], pemilihan siswa terbaik berdasarkan kriteria akademik dan perilaku [4], serta sistem pendukung keputusan promosi kenaikan jabatan karyawan guna meminimalkan kesalahan input data [5].

Sehingga, akurasi pemeringkatan nasabah pada sistem penilaian kredit sangat bergantung pada penentuan bobot kriteria. Konsep pembobotan kriteria pada umumnya diperoleh melalui beberapa pendekatan, baik secara subjektif langsung dari pakar maupun objektif menggunakan bantuan metode terstruktur. Salah satu pendekatan terstruktur yang sering digunakan untuk penentuan struktur hierarki kriteria atau kepentingan fungsi adalah metode AHP, seperti pada kasus seleksi pemasok [8]. Di sisi lain, dalam domain penilaian kelayakan kredit, pembobotan kriteria dapat dirumuskan menggunakan metode *Rank Order Centroid* (ROC) yang dipadukan dengan SAW (ROC-SAW) untuk menentukan prioritas penerima kredit berdasarkan urutan kepentingan kriteria yang dinilai oleh pakar [9]. Untuk menjamin stabilitas, keandalan, dan validitas dari rekomendasi yang dihasilkan oleh model keputusan berbasis SAW, pelaksanaan analisis sensitivitas (*sensitivity analysis*) menjadi tahapan yang wajib dilakukan. Analisis sensitivitas berfungsi untuk menguji seberapa kuat atau rentan peringkat alternatif nasabah (debitur) berubah apabila terjadi variasi atau pergeseran pada koefisien bobot kriteria [13].

III. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan *Computational Experimental Research* dengan memanfaatkan simulasi berbasis komputer untuk menganalisis pengaruh perubahan bobot kriteria terhadap hasil penilaian kelayakan kredit menggunakan metode *Simple Additive Weighting* (SAW). Seluruh proses simulasi diimplementasikan menggunakan bahasa pemrograman Python pada lingkungan Google Colab dengan memanfaatkan German Credit Dataset sebagai data uji.

3.1 Definisi Matriks Keputusan

Sistem penilaian ini memodelkan himpunan alternatif calon nasabah $A = \{A_1, A_2, \ldots, A_m\}$ sebanyak $m = 1000$

nasabah, terhadap himpunan kriteria keputusan $C_m = \{C_1, C_2, \ldots, C_n\}$ sebanyak $n = 12$ kriteria. Hubungan antara alternatif dan kriteria direpresentasikan ke dalam bentuk Matriks keputusan X berukuran $m \times n$ sebagai berikut:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Dimana x_{ij} merupakan nilai mentah (*raw score*) dari alternatif ke-i pada kriteria ke-j setelah melalui proses ordinal mapping

3.2 Normalisasi Matriks Keputusan

Untuk menyelaraskan skala parameter yang berbeda, matriks keputusan X ditransformasikan menjadi matriks normalisasi $R = [r_{ij}]_{m \times n}$ menggunakan prinsip metode *Simple Additive Weighting* (SAW). Formulasi pembagian kriteria keuntungan (benefit criteria, J_B) dan kriteria biaya (cost criteria, J_C) didefinisikan sebagai:

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{\max_i(x_{ij})}, & \text{Jika } j \in J_B \\ \dfrac{\min_i(x_{ij})}{x_{ij}}, & \text{Jika } j \in J_C \end{cases}$$

Dimana $max_i(x_{ij})$ adalah nilai tertinggi pada kriteria j dan $min_i(x_{ij})$ adalah nilai terendah pada kriteria j.

3.3 Perhitungan Skor Preferensi Acuan

Vektor bobot kriteria dinyatakan sebagai $W = [w_i, w_2, \ldots, w_n]$, dengan syarat batas total bobot yang memenuhi integrasi normalisasi berikut:

$$\sum_{j=1}^{n} w_j = 1$$

Skor preferensi akhir (V_i) untuk setiap alternatif A_i dihitung melalui fungsi kombinasi linear terbobot:

$$V_i = \sum_{j=1}^{n} w_j \cdot r_{ij}$$

3.4 *Classification Threshold*

Penentuan status akhir kelayakan kredit nasabah (K_i) dievaluasi secara dinamis menggunakan persentil ke-30 dari seluruh populasi skor preferensi sebagai ambang batas (threshold). Nilai persentil ini dipilih untuk mencerminkan proporsi kelas aktual dataset *German Credit*, yaitu 70% nasabah berkelas Layak dan 30% berkelas Tidak Layak, sehingga keputusan klasifikasi tidak dipaksa terdistribusi secara simetris:

3.5 Simulasi Analisis Sensitivas

Eksperimen komputasi dijalankan secara iteratif dengan memodifikasi komponen vektor bobot dasar menjadi vektor bobot skenario baru $W^k = [w_1^k, w_2^k, \ldots, w_n^k]$, di mana $k \in \{Fokus\ Finansial, Bobot\ Merata, Fokus\ Riwayat\ Kredit\}$. Gejala pembalikan peringkat (rank reversal) pada setiap alternatif diukur berdasarkan nilai absolut perubahan urutan indeks preferensi:

$$\Delta Rank_i = \left| Rank\left(V_i^{baseline}\right) - Rank\left(V_i^k\right) \right|$$

Sedangkan tingkat akurasi (Acc) dari fungsi keputusan terhadap label target aktual data (Y_i) pada setiap skenario dihitung menggunakan fungsi indikator:

$$Acc^k = \frac{1}{m} \sum_{i=1}^{m} I\left(K_i^k = Y_i\right)$$

Seluruh rangkaian matriks transisi hasil iterasi parameter tersebut kemudian diekstrak ke dalam representasi visual grafis untuk mendeteksi fluktuasi stabilitas model secara instan sebelum diimplementasikan pada lingkungan manajemen risiko operasional perbankan yang nyata.

IV. HASIL DAN PEMBAHASAN

Pengujian model dilakukan menggunakan dataset German Credit yang memuat 1.000 data calon debitur dengan 20 atribut penilaian serta 1 atribut target. Dalam penelitian ini, tidak seluruh atribut pada dataset digunakan sebagai dasar pengambilan keputusan. Pemilihan atribut dilakukan dengan mempertimbangkan keterkaitannya terhadap proses penilaian kelayakan kredit. Hasil seleksi menghasilkan 12 kriteria yang digunakan dalam penelitian, terdiri atas 8 kriteria bertipe *benefit*.

Pemilihan kedua belas kriteria tersebut mengacu pada aspek-aspek yang umum digunakan dalam proses analisis kredit perbankan, khususnya yang berkaitan dengan prinsip 5C. Kriteria yang dipilih dinilai mampu menggambarkan kondisi keuangan calon debitur, riwayat pembayaran kredit, serta kepemilikan aset yang dapat mendukung tingkat kepercayaan dalam pemberian kredit. Selain itu, atribut yang dianggap kurang berpengaruh tidak disertakan dalam proses analisis agar data yang digunakan lebih relevan. Langkah ini juga membantu mengurangi kompleksitas pengolahan data sehingga proses komputasi dapat dilakukan dengan lebih efisien.

Tabel 1. Atribut Dataset

Kriteria	Tipe Kriteria	Presentase
checking_balance	Benefit	15%
credit_history	Benefit	13%
savings_balance	Benefit	12%
employment_length	Benefit	10%
amount	Cost	10%
property	Benefit	8%
months_loans_duration	Cost	8%
housing	Benefit	7%
age	Benefit	5%
job	Benefit	5%
installment_rate	Cost	5%
other_debtors	Cost	2%

Sebelum proses perhitungan SAW dilakukan, seluruh atribut yang masih berbentuk kategorikal terlebih dahulu dikonversi ke dalam bentuk numerik. Proses konversi menggunakan pendekatan *ordinal mapping* sehingga urutan dan tingkat risiko dari setiap kategori tetap dapat dipertahankan. Pada tahap awal pengujian, digunakan skema bobot dasar (*baseline*) yang disusun berdasarkan pertimbangan terhadap kondisi finansial dan karakteristik calon nasabah.

Selanjutnya dilakukan proses normalisasi menggunakan metode Simple Additive Weighting (SAW), di mana kriteria benefit dinormalisasi menggunakan nilai maksimum dan kriteria cost dinormalisasi menggunakan nilai minimum.

```
# Normalisasi matriks keputusan
df_saw = df_encoded[all_criteria].copy().astype(float)

df_saw = df_saw.replace(0, 0.001)

df_norm = pd.DataFrame(index=df_saw.index)

for col in benefit_criteria:
    max_val = df_saw[col].max()
    df_norm[col] = df_saw[col] / max_val  # r_ij = x_ij / max_j

for col in cost_criteria:
    min_val = df_saw[col].min()
    df_norm[col] = min_val / df_saw[col]  # r_ij = min_j / x_ij

df_norm.head().round(4)
```

Gambar 1. Kode Program Normalisasi Matriks Keputusan

Selanjutnya dilakukan perhitungan skor SAW yang menghasilkan skor preferensi (SAW Score) untuk setiap nasabah yang kemudian digunakan sebagai dasar pemeringkatan dan penentuan status kelayakan kredit. Untuk menentukan keputusan kredit, digunakan nilai median skor SAW sebagai ambang batas (threshold). Nasabah yang memiliki skor di atas threshold dikategorikan sebagai "Layak", sedangkan nasabah dengan skor di bawah threshold dikategorikan sebagai "Tidak Layak".

```
Statistik skor SAW:
count    1000.0000
mean        0.4398
std         0.0839
min         0.2006
25%         0.3832
50%         0.4385
75%         0.4949
max         0.7248
dtype: float64

Threshold (median): 0.3950

Prediksi SAW:
saw_label
Layak          700
Tidak Layak    300
Name: count, dtype: int64
```

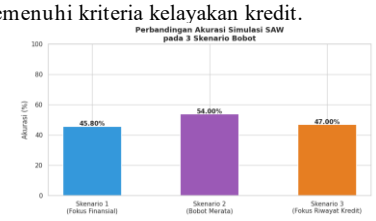
Commented [1]: Buatkan perhitungan dan analisis terkait dengan pemilihan atribut bobot kriterianya

Gambar 2. Statistik Skor SAW dan Nilai Threshold

Untuk mengamati pengaruh perubahan kebijakan penilaian kredit, dilakukan simulasi menggunakan beberapa skenario pembobotan yang berbeda. Simulasi dijalankan secara komputasional menggunakan Python pada Google Colab dengan memodifikasi proporsi bobot setiap kriteria tanpa mengubah struktur data nasabah. Terdapat tiga skenario pembobotan yang digunakan dalam penelitian ini, yaitu:

1. Skenario Fokus Finansial, yang memberikan bobot lebih besar pada checking_balance dan savings_balance.
2. Skenario Bobot Merata, yang memberikan distribusi bobot yang sama pada seluruh kriteria.
3. Skenario Fokus Riwayat Kredit, yang memberikan bobot dominan pada credit_history.

Setiap skenario menghasilkan nilai preferensi baru yang kemudian digunakan untuk menentukan status kelayakan kredit seluruh nasabah. Selain menghasilkan skor SAW yang berbeda, perubahan bobot juga mempengaruhi urutan prioritas nasabah dan jumlah nasabah yang memenuhi kriteria kelayakan kredit.



Gambar 3. Perbandingan Akurasi Simulasi SAW pada 3 Skenario Bobot

Berdasarkan hasil simulasi, skenario bobot merata menghasilkan tingkat kesesuaian tertinggi terhadap data aktual dibandingkan dua skenario lainnya. Skenario ini memperoleh nilai akurasi sebesar 54,00%, sedangkan skenario fokus finansial menghasilkan akurasi sebesar 45,80% dan skenario fokus riwayat kredit menghasilkan akurasi sebesar 46,80%.

Perlu dicatat bahwa nilai akurasi yang dihasilkan pada seluruh skenario berkisar antara 45–54%, yang tergolong rendah jika dibandingkan dengan metode klasifikasi machine learning konvensional. Hal ini disebabkan karena metode SAW pada dasarnya merupakan alat perangkingan (ranking tool), bukan sistem klasifikasi biner. Penerapan ambang batas (threshold) pada skor SAW merupakan adaptasi untuk keperluan evaluasi, sehingga nilai akurasi tidak menjadi tolok ukur utama keberhasilan metode ini. Fokus utama penelitian ini adalah pada perubahan relatif akurasi dan peringkat nasabah ketika bobot kriteria divariasikan, bukan pada nilai akurasi absolut. Oleh karena itu, temuan yang paling relevan adalah bahwa pergeseran bobot dari satu skenario ke skenario lain secara konsisten mengubah urutan prioritas nasabah dan status kelayakan sebagian dari mereka. Berikut 5 Nasabah Paling Layak dan Paling Tidak Layak Berdasarkan Skor SAW:

Tabel 2. 5 Nasabah Paling Layak

ID Nasabah	Checking Balance	Credit History	Amount
27	> 200 DM	Fully repaid this bank	409
502	> 200 DM	Repaid	1126
94	1 - 200 DM	Repaid	1318
109	1 - 200 DM	Repaid	1410
140	> 200 DM	Repaid	709

Tabel 3. 5 Nasabah Paling Tidak Layak

ID Nasabah	Checking Balance	Credit History	Amount
683	Unknown	Critical	5103
549	Unknown	Critical	8858
857	Unknown	Critical	3343
59	< 0 DM	Critical	6229
665	Unknown	Critical	6314

V. KESIMPULAN DAN SARAN

Hasil simulasi menunjukkan bahwa perubahan bobot kriteria memberikan pengaruh yang terukur terhadap stabilitas hasil penilaian kelayakan kredit menggunakan metode SAW. Skenario bobot merata menghasilkan tingkat kesesuaian tertinggi terhadap data aktual (54,00%), diikuti skenario fokus riwayat kredit (47,00%) dan fokus finansial (45,80%). Rentang perubahan akurasi sebesar ±8% antar skenario mengonfirmasi bahwa keputusan kredit merupakan hasil interaksi dari berbagai faktor secara bersamaan, bukan ditentukan oleh satu kriteria dominan. Temuan ini sekaligus menunjukkan bahwa metode SAW lebih tepat diposisikan sebagai alat simulasi kebijakan yang memungkinkan manajemen perbankan mengamati dampak pergeseran bobot kriteria sebelum diterapkan pada sistem operasional nyata dibandingkan sebagai sistem klasifikasi biner dengan akurasi tinggi.

Untuk penelitian selanjutnya, disarankan agar penentuan bobot kriteria dilakukan secara lebih objektif menggunakan metode seperti Entropy Weight Method atau AHP berbasis data, guna mengurangi unsur subjektivitas. Selain itu, perbandingan dengan metode MCDM lain seperti TOPSIS atau VIKOR serta penggunaan threshold yang dioptimalkan melalui kurva ROC dapat menjadi langkah pengembangan yang relevan untuk meningkatkan validitas dan keandalan sistem penilaian kelayakan kredit di masa mendatang.

VI. UCAPAN TERIMAKASIH

Terima kasih kepada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram atas dukungan akademik yang diberikan selama pelaksanaan penelitian ini. Kami menyampaikan apresiasi kepada dosen pengampu Mata Kuliah Pemodelan dan Simulasi yang telah memberikan arahan, masukan, dan bimbingan selama proses penyusunan artikel ilmiah ini. Tidak lupa, penulis mengucapkan terima kasih kepada seluruh pihak yang telah membantu dalam penyediaan data, pengembangan sistem simulasi, serta penyusunan artikel ilmiah ini sehingga penelitian dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

[1] A. N. Pramudhita, H. Suyono, dan E. Yudaningtyas, "Penggunaan Algoritma Multi Criteria Decision Making dengan Metode Topsis dalam Penempatan Karyawan," *Jurnal Ilmiah*, vol. x, no. x, pp. xx-xx, Tahun. *(Asal file: Penggunaan Algoritma Multi Criteria Decision Making dengan Metode Topsis dalam Penempatan Karyawan.pdf)*

[2] A. H. Rangkuti, "Teknik Pengambilan Keputusan Multi Kriteria Menggunakan Metode Bayes, MPE, CPI dan AHP," *ComTech*, vol. 2, no. 1, pp. 229-238, Juni 2011.

[3] D. M. Rajagukguk dan R. Limbong, "Implementasi Metode Simple Additive Weighting (SAW) pada Sistem Pendukung Keputusan Pemilihan Dosen Berprestasi," *Means (Media Informasi Analisa dan Sistem)*, vol. 2, no. 2, Desember 2017.

[4] A. Setiadi, Yunita, dan A. R. Ningsih, "Penerapan Metode Simple Additive Weighting(SAW) untuk Pemilihan Siswa Terbaik," *Jurnal SISFOKOM*, vol. 7, no. 2, September 2018.

[5] Frieyadie, "Penerapan Metode Simple Additive Weight (SAW) dalam Sistem Pendukung Keputusan Promosi Kenaikan Jabatan," *Jurnal Pilar Nusa Mandiri*, vol. 12, no. 1, pp. 37-45, Maret 2016.

[6] E. Ridhawati, G. K. Siregar, dan D. Iriawan, "Metode Simple Additive Weighting (SAW) pada Sistem Pendukung Keputusan Penilai Kinerja Guru (PKG) (Studi Kasus SMP 17 1

Pagelaran)," *Jurnal Informasi Dan Komputer*, vol. 6, no. 2, 2018.

[7] M. Muhammad, N. Safriadi, dan N. Prihartini, "Implementasi Metode Simple Additive Weighting(SAW) pada Sistem Pendukung Keputusan dalam Menentukan Prioritas Perbaikan Jalan," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 5, no. 4, 2017.

[8] W. Habsari, T. Djatna, F. Udin, dan Y. Arkeman, "Pendekatan Pengambilan Keputusan Multi Kriteria Menggunakan AHP untuk Seleksi Pemasok Kemasan Pudak," *Jurnal Teknologi Industri Pertanian*, vol. 32, no. 2, pp. 197-203, Agustus 2022.

[9] B. K. Wijaya, I G. I. Sudipa, Rachmat, I K. Wiratama, dan N. M. C. Sasri, "Sistem Penentuan Keputusan Kelayakan Penerima Kredit Menggunakan Metode ROC - SAW," *JUISIK (Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer)*, vol. 2, no. 2, Juli 2022.

[10] A. Nadali, S. Pourdarab, dan H. E. Nosratabadi, "Class Labeling of Bank Credit's Customers Using AHP and SAW for Credit Scoring with Data Mining Algorithms," *International Journal of Computer Theory and Engineering*, vol. 4, no. 3, June 2012.

[11] N. Yoshino dan F. Taghizadeh-Hesary, "A Comprehensive Method For The Credit Risk Assessment Of Small And Medium-Sized Enterprises Based On Asian Data," *ADBI Working Paper Series*, no. 907, Tokyo: Asian Development Bank Institute, December 2018.

[12] M. T. I. Ma'arif, "Penerapan Metode Simple Additive Weighting (SAW) dalam Simulasi Operasional Supermarket," Skripsi, Program Studi Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Malang, 2025.

[13] G. Demir dan R. Arslan, "Sensitivity Analysis in Multi-Criteria Decision-Making Problems," *Ankara Hacı Bayram Veli Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 24, no. 3, pp. 1025-1056, 2022.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

PREDIKSI JENIS TANAMAN YANG COCOK BERDASARKAN KONDISI TANAH MENGGUNAKAN ANN YANG DIOPTIMALKAN DENGAN GRID SEARCH DAN DROPOUT

Didy Ardiyanto [1)], Lutfi Alfarizi.[2)] , Lalu Ahmad Purwadi.[3)]
Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: ardindidy@gmail.com , contohakun3562@gmail.com , lutfial1908@gmail.com

Abstrak

Pemilihan jenis tanaman yang sesuai dengan kondisi tanah merupakan faktor penting dalam meningkatkan produktivitas pertanian. Tanah memiliki karakteristik yang beragam seperti pH, kadar kelembapan, kandungan nitrogen (N), fosfor (P), dan kalium (K) yang berpengaruh langsung terhadap pertumbuhan tanaman. Penelitian ini bertujuan untuk menerapkan *Jaringan Syaraf Tiruan (Artificial Neural Network/ANN)* dalam memprediksi jenis tanaman yang paling sesuai berdasarkan data kondisi tanah. Metode ANN dipilih karena kemampuannya dalam mengenali hubungan non-linear antar variabel yang kompleks serta menghasilkan klasifikasi yang akurat. Dataset yang digunakan berasal dari *Crop Recommendation Dataset* yang tersedia di platform *Kaggle*, berisi parameter tanah dan jenis tanaman yang ideal untuk ditanam. Hasil kajian literatur menunjukkan bahwa penerapan ANN mampu mencapai akurasi di atas 95% dalam klasifikasi tanaman. Dengan pendekatan ini, sistem prediksi berbasis ANN diharapkan dapat membantu petani dalam pengambilan keputusan tanam yang lebih efisien dan adaptif terhadap kondisi lingkungan.
Kata kunci: *Artificial Neural Network*; Jaringan Syaraf Tiruan; kondisi tanah; prediksi tanaman; pertanian cerdas.

Abstract

*Selecting the appropriate crop species based on soil conditions is a crucial factor in enhancing agricultural productivity. Soil possesses diverse characteristics, such as pH, moisture levels, and the content of nitrogen (N), phosphorus (P), and potassium (K), which directly influence crop growth. This study aims to implement an Artificial Neural Network (ANN) to predict the most suitable crop types based on soil condition data. The ANN method was selected due to its capability to recognize complex, non-linear relationships among variables and generate accurate classifications. The dataset utilized is derived from the Crop Recommendation Dataset available on the Kaggle platform, comprising soil parameters and the corresponding ideal crops for cultivation. Findings from the literature review indicate that the application of ANN can achieve an accuracy exceeding **95%** in crop classification. Through this approach, the ANN-based prediction system is expected to assist farmers in making more efficient planting decisions that are highly adaptive to environmental conditions.*
Keyword: *Artificial Neural Network; Soil Conditions; Crop Prediction; Smart Agriculture.*

I. PENDAHULUAN

Sektor pertanian merupakan pilar utama dalam menjaga ketahanan pangan nasional. Salah satu tantangan krusial yang dihadapi petani adalah menentukan jenis tanaman yang paling sesuai dengan kondisi fisik dan kimiawi tanah di lokasi spesifik [1]. Kesalahan dalam pemilihan jenis tanaman, yang acap kali masih menggunakan pendekatan coba-coba (*trial and error*), dapat memicu penurunan drastis pada hasil panen serta memicu ketidakefisienan pemanfaatan lahan. Oleh karena itu, diperlukan sebuah sistem komputasi cerdas yang dapat menganalisis karakteristik tanah secara mendalam—seperti tingkat pH, kadar kelembapan, serta kandungan unsur hara makro yakni nitrogen (N), fosfor (P), dan kalium (K)—untuk memberikan rekomendasi komoditas pertanian secara otomatis dan akurat.

Seiring dengan masifnya adopsi teknologi, *Artificial Intelligence* (AI) dan *Machine Learning* telah menjadi tulang punggung dalam mendukung konsep pertanian cerdas, termasuk dalam analisis kesesuaian lahan dan prediksi tanaman yang cerdas [7f]. Salah satu algoritma yang sangat mumpuni adalah *Artificial Neural Network* (ANN), sebuah Jaringan Syaraf Tiruan yang dirancang untuk mampu belajar dari data historis guna memetakan hubungan non-linear yang sangat kompleks antara parameter input (kondisi lingkungan) dan variabel target (rekomendasi

tanaman) [2], [4]. ANN telah terbukti handal dalam melakukan tugas klasifikasi presisi tinggi untuk sektor pertanian [3]. Namun, implementasi model ANN standar pada penelitian-penelitian terdahulu sering kali terhambat oleh arsitektur yang rentan mengalami *overfitting*—di mana model sangat baik pada data latih namun gagal beradaptasi pada data baru—serta sulitnya menemukan kombinasi *hyperparameter* yang paling optimal secara manual.

Untuk mengatasi limitasi tersebut, penelitian ini bertujuan untuk mengkaji, membangun, dan menerapkan arsitektur *Artificial Neural Network* (ANN) dalam memprediksi kecocokan jenis tanaman, dengan mengintegrasikan algoritma *Grid Search* dan *Dropout layer*. *Grid Search* diterapkan untuk mencari dan mengevaluasi konfigurasi parameter secara komprehensif, sementara *Dropout* diimplementasikan untuk menonaktifkan sebagian neuron secara acak guna meminimalisasi risiko *overfitting*.

Hasil dari penelitian ini diharapkan dapat menjadi fondasi teknis bagi pengembangan sistem rekomendasi tanam yang adaptif dan berbasis data (*data-driven*). Bagi para petani dan pemangku kepentingan, sistem ini akan memberikan panduan praktis yang secara signifikan dapat meningkatkan efisiensi lahan, memaksimalkan produktivitas komoditas, dan mempercepat transisi menuju pertanian presisi (*precision agriculture*).

Hipotesis utama dalam penelitian ini adalah bahwa optimasi arsitektur ANN menggunakan teknik *Grid Search* dan *Dropout* akan menghasilkan kemampuan prediksi jenis tanaman yang secara empiris lebih akurat, lebih stabil, serta memiliki daya generalisasi yang jauh lebih baik dibandingkan dengan penggunaan model ANN standar.

II. TINJAUAN PUSTAKA

Beberapa penelitian terkini menunjukkan keberhasilan penerapan Jaringan Syaraf Tiruan (JST) dalam sistem rekomendasi tanaman berdasarkan data tanah dan iklim. Penelitian oleh Patel et al. [2] telah membuktikan hal ini dengan menerapkan model *Feedforward Neural Network* menggunakan *Crop Recommendation Dataset* yang bersumber dari platform Kaggle. Penelitian tersebut berhasil mencapai tingkat akurasi sebesar 96% dan menghasilkan model yang stabil untuk klasifikasi data tabular. Namun, model yang diusulkan memiliki kelemahan utama, yakni tidak adanya tahapan optimisasi parameter yang spesifik, sehingga model tersebut masih berisiko tinggi mengalami *overfitting* saat dihadapkan pada data baru.

Dalam konteks analisis lokal, Ernawati et al. [3] merancang model prediksi kesesuaian lahan budidaya tanaman pangan menggunakan algoritma *Backpropagation Neural Network*. Dengan menggunakan data analisis lahan dari Sub-DAS Bengkulu Hilir, model ini berhasil mencatatkan akurasi sebesar 95,8%, yang semakin menegaskan bahwa ANN sangat efektif untuk memprediksi karakteristik lahan. Sayangnya, model yang dibangun belum mengimplementasikan teknik regularisasi maupun optimisasi parameter, yang berpotensi menyebabkan model hanya menghafal pola pada data latih.

Hal yang sama juga ditemukan pada penelitian Madhuri dan Indiramma [4] yang mengaplikasikan ANN untuk sistem rekomendasi tanaman terintegrasi. Menggunakan data kondisi tanah dan iklim dari wilayah Hadonahalli dan Durgenahalli di India, sistem ini mencapai akurasi 96%. Meskipun sangat relevan dengan fokus prediksi tanaman, penelitian ini masih menggunakan dataset yang relatif terbatas dan belum menerapkan optimisasi *hyperparameter* atau regularisasi, sehingga daya generalisasi modelnya dinilai masih rendah. Serupa dengan itu, penelitian dari Bangaru Kamatchi et al. (2021) yang menggunakan data agrometeorologi lokal juga berhasil mencapai akurasi 95% dengan metode ANN, namun masih menyisakan celah penelitian karena belum menyertakan teknik regularisasi seperti *Dropout*.

Selain JST berbasis *Deep Learning*, metode *Machine Learning* lainnya juga kerap dikaji dalam ranah pertanian presisi. Abdul Rahman et al. [5] menguji metode *ensemble learning* dengan membandingkan algoritma *Random Forest* dan *XGBoost Regressor* untuk rekomendasi jenis tanaman. Dengan menggunakan 2.200 baris data dari 22 klasifikasi tanaman, penelitian ini menghasilkan akurasi yang impresif mencapai 98%. Walaupun performanya

tinggi, penelitian tersebut murni berfokus pada perbandingan algoritma ansambel tanpa menyentuh optimisasi arsitektur jaringan syaraf. Di ranah analisis spasial, Sarastika et al. [6] memadukan ANN dengan *Cellular Automata* (ANN-CA) menggunakan data ekstraksi citra SPOT secara temporal. Meskipun efektif memprediksi konversi lahan dengan akurasi 86,66%, model prediktif tersebut belum mempertimbangkan variabel iklim dan belum dioptimalkan dengan teknik *regularization* untuk menjamin stabilitas hasil prediksi.

Berdasarkan kajian terhadap berbagai literatur di atas, mayoritas penelitian telah membuktikan kapabilitas algoritma klasifikasi cerdas dalam memetakan kompleksitas kondisi tanah. Akan tetapi, terdapat kesenjangan teknis (*research gap*) yang konsisten di seluruh studi terdahulu: absennya optimisasi *hyperparameter* yang sistematis dan tidak digunakannya teknik regularisasi *Dropout*. Kesenjangan ini menjadi landasan kuat bagi penelitian ini untuk mengajukan arsitektur ANN yang lebih tangguh. Model yang diusulkan akan dioptimalkan secara menyeluruh menggunakan metode *Grid Search* untuk pencarian parameter ideal dan mengimplementasikan lapisan *Dropout* guna mencegah *overfitting*, sehingga diharapkan mampu menghasilkan performa prediksi yang jauh lebih stabil dan akurat.

III. METODOLOGI PENELITIAN

Penelitian ini dilakukan melalui pendekatan *studi literatur dan simulasi data* untuk memahami efektivitas penerapan ANN dalam prediksi jenis tanaman berdasarkan kondisi tanah. Literatur dikumpulkan dari basis data ilmiah seperti Google Scholar, ScienceDirect, dan IEEE Xplore menggunakan kata kunci *"crop recommendation using ANN"*, *"soil classification neural network"*, dan *"AI for precision agriculture"*.

Kriteria pemilihan literatur meliputi:

1. Membahas penerapan ANN untuk klasifikasi tanaman berdasarkan data tanah.
2. Menjelaskan struktur jaringan, parameter, serta algoritma pelatihan yang digunakan.
3. Menyertakan dataset dan hasil evaluasi (akurasi, presisi, recall, F1-score).
4. Fokus pada penelitian tahun 2020–2024 untuk menjaga relevansi.

Metode pelatihan model ANN mencakup tahapan:

1. **Input Layer:** menerima data fitur tanah (pH, N, P, K, kelembapan, suhu).
2. **Hidden Layer:** menggunakan fungsi aktivasi *ReLU*.
3. **Output Layer:** memprediksi jenis tanaman (misalnya padi, jagung, gandum, tebu).
4. **Optimisasi Internal:** algoritma *Adam Optimizer* dan fungsi *categorical cross-entropy loss*.
5. **Optimasi Eksternal dan Regulasi:** Teknik *Grid Search* digunakan untuk mencari kombinasi hiperparameter terbaik seperti jumlah neuron, fungsi aktivasi, learning rate, dan ukuran batch. Setelah parameter optimal diperoleh, ditambahkan *Dropout Layer* dengan tingkat dropout sebesar 0.2–0.3 pada setiap lapisan tersembunyi untuk mencegah *overfitting*. Kombinasi kedua teknik ini diharapkan meningkatkan stabilitas pelatihan dan akurasi model.

Evaluasi dilakukan dengan menggunakan pembagian data *train-test split* (80:20) dan metrik performa utama adalah akurasi dan *confusion matrix*.

IV. HASIL DAN PEMBAHASAN

4.1 Hasil Pelatihan Model

Penelitian ini mengimplementasikan model Artificial Neural Network (ANN) untuk melakukan klasifikasi dan rekomendasi tanaman berdasarkan kondisi tanah dan lingkungan. Dataset yang digunakan terdiri dari enam parameter masukan, yaitu Nitrogen (N), Fosfor (P), Kalium (K), suhu (temperature), kelembapan (humidity), dan tingkat keasaman tanah (pH). Model dirancang untuk mengklasifikasikan 22 jenis tanaman yang berbeda.

Sebelum proses pelatihan dilakukan, data dibagi menjadi data latih sebesar 80% dan data uji sebesar 20% menggunakan metode stratified split agar distribusi kelas tetap seimbang. Selanjutnya dilakukan proses normalisasi data menggunakan StandardScaler sehingga seluruh fitur berada pada skala yang seragam.

Untuk memperoleh konfigurasi model terbaik, dilakukan Grid Search terhadap beberapa kombinasi hyperparameter yang meliputi jumlah neuron, learning rate, batch size, dan dropout rate. Sebanyak 36 kombinasi parameter diuji selama proses optimasi.

Hasil Grid Search menunjukkan bahwa konfigurasi terbaik diperoleh dengan parameter sebagai berikut:

Parameter	Nilai Output
Jumlah Neuron	128
Learning Rate	0.01
Batch Size	64
Dropout Rate	0.2

Dengan konfigurasi tersebut, model mampu mencapai akurasi pengujian sebesar **96,59%**.

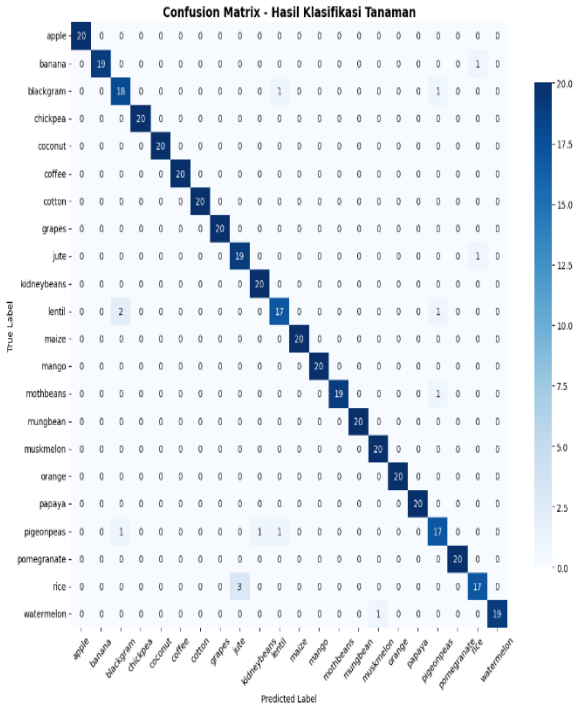
4.2 Evaluasi Kinerja Model

Evaluasi model dilakukan menggunakan confusion matrix, accuracy, precision, recall, dan F1-score. Hasil pengujian menunjukkan bahwa model memiliki performa yang sangat baik dalam mengklasifikasikan tanaman berdasarkan kondisi tanah yang diberikan.

Nilai evaluasi keseluruhan ditunjukkan pada Tabel berikut.

Metrik	Nilai
Accuracy	96.59 %
Precision Macro	96.70 %
Reccal Macro	96.60 %
F1-Score Macro	96.60 %
Precision Weighted	96,70%
Recall Weighted	96,60%
F1-Score Weighted	96,60%

Berdasarkan hasil tersebut dapat dilihat bahwa nilai precision, recall, dan F1-score memiliki nilai yang relatif seimbang. Hal ini menunjukkan bahwa model tidak mengalami bias terhadap kelas tertentu dan mampu melakukan klasifikasi dengan tingkat konsistensi yang tinggi.



Gambar 4.1 Confusion Matrix Klasifikasi

Berdasarkan confusion matrix, sebagian besar data berhasil diklasifikasikan dengan benar. Kesalahan klasifikasi hanya terjadi pada beberapa kelas tertentu seperti blackgram, lentil, pigeonpeas, dan rice. Namun jumlah kesalahan tersebut relatif kecil dibandingkan total data pengujian sehingga tidak memberikan pengaruh signifikan terhadap akurasi keseluruhan model.

4.3 Analisis Performa Setiap Kelas

Pengujian dilakukan terhadap 22 kelas tanaman yang tersedia pada dataset. Sebagian besar kelas memperoleh nilai akurasi 100%, menunjukkan bahwa model mampu mengenali karakteristik masing-masing tanaman dengan sangat baik.

Beberapa tanaman yang memperoleh akurasi sempurna antara lain:

- Apple
- Chickpea
- Coconut
- Coffee
- Cotton
- Grapes
- Maize
- Mango
- Mungbean
- Orange
- Papaya
- Pomegranate

Sementara itu beberapa kelas memiliki tingkat akurasi yang sedikit lebih rendah, yaitu:

Tanaman	**Akurasi**
Blackgram	90 %
Lentil	85 %
Pigeonpeas	85 %
Rice	85 %
Banana	95%
Jute	95%
Mothbeans	95%
Watermelon	95%

Perbedaan performa tersebut menunjukkan bahwa beberapa tanaman memiliki karakteristik fitur yang saling berdekatan sehingga model mengalami kesulitan dalam membedakannya secara sempurna.

4.4 Analisis Feature Importance

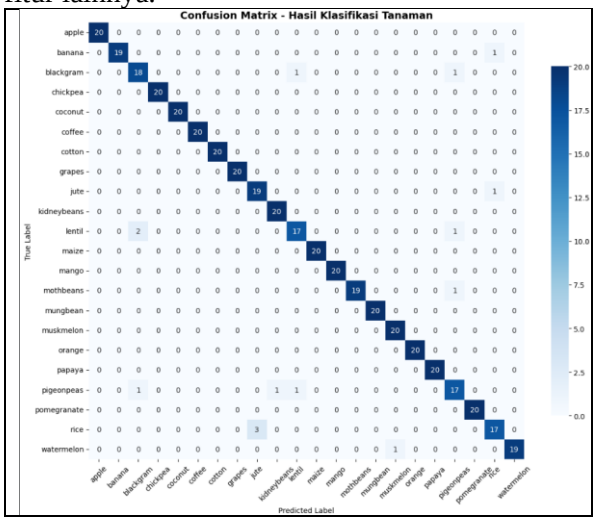
Untuk mengetahui pengaruh masing-masing fitur terhadap keputusan model, dilakukan analisis menggunakan metode Permutation Feature Importance. Metode ini mengukur penurunan akurasi ketika suatu fitur diacak secara acak (permutasi).

Hasil analisis feature importance ditunjukkan pada Tabel berikut.

Fitur	Importance Score
Humidity	0,2374
P	0,2134
K	0,2078
N	0,1668
Temperature	0,1287
pH	0,0459

Tabel di atas menunjukkan tingkat kontribusi masing-masing fitur terhadap proses pengambilan keputusan model ANN berdasarkan metode Permutation Feature Importance. Semakin besar nilai importance score yang dimiliki suatu fitur, maka semakin besar pula pengaruh fitur tersebut terhadap hasil prediksi tanaman. Sebaliknya, fitur dengan nilai importance yang lebih kecil memiliki kontribusi yang relatif lebih rendah dalam menentukan kelas tanaman yang direkomendasikan oleh sistem.

Berdasarkan hasil analisis, fitur humidity memiliki nilai importance tertinggi yaitu sebesar 0,2374, diikuti oleh fitur P (fosfor) sebesar 0,2134 dan K (kalium) sebesar 0,2078. Hal ini menunjukkan bahwa kondisi kelembapan lingkungan dan kandungan unsur hara tanah menjadi faktor utama yang dipertimbangkan model dalam menentukan jenis tanaman yang sesuai. Sementara itu, fitur **pH** memiliki nilai importance paling rendah yaitu **0,0459**, namun tetap memberikan kontribusi terhadap proses klasifikasi ketika dikombinasikan dengan fitur-fitur lainnya.



Gambar 4.2 Feature Importance Hasil Permutation Importance

Gambar 4.2 memperlihatkan visualisasi tingkat kepentingan masing-masing fitur terhadap hasil klasifikasi tanaman. Grafik menunjukkan bahwa fitur humidity memiliki kontribusi terbesar terhadap prediksi model, diikuti oleh kandungan unsur hara P dan K. Sementara itu, fitur pH memiliki kontribusi paling kecil dibandingkan fitur lainnya. Hasil ini konsisten dengan nilai importance score pada Tabel 4.4 dan menunjukkan bahwa kondisi kelembapan lingkungan serta kandungan unsur hara tanah merupakan faktor utama yang digunakan model ANN dalam menentukan rekomendasi tanaman.

Berdasarkan hasil tersebut dapat diketahui bahwa fitur humidity memiliki pengaruh terbesar terhadap proses klasifikasi tanaman dengan nilai importance sebesar 0,2374. Hal ini menunjukkan bahwa tingkat kelembapan menjadi faktor yang paling dominan dalam menentukan jenis tanaman yang sesuai.

Sebaliknya, fitur pH memiliki pengaruh paling kecil dengan nilai importance sebesar 0,0459. Meskipun demikian, fitur tersebut tetap berkontribusi dalam proses pengambilan keputusan model karena digunakan bersama dengan fitur-fitur lainnya.

4.5 Implementasi Sistem Rekomendasi Tanaman

Model yang telah dilatih kemudian diimplementasikan ke dalam sistem rekomendasi tanaman. Sistem menerima enam parameter masukan berupa nilai Nitrogen (N), Fosfor (P), Kalium (K), suhu, kelembapan, dan pH tanah. Berdasarkan parameter tersebut, sistem menghasilkan rekomendasi tanaman beserta tingkat keyakinan (confidence score).

Pengujian dilakukan pada beberapa kondisi tanah yang memiliki tingkat keasaman berbeda, mulai dari sangat asam hingga basa. Hasil pengujian menunjukkan bahwa model mampu memberikan rekomendasi tanaman yang berbeda sesuai dengan karakteristik lingkungan yang diberikan.

Kondisi Tanah	pH	Rekomendasi	Confidence
Sangat Asam	4.3	Pomegranate	53.2%
Asam	5.0	Rice	92.6%
Agak Asam	6.0	Rice	72.3%
Netral	7.0	Rice	93.5%
Agak Basa	8.0	Jute	97.1%
Basa	8.8	Chickpea	74.6%

Tabel tersebut menunjukkan bahwa sistem mampu menghasilkan rekomendasi tanaman yang berbeda sesuai dengan perubahan kondisi lingkungan, khususnya tingkat keasaman tanah. Hal ini menunjukkan bahwa model telah berhasil mempelajari hubungan antara karakteristik tanah dengan jenis tanaman yang sesuai untuk ditanam.

Contohnya, pada kondisi tanah sangat asam (pH = 4,3), sistem merekomendasikan tanaman Pomegranate dengan tingkat keyakinan sebesar 53,2%. Pada kondisi tanah netral (pH = 7,0), sistem merekomendasikan tanaman Rice dengan tingkat keyakinan sebesar 93,5%. Sedangkan pada kondisi tanah basa (pH = 8,8), sistem merekomendasikan tanaman Chickpea dengan tingkat keyakinan sebesar 74,6%.

Hasil tersebut menunjukkan bahwa sistem berhasil memanfaatkan informasi kondisi tanah dan lingkungan untuk menghasilkan rekomendasi tanaman yang sesuai dengan karakteristik lahan.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan metode Artificial Neural Network (ANN) untuk membangun sistem rekomendasi tanaman berdasarkan kondisi tanah dan lingkungan. Model menggunakan enam parameter masukan yaitu Nitrogen (N), Fosfor (P), Kalium (K), suhu (temperature), kelembapan (humidity), dan tingkat keasaman tanah (pH) untuk mengklasifikasikan 22 jenis tanaman. Melalui proses optimasi hyperparameter menggunakan Grid Search terhadap 36 kombinasi parameter, diperoleh konfigurasi terbaik dengan jumlah neuron 128, learning rate 0,01, batch size 64, dan dropout rate 0,2. Model yang dihasilkan mampu mencapai akurasi sebesar 96,59% dengan nilai precision, recall, dan F1-score yang seimbang, sehingga menunjukkan kemampuan klasifikasi yang sangat baik. Hasil evaluasi menggunakan confusion matrix memperlihatkan bahwa sebagian besar data berhasil diklasifikasikan dengan benar, sementara kesalahan klasifikasi hanya terjadi pada beberapa kelas tertentu dengan jumlah yang relatif kecil. Analisis feature importance menunjukkan bahwa fitur humidity memiliki pengaruh terbesar dalam proses pengambilan keputusan model, diikuti oleh unsur hara Fosfor (P) dan Kalium (K), sedangkan fitur pH memiliki pengaruh yang lebih rendah dibandingkan fitur lainnya. Selain itu, implementasi sistem rekomendasi tanaman menunjukkan bahwa model mampu memberikan rekomendasi tanaman yang sesuai dengan kondisi tanah yang diberikan beserta tingkat keyakinan prediksinya, sehingga sistem dapat digunakan sebagai alat bantu dalam menentukan jenis tanaman yang cocok untuk ditanam.

Berdasarkan hasil penelitian yang telah dilakukan, pengembangan lebih lanjut dapat dilakukan dengan menambahkan parameter lingkungan lainnya seperti curah hujan, jenis tanah, dan intensitas cahaya agar sistem mampu memberikan rekomendasi yang lebih akurat. Selain itu, penelitian selanjutnya dapat membandingkan performa Artificial Neural Network dengan metode machine learning lainnya untuk mengetahui metode yang paling optimal dalam permasalahan rekomendasi tanaman. Sistem yang telah dikembangkan juga dapat diimplementasikan dalam bentuk aplikasi berbasis web atau mobile sehingga lebih mudah digunakan

oleh petani maupun masyarakat umum dalam menentukan tanaman yang sesuai dengan kondisi lahan yang dimiliki.

VI. UCAPAN TERIMAKASIH (OPTIONAL)

Penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah yang telah memberikan arahan, bimbingan, serta masukan selama proses penyusunan penelitian ini. Penulis juga menyampaikan terima kasih kepada seluruh anggota kelompok yang telah berkontribusi dalam pengumpulan data, pengembangan sistem, dan penyusunan laporan penelitian. Selain itu, penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika Universitas Mataram yang telah menyediakan sarana dan lingkungan akademik yang mendukung pelaksanaan penelitian ini.

DAFTAR PUSTAKA

[1] W.K. Chen. *Linear Networks and Systems*. Belmont, CA: Wadsworth, 1993, hlm. 123-135.

[2] M.D. Dahleh. "6.5 Matrix methods," dalam *Vibration and Shock Handbook*. C.W. De Silva, Ed. Boca Raton: Taylor & Francis, 2005, hlm. 6-14.

[3] D. Casadei, G. Serra, K. Tani. "Implementation of a direct control algorithm for induction motors based on discrete space vector modulation." *IEEE Transactions on Power Electronics*, 15(4), hlm. 769-777, 2007.

[4] R. Nuryadi dan D. Hartanto. "Computer simulation of quantum confinement effect in silicon nano wire," dalam *Proc. The 12th International Conference on QiR (Quality in Research*, 2011, hlm. 160-166.

[5] E.E. Rebecca. "Alternating current fed power supply." U.S. Patent 7 897 777, 3 Nov. 1987.

[6] D.E. Winterbone. (1997). *Advanced thermodynamics for engineers.* [Online]. Tersedia di: www.knovel.com [6 Jan. 2011].

[7] A. Paul. (1987, Okt.). "Electrical properties of flying machines". *Flying Machines*. [On-line]. 38(1), hlm. 778-998. Tersedia di: www.flyingmachjourn/properties/fly.edu [1 Des 2003].

[8] N. Pakvilai. (2013). "Plant macronutrient analysis of enzyme ionic plasma and organic fertilizer from biodegradable waste." *The 2nd Annual South East Asian International Seminar (ASAIS)*. [On-line].hlm. 1-6. Tersedia di: http://asais-pnj.org [17 Jan 2014].

[9] M. Duncan. "Engineering Concepts on Ice." Internet: www.iceengg.edu/staff.html, 25 Okt. 2000 [29 Nov. 2003].

[10] T. Pangaribuan. "Perkembangan Kompetensi Kewacanaan di LPTK." Disertasi Doktor. IKIP Malang, Malang, 1992.

[11] S. Maw. Bahan Kuliah, Engg 251. Topik: “Speed skating.” ICT 224, Faculty of Engineering, University of Calgary, Calgary, Alberta, 31 Okt. 2003.

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN CAFE TERBAIK UNTUK BELAJAR MAHASISWA MENGGUNAKAN METODE WEIGHTED PRODUCT

Muhammad Alfath Mavianza [1)], I Nyoman Sudarsana [1)], Muhammad Raechan Ulwan Zacky [1)], Ida Bagus Amanta Pradipa Khrisna [1)], Santi Ika Murpratiwi S.Kom., M.T [2)]
1) Mahasiswa Program Studi Teknik Informatika Universitas Mataram
2) Dosen Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: amav1305@gmail.com

Abstrak

Cafe telah menjadi salah satu tempat favorit mahasiswa untuk belajar di luar kampus. Namun, banyaknya pilihan cafe yang tersedia membuat mahasiswa kesulitan dalam menentukan cafe yang paling sesuai dengan kebutuhan belajar mereka. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah sistem pendukung keputusan (SPK) berbasis web yang dapat membantu mahasiswa memilih cafe terbaik untuk belajar berdasarkan kriteria-kriteria yang relevan. Metode yang digunakan adalah Weighted Product (WP), yaitu metode pengambilan keputusan multi-kriteria yang melakukan perkalian terbobot antar kriteria untuk menghasilkan nilai preferensi setiap alternatif. Data kriteria dan bobot diperoleh dari kuesioner yang diisi oleh 71 responden mahasiswa. Lima kriteria yang digunakan adalah harga menu (cost), kecepatan WiFi (benefit), kelayakan fasilitas belajar (benefit), jarak dari kampus (cost), dan ketersediaan colokan listrik (benefit), dengan lima alternatif cafe yaitu Kopi Kenangan, Éclair, Bento Kopi, Upnormal, dan Locus Coffee. Sistem dikembangkan menggunakan Next.js dan Tailwind CSS dalam bentuk kalkulator keputusan interaktif yang menuntun pengguna melalui enam tahapan perhitungan WP secara bertahap. Hasil pengujian sistem menunjukkan bahwa Upnormal merupakan cafe terbaik dengan nilai vektor V tertinggi sebesar 0,2698, diikuti oleh Bento Kopi (0,2579), Kopi Kenangan (0,1766), Éclair (0,1639), dan Locus Coffee (0,1319). Sistem ini diharapkan dapat membantu mahasiswa dalam pengambilan keputusan secara lebih objektif, terstruktur, dan mudah diakses melalui browser.
Kata kunci: Aplikasi Web, Cafe, Mahasiswa, Multi-Kriteria, Sistem Pendukung Keputusan, Weighted Product

Abstract

Cafes have become one of the most popular study spots for university students outside of campus. However, the wide variety of available options makes it difficult for students to determine the most suitable cafe for their learning needs. This study aims to design and implement a web-based Decision Support System (DSS) to assist students in selecting the best cafe for studying based on relevant criteria. The method employed is the Weighted Product (WP) method, a multi-criteria decision-making approach that performs weighted multiplication across criteria to produce a preference value for each alternative. Criteria weights were derived from a questionnaire completed by 71 student respondents. Five criteria were used: menu price (cost), WiFi speed (benefit), study facility adequacy (benefit), distance from campus (cost), and availability of power outlets (benefit), with five cafe alternatives: Kopi Kenangan, Éclair, Bento Kopi, Upnormal, and Locus Coffee. The system was developed using Next.js and Tailwind CSS as an interactive decision calculator that guides users through six sequential WP computation steps. Testing results show that Upnormal is the best cafe with the highest V vector value of 0.2698, followed by Bento Kopi (0.2579), Kopi Kenangan (0.1766), Éclair (0.1639), and Locus Coffee (0.1319). This system is expected to assist students in making decisions that are more objective, structured, and easily accessible through a browser.
Keyword: *Cafe, Decision Support System, Multi-Criteria, Students, Web Application, Weighted Product*

I. PENDAHULUAN

Perkembangan budaya belajar di kalangan mahasiswa saat ini tidak lagi terbatas pada ruang kelas atau perpustakaan. Cafe telah menjadi salah satu alternatif tempat belajar yang diminati karena menawarkan suasana yang lebih santai, fleksibel, dan kondusif bagi aktivitas akademik [1]. Fasilitas seperti Wi-Fi, colokan listrik, serta suasana yang nyaman menjadikan cafe pilihan menarik bagi mahasiswa untuk mengerjakan tugas, diskusi kelompok, maupun belajar mandiri.

Namun, banyaknya cafe yang tersedia di sekitar kampus menciptakan dilema tersendiri. Tidak semua cafe mampu memenuhi seluruh kriteria kebutuhan belajar mahasiswa secara seimbang. Beberapa cafe menawarkan harga menu yang terjangkau namun memiliki koneksi WiFi yang lambat, sementara cafe lain menyediakan

fasilitas belajar yang lengkap namun berlokasi jauh dari kampus.

Kondisi ini menyebabkan mahasiswa kesulitan membuat keputusan yang tepat dalam memilih cafe yang paling sesuai. Pemilihan yang dilakukan secara subjektif berdasarkan kebiasaan atau rekomendasi teman saja berisiko menghasilkan keputusan yang kurang sesuai dengan kebutuhan belajar yang sebenarnya [2].

Sistem Pendukung Keputusan (SPK) merupakan solusi berbasis teknologi informasi yang dapat membantu proses pengambilan keputusan secara lebih sistematis, objektif, dan terukur. Dengan memanfaatkan metode pengambilan keputusan multi-kriteria, SPK mampu mempertimbangkan berbagai faktor secara simultan dan menghasilkan rekomendasi berdasarkan data yang valid. Agar dapat dimanfaatkan secara praktis oleh mahasiswa, SPK dalam penelitian ini dirancang dalam bentuk aplikasi web yang dapat diakses langsung melalui browser tanpa memerlukan instalasi perangkat lunak tambahan, sehingga proses input data dan perhitungan dapat dilakukan secara interaktif oleh pengguna.

Metode Weighted Product (WP) dipilih dalam penelitian ini karena kesederhanaannya serta kemampuannya dalam menangani kriteria bertipe cost dan benefit secara bersamaan. Metode WP bekerja dengan mengalikan nilai setiap alternatif yang telah dipangkatkan dengan bobot masing-masing kriteria, sehingga menghasilkan nilai preferensi yang komprehensif.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan SPK berbasis web untuk pemilihan cafe terbaik bagi belajar mahasiswa Universitas Mataram menggunakan metode WP. Data kriteria dan bobot preferensi diperoleh melalui kuesioner yang disebarkan kepada 71 mahasiswa sebagai responden. Lima cafe di sekitar kampus Universitas Mataram menjadi objek penelitian, yaitu Kopi Kenangan, Éclair, Bento Kopi, Upnormal, dan Locus Coffee. Sistem dikembangkan sebagai kalkulator keputusan interaktif yang menuntun pengguna melalui tahapan input kriteria, input data alternatif, normalisasi bobot, perhitungan vektor S, perhitungan vektor V, hingga penyajian hasil akhir berupa ranking dan rekomendasi cafe terbaik.

II. TINJAUAN PUSTAKA

Sistem Pendukung Keputusan (SPK) merupakan sistem informasi berbasis komputer yang dirancang untuk membantu pengambilan keputusan dalam situasi yang kompleks, tidak terstruktur, atau semi-terstruktur. SPK tidak dimaksudkan untuk menggantikan keputusan manusia, melainkan menyediakan data, model, dan antarmuka dialog yang mendukung proses berpikir pengambil keputusan [3]. Secara umum, arsitektur SPK terdiri atas tiga komponen utama, yaitu manajemen data untuk mengelola dan menyaring informasi yang relevan, manajemen model untuk menerapkan metode analitik pengambilan keputusan, dan antarmuka pengguna yang menyajikan hasil dalam format yang mudah dipahami dan ditindaklanjuti [3]. Ketiga komponen tersebut menjadi acuan dalam perancangan SPK pemilihan cafe belajar mahasiswa pada penelitian ini: manajemen data berupa data kriteria dan alternatif cafe, manajemen model berupa metode Weighted Product, dan antarmuka pengguna berupa aplikasi web yang dikembangkan. Aplikasi yang dibangun juga menyertakan halaman dokumentasi teori SPK dan metode WP bagi pengguna yang ingin memahami dasar perhitungan secara lebih mendalam.

Multi-Criteria Decision Making (MCDM) merupakan pendekatan pengambilan keputusan yang digunakan untuk memilih alternatif terbaik berdasarkan beberapa kriteria yang memiliki tingkat kepentingan berbeda [4]. Dalam penelitian ini, pemilihan cafe belajar mahasiswa termasuk dalam permasalahan multikriteria karena setiap cafe memiliki nilai yang berbeda pada setiap kriteria; misalnya, suatu cafe dapat unggul dari sisi kecepatan WiFi tetapi kurang baik dari sisi harga atau jarak dari kampus. Penentuan bobot kriteria menjadi bagian penting dalam MCDM karena bobot menunjukkan tingkat kepentingan relatif antar kriteria [5].

Weighted Product (WP) merupakan salah satu metode MCDM yang digunakan untuk menentukan alternatif terbaik berdasarkan nilai preferensi. Metode ini menggunakan perkalian nilai kriteria yang dipangkatkan dengan bobot masing-masing kriteria, dengan kriteria benefit menggunakan bobot bernilai positif dan kriteria cost menggunakan bobot bernilai negatif. Tahapan

umum metode WP dimulai dari menentukan alternatif dan kriteria, menentukan bobot setiap kriteria, melakukan normalisasi bobot, menghitung nilai vektor S, menghitung nilai preferensi V, kemudian menentukan ranking alternatif berdasarkan nilai preferensi terbesar. Nilai vektor S dihitung dengan Persamaan (1).

$$S_i = \prod_{j=1}^{n} x_{ij}^{w_j} \quad (1)$$

Keterangan: Si merupakan nilai vektor S untuk alternatif ke-i, xij merupakan nilai alternatif ke-i pada kriteria ke-j, dan wj merupakan bobot kriteria ke-j. Setelah nilai vektor S diperoleh, nilai preferensi V dihitung dengan Persamaan (2).

$$V_i = \frac{S_i}{\sum_{i=1}^{m} S_i} \quad (2)$$

Nilai Vi digunakan sebagai dasar dalam proses perangkingan alternatif. Alternatif dengan nilai Vi terbesar menjadi alternatif yang paling direkomendasikan.

Beberapa penelitian sebelumnya telah menerapkan metode WP dalam SPK pemilihan cafe maupun rekomendasi tempat kuliner. Apsiswanto dan Pamungkas menerapkan WP untuk pemilihan cafe bagi mahasiswa pendatang di Kota Metro [6]. Kaslumin dan Purnomo menggunakan WP dalam pemilihan cafe bagi mahasiswa di Yogyakarta [7]. Ramadhan, Priskila, dan Parhusip menerapkan WP dalam SPK pemilihan cafe di Kota Palangka Raya berbasis website [8], yaitu pendekatan yang sejalan dengan penelitian ini karena turut menghadirkan SPK WP dalam bentuk aplikasi web yang dapat diakses pengguna secara langsung. Putri dkk. menggunakan WP dalam pemilihan cafe favorit di Kota Bengkayang [9], sedangkan Wardhani dan Lutfina [10] serta Khasanah [11] menerapkan WP pada rekomendasi tempat kuliner berbasis aplikasi mobile dan web.

Berdasarkan penelitian terdahulu tersebut, metode WP terbukti dapat diimplementasikan dalam bentuk sistem berbasis web untuk membantu pengambilan keputusan pemilihan tempat. Perbedaan penelitian ini terletak pada fokus objek penelitian, yaitu pemilihan cafe khusus untuk kebutuhan belajar mahasiswa dengan kriteria yang disesuaikan secara spesifik (harga menu, kecepatan WiFi, kelayakan fasilitas belajar, jarak dari kampus, dan ketersediaan colokan listrik), serta implementasi sistem dalam bentuk kalkulator keputusan interaktif bertahap (step-by-step) yang menuntun pengguna mulai dari input kriteria hingga penyajian hasil akhir, alih-alih hanya menampilkan hasil akhir secara statis.

III. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan model pengambilan keputusan multikriteria, yang diimplementasikan dalam bentuk SPK berbasis web. Pengembangan sistem dilakukan menggunakan model Waterfall, yang terdiri atas tahapan analisis kebutuhan, desain sistem, implementasi (coding), pengujian, dan pemeliharaan secara berurutan [12]. Tahapan analisis kebutuhan dilakukan dengan mengidentifikasi kriteria dan alternatif cafe yang relevan bagi kebutuhan belajar mahasiswa; tahapan desain meliputi perancangan alur perhitungan WP dan antarmuka pengguna; tahapan implementasi dilakukan dengan membangun aplikasi web menggunakan framework Next.js dan Tailwind CSS pada sisi front-end serta Node.js pada sisi server; dan tahapan pengujian dilakukan dengan memverifikasi kesesuaian hasil perhitungan sistem terhadap perhitungan manual.

Sistem SPK yang dibangun berupa kalkulator keputusan interaktif yang terdiri atas enam tahapan berurutan, yaitu: (1) penyajian kriteria penilaian beserta tipe dan bobot awal; (2) input data alternatif cafe pada setiap kriteria; (3) normalisasi bobot kriteria; (4) perhitungan nilai vektor S; (5) perhitungan nilai preferensi V dan ranking sementara; dan (6) penyajian hasil akhir berupa rekomendasi cafe terbaik. Pengguna dapat berpindah antar tahapan menggunakan tombol navigasi pada setiap halaman, sementara seluruh perhitungan dilakukan secara otomatis oleh sistem berdasarkan data yang diinput. Sistem ini bersifat publik tanpa mekanisme autentikasi, karena difokuskan sebagai alat bantu kalkulasi yang dapat langsung digunakan oleh siapa pun tanpa proses pendaftaran.

Data yang digunakan terdiri dari data observasi cafe dan data bobot kriteria. Data

observasi digunakan untuk memperoleh nilai setiap alternatif pada masing-masing kriteria, sedangkan data bobot diperoleh dari penilaian 71 responden terhadap tingkat kepentingan kriteria. Setiap responden memberikan urutan prioritas terhadap kriteria yang dianggap penting dalam memilih cafe belajar.

Tabel 1. Alternatif cafe

Kode	Alternatif
A1	Kopi Kenangan
A2	Éclair
A3	Bento Kopi
A4	Upnormal
A5	Locus Coffee

Kriteria yang digunakan dalam penelitian ini terdiri dari lima kriteria, yaitu harga menu, kecepatan WiFi, kelayakan fasilitas belajar, jarak dari kampus, dan ketersediaan colokan listrik. Harga menu dan jarak dari kampus merupakan kriteria cost karena semakin kecil nilainya semakin baik. Kecepatan WiFi, kelayakan fasilitas belajar, dan ketersediaan colokan listrik merupakan kriteria benefit karena semakin besar nilainya semakin baik.

Tabel 2. Kriteria penelitian

Kode	Kriteria	Jenis
C1	Harga menu	Cost
C2	Kecepatan WiFi	Benefit
C3	Kelayakan fasilitas belajar	Benefit
C4	Jarak dari kampus	Cost
C5	Ketersediaan colokan listrik	Benefit

Nilai harga menu diperoleh dari kombinasi rata-rata harga makanan dan rata-rata harga minuman, dengan bobot minuman dibuat lebih besar karena mahasiswa yang belajar di cafe cenderung menjadikan minuman sebagai pembelian utama. Nilai harga menu dihitung menggunakan Persamaan (3).

Harga Menu = (0,4 × Rata-rata Harga Makanan) + (0,6 × Rata-rata Harga Minuman) (3)

Kelayakan fasilitas belajar diperoleh dari kombinasi jumlah kursi dan skor ketenangan lingkungan (skala 1-5), dengan bobot ketenangan dibuat lebih besar karena suasana yang tenang lebih mendukung aktivitas belajar dibanding jumlah kursi semata. Nilai kelayakan fasilitas belajar dihitung menggunakan Persamaan (4).

Kelayakan Fasilitas Belajar = (0,4 × Jumlah Kursi) + (0,6 × Skor Ketenangan) (4)

Ketersediaan colokan listrik merepresentasikan kemudahan mahasiswa dalam mengakses sumber daya listrik, dengan mempertimbangkan jumlah fisik colokan, sebaran colokan, jumlah meja yang memiliki akses listrik, serta kemudahan pengguna menjangkau colokan dari area tempat duduk.

Tabel 3. Data alternatif berdasarkan kriteria

Kode	C1	C2	C3	C4	C5
A1	18.243	10	51,4	1,5	30
A2	28.996	100	30,4	1,4	5
A3	17.168,4	20	72,6	2,5	75
A4	22.300	25	57,8	3,1	150
A5	24.560	10	39	0,9	10

Bobot kriteria diperoleh dari hasil penilaian responden dengan sistem ranking, dengan ranking 1 diberi 5 poin hingga ranking 5 diberi 1 poin. Total poin setiap kriteria kemudian dinormalisasi agar jumlah seluruh bobot bernilai 1.

Tabel 4. Bobot kriteria

Kode	Poin	Bobot
C1	216	0,202817
C2	233	0,218779
C3	288	0,270423
C4	120	0,112676
C5	208	0,195305

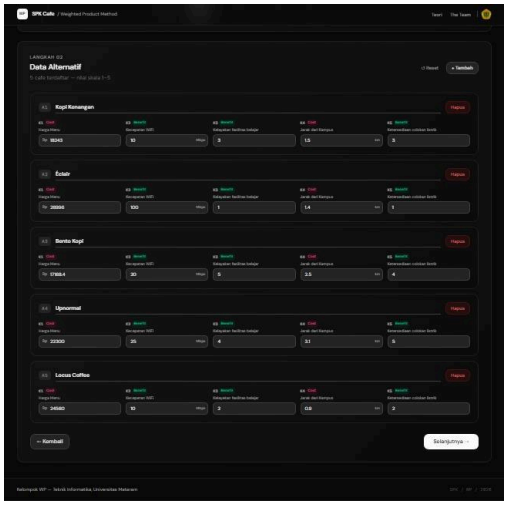
Normalisasi bobot dilakukan menggunakan Persamaan (5).

$$W_j = \frac{w_j}{\sum_{j=1}^{n} w_j} \quad (5)$$

Setelah bobot dinormalisasi, nilai vektor S dan nilai preferensi V masing-masing dihitung menggunakan Persamaan (1) dan (2), dengan kriteria benefit menggunakan bobot positif dan kriteria cost menggunakan bobot negatif. Nilai Vi yang dihasilkan oleh sistem digunakan untuk menentukan ranking alternatif cafe belajar mahasiswa.

IV. HASIL DAN PEMBAHASAN

Berdasarkan data alternatif, jenis kriteria, dan bobot normalisasi pada Tabel 1 hingga Tabel 4, perhitungan nilai S, V, dan ranking dilakukan langsung melalui aplikasi web yang telah dibangun. Kriteria harga menu dan jarak dari kampus diberi eksponen bernilai negatif karena termasuk kriteria cost, sedangkan kecepatan WiFi, kelayakan fasilitas belajar, dan ketersediaan colokan listrik diberi eksponen bernilai positif karena termasuk kriteria benefit. Gambar 1 menunjukkan antarmuka input data alternatif pada sistem, yang memuat data yang sama dengan Tabel 3.



Gambar 1. Antarmuka input data alternatif pada SPK Cafe

Tabel 5 menyajikan hasil akhir perhitungan nilai S, V, dan ranking yang dihasilkan oleh sistem.

Tabel 5. Hasil perhitungan nilai S, V, dan ranking

Kode	Alt	S	V	Rank
A1	Kopken	1,21867	0,17656	3
A2	Éclair	1,13128	0,16390	4
A3	Bento	1,77981	0,25786	2
A4	Upno	1,86222	0,26980	1
A5	Locus	0,91009	0,13185	5

Hasil perhitungan pada sistem menunjukkan bahwa Upnormal memperoleh nilai preferensi tertinggi sebesar 0,269806 sehingga menempati ranking pertama. Nilai tersebut dipengaruhi oleh ketersediaan colokan listrik yang sangat tinggi dan nilai kelayakan fasilitas belajar yang cukup baik. Meskipun jarak Upnormal dari kampus lebih jauh dibanding beberapa alternatif lain, pengaruh kriteria benefit seperti ketersediaan colokan dan fasilitas belajar membuat nilai akhirnya tetap lebih unggul dibanding alternatif lain.

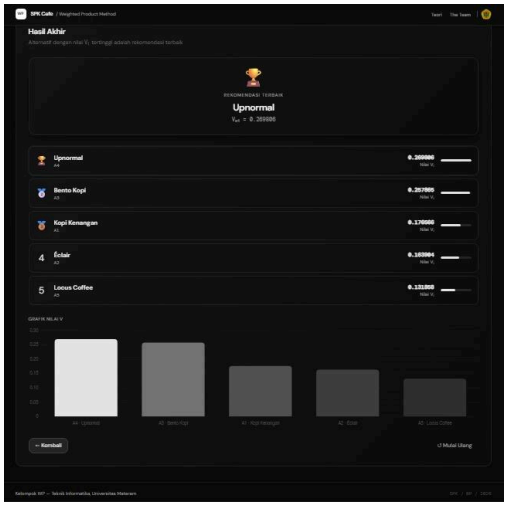
Bento Kopi menempati ranking kedua dengan nilai preferensi sebesar 0,257865. Alternatif ini memiliki harga menu yang relatif rendah, nilai kelayakan fasilitas belajar tertinggi, dan ketersediaan colokan listrik yang cukup besar. Namun, nilai WiFi yang lebih rendah dibanding Éclair dan jarak yang lebih jauh dari kampus membuat nilai preferensinya masih berada di bawah Upnormal.

Kopi Kenangan menempati ranking ketiga dengan nilai preferensi sebesar 0,176566. Alternatif ini memiliki harga menu yang lebih rendah dibanding Éclair dan Upnormal serta jarak yang masih cukup dekat dari kampus, namun nilai WiFi dan ketersediaan colokan listrik tidak sebesar alternatif dengan ranking lebih tinggi.

Éclair menempati ranking keempat dengan nilai preferensi sebesar 0,163904. Keunggulan utama Éclair terdapat pada kecepatan WiFi yang paling tinggi, namun harga menu yang relatif tinggi, ketersediaan colokan listrik yang rendah, dan nilai kelayakan fasilitas belajar yang lebih kecil menyebabkan nilai preferensinya tidak menjadi yang tertinggi. Hal ini menunjukkan bahwa satu kriteria yang unggul tidak selalu cukup untuk menghasilkan ranking tertinggi apabila kriteria lain memiliki nilai rendah.

Locus Coffee menempati ranking kelima dengan nilai preferensi sebesar 0,131858. Alternatif ini memiliki keunggulan dari sisi jarak yang paling dekat dengan kampus, tetapi nilai WiFi, ketersediaan colokan listrik, dan kelayakan fasilitas belajar masih lebih rendah dibanding beberapa alternatif lainnya, sehingga nilai preferensi akhirnya menjadi yang paling rendah.

Gambar 2 menunjukkan antarmuka hasil akhir pada sistem, yang menyajikan rekomendasi cafe terbaik beserta nilai preferensi V dari seluruh alternatif dalam bentuk daftar ranking dan grafik batang, sehingga pengguna dapat secara langsung membandingkan nilai antar alternatif tanpa perlu melakukan perhitungan manual.



Gambar 2. Antarmuka hasil akhir rekomendasi pada SPK Cafe

Secara umum, implementasi metode Weighted Product pada aplikasi web ini berhasil mereplikasi hasil perhitungan manual secara konsisten, dengan kombinasi nilai pada seluruh criteria bukan satu kriteria tunggal yang menentukan ranking akhir. Kriteria dengan bobot tertinggi adalah kelayakan fasilitas belajar sebesar 0,270423, sehingga faktor yang mendukung aktivitas belajar memiliki pengaruh besar terhadap hasil akhir rekomendasi yang ditampilkan sistem.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan Sistem Pendukung Keputusan (SPK) berbasis web untuk pemilihan cafe belajar mahasiswa menggunakan metode Weighted Product. Sistem dikembangkan menggunakan Next.js dan Tailwind CSS dalam bentuk kalkulator keputusan interaktif yang menuntun pengguna melalui tahapan input kriteria, input data alternatif, normalisasi bobot, perhitungan vektor S, perhitungan vektor V, hingga penyajian hasil akhir. Alternatif yang digunakan terdiri dari Kopi Kenangan, Éclair, Bento Kopi, Upnormal, dan Locus Coffee, dengan kriteria harga menu, kecepatan WiFi, kelayakan fasilitas belajar, jarak dari kampus, dan ketersediaan colokan listrik. Hasil pengujian sistem menunjukkan bahwa Upnormal menjadi alternatif terbaik dengan nilai preferensi 0,269806, diikuti oleh Bento Kopi, Kopi Kenangan, Éclair, dan Locus Coffee. Hasil ini menunjukkan bahwa SPK berbasis web mampu memberikan rekomendasi yang objektif sekaligus mudah diakses oleh mahasiswa tanpa memerlukan instalasi atau proses pendaftaran. Untuk penelitian selanjutnya, sistem dapat dikembangkan dengan menambahkan fitur input data alternatif baru secara dinamis oleh pengguna, penyimpanan riwayat perhitungan, serta perluasan jumlah alternatif cafe agar hasil rekomendasi semakin representatif terhadap kondisi terkini.

VI. UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika Universitas Mataram serta dosen pembimbing yang telah memberikan arahan dalam penelitian ini.

DAFTAR PUSTAKA

[1] S. S. K. Adityawirawan and H. E. Kusuma, “Café as student’s informal learning space: A case study in Bandung, Indonesia,” DIMENSI: Journal of Architecture and Built Environment, vol. 48, no. 2, pp. 109–120, 2021.

[2] E. B. Mandinach, “A perfect time for data use: Using data-driven decision making to inform practice,” Educational Psychologist, vol. 47, no. 2, pp. 71–85, 2012.

[3] R. H. Sprague Jr., “A framework for the development of decision support systems,” MIS Quarterly, vol. 4, no. 4, pp. 1–26, 1980.

[4] H. Taherdoost and M. Madanchian, “Multi-Criteria Decision Making (MCDM) methods and concepts,” Encyclopedia, vol. 3, no. 1, pp. 77–87, 2023.

[5] M. Žižović and D. Pamučar, “New model for determining criteria weights: Level Based Weight Assessment (LBWA) model,” Decision Making: Applications in Management and Engineering, vol. 2, no. 2, pp. 126–137, 2019.

[6] U. Apsiswanto and C. A. Pamungkas, “Penerapan metode Weight Product (WP) pada pemilihan kafe bagi mahasiswa pendatang di Kota Metro,” Jurnal Informatika, vol. 22, no. 2, pp. 172–182, 2022.

[7] W. S. Kaslumin and A. S. Purnomo, “Sistem penunjang keputusan pemilihan café bagi mahasiswa Yogyakarta menggunakan metode Weighted Product (WP),” Jurnal Syntax Admiration, vol. 4, no. 12, 2023.

[8] A. F. Ramadhan, R. Priskila, and J. Parhusip, “Sistem pendukung keputusan pemilihan cafe di

Kota Palangka Raya berbasis website dengan menggunakan metode Weighted Product," Prosiding KONSTELASI, vol. 2, no. 1, pp. 159–171, 2025.

[9] M. M. A. Putri, Veronika, J. T. Sumar, Y. M. Charolin, and N. P., "Sistem pendukung keputusan pemilihan cafe favorit di Kota Bengkayang menggunakan metode Weighted Product (WP)," Instink: Inovasi Pendidikan, Teknologi Informasi dan Komputer, vol. 5, no. 1, pp. 34–46, 2026.

[10] A. K. Wardhani and E. Lutfina, "Application culinary decision support system in Kudus City with Weighted Product method based on mobile phone," Journal of Computer Science and Engineering, vol. 1, no. 1, pp. 10–16, 2020.

[11] F. N. Khasanah, "Rekomendasi hasil metode Weighted Product terhadap pemilihan tempat kuliner di sekitar Universitas Bhayangkara Bekasi," Techno.Com, vol. 20, no. 3, pp. 382–391, 2021.

[12] W. W. Royce, "Managing the development of large software systems," in Proc. IEEE WESCON, vol. 26, 1970, pp. 1–9.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

ANALISIS PERFORMA ALGORITMA PARALLEL MERGE SORT MENGGUNAKAN MPI TERHADAP WAKTU EKSEKUSI PENGURUTAN DATASET IPK MAHASISWA

Ni Wayan Eka Aprilianti[1)], Musfiqoh Rizkia Aulia[2)], Qomari Auliyah[3)], Annisa Makarima Akhlak[4)]
1,2,3,4) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: ekalia@gmail.com

Abstrak

Pertumbuhan data yang semakin besar mendorong kebutuhan terhadap metode pengolahan data yang efisien, terutama dalam proses pengurutan data berskala besar. Pendekatan sekuensial memiliki keterbatasan karena waktu eksekusi yang cenderung meningkat seiring bertambahnya jumlah data. Penelitian ini bertujuan untuk menganalisis performa algoritma Parallel Merge Sort menggunakan Message Passing Interface (MPI) terhadap waktu eksekusi pengurutan dataset IPK mahasiswa. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan pustaka mpi4py. Data dibagi ke beberapa proses menggunakan metode scatter, kemudian setiap proses melakukan pengurutan secara paralel menggunakan Merge Sort, dan hasil akhirnya digabungkan kembali melalui metode gather. Pengujian dilakukan dengan variasi jumlah proses MPI mulai dari 1 hingga 64 proses dan ukuran dataset mulai dari 10.000 hingga 1.000.000 data. Hasil pengujian menunjukkan bahwa pendekatan paralel mampu meningkatkan efisiensi waktu eksekusi dibandingkan metode serial. Pada dataset 1.000.000 data, waktu eksekusi menurun dari 14,0442 detik pada 1 proses menjadi 6,0331 detik pada 32 proses. Namun, peningkatan jumlah proses tidak selalu menghasilkan performa yang lebih baik karena adanya overhead komunikasi antarproses pada tahap penggabungan data. Penelitian ini menunjukkan bahwa terdapat jumlah proses optimal untuk memperoleh performa terbaik dalam implementasi Parallel Merge Sort menggunakan MPI.
Kata kunci: Merge Sort; MPI; Pemrosesan Paralel; Sorting; Waktu Eksekusi

Abstract

The increasing growth of data has driven the need for efficient data processing methods, particularly in large-scale sorting operations. Sequential approaches have limitations because execution time tends to increase as the amount of data grows. This study aims to analyze the performance of the Parallel Merge Sort algorithm using the Message Passing Interface (MPI) in terms of the execution time required to sort a student GPA dataset. The implementation was carried out using the Python programming language and the mpi4py library. The dataset was distributed among multiple processes using the scatter method, after which each process performed sorting in parallel using the Merge Sort algorithm. The sorted results were then combined using the gather method. Experiments were conducted using varying numbers of MPI processes, ranging from 1 to 64 processes, and dataset sizes ranging from 10,000 to 1,000,000 records. The results indicate that the parallel approach improves execution time efficiency compared to the serial method. For a dataset containing 1,000,000 records, the execution time decreased from 14.0442 seconds with 1 process to 6.0331 seconds with 32 processes. However, increasing the number of processes does not always lead to better performance due to the communication overhead incurred during the data merging stage. This study demonstrates that there is an optimal number of processes for achieving the best performance in the implementation of Parallel Merge Sort using MPI.
Keyword: *Execution Time; Merge Sort; MPI; Parallel Processing; Sorting.*

I. PENDAHULUAN

Perkembangan teknologi informasi menyebabkan pertumbuhan data yang sangat cepat di berbagai bidang, termasuk dalam dunia pendidikan. Volume data yang terus meningkat menuntut sistem pengolahan data yang mampu bekerja secara cepat dan efisien, terutama pada proses pengolahan data berukuran besar. Salah satu proses penting dalam pengolahan data adalah pengurutan data (sorting), karena data yang terurut dapat mempermudah proses pencarian, analisis, dan pengolahan informasi lebih lanjut. Namun, pada dataset berukuran besar, proses pengurutan membutuhkan waktu komputasi yang cukup tinggi apabila dilakukan menggunakan metode konvensional [1].

Dalam lingkungan akademik, data mahasiswa seperti data Indeks Prestasi Kumulatif (IPK) sering digunakan untuk kebutuhan evaluasi, pengelompokan, maupun analisis akademik. Pengolahan dataset IPK mahasiswa dalam jumlah besar memerlukan metode pengurutan yang efisien agar proses komputasi dapat dilakukan dengan lebih cepat dan terstruktur. Salah satu algoritma pengurutan yang memiliki performa baik untuk pengolahan data berskala besar adalah Merge Sort. Algoritma ini menggunakan pendekatan *divide* and *conquer* dengan cara membagi data menjadi beberapa bagian kecil, melakukan proses pengurutan pada masing-masing bagian, kemudian menggabungkannya kembali menjadi data yang terurut. Merge Sort memiliki kompleksitas waktu sebesar O(n log n) sehingga dinilai cukup stabil dan efisien [2].

Meskipun Merge Sort memiliki performa yang baik, implementasi secara sekuensial masih memiliki keterbatasan dalam hal waktu eksekusi. Pada metode serial, seluruh proses pengolahan data dijalankan menggunakan satu prosesor sehingga waktu komputasi akan meningkat seiring bertambahnya jumlah data yang diproses. Kondisi tersebut menyebabkan proses pengurutan data berskala besar menjadi kurang optimal, terutama ketika ukuran dataset mencapai ratusan ribu hingga jutaan data [3]. Selain itu, peningkatan jumlah data juga dapat menyebabkan bottleneck pada proses komputasi karena seluruh instruksi dijalankan secara berurutan.

Untuk mengatasi permasalahan tersebut, digunakan pendekatan pemrosesan paralel menggunakan Message Passing Interface (MPI). MPI merupakan teknologi komputasi paralel yang memungkinkan pembagian beban kerja ke beberapa proses sehingga proses pengolahan data dapat dilakukan secara simultan. Pada implementasi Parallel Merge Sort menggunakan MPI, data dibagi ke beberapa proses menggunakan metode scatter, kemudian setiap proses melakukan pengurutan data secara paralel menggunakan Merge Sort, dan hasil akhirnya digabungkan kembali melalui metode gather [4]. Pendekatan ini diharapkan mampu meningkatkan efisiensi waktu eksekusi dibandingkan metode serial, terutama pada pengolahan dataset berukuran besar.

Berdasarkan uraian tersebut, permasalahan yang diangkat dalam penelitian ini adalah masih tingginya waktu eksekusi pada proses pengurutan dataset IPK mahasiswa berukuran besar apabila dilakukan secara sekuensial. Oleh karena itu, diperlukan suatu pendekatan yang mampu meningkatkan efisiensi komputasi sehingga proses pengurutan dapat dilakukan lebih cepat. Penelitian ini bertujuan untuk menganalisis waktu eksekusi algoritma Parallel Merge Sort menggunakan MPI pada pengolahan dataset IPK mahasiswa dengan variasi jumlah proses MPI. Pengujian dilakukan untuk mengetahui pengaruh jumlah proses terhadap waktu eksekusi serta menentukan konfigurasi proses yang memberikan performa terbaik. Hasil penelitian diharapkan dapat menunjukkan efektivitas penggunaan MPI dalam meningkatkan efisiensi pengolahan data berskala besar.

II. TINJAUAN PUSTAKA

2.1 Algoritma *Merge Sort*

Merge Sort merupakan algoritma pengurutan berbasis pendekatan *divide and conquer* yang bekerja dengan membagi data menjadi beberapa bagian kecil, mengurutkan setiap bagian, kemudian menggabungkannya kembali hingga membentuk data yang terurut. Algoritma ini memiliki kompleksitas waktu O(n log n), sehingga dinilai stabil untuk

pengolahan data berukuran besar [2]. Lestari et al. [5] menunjukkan bahwa *Merge Sort* lebih efisien dan konsisten dibandingkan *Bubble Sort* dan *Insertion Sort* pada pengurutan data nilai akademik mahasiswa. Asra et al. [6] juga menunjukkan bahwa *Merge Sort* memiliki kestabilan performa pada berbagai kondisi data, meskipun *Quick Sort* dapat lebih unggul pada kondisi tertentu. Berdasarkan temuan tersebut, *Merge Sort* sesuai digunakan pada data akademik, tetapi implementasi sekuensialnya tetap memiliki keterbatasan ketika ukuran data semakin besar.

2.2 *Parallel Processing* dan *Parallel Merge Sort*

Parallel processing merupakan teknik komputasi yang membagi pekerjaan ke beberapa proses agar dapat dieksekusi secara bersamaan. Rahmatullah et al. [7] menunjukkan bahwa komputasi paralel dapat memberikan peningkatan performa dibandingkan komputasi sekuensial, meskipun hasilnya dipengaruhi oleh ukuran data, jenis kasus, dan sumber daya komputasi. Pada *Parallel Merge Sort*, data dibagi ke beberapa proses, setiap proses melakukan pengurutan lokal, kemudian hasilnya digabungkan kembali melalui proses *merge*. Yang [3] menunjukkan bahwa paralelisasi *Merge Sort* dalam Python dapat meningkatkan performa dibandingkan *Merge Sort* sekuensial. Panggabean et al. [1] juga membuktikan bahwa *Parallel Merge Sort* berbasis MPI mampu menurunkan waktu eksekusi pada *big data dataset*. Namun, peningkatan jumlah proses tidak selalu menghasilkan percepatan linear karena terdapat biaya komunikasi antarproses.

2.3 *Message Passing Interface* (MPI)

Message Passing Interface (MPI) merup *Message Passing Interface* (MPI) merupakan standar komunikasi dalam pemrograman paralel yang memungkinkan beberapa proses saling bertukar data melalui mekanisme pengiriman dan penerimaan pesan. MPI banyak digunakan pada sistem *distributed memory* karena setiap proses memiliki ruang memori sendiri sehingga komunikasi dilakukan secara eksplisit [8]. Dalam *Parallel Merge Sort*, MPI digunakan untuk membagi data melalui *scatter*, menjalankan pengurutan lokal pada setiap proses, kemudian mengumpulkan kembali hasil pengurutan melalui *gather*. Pendekatan ini berbeda dari metode sekuensial karena beban komputasi dapat dibagi ke beberapa proses, tetapi efektivitasnya tetap bergantung pada keseimbangan antara beban komputasi dan biaya komunikasi [1].

2.4 Metrik Performa Komputasi Paralel

Evaluasi performa algoritma paralel umumnya dilakukan menggunakan waktu eksekusi, *speedup*, dan efisiensi [4]. Waktu eksekusi menunjukkan durasi yang dibutuhkan program untuk menyelesaikan proses komputasi. *Speedup* digunakan untuk mengukur peningkatan kecepatan program paralel dibandingkan program sekuensial, dengan rumus:

$$S = T\_serial / T\_parallel$$

Efisiensi menunjukkan tingkat pemanfaatan proses yang digunakan, dengan rumus:

$$E = S / P$$

Kartawidjaja [9] menjelaskan bahwa performa komputasi paralel tidak hanya dipengaruhi oleh algoritma, tetapi juga oleh jumlah prosesor, pembagian beban kerja, komunikasi antarprosesor, dan *overhead*. Oleh karena itu, penggunaan proses yang lebih banyak belum tentu selalu menghasilkan performa yang lebih baik.

2.5 *Overhead* Komunikasi pada MPI

Dalam konteks MPI, komunikasi antarproses merupakan mekanisme penting yang memungkinkan pertukaran data selama proses komputasi paralel. Namun, komunikasi tersebut menimbulkan *overhead* berupa biaya tambahan akibat koordinasi, sinkronisasi, pembagian data, pengiriman pesan, dan penggabungan hasil antarproses. Alghamdi dan Alaghband [10] menjelaskan bahwa pada algoritma pengurutan paralel, biaya komunikasi dapat menjadi lebih besar daripada keuntungan paralelisasi apabila ukuran data kecil atau pembagian kerja tidak dirancang dengan baik. Altarawneh et al. [4] juga menunjukkan bahwa pada *Parallel Merge Sort*, jumlah unit paralel yang terlalu banyak dapat menurunkan efisiensi karena proses *merge* dan sinkronisasi dapat menjadi *bottleneck*.

Dengan demikian, masih terdapat ruang kajian untuk menganalisis *Parallel Merge Sort* berbasis MPI pada *dataset* akademik

terstruktur, khususnya data IPK mahasiswa, dengan variasi jumlah proses yang lebih luas. Penelitian ini berfokus pada analisis waktu eksekusi *Parallel Merge Sort* menggunakan MPI dengan variasi proses 1 hingga 64 untuk menentukan konfigurasi proses yang paling optimal.

III. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif untuk menganalisis performa algoritma *Parallel Merge Sort* menggunakan *Message Passing Interface* (MPI) terhadap waktu eksekusi pengurutan data IPK mahasiswa. Pendekatan ini digunakan karena hasil penelitian diperoleh dari pengukuran waktu eksekusi pada beberapa variasi ukuran data dan jumlah proses MPI.

Tahap pertama adalah pengumpulan data. Data yang digunakan berupa *dataset* mahasiswa terkait perubahan IPK sebelum dan sesudah penggunaan *AI tools*. Data disimpan dalam format CSV dengan atribut utama berupa nama *AI tool*, IPK sebelum penggunaan *AI tool*, dan IPK setelah penggunaan *AI tool*. Selanjutnya, dilakukan perhitungan selisih IPK sebagai dasar pengurutan data.

Tahap kedua adalah pengolahan data awal. Pada tahap ini, data disiapkan dalam beberapa ukuran pengujian, yaitu 10.000, 100.000, 500.000, dan 1.000.000 data. Variasi ukuran data digunakan untuk mengetahui pengaruh jumlah data terhadap waktu eksekusi algoritma.

Tahap ketiga adalah implementasi algoritma. Proses pengurutan dilakukan menggunakan algoritma *Merge Sort* dengan pendekatan *divide and conquer*. Untuk pemrosesan paralel, data dibagi ke beberapa proses menggunakan MPI, kemudian setiap proses mengurutkan bagiannya masing-masing. Hasil pengurutan dari setiap proses selanjutnya digabungkan kembali untuk memperoleh data akhir yang terurut.

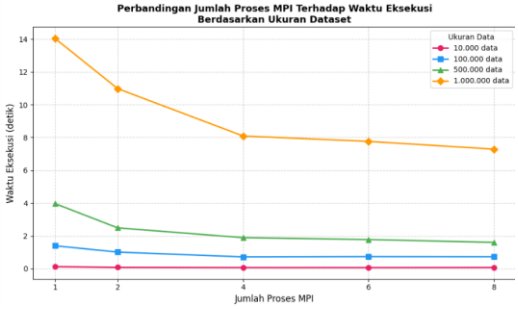
Tahap keempat adalah pengujian performa. Pengujian dilakukan dengan variasi jumlah proses MPI, yaitu 1, 2, 4, 8, 16, 32, dan 64 proses. Setiap skenario pengujian dijalankan sebanyak tiga kali, kemudian waktu eksekusinya dirata-ratakan agar hasil pengukuran lebih stabil. Penggunaan jumlah proses besar tetap disesuaikan dengan kemampuan perangkat yang digunakan.

Tahap terakhir adalah analisis hasil. Analisis dilakukan dengan membandingkan waktu eksekusi pada satu proses sebagai pendekatan sekuensial dan beberapa proses sebagai pendekatan paralel. Hasil analisis digunakan untuk mengetahui pengaruh jumlah proses terhadap performa algoritma serta menentukan konfigurasi proses yang paling optimal.

IV. HASIL DAN PEMBAHASAN

4.1 Pengaruh Jumlah Proses Terhadap Waktu Eksekusi

Untuk menganalisis pengaruh jumlah proses MPI terhadap waktu eksekusi program, dilakukan pengujian menggunakan beberapa ukuran dataset, yaitu 10.000, 100.000, 500.000, dan 1.000.000 data. Setiap dataset dieksekusi menggunakan variasi jumlah proses MPI sebanyak 1, 2, 4, 6, dan 8 proses. Hasil pengujian ditunjukkan pada Gambar 1.



Gambar 1. Hasil Perbandingan Waktu Eksekusi pada Berbagai Jumlah Proses MPI

Berdasarkan Gambar 1, peningkatan jumlah proses MPI secara umum berpengaruh terhadap penurunan waktu eksekusi program. Penurunan paling terlihat pada dataset besar, terutama 500.000 dan 1.000.000 data. Pada dataset 1.000.000 data, waktu eksekusi turun dari 14,0442 detik pada 1 proses menjadi 10,9784 detik pada 2 proses, 8,0910 detik pada 4 proses, dan 7,2893 detik pada 8 proses. Hasil ini menunjukkan bahwa pembagian data ke beberapa proses mampu mengurangi beban kerja setiap proses sehingga pengurutan berjalan lebih cepat.

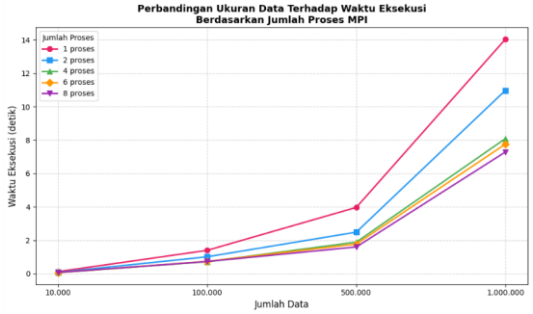
Pola serupa terlihat pada dataset 500.000 data, yaitu waktu eksekusi menurun dari 3,9688 detik pada 1 proses menjadi 2,4882

detik pada 2 proses, 1,8915 detik pada 4 proses, dan 1,6022 detik pada 8 proses. Penurunan ini menunjukkan bahwa pemrosesan paralel lebih efektif ketika ukuran data cukup besar karena setiap proses hanya menangani sebagian data. Sebaliknya, pada dataset 10.000 data, penurunan waktu tidak terlalu signifikan. Waktu eksekusi turun dari 0,1229 detik pada 1 proses menjadi 0,0664 detik pada 6 proses, tetapi meningkat kembali menjadi 0,0723 detik pada 8 proses.

Hasil pengujian menunjukkan bahwa jumlah proses MPI memengaruhi waktu eksekusi, tetapi pengaruhnya bergantung pada ukuran dataset. Semakin besar ukuran data, semakin jelas manfaat pembagian beban kerja melalui pemrosesan paralel. Namun, penurunan waktu eksekusi tidak selalu linear karena masih terdapat *overhead* komunikasi, distribusi data, dan proses *merge* antarproses.

4.2 Pengaruh Ukuran Data terhadap Waktu Eksekusi

Pengujian berikutnya dilakukan untuk melihat pengaruh ukuran data terhadap waktu eksekusi pada beberapa konfigurasi jumlah proses MPI. Pengujian ini bertujuan untuk mengetahui bagaimana peningkatan jumlah data memengaruhi beban komputasi algoritma *Merge Sort* dalam pemrosesan paralel. Hasil pengujian ditampilkan pada Gambar 2.



Gambar 2. Perbandingan Ukuran Data terhadap Waktu Eksekusi Berdasarkan Jumlah Proses

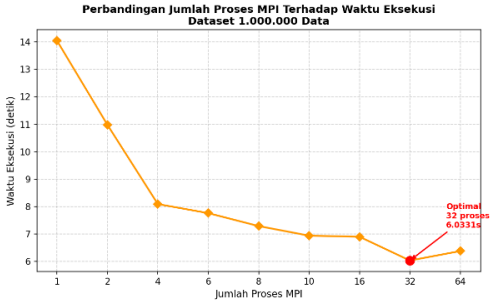
Berdasarkan Gambar 2, terlihat bahwa semakin besar ukuran dataset yang diproses, waktu eksekusi program juga semakin meningkat. Hal ini terjadi karena bertambahnya jumlah elemen data membuat proses pembagian, perbandingan, pengurutan lokal, dan penggabungan data menjadi lebih besar. Pada konfigurasi 1 proses, peningkatan waktu eksekusi paling tinggi karena seluruh proses pengurutan dilakukan oleh satu proses tanpa pembagian beban kerja.

Pada dataset 10.000 data, waktu eksekusi pada seluruh konfigurasi proses relatif kecil dan perbedaannya belum signifikan. Hal ini menunjukkan bahwa pemrosesan paralel belum memberikan keuntungan besar pada ukuran data kecil karena beban komputasi masih rendah. Namun, pada dataset 100.000, 500.000, dan 1.000.000 data, perbedaan waktu eksekusi mulai terlihat jelas. Misalnya, pada 100.000 data waktu terbaik diperoleh pada 16 proses sebesar 0,6597 detik, sedangkan pada 500.000 dan 1.000.000 data waktu terbaik diperoleh pada 32 proses, masing-masing sebesar 1,3325 detik dan 6,0331 detik.

Hasil pengujian menunjukkan bahwa ukuran data berpengaruh langsung terhadap waktu eksekusi, tetapi penggunaan MPI mampu menekan peningkatan waktu tersebut melalui pembagian beban kerja. Semakin besar dataset, semakin besar manfaat yang diperoleh dari pemrosesan paralel. Parallel Merge Sort lebih efektif diterapkan pada dataset besar dibandingkan dataset kecil.

4.3 Titik Optimal dan Efektivitas *Parallel Merge Sort*

Untuk mengetahui titik optimal jumlah proses MPI, dilakukan analisis terhadap waktu eksekusi pada setiap ukuran *dataset* dengan variasi jumlah proses yang digunakan. Analisis ini diperlukan karena peningkatan jumlah proses tidak selalu menghasilkan waktu eksekusi yang lebih rendah. Pada jumlah proses tertentu, pembagian beban kerja dapat mempercepat pengurutan, tetapi pada jumlah proses yang terlalu besar, *overhead* komunikasi dan penggabungan data dapat menurunkan performa.



Gambar 3. Titik Optimal Jumlah Proses MPI pada *Dataset* 1.000.000 Data

Berdasarkan Gambar 3, waktu eksekusi pada dataset 1.000.000 data menurun dari 1 proses hingga 32 proses. Pada 1 proses, waktu eksekusi sebesar 14,0442 detik. Waktu tersebut menurun menjadi 10,9784 detik pada 2 proses, 8,0910 detik pada 4 proses, 7,2893 detik pada 8 proses, 6,9005 detik pada 16 proses, dan mencapai nilai terbaik sebesar 6,0331 detik pada 32 proses. Hasil ini menunjukkan bahwa penambahan proses mampu meningkatkan performa selama keuntungan pembagian beban kerja masih lebih besar dibandingkan biaya komunikasi antarproses.

Namun, ketika jumlah proses ditingkatkan menjadi 64 proses, waktu eksekusi meningkat menjadi 6,3794 detik. Kenaikan ini menunjukkan bahwa 64 proses tidak lagi memberikan performa terbaik pada dataset 1.000.000 data. Kondisi tersebut terjadi karena semakin banyak proses yang digunakan, semakin besar kebutuhan komunikasi, sinkronisasi, distribusi data, dan penggabungan hasil. Hasil pengujian menunjukkan bahwa 32 proses memberikan keseimbangan terbaik antara pembagian beban kerja dan biaya komunikasi.

Tabel 2. Ringkasan Titik Optimal Jumlah Proses MPI pada Setiap Ukuran Dataset

Ukuran Dataset	Proses Optimal	Waktu Terbaik
10.000	6 Proses	0,0664 detik
100.000	16 Proses	0,6597 detik
500.00	32 Proses	1,3325 detik
1.000.000	32 Proses	6,0331 detik

Tabel 2 menunjukkan bahwa jumlah proses optimal berbeda pada setiap ukuran *dataset*. Pada *dataset* 10.000 data, waktu terbaik diperoleh pada 6 proses, sedangkan pada *dataset* 100.000 data waktu terbaik diperoleh pada 16 proses. Untuk *dataset* 500.000 dan 1.000.000 data, konfigurasi terbaik diperoleh pada 32 proses. Temuan ini menunjukkan bahwa semakin besar ukuran data, semakin besar pula peluang pemrosesan paralel untuk memberikan peningkatan performa.

Secara keseluruhan, hasil pengujian menunjukkan bahwa implementasi *Parallel Merge Sort* menggunakan MPI efektif untuk pengurutan data berukuran besar. Akan tetapi, efektivitas tersebut dipengaruhi oleh keseimbangan antara beban komputasi dan *overhead* komunikasi. Oleh karena itu, pemilihan jumlah proses perlu disesuaikan dengan ukuran *dataset* agar performa yang diperoleh tetap optimal.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, algoritma *Parallel Merge Sort* menggunakan *Message Passing Interface* (MPI) terbukti mampu meningkatkan performa pengurutan *dataset* IPK mahasiswa ditinjau dari waktu eksekusi. Penerapan MPI memungkinkan proses pengurutan dibagi ke beberapa proses sehingga waktu komputasi berkurang, terutama pada dataset besar. Pada dataset 1.000.000 data, waktu eksekusi menurun dari 14,0442 detik pada 1 proses menjadi 6,0331 detik pada 32 proses. Akan tetapi, penambahan jumlah proses tidak selalu menghasilkan peningkatan performa secara linear. Pada 64 proses, waktu eksekusi meningkat menjadi 6,3794 detik, yang menunjukkan adanya titik optimal dalam penggunaan proses MPI. Titik optimal tersebut berbeda pada setiap ukuran *dataset*, yaitu 6 proses untuk 10.000 data, 16 proses untuk 100.000 data, serta 32 proses untuk 500.000 dan 1.000.000 data. Hasil ini memperlihatkan bahwa performa *Parallel Merge Sort* menggunakan MPI dipengaruhi oleh ukuran *dataset*, jumlah proses, pembagian beban kerja, serta *overhead* komunikasi antarproses.

Untuk pengembangan penelitian selanjutnya, pengujian sebaiknya dilakukan pada lingkungan komputasi yang lebih stabil dan representatif serta memiliki sumber daya memadai, misalnya menggunakan perangkat dengan inti prosesor lebih besar atau beberapa komputer dalam sistem *cluster*. Selain itu, penelitian berikutnya dapat menggunakan dataset lebih besar dan beragam agar konsistensi performa algoritma dapat dianalisis pada karakteristik data berbeda. Pengujian juga dapat dikembangkan dengan menambahkan metrik seperti speedup, efisiensi, dan skalabilitas, sehingga dapat diketahui hubungan antara ukuran data, jumlah proses optimal, dan waktu eksekusi yang dihasilkan.

DAFTAR PUSTAKA

[1] E. Panggabean *et al.*, "Performance Analysis of Parallel Merge Sort Using MPI (Message Passing Interface) on Big Data Dataset," *J. Sist. Komput. dan Inform. Hal*, vol. 7, no. 2, pp. 594–600, 2025, doi: 10.30865/json.v7i2.9307.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Fourth Edition*. 2022. [Online]. Available: https://dl.konkur.in/2025/03/Algorithms2022-%5Bwww.konkur.in%5D.pdf

[3] A. Yang, "Approaches to the Parallelization of Merge Sort in Python," 2022, [Online]. Available: http://arxiv.org/abs/2211.16479

[4] M. Altarawneh, U. Inan, and B. Elshqeirat, "Empirical Analysis Measuring the Performance of Multi-threading in Parallel Merge Sort," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 1, pp. 72–78, 2022, doi: 10.14569/IJACSA.2022.0130110.

[5] S. L. Lestari, F. Al. Matondang, D. Syafitri, K. A. Putri, and A. Perdana, "Implementasi Algoritma Merge Sort Berbasis Divide and Conquer untuk Pengurutan Data Nilai Akademik Mahasiswa pada Sistem Informasi Akademik Universitas," vol. 2, no. 4, pp. 317–323, 2026.

[6] N. A. Asra, M. R. Albani, G. R. Pandiagan, and A. Barus, "Analisis Komparatif Efisiensi Merge Sort dan Quick Sort Menggunakan Pendekatan Divide and Conquer pada Berbagai Kondisi Data Produk," no. April, pp. 1–15, 2026.

[7] G. M. Rahmatullah, A. Fajar Zulkarnain, and M. Reza Hidayat, "Analisis Perbandingan Performa Pemrograman Sekuensial Dan Paralel Dengan Skema Uji Matrix, Filter Dan Quick Sort," *J. Teknol. Inf. Univ. Lambung Mangkurat*, vol. 6, no. 1, pp. 19–24, 2021, doi: 10.20527/jtiulm.v6i1.69.

[8] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI Portable Parallel Programming with the Message-Passing Interface Second Edition Anthony Skjellum The MIT Press Cambridge , Massachusetts*. 2014.

[9] M. A. Kartawidjaja, "Analisis kinerja perkalian matriks paralel menggunakan metrik isoefisiensi," pp. 51–54, 2008.

[10] T. Alghamdi and G. Alaghband, "High Performance Parallel Sort for Shared and Distributed Memory Mimd," pp. 113–122, 2019, doi: 10.33965/ac2019_201912l014.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

PEMODELAN DAN SIMULASI RISIKO KREDIT NASABAH BERBASIS RANDOM FOREST

Raffi Fatthoni [1)], Nabila Zahirani [2)], Arya Rayan Utama [3)], Muhammad Ridho Aidil Furqon.[4)]
1) Program Studi Teknik Informatika Universitas Mataram
2) Program Studi Teknik Informatika Universitas Mataram
3) Program Studi Teknik Informatika Universitas Mataram
4) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: f1d02310133@student.unram.ac.id

Abstrak

Risiko kredit merupakan permasalahan penting pada layanan perbankan karena keputusan pemberian pinjaman harus mempertimbangkan peluang gagal bayar dan keberlanjutan penyaluran kredit. Penelitian ini bertujuan membangun model prediksi risiko kredit nasabah dan melakukan simulasi kebijakan persetujuan pinjaman berdasarkan probabilitas default. Dataset yang digunakan adalah Credit Risk Dataset dari Kaggle yang berisi 32.581 data awal dan 12 atribut. Setelah pembersihan anomali dasar, data yang digunakan menjadi 32.573 baris dengan proporsi default 21,82% dan non-default 78,18%. Metode yang dibandingkan meliputi Logistic Regression, Random Forest, dan Gradient Boosting. Preprocessing dilakukan menggunakan imputasi median untuk fitur numerik, imputasi modus untuk fitur kategorikal, standardisasi fitur numerik, serta one-hot encoding untuk fitur kategorikal. Data dibagi dengan rasio 80:20 secara stratified dan dievaluasi menggunakan accuracy, precision, recall, F1-score, confusion matrix, dan ROC-AUC. Hasil pengujian menunjukkan bahwa Random Forest menjadi model terbaik berdasarkan ROC-AUC sebesar 0,927, accuracy 91,86%, precision default 88,24%, recall default 72,34%, dan F1-score 79,51%. Simulasi threshold menunjukkan bahwa threshold 0,30 memberikan biaya risiko proxy terendah, dengan approval rate 71,71% dan default capture rate 82,55%. Hasil ini menunjukkan bahwa model dapat digunakan sebagai alat bantu analisis risiko kredit, terutama untuk mendukung proses screening awal dan simulasi kebijakan persetujuan pinjaman.
Kata kunci: kebijakan kredit; pemodelan simulasi; Random Forest; risiko kredit; ROC-AUC

Abstract

Credit risk is an important issue in banking because loan approval decisions must consider the probability of default and the sustainability of credit distribution. This study aims to develop a customer credit risk prediction model and simulate loan approval policies based on default probability. The dataset used is the Kaggle Credit Risk Dataset, consisting of 32,581 initial records and 12 attributes. After basic anomaly cleaning, 32,573 records were used, with 21.82% default and 78.18% non-default observations. The compared methods include Logistic Regression, Random Forest, and Gradient Boosting. Preprocessing was conducted using median imputation for numerical features, mode imputation for categorical features, numerical feature standardization, and one-hot encoding for categorical features. The data were split into 80:20 stratified training and testing sets and evaluated using accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC. The test results show that Random Forest is the best model based on a ROC-AUC of 0.927, accuracy of 91.86%, default precision of 88.24%, default recall of 72.34%, and F1-score of 79.51%. The threshold simulation indicates that a 0.30 threshold produces the lowest proxy risk cost, with an approval rate of 71.71% and a default capture rate of 82.55%. These results indicate that the model can support credit risk analysis, particularly for preliminary screening and loan approval policy simulation.
Keyword: *credit policy; credit risk; modeling simulation; Random Forest; ROC-AUC*

I. PENDAHULUAN

Risiko kredit adalah salah satu risiko utama pada aktivitas pemberian pinjaman karena pihak bank atau lembaga pembiayaan harus memperkirakan apakah calon nasabah memiliki peluang gagal bayar yang tinggi. Keputusan kredit yang terlalu longgar dapat meningkatkan kerugian akibat default, sedangkan kebijakan yang terlalu ketat dapat menurunkan jumlah nasabah yang layak memperoleh pinjaman. Oleh karena itu, pemodelan berbasis data dapat digunakan sebagai alat bantu untuk menilai probabilitas default secara lebih konsisten sebelum keputusan akhir dilakukan oleh analis kredit.

Dalam konteks pemodelan dan simulasi, penelitian ini tidak hanya membangun model klasifikasi, tetapi juga menyimulasikan perubahan

threshold probabilitas default untuk melihat dampaknya terhadap approval rate, reject atau review rate, default capture rate, dan biaya risiko proxy. Pendekatan ini relevan dengan kebutuhan bisnis perbankan karena model prediksi perlu diterjemahkan menjadi skenario kebijakan yang dapat dibandingkan secara kuantitatif. Dataset yang digunakan adalah Credit Risk Dataset dari Kaggle yang memuat atribut demografis nasabah, informasi pinjaman, dan status default [1].

Tujuan penelitian ini adalah membandingkan performa Logistic Regression, Random Forest, dan Gradient Boosting dalam memprediksi risiko kredit nasabah, menentukan model terbaik berdasarkan ROC-AUC dan F1-score, serta mengevaluasi skenario threshold sebagai simulasi kebijakan persetujuan pinjaman. Manfaat penelitian ini adalah memberikan gambaran proses analitik yang dapat digunakan untuk screening awal risiko kredit dan mendukung pengambilan keputusan berbasis data.

Template: Berisi latar belakang atau alasan penelitian; Teori pendukung; Tujuan penelitian; Manfaat hasil penelitian, hipotesis (bila ada). Daftar pustaka yang digunakan harus relevan dengan konsep penelitian [1].

II. TINJAUAN PUSTAKA

Credit scoring merupakan proses penilaian kelayakan kredit berdasarkan karakteristik peminjam dan histori kredit. Penelitian terkini menunjukkan bahwa metode machine learning dapat membantu proses credit scoring karena mampu mempelajari pola nonlinier dan hubungan antarfitur yang sulit ditangkap oleh aturan manual sederhana [2]. Dalam penelitian ini, Logistic Regression digunakan sebagai baseline yang mudah diinterpretasikan, sedangkan Random Forest dan Gradient Boosting digunakan sebagai model ensemble untuk menangkap hubungan yang lebih kompleks.

Random Forest membangun banyak pohon keputusan dari sampel data dan fitur yang dipilih secara acak, kemudian menggabungkan hasilnya untuk meningkatkan stabilitas prediksi [3]. Gradient Boosting membangun model secara bertahap dengan memperbaiki kesalahan model sebelumnya [4]. Kinerja model dievaluasi menggunakan confusion matrix, accuracy, precision, recall, F1-score, dan ROC-AUC. ROC-AUC digunakan karena mampu mengukur kemampuan model membedakan kelas default dan non-default pada berbagai threshold klasifikasi [5]. Implementasi pemodelan dilakukan dalam notebook Google Colab berbasis Python menggunakan pustaka scikit-learn karena menyediakan pipeline preprocessing, evaluasi model, validasi silang, dan algoritma klasifikasi yang digunakan dalam penelitian [6].

III. METODOLOGI PENELITIAN

Penelitian menggunakan pendekatan kuantitatif berbasis dataset sekunder. Data awal berjumlah 32.581 baris dan setelah pembersihan anomali dasar menjadi 32.573 baris. Aturan pembersihan yang digunakan mencakup rentang usia 18 sampai 100 tahun, lama kerja 0 sampai 60 tahun atau kosong, dan pendapatan maksimum 2.000.000. Target prediksi adalah loan_status, dengan nilai 0 sebagai non-default dan nilai 1 sebagai default. Seluruh tahapan eksperimen dijalankan dalam notebook Google Colab, sehingga hasil evaluasi dan visualisasi ditampilkan langsung pada output cell notebook.

Fitur numerik yang digunakan meliputi person_age, person_income, person_emp_length, loan_amnt, loan_int_rate, loan_percent_income, dan cb_person_cred_hist_length. Fitur kategorikal meliputi person_home_ownership, loan_intent, loan_grade, dan cb_person_default_on_file. Missing value pada fitur numerik ditangani dengan median imputation, sedangkan fitur kategorikal menggunakan most frequent imputation. Fitur numerik distandardisasi dengan StandardScaler dan fitur kategorikal diubah dengan OneHotEncoder.

Data dibagi menjadi data latih dan data uji dengan rasio 80:20 menggunakan stratified split berdasarkan loan_status. Jumlah data latih adalah 26.058 baris dan data uji 6.515 baris. Selain pengujian pada data uji, validasi silang 5-fold dilakukan pada data latih untuk melihat kestabilan performa model. Model terbaik dipilih berdasarkan ROC-AUC tertinggi dan F1-score sebagai pembanding apabila nilai ROC-AUC sama.

Simulasi kebijakan dilakukan dengan mengubah threshold probabilitas default dari 0,10 sampai 0,90. Nasabah dengan probabilitas default di atas threshold diklasifikasikan sebagai berisiko dan masuk kategori ditolak atau perlu review lanjutan. Untuk menyederhanakan analisis bisnis,

penelitian ini menggunakan biaya risiko proxy yang memberi bobot kesalahan false negative lima kali lebih besar daripada false positive karena nasabah default yang lolos sebagai non-default lebih berisiko bagi pemberi kredit. Rumus biaya risiko proxy ditunjukkan pada Persamaan (1).

Cost = 5 x FN + 1 x FP (1)

IV. HASIL DAN PEMBAHASAN

Setelah pembersihan, dataset memiliki 32.573 baris dengan 12 kolom. Kelas non-default berjumlah 25.466 data atau 78,18%, sedangkan kelas default berjumlah 7.107 data atau 21,82%. Komposisi ini menunjukkan adanya ketidakseimbangan kelas, sehingga metrik recall, F1-score, dan ROC-AUC perlu diperhatikan selain accuracy.

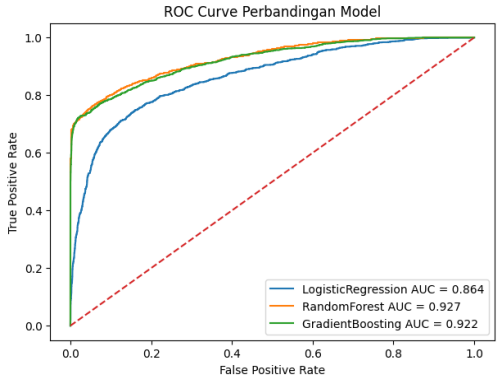
Tabel 1. Dataset setelah pembersihan

Komponen	Nilai
Jumlah data awal	32.581
Jumlah data setelah pembersihan	32.573
Jumlah fitur	11 fitur prediktor + 1 target
Non-default	25.466 data (78,18%)
Default	7.107 data (21,82%)
Missing person_emp_length	895 data (2,75%)
Missing loan_int_rate	3.115 data (9,56%)

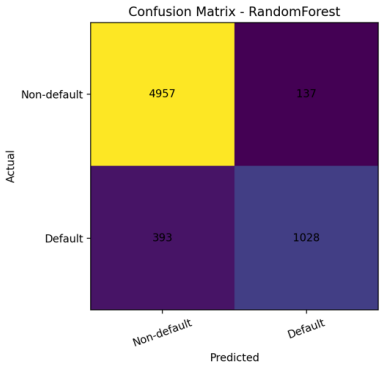
Hasil pengujian pada data uji menunjukkan bahwa Gradient Boosting memperoleh accuracy tertinggi, yaitu 92,54%, dan precision default tertinggi, yaitu 95,17%. Namun, Random Forest memperoleh ROC-AUC tertinggi sebesar 0,927 dan recall default 72,34%, lebih tinggi dibandingkan Gradient Boosting yang memiliki recall default 69,32%. Dalam konteks risiko kredit, recall default penting karena menunjukkan proporsi nasabah default aktual yang berhasil terdeteksi oleh model.

Tabel 2. Hasil evaluasi model pada data uji

Model	Accuracy	Precision default	Recall default	F1 default	ROC-AUC
Logistic Regression	81,00%	54,64%	75,79%	63,50%	0,864
Random Forest	91,86%	88,24%	72,34%	79,51%	0,927
Gradient Boosting	92,54%	95,17%	69,32%	80,21%	0,922



Gambar 1. ROC Curve



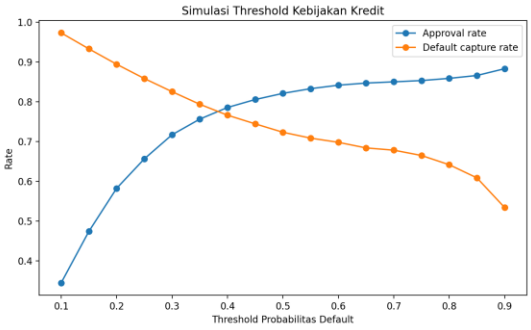
Gambar 2. Confusion matrix model Random Forest

Confusion matrix Random Forest menunjukkan 4.957 non-default diprediksi benar sebagai non-default dan 1.028 default berhasil diprediksi benar sebagai default. Kesalahan false positive berjumlah 137, yaitu nasabah non-default yang diprediksi default, sedangkan false negative berjumlah 393, yaitu nasabah default yang diprediksi non-default. Nilai false negative ini menjadi perhatian utama karena merepresentasikan potensi kredit berisiko yang tetap disetujui apabila threshold standar digunakan.

Simulasi threshold menunjukkan adanya trade-off antara pertumbuhan approval rate dan kemampuan menangkap nasabah default. Threshold yang semakin rendah membuat kebijakan lebih konservatif karena lebih banyak nasabah masuk kategori berisiko, sedangkan threshold yang semakin tinggi meningkatkan approval rate tetapi menurunkan default capture rate.

Tabel 3. Hasil simulasi threshold Random Forest

Threshold	Approval rate	Default capture	False reject	Cost proxy
0,30	71,71%	82,55%	13,15%	1.910
0,40	78,57%	76,64%	6,03%	1.967
0,50	82,12%	72,34%	2,69%	2.102
0,60	84,17%	69,81%	0,77%	2.184
0,70	85,02%	67,84%	0,24%	2.297



Gambar 3. Simulasi perubahan threshold terhadap approval rate dan default capture rate

Berdasarkan biaya risiko proxy, threshold 0,30 menghasilkan nilai biaya terendah, yaitu 1.910. Pada threshold tersebut, approval rate sebesar 71,71% dan default capture rate sebesar 82,55%. Jika bank ingin menjaga jumlah persetujuan lebih tinggi, threshold 0,50 dapat digunakan karena approval rate mencapai 82,12%, tetapi biaya proxy meningkat menjadi 2.102 dan default capture turun menjadi 72,34%. Dengan demikian, threshold 0,30 lebih sesuai untuk strategi konservatif yang memprioritaskan pengendalian risiko, sedangkan threshold 0,50 lebih sesuai untuk strategi yang menyeimbangkan risiko dan volume persetujuan.

Tabel 4. Analisis risk band

Risk band	Jumlah data	Rata-rata prob. default	Default aktual
Low	4277	10,35%	201 (4,70%)
Medium	1073	33,97%	192 (17,89%)
High	210	58,96%	83 (39,52%)
Very High	955	94,24%	945 (98,95%)

Analisis risk band memperlihatkan bahwa kelompok Low memiliki default aktual 4,70%, sedangkan kelompok Very High memiliki default aktual 98,95%. Pola ini menunjukkan bahwa probabilitas yang dihasilkan model memiliki urutan risiko yang masuk akal untuk kebutuhan segmentasi awal. Berdasarkan feature importance, variabel yang paling berpengaruh adalah loan_percent_income, person_income, loan_int_rate, loan_grade_D, dan loan_amnt. Secara bisnis, temuan ini logis karena rasio pinjaman terhadap pendapatan, kemampuan pendapatan, bunga pinjaman, dan kualitas grade pinjaman berkaitan langsung dengan kemampuan bayar nasabah.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil membangun model pemodelan dan simulasi risiko kredit nasabah menggunakan Credit Risk Dataset. Dari tiga model yang dibandingkan, Random Forest dipilih sebagai model terbaik karena memperoleh ROC-AUC tertinggi sebesar 0,927 dengan accuracy 91,86%, precision default 88,24%, recall default 72,34%, dan F1-score 79,51%. Hasil simulasi threshold menunjukkan bahwa threshold 0,30 memberikan biaya risiko proxy paling rendah, sedangkan threshold 0,50 dapat menjadi alternatif apabila bank ingin mempertahankan approval rate yang lebih tinggi. Model ini dapat digunakan sebagai alat bantu screening awal risiko kredit, bukan sebagai satu-satunya dasar keputusan final. Penelitian selanjutnya disarankan menambahkan data riwayat pembayaran yang lebih rinci, melakukan kalibrasi probabilitas, serta menguji fairness model agar kebijakan kredit lebih stabil dan bertanggung jawab.

DAFTAR PUSTAKA

[1] Lao Tse, "Credit Risk Dataset," Kaggle. [Online]. Tersedia di: https://www.kaggle.com/datasets/laotse/credit-risk-dataset [Diakses: 6 Jun. 2026].

[2] O. Koc, O. Ugur, dan A. S. Kestel, "The impact of feature selection and transformation on machine learning methods in determining the credit scoring," arXiv:2303.05427, 2023.

[3] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, hlm. 5-32, 2001.

[4] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," The Annals of Statistics, vol. 29, no. 5, hlm. 1189-1232, 2001.

[5] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, hlm. 861-874, 2006.

[6] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, hlm. 2825-2830, 2011.

SISTEM PENDUKUNG KEPUTUSAN REKOMENDASI LABORATORIUM KEAHLIAN MAHASISWA INFORMATIKA UNRAM MENGGUNAKAN METODE SIMPLE ADDITIVE WEIGHTING

Gusti Ayu Marsha Widyaswari [1)], Winona Andien Jihan Habbibah [2)], Nadin Mufida [3)], Nazril Hidayat [4)], Fatriya Annastha Putra [5)], Herliana Rosika [6)]
Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: ayumarsha141@gmail.com

Abstrak

Pemilihan laboratorium keahlian yang tepat merupakan keputusan strategis bagi mahasiswa Program Studi Teknik Informatika Universitas Mataram dalam menentukan fokus penelitian Tugas Akhir. Namun, sebaran peminatan mahasiswa pada periode 2023–2026 menunjukkan ketimpangan akibat proses pemilihan yang masih bersifat subjektif. Dari total 232 mahasiswa, akumulasi bimbingan Tugas Akhir didominasi oleh Laboratorium Sistem Informasi Enterprise (SIE) sebanyak 103 mahasiswa, diikuti Laboratorium Komunikasi Data dan Sistem Tertanam (KDST) sebanyak 78 mahasiswa, serta Laboratorium Sistem Cerdas dan Aplikasinya (SCA) sebanyak 51 mahasiswa. Penelitian ini mengimplementasikan metode Simple Additive Weighting (SAW) sebagai model pengambilan keputusan untuk memberikan rekomendasi laboratorium secara objektif melalui pendekatan pemetaan kompetensi berbasis Outcome-Based Education (OBE). Kriteria penilaian diturunkan dari capaian akademik mahasiswa pada mata kuliah inti semester 6 yang dipetakan ke kompetensi laboratorium berdasarkan kurikulum OBE dan dikonversi ke dalam skala numerik menggunakan pendekatan sub-weighting nilai huruf. Alternatif keputusan meliputi Laboratorium SCA, KDST, dan SIE. Pengujian dilakukan menggunakan berbagai skenario profil akademik yang merepresentasikan karakteristik kompetensi masing-masing laboratorium. Hasil pengujian menunjukkan bahwa metode SAW mampu menghasilkan rekomendasi yang konsisten terhadap kompetensi dominan pada setiap skenario, baik pada profil yang berorientasi kecerdasan buatan, jaringan komputer, maupun rekayasa perangkat lunak. Pada salah satu skenario dengan dominasi kompetensi rekayasa perangkat lunak dan sistem informasi, Laboratorium SIE memperoleh nilai preferensi tertinggi sebesar 0,909, diikuti Laboratorium SCA sebesar 0,865 dan Laboratorium KDST sebesar 0,842. Hasil penelitian menunjukkan bahwa model pemetaan kompetensi berbasis OBE yang diusulkan mampu menyediakan rekomendasi laboratorium yang transparan, terukur, dan selaras dengan karakteristik kompetensi akademik mahasiswa sebagai dasar pengambilan keputusan peminatan Tugas Akhir.
Kata kunci: Kurikulum OBE; Laboratorium Keahlian; SAW; Sistem Pendukung Keputusan.

Abstract

Selecting the appropriate specialization laboratory is a strategic decision for students of the Informatics Engineering Study Program at the University of Mataram in determining the focus of their Final Project research. However, the distribution of student specialization interests during the 2023–2026 period indicates an imbalance caused by a selection process that remains subjective. Out of 232 students, the distribution of Final Project supervision was dominated by the Enterprise Information Systems Laboratory (SIE) with 103 students, followed by the Data Communication and Embedded Systems Laboratory (KDST) with 78 students, and the Intelligent Systems and Applications Laboratory (SCA) with 51 students. This study implements the Simple Additive Weighting (SAW) method as a decision-making model to provide objective laboratory recommendations through an Outcome-Based Education (OBE)-based competency mapping approach. The assessment criteria are derived from students' academic performance in core sixth-semester courses, mapped to laboratory competencies based on the OBE curriculum and converted into numerical values using a letter-grade sub-weighting approach. The decision alternatives include the SCA, KDST, and SIE laboratories. Evaluation was conducted using various academic profile scenarios representing the competency characteristics of each laboratory. The results show that the SAW method consistently generates recommendations aligned with the dominant competencies represented in each scenario, including profiles oriented toward artificial intelligence, computer networks, and software engineering. In one scenario characterized by competencies in software engineering and information systems, the SIE Laboratory achieved the highest preference value of 0.909, followed by the SCA Laboratory with 0.865 and the KDST Laboratory with 0.842. The findings indicate that the proposed OBE-based competency mapping model can provide transparent, measurable, and competency-aligned laboratory recommendations to support students' Final Project specialization decisions.
Keyword: *Decision Support System; Laboratory Specialization; OBE Curriculum; SAW.*

I. PENDAHULUAN

Program Studi Teknik Informatika Universitas Mataram (PSTI Unram) menerapkan kurikulum berbasis Outcome-Based Education (OBE) yang dirancang untuk menghasilkan lulusan dengan kompetensi terukur dan terarah [1]. Berdasarkan dokumen kurikulum resmi prodi, jalur pendalaman kompetensi mahasiswa untuk penyelesaian Tugas Akhir dibagi ke dalam tiga laboratorium keahlian utama, yaitu: (1) Lab Sistem Cerdas dan Aplikasinya di bidang AI dan data science; (2) Lab Komunikasi Data dan Sistem Tertanam di bidang jaringan, IoT, dan HPC; serta (3) Lab Sistem Informasi Enterprise di bidang rekayasa perangkat lunak dan proses bisnis. Namun, proses pemilihan laboratorium saat ini belum didukung oleh instrumen rekomendasi berbasis data. Akibatnya, keputusan peminatan masih sangat bergantung pada pertimbangan masing-masing mahasiswa sehingga berpotensi menimbulkan ketidaksesuaian antara kompetensi akademik dan fokus penelitian yang dipilih.

Urgensi ini diperkuat oleh data internal pada platform Sistem Informasi Tugas Akhir PSTI Unram dalam tiga tahun terakhir yang menunjukkan distribusi peminatan tidak merata [2]. Dari 232 mahasiswa aktif, akumulasi sebaran didominasi oleh Lab SIE sebanyak 103 mahasiswa, diikuti Lab KDST sebanyak 78 mahasiswa, dan Lab SCA sebanyak 51 mahasiswa. Ketimpangan beban bimbingan antar-laboratorium tersebut mempertegas perlunya standardisasi rekomendasi yang adil. Ketidakseimbangan tersebut tidak selalu menunjukkan perbedaan minat yang sebenarnya, tetapi juga dapat mengindikasikan belum optimalnya proses pemetaan kompetensi mahasiswa terhadap laboratorium yang sesuai.

Beberapa penelitian sebelumnya telah menerapkan metode Simple Additive Weighting (SAW) untuk mendukung pengambilan keputusan pada bidang pendidikan, seperti pemilihan jurusan, peminatan akademik, maupun rekomendasi laboratorium. Namun, sebagian besar penelitian tersebut masih menggunakan parameter akademik umum seperti IPK, nilai mata kuliah, atau minat mahasiswa tanpa mengaitkannya secara langsung dengan capaian pembelajaran dan kompetensi yang ditetapkan dalam kurikulum Outcome-Based Education (OBE). Selain itu, belum ditemukan penelitian yang secara khusus merekomendasikan laboratorium keahlian berdasarkan pemetaan kompetensi akademik mahasiswa yang mengacu pada CPL dan kurikulum OBE menggunakan metode SAW.

Berdasarkan research gap tersebut, penelitian ini mengusulkan penerapan metode Simple Additive Weighting (SAW) yang mengintegrasikan pemetaan kompetensi berbasis OBE untuk menghasilkan rekomendasi laboratorium keahlian mahasiswa PSTI Universitas Mataram. Metode SAW dipilih karena mampu mendukung pengambilan keputusan multikriteria secara lebih objektif, transparan, dan mudah diinterpretasikan, serta sesuai untuk proses pemeringkatan alternatif berdasarkan sejumlah kriteria bertipe benefit [3], [4]. Kontribusi utama penelitian ini terletak pada integrasi pemetaan kompetensi berbasis CPL dan kurikulum OBE ke dalam mekanisme rekomendasi laboratorium keahlian, sehingga rekomendasi yang dihasilkan tidak hanya mempertimbangkan performa akademik mahasiswa, tetapi juga kesesuaiannya dengan kompetensi yang menjadi fokus masing-masing laboratorium.

II. TINJAUAN PUSTAKA

A. Multi-Attribute Decision Making

Multi-Attribute Decision Making (MADM) adalah pendekatan matematis untuk mengevaluasi atau menyeleksi sejumlah m alternatif (A_i) berdasarkan sekumpulan n kriteria (C_j) yang saling bebas. Prosesnya dieksekusi melalui pembentukan matriks keputusan (X) sebagai berikut [4]:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \quad (1)$$

Dimana x_{ij} merupakan rating kinerja alternatif ke-i terhadap kriteria ke-j. Tingkat kepentingan setiap kriteria ditentukan oleh vektor bobot $W = \{w_1, w_2, \ldots, w_n\}$[4].

B. Metode Simple Additive Weighting

Metode SAW mencari jumlah terbobot dari rating kinerja pada setiap alternatif untuk seluruh atribut. Tahap awal dilakukan dengan menormalisasi matriks Keputusan (X) menjadi matriks (R) menggunakan rumus [4]:

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{\max\limits_i x_{ij}} & \text{jika } j \text{ adalah kriteria keuntungan (benefit)} \\ \dfrac{\min\limits_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah kriteria biaya (cost)} \end{cases} \quad (2)$$

Nilai preferensi akhir (V_i) untuk menentukan alternatif terbaik dihitung dengan persamaan [4]:

$$V_i = \sum_{j=1}^{n} w_j \, r_{ij} \quad (3)$$

Nilai V_i terbesar menunjukkan prioritas alternatif terbaik yang direkomendasikan sistem [5]. Penerapan SAW untuk peminatan studi terbukti menghasilkan akurasi kesesuaian yang tinggi [6], serta terbukti meningkatkan efisiensi proses penempatan bidang keahlian mahasiswa di perguruan tinggi [7].

C. Penelitian Terdahulu

Kajian terhadap penelitian terdahulu yang relevan dengan penerapan metode MADM di bidang akademik dipetakan pada Tabel 1.

Tabel 1. Perbandingan Peneliti Terdahulu

Penulis	Metode	Objek Penelitian
Tinaliah & Elizabeth [3]	SAW	Pemilihan Peminatan Program Studi
Zakaria et al. [6]	SAW	Pemilihan Bidang Minat Mahasiswa
Febryanti et al. [7]	SAW	Pemilihan Bidang Peminatan Mahasiswa
Badruddin et al. [8]	SAW	Penentuan Jurusan Calon Mahasiswa Baru
Rosmiati [9]	SAW	Pemilihan Asisten Laboratorium

Berdasarkan hasil kajian pada Tabel 1, belum ditemukan penelitian yang secara khusus menerapkan metode Simple Additive Weighting (SAW) untuk merekomendasikan laboratorium keahlian berdasarkan pemetaan kompetensi yang mengacu pada Capaian Pembelajaran Lulusan (CPL) dan kurikulum Outcome-Based Education (OBE). Temuan tersebut menjadi dasar pengembangan model rekomendasi yang diusulkan dalam penelitian ini. Perbedaan kontribusi penelitian ini terhadap penelitian terdahulu dirinci lebih lanjut pada Tabel 2.

Tabel 2. Perbandingan Kontribusi Penelitian

Aspek	Penelitian Terdahulu	Penelitian Ini
Metode	SAW	SAW
Objek	Peminatan/Jurusan/Laboratorium Umum	Laboratorium Keahlian PSTI Unram
Dasar Penilaian	IPK atau nilai akademik umum	Mata kuliah inti semester 6
Pendekatan OBE	Tidak digunakan	Digunakan
Pemetaan Kompetensi Laboratorium	Terbatas	Ya
Bobot Fleksibel	Tidak dibahas	Ya
Luaran	Rekomendasi peminatan	Rekomendasi laboratorium berbasis kompetensi OBE

D. Kurikulum OBE dan Pemetaan Kompetensi Laboratorium

Kurikulum OBE berfokus pada capaian kompetensi lulusan (Program Learning Outcomes/PLO) [1], yang menuntut pengukuran capaian pembelajaran berkala untuk mengevaluasi kapabilitas mahasiswa [10]. Rekam jejak nilai mata kuliah mahasiswa diinterpretasikan langsung sebagai indikator penguasaan kompetensi spesifik laboratorium berikut : Laboratorium SCA berfokus pada kecerdasan buatan dan data science, KDST pada jaringan komputer, IoT, dan sistem tertanam, sedangkan SIE berfokus pada sistem informasi dan rekayasa perangkat lunak [1]. Dalam pendekatan OBE, capaian pembelajaran mata kuliah dirancang untuk merepresentasikan penguasaan kompetensi tertentu sehingga nilai akademik dapat digunakan sebagai indikator pencapaian kompetensi mahasiswa [1], [10].

Tabel 3. Pemetaan Mata Kuliah terhadap Kompete Laboratorium

Mata Kuliah	Kompetensi Dominan	Laboratorium
Internet Of Things (IoT)	Embedded System & Network	KDST
Fuzzy Logic	Artificial Intelligence	SCA
Distributed System	Enterprise System	SIE
Mobile Programming	Software Development	SIE
Visual Programming	Software Engineering	SIE

Pemetaan pada Tabel 3 disusun berdasarkan kesesuaian capaian pembelajaran mata kuliah dengan fokus bidang keahlian laboratorium pada kurikulum OBE PSTI Universitas Mataram [1]. Namun, dalam proses perhitungan SAW, setiap mata kuliah tetap dinilai relevansinya terhadap ketiga laboratorium secara proporsional, sehingga satu mata kuliah dapat memberikan kontribusi nilai pada lebih dari satu laboratorium dengan besaran yang berbeda-beda sesuai tingkat kecocokannya.

III. METODOLOGI PENELITIAN

A. Tahapan Penelitian

Penelitian ini dilaksanakan melalui lima tahapan sistematis guna memastikan hasil rekomendasi secara lebih objektif, transparan, dan dapat dipertanggungjawabkan:

1) Pengumpulan Data Kurikulum dan Riwayat Peminatan: Tahap awal melibatkan ekstraksi data kurikulum OBE PSTI Unram [1], serta data historis sebaran topik Tugas Akhir mahasiswa selama tiga tahun terakhir yang diperoleh dari platform Sistem Informasi Tugas Akhir PSTI Unram [2].

2) Penentuan Kriteria dan Alternatif: Kriteria keputusan diturunkan dari sebaran kelompok mata kuliah inti pada semester 6. Mata kuliah semester 6 dipilih karena merupakan kelompok mata kuliah yang paling

merepresentasikan kompetensi spesifik laboratorium pada kurikulum OBE PSTI Universitas Mataram. Sementara itu, alternatif keputusan meliputi Lab SCA, Lab KDST, dan Lab SIE [1].

3) Konfigurasi Pembobotan Fleksibel: Admin sistem melakukan pengaturan distribusi bobot kriteria untuk masing-masing laboratorium melalui basis data pengelolaan. Total alokasi nilai bobot kepentingan kriteria (W) dari kriteria C_1 sampai C_5 secara global diwajibkan bernilai mutlak 1,00 (100%) untuk menjamin validitas penjumlahan terbobot standar pada metode SAW.

4) Normalisasi Skor Akademik (Sub-Weighting): Mengonversi nilai huruf semester 6 sampel mahasiswa ke skor numerik dengan skala 0–1 (A = 1,00; B+ = 0,90; B = 0,80; B- = 0,70; C+ = 0,65; C = 0,60; D = 0,50; E = 0,00). Pendekatan sub-weighting ini sangat penting guna mencegah bias penilaian pada standarisasi parameter MADM [4].

5) Kalkulasi Perankingan SAW: Proses eksekusi perhitungan menggunakan algoritma SAW untuk memproduksi nilai preferensi akhir mahasiswa pada tiap-tiap laboratorium alternatif.

B. Formulasi Matematis SAW

Seluruh nilai mata kuliah diperlakukan sebagai atribut benefit dan dinormalisasi menggunakan Persamaan (2). Seluruh kriteria dikategorikan sebagai atribut benefit karena nilai akademik yang lebih tinggi menunjukkan tingkat penguasaan kompetensi yang lebih baik terhadap bidang keahlian terkait. Selanjutnya nilai preferensi dihitung menggunakan Persamaan (3) melalui perkalian matriks normalisasi dengan bobot kriteria untuk memperoleh peringkat laboratorium.

C. Matriks Konfigurasi Bobot Kriteria

Bobot setiap kriteria ditentukan berdasarkan tingkat keterkaitan mata kuliah terhadap kompetensi yang menjadi fokus masing-masing laboratorium keahlian. Penentuan bobot mengacu pada dokumen kurikulum Outcome-Based Education (OBE) PSTI Universitas Mataram yang memetakan hubungan antara capaian pembelajaran mata kuliah dengan bidang keahlian laboratorium. Semakin tinggi tingkat relevansi suatu mata kuliah terhadap kompetensi laboratorium, maka semakin besar bobot yang diberikan pada kriteria tersebut. Pemilihan lima mata kuliah pada penelitian ini didasarkan pada tingkat relevansi tertinggi terhadap kompetensi utama yang direpresentasikan oleh masing-masing laboratorium keahlian dalam kurikulum OBE, sekaligus menjaga kompleksitas model tetap sederhana dan mudah diinterpretasikan.

Perlu ditegaskan bahwa bobot kriteria W_j pada Tabel 4 bersifat global dan diterapkan secara identik untuk ketiga alternatif laboratorium (SCA, KDST, SIE) dalam proses perhitungan SAW. Perbedaan nilai preferensi akhir antar-laboratorium murni dihasilkan dari perbedaan nilai rating kecocokan X_{ij} setiap mata kuliah terhadap masing-masing laboratorium, bukan dari perbedaan bobot kriterianya. Dengan kata lain, bobot menentukan seberapa penting suatu mata kuliah secara keseluruhan, sedangkan nilai X_{ij} menentukan seberapa relevan capaian mata kuliah tersebut terhadap kompetensi laboratorium tertentu.

Tabel 4. Sampel Matriks Distribusi Bobot Mata Kuliah

Kriteria	Mata Kuliah	Bobot Kepentingan
C_1	Internet Of Things (IoT)	0,20
C_2	Fuzzy Logic	0,10
C_3	Distributed System	0,25
C_4	Mobile Programming	0,20
C_5	Visual Programming	0,25

Distribusi bobot pada Tabel 4 disusun berdasarkan tingkat relevansi mata kuliah terhadap kompetensi utama masing-masing laboratorium sesuai kurikulum OBE PSTI Universitas Mataram, dengan syarat total bobot bernilai 1,00 [1].

D. Arsitektur Sistem

Arsitektur sistem yang diusulkan menggambarkan alur pengolahan data akademik mahasiswa hingga menghasilkan rekomendasi laboratorium keahlian menggunakan metode Simple Additive Weighting (SAW). Sistem memanfaatkan nilai mata kuliah semester 6 sebagai masukan utama yang dipetakan terhadap kompetensi laboratorium berdasarkan kurikulum Outcome-Based Education (OBE). Alur proses sistem ditunjukkan pada Gambar 1.

Gambar 1. Arsitektur Sistem Rekomendasi Laboratorium Keahlian Menggunakan Metode SAW

Berdasarkan Gambar 1, sistem mengolah data nilai akademik mahasiswa melalui tahapan

konversi nilai huruf ke skala numerik, pembentukan matriks keputusan, penentuan bobot kriteria, normalisasi SAW, perhitungan nilai preferensi, dan perankingan alternatif laboratorium. Hasil akhir proses berupa rekomendasi laboratorium keahlian yang sesuai dengan kompetensi akademik mahasiswa.

IV. HASIL DAN PEMBAHASAN

A. Konfigurasi Parameter Bobot Fleksibel

Bobot kriteria dan nilai kecocokan mata kuliah terhadap laboratorium disimpan dalam basis data sehingga dapat diperbarui ketika terjadi perubahan kurikulum atau fokus keilmuan laboratorium tanpa mengubah algoritma SAW. Pendekatan ini membuat sistem lebih adaptif dibandingkan penggunaan bobot statis.

B. Simulasi Kasus Perhitungan SAW

Sebagai ilustrasi kalkulasi manual, digunakan satu studi kasus data akademik Mahasiswa A. Nilai huruf tiap mata kuliah dikonversi ke skor dasar lalu diboboti menurut kecocokan mata kuliah terhadap profil tiap laboratorium, sehingga satu nilai huruf dapat menghasilkan nilai X berbeda pada kolom SCA, KDST, dan SIE sebelum dinormalisasi dengan Persamaan (2). Studi kasus ini berbeda dari pengujian sepuluh skenario pada bagian E.

Tabel 5. Data Nilai Matriks Keputusan (X) dan Hasil Normalisasi (R) Mahasiswa A

Kriteria	Nilai Huruf	X_{SCA}	X_{KDST}	X_{SIE}	r_{ij}
C_1	B+	0,90	0,80	0,90	1,00 (SCA/SIE) / 0.889 (KDST)
C_2	B+	0,90	0,80	0,80	1,00 (SCA) / 0.889 (KDST/SIE)
C_3	A	0,90	0,80	1,00	0.90 (SCA) / 0.8 (KDST) / 1.00 (SIE)
C_4	A	0,70	1,00	0,60	0.70 (SCA) / 1.00 (KDST) / 0.60 (SIE)
C_5	A	0,80	0,70	1,00	0.80 (SCA) / 0.70 (KDST) / 1.00 (SIE)

C. Perhitungan Nilai Preferensi SAW

Berdasarkan matriks rating kinerja (r_{ij}) dan matriks distribusi bobot kepentingan kriteria (w_j), Model kalkulasi mengeksekusi perhitungan penjumlahan terbobot menggunakan fungsi preferensi untuk mengalkulasi nilai preferensi total (V_i) pada masing-masing alternatif laboratorium keahlian. Rincian hasil akhir proses perkalian matriks terbobot sebagai berikut:

Tabel 6. Rincian Hasil Perkalian Matriks Terbobot ($r_{ij} \times w_j$)

Kriteria	Alternatif A_1 (SCA)	Alternatif A_2 (KDST)	Alternatif A_3 (SIE)
C_1	1,00 × 0,20 = 0,200	0,889 × 0,20 = 0,178	1,00 × 0,20 = 0,200
C_2	1,00 × 0,10 = 0,100	0,889 × 0,10 = 0,089	0,889 × 0,10 = 0,089
C_3	0,90 × 0,25 = 0,225	0,80 × 0,25= 0,200	1,00 × 0,25 = 0,250
C_4	0,70 × 0,20 = 0,140	1,00 × 0,20 = 0,200	0,60 × 0,20 = 0,120
C_5	0,80 × 0,25 = 0,200	0,70 × 0,25 = 0,175	1,00 × 0,25 = 0,250
Total Preferensi (V_i)	0,865	0,842	0,909

D. Perankingan Alternatif Laboratorium

Setelah seluruh nilai preferensi dikalkulasi menggunakan algoritma SAW, model menyusun peringkat alternatif secara menurun untuk menentukan prioritas luaran rekomendasi. Berdasarkan akumulasi kalkulasi penjumlahan terbobot, diperoleh keputusan akhir urutan prioritas rekomendasi laboratorium keahlian untuk mahasiswa yang dipaparkan pada Tabel 7.

Tabel 7. Hasil Akhir Perankingan Laboratorium

Peringkat	Laboratorium Keahlian	Nilai Preferensi (V_i)
1	Lab SIE	0,909
2	Lab SCA	0,865
3	Lab KDST	0,842

Hasil perangkingan menunjukkan bahwa Laboratorium Sistem Informasi Enterprise (SIE) memperoleh nilai preferensi tertinggi ($V_{\text{SIE}} = 0{,}909$). Hasil tersebut didorong kuat oleh perolehan nilai sempurna (A) mahasiswa pada mata kuliah pilar arsitektur data seperti Distributed System. Sementara itu, Laboratorium Sistem Cerdas dan Aplikasinya (SCA) menempati peringkat kedua ($V_{\text{SCA}} = 0{,}865$), disusul oleh Laboratorium Komunikasi Data dan Sistem Tertanam (KDST) di peringkat terakhir dengan nilai ($V_{\text{KDST}} = 0{,}842$).

E. Pengujian pada Berbagai Profil Akademik

Untuk mengevaluasi konsistensi rekomendasi, dilakukan pengujian menggunakan sepuluh skenario profil akademik hipotetis yang dikonstruksi secara sintetis (bukan data transkrip riil) untuk merepresentasikan karakteristik

kompetensi tiap laboratorium. Setiap skenario dibangun dari kombinasi nilai mata kuliah semester 6 sesuai kurikulum OBE, sehingga pengujian ini memverifikasi konsistensi internal algoritma SAW dalam memetakan nilai ke laboratorium yang sesuai.

Tabel 8. Pengujian pada Berbagai Profil Akademik

Skenario	Karakteristik Dominan	Lab yang Diharapkan	Hasil SAW
S1	AI dan Data Science	SCA	SCA
S2	AI dan Data Science	SCA	SCA
S3	AI dan Data Science	SCA	SCA
S4	Jaringan dan IoT	KDST	KDST
S5	Jaringan dan IoT	KDST	KDST
S6	Jaringan dan IoT	KDST	KDST
S7	RPL dan Sistem Informasi	SIE	SIE
S8	RPL dan Sistem Informasi	SIE	SIE
S9	RPL dan Sistem Informasi	SIE	SIE
S10	Kompetensi Campuran dengan Dominasi RPL	SIE	SIE

Seluruh skenario menghasilkan rekomendasi yang sesuai dengan laboratorium yang diharapkan, dengan tingkat kesesuaian 100% dari 10 skenario yang diuji. Hasil ini menunjukkan bahwa algoritma SAW konsisten dalam memetakan nilai mata kuliah ke laboratorium yang relevan sesuai rancangan skenario. Perlu dicatat bahwa nilai kesesuaian ini merefleksikan konsistensi logika perhitungan terhadap skenario yang memang dirancang demikian, bukan tingkat akurasi terhadap data akademik mahasiswa riil.

F. Keterbatasan Penelitian

Penelitian ini masih terbatas pada pengujian menggunakan skenario profil akademik representatif dan belum melibatkan data akademik aktual dalam jumlah besar maupun validasi langsung oleh pengelola laboratorium. Penelitian selanjutnya disarankan melakukan validasi empiris menggunakan data mahasiswa aktual serta membandingkan metode SAW dengan metode MADM lainnya. Model yang diusulkan berfokus pada aspek kompetensi akademik dan belum mempertimbangkan faktor minat personal mahasiswa dalam proses rekomendasi. Selain itu, bobot kriteria masih ditentukan berdasarkan pemetaan kurikulum OBE dan belum divalidasi melalui penilaian pakar atau metode pembobotan khusus seperti AHP.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Metode Simple Additive Weighting (SAW) berhasil diterapkan sebagai model pengambilan keputusan untuk merekomendasikan laboratorium keahlian mahasiswa PSTI Unram berdasarkan capaian mata kuliah semester 6. Melalui proses konversi nilai huruf ke skala numerik dan perhitungan SAW, diperoleh urutan prioritas Laboratorium Sistem Informasi Enterprise (0,909), Laboratorium Sistem Cerdas dan Aplikasinya (0,865), dan Laboratorium Komunikasi Data dan Sistem Tertanam (0,842). Pengujian terhadap 10 skenario profil akademik hipotetis menunjukkan konsistensi internal algoritma sebesar 100% dalam memetakan kompetensi dominan ke laboratorium yang sesuai, sehingga model ini berpotensi mendukung pemilihan laboratorium secara objektif dan terukur, selaras dengan pendekatan pemetaan kompetensi OBE, meskipun validasi menggunakan data akademik riil masih diperlukan.

B. Saran

Penelitian selanjutnya dapat menambahkan kriteria selain nilai akademik, seperti portofolio, sertifikasi, dan minat karier mahasiswa untuk meningkatkan akurasi rekomendasi. Selain itu, pembobotan kriteria dapat dikembangkan menggunakan metode hibrida, seperti AHP-SAW, guna meningkatkan objektivitas penentuan bobot [11]. Validasi lebih lanjut juga perlu dilakukan menggunakan data akademik aktual dan studi longitudinal untuk mengevaluasi kesesuaian rekomendasi dengan topik Tugas Akhir yang dipilih mahasiswa.

DAFTAR PUSTAKA

[1] Program Studi Teknik Informatika Universitas Mataram, "Dokumen Kurikulum Outcome-Based Education (OBE) PSTI Unram," Mataram: Universitas Mataram, 2022.

[2] Program Studi Teknik Informatika Universitas Mataram, "Riwayat Topik Tugas Akhir Mahasiswa," Sistem Informasi Tugas Akhir (SITA) PSTI Unram. https://ta.if.unram.ac.id/index.php/riwayat-topik.

[3] Tinaliah dan T.Elizabeth, "Sistem Pendukung Keputusan Pemilihan Peminatan Program Studi Teknik Informatika Menggunakan Metode SAW," JATISI (Jurnal Teknik Informatika dan Sistem Informasi), vol. 5, no. 2, hlm. 210-218, 2019.

[4] S. Kusumadewi, S. Hartati, A. Harjoko, dan R. Wardoyo, Fuzzy Multi-Attribute Decision Making (Fuzzy MADM). Yogyakarta: Graha Ilmu, 2006.

[5] M. B. Al-Fiqrie, D. Sunarto, dan A. Sukmaaji, "Sistem Pendukung Keputusan Pemilihan Rumpun Mata Kuliah di Program Studi dengan Metode Simple Additive Weighting," Jurnal Sistem Informasi dan Komputer Akuntansi, vol. 10, no. 2, hlm. 62-69, 2021.

[6] I. Zakaria, G. I. Marthasari, dan I. Nuryasin, "Penerapan Metode Simple Additive Weighting (SAW) Pada Sistem Pendukung Keputusan Pemilihan Bidang Minat Oleh Mahasiswa," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 9, no. 3, hlm. 5267-5274, 2025.

[7] A. C. Febryanti, I. Darmawan, dan R. Andreswari, "Pemodelan Sistem Pendukung Keputusan Pemilihan Bidang Peminatan Menggunakan Metode Simple Additive Weighting Studi Kasus: Program Studi Sistem Informasi Universitas Telkom," eProceedings of Engineering, vol. 4, no. 2, 2017.

[8] M. A. Badruddin, D. M. Hutagalung, I. H. Manurung, dan R. U. Ginting, "Sistem Pendukung Keputusan Penentuan Jurusan Calon Mahasiswa Baru Dengan Metode Simple Additive Weighting (SAW) (Studi Kasus: Universitas Sari Mutiara Indonesia, Medan)," Jurnal Teknologi Kesehatan dan Ilmu Sosial (TEKESNOS), vol. 2, no. 1, hlm. 33-39, 2020.

[9] R. Rosmiati, "Sistem Pendukung Keputusan Pemilihan Asisten Laboratorium Menggunakan Metode Simple Additive Weighting (SAW)," Jurnal Saintekom: Sains, Teknologi, Komputer dan Manajemen, vol. 6, no. 1, hlm. 16-26, 2016.

[10] H. S. Bintoro, Gudnanto, A. C. Ruby, P. M. Ratri, dan Z. Romadhon, "Evaluasi Analisis Capaian Pembelajaran Lulusan melalui Sistem Informasi CPL Berbasis OBE," Paedagogie, vol. 21, no. 1, hlm. 533-544, 2026.

[11] W. Bagye dan A. Hamid, "Komparasi Perhitungan Gabungan AHP dan SAW Dengan Perhitungan Konvensional pada Perangkingan Siswa," Jurnal Fasilkom, vol. 15, no. 2, hlm. 423-429, 2025.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

MODEL PREDIKSI FINANCIAL DISTRESS EMITEN SUBSEKTOR FARMASI DI BURSA EFEK INDONESIA: EVALUASI KINERJA XGBOOST BERBASIS MULTI-HORIZON LAG

I Ketut Yudista Widnya Pramana [1)], Rasya Gonawi [2)], Ghaida Fadila Saniy [3)], Maesa Dwi Nurrizki [4)]

1) Program Studi Sains Aktuaria Institut Teknologi Sepuluh Nopember
2) Program Studi Rekayasa Kecerdasan Artifisial Institut Teknologi Sepuluh Nopember
3) Program Studi Bisnis Digital Institut Teknologi Sepuluh Nopember
4) Program Studi Sains Aktuaria Institut Teknologi Sepuluh Nopember

Corresponding author e-mail: yudistapramana05@gmail.com

Abstrak

Deteksi dini financial distress umumnya menggunakan model Altman Z"-Score yang bersifat reaktif sehingga menciptakan time-lag dalam mitigasi risiko karena harus menunggu rilisnya laporan keuangan kuartalan. Padahal, subsektor farmasi dituntut memiliki respons cepat karena sangat rentan terhadap tekanan biaya operasional, fluktuasi harga bahan baku, dan perubahan kondisi makroekonomi. Penelitian ini bertujuan untuk mengevaluasi model XGBoost dalam memprediksi financial distress emiten subsektor farmasi dengan memanfaatkan multi-horizon lag untuk menutup kekurangan dari Altman Z"-Score. Model mengintegrasikan beberapa fitur keuangan perusahaan meliputi pertumbuhan penjualan, pertumbuhan aset, pertumbuhan utang, rasio arus kas operasional terhadap total utang, dan ukuran perusahaan serta fitur makroekonomi yaitu BI-Rate, nilai tukar Rupiah, dan tingkat inflasi melalui skenario multi-horizon lag. Untuk menjaga validitas hasil prediksi dan mencegah kebocoran data, penelitian ini menerapkan metode time-based holdout dan walk-forward validation. Evaluasi model XGBoost dilakukan melalui komparasi model terhadap Random Forest dan Regresi Logistik yang diukur melalui metrik accuracy, precision, recall, F1-Score dan ROC-AUC. Berdasarkan hasil evaluasi, Random Forest menunjukkan performa terbaik dengan accuracy 93,87%, F1-Score 89,65%, dan ROC-AUC 97,75%, sementara XGBoost menunjukkan performa kompetitif dengan accuracy 89,79%, F1-Score 81,48%, dan ROC-AUC 97,34%. Analisis SHAP dan feature importance membuktikan bahwa fitur keuangan perusahaan lebih berpengaruh dalam memprediksi financial distress dibandingkan fitur makroekonomi. Penelitian ini menunjukkan bahwa pendekatan machine learning berbasis multi-horizon lag mampu memprediksi financial distress sejak dini pada emiten subsektor farmasi, sekaligus memberikan interpretasi yang jelas mengenai fitur-fitur yang berkontribusi terhadap prediksi financial distress.

Kata kunci: Financial Distress; Fitur Keuangan; Makroekonomi; Subsektor Farmasi; XGBoost

Abstract

Early detection of financial distress commonly relies on the Altman Z"-Score model, which is reactive and creates a time lag in risk mitigation because it depends on the release of quarterly financial reports. However, the pharmaceutical subsector requires a rapid response because it is highly vulnerable to operational cost pressures, raw material price fluctuations, and changes in macroeconomic conditions. This study aims to evaluate the XGBoost model in predicting financial distress among pharmaceutical subsector issuers by applying multi-horizon lag to address the limitations of the Altman Z"-Score. The model integrates firm-level financial features, including sales growth, asset growth, debt growth, the operating cash flow to total debt ratio, and firm size, as well as macroeconomic features consisting of the BI-Rate, Rupiah exchange rate, and inflation rate through multi-horizon lag scenarios. To maintain prediction validity and prevent data leakage, this study applies time-based holdout and walk-forward validation methods. XGBoost is evaluated comparatively against Random Forest and Logistic Regression using accuracy, precision, recall, F1-Score, and ROC-AUC. The evaluation results show that Random Forest achieved the best performance with 93.87% accuracy, 89.65% F1-Score, and 97.75% ROC-AUC, while XGBoost remained competitive with 89.79% accuracy, 81.48% F1-Score, and 97.34% ROC-AUC. SHAP and feature importance analyses indicate that firm-level financial features have a stronger influence on financial distress prediction than macroeconomic features. This study shows that a multi-horizon lag-based machine learning approach can predict financial distress early among pharmaceutical subsector issuers while providing clear interpretations of the features contributing to financial distress prediction.

Keywords: *Financial Distress; Financial Features; Macroeconomic; Pharmaceutical Subsector; XGBoost*

I. PENDAHULUAN

Financial distress merupakan kondisi ketika perusahaan mengalami penurunan kemampuan keuangan dalam memenuhi kewajiban operasional maupun finansialnya yang apabila tidak terdeteksi secara dini dapat berkembang menjadi kegagalan usaha. Kondisi tersebut tidak hanya berdampak terhadap perusahaan, tetapi juga dapat memengaruhi investor, kreditur, serta keberlangsungan aktivitas bisnis [1]. Oleh karena itu, kemampuan dalam memprediksi *financial distress* sejak dini menjadi penting sebagai dasar pengembangan sistem peringatan dini (*early warning system*) dalam mendukung pengambilan keputusan.

Emiten subsektor farmasi di Bursa Efek Indonesia (BEI) menjadi objek yang relevan untuk dikaji karena karakteristik industrinya yang dipengaruhi oleh tekanan biaya operasional, perubahan harga bahan baku, serta kondisi ekonomi makro. Prediksi financial distress pada subsektor ini tidak hanya bergantung pada kondisi historis laporan keuangan, tetapi juga memerlukan pendekatan yang mampu menangkap pola perubahan kondisi perusahaan secara lebih dinamis. Pendekatan konvensional seperti Altman Z"-Score telah banyak digunakan untuk mengidentifikasi potensi financial distress berdasarkan rasio keuangan [2]. Namun, model tersebut masih memiliki keterbatasan karena bersifat retrospektif dan bergantung pada informasi laporan keuangan yang telah tersedia.

Perkembangan metode machine learning memungkinkan proses prediksi dilakukan dengan memanfaatkan pola hubungan kompleks antarvariabel keuangan yang sulit ditangkap oleh metode statistik konvensional. Salah satu algoritma yang banyak digunakan pada data berbasis tabel adalah Extreme Gradient Boosting (XGBoost), yaitu metode ensemble learning berbasis gradient boosting yang memiliki kemampuan klasifikasi yang baik [3] . Huang dan Yen [4] membandingkan enam algoritma machine learning untuk prediksi financial distress dan menemukan bahwa XGBoost memberikan akurasi prediksi terbaik di antara model supervised learning lainnya.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan dan mengevaluasi kinerja model XGBoost berbasis multi-horizon lag dalam memprediksi *financial distress* pada emiten subsektor farmasi di Bursa Efek Indonesia. Model yang dikembangkan menggunakan fitur keuangan perusahaan dan fitur makroekonomi sebagai variabel prediktor, kemudian dibandingkan dengan Random Forest dan Regresi Logistik menggunakan metrik evaluasi klasifikasi. Penelitian ini juga menerapkan SHAP (*Shapley Additive Explanations*) untuk mengidentifikasi kontribusi setiap fitur terhadap hasil prediksi sehingga model tidak hanya menghasilkan prediksi, tetapi juga memberikan interpretasi mengenai faktor yang memengaruhi risiko financial distress.

II. TINJAUAN PUSTAKA

A. *Financial Distress* dan Formula Altman Z"- Score

Financial distress merupakan suatu tahap penurunan kondisi keuangan di mana perusahaan mengalami kesulitan dalam memenuhi kewajiban keuangannya. Untuk mengidentifikasi kondisi ini secara akurat, Altman mengembangkan indikator prediktif berbasis rasio keuangan berganda [5]. Dalam konteks perusahaan di negara berkembang (*emerging markets*) serta mencakup sektor manufaktur dan non-manufaktur, variasi yang digunakan adalah Altman Z"-Score. Formulasi matematika dari fungsi Altman Z"-Score didefinisikan sebagai berikut:

$$Z'' = 6{,}56X_1 + 3{,}26X_2 + 6{,}72X_3 + 1{,}05X_4 \quad (1)$$

dimana:
$X_1 =$ modal kerja / total aset,
$X_2 =$ saldo laba / total aset,
$X_3 =$ *earnings before interest and tax* (EBIT) / total aset, dan
$X_4 =$ nilai buku ekuitas / total aset.

B. Machine Learning untuk Prediksi Financial Distress

Penggunaan *machine learning* dalam memprediksi kondisi *financial distress* kini menjadi alternatif yang sangat efektif di samping penggunaan model statistik konvensional. Pendekatan ini mampu menangani hubungan non-linier yang kompleks antar rasio keuangan serta memiliki fleksibilitas tinggi dalam memproses data berskala besar, sehingga dapat menghasilkan

estimasi risiko kebangkrutan yang lebih dinamis dan adaptif terhadap perubahan kondisi pasar.

C. XGBoost, Random Forest, dan Regresi Logistik

Dalam penelitian ini, tiga algoritma machine learning digunakan dengan peran dan karasteristik yang berbeda. XGBoost (Extreme Gradient Boosting) menjadi model utama yang diuji menggunakan pendekatan *multi-horizon lag features* karena kemampuannya dalam meminimalkan fungsi kerugian secara sekuensial berbasis *ensemble trees*. Selanjutnya, Random Forest berperan sebagai model pembanding berbasis *tree* yang dibangun secara paralel untuk mengurangi varians pada dataset berskala kecil. Sementara itu, Regresi Logistik digunakan sebagai model baseline klasik yang mengukur hubungan *log-odds* antara fitur prediktor dan label biner.

D. *Multi-Horizon Lag* dan Validasi Berbasis Waktu

Untuk menangani sifat data keuangan yang berorientasi dengan waktu, penelitian ini menggunakan pendekatan *Multi-Horizon Lag*. Pendekatan ini memungkinkan model untuk memprediksi potensi *financial distress* dalam berbagai runtutan waktu ke depan (misalnya 1 tahun, 2 tahun, atau 3 tahun sebelum kejadian). Pentingnya penggunaan beberapa runtutan waktu ini dipertegas oleh Yan et al. [6] yang menyatakan bahwa pola pelemahan finansial bersifat gradual, sehingga fitur lagged multi-periode mampu menangkap sinyal bahaya secara lebih dini. Selain itu, validasi model dilakukan dengan metode validasi berbasis waktu (time-based validation) guna memastikan bahwa pengujian model mencerminkan situasi riil di masa depan tanpa adanya kebocoran data historis (*data leakage*).

E. SHAP dan *Feature Importance*

Meskipun model Machine Learning seperti XGBoost dan Random Forest memiliki akurasi yang tinggi, model tersebut sering kali dianggap sebagai kotak hitam (*black box*). Oleh karena itu, penelitian ini mengintegrasikan metode SHAP (*SHapley Additive exPlanations*) untuk memberikan transparansi dan interpretabilitas. SHAP bekerja dengan mengukur kontribusi spesifik dari setiap variabel rasio keuangan (X1 hingga X4) terhadap hasil prediksi akhir, sehingga dapat diketahui rasio mana yang paling berpengaruh secara signifikan (feature importance) dalam mendorong atau menahan status financial distress suatu perusahaan.

III. METODOLOGI PENELITIAN

A. Desain, Data, dan Variabel Penelitian

Penelitian ini menggunakan pendekatan kuantitatif berbasis machine learning dengan desain klasifikasi biner untuk memprediksi financial distress emiten subsektor farmasi di Bursa Efek Indonesia (BEI). Label penelitian dikategorikan ke dalam dua kelas, yaitu *financial distress* dan *non-financial distress*. XGBoost digunakan sebagai model utama, sedangkan Random Forest dan Regresi Logistik digunakan sebagai model pembanding.

Data yang digunakan merupakan data emiten kuartalan pada periode kuartal I 2021 hingga kuartal I 2026 dari 10 emiten subsektor farmasi di BEI dengan kode saham DVLA, KAEF, KLBF, MERK, PYFA, SCPI, SIDO, SOHO, TSPC, dan INAF. Data keuangan perusahaan diperoleh dari laporan keuangan triwulan emiten, sedangkan data makroekonomi diperoleh dari laman resmi Bank Indonesia. Variabel prediktor dalam penelitian ini terdiri atas fitur keuangan perusahaan dan fitur makroekonomi. Fitur keuangan perusahaan meliputi pertumbuhan penjualan, pertumbuhan aset, pertumbuhan utang, rasio arus kas operasi terhadap total utang, dan ukuran perusahaan. Sementara itu, fitur makroekonomi meliputi BI-Rate, nilai tukar Rupiah, dan tingkat inflasi.

B. Bentuk Label dan Multi-Horizon Lag

Label financial distress dibentuk menggunakan Altman Z"-Score yang telah dijelaskan pada bagian tinjauan pustaka. Nilai Z"-Score kemudian dikodekan secara biner, yaitu *financial distress* untuk $Z'' \leq 2{,}60$ dan *non-financial distress* untuk $Z'' > 2{,}60$.

Pada skenario *multi-horizon lag*, fitur pada periode t digunakan untuk memprediksi label financial distress pada periode $t+1$, $t+2$, dan $t+4$. Pendekatan ini memungkinkan model memanfaatkan informasi historis sebelum periode target, sehingga prediksi *financial distress* dapat dilakukan secara lebih dini.

C. Validasi dan Pencegahan Kebocoran Data

Validasi model dilakukan menggunakan *time-based holdout* dan *walk-forward validation* karena data penelitian memiliki struktur waktu kuartalan. Pada *time-based holdout*, data periode awal digunakan sebagai data pelatihan, sedangkan data periode akhir digunakan sebagai data pengujian. Skema ini dipilih agar proses evaluasi menyerupai kondisi prediksi aktual, yaitu model dilatih menggunakan data historis untuk memprediksi kondisi pada periode berikutnya.

Walk-forward validation digunakan untuk mengevaluasi kestabilan performa model pada beberapa periode pengujian. Dalam skema ini, model dilatih menggunakan data historis hingga periode tertentu, kemudian diuji pada periode setelahnya secara bertahap mengikuti urutan waktu. Pencegahan kebocoran data dilakukan dengan memastikan bahwa Altman Z"-Score tidak digunakan sebagai fitur prediktor. Selain itu, proses *preprocessing* seperti imputasi missing value dan standardisasi dilakukan berdasarkan data pelatihan pada setiap *fold*, kemudian diterapkan pada data pengujian.

D. Evaluasi Model dan Interpretasi

Model yang digunakan dalam penelitian ini terdiri atas XGBoost, Random Forest, dan Regresi Logistik. XGBoost digunakan sebagai model utama karena mampu menangkap hubungan nonlinier dan interaksi antarfitur pada data tabular. Random Forest digunakan sebagai model pembanding sesama *ensemble* berbasis pohon keputusan, sedangkan Regresi Logistik digunakan sebagai baseline klasifikasi biner.

Evaluasi performa model dilakukan menggunakan *accuracy*, *precision*, *recall*, F1-score, dan ROC-AUC. Accuracy untuk mengukur ketepatan klasifikasi secara keseluruhan, precision digunakan untuk melihat ketepatan prediksi pada kelas financial distress, recall digunakan untuk mengukur kemampuan model dalam mendeteksi observasi yang benar-benar mengalami financial distress, F1-score digunakan untuk menilai keseimbangan antara precision dan recall, sedangkan ROC-AUC digunakan untuk mengukur kemampuan model dalam membedakan kelas *financial distress* dan *non-financial distress*. Selain evaluasi performa, penelitian ini menggunakan SHAP dan *feature importance* untuk menginterpretasikan kontribusi fitur terhadap hasil prediksi model.

IV. HASIL DAN PEMBAHASAN

A. Karakteristik Dataset dan Pembentukan Lag

Label *financial distress* yang dibentuk berdasarkan Altman Z"-Score, dengan komposisi 155 observasi sehat (74,16%) dan 54 observasi distress (25,84%). Distribusi ini menunjukkan adanya *class imbalance*, sehingga model Regresi Logistik dan Random Forest menggunakan class_weight='balanced', sedangkan XGBoost menggunakan scale_pos_weight berdasarkan rasio kelas pada *fold training*.

Tabel 4.1 Ringkasan Dataset, Validasi Data, dan Skenario Lag

Aspek	Ringkasan Temuan
Distribusi label	209 observasi: 155 sehat (74,16%) dan 54 distress (25,84%). INAF dan KAEF selalu distress; PYFA campuran; SOHO mayoritas sehat; emiten lain konsisten sehat.
Kualitas data	Missing value hanya 0,48% pada pertumbuhan aset dan pertumbuhan utang; ditangani dengan median imputation di setiap fold agar tidak terjadi data leakage.
Validasi makro	BI Rate, nilai tukar, dan inflasi konsisten antar emiten pada periode yang sama; tidak ditemukan inkonsistensi input makro.
Lag-1 / t+1	199 observasi digunakan; 10 observasi dibuang secara struktural.
Lag-2 / t+2	189 observasi digunakan; 20 observasi dibuang secara struktural.
Lag-4 / t+4	169 observasi digunakan; 40 observasi dibuang secara struktural.

Pembentukan multi-horizon lag dilakukan dengan menggunakan fitur periode t untuk memprediksi

label distress pada $t+1$, $t+2$, dan $t+4$. Lag fitur perusahaan dibentuk berbasis emiten dan indeks periode, sedangkan lag makroekonomi dibentuk berbasis indeks periode, sehingga informasi masa depan tidak masuk ke data pelatihan.

B. Evaluasi Performa Model

Evaluasi utama dilakukan melalui *time-based holdout*, yaitu data sebelum 2025 sebagai data latih dan Q1 2025-Q1 2026 sebagai data uji. Random Forest menjadi model dengan performa holdout terbaik pada seluruh horizon, sedangkan XGBoost tetap kompetitif dan lebih stabil pada evaluasi *walk-forward*.

Tabel 4.2 Performa Holdout Test Skenario All_Features

Lag	Model	Accuracy	Recall	F1-Score	ROC-AUC
Lag-1	Logistic Regression	89,80%	85,71%	82,76 %	95,31 %
Lag-1	Random Forest	93,88%	92,86%	89,66 %	97,76 %
Lag-1	XGBoost	89,80%	78,57%	81,48 %	97,35 %
Lag-2	Logistic Regression	85,71%	92,86%	78,79 %	95,71 %
Lag-2	Random Forest	91,84%	85,71%	85,71 %	98,37 %
Lag-2	XGBoost	89,80%	85,71%	82,76 %	96,12 %
Lag-4	Logistic Regression	67,35%	100,00 %	63,64 %	92,65 %

Pada Lag-1, Random Forest menghasilkan F1-Score tertinggi sebesar 89,66% dan hanya 1 *false negative*, sehingga paling andal untuk peringatan dini jangka pendek. Pada Lag-2, performa Random Forest tetap kuat dengan F1-Score 85,71% dan ROC-AUC 98,37%. Pada Lag-4, seluruh model mengalami penurunan karena horizon prediksi lebih jauh, tetapi Random Forest masih mempertahankan F1-Score 78,79% dan ROC-AUC 94,69%.

C. Walk-Forward Validation dan Uji Statistik

Walk-forward validation digunakan untuk melihat kestabilan model antarperiode. Pada Lag-1 dan Lag-2 tersedia dua *fold*, sedangkan Lag-4 hanya satu *fold* sehingga interpretasinya perlu hati-hati. XGBoost menunjukkan mean F1 tertinggi dan simpangan baku lebih rendah daripada Random Forest pada Lag-1 dan Lag-2, yang menandakan generalisasi temporal lebih stabil.

Tabel 4.3 Ringkasan Walk-Forward dan McNemar Test

Lag	Mean F1 Walk-Forward	Ringkasan McNemar Holdout
Lag-1	LR 0,275; RF 0,689; XGB 0,718	Seluruh pasangan tidak signifikan; power rendah karena discordant pairs kecil.
Lag-2	LR 0,374; RF 0,700; XGB 0,729	Seluruh pasangan tidak signifikan; power rendah.
Lag-4	LR 0,750; RF 0,786; XGB 0,786	XGB vs LR signifikan (p=0,039); RF vs LR signifikan (p=0,012); XGB vs RF tidak signifikan.

Hasil McNemar menunjukkan bahwa perbedaan antar model pada Lag-1 dan Lag-2 belum signifikan secara statistik, terutama karena jumlah *discordant pairs* kecil. Namun, pada Lag-4, model *ensemble* berbeda signifikan dibanding Regresi Logistik. Dengan demikian, keunggulan Random Forest dan XGBoost secara praktis terlihat kuat, terutama untuk horizon prediksi yang lebih panjang.

D. Skenario Fitur, Feature Importance, dan SHAP

Analisis skenario fitur menunjukkan bahwa fitur keuangan perusahaan jauh lebih informatif dibandingkan fitur makroekonomi. Pada XGBoost Lag-1, skenario Firm_Only bahkan menghasilkan F1-Score lebih tinggi daripada All_Features, sedangkan Macro_Only hanya mencapai ROC-AUC 50,00%, setara prediksi acak.

Tabel 4.4 Ringkasan Interpretasi SHAP XGBoost All_Features

Lag	Fitur Utama	Nilai Utama	Interpretasi
Lag-1	OCF/Total Debt	Mean |SHAP| 1,6558	Fitur paling dominan; makro = 0.
Lag-2	OCF/Total Debt	Mean |SHAP| 1,5757	Dominasi fitur kas operasi tetap konsisten.
Lag-4	OCF/Total Debt	Mean |SHAP| 1,5107	Pertumbuhan utang naik sebagai fitur sekunder; inflasi sangat kecil (0,0067).

Feature importance XGBoost dan SHAP sama-sama menunjukkan bahwa rasio arus kas operasional terhadap total utang merupakan prediktor paling dominan pada seluruh horizon. Secara ekonomis, semakin kuat arus kas operasi relatif terhadap total utang, semakin kecil kecenderungan perusahaan mengalami financial distress. Sebaliknya, BI Rate, nilai tukar, dan inflasi hampir tidak berkontribusi dalam model, sehingga financial distress emiten farmasi pada periode ini lebih dipengaruhi oleh kondisi internal perusahaan daripada variasi makroekonomi.

E. Implikasi, Ilustrasi, dan Keterbatasan

Implikasi praktisnya, investor dan analis dapat memprioritaskan pemantauan rasio arus kas operasional terhadap total utang, pertumbuhan utang, dan pertumbuhan penjualan sebagai indikator risiko. Bagi regulator, model tersebut dapat mendukung sistem early warning berbasis laporan keuangan kuartalan.

Sebagai ilustrasi penerapan, model dapat digunakan dengan memasukkan fitur keuangan dan makroekonomi yang telah tersedia pada periode t, untuk memprediksi potensi financial distress pada periode $t+1$, $t+2$, dan $t+4$. Misalnya, apabila suatu emiten mengalami penurunan rasio arus kas operasional terhadap total utang, peningkatan pertumbuhan utang, dan perlambatan pertumbuhan penjualan, sementara pada saat yang sama terjadi tekanan makroekonomi berupa kenaikan BI-Rate atau pelemahan nilai tukar Rupiah, maka model dapat menghasilkan sinyal risiko distress untuk beberapa kuartal berikutnya. Dengan mekanisme ini, investor, analis, maupun regulator tidak perlu menunggu nilai Altman Z''-Score pada periode target tersedia, tetapi dapat memperoleh indikasi awal berdasarkan pola historis yang telah dipelajari model.

Keterbatasan utama penelitian terletak pada ukuran dataset yang relatif kecil, jumlah *fold walk-forward* yang terbatas, cakupan hanya pada 10 emiten farmasi BEI, serta penggunaan Altman Z"-Score sebagai satu-satunya dasar label distress. Oleh karena itu, penelitian selanjutnya dapat memperluas sektor, menambah periode observasi, menggunakan definisi distress alternatif, dan melakukan optimasi hyperparameter yang lebih sistematis.

V. KESIMPULAN DAN SARAN

Pendekatan *machine learning* berbasis *multi-horizon lag* efektif menjadi sistem deteksi dini *financial distress* emiten farmasi di BEI. Random Forest memberikan akurasi tertinggi pada prediksi jangka pendek (Lag-1) dan panjang (Lag-4), sementara XGBoost unggul dalam stabilitas temporal (*walk-forward*) serta transparansi fitur melalui SHAP. Analisis SHAP konsisten menunjukkan bahwa rasio arus kas operasional terhadap total utang merupakan prediktor paling dominan, menandakan risiko keuangan lebih dipengaruhi oleh faktor internal fundamental perusahaan dibanding variasi makroekonomi.

Investor dan analis disarankan memprioritaskan pemantauan rasio arus kas operasi dan pertumbuhan utang. Manajemen emiten perlu fokus memperkuat arus kas, sementara regulator dapat mengadopsi model ini untuk pengawasan kuartalan. Penelitian selanjutnya disarankan memperluas sektor industri, memperpanjang periode observasi, serta menerapkan kriteria *distress* alternatif di luar Altman Z"-Score.

DAFTAR PUSTAKA

[1] E. I. Altman, “Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy,” *The Journal of Finance*, vol. 23, no. 4, pp. 589–609, 1968.

[2] E. I. Altman, “Predicting Financial Distress of Companies: Revisiting the Z-Score and ZETA Models,” in *Handbook of Research Methods and Applications in Empirical Finance*, pp. 428–456, 2013.

[3] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[4] Y.-P. Huang and M.-F. Yen, "A new perspective of performance comparison among machine learning algorithms for financial distress prediction," Applied Soft Computing, vol. 83, p. 105663, 2019.

[5] M. P. Pantoja-Aguilar, G. de M. Pizano-Ramírez, B. Lerma-Torres, dan M. Á. Zavala-Vargas, “*Testing Altman's Z''-Score to assess the level of accuracy of the model in Mexican companies*,” *Nova Scientia*, vol. 13, no. 27, Nov. 2021.

[6] D. Yan, G. Chi, dan K. K. Lai, *“Financial Distress Prediction and Feature Selection in Multiple Periods by Lassoing Unconstrained Distributed Lag Non-linear Models” Mathematics*, vol. 8, no. 8, hal. 1275, Agu. 2020.

ANALISIS KOMPARATIF CAKUPAN DATA HISTORIS PADA PERAMALAN KURS USD/IDR MENGGUNAKAN METODE EXPONENTIAL SMOOTHING (ETS)

Azizah Indriani Putri [1], Baiq Mutia Dewi Edelweis [2], Syazwani[3], Herliana Rosika[4]
Program Studi Teknik Informatika Universitas Mataram, 2) Program Studi Teknik Informatika Universitas Mataram, 3) Program Studi Teknik Informatika Universitas Mataram, 4) Program Studi Teknik Informatika Universitas Mataram
e-mail: f1d02310041@student.unram.ac.id, f1d02310107@student.unram.ac.id, f1d02310140@student.unram.ac.id, herliana2014@staff.unram.ac.id

Abstrak

Pergerakan nilai tukar mata uang asing, khususnya kurs Dolar Amerika terhadap Rupiah (USD/IDR), memiliki karakteristik runtun waktu finansial yang fluktuatif dan sensitif terhadap kejut ekonomi. Dalam pemodelan statistik konvensional seperti metode *Exponential Smoothing* (ETS), penentuan cakupan data historis (data horizon) menjadi faktor krusial yang memicu dilema antara pemanfaatan volume data jangka menengah atau relevansi data kontemporer jangka pendek. Penelitian ini bertujuan untuk melakukan analisis komparatif terhadap pengaruh panjang data historis dalam memprediksi kurs USD/IDR serta mensimulasikan hasil peramalan untuk 30 hari ke depan. Penelitian kuantitatif eksperimental ini mengevaluasi dua skenario data horizon dari API Yahoo Finance, yaitu skenario jangka pendek kontemporer (Januari – Mei 2026) dan skenario jangka menengah (2021 – 2026). Pengujian stasioneritas dilakukan melalui uji *Augmented Dickey-Fuller* (ADF), dilanjutkan proses *differencing*, pembagian dataset (80% *training*, 20% *testing*), dan pemodelan menggunakan ETS dengan komponen tren linear (*Holt's method*). Hasil eksperimen menunjukkan bahwa skenario data horizon jangka pendek menghasilkan akurasi terbaik dengan MAPE sebesar 1,21% dan RMSE sebesar Rp259, sedangkan skenario jangka menengah menghasilkan MAPE sebesar 16,68% dan RMSE sebesar Rp 2.930. Uji Diebold-Mariano mengonfirmasi bahwa perbedaan akurasi ini signifikan secara statistik (DM = -49,57, $p < 0,01$). Simulasi proyeksi menggunakan model terbaik menunjukkan bahwa kurs USD/IDR untuk 30 hari ke depan diperkirakan akan bergerak dengan tren meningkat hingga Rp18.175.
Kata kunci: ETS, Kurs USD/IDR, Data Horizon, Peramalan, Komparasi Akurasi.

Abstract

Movements in foreign exchange rates, particularly the U.S. dollar against the Indonesian rupiah (USD/IDR), exhibit the characteristics of a financial time series that is volatile and sensitive to economic shocks. In conventional statistical modeling such as the Exponential Smoothing (ETS) method, determining the historical data scope (data horizon) is a crucial factor that creates a dilemma between utilizing long-term data volume or the relevance of short-term contemporary data. This study aims to conduct a comparative analysis of the impact of historical data length on predicting the USD/IDR exchange rate and to simulate forecast results for the next 30 days. This experimental quantitative study evaluates two data horizon scenarios from the Yahoo Finance API: a short-term contemporary scenario (January–May 2026) and a medium-term scenario (2021–2026). Stationarity testing was performed using the Augmented Dickey-Fuller (ADF) test, followed by differencing, dataset splitting (80% training, 20% testing), and modeling using the ETS with a linear trend component (Holt's method). The experimental results show that the short-term data horizon scenario yields the best accuracy with a MAPE of 1.21% and an RMSE of Rp259, while the medium-term scenario yields a MAPE of 16,68% and an RMSE of Rp2.930. The Diebold-Mariano test confirmed that this difference in accuracy is statistically significant (DM = -49.57, $p < 0.01$). Projection simulations using the best model indicate that the USD/IDR exchange rate is expected to trend upward over the next 30 days, reaching Rp18,175.
***Keywords**: ETS, USD/IDR Exchange Rate, Data Horizon, Forecasting, Accuracy Comparison.*

I. PENDAHULUAN

Perubahan nilai tukar mata uang, khususnya nilai tukar antara USD (Dolar Amerika Serikat) dan IDR (Rupiah Indonesia), menjadi isu yang sangat penting dalam perekonomian global dan nasional [1]. Nilai tukar mata uang asing, khususnya Dolar Amerika Serikat terhadap Rupiah (USD/IDR), memegang peranan yang sangat krusial dalam stabilitas makroekonomi Indonesia. Fluktuasi nilai tukar mata uang memiliki peran penting dalam perekonomian suatu negara, terutama bagi negara berkembang seperti Indonesia. Kurs USD/IDR, sebagai representasi

hubungan nilai tukar antara Dolar Amerika Serikat (USD) dan Rupiah Indonesia (IDR), menjadi salah satu indikator utama yang mencerminkan stabilitas ekonomi, daya saing perdagangan internasional, dan sentimen pasar global. Perubahan nilai tukar dapat berdampak langsung pada berbagai sektor, seperti perdagangan internasional, investasi asing, dan inflasi domestik. Berdasarkan data Yahoo Finance yang digunakan dalam penelitian ini, periode waktu Januari–Mei 2026 memiliki rentang nilai kurs antara Rp16.668 hingga Rp17.840. Sementara itu, periode 2021–2026 memiliki rentang nilai yang lebih luas, yaitu Rp13.828 hingga Rp17.840. Perbedaan rentang nilai tersebut menunjukkan adanya variasi karakteristik data yang berpotensi memengaruhi hasil peramalan [2].

Dinamika pergerakan kurs yang sangat fluktuatif dan dipengaruhi oleh sentimen global terkini seperti kebijakan suku bunga bank sentral dunia hingga ketegangan geopolitik menuntut adanya sebuah sistem proyeksi nilai tukar yang presisi. Peramalan kurs sangat penting karena membantu pemerintah dalam pengambilan keputusan yang tepat terkait kebijakan ekonomi, terutama dalam konteks ekspor-impor. Peramalan yang akurat dapat meningkatkan efektivitas rencana bisnis dan mendukung pertumbuhan ekonomi dengan memprediksi perubahan nilai tukar yang mempengaruhi harga barang dan komoditas selain itu, pemahaman tentang fluktuasi kurs juga penting untuk stabilitas ekonomi domestik dan internasional [3].

Salah satu teknik peramalan yang seringkali digunakan yakni dengan menggunakan analisis runtun waktu. Analisis runtun waktu didefinisikan sebagai pola pergerakan data dengan interval waktu yang sama [4]. Metode ETS merupakan pendekatan analisis runtuh waktu yang fleksibel untuk menangkap komponen *error*, *trend*, dan musiman dalam data *time series* [5]. *Holt's Linear Trend method* (ETS (A, A, N)) adalah varian ETS yang mampu menangkap tren linear dalam data, sehingga cocok untuk data yang menunjukkan pola peningkatan atau penurunan yang konsisten [5]. Fenomena ini menimbulkan sebuah dilema metodologis yang krusial dalam prediksi indikator ekonomi makro seperti kurs USD/IDR. Di satu sisi, pemanfaatan data historis jangka menengah seperti data runtun waktu dari tahun 2021 hingga 2026, menguntungkan bagi model statistik karena mampu merekam berbagai siklus fluktuasi ekonomi makro serta peristiwa ekstrem (*outliers*) masa lalu, seperti krisis moneter masa pandemi global. Namun, cakupan data yang terlalu panjang berisiko menimbulkan distorsi [6], di mana model menjadi kurang sensitif terhadap volatilitas dan tren linier kontemporer karena terbebani oleh pola masa lalu yang sudah tidak relevan dengan kondisi pasar saat ini.

Pendekatan alternatif dapat dilakukan dengan membatasi cakupan data pada rentang jangka pendek kontemporer, yaitu data tahun berjalan dari Januari hingga Mei 2026. Pemanfaatan data jangka pendek ini diasumsikan memiliki keunggulan dalam menangkap kondisi riil, sentimen pasar terbaru, serta kebijakan moneter paling aktual (*contemporary trend*) [7]. Skenario jangka pendek ini dihadapkan pada keterbatasan jumlah sampel data yang relatif kecil, yang secara statistik berisiko menyebabkan *overfitting* atau ketidakmampuan model dalam melakukan generalisasi prediksi jangka menengah. Oleh karena itu, melalui pengujian berbasis kode Python dalam penelitian ini, kedua skenario data horizon tersebut disimulasikan secara terpisah menggunakan model *Exponential Smoothing* (ETS) dengan komponen tren linear (*Holt's method*). Pengujian ini dilakukan untuk mengevaluasi secara empiris apakah volume data masa lalu yang besar ataukah aktualitas data jangka pendek yang memberikan tingkat presisi dan akurasi peramalan tertinggi berdasarkan metrik kesalahan *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Square Error* (RMSE).

II. TINJAUAN PUSTAKA

1. Nilai Tukar (Kurs) USD/IDR

Nilai tukar adalah harga mata uang asing yang dinyatakan dalam mata uang domestik dan memainkan peran sentral dalam perdagangan internasional [2]. Nilai tukar memungkinkan perbandingan harga barang dan jasa di seluruh dunia. Perdagangan internasional menghubungkan berbagai negara dan menjadikannya lebih terintegrasi [2]. Perekonomian Indonesia banyak dipengaruhi perekonomian internasional sehingga nilai tukar rupiah sanagat dibutuhkan oleh masyarakat dalam kehidupan perekonomiannya. Perekonomian Indonesia dipengaruhi juga kondisi perekonomian dari negara lain, salah satunya

adalah perekonomian Amerika dimana Indonesia masih memiliki hutang internasional. Ketika terjadi penurunan nilai rupiah terhadap dolar Amerika akan mempengaruhi jumlah hutang luar negeri yang harus dibayarkan [8].

2. Konsep Cakupan Data Historis (*Data Horizon*) dalam Peramalan

Peramalan adalah kegiatan mengestimasi apa yang akan terjadi pada masa yang akan datang. Peramalan diperlukan karena adanya perbedaan kesenjangan waktu antara kesadaran akan dibutuhkannya suatu kebijakan baru dengan waktu pelaksanaan kebijakan tersebut [9].

Berdasarkan horizon waktu, peramalan dapat dibagi menjadi tiga jenis yaitu:

a. Peramalan jangka panjang, yaitu mencakup waktu lebih besar dari 18 bulan. Misalnya untuk Memprediksi tren makro beberapa bulan atau tahun ke depan. Namun, kelemahannya adalah data masa lalu yang terlalu jauh sering kali mengandung informasi yang sudah tidak relevan (*outdated*) dengan dinamika pasar saat ini, sehingga berisiko menciptakan distorsi estimasi parameter model.
b. Peramalan jangka menengah, yaitu mencakup waktu antara 3 sampai 18 bulan. Misalnya perencanaan produksi musiman
c. Peramalan Jangka Pendek, yaitu mencakup jangka waktu kurang dari 3 bulan. Misalnya untuk memprediksi nilai kurs untuk 30 hari ke depan. Namun, keterbatasan ukuran sampel (n kecil) dapat memicu risiko ketidakstabilan model (*overfitting*) dan kurangnya informasi untuk melakukan generalisasi prediksi ke depan.

3. Stasioneritas Data

Suatu data time series dikatakan stasioner apabila mean dan variansnya tetap sepanjang waktu [10]. Berikut merupakan proses stasioner

$$E\,(Zt) = \mu\ (2)$$

$$Var\,(Zt) = E(Zt - \mu)\,2 = \gamma o\ (2)\ (3)$$

dengan μ untuk semua lag *k* adalah konstan. Uji akar unit *Dickey-Fuller* (DF) merupakan salah satu metode untuk menguji stasioneritas data dalam mean [11]. Uji akar unit mengaplikasikan model regresi yang dituliskan pada Persamaan (4).

$$\Delta Z_t = (\rho - 1) Z_{t-1} + \varepsilon_t$$

$$\Delta Z_t = \delta Z_{t-1} + \varepsilon_t$$

Zt dikatakan tidak stasioner (memiliki akar unit) apabila δ = 0 atau ρ = 1. Hipotesis untuk uji akar unit Dickey-Fuller (DF) adalah: H0: δ = 0 (Data tidak stasioner dalam mean) H1: δ < 0 (Data stasioner dalam mean) Statistik uji untuk uji DF dinyatakan oleh Persamaan (5).

$$\tau = \frac{\hat{\delta}}{SE(\hat{\delta})}$$

dengan δ: penduga dari δ, dan SE(δ): standard error dari δ.

Kriteria uji: Ho ditolak jika nilai τhitung < $\tau(\alpha;n-1)$ atau p − value < α.

4. Metode *Exponential Smoothing* (ETS)

Exponential Smoothing (ETS) merupakan metode peramalan runtun waktu yang memberikan bobot eksponensial terhadap data masa lalu, di mana data yang lebih baru diberi bobot lebih besar dibandingkan data yang lebih lama [5]. Metode ETS dikelompokkan berdasarkan tiga komponen utama: *error* (E), *trend* (T), dan *seasonal* (S). Untuk data yang menunjukkan pola peningkatan atau penurunan konsisten tanpa pola musiman, digunakan model *Holt's Linear Trend* atau dinotasikan sebagai ETS(A,A,N) komponen *error* aditif, *trend* aditif, dan tanpa musiman [5].

5. Metrik Evaluasi Akurasi

a. MAPE

Mean Absolute Percentage Error (MAPE) merupakan metrik evaluasi yang umum digunakan dalam bidang statistik dan machine learning untuk mengukur tingkat kesalahan prediksi pada model regresi. MAPE menghitung rata-rata persentase selisih antara nilai prediksi dan nilai sebenarnya, sehingga hasilnya dinyatakan dalam bentuk persentase. MAPE cocok digunakan ketika tujuan analisis adalah mengukur tingkat kesalahan relatif dalam bentuk persentase, bukan selisih absolut [12].

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right|$$

b. RMSE

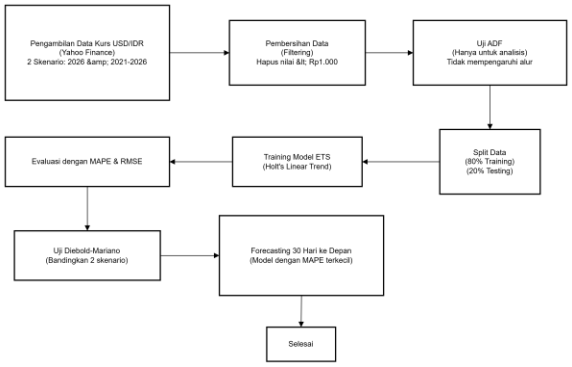
Root Mean Square Error (RMSE) adalah akar dari rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi. RMSE mengukur besarnya kesalahan prediksi dalam satuan yang sama dengan data asli, sehingga mudah

diinterpretasikan. RMSE lebih sensitif terhadap *outlier* dibandingkan MAPE [13].

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}$$

III. METODOLOGI PENELITIAN

Gambar 1 memperlihatkan alur penelitian secara keseluruhan, mulai dari pengambilan data, pemodelan, evaluasi, hingga peramalan.



Gambar 1 Diagram Alur Penelitian

Berdasarkan diagram alir penelitian pada Gambar 1, berikut adalah penjelasan rinci dari setiap tahapan prosedur penelitian yang dilakukan:

1. Pengambilan Data Kurs

Data yang digunakan merupakan data sekunder harga penutupan harian (*daily closing price*) kurs USD/IDR. Data diunduh secara otomatis memanfaatkan *Application Programming Interface* (API) Yahoo Finance melalui pustaka yfinance di Python. Pada tahap ini, data langsung dibagi menjadi dua variabel horizon waktu yang kontras, yaitu data kontemporer tahun berjalan 2026 dan data historis makro jangka menengah (2021–2026).

2. Pembersihan Data (*Data Preprocessing*):

Untuk menjaga kualitas data dari adanya galat sistem (*data glitch*) atau anomali pencatatan dari sumber, dilakukan penyaringan (*filtering*). Algoritma pengondisian diterapkan untuk menghapus baris data jika nilai kurs tercatat di bawah Rp1.000, karena nilai tersebut tidak logis untuk rentang waktu pengamatan.

3. Uji Stasioneritas Data

Data kurs yang telah bersih kemudian diuji tingkat stasioneritasnya secara statistik menggunakan uji *Augmented Dickey-Fuller* (ADF) melalui fungsi adfuller(). Stasioneritas tercapai apabila nilai *p-value* yang dihasilkan $\le 0,05$.

4. Pembagian Dataset (*Data Splitting*)

Sebelum masuk ke fase pelatihan model, masing-masing dataset skenario dipotong secara kronologis (berdasarkan urutan waktu) dengan proporsi 80% dialokasikan sebagai data latih (training data) dan 20% sisanya dialokasikan sebagai data uji (testing data).

5. Pelatihan Model ETS

Data latih (80%) dimasukkan ke dalam model *Exponential Smoothing* (ETS) dengan komponen tren linear (Holt's method) untuk melatih model (*fitting process*). Proses ini menghasilkan dua model terpisah: model berbasis data jangka pendek (2026) dan model berbasis data jangka menengah (2021-2026).

6. Evaluasi Kinerja Komparatif

Model yang telah terlatih diuji kemampuannya untuk memprediksi data uji (20%). Hasil prediksi tersebut kemudian diadu dengan data aktual di dunia nyata menggunakan metrik evaluasi kesalahan *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Square Error* (RMSE).

7. Uji Diebold-Mariano

Uji Diebold-Mariano (DM) digunakan untuk menguji apakah perbedaan akurasi antara dua model peramalan signifikan secara statistik [14]. Hipotesis yang diuji adalah H_0: kedua model memiliki akurasi yang sama, melawan H_1: model pertama lebih akurat daripada model kedua. Statistik uji DM dihitung berdasarkan selisih kuadrat error kedua model, dengan rumus:

$$DM = \frac{\bar{d}}{\sqrt{Var(\bar{d})}}$$

di mana $\bar{d}$ adalah rata-rata dari selisih loss function ($d_t = e_{1t}^2 - e_{2t}^2$), dan $Var(\bar{d})$ adalah estimasi varians yang dikoreksi dengan autokorelasi menggunakan metode Newey-West. Statistik DM berdistribusi normal standar. Jika nilai DM negatif dan p-value < 0,05, maka model pertama signifikan lebih akurat [14].

8. Simulasi Peramalan (*Forecasting*)

Setelah mendapatkan evaluasi performa dari kedua skenario cakupan data historis, model terbaik dengan tingkat kesalahan (MAPE/RMSE) terkecil digunakan untuk melakukan simulasi proyeksi arah pergerakan kurs USD/IDR untuk 30 hari ke depan.

IV. HASIL DAN PEMBAHASAN

1. Analisis Eksperimen Skenario Jangka Pendek

Pengujian skenario pertama menggunakan data kontemporer harian kurs USD/IDR dari 1 Januari hingga 31 Mei 2026. Kurs USD/IDR memiliki kecenderungan meningkat sepanjang periode Januari–Mei 2026. Nilai kurs bergerak dari kisaran Rp16.700 pada awal periode menjadi sekitar Rp17.800 pada akhir periode. Kenaikan tersebut mengindikasikan pelemahan nilai Rupiah terhadap Dolar Amerika Serikat. Meskipun terdapat beberapa fluktuasi jangka pendek, pola pergerakan secara umum menunjukkan tren naik yang relatif konsisten. Kondisi ini mengindikasikan adanya komponen tren yang cukup kuat sehingga berpotensi dimodelkan menggunakan metode Holt's Linear Trend (ETS(A,A,N)).

Hasil evaluasi model ETS menunjukkan nilai MAPE sebesar 1,21% dan RMSE sebesar Rp259. Secara matematis, nilai MAPE dihitung menggunakan rata-rata persentase kesalahan absolut seluruh data pengujian, sedangkan RMSE dihitung menggunakan akar dari rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi. Berdasarkan klasifikasi akurasi peramalan, nilai MAPE di bawah 10% menunjukkan bahwa model memiliki tingkat akurasi yang sangat baik dalam memprediksi pergerakan kurs USD/IDR. Selain itu, nilai RMSE sebesar Rp259 menunjukkan bahwa rata-rata deviasi hasil prediksi terhadap nilai aktual berada pada kisaran Rp259. Temuan ini mengindikasikan bahwa data kontemporer mampu memberikan informasi yang cukup untuk menangkap arah pergerakan kurs dalam jangka pendek.

2. Analisis Eksperimen Skenario Jangka Menengah (2021-2026)

Skenario kedua menggunakan data historis yang lebih panjang, yaitu periode 1 Januari 2021 hingga 31 Mei 2026 dengan total 1.406 observasi. Dataset ini mencakup berbagai kondisi ekonomi yang berbeda, mulai dari fase pemulihan pasca-pandemi COVID-19, periode lonjakan inflasi global, hingga fase pengetatan kebijakan moneter yang dilakukan berbagai bank sentral dunia.

Hasil evaluasi menunjukkan bahwa model ETS menghasilkan nilai MAPE sebesar 16,68% dan RMSE sebesar Rp2.930. Dibandingkan dengan skenario horizon kontemporer, terjadi penurunan performa yang cukup signifikan. Meskipun nilai MAPE tersebut masih berada pada kategori cukup akurat (10–20%), tingkat kesalahannya jauh lebih besar dibandingkan model yang hanya menggunakan data tahun 2026.

Penurunan akurasi ini mengindikasikan bahwa penambahan data historis yang lebih panjang tidak selalu meningkatkan performa model peramalan. Salah satu penyebab yang mungkin adalah adanya structural break atau perubahan struktur pasar selama periode pengamatan. Data tahun 2021–2023 dipengaruhi oleh dampak pemulihan ekonomi pasca-pandemi dan volatilitas global yang tinggi, sedangkan data tahun 2024–2026 menunjukkan pola pergerakan kecenderungan tren yang lebih stabil.

Akibatnya, model ETS harus menyesuaikan parameter terhadap berbagai pola yang berbeda dalam satu rentang waktu yang panjang. Kondisi ini menyebabkan model menjadi kurang sensitif terhadap tren terkini yang terjadi pada tahun 2026. Dengan kata lain, informasi historis yang terlalu lama berpotensi mengurangi kemampuan model dalam menangkap dinamika pasar yang lebih relevan dengan kondisi saat ini.

3. Analisis Komparatif Performa Model

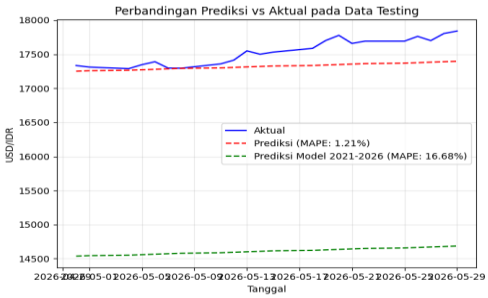
Rekapitulasi perbandingan performa akurasi berdasarkan kedua skenario *data horizon* disajikan pada Tabel 4.2.

Tabel 1. Perbandingan Akurasi Model ETS Berdasarkan Horizon Data

Metrik	Data 2026 (Jan-Mei)	Data 2021 – 2026
Jumlah Data (hari)	106	1.406
Rentang Nilai (Rp)	16.668 - 17.840	13.828 – 17.840
MAPE (%)	1,21%	16.68%
RMSE (Rp)	Rp259	Rp2.930
Kategori	Sangat Akurat	Cukup Akurat

Berdasarkan Tabel 1, terlihat bahwa model dengan data horizon jangka pendek (2026) jauh lebih unggul dibandingkan model dengan data historis 5 tahun. Perbedaan akurasi ini sangat kontras: MAPE model 2026 hanya 1,21%, sementara model 2021-2026 mencapai 16,68%.

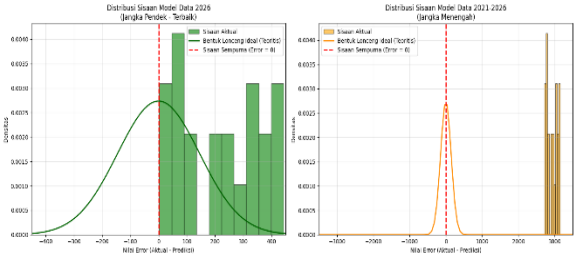
Sebagai bagian dari evaluasi model, dilakukan perbandingan antara nilai aktual dengan hasil prediksi pada data pengujian. Hasil perbandingan tersebut disajikan pada Gambar 2.



Gambar 2 Perbandingan Prediksi vs Aktual

Perbedaan akurasi yang signifikan ini disebabkan oleh dua faktor utama. Pertama, periode 2021–2023 didominasi oleh pemulihan pasca-pandemi dan lonjakan inflasi global yang menyebabkan volatilitas ekstrem, sementara periode 2024–2026 menunjukkan struktur pasar yang lebih stabil dengan tren kenaikan bertahap. Kedua, terjadi *structural break* pada awal 2024 akibat perubahan kebijakan suku bunga The Fed dan stabilitas politik pasca-pemilu, sehingga data sebelum periode tersebut mengandung pola yang sudah tidak relevan dengan kondisi pasar saat ini. Akibatnya, model yang dilatih dengan data jangka menengah (2021–2026) menjadi kurang responsif terhadap tren kontemporer.

Selain menggunakan metrik MAPE dan RMSE, kualitas model juga dianalisis melalui distribusi sisaan. Analisis ini bertujuan untuk melihat sebaran kesalahan prediksi pada masing-masing model sebagai pendukung hasil evaluasi kuantitatif.



Gambar 3 Distribusi Sisaan Model ETS

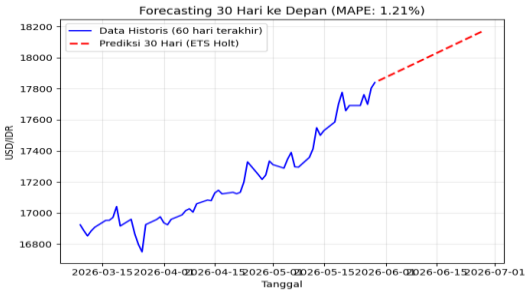
Berdasarkan Gambar 3, model dengan horizon kontemporer tahun 2026 menunjukkan distribusi sisaan pada rentang error yang lebih kecil dibandingkan model dengan horizon historis 2021–2026. Sebaliknya, model horizon historis memiliki sebaran error yang lebih besar, sehingga mengindikasikan tingkat kesalahan prediksi yang lebih tinggi. Pola distribusi sisaan tersebut mendukung hasil evaluasi menggunakan MAPE dan RMSE, di mana model horizon kontemporer memperoleh nilai MAPE sebesar 1,21% dan RMSE sebesar Rp259, sedangkan model horizon historis memperoleh MAPE sebesar 16,68% dan RMSE sebesar Rp2.930.

4. Uji Diebold-Mariano

Hasil uji Diebold-Mariano pada data *testing* yang sama (20% terakhir dari periode 2026) menunjukkan nilai DM statistik sebesar -49,57 dengan p-value < 0,001. Karena p-value < 0,05 dan DM bernilai negatif, maka hipotesis nol ditolak. Dengan demikian, model dengan data horizon jangka pendek (2026) secara signifikan lebih akurat secara statistik dibandingkan model dengan data jangka menengah (2021–2026) pada tingkat kepercayaan 95% [14].

5. Hasil Simulasi Peramalan 30 Hari ke Depan

Mengacu pada keunggulan akurasi skenario jangka pendek (MAPE 1,21% dan terbukti signifikan secara statistik), model ETS (Holt's Linear Trend) berbasis data kontemporer 2026 dipilih untuk melakukan simulasi proyeksi pergerakan kurs USD/IDR untuk 30 hari ke depan (Juni 2026).



Gambar 4 Forecasting 30 Hari ke Depan

Grafik pada Gambar 4 menunjukkan bahwa model ETS (Holt's Linear Trend) menghasilkan proyeksi dengan tren meningkat secara linear. Nilai prediksi pada akhir periode 30 hari (Juni 2026) mencapai Rp18.175, naik sebesar +335 poin (+1,88%) dari nilai terakhir data historis Rp17.840. Pola ini mencerminkan kemampuan model ETS dalam menangkap tren jangka pendek yang kontemporer.

V. KESIMPULAN DAN SARAN

Berdasarkan eksperimen peramalan kurs USD/IDR dengan metode ETS (Holt's method), disimpulkan bahwa horizon data jangka pendek (Jan–Mei 2026) menghasilkan akurasi terbaik (MAPE 1,21%; RMSE Rp259), jauh lebih unggul dibandingkan data jangka menengah 2021–2026 (MAPE 16,68%; RMSE Rp2.930). Uji Diebold-Mariano menunjukkan perbedaan ini signifikan

secara statistik (DM = -49,57; p < 0,01), membuktikan bahwa data historis yang terlalu panjang mengandung noise dan struktur ekonomi yang tidak relevan. Proyeksi 30 hari ke depan dengan model terbaik menunjukkan tren meningkat hingga Rp18.175.

DAFTAR PUSTAKA

[1] A. B. Santoso, *Ekonomi Internasional*. Semarang, Indonesia: Badan Penerbit Universitas Stikubank Semarang, 2020.

[2] S. A. Pakasi, T. O. Rotinsulu, dan M. T. B. Maramis, "Analisis Fundamental Fluktuasi Kurs USD/IDR 2009–2023," *Jurnal EMBA: Jurnal Riset Ekonomi, Manajemen, Bisnis dan Akuntansi*, vol. 13, no. 1, pp. 709–720, Jan. 2025, doi: 10.35794/emba.v13i01.60569.

[3] Y. Sari, E. Winarni, dan Febriani, "Peramalan Kurs Indonesia Menggunakan Model ARIMA (Autoregressive Integrated Moving Average)," *Jurnal Development*, vol. 12, no. 2, pp. 31–38, Des. 2024.

[4] R. H. Shumway and D. S. Stoffer, Time Series Analysis and Its Applications. New York, NY: Springer New York, 2011. doi: 10.1007/978-1-4419-7865-3.

[5] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed. Melbourne, Australia: OTexts, 2021.

[6] A. Abdullah, H. Kuswanto, D. D. Prastyo, and Suhartono, "Modeling the volatility of gross domestic product based on the volatility of money supply, inflation, interest rate, and exchange rate using the fractional cointegration model," *Journal of Physics: Conference Series*, vol. 1538, p. 012052, 2020. doi: 10.1088/1742-6596/1538/1/012052.

[7] P. Fariska, Nugraha, dan M. M. A. Rohandi, "Hubungan Sentimen Investor, Volume Perdagangan dan Kebijakan Moneter Pada Perkembangan Pasar Modal di Indonesia," Jurnal Manajemen dan Bisnis: Performa, vol. 17, no. 1, pp. Mar. 2020.

[8] I. H. Susilowati dan Rosento, "Peramalan Nilai Tukar Kurs IDR Terhadap Dollar USD Dengan Metode Moving Average dan Exponential Smoothing," *Perspektif: Jurnal Ekonomi & Manajemen Universitas Bina Sarana Informatika*, vol. 18, no. 1, pp. 87–94, Mar. 2020, doi: 10.31294/jp.v18i1.7837.

[9] A. Yuliana, A. Rizkiani, A. Bashyar, Chotibin, N. A. Jamil, dan I. Yuliani, *Metode Peramalan (Forecasting)*. Purwakarta, Indonesia: Program Studi Manajemen, STIE Dr. Khez. Muttaqien Purwakarta, 2019.

[10] G. E. P. Box, G. M. Jenkins, G. C. Reinsel and G. M. Ljung, Time Series Analysis Forecasting and Control Fifth Edition. John Wiley & Sons. 2016.

[11] Suparti and R. Santoso, "Analisis Data Time Series Menggunakan Model Kernel: Pemodelan Data Harga Saham MDKA," *Indonesian Journal of Applied Statistics*, vol. 6, no. 1, pp. 22–32, May 2023. doi: 10.13057/ijas.v6i1.79385.

[12] H. H. Nuha, "Mean Absolute Percentage Error (MAPE) dan Penggunaannya," *Jurnal Pemanfaatan Teknologi untuk Masyarakat (JAPATUM)*, vol. 3, no. 4, pp. 26–29, Dec. 2024, doi: 10.59328/JAPATUM.2024.3.4.107.

[13] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014, doi: 10.5194/gmd-7-1247-2014.

[14] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014, doi: 10.5194/gmd-7-1247-2014.

PEMODELAN DAN PREDIKSI POPULASI TERNAK UNGGAS DI PROVINSI NTB MENGGUNAKAN METODE REGRESI LINEAR DAN CARLO MONTE

Baiq Kustina Amriana [1)], Alifah Rizki Saputri [2)], Zahratu Syita [3)], Herliana Rosika [4)]
Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: f1d02310043@student.unram.ac.id, f1d02310103@student.unram.ac.id, f1d02310148@student.unram.ac.id, herliana2014@staff.unram.ac.id

Abstrak

Populasi unggas merupakan salah satu indikator penting dalam sektor peternakan karena berperan dalam mendukung ketahanan pangan dan perekonomian daerah. Penelitian ini bertujuan untuk memprediksi populasi unggas di Provinsi Nusa Tenggara Barat (NTB) menggunakan metode Regresi Linear dan simulasi Monte Carlo. Data yang digunakan merupakan data populasi unggas Provinsi NTB periode 2016–2025 yang diperoleh dari publikasi resmi Dinas Peternakan dan Kesehatan Hewan Provinsi NTB. Populasi unggas dihitung berdasarkan akumulasi beberapa jenis unggas, yaitu ayam ras pedaging, ayam ras petelur, ayam buras, dan itik. Metode Regresi Linear digunakan untuk membangun model prediksi berdasarkan tren historis populasi unggas, sedangkan simulasi Monte Carlo digunakan untuk menggambarkan kemungkinan variasi hasil prediksi pada masa mendatang. Berdasarkan hasil pemodelan, total populasi unggas di Provinsi NTB diprediksi terus mengalami peningkatan selama periode 2026–2030. Hasil prediksi menunjukkan populasi unggas mencapai 59.134.154 ekor pada tahun 2026, meningkat menjadi 63.533.750 ekor pada tahun 2027, 67.933.342 ekor pada tahun 2028, 72.332.937 ekor pada tahun 2029, dan mencapai 76.732.531 ekor pada tahun 2030. Selama periode prediksi 2026–2030, populasi unggas didominasi oleh ayam ras pedaging yang meningkat dari 48.175.589 ekor menjadi 65.552.018 ekor, diikuti ayam buras sebesar 6.552.830-5.650.220 ekor, ayam ras petelur sebesar 4.049.735-5.501.982 ekor, dan itik sebesar 356.000-28.311 ekor sebagai populasi terkecil.

Kata kunci: Monte Carlo; NTB; Populasi Unggas; Regresi Time series; Simulasi

Abstract

The poultry population is an important indicator in the livestock sector because it plays a significant role in supporting food security and regional economic development. This study aims to predict the poultry population in West Nusa Tenggara (NTB) Province using Linear Regression and Monte Carlo simulation methods. The data used in this study consist of poultry population data for NTB Province from 2016 to 2025, obtained from official publications of the West Nusa Tenggara Provincial Livestock and Animal Health Service. The poultry population was calculated as the accumulation of several poultry types, namely broiler chickens, layer chickens, native chickens, and ducks. Linear Regression was employed to develop a prediction model based on historical poultry population trends, while Monte Carlo simulation was used to represent possible variations in future prediction results. Based on the modeling results, the total poultry population in NTB Province is predicted to continue increasing during the 2026–2030 period. The forecasted poultry population reaches 59,134,154 birds in 2026, increasing to 63,533,750 birds in 2027, 67,933,342 birds in 2028, 72,332,937 birds in 2029, and 76,732,531 birds in 2030. During the prediction period, the poultry population is dominated by broiler chickens, which are projected to increase from 48,175,589 birds to 65,552,018 birds, followed by native chickens ranging from 6,552,830 to 5,650,220 birds, layer chickens ranging from 4,049,735 to 5,501,982 birds, and ducks ranging from 356,000 to 28,311 birds as the smallest population group.

Keyword: *Monte Carlo; NTB; poultry population; simulation; time series regression*

I. PENDAHULUAN

Sektor peternakan merupakan salah satu bagian penting dalam pembangunan pertanian karena berperan sebagai penyedia sumber protein hewani bagi masyarakat. Bertambahnya jumlah penduduk dan meningkatnya kesadaran masyarakat akan pentingnya gizi menyebabkan kebutuhan terhadap produk peternakan terus meningkat setiap tahun. Kondisi tersebut menjadikan

subsektor peternakan memiliki peran strategis dalam mendukung ketahanan pangan serta peningkatan kualitas sumber daya manusia [1], serta menjadi salah satu sektor strategis dalam pembangunan ekonomi dan penyediaan pangan nasional [2].

Salah satu komoditas peternakan yang memiliki kontribusi besar dalam pemenuhan kebutuhan protein hewani adalah unggas, terutama ayam pedaging dan ayam petelur. Produk unggas banyak diminati karena memiliki kandungan protein yang tinggi, harga yang relatif terjangkau, serta mudah diperoleh oleh masyarakat. Selain itu, meningkatnya konsumsi produk unggas menunjukkan bahwa unggas memiliki peran penting dalam mendukung ketersediaan pangan dan pemenuhan gizi masyarakat [1][3]. Oleh karena itu, pemantauan dan prediksi perkembangan populasi unggas menjadi penting untuk mendukung perencanaan produksi serta menjaga ketersediaan pasokan di masa mendatang [4][5].

Perkembangan sektor unggas tidak hanya ditentukan oleh tingkat produksi, tetapi juga dipengaruhi oleh dinamika populasi ternak yang tersedia. Berdasarkan data Dinas Peternakan dan Kesehatan Hewan Provinsi Nusa Tenggara Barat (DPKH NTB), total populasi ternak unggas di Provinsi NTB mengalami perubahan yang cukup signifikan selama periode 2016–2025, yaitu meningkat dari 17.275.409 ekor pada tahun 2016 menjadi 49.389.281 ekor pada tahun 2025. Meskipun demikian, peningkatan tersebut tidak terjadi secara konsisten karena terjadi fluktuasi pada beberapa tahun, seperti penurunan populasi pada tahun 2020, 2022, dan 2024. Kondisi tersebut menunjukkan bahwa perkembangan populasi unggas dipengaruhi oleh berbagai faktor sehingga diperlukan suatu model prediksi yang mampu menggambarkan kecenderungan populasi pada periode mendatang sebagai dasar dalam penyusunan kebijakan, perencanaan produksi, dan pengelolaan sektor peternakan. Hal ini sejalan dengan penelitian Kulyk dkk. yang menunjukkan bahwa analisis deret waktu mampu menggambarkan tren perkembangan sektor peternakan unggas dan menghasilkan prediksi yang bermanfaat dalam mendukung pengambilan keputusan [5].

Berbagai penelitian sebelumnya telah menerapkan metode prediksi pada sektor peternakan dengan pendekatan yang beragam. Dewi dkk. menggunakan Regresi Linear untuk memprediksi konsumsi dan produksi daging unggas [1], sedangkan Prabowo dkk. memanfaatkan metode Long Short-Term Memory (LSTM) untuk melakukan forecasting populasi ternak di Indonesia [6]. Penelitian lain oleh Hanni dkk. melakukan forecasting produksi dan konsumsi daging ayam broiler [4], sementara Kulyk dkk. menunjukkan bahwa analisis deret waktu mampu menggambarkan perkembangan sektor peternakan unggas secara efektif [5]. Di sisi lain, Muthmainnah dkk. [3] serta Purwaningsih dkk. [7] membuktikan bahwa Simulasi Monte Carlo dapat digunakan untuk memodelkan ketidakpastian dan mendukung proses prediksi maupun pengambilan keputusan. Meskipun demikian, penelitian yang mengombinasikan metode Regresi Linear dan Simulasi Monte Carlo untuk melakukan pemodelan dan prediksi populasi ternak unggas di Provinsi Nusa Tenggara Barat (NTB) masih sangat terbatas. Oleh karena itu, penelitian ini bertujuan untuk melakukan pemodelan dan prediksi populasi ternak unggas di Provinsi NTB menggunakan metode Regresi Linear dan Simulasi Monte Carlo sebagai pendukung pengambilan keputusan di sektor peternakan.

II. TINJAUAN PUSTAKA

Populasi unggas memiliki peran besar sebagai penyediaan protein hewani bagi masyarakat. Catatan data populasi yang rutin tidak hanya berfungsi untuk mengevaluasi sektor peternakan dan membaca tren pertumbuhan, tetapi juga memudahkan pemerintah dalam merancang kebijakan produksi serta distribusi. Menurut Prabowo dkk., pemanfaatan data ini untuk *forecasting* sangat membantu menghasilkan informasi

akurat demi mendukung perencanaan pembangunan peternakan yang berkelanjutan [3].

Regresi Linear merupakan metode statistik yang digunakan untuk memodelkan hubungan antara variabel bebas dan variabel terikat sehingga dapat dimanfaatkan untuk melakukan prediksi berdasarkan pola historis data. Pada penelitian forecasting, variabel waktu sering digunakan sebagai variabel independen untuk memperkirakan nilai pada periode berikutnya. Dewi dkk. menunjukkan bahwa metode Regresi Linear mampu menghasilkan prediksi konsumsi dan produksi daging unggas dengan tingkat kesalahan yang relatif rendah. Hasil tersebut didukung oleh penelitian lain yang menerapkan Regresi Linear pada data time series dan menunjukkan bahwa metode ini mampu menghasilkan model prediksi yang sederhana, mudah diinterpretasikan, serta memiliki akurasi yang baik untuk berbagai kasus prediksi [1][8].

Simulasi Monte Carlo merupakan metode berbasis probabilitas yang memanfaatkan bilangan acak untuk menghasilkan berbagai kemungkinan hasil dari suatu sistem yang mengandung ketidakpastian. Metode ini tidak hanya menghasilkan satu nilai prediksi, tetapi juga memberikan gambaran rentang kemungkinan hasil yang dapat terjadi. Muthmainnah dkk. berhasil menerapkan Monte Carlo untuk memprediksi produksi bibit ayam kampung, sedangkan Purwaningsih dkk. menggunakan metode yang sama untuk menganalisis risiko pada industri unggas. Penelitian lain juga menunjukkan bahwa Monte Carlo efektif digunakan untuk forecasting serta analisis risiko pada berbagai bidang, termasuk sektor kesehatan dan pertanian [3][7][9][10].

Forecasting adalah cara memperkirakan masa depan memakai data masa lalu, yang di sektor peternakan berfungsi membantu menyusun strategi produksi dan distribusi. Melalui metode *Long Short-Term Memory* (LSTM), penelitian Prabowo dkk. membuktikan bahwa data populasi ternak memiliki pola yang bisa dimodelkan. Hasilnya menunjukkan adanya tren kenaikan pada populasi ayam pedaging dan ayam petelur di Indonesia [3].

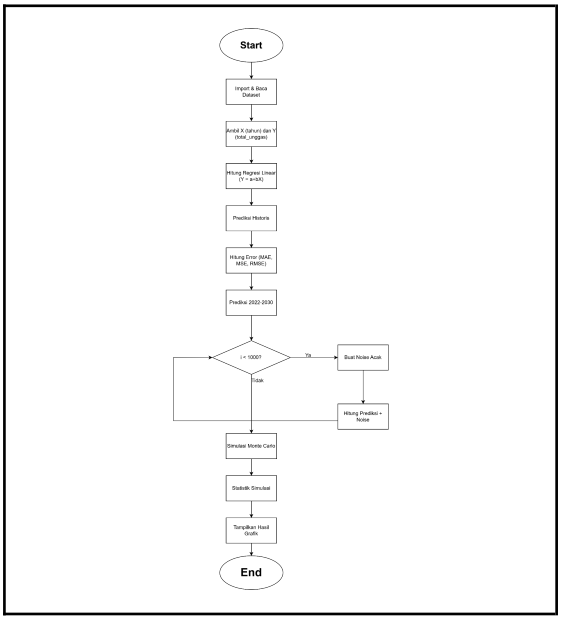
Berbagai penelitian terdahulu telah menerapkan metode prediksi pada sektor peternakan menggunakan pendekatan yang berbeda. Dewi dkk. dan penelitian lain menerapkan Regresi Linear untuk menghasilkan model prediksi berbasis data historis [1][8], Hanni dkk., Prabowo dkk., dan Kulyk dkk. melakukan forecasting menggunakan berbagai pendekatan time series [4][5][6], sedangkan Muthmainnah dkk., Purwaningsih dkk., serta penelitian Monte Carlo lainnya menunjukkan bahwa Simulasi Monte Carlo mampu mendukung proses prediksi dan analisis ketidakpastian [3][7][9][10]. Berdasarkan penelitian tersebut, sebagian besar studi masih menggunakan satu metode prediksi secara terpisah. Oleh karena itu, penelitian ini mengombinasikan Regresi Linear sebagai model utama prediksi dengan Simulasi Monte Carlo untuk menggambarkan ketidakpastian hasil prediksi populasi ternak unggas di Provinsi Nusa Tenggara Barat.

III. METODOLOGI PENELITIAN

Penelitian ini merupakan penelitian kuantitatif yang bertujuan untuk melakukan pemodelan dan prediksi populasi ternak unggas di Provinsi Nusa Tenggara Barat (NTB). Pendekatan yang digunakan adalah analisis data time series dengan metode Regresi Linear untuk membangun model prediksi berdasarkan pola historis data populasi unggas. Selanjutnya, Simulasi Monte Carlo digunakan untuk mengukur ketidakpastian hasil prediksi dan menghasilkan berbagai kemungkinan nilai populasi pada periode mendatang.

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari Dinas Peternakan dan Kesehatan Hewan Provinsi Nusa Tenggara Barat. Data yang dianalisis meliputi populasi empat jenis unggas, yaitu ayam buras, ayam petelur, ayam pedaging, dan itik pada periode tahun 2019–2021. Data dikumpulkan dalam bentuk deret waktu (time series) yang kemudian digunakan sebagai dasar dalam pembangunan model prediksi.

Penelitian dilakukan melalui beberapa tahapan yang meliputi pengumpulan data, pembangunan model Regresi Linear, evaluasi model, prediksi populasi unggas, serta Simulasi Monte Carlo untuk mengukur ketidakpastian hasil prediksi. Berikut adalah alur penelitian nya :



Gambar 3.1 Alur Penelitian

Penelitian diawali dengan proses impor dan pembacaan dataset populasi unggas Provinsi NTB. Selanjutnya dipilih variabel tahun sebagai variabel independen (X) dan total populasi unggas sebagai variabel dependen (Y). Data tersebut digunakan untuk membangun model Regresi Linear dengan persamaan Y=a+bX. Model yang dihasilkan kemudian digunakan untuk melakukan prediksi historis dan dievaluasi menggunakan MAE, MSE, dan RMSE. Setelah model dianggap layak, dilakukan prediksi populasi unggas untuk periode 2022–2030. Hasil prediksi selanjutnya digunakan pada Simulasi Monte Carlo dengan 1000 iterasi menggunakan noise acak untuk menghasilkan berbagai kemungkinan nilai populasi. Dari hasil simulasi dihitung statistik seperti rata-rata, nilai minimum, dan nilai maksimum, kemudian hasil ditampilkan dalam bentuk grafik untuk dianalisis lebih lanjut.

Metode Regresi Linear digunakan untuk membangun model hubungan antara variabel waktu (X) dan populasi unggas (Y). Persamaan Regresi Linear yang digunakan dalam penelitian ini adalah:

$$Y = a + bX$$

Gambar 3.2 Persamaan Regresi Linear

keterangan :
Y = nilai prediksi populasi unggas
X = periode waktu
a = konstanta (intercept)
b = koefisien regresi

Simulasi Monte Carlo digunakan untuk menganalisis ketidakpastian hasil prediksi yang diperoleh dari model Regresi Linear. Metode ini dilakukan dengan membangkitkan sejumlah bilangan acak berdasarkan karakteristik data historis sehingga diperoleh banyak kemungkinan nilai populasi pada masa mendatang. Hasil simulasi kemudian digunakan untuk menghitung nilai rata-rata, nilai minimum, nilai maksimum, serta rentang kemungkinan populasi yang dapat terjadi sehingga prediksi yang dihasilkan tidak hanya berupa satu nilai tunggal tetapi juga distribusi kemungkinan hasil.

IV. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan data populasi ternak unggas Provinsi Nusa Tenggara Barat (NTB) yang diperoleh dari publikasi resmi Dinas Peternakan dan Kesehatan Hewan Provinsi NTB. Data yang digunakan merupakan data tahunan selama periode 2016-2025 sehingga mampu menggambarkan perkembangan populasi unggas dalam rentang waktu yang cukup panjang.

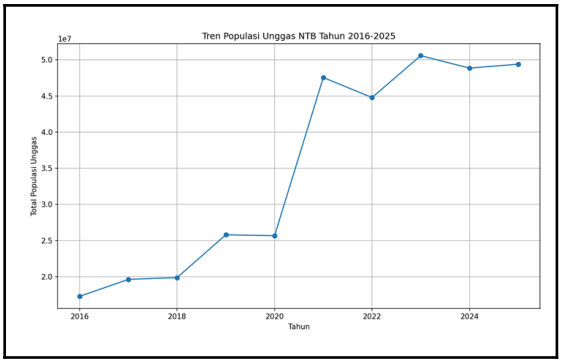
Variabel yang tersedia dalam dataset meliputi populasi ayam buras, ayam petelur, ayam pedaging, itik, serta total populasi unggas.

Tabel 4.1 Dataset Unggas

tahun	ayam_pedaging	ayam_petelur	ayam_buras	itik	Total unggas
2016	7.536.124	488.863	8.130.771	111.9651	17.275.409
2017	9.796.26	740.195	8.127.37	960.857	19.624.6

	8		4		94
2018	9.824.575	781.412	8.155.768	1.110.870	19.872.625
2019	15.107.542	1.246.095	8.262.646	1.176/647	25.792.930
2020	15.787.388	1.438.497	7.697.844	737.703	25.661.432
2021	34.124.279	3.444.223	9.201.252	777.467	47.547.212
2022	31.713.828	3.131.512	9.206.819	727.130	44.779.289
2023	39.045.383	2.780.498	8.223.707	524.339	50.573.927
2024	40.067.329	3.235.055	5.118.430	426.281	48.847.095
2025	39.827.290	3.242.613	5.814.585	504.793	49.389.281

Berdasarkan **Tabel 4.1,** terlihat bahwa populasi unggas di Provinsi NTB cenderung mengalami peningkatan dari tahun ke tahun, meskipun pada beberapa periode terdapat fluktuasi yang dipengaruhi oleh berbagai faktor, seperti kondisi ekonomi, perubahan pola konsumsi masyarakat, kebijakan pemerintah, maupun kondisi kesehatan ternak. Jika ditinjau per jenis unggas, ayam pedaging menjadi penyumbang terbesar dengan pertumbuhan yang paling pesat, sedangkan ayam buras dan itik justru cenderung menurun pada beberapa tahun terakhir. Kecenderungan peningkatan pada total populasi menunjukkan bahwa data memiliki pola tren yang cukup jelas sehingga berpotensi dimodelkan menggunakan pendekatan Regresi Linear.



Gambar 4.1 Grafik Data Historis

Berdasarkan grafik tren populasi unggas di Provinsi NTB periode 2016–2025, terlihat pola pertumbuhan yang tidak sepenuhnya linear. Populasi tumbuh perlahan pada awal periode, kemudian melonjak signifikan pada 2021, dan relatif stabil di kisaran 45–50 juta ekor pada tahun-tahun setelahnya. Pola pertumbuhan ini menjadi dasar pertimbangan digunakannya metode regresi time series dalam penelitian ini.

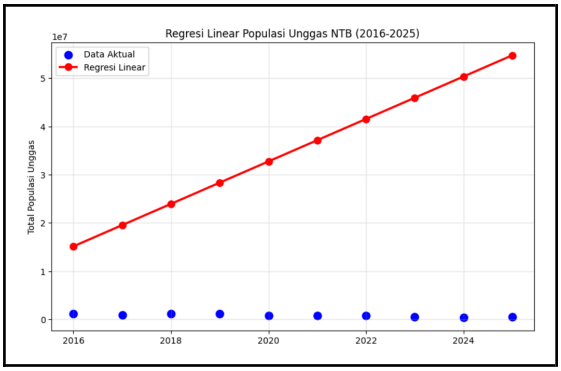
Proses pemodelan dilakukan menggunakan metode Regresi Linear sederhana dengan variabel tahun sebagai variabel independen dan total populasi unggas sebagai variabel dependen. Persamaan regresi yang dihasilkan adalah:

```
Persamaan Regresi:
Y = -8854443434.55 + 4399594.07X
```

Proses pemodelan dilakukan menggunakan metode Regresi Linear sederhana dengan variabel tahun sebagai variabel independen dan total populasi unggas sebagai variabel dependen.

```
[15138216.0727272  19537810.14545441 23937404.21818161 28336998.29090881
 32736592.36363602 37136186.43636322 41535780.50909042 45935374.58181763
 50334968.65454483 54734562.72727203]
```

Hasil tersebut menunjukkan pemodelan regresi linear terhadap total populasi unggas di Provinsi Nusa Tenggara Barat periode 2016–2025. Berdasarkan perhitungan, diperoleh persamaan regresi linear sebesar Y = -8.854.443.434,55 + 4.399.594,07X. Persamaan tersebut menunjukkan bahwa populasi unggas mengalami tren peningkatan rata-rata sebesar 4.399.594 ekor setiap tahun.



Gambar 4.2 Grafik Regresi Linear

Berdasarkan Gambar 4.2 Nilai prediksi yang dihasilkan model untuk data historis berkisar antara 15.138.216 ekor pada tahun 2016 hingga 54.734.563 ekor pada tahun 2025. Nilai tersebut direpresentasikan oleh garis regresi pada Gambar 4.2, sedangkan titik-titik biru menunjukkan data aktual. Perbedaan antara nilai prediksi dan data aktual pada beberapa tahun disebabkan oleh adanya fluktuasi populasi yang cukup signifikan, terutama pada periode 2020–2021 yang mengalami peningkatan tajam.

Tabel 4.2 Evaluasi Model

Metrik	Nilai
R^2	0.8652
MAE	4103432.6509090425
MSE	24883952405442.395
RMSE	4988381.742152699

Evaluasi model pada **Tabel 4.2** dilakukan menggunakan koefisien determinasi (R^2), MAE, MSE, dan RMSE. Nilai R^2 sebesar 0,8652 menunjukkan bahwa model mampu menjelaskan sekitar 86,52% variasi data populasi unggas. Hasil pengujian menunjukkan nilai MAE sebesar 4.103.432,65 yang berarti rata-rata selisih antara nilai aktual dan nilai prediksi adalah sekitar 4,1 juta ekor. Nilai MSE yang diperoleh sebesar 24.883.952.405.442,39 menunjukkan besarnya rata-rata kuadrat kesalahan prediksi yang dihasilkan model. Selanjutnya, nilai RMSE sebesar 4.988.381,74 menunjukkan bahwa tingkat kesalahan prediksi model berada pada kisaran 4,9 juta ekor. Nilai ini masih dapat diterima mengingat data populasi unggas yang digunakan memiliki skala puluhan juta ekor dan mengalami fluktuasi yang cukup signifikan pada beberapa tahun pengamatan.

Tabel 4.4 Prediksi Unggas 2026-2030

Tahun	Ayam Pedaging	Ayam Petelur	Ayam Buras	Itik	Total Unggas
2026	48.175.589	4.049.735	6.552.830	356.000	59.134.154
2027	52.519.697	4.412.797	6.327.178	274.078	63.533.750
2028	56.863.804	4.775.858	6.101.525	192.155	67.933.342
2029	61.207.911	5.138.920	5.875.873	110.233	72.332.937
2030	65.552.018	5.501.982	5.650.220	28.311	76.732.531

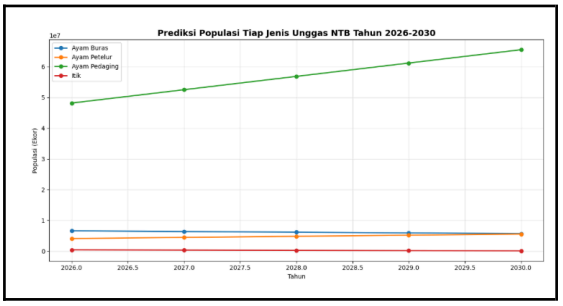
Berdasarkan hasil prediksi menggunakan metode Regresi Linear, total populasi unggas di Provinsi Nusa Tenggara Barat diperkirakan terus mengalami peningkatan selama periode 2026-2030. Total populasi unggas diprediksi meningkat dari 59.134.154 ekor pada tahun 2026 menjadi 76.732.531 ekor pada tahun 2030. Jenis unggas dengan populasi tertinggi adalah ayam ras pedaging yang diprediksi meningkat dari 48.175.589 ekor menjadi 65.552.018 ekor. Sementara itu, populasi ayam ras petelur juga menunjukkan tren peningkatan dari 4.049.735 ekor menjadi 5.501.982 ekor. Sebaliknya, populasi ayam buras dan itik cenderung mengalami penurunan selama periode prediksi. Pada tahun 2030, populasi ayam buras diprediksi mencapai 5.650.220 ekor, sedangkan populasi itik menjadi yang terendah dengan jumlah 28.311 ekor.

Tabel 4.5 Simulasi Monte Carlo

Parameter	Nilai
Rata-rata	76.619.223

Minimum	40.271.942
Maksimum	123.629.198

Hasil simulasi Monte Carlo menunjukkan bahwa populasi unggas di Provinsi Nusa Tenggara Barat pada periode prediksi memiliki nilai rata-rata sebesar 76.038.397 ekor. Dari 1.000 kali simulasi yang dilakukan, diperoleh nilai minimum sebesar 38.118.429 ekor dan nilai maksimum sebesar 111.746.047 ekor. Rentang hasil tersebut menunjukkan adanya variasi kemungkinan jumlah populasi unggas di masa mendatang akibat ketidakpastian yang terdapat dalam data historis. Meskipun demikian, nilai rata-rata hasil simulasi masih berada di sekitar hasil prediksi regresi linear, sehingga menunjukkan bahwa model yang digunakan memiliki konsistensi dalam menggambarkan tren pertumbuhan populasi unggas di Provinsi Nusa Tenggara Barat.



Gambar 4.3 Grafik Prediksi Populasi

Gambar 4.3 menunjukkan hasil prediksi populasi unggas berdasarkan jenisnya untuk periode 2026-2030. Hasil prediksi menunjukkan bahwa ayam ras pedaging menjadi jenis unggas dengan populasi tertinggi dan mengalami peningkatan yang paling signifikan, yaitu dari 48.175.589 ekor pada tahun 2026 menjadi 65.552.018 ekor pada tahun 2030. Populasi ayam ras petelur juga diprediksi mengalami peningkatan dari 4.049.735 ekor menjadi 5.501.982 ekor pada periode yang sama.

Sementara itu, populasi ayam buras menunjukkan kecenderungan menurun dari 6.552.830 ekor pada tahun 2026 menjadi 5.650.220 ekor pada tahun 2030. Penurunan yang lebih signifikan terjadi pada populasi itik, yang diprediksi turun dari 356.000 ekor menjadi 28.311 ekor pada tahun 2030. Secara keseluruhan, total populasi unggas diperkirakan meningkat dari 59.134.154 ekor pada tahun 2026 menjadi 76.732.531 ekor pada tahun 2030. Hasil tersebut menunjukkan bahwa peningkatan total populasi unggas di Provinsi Nusa Tenggara Barat terutama didorong oleh pertumbuhan populasi ayam ras pedaging yang mendominasi dibandingkan jenis unggas lainnya.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil membangun model prediksi populasi unggas di Provinsi Nusa Tenggara Barat (NTB) menggunakan metode regresi linear berbasis data historis tahun 2016–2025. Hasil evaluasi model menunjukkan nilai R^2 sebesar 0,8652 yang berarti model mampu menjelaskan 86,52% variasi data populasi unggas, dengan nilai MAE sebesar 4.103.432 ekor dan RMSE sebesar 4.988.382 ekor, sehingga model dinilai cukup baik dalam merepresentasikan tren pertumbuhan populasi unggas di NTB. Berdasarkan model tersebut, prediksi total populasi unggas menunjukkan tren pertumbuhan yang konsisten dari 59.134.154 ekor pada tahun 2026 hingga mencapai 76.732.531 ekor pada tahun 2030, dengan rata-rata pertambahan sekitar 4,4 juta ekor per tahun. Apabila ditinjau per jenis, populasi didominasi ayam pedaging, diikuti ayam petelur, ayam buras, dan itik sebagai populasi terkecil. Hasil prediksi ini mengindikasikan bahwa sektor peternakan unggas di NTB memiliki potensi pertumbuhan yang cukup besar dan perlu didukung dengan perencanaan kapasitas produksi, ketersediaan pakan, serta kebijakan yang memadai dari pemerintah daerah guna mengantisipasi peningkatan populasi unggas dalam jangka panjang.

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan pertimbangan untuk penelitian selanjutnya. Pertama, meskipun model regresi linear menghasilkan nilai R^2 sebesar 0,8652, nilai RMSE yang cukup besar yakni 4.988.382 ekor mengindikasikan bahwa model masih memiliki ruang untuk ditingkatkan, sehingga disarankan untuk mencoba metode yang lebih kompleks seperti regresi polynomial, ARIMA, atau pendekatan machine learning guna

memperoleh akurasi yang lebih tinggi. Kedua, penurunan tren pada populasi ayam buras dan itik perlu dikaji lebih lanjut, misalnya dengan menambahkan variabel independen seperti harga pakan ternak, jumlah rumah tangga peternak, dan kondisi iklim, agar model dapat menangkap faktor-faktor yang turut memengaruhi dinamika populasi tiap jenis unggas secara lebih komprehensif.

DAFTAR PUSTAKA

[1] M. M. Dewi, L. D. Farida, and M. Nuraminudin, "Regresi Linier untuk Prediksi Konsumsi dan Produksi Daging Unggas (Studi Kasus: Provinsi Jawa Barat)," Journal of Information System Management (JOISM), vol. 4, no. 2, pp. 81–85, 2023.

[2] D. C. Widianingrum and H. Khasanah, "Tren Perkembangan, Kondisi, Permasalahan, Strategi, dan Prediksi Komoditas Peternakan Indonesia (2010–2030)," ANIMPRO: Conference of Applied Animal Science Proceeding Series, pp. 6–17, 2021.

[3] I. Muthmainnah, A. Richi, and I. M. Karo-Karo, "Optimization of Kampung Chicken Seedling Production Using the Monte Carlo Method (Case Study on Hacci Farm)," Journal of Artificial Intelligence and Engineering Applications, vol. 3, no. 3, pp. 852–857, Jun. 2024.

[4] M. Hanni, I. Baroh, and B. Y. Ariadi, "Forecasting Produksi dan Konsumsi Daging Ayam Broiler di Provinsi Jawa Timur," Jurnal Peternakan Sriwijaya, vol. 11, no. 1, pp. 33–41, Jun. 2022.

[5] A. Kulyk, K. Fokina-Mezentseva, A. Saiun, and D. Saiun, "Forecasting the Development of Poultry Farming Based on Time Series," Agricultural and Resource Economics, vol. 11, no. 1, 2025.

[6] T. Prabowo, Lestariningsih, A. C. Fauzan, and V. Y. Mafula, "Analisis Deret Waktu untuk Forecasting Populasi Ternak di Indonesia dengan Model LSTM," JSAI: Journal Scientific and Applied Informatics, vol. 8, no. 1, pp. 109–121, Jan. 2025.

[7] R. Purwaningsih, M. Arief, N. U. Handayani, D. Rahmawati, and A. Mustikasari, "Market Risk Assessment on Poultry Industry Using Monte Carlo Simulation," IOP Conference Series: Materials Science and Engineering, vol. 403, no. 1, 012044, 2018.

[8] A. Sitohang, H. Simamora, and A. Siregar, "Application of Linear Regression Method in Predicting Chicken Egg Sales," *InfoSains: Jurnal Ilmiah Ilmu Pengetahuan dan Teknologi*, vol. 1, no. 4, 2025.

[9] M. Rizki, R. Gunawan, A. Pratama, *et al*., "Implementation of Forecasting with the Monte Carlo Simulation Method to Predict Supply and Demand for Psychotropic Drug Products," *Brilliance: Research of Artificial Intelligence*, vol. 3, no. 4, pp. 629–637, 2023.

[10] F. R. de Amorim, C. C. Guimarães, P. Afonso, and M. S. G. Tobias, "Forecasting Cost Risks of Corn and Soybean Crops through Monte Carlo Simulation," *Applied Sciences*, vol. 14, no. 17, Art. no. 8030, 2024.

OPTIMALISASI EKSTRAKSI FITUR TF-IDF PADA ANALISIS TEKS SKALA BESAR MELALUI IMPLEMENTASI KOMPUTASI PARALEL BERBASIS MESSAGE PASSING INTERFACE

Jauda Avaro Zufaru [1)], Muhammad Akbar [2)], M. Sagos [3)], Rendy Wahyu Islami [4)], Arya Desyahranie.[5)]
1, 2, 3, 4, 5) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: f1d02410059@student.unram.ac.id

Abstrak

Pertumbuhan eksponensial data teks tidak terstruktur menuntut metode ekstraksi fitur yang cepat dan terukur, namun algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) pada arsitektur serial menghadapi kemacetan eksekusi berurutan yang membatasi skalabilitasnya pada korpus berukuran besar. Penelitian ini bertujuan merancang dan mengimplementasikan TF-IDF paralel berbasis Message Passing Interface (MPI) melalui pustaka mpi4py dengan pola *master-worker*, kemudian membandingkan kinerjanya terhadap eksekusi serial untuk menemukan konfigurasi proses yang paling optimal pada berbagai skala data. Penelitian dilaksanakan melalui eksperimen kuantitatif berbasis pengujian kinerja sistem pada korpus berbasis data spam dengan lima ukuran dokumen, yaitu 10.000, 25.000, 50.000, 75.000, dan 100.000 dokumen. Setiap ukuran diuji pada delapan konfigurasi proses, $P \in \{1, 2, 4, 6, 8, 12, 16, 24\}$, masing-masing diulang lima kali untuk memperoleh waktu eksekusi, *speedup*, dan efisiensi rata-rata. Hasil menunjukkan bahwa konfigurasi dua proses secara konsisten kurang efektif, bahkan lebih lambat daripada serial pada tiga dari lima ukuran data, akibat beban komunikasi kamus *document frequency* yang tidak berkurang proporsional terhadap jumlah proses. Konfigurasi optimal bergeser seiring pertumbuhan data, dari P=12 pada 10.000 dokumen dengan *speedup* tertinggi 2,60×, ke P=16 pada 25.000 dan 50.000 dokumen, hingga P=24 pada 75.000 dan 100.000 dokumen dengan *speedup* stabil pada kisaran 1,73×–1,78×. Efisiensi menurun secara monoton sejalan dengan Hukum Amdahl. Temuan ini membuktikan bahwa komputasi paralel berbasis MPI efektif mengatasi kemacetan eksekusi TF-IDF, dengan pemilihan konfigurasi proses yang tepat ditentukan oleh rasio granularitas data terhadap biaya komunikasi, serta membuka peluang penelitian lanjutan pada konfigurasi proses dan volume data yang lebih besar.
Kata kunci: Komputasi Paralel; Message Passing Interface; Skalabilitas; TF-IDF.

Abstract

The exponential growth of unstructured text data demands fast, scalable feature extraction methods, yet the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm on a serial architecture suffers from a serial bottleneck that limits its scalability on large corpora. This study designs and implements a parallel TF-IDF based on the Message Passing Interface (MPI) through the mpi4py library using a master-worker pattern, then compares its performance against serial execution to identify the optimal process configuration across data scales. The research was conducted as a quantitative benchmarking experiment on a spam-based corpus with five document sizes: 10,000, 25,000, 50,000, 75,000, and 100,000. Each size was tested across eight process configurations, $P \in \{1, 2, 4, 6, 8, 12, 16, 24\}$, with five repetitions each, to obtain average execution time, speedup, and efficiency. Results show the two-process configuration was consistently ineffective, even slower than serial in three of the five sizes, due to the communication cost of the document frequency dictionary not decreasing proportionally with the number of processes. The optimal configuration shifted as data size grew, from P=12 at 10,000 documents with the highest speedup of 2.60×, to P=16 at 25,000 and 50,000 documents, and to P=24 at 75,000 and 100,000 documents with a stable speedup of 1.73×-1.78×. Efficiency decreased monotonically in line with Amdahl's Law. These findings demonstrate that MPI-based parallel computing effectively overcomes the TF-IDF execution bottleneck, with the optimal process configuration determined by the ratio of data granularity to communication cost, opening opportunities for further research on larger process configurations and data volumes.
Keyword: *Message Passing Interface; Parallel Computing; Scalability; TF-IDF.*

I. PENDAHULUAN

Era disrupsi digital dan komputasi modern, pertumbuhan eksponensial dari data berskala masif telah mengubah paradigma fundamental dalam bidang ilmu komputer, khususnya pada pemrosesan bahasa alami dan penambangan teks. Data yang dihasilkan oleh interaksi manusia melalui platform digital, mulai dari ulasan perdagangan elektronik, percakapan media sosial, surel, hingga repositori dokumen akademis, sebagian besar berwujud data teks

yang sama sekali tidak terstruktur [1]. Berbeda dengan data terstruktur yang tersimpan rapi di dalam basis data relasional, data teks mentah memiliki tingkat kompleksitas yang tinggi dan tidak dapat langsung diproses oleh algoritma komputasi maupun model pembelajaran mesin (*machine learning*). Oleh karena itu, diperlukan sebuah proses transformasi matematis yang sangat presisi untuk menerjemahkan teks menjadi representasi numerik atau vektor fitur yang bermakna. Berbagai metode dan teknik ekstraksi fitur yang terus berkembang, algoritma pembobotan statistik tetap menjadi standar fondasi yang esensial karena kemampuannya dalam mengukur signifikansi suatu kata secara objektif, baik dalam konteks lokal pada sebuah dokumen tunggal maupun dalam konteks global yang mencakup seluruh korpus dokumen yang ada [1][2].

Algoritma yang secara konsisten membuktikan keandalannya dalam proses ekstraksi fitur ini adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF tidak sekadar menghitung berapa kali sebuah kata muncul secara kasar, melainkan melakukan kalkulasi bobot relevansi yang sangat tajam untuk memfilter derau bahasa [2]. Logika fundamental dari TF-IDF bekerja dengan dua ukuran matriks utama. Pertama, ukuran lokal atau *Term Frequency* (TF) yang mengukur seberapa sering sebuah kata muncul di dalam satu dokumen tertentu. Semakin tinggi frekuensi kemunculan lokal ini, semakin penting kata tersebut dalam merepresentasikan isi dokumen. Namun, pengukuran ini harus dikalibrasi menggunakan ukuran global, yaitu *Inverse Document Frequency* (IDF). IDF berfungsi untuk mengukur seberapa langka kata tersebut di seluruh himpunan dokumen yang ada. Pengali global ini sangat vital karena kata hubung yang sangat umum akan memiliki frekuensi lokal yang sangat tinggi. Namun, karena kata tersebut muncul di hampir semua dokumen, nilai IDF-nya akan meredam bobot kata tersebut mendekati angka nol. Sebaliknya, terminologi spesifik yang mendefinisikan konteks unik dari sebuah teks akan mendapatkan lonjakan bobot yang sangat tinggi, menghasilkan sebuah vektor representasi yang akurat dan siap untuk dianalisis lebih lanjut [3].

Meskipun TF-IDF memiliki keunggulan absolut dalam hal akurasi representasi, kesederhanaan logika matematis, dan kemampuan interpretasi, implementasi tradisional dari algoritma ini menyimpan sebuah kelemahan arsitektural yang sangat krusial ketika dihadapkan pada dimensi data yang sangat besar. Permasalahan utama ini dikenal secara luas di kalangan peneliti komputasi berkinerja tinggi sebagai kemacetan eksekusi berurutan atau *serial bottleneck* [4]. Pada arsitektur komputasi tradisional berbasis skalar, seluruh tahapan algoritma, yang membentang dari operasi pembacaan berkas, prapemrosesan teks, perhitungan frekuensi kemunculan kata, hingga agregasi global dan kalkulasi nilai akhir, dieksekusi sepenuhnya oleh sebuah prosesor atau inti komputasi tunggal secara linear [5]. Mengandalkan satu unit pemrosesan untuk mengurai dan merangkai kosakata dari *dataset* berskala jutaan entri dokumen pada akhirnya akan menciptakan tingkat inefisiensi yang fatal bagi skalabilitas sistem analitik [6].

Fenomena kemacetan eksekusi berurutan ini dapat divisualisasikan secara tajam melalui metafora arsitektural berupa sebuah jam pasir yang tersumbat di bagian tengah lehernya. Dalam visualisasi ini, hamparan pasir di bagian atas melambangkan ledakan volume data teks digital yang sifatnya eksponensial dan terus bertambah tanpa henti. Sementara itu, perumpamaan leher jam pasir yang sangat sempit melambangkan arsitektur pemrosesan inti tunggal. Seberapa pun tingginya kecepatan spesifik dari sebuah prosesor modern, jika seluruh beban komputasi ekstraksi dan pembobotan fitur dipaksa melewati satu jalur eksekusi tunggal, maka aliran penyelesaian data akan tertahan dan macet. Ketika dimensi dari dokumen membesar dan matriks kosakata unik bertambah, beban kerja yang ditumpuk pada satu titik ini akan menguras kapasitas memori lokal. Akibatnya, sistem akan mengalami perlambatan drastis, meningkatkan waktu komputasi dari hitungan detik menjadi berjam-jam, dan pada batas ekstremnya, perangkat keras akan lumpuh karena ketidakmampuan mengalokasikan sumber daya komputasi secara wajar.

Menghadapi krisis data modern dan risiko kelumpuhan akibat penyumbatan serial tersebut, adaptasi terhadap pendekatan komputasi paralel dan terdistribusi telah bertransformasi dari sekadar pilihan optimasi menjadi sebuah kebutuhan mutlak yang tidak dapat dihindari. Peralihan paradigma dari pemrosesan data monolitik yang berat dan terpusat menuju mode pemrosesan melalui *node* yang terdistribusi menuntut kerangka kerja perangkat lunak yang tangguh dan memiliki standar industri yang teruji. Dalam konteks ini,

Message Passing Interface (MPI) diakui sebagai spesifikasi komunikasi yang paling dominan dan terandalkan untuk sistem komputasi berbasis memori terdistribusi [7][8]. Berbeda dengan model memori terbagi yang terbatas pada jumlah inti dalam satu perangkat, MPI memungkinkan distribusi beban komputasi untuk melintasi batas-batas fisik perangkat keras, menghubungkan ratusan hingga ribuan unit pemrosesan agar dapat bekerja secara simultan [9]. MPI memecah beban dokumen yang tadinya menggunung pada satu titik menjadi serpihan-serpihan tugas diskret yang disebarkan secara merata.

Mekanisme fungsional dari Message Passing Interface dapat dipahami melalui analogi hierarkis mengenai sebuah tim profesional yang bekerja secara terkoordinasi dan sangat efisien. Dalam struktur ini, arsitektur menunjuk satu proses utama yang disebut sebagai *root* untuk bertindak layaknya seorang manajer proyek, sementara prosesor-prosesor lainnya beroperasi sebagai pekerja aktif. Seorang manajer yang efektif memahami bahwa mengerjakan keseluruhan proyek secara mandiri adalah tindakan yang kontraproduktif. Oleh karena itu, manajer menggunakan operasi penyebaran untuk memecah *dataset* raksasa secara berimbang dan mendelegasikannya kepada seluruh pekerja. Segera setelah pendelegasian selesai, para pekerja akan mengeksekusi komputasi lokal mereka masing-masing secara serentak, menghitung frekuensi kata di area data mereka tanpa perlu menunggu proses pekerja lain selesai. Setelah pekerja menuntaskan tugasnya, operasi pengumpulan menarik kembali semua temuan lokal tersebut ke meja manajer untuk disatukan menjadi nilai agregat global. Sebagai penutup dari alur kerja yang elegan ini, manajer menghitung nilai final dan menyiarkan kembali hasil tersebut kepada seluruh pekerja agar mereka dapat memfinalisasi kalkulasi nilai akhir secara serempak. Pendekatan berbasis kerja tim ini memastikan bahwa perangkat keras dapat dieksploitasi hingga batas maksimalnya.

Kendati komputasi paralel MPI menawarkan penyelesaian yang menjanjikan terhadap masalah, implementasi terdistribusi ini membawa sebuah konsekuensi logis berupa pertukaran nilai atau *trade-off* operasional yang sangat krusial. Peralihan dari memori tunggal ke memori terdistribusi memperkenalkan variabel baru ke dalam waktu eksekusi sistem, yaitu *overhead* komunikasi jaringan [10]. Pemrosesan paralel memang mampu memangkas waktu komputasi matematis secara agresif, tetapi sebagai gantinya, sistem menuntut alokasi waktu tambahan untuk mentransmisikan data antara manajer dan para pekerja melalui jaringan interkoneksi. Keseimbangan antara waktu komputasi dan biaya komunikasi ini adalah faktor penentu utama dari skalabilitas sistem. Jika algoritma tidak dirancang dengan hati-hati, interaksi pengiriman pesan ini dapat menjelma menjadi sumber penyumbatan baru.

Fenomena ketidakseimbangan ini didefinisikan secara konseptual sebagai paradoks granularitas halus. Berdasarkan uji empiris dalam pengembangan arsitektur paralel, penambahan jumlah prosesor tidak selalu berbanding lurus dengan peningkatan kecepatan yang linear [10]. Terdapat sebuah titik jenuh yang beban komputasi per proses menjadi terlalu kecil karena data telah dibagi ke terlalu banyak pekerja. Ketika hal ini terjadi, waktu sepersekian detik yang dihemat dari perhitungan matematis justru hangus oleh latensi yang ditimbulkan saat manajer harus menyebarkan dan mengumpulkan pesan dari sekian banyak pekerja [11]. Alih-alih menjadi lebih cepat, eksekusi paralel pada jumlah prosesor yang masif dengan beban data yang ringan justru menyebabkan waktu penyelesaian membengkak lebih lama dari kondisi optimalnya, menjadikan sistem lebih sibuk berkomunikasi daripada melakukan komputasi nyata.

Berpijak pada uraian latar belakang yang ekstensif, identifikasi masalah arsitektural, dan analisis konseptual mengenai tantangan komputasi terdistribusi tersebut, penelitian ini diformulasikan dengan serangkaian tujuan yang spesifik dan terukur. Pertama, penelitian ini bertujuan untuk merancang dan mengimplementasikan algoritma TF-IDF secara utuh untuk memproses representasi numerik pada *dataset* teks. Kedua, penelitian ini akan mengintegrasikan secara mendalam pendekatan paralel menggunakan protokol Message Passing Interface (MPI) untuk meruntuhkan kendala eksekusi berurutan dengan mendelegasikan beban secara simultan. Ketiga, penelitian ini berupaya untuk membedah dan membandingkan performa empiris antara metode komputasi serial dan metode komputasi paralel, menilik secara spesifik rasio percepatan dan tingkat efisiensi sistem. Terakhir, penelitian ini akan melakukan analisis skalabilitas yang tajam untuk mengidentifikasi dinamika pengaruh penambahan jumlah inti pemrosesan terhadap

pergeseran waktu komunikasi, guna menemukan titik optimal yang mana perangkat keras dan arsitektur perangkat lunak bekerja dalam harmoni yang absolut.

Pencapaian dari tujuan-tujuan penelitian ini diharapkan mampu memberikan kontribusi yang bernilai tinggi secara teoritis maupun praktis. Dari sudut pandang teoritis, hasil analisis akan memberikan pemahaman mendalam tentang perilaku algoritma ekstraksi fitur ketika dihadapkan pada hambatan komunikasi sistem paralel, memperkaya literatur di bidang penambangan teks terdistribusi. Secara praktis, penelitian ini akan menjadi rujukan arsitektural bagi para insinyur perangkat lunak dan ilmuwan data dalam memilih strategi pemrosesan yang paling efisien ketika berhadapan dengan data teks berdimensi raksasa. Dengan demikian, pemahaman terhadap batasan granularitas tugas dan optimalisasi waktu eksekusi dapat diaplikasikan untuk membangun infrastruktur kecerdasan buatan yang tidak hanya cerdas secara analitik, tetapi juga tangguh, terukur, dan responsif terhadap tuntutan kinerja di dunia nyata.

II. TINJAUAN PUSTAKA

Evolusi metode pemrosesan teks dan komputasi berkinerja tinggi telah melahirkan lanskap penelitian yang luas dan dinamis. Pada bagian tinjauan pustaka ini, literatur-literatur kontemporer yang dipublikasikan dalam beberapa tahun terakhir akan dianalisis secara kritis untuk memahami posisi metode ekstraksi fitur tradisional dalam ekosistem komputasi modern, membandingkan kerangka kerja terdistribusi yang dominan, serta mengeksplorasi tantangan skalabilitas dan latensi komunikasi pada sistem terdistribusi. Analisis ini ditujukan tidak untuk mendiskreditkan temuan terdahulu, melainkan untuk mengidentifikasi celah penelitian yang spesifik dan menempatkan kontribusi eksperimen paralel TF-IDF ini sebagai pelengkap yang substansial bagi korpus pengetahuan yang telah ada.

Dalam ranah *Natural Language Processing* (NLP), teknik ekstraksi fitur merupakan komponen inti yang menentukan kualitas pemodelan. Meskipun era saat ini didominasi oleh arsitektur *deep learning* dan *transformer* yang mampu menangkap semantik kontekstual tingkat tinggi, metode berbasis frekuensi statistik seperti TF-IDF terbukti mempertahankan relevansinya dalam penerapan sistem berskala industri yang menuntut efisiensi komputasi dan transparansi model. Penelitian yang dilakukan oleh Wang [12] menguraikan bahwa dalam sistem pencarian teks lengkap dan platform analitik produksi, TF-IDF tetap menjadi blok pembangun yang esensial. Wang menegaskan bahwa TF-IDF menawarkan jaminan interpretasi yang absolut (*explainability*), yang sangat krusial ketika pengembang perlu mengaudit mengapa sebuah istilah tertentu mendominasi hasil peringkat, sebuah karakteristik yang sering kali absen pada model pencarian berbasis vektor padat (*dense embeddings*). Temuan ini memperkuat landasan bahwa optimalisasi TF-IDF bukanlah upaya yang usang, melainkan investasi strategis pada infrastruktur fundamental *text mining*.

Seiring dengan pemanfaatan TF-IDF pada korpus berskala raksasa, kendala utama yang diidentifikasi oleh para akademisi bergeser dari kompleksitas algoritma menuju beban waktu komputasi. Chaurasia *et al*. [13] merespons tantangan ini dengan mengembangkan pendekatan paralelisasi TF-IDF yang ditujukan untuk meningkatkan kecepatan sistem peringkasan teks ekstraktif. Penelitian tersebut menunjukkan bahwa paralelisasi mengurangi waktu komputasi secara drastis, membuat proses pengambilan informasi pada lanskap digital yang bergerak cepat menjadi jauh lebih efisien. Demikian pula, Paulsen *et al*. [4] memperkenalkan Sparkly, sebuah pemblokir berbasis TF-IDF untuk sistem *entity matching* yang dieksekusi secara terdistribusi tanpa pembagian memori (*share-nothing fashion*) di atas klaster Apache Spark. Paulsen *et al*. [4] juga menyimpulkan bahwa pemblokiran TF-IDF top-k di lingkungan terdistribusi membentuk garis dasar (*baseline*) performa yang sangat kuat dan sangat terukur dalam menangani ukuran data yang masif. Kedua penelitian ini membuktikan validitas pendekatan paralel untuk mempercepat matriks TF-IDF, namun keduanya sebagian besar bergantung pada implementasi model memori bersama seperti OpenMP atau *framework* tingkat tinggi yang memiliki *overhead* lapisan mesin virtual (JVM) yang besar.

Bahasan kerangka kerja sistem terdistribusi, perbandingan antara Apache Spark dan Message Passing Interface (MPI) menjadi diskursus yang signifikan di kalangan pengembang komputasi berkinerja tinggi atau *High Performance Computing* (HPC) [15]. Evaluasi *benchmarking* ekstensif yang

dilakukan oleh Han *et al*. [8], serta studi terkait pemrosesan Big Data memberikan wawasan kuantitatif bahwa MPI secara konsisten mengungguli Apache Spark hingga dua sampai sepuluh kali lipat dalam penanganan tugas-tugas yang murni berfokus pada intensitas komputasi (*compute-intensive tasks*). Spark, di satu sisi, lebih diutamakan untuk produktivitas pengembang dan memiliki sistem toleransi kesalahan (*fault tolerance*) yang dibangun secara *default*. Namun, Dalcin dan Fang [16] dalam ulasan mereka mengenai evolusi paket *mpi4py* menegaskan bahwa penerapan antarmuka Python untuk MPI telah matang, memungkinkan eksekusi kode dengan kecepatan yang nyaris menyamai bahasa C untuk distribusi data numerik secara langsung, menghilangkan hambatan pemrograman tingkat rendah yang biasanya melekat pada arsitektur MPI. Oleh karenanya, mengeksplorasi performa MPI untuk TF-IDF di lingkungan Python menggunakan *mpi4py* menjadi celah yang menjanjikan untuk menghadirkan komputasi tingkat HPC pada pengembangan kecerdasan buatan berbasis teks.

Namun demikian, paralelisasi menggunakan MPI memiliki tantangan tersendiri, khususnya mengenai keseimbangan antara kecepatan dan beban komunikasi, yang dalam eksperimen awal disebut sebagai paradoks granularitas. Penelitian yang meninjau toleransi kesalahan dan analisis *bottleneck* oleh Bland *et al*. [17] mengungkapkan bahwa mengidentifikasi jalur kritis komputasi adalah isu utama. Ketidakseimbangan beban komputasi di antara unit pekerja menyebabkan waktu yang berharga terbuang sia-sia saat menunggu proses kolektif seperti MPI_Gather atau hambatan sinkronisasi (*barrier wait*). Fenomena inilah yang membatasi skalabilitas linear, serta penambahan *node* pemrosesan akan menurunkan performa ketika latensi komunikasi untuk mengatur potongan-potongan matriks kecil melampaui waktu yang dihemat dari perhitungan paralel itu sendiri.

Lebih lanjut, dalam konteks optimasi pemrograman hibrida, Althiban *et al*. [17] menganalisis bahwa penggabungan model memori terdistribusi (MPI) dengan model memori bersama (OpenMP) sering kali memicu kompleksitas cacat perangkat lunak seperti kebuntuan komunikasi (*deadlocks*) dan kondisi balapan (*race conditions*). Penelitian Althiban mendemonstrasikan bahwa algoritma klasifikasi bahkan harus diekstraksi menggunakan fitur TF-IDF untuk memprediksi cacat pada kode paralel itu sendiri. Hal ini mengonfirmasi betapa rentan dan sensitifnya sistem terdistribusi terhadap desain komunikasi yang tidak optimal.

III. METODOLOGI PENELITIAN

Metodologi yang dikembangkan dalam penelitian ini dirancang secara sistematis untuk menjawab tantangan skalabilitas ekstraksi fitur teks dengan mengadopsi paradigma komputasi paralel. Pendekatan yang digunakan merupakan penelitian eksperimental kuantitatif berbasis *benchmarking* sistem (*system performance benchmarking*). Implementasi algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) dibangun dalam dua variasi arsitektur, arsitektur serial murni dan terdistribusi paralel guna memungkinkan komparasi langsung dan objektif. Arsitektur paralel direalisasikan menggunakan spesifikasi Message Passing Interface (MPI) yang dieksekusi melalui pustaka *mpi4py* dalam bahasa pemrograman Python. Desain sistem mendayagunakan pola *master-worker* yang mana fungsi pengelolaan data global dikonsolidasikan pada *root process*, sementara beban kerja komputasi repetitif didistribusikan ke seluruh pekerja yang ada dalam komunikator.

III.I. Akuisisi *Dataset* dan Mekanisme Prapemrosesan

Untuk menguji ketahanan dan kinerja eksekusi sistem, eksperimen ini menggunakan *dataset* tekstual publik, yakni himpunan data ulasan konsumen (*Amazon Fine Food Reviews*) dan himpunan data identifikasi pesan (spam.csv) yang memuat sekitar 100.000 dokumen teks mentah berformat *comma separated values*. Meskipun secara volumetrik jumlah dokumen ini tergolong pada skala menengah, varians kosakata unik yang terkandung di dalamnya menciptakan ekspansi dimensi fitur yang sangat masif dan asimetris. Hal ini menjadikannya instrumen uji yang sangat merepresentasikan beban komputasi matriks renggang (*sparse matrix*) pada kondisi industri sesungguhnya. Sebelum ekstraksi fitur komputasional dapat dieksekusi, teks mentah wajib dikonversi menjadi unit-unit logis melalui tahap prapemrosesan secara lokal di tiap *node*. Tahap ini secara sengaja dirancang ramping (*lightweight*) untuk memastikan bahwa fokus beban komputasi diletakkan pada operasi agregasi algoritma TF-IDF. Proses yang dilakukan mencakup:

a. *Case Folding* (Penyeragaman Karakter) yang mengubah seluruh karakter alfabet menjadi huruf kecil (*lowercase*) melalui instruksi internal. Normalisasi ini esensial agar model tidak menciptakan dimensi matriks terpisah untuk kata yang identik secara semantik namun berbeda secara tipografi (misalnya, "Sistem", "SISTEM", dan "sistem").
b. Tokenisasi Berbasis Spasi (*Whitespace Tokenization*), yaitu memecah untaian kalimat panjang menjadi sekumpulan *array* kata tunggal (token).

Dalam skenario metodologis ini, langkah eliminasi kata hubung (*stopwords removal*) dan pemotongan imbuhan (*stemming*) tidak diaktifkan. Keputusan taktis ini diambil untuk mempertahankan integritas beban komputasi yang tinggi [19]. Keberadaan kata-kata umum yang memiliki frekuensi luar biasa tinggi akan menguji secara langsung seberapa tangguh arsitektur MPI mengelola struktur data kamus (*dictionary*) yang membengkak saat fase agregasi global dilakukan.

III.II. Konstruksi Matematis Algoritma Pembobotan

Algoritma TF-IDF mentransformasi token teks menjadi skalar fitur berkelanjutan yang merepresentasikan tingkat signifikansi informasi. Komputasi ini disusun berdasarkan tiga tahapan matematis terpisah yang dalam arsitektur paralel dipetakan pada area penyelesaian yang berbeda:

a. Komponen *Term Frequency* (TF)

Komponen ini mengukur kepadatan lokal sebuah kata t di dalam batas-batas dokumen d Untuk menyeimbangkan rentang hasil antara dokumen yang sangat pendek dan dokumen yang sangat panjang, nilai dihitung melalui pembagian kemunculan absolut terhadap panjang dokumen.

$$TF(t,d) = \frac{f(t,d)}{|d|}$$

Keterangan:
- $f(t,d)$ merupakan frekuensi observasi term *t* di dalam dokumen d.
- $|d|$ merupakan jumlah total absolut keseluruhan token di dalam dokumen d.

b. Komponen *Inverse Document Frequency* (IDF)

Komponen ini merupakan metrik peredam global yang memberikan penalti pada kata yang memiliki tingkat generalisasi tinggi melintasi seluruh himpunan dokumen *D*. Karena mengharuskan pengetahuan absolut dari keseluruhan korpus, perhitungan ini wajib diagregasi secara sentral.

$$IDF(t) = \log_{10}\left(\frac{|D|}{DF(t)}\right)$$

Keterangan:
- $|D|$ merepresentasikan jumlah agregat seluruh dokumen di dalam *dataset*.
- $DF(t)$ merepresentasikan jumlah observasi dokumen yang memuat sekurang-kurangnya satu term *t.*

c. Vektor Fitur *Representatif* Akhir

Integrasi dari metrik lokal dan global menghasilkan bobot ekuilibrium yang digunakan oleh sistem pembelajaran mesin untuk klasifikasi.

$$TF - IDF(t,d) = TF(t,d) \times IDF(t)$$

Sistem mengiterasi perkalian ini untuk setiap elemen token unik, menghasilkan sebuah ruang vektor yang dikuantifikasi secara akurat.

III.III. Arsitektur Paralelisasi dan Pemetaan Transmisi MPI

Mengonversi algoritma linear menjadi pemrosesan bersama menuntut orkestrasi distribusi data dan sinkronisasi komunikasi yang cermat untuk menghindari penyumbatan jaringan (*network bottleneck*). Alur logika terdistribusi ini dipetakan memanfaatkan fungsi kolektif dari antarmuka *mpi4py*.

Penggunaan operasi langsung terhadap objek dinamis Python (melalui pemanggilan metode huruf kecil seperti *scatter*, *gather*, *bcast*) memberikan keunggulan berupa fleksibilitas tanpa manajemen penyangga memori mentah (*buffer*) tingkat bahasa C. Walaupun proses *pickling* serialisasi dari objek struktur data kamus (*dictionary*) menyuntikkan tambahan latensi komunikasi, hal ini tetap menjadi pendekatan yang sangat pragmatis untuk pemrosesan teks berbasis senarai asimetris jika dikompensasi oleh granularitas tugas yang rasional.

III.IV. Evaluasi Kinerja dan Analisis Skalabilitas

Validasi efektivitas pendekatan diukur menggunakan infrastruktur pengujian iteratif otomatis yang mengeksekusi kompilasi kode berulang kali untuk menjamin konsistensi empiris. Variabel manipulasi difokuskan pada manipulasi topologi ukuran klaster paralel

dengan konfigurasi penugasan prosesor $P \in \{1, 2, 4, 8\}$. Tiga parameter fundamental diekstraksi dari instrumen fungsi waktu, yaitu:

a. *Execution time* (waktu penyelesaian eksak) dengan mencatat total durasi komputasi dan transfer memori yang diperlukan suatu arsitektur. Nilai dasar (*baseline*) diambil dari durasi eksekusi menggunakan algoritma serial (T_1), sedangkan kinerja dihitung pada operasi multinoda (T_p).
b. *Speedup acceleration* (rasio akselerasi) yang mengukur kemampuan algoritma mengatasi hambatan Hukum Amdahl, didefinisikan secara komputasional melalui persamaan $S(p) = \frac{T_1}{T_p}$. Akselerasi ini merupakan representasi peningkatan waktu nyata akibat distribusi beban matematis.
c. *System efficiency* (efisiensi sumber daya) yang mengukur optimasi investasi perangkat keras sistem yang dirumuskan $E(p) = \frac{S_{(p)}}{p}$. Efisiensi mencerminkan seberapa berharganya sebuah penambahan inti prosesor terhadap keuntungan kecepatan. Persentase rendah menandakan bahwa arsitektur telah melampaui puncak optimum operasional dan bergeser menuju stagnasi komputasi.

Metodologi yang sistematis, transparan, dan diarsiteki dengan teliti ini pada akhirnya memungkinkan eksplorasi presisi dalam memetakan batasan operasional dari algoritma linguistik fundamental di bawah naungan arsitektur Message Passing Interface (MPI).

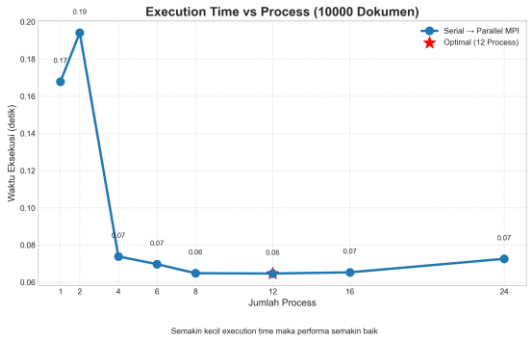
IV. HASIL DAN PEMBAHASAN

Pengujian performa dilakukan secara menyeluruh untuk membandingkan efektivitas komputasi serial dan paralel dalam menyelesaikan ekstraksi fitur TF-IDF pada berbagai skala korpus. Eksperimen ini menggunakan satu korpus yang dikembangkan dari *dataset* gabungan teks berbasis spam, dengan lima variasi ukuran dokumen yang meningkat secara progresif, yaitu 10.000, 25.000, 50.000, 75.000, dan 100.000 dokumen. Setiap ukuran diuji menggunakan tujuh konfigurasi proses paralel, yakni $P \in \{2, 4, 6, 8, 12, 16, 24\}$, serta satu konfigurasi serial sebagai *baseline*. Setiap kombinasi dijalankan sebanyak lima kali percobaan, dan nilai rata-ratanya dijadikan representasi performa yang stabil secara statistik. Tiga metrik utama diekstrak secara otomatis dari setiap konfigurasi, seperti waktu eksekusi rata-rata, rasio percepatan atau *speedup* (*S = T_serial / T_paralel*), serta efisiensi sistem (*E = S / P*). Ketiga metrik ini dianalisis secara bersamaan untuk mendapatkan gambaran menyeluruh mengenai kondisi operasional optimal komputasi paralel MPI pada setiap skala data.

IV.I. Analisis Waktu Eksekusi per Skala *Dataset*

A. Korpus 10.000 Dokumen

Pada skala 10.000 dokumen, eksekusi serial mencatatkan waktu rata-rata 0,168 detik. Konfigurasi 2 proses justru menghasilkan waktu yang lebih panjang dari serial, yakni 0,194 detik, sehingga nilai *speedup*-nya hanya 0,86 yang artinya konfigurasi ini 13,5% lebih lambat dari proses tunggal. Ini adalah indikasi yang sangat jelas bahwa pada skala data ini, biaya inisialisasi proses MPI dan komunikasi antar-proses melalui operasi *scatter*, *gather*, dan *broadcast* belum sebanding dengan porsi komputasi yang berhasil didistribusikan. Sebaliknya, lonjakan performa terjadi secara dramatis begitu jumlah proses naik ke konfigurasi 4 dengan waktu eksekusi turun tajam ke 0,074 detik, menghasilkan *speedup* 2,27×. Konfigurasi 6 proses (0,070 detik, *speedup* 2,41×) dan 8 proses (0,065 detik, *speedup* 2,59×) meneruskan penurunan tersebut secara konsisten. Konfigurasi 12 proses mencapai titik optimal dengan waktu 0,065 detik dan *speedup* tertinggi sebesar 2,60×. Setelah titik ini, konfigurasi 16 proses sedikit melambat ke 0,065 detik (*speedup* 2,57×), dan konfigurasi 24 proses memperlihatkan degradasi yang lebih nyata ke 0,073 detik (*speedup* 2,31×).

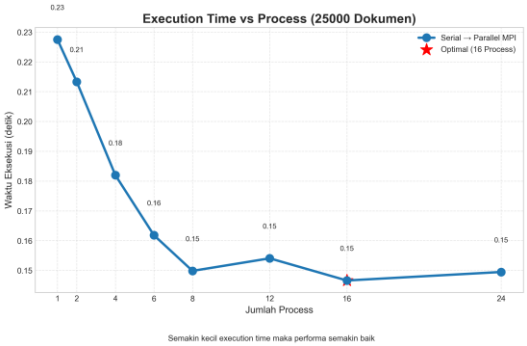


Gambar 1. *Execution Time* vs *Process* (10.000 Dokumen)

Pola pada skala 10.000 dokumen ini mengungkap fenomena yang penting: terdapat titik puncak *speedup* di *P*=12, diikuti oleh penurunan performa pada *P*=16 dan *P*=24. Fenomena penurunan setelah titik puncak ini merupakan karakteristik *oversubscription*, kondisi ketika jumlah proses melebihi kapasitas optimal perangkat keras untuk *dataset* berukuran kecil. Pada *P*=24 dengan hanya 10.000 dokumen, setiap proses rata-rata hanya menanggung sekitar 417 dokumen. Porsi ini terlalu ringan untuk mengimbangi biaya koordinasi dari 24 proses yang harus saling berkomunikasi, sehingga sistem lebih banyak menghabiskan waktu untuk sinkronisasi daripada komputasi TF itu sendiri. Perlu dicatat bahwa *speedup* tertinggi secara absolut dalam seluruh pengujian ini (2,60×) justru diraih pada skala terkecil dengan konfigurasi *P*=12 sebuah temuan yang menunjukkan bahwa pada *dataset* berukuran kecil, distribusi ke sejumlah proses yang moderat dapat mengeksploitasi sumber daya komputasi secara sangat efisien selama beban per proses masih cukup substansial.

B. Korpus 25.000 Dokumen

Pada 25.000 dokumen, eksekusi serial membutuhkan 0,227 detik. Konfigurasi 2 proses berhasil sedikit mengalahkan serial dengan 0,213 detik, namun *speedup*-nya yang hanya 1,07× masih terlalu kecil untuk dianggap sebagai keuntungan paralel yang berarti. Tren penurunan waktu berlanjut ke konfigurasi 4 proses (0,182 detik, *speedup* 1,25×) dan 6 proses (0,162 detik, *speedup* 1,41×). Konfigurasi 8 proses mencatatkan 0,150 detik (*speedup* 1,52×). Pada konfigurasi 12 proses, terjadi sedikit fluktuasi nonmonoton dengan waktu yang sedikit naik ke 0,154 detik (*speedup* 1,48×), sebelum kembali turun secara mulus ke 0,147 detik pada konfigurasi 16 proses (*speedup* 1,55×) yang menjadi konfigurasi optimal untuk skala ini. Konfigurasi 24 proses sedikit naik ke 0,149 detik (*speedup* 1,52×).



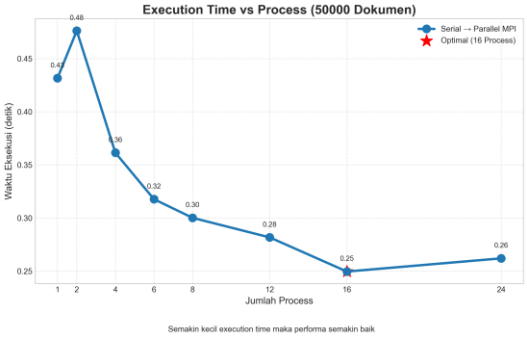
Gambar 2. *Execution Time* vs *Process* (25.000 Dokumen)

Fluktuasi kecil pada konfigurasi 12 proses di skala 25.000 dokumen ini patut mendapat perhatian analitis. Kenaikan waktu yang kecil dari *P*=8 ke *P*=12 (0,150 detik ke 0,154 detik) sebelum kembali turun di *P*=16 mengindikasikan adanya interferensi penjadwalan proses sistem operasi pada titik tertentu. Pada sistem berbasis Windows dengan 24 *logical core*, pembagian 12 proses kepada *scheduler* dapat menciptakan pola *context switch* yang kurang optimal dibandingkan konfigurasi 8 atau 16 proses. Meskipun demikian, selisih absolut antar konfigurasi di rentang *P*=8 hingga *P*=24 sangatlah kecil (hanya sekitar 7 milidetik), yang menunjukkan bahwa sistem telah memasuki zona *plateau*, sebuah kondisi saturasi ringan yang menandakan skalabilitas sudah mendekati batas untuk ukuran data ini. Titik optimal *P*=16 dengan *speedup* 1,55× memberikan reduksi waktu yang berarti dibanding serial sambil mempertahankan rasio komunikasi-terhadap-komputasi yang masih efisien.

C. Korpus 50.000 Dokumen

Skala 50.000 dokumen kembali memperlihatkan fenomena konfigurasi 2 proses yang lebih lambat dari serial. Serial membutuhkan 0,432 detik, sementara *P*=2 mencatatkan 0,476 detik dengan *speedup* 0,91× atau 10,3% lebih lambat dari proses tunggal. Ini mengonfirmasi bahwa ketidakefektifan konfigurasi 2 proses bukan terbatas pada skala kecil, melainkan muncul kembali ketika beban per proses terlalu besar untuk ditangani dua partisi secara efisien

tanpa biaya komunikasi yang signifikan. Tren penurunan berlanjut konsisten dari *P*=4 (0,361 detik, *speedup* 1,19×) hingga *P*=12 (0,282 detik, *speedup* 1,53×). Konfigurasi 16 proses mencapai titik optimal dengan 0,250 detik (*speedup* 1,73×). Konfigurasi 24 proses sedikit naik ke 0,262 detik (*speedup* 1,65×), menandakan sistem mulai memasuki batas efisiensinya untuk ukuran data ini.

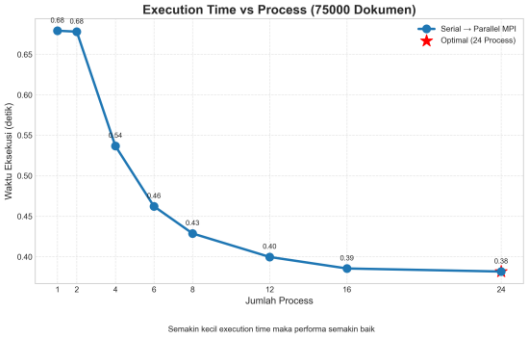


Gambar 3. *Execution Time* vs *Process* (50.000 Dokumen)

Profil kurva pada 50.000 dokumen menggambarkan manifestasi Hukum Amdahl secara gamblang. Meskipun setiap tambahan proses terus memangkas waktu eksekusi dari *P*=4 hingga *P*=16, laju penurunannya semakin landai dan grafik membentuk kurva cembung yang tipikal dari distribusi beban paralel. Penurunan dari serial (0,432 detik) ke *P*=4 (0,361 detik) menghasilkan reduksi 16,4%, sementara penurunan dari *P*=4 ke *P*=8 hanya 16,8%, dari *P*=8 ke *P*=12 sebesar 6,1%, dan dari *P*=12 ke *P*=16 sebesar 11,4%. Penurunan yang tidak seragam ini mencerminkan bahwa fraksi sekuensial dari algoritma, terutama fase agregasi kamus *document frequency* global dan penyiaran nilai IDF ke seluruh proses tetap konstan dan tidak tereduksi meskipun proses paralel ditambah, sesuai dengan prediksi teoritis Hukum Amdahl. Kenaikan tipis pada *P*=24 (0,262 detik) mengindikasikan titik jenuh yang mulai terbentuk pada konfigurasi ini untuk skala 50.000 dokumen.

D. Korpus 75.000 Dokumen

Pada 75.000 dokumen, terjadi pergeseran pola yang sangat menarik pada konfigurasi 2 proses. Serial mencatatkan 0,679 detik, dan *P*=2 menghasilkan 0,678 detik dengan *speedup* 1,0017×, secara praktis identik dengan eksekusi serial, atau berada tepat di titik impas (*break-even point*). Ini adalah demonstrasi empiris langsung dari kondisi keseimbangan antara keuntungan distribusi komputasi dan biaya komunikasi yang saling menganulir. Setelah titik impas ini, kurva turun secara konsisten: *P*=4 (0,537 detik, *speedup* 1,27×), *P*=6 (0,462 detik, 1,47×), *P*=8 (0,429 detik, 1,58×), *P*=12 (0,400 detik, 1,70×), *P*=16 (0,385 detik, 1,76×), dan *P*=24 (0,382 detik, *speedup* 1,78×) sebagai konfigurasi optimal.



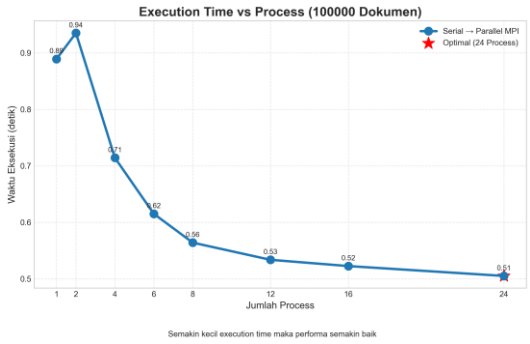
Gambar 4. *Execution Time* vs *Process* (75.000 Dokumen)

Seluruh konfigurasi mulai dari *P*=4 ke atas secara definitif mengungguli serial pada skala 75.000 dokumen, dengan margin keunggulan yang terus membesar seiring penambahan proses. Berbeda dari dua skala sebelumnya (50.000 dokumen) yang menunjukkan sedikit kenaikan di *P*=24, pada skala 75.000 dokumen kurva masih terus menurun secara monoton hingga konfigurasi terakhir yang diuji (*P*=24). Penurunan dari *P*=16 ke *P*=24 memang sangat kecil (0,385 detik ke 0,382 detik, hanya 3 milidetik), tetapi arahnya masih turun yang menjadi sebuah indikasi kuat bahwa titik saturasi sesungguhnya untuk ukuran data ini berada di luar rentang konfigurasi yang diuji. Ini membuka proyeksi bahwa konfigurasi *P*=32 atau lebih tinggi masih berpotensi memberikan keuntungan tambahan yang terukur.

E. Korpus 100.000 Dokumen

Pada skala tertinggi, 100.000 dokumen, pola P=2 yang lebih lambat dari serial kembali muncul. Serial membutuhkan 0,889 detik, sementara *P*=2 menghasilkan 0,935 detik (*speedup* 0,95×) atau 5,2% lebih lambat. Tren penurunan yang konsisten kemudian terbentuk *P*=4 (0,714 detik, *speedup* 1,24×), P=6 (0,615 detik, 1,45×), *P*=8 (0,564 detik, 1,58×), *P*=12 (0,534 detik, 1,67×), *P*=16 (0,522 detik, 1,70×), hingga P=24 (0,505 detik, *speedup* 1,76×).

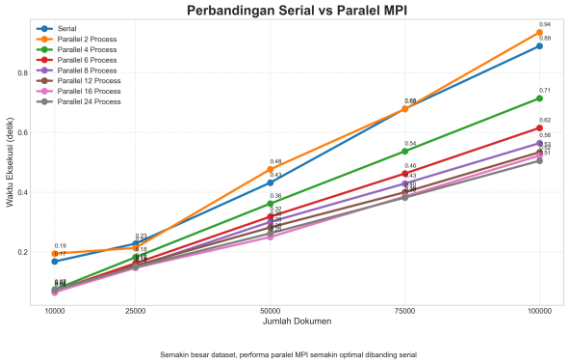
Konfigurasi 24 proses menjadi yang terbaik untuk skala ini, namun kurva penurunan masih terus berlanjut tanpa tanda-tanda saturasi hingga di ujung konfigurasi yang diuji.



Gambar 5. *Execution Time* vs *Process* (100.000 Dokumen)

Karakteristik yang paling menonjol pada skala 100.000 dokumen adalah bahwa grafik tidak memperlihatkan titik balik atau *leveling off* hingga *P*=24 yang berbeda dari skala 10.000 dokumen yang jelas memperlihatkan degradasi di *P*>12. Setiap tambahan proses masih menghasilkan reduksi waktu yang positif, meskipun laju penurunannya semakin kecil sesuai dengan prinsip *diminishing returns*. Penurunan dari *P*=12 ke *P*=16 hanya menghasilkan penghematan 12 milidetik, dan dari *P*=16 ke *P*=24 hanya 17 milidetik, angka yang kecil secara absolut namun tetap positif secara arahnya. Ini menunjukkan bahwa pada skala 100.000 dokumen, perangkat keras masih belum dieksploitasi secara maksimal pada *P*=24, dan kemungkinan besar konfigurasi yang lebih tinggi masih dapat memberikan manfaat tambahan.

IV.II. Tinjauan Komparatif Lintas Seluruh Skala



Gambar 6. Perbandingan Serial vs Paralel MPI pada Seluruh Skala Data

Grafik komparatif pada gambar tersebut menyajikan perspektif menyeluruh tentang bagaimana setiap konfigurasi proses merespons pertambahan volume data dari 10.000 hingga 100.000 dokumen. Beberapa pola struktural yang sangat informatif dapat dibaca dari visualisasi ini. Garis serial (biru) memperlihatkan pertumbuhan yang paling curam dan konsisten, yang mengonfirmasi bahwa tanpa distribusi beban, waktu komputasi tumbuh secara hampir linear proporsional terhadap ukuran data. Garis konfigurasi 2 proses (oranye) bergerak sejajar dengan serial bahkan sesekali melampaui garis serial menunjukkan inefisiensi konfigurasi ini secara sistematis di berbagai skala. Mulai dari *P*=4 ke atas, seluruh konfigurasi paralel berada di bawah garis serial dengan jarak yang semakin melebar seiring bertambahnya data, sebuah bukti visual langsung dari skalabilitas sistem MPI. Konfigurasi dengan jumlah proses lebih tinggi secara konsisten memiliki gradien pertumbuhan waktu yang lebih rendah, artinya setiap tambahan dokumen ke korpus menghasilkan penambahan waktu yang lebih kecil dibandingkan konfigurasi dengan proses lebih sedikit.

Satu observasi penting dari grafik ini adalah persilangan garis-garis yang terjadi di sekitar skala 25.000 dokumen. Pada 10.000 dokumen, perbedaan waktu antarkonfigurasi paralel sangat besar (*P*=4 berada di 0,074 detik sementara *P*=24 di 0,073 detik yang menunjukkan hampir setara). Namun, seiring skala data bertambah, garis-garis konfigurasi tinggi (*P*=12, *P*=16, *P*=24) mulai terpisah lebih jauh dari konfigurasi rendah (*P*=4, *P*=6), mengindikasikan bahwa manfaat relatif dari konfigurasi tinggi semakin menonjol pada data yang lebih besar. Pada 100.000 dokumen, perbedaan antara *P*=4 (0,714 detik) dan *P*=24 (0,505 detik) sudah mencapai selisih 209 milidetik yang cukup signifikan untuk aplikasi pemrosesan teks skala industri.

IV.III. Analisis *Speedup*, Efisiensi, dan Identifikasi Konfigurasi Optimal

Tabel 1 merangkum konfigurasi dengan performa terbaik beserta nilai *speedup* dan efisiensi yang dicapai pada setiap ukuran *dataset*.

Tabel 1. Rekapitulasi Konfigurasi Optimal per Skala *Dataset*

Ukuran *Dataset*	Konfigurasi Optimal	Waktu Eksekusi (s)	*Speedup*	Efisiensi
10.000 dokumen	12 Proses	0,065	2,60×	0,2163
25.000 dokumen	16 Proses	0,147	1,55×	0,0970
50.000 dokumen	16 Proses	0,250	1,73×	0,1080
75.000 dokumen	24 Proses	0,382	1,78×	0,0741
100.000 dokumen	24 Proses	0,505	1,76×	0,0734

Pola pergeseran konfigurasi optimal yang terlihat pada **Tabel 1** merupakan temuan paling penting dari eksperimen ini. Pada 10.000 dokumen, konfigurasi optimal berada di P=12. Pada 25.000 dan 50.000 dokumen, titik optimalnya bergeser ke P=16. Untuk 75.000 dan 100.000 dokumen, konfigurasi P=24 menjadi yang terbaik. Pergeseran ini tidak terjadi secara acak, melainkan mengikuti logika granularitas tugas yang sangat sistematis. Setiap peningkatan konfigurasi optimal berlanjut seiring dengan pertambahan beban dokumen per proses yang cukup besar untuk mengimbangi tambahan biaya komunikasi. Pada P=12 untuk 10.000 dokumen, setiap proses menanggung sekitar 833 dokumen, porsi yang optimal untuk perangkat keras yang digunakan. Ketika data bertambah ke 25.000 dokumen, beban per proses dengan P=12 naik menjadi sekitar 2.083 dokumen, dan P=16 dengan 1.563 dokumen per proses ternyata memberikan keseimbangan yang lebih baik. Pada 75.000 dan 100.000 dokumen, konfigurasi P=24 dengan 3.125 dan 4.167 dokumen per proses masing-masing mencapai rasio komputasi-terhadap-komunikasi yang paling menguntungkan.

Konsistensi fenomena konfigurasi 2 proses yang tidak efektif patut dianalisis lebih dalam. Dari lima skala yang diuji, P=2 menghasilkan *speedup* di bawah 1,0 pada tiga skala (10.000, 50.000, 100.000 dokumen), hampir tepat di titik impas pada 75.000 dokumen, dan hanya memberikan akselerasi sangat kecil (1,07×) pada 25.000 dokumen. Akar permasalahannya terletak pada karakteristik unik dari beban komunikasi MPI dalam konteks TF-IDF. Dalam algoritma ini, operasi *gather* untuk mengumpulkan kamus *document frequency* dari semua proses pekerja melibatkan transmisi struktur data kamus Python berukuran besar yang ukurannya ditentukan bukan oleh jumlah dokumen per proses, melainkan oleh kekayaan kosakata unik dalam seluruh korpus. Dengan hanya 2 proses, meskipun masing-masing proses hanya membaca setengah dokumen, setiap proses tetap harus mengirim kamus kosakata yang cukup besar ke proses *root*. Beban komunikasi ini tidak berkurang secara proporsional dengan pengurangan jumlah proses, sehingga keuntungan dari pembagian komputasi TF tidak cukup besar untuk mengimbangi biaya sinkronisasi tersebut.

Dari sudut pandang efisiensi sistem, **Tabel 1** memperlihatkan tren penurunan yang konsisten seiring bertambahnya jumlah proses sebuah karakteristik yang selaras sempurna dengan prediksi teoritis. Pada konfigurasi optimal P=12 untuk 10.000 dokumen, efisiensi mencapai 0,2163, artinya setiap proses yang ditambahkan secara rata-rata berkontribusi sebesar 21,6% dari kapasitas maksimum idealnya. Nilai ini menurun ke 0,0970 pada P=16 (25.000 dokumen) dan lebih jauh ke 0,0741 pada P=24 (75.000 dokumen). Penurunan efisiensi yang monoton ini merupakan konsekuensi langsung dari Hukum Amdahl, fraksi sekuensial algoritma yang tidak dapat diparalelkan, tetap memakan waktu absolut yang sama terlepas dari berapa banyak proses yang bekerja secara paralel. Implikasinya bagi pemilihan konfigurasi praktis adalah bahwa penambahan proses di luar batas optimal tidak direkomendasikan jika nilai efisiensi sudah sangat rendah, kecuali jika volume data juga turut bertambah secara proporsional untuk mengisi kapasitas komputasi tambahan tersebut.

Tren *speedup* yang dipetakan dari Tabel 1 memperlihatkan pola yang tidak sepenuhnya linear terhadap ukuran data. Nilai *speedup* tertinggi (2,60×) justru diraih pada ukuran data terkecil (10.000 dokumen) dengan konfigurasi P=12, bukan pada *dataset* terbesar. Hal ini disebabkan oleh rasio kosakata-terhadap-dokumen yang lebih rendah pada *dataset* kecil, sehingga biaya operasi *gather* dan *broadcast* relatif lebih ringan terhadap beban komputasi TF lokal. Pada skala yang lebih besar, kosakata unik tumbuh lebih cepat daripada jumlah dokumen, sehingga biaya komunikasi agregasi kamus meningkat dan menghambat laju akselerasi. Meski demikian, nilai *speedup* pada skala 50.000–100.000 dokumen tetap stabil di kisaran 1,73×–1,78×, menunjukkan bahwa sistem mempertahankan tingkat akselerasi yang konsisten dan bermakna pada rentang data yang lebih besar. Stabilitas ini, ditambah dengan fakta bahwa kurva waktu eksekusi pada 75.000 dan 100.000 dokumen masih terus menurun pada

P=24 tanpa tanda saturasi, mengindikasikan bahwa kapasitas skalabilitas sistem MPI ini belum mencapai batasnya pada rentang konfigurasi yang diuji, membuka potensi peningkatan akselerasi lebih lanjut apabila jumlah proses dan skala data ditingkatkan secara bersamaan pada infrastruktur komputasi yang lebih besar.

V. KESIMPULAN DAN SARAN

V.I. Kesimpulan

Eksperimen dan *research* ini berhasil merancang dan mengimplementasikan algoritma TF-IDF paralel berbasis Message Passing Interface menggunakan pustaka *mpi4py*, serta membandingkan kinerjanya terhadap eksekusi serial pada lima skala korpus dan delapan konfigurasi proses. Hasil pengujian menunjukkan bahwa implementasi paralel secara umum berhasil mengatasi kemacetan eksekusi berurutan pada algoritma TF-IDF, tetapi besarnya keuntungan yang diperoleh sangat bergantung pada keseimbangan antara granularitas beban kerja per proses dan biaya komunikasi kolektif MPI. Konfigurasi dua proses terbukti secara konsisten tidak efektif karena beban komunikasi kamus *document frequency* tidak menyusut sebanding dengan pengurangan jumlah proses, sehingga pada tiga dari lima ukuran data justru lebih lambat daripada eksekusi serial. Konfigurasi optimal bergeser secara sistematis seiring pertambahan ukuran data, dari dua belas proses pada korpus 10.000 dokumen yang menghasilkan *speedup* tertinggi sebesar 2,60 kali, ke enam belas proses pada korpus 25.000 dan 50.000 dokumen, hingga dua puluh empat proses pada korpus 75.000 dan 100.000 dokumen dengan *speedup* yang stabil pada kisaran 1,73 hingga 1,78 kali. Pola penurunan efisiensi yang monoton seiring penambahan proses sejalan dengan Hukum Amdahl, yang menegaskan bahwa fraksi komputasi sekuensial pada agregasi kamus global dan penyiaran nilai *inverse document frequency* tetap menjadi batas struktural terhadap akselerasi yang dapat dicapai.

V.II. Saran

Temuan ini memberikan kontribusi teoretis berupa bukti empiris mengenai paradoks granularitas dalam pemrosesan teks terdistribusi, sekaligus memperkaya pemahaman tentang interaksi antara volume data, jumlah proses, dan biaya komunikasi pada arsitektur MPI. Secara praktis, hasil penelitian ini dapat menjadi acuan bagi pengembang sistem analitik teks dalam menentukan jumlah proses yang proporsional terhadap volume data agar investasi sumber daya komputasi memberikan manfaat akselerasi yang nyata, sekaligus menegaskan bahwa konfigurasi dua proses sebaiknya dihindari untuk tugas serupa. Mengingat kurva waktu eksekusi pada korpus 75.000 dan 100.000 dokumen belum menunjukkan tanda saturasi pada dua puluh empat proses, penelitian selanjutnya disarankan untuk menguji konfigurasi proses yang lebih tinggi serta volume data yang lebih besar guna menemukan titik jenuh sesungguhnya, dan dapat pula mengeksplorasi pendekatan komunikasi *non-blocking* atau model hibrida MPI dengan OpenMP untuk menekan biaya komunikasi lebih lanjut.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Heri Wijayanto S.T., M.T., Ph.D selaku dosen pengampu mata kuliah Pemrosesan Paralel di Program Studi Teknik Informatika, Universitas Mataram, atas bimbingan, arahan, dan masukan yang sangat berharga selama proses penyusunan penelitian ini. Apresiasi juga disampaikan kepada seluruh anggota kelompok atas kontribusi dan kerja sama yang terjalin dengan baik dalam penyelesaian artikel penelitian ini.

DAFTAR PUSTAKA

[1] A. Bagheri, A. Giachanou, P. Mosteiro, and S. Verberne, “Natural Language Processing and Text Mining (Turning Unstructured Data into Structured) BT - Clinical Applications of Artificial Intelligence in Real-World Data,” F. W. Asselbergs, S. Denaxas, D. L. Oberski, and J. H. Moore, Eds., Cham: Springer International Publishing, 2023, pp. 69–93. doi: 10.1007/978-3-031-36678-9_5.

[2] K. P. Harmandini and K. M. L, “Analysis of TF-IDF and TF-RF Feature Extraction on Product Review Sentiment,” *Sinkron*, vol. 8, no. 2, pp. 929–937, 2024, doi: 10.33395/sinkron.v8i2.13376.

[3] M. Das, S. Kamalanathan, and P. Alphonse, “A Comparative Study on TF-IDF feature weighting method and

its analysis using unstructured dataset," *CEUR Workshop Proc.*, vol. 2870, pp. 98–107, 2021.

[4] V. Korkhov *et al.*, "Finding Bottlenecks in Message Passing Interface Programs by Scalable Critical Path Analysis," *Algorithms*, vol. 16, no. 11, 2023, doi: 10.3390/a16110505.

[5] U. Pujianto and A. Y. Wijaya, "Pemilihan Korpus Statis Bersesuaian dengan Cosine Similarity dan Penggunaan IDF Global Pada Penambahan Dokumen Baru," *E-Link J. Tek. Elektro dan Inform.*, vol. 14, no. 2, p. 1, 2020, doi: 10.30587/e-link.v14i2.1215.

[6] Y. Wu, Y. Xiang, J. Ge, and P. Muller, "High-Performance Computing for Big Data Processing," *Futur. Gener. Comput. Syst.*, vol. 88, pp. 693–695, 2018, doi: https://doi.org/10.1016/j.future.2018.07.054.

[7] M. Shukla, D. Dharme, P. Ramnarain, R. Dos Santos, and C.-T. Lu, " DIGDUG: Scalable Separable Dense Graph Pruning and Join Operations in MapReduce ," *IEEE Trans. Big Data*, vol. 7, no. 06, pp. 930–951, 2021, doi: 10.1109/TBDATA.2020.2983650.

[8] J. Han, J. Liu, Y. Liu, Y. Shen, D. S. L. Wei, and K. Xue, "Breaking the Communication Bottleneck: A Novel MPI Framework Based on Priority Flow Scheduling for Distributed Scientific Computing Tasks," *IEEE Commun. Mag.*, vol. 63, no. 9, pp. 118–124, 2025, doi: 10.1109/MCOM.006.2400544.

[9] D. Buell, L. Little, K. J. Liszka, and A. N. Sloss, *In Praise of An Introduction to Parallel Programming*. 2011. doi: 10.1016/b978-0-12-374260-5.00012-9.

[10] J. Dongarra *et al.*, "The international exascale software project roadmap," *Int. J. High Perform. Comput. Appl.*, vol. 25, no. 1, pp. 3–60, 2011, doi: 10.1177/1094342010391989.

[11] B. Ramesh *et al.*, "Scalable MPI Collectives using SHARP: Large Scale Performance Evaluation on the TACC Frontera System," in *2020 Workshop on Exascale MPI (ExaMPI)*, 2020, pp. 11–20. doi: 10.1109/ExaMPI52011.2020.00007.

[12] Y. Wang, "Research on the TF–IDF algorithm combined with semantics for automatic extraction of keywords from network news texts," *J. Intell. Syst.*, vol. 33, no. 1, 2024, doi: 10.1515/jisys-2023-0300.

[13] D. Chaurasia, P. V. D. K, and M. Bhatta, "Enhancing Text Summarization through Parallelization: A TF-IDF Algorithm Approach," in *2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, 2024, pp. 1503–1508. doi: 10.1109/ICoICI62503.2024.10696641.

[14] D. Paulsen, Y. Govind, and A. Doan, "Sparkly: A Simple yet Surprisingly Strong TF/IDF Blocker for Entity Matching," *Proc. VLDB Endow.*, vol. 16, no. 6, pp. 1507–1519, 2023, doi: 10.14778/3583140.3583163.

[15] M. Saxena, S. Jha, S. Khan, J. Rodgers, P. Lindner, and E. Gabriel, "Comparison of MPI and spark for data science applications," *Proc. - 2020 IEEE 34th Int. Parallel Distrib. Process. Symp. Work. IPDPSW 2020*, no. 9150426, pp. 682–690, 2020, doi: 10.1109/IPDPSW50202.2020.00123.

[16] L. Dalcin and Y. L. L. Fang, "Mpi4py: Status Update after 12 Years of Development," *Comput. Sci. Eng.*, vol. 23, no. 4, pp. 47–54, 2021, doi: 10.1109/MCSE.2021.3083216.

[17] W. Bland, A. Bouteiller, T. Herault, G. Bosilca, and J. Dongarra, "Post-failure recovery of MPI communication capability: Design and rationale," *Int. J. High Perform. Comput. Appl.*, vol. 27, no. 3, pp. 244–254, 2013, doi: 10.1177/1094342013488238.

[18] A. S. Althiban, H. M. Alharbi, L. A. Al Khuzayem, and F. E. Eassa, "Predicting Software Defects in Hybrid MPI and OpenMP Parallel Programs Using Machine Learning," *Electron.*, vol. 13, no. 1, 2024, doi: 10.3390/electronics13010182.

[19] R. R. A. Siregar, F. A. Sinaga, and R. Arianto, "Aplikasi Penentuan Dosen Penguji Skripsi Menggunakan Metode TF-IDF dan Vector Space Model," *Comput. J. Comput. Sci. Inf. Syst.*, vol. 1, no. 2, p. 171, 2017, doi: 10.24912/computatio.v1i2.1014.

Simulasi Efisiensi Beban Pendingin dan Pemanas pada Bangunan Pintar Berbasis Pemodelan Komputasional

Alif Taftazani Sanjaya [1)], Kamrun Syah Syahidu[1)], Muhammad Mikroju Raseprian Sahid[1)] ,Azkal Arya Habibie[1)], Nisa Aulia Kirani[1)], Herliana Rosika[1)]
Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: aliftaftas.28@gmail.com

Abstrak

Penelitian ini mengembangkan pendekatan probabilistik berbasis simulasi Monte Carlo untuk mengevaluasi distribusi beban pendingin dan pemanas pada bangunan pintar. Menggunakan Energy Efficiency Dataset dari UCI Machine Learning Repository (768 sampel dengan 8 parameter arsitektur), kami membangun surrogate model regresi polinomial derajat dua sebagai fungsi transfer dalam simulasi Monte Carlo dengan Latin Hypercube Sampling (10.000 iterasi). Simulasi menghasilkan interval kepercayaan 95% heating load [8,83–41,22] kWh/m² dan cooling load [12,94–41,67] kWh/m², yang jauh lebih informatif dibanding prediksi deterministik tunggal untuk sizing HVAC. Analisis sensitivitas Spearman mengidentifikasi Overall Height sebagai parameter paling dominan ($\rho \approx 0{,}86$), memungkinkan desainer memfokuskan optimasi pada variabel dengan leverage terbesar. Pendekatan ini memberikan decision support tool berbasis probabilitas yang memungkinkan perancang memperhitungkan risiko dan margin ketidakpastian secara eksplisit dalam desain sistem HVAC.

Kata kunci: bangunan pintar; beban pendingin; beban pemanas; efisiensi energi; simulasi Monte Carlo; kuantifikasi ketidakpastian

Abstract

This study employs a probabilistic approach based on Monte Carlo simulation to characterize the distribution of cooling and heating loads in smart buildings. Utilizing the Energy Efficiency Dataset from the UCI Machine Learning Repository (768 samples with 8 architectural parameters), we constructed a second-degree polynomial regression surrogate model to serve as the transfer function within a Monte Carlo simulation using Latin Hypercube Sampling (10,000 iterations). The simulation yielded 95% confidence intervals of [8.83–41.22] kWh/m² for heating loads and [12.94–41.67] kWh/m² for cooling loads, offering far more insight than single deterministic predictions for HVAC sizing. Spearman sensitivity analysis identified "Overall Height" as the most dominant parameter ($\rho \approx 0.86$), enabling designers to focus optimization efforts on the variable with the greatest leverage. This approach provides a probability-based decision-support tool that allows for the explicit calculation of design risks and performance margins during HAC system design.

Keywords: smart buildings; cooling load; heating load; energy efficiency; Monte Carlo simulation; uncertainty quantification

I. PENDAHULUAN

Sektor bangunan komersial dan residensial mengonsumsi sekitar 40% total energi global, di mana sistem HVAC menyumbang porsi terbesar yaitu 50–60% [1]. Di Indonesia, pesatnya urbanisasi mendorong lonjakan konsumsi energi di sektor ini, yang berdampak langsung pada emisi karbon dan tantangan keberlanjutan. Praktik desain konvensional masih sering mengandalkan metode trial-and-error karena keterbatasan alat prediksi yang praktis pada tahap awal perancangan.

Meskipun smart building menawarkan efisiensi lewat otomatisasi [2], keberhasilannya sangat bergantung pada akurasi estimasi beban termal sejak fase konsep. Penelitian sebelumnya (Tsanas & Xifara, 2012; Aydinalp-Koksal & Ugursal, 2008) telah membangun model prediktif dengan akurasi tinggi ($R^2 > 0{,}95$), namun terbatas pada prediksi deterministik tunggal tanpa menyertakan batas ketidakpastian. Padahal, dalam praktik desain HVAC, insinyur membutuhkan rentang kemungkinan beban untuk menentukan kapasitas sistem dengan margin keamanan yang memadai.

Penelitian ini mengusulkan pendekatan hybrid dengan mengintegrasikan polynomial regression sebagai surrogate model ke dalam simulasi Monte Carlo [4]. Pendekatan ini bertujuan untuk: (1) mengembangkan model prediktif beban pendingin dan pemanas berakurasi tinggi ($R2 \geq 0.95$), (2) mengidentifikasi parameter arsitektur paling berpengaruh melalui analisis sensitivitas Spearman, dan (3) menghasilkan confidence interval 95% beban energi untuk kuantifikasi ketidakpastian. Hasilnya diharapkan menjadi decision support tool bagi arsitek untuk mengevaluasi trade-off desain secara real-time dan akurat.

II. TINJAUAN PUSTAKA

2.1 Efisiensi Energi Bangunan

Efisiensi energi bangunan adalah rasio antara output layanan bangunan dengan input energi yang dikonsumsi [5]. Indikator utama adalah beban pendingin (cooling load) dan beban pemanas (heating load), yang merepresentasikan energi untuk mempertahankan kondisi termal ruangan pada setpoint yang diinginkan. Kedua parameter dipengaruhi oleh karakteristik geometris, material konstruksi, kondisi iklim, dan perilaku penghuni.

2.2 Surrogate Model untuk Simulasi Energi Bangunan

Dalam simulasi Monte Carlo, surrogate model (metamodel) berfungsi sebagai pengganti simulasi fisik yang mahal komputasi [6]. Pendekatan ini memungkinkan evaluasi cepat ribuan skenario tanpa menjalankan EnergyPlus. Tsanas dan Xifara [7] mendemonstrasikan bahwa regresi polinomial dapat mencapai akurasi $R^2 > 0,95$ pada dataset efisiensi bangunan, menjadikannya kandidat kuat sebagai surrogate model. Penelitian ini mengadopsi regresi polinomial derajat dua sebagai fungsi transfer dalam loop Monte Carlo, dengan akurasi tinggi sebagai syarat validasi.

2.3 Parameter Arsitektural dan Beban Termal

Parameter geometris mempengaruhi beban termal melalui mekanisme konduksi, konveksi, dan radiasi. Kompakness relatif (rasio volume terhadap luas permukaan) menentukan rasio envelope terhadap volume yang dikondisikan [9]. Bangunan kompak memiliki rasio surface-to-volume rendah, mengurangi kehilangan panas di musim dingin namun juga mengurangi disipasi panas di musim panas. Proporsi kaca mempengaruhi radiasi solar masuk ke bangunan, sumber panas signifikan terutama pada iklim tropis [10].

2.4 Simulasi Monte Carlo pada Analisis Energi Bangunan

Metode Monte Carlo adalah teknik simulasi komputasional yang menggunakan sampling acak berulang untuk memahami sistem kompleks [11]. Dalam konteks energi bangunan, Monte Carlo memungkinkan kuantifikasi ketidakpastian parameter input secara eksplisit. Fan et al. menerapkan Monte Carlo untuk meningkatkan keandalan prediksi cooling load, menunjukkan bahwa ketidakpastian input (cuaca, parameter bangunan) signifikan mempengaruhi estimasi beban. Wang dan Chen [4] menggunakan Monte Carlo untuk sizing sistem HVAC, membuktikan bahwa pendekatan probabilistik menghasilkan desain lebih andal dibanding deterministik.

Dalam konteks energi bangunan, Monte Carlo memungkinkan kuantifikasi risiko (probabilitas beban melebihi kapasitas HVAC), analisis sensitivitas (identifikasi input paling berpengaruh), dan optimalisasi robust (desain sistem andal dalam berbagai skenario)

III. METODOLOGI PENELITIAN

3.1 Dataset dan Variabel Penelitian

Dataset yang digunakan adalah Energy Efficiency Dataset dari UCI Machine Learning Repository (ID: 242), terdiri dari 768 sampel bangunan residensial dengan delapan variabel prediktor dan dua variabel target. Tabel 1 merangkum variabel-variabel tersebut.

Tabel 1. Variabel Dataset Efisiensi Energi Bangunan

Kode	Variabel	Nilai	Satuan
X1	Relative Compactness	0,62 – 0,98	Adimensional
X2	Surface Area	514,50 – 808,50	m²
X3	Wall Area	245,00 – 416,50	m²
X4	Roof Area	110,25 – 220,50	m²
X5	Overall Height	3,50 – 7,00	m
X6	Orientation	2 – 5	Kategorik
X7	Glazing Area	0,00 – 0,40	Proporsi
X8	Glazing Area Distribution	0 – 5	Kategorik
Y1	Heating Load (Target)	6,01 – 43,10	kWh/m²
Y2	Cooling Load (Target)	10,90 – 48,03	kWh/m²

3.2 Pembangunan Surrogate Model

Model regresi polinomial derajat dua dibangun sebagai fungsi transfer, dilatih dengan 80% data (614 sampel) dan divalidasi pada 20% sisanya (154 sampel). Model mencapai $R^2 = 0,994$ untuk heating load dan 0,968 untuk cooling load (Tabel 1), dengan eksekusi < 0,1 detik per 10.000 evaluasi — memenuhi syarat sebagai surrogate

model yang cepat dan akurat untuk simulasi Monte Carlo.

3.3 Implementasi Monte Carlo dengan LHS

Simulasi Monte Carlo dengan LHS diimplementasikan melalui:

1. Definisi distribusi input Setiap variabel (X1–X8) diberi distribusi probabilistik berdasarkan statistik data aktual (min, max, mean, std).
2. Pembangkitan sampel LHS yaitu N = 10.000 sampel menggunakan Latin Hypercube Sampling [12] untuk cakupan distribusi merata.
3. Evaluasi surrogate model Setiap sampel dievaluasi menggunakan persamaan regresi polinomial untuk heating load ($\hat{Y}_1$) dan cooling load ($\hat{Y}_2$).
4. Analisis output Statistik deskriptif (mean, std, P5, P95), interval kepercayaan 95%, analisis sensitivitas Spearman, dan konvergensi.

Validasi simulasi dilakukan melalui perbandingan distribusi dengan data asli dan plot konvergensi mean-std kumulatif.

3.4 Metrik Evaluasi dan Validasi Simulasi

Kualitas simulasi Monte Carlo dinilai melalui: (1) konvergensi mean dan standar deviasi terhadap jumlah iterasi, (2) lebar interval kepercayaan 95% sebagai ukuran ketidakpastian, dan (3) koefisien variasi (CV = σ/μ) pada 2.000 iterasi terakhir (CV < 1% sebagai indikator stabilitas). Akurasi surrogate model (MAE, RMSE, R^2) digunakan sebagai prasyarat validasi, bukan fokus utama evaluasi.

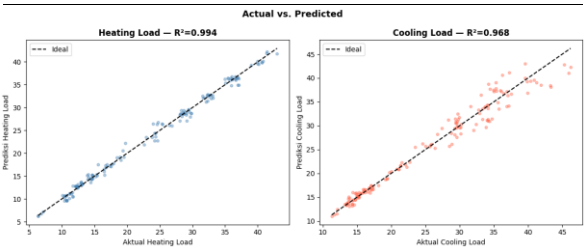
IV. HASIL DAN PEMBAHASAN

4.1 Validasi Surrogate Model

Sebelum simulasi Monte Carlo, surrogate model divalidasi. Regresi polinomial derajat dua mencapai R^2 = 0,994 untuk heating load dan 0,968 untuk cooling load pada data uji (Tabel 1). Plot aktual vs prediksi (Gambar 1) mengikuti garis identitas dengan sangat baik, mengonfirmasi kelayakan model sebagai fungsi transfer dalam loop Monte Carlo.

Tabel 1.Metrik Kinerja Model Regresi Polinomial

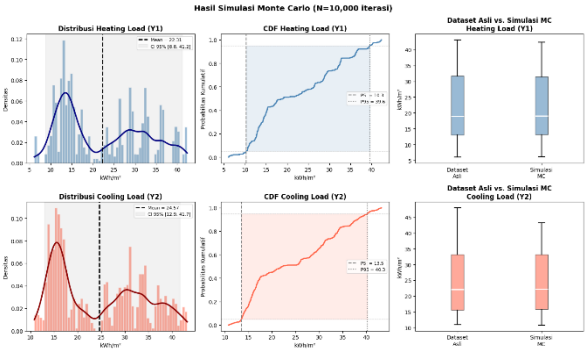
milu	Heating Load (Y1)	Cooling Load (Y2)
MAE (Mean Absolute Error)	0.60	1.19
RMSE (Root Mean Square Error)	0.80	1.73
R^2 (Koefisien Determinasi)	0.994	0.968



Gambar 1. Actual Vs Predicted

Model *Heating Load* mencapai R^2 sebesar 0.994, sementara model *Cooling Load* mencapai R^2 0.968. Nilai ini mengindikasikan bahwa parameter arsitektur bangunan (X1...X8) mampu menjelaskan lebih dari 96% variasi pada kedua beban energi. Plot antara nilai aktual dan prediksi (Gambar 1) menunjukkan sebaran titik yang mengikuti garis identitas (y=x) dengan sangat baik, terutama untuk *Heating Load*, yang mengkonfirmasi bahwa model regresi polinomial layak digunakan sebagai fungsi transfer (*surrogate model*) dalam loop simulasi Monte Carlo. Tingginya akurasi ini sesuai dengan studi Tsanas & Xifara (2012) yang juga mendapatkan $R^2 > 0.95$ pada dataset yang sama.

4.2 Hasil Simulasi Monte Carlo



Gambar 2. Hasil Simulasi Monte Carlo

Simulasi Monte Carlo dengan metode *Latin Hypercube Sampling* (LHS) dilakukan sebanyak 10.000 iterasi menggunakan model regresi polinomial derajat dua sebagai fungsi transfer. Berikut adalah ringkasan statistik hasil simulasi:

Statistik	Heating Load (kWh/m^2)	Cooling Load (kWh/m^2)

Rata-rata (Mean)	22,31	24,57
Standar Deviasi	10,05	9,37
Persentil ke-5 (P5)	10,29	13,50
Persentil ke-95 (P95)	39,60	40,29
Interval Kepercayaan 95%	8,83 – 41,22	12,94 – 41,67

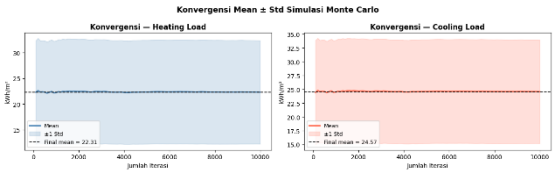
Hasil simulasi menunjukkan rata-rata heating load (22,31 kWh/m²) dan cooling load (24,57 kWh/m²) yang hampir identik dengan data asli (22,31 dan 24,59 kWh/m²), artinya simulasi mampu meniru distribusi data dengan baik. Namun, interval kepercayaan 95% dan deviasi standar sekitar 10 kWh/m² mengindikasikan variasi beban termal yang besar, menandakan bahwa parameter arsitektur sangat berpengaruh terhadap efisiensi energi bangunan.

Gambar 2 menunjukkan bahwa distribusi *heating load* cenderung menceng ke kanan, dengan frekuensi puncak di 10–30 kWh/m² dan ekor panjang hingga >40 kWh/m². Sementara itu, *cooling load* terdistribusi lebih simetris dengan rentang bawah yang lebih sempit (P5 = 13,50 vs 10,29 kWh/m²). Dari kurva CDF di Gambar 6, terlihat misalnya bahwa peluang *heating load* tidak melebihi 39,60 kWh/m² mencapai 95%, yang dapat dijadikan acuan dalam menentukan batas aman desain HVAC.

Validasi simulasi diperkuat oleh Gambar 7, yang membandingkan distribusi data asli (768 sampel) dengan simulasi (10.000 sampel) melalui boxplot. Hasil simulasi sangat mendekati data asli, baik dari median, rentang antarkuartil, maupun nilai ekstremnya (*heating load*: 6,01–43,10 kWh/m²; *cooling load*: 10,90–48,03 kWh/m²). Ini membuktikan bahwa metode LHS dengan resampling dari baris data asli mampu mempertahankan korelasi antarvariabel, sehingga setiap kombinasi parameter yang dihasilkan tetap realistis secara fisik.

4.3 Analisis Konvergensi

Untuk memastikan bahwa jumlah iterasi (N=10.000) cukup untuk mencapai stabilitas statistik, dilakukan analisis konvergensi. Gambar 3 menunjukkan plot nilai rata-rata (mean) dan standar deviasi kumulatif terhadap jumlah iterasi.

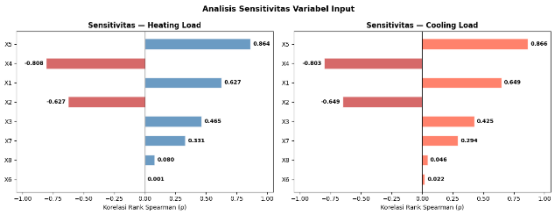


Gambar 3. Konvergensi Mean +- Std Simulasi Monte Carlo

Setelah sekitar 2.000 iterasi, rata-rata dan standar deviasi kedua beban mulai menunjukkan kestabilan, dan pada 10.000 iterasi kurva mencapai *plateau* tanpa fluktuasi berarti. Hal ini menegaskan bahwa jumlah sampel N=10.000 dengan metode LHS sudah cukup untuk menghasilkan estimasi yang konvergen dan andal..

4.4 Analisis Sensivitas

Untuk mengidentifikasi parameter arsitektur yang paling berpengaruh terhadap beban energi, dilakukan analisis sensitivitas menggunakan korelasi rank Spearman antara setiap variabel input (X1-X8) terhadap output simulasi (Heating & Cooling Load). Hasilnya disajikan pada Gambar 4.



Gambar 4. Analisis Sensitivitas Spearman

Berdasarkan analisis sensitivitas Spearman Gambar 4, dua parameter utama yang paling dominan menentukan beban energi adalah Overall Height (X5) dengan korelasi positif tertinggi (ρ = 0,866 untuk cooling, ρ = 0,864 untuk heating), mengindikasikan bahwa peningkatan tinggi bangunan signifikan menaikkan beban termal akibat volume ruang yang lebih besar dan luas selubung bangunan yang bertambah. Kedua, Roof Area (X4) menunjukkan korelasi negatif yang kuat (ρ = -0,808, ρ = -0,803), artinya semakin luas atap, semakin rendah beban pemanas dan pendingin karena rasio luas permukaan terhadap volume yang lebih rendah. Parameter lain seperti Surface Area (X2), Relative Compactness (X1), Wall Area (X3), dan Glazing Area (X7) menunjukkan pengaruh yang lebih rendah dengan |ρ| berkisar antara 0,29 hingga 0,65, sementara Orientation (X6) dan Glazing Distribution (X8)

memiliki pengaruh yang dapat diabaikan ($\rho < 0,10$). Hal ini mengindikasikan bahwa pada rentang variabilitas dataset, faktor geometris fundamental (tinggi, bentuk, dan proporsi) jauh lebih kritis terhadap efisiensi energi bangunan dibandingkan detail spesifik seperti orientasi atau distribusi kaca.

4.5 Implikasi Desain

Hasil simulasi mengkonfirmasi teori fisika bangunan bahwa tinggi bangunan dan kompakness adalah kunci efisiensi energi. Untuk mencapai beban pendingin dan pemanas yang rendah (kuartil bawah distribusi), rekomendasi desain yang dapat diambil adalah:

1. Meminimalkan tinggi bangunan (preferensi 3.5m daripada 7.0m) untuk volume ruang yang lebih kecil, yang terbukti menurunkan beban termal secara signifikan.
2. Meningkatkan nilai Relative Compactness (mendekati 1.0) dengan merancang bentuk bangunan yang lebih mendekati kubus atau bola, sehingga meminimalisir luasan dinding eksterior yang berfungsi sebagai jalur transmisi panas.
3. Memperhatikan desain atap (Roof Area) karena memiliki korelasi negatif kuat terhadap beban energi, meskipun faktor ini perlu dipertimbangkan bersama aspek struktural dan estetika bangunan.
4. Parameter seperti orientasi dan distribusi kaca memiliki pengaruh minimal pada rentang dataset, sehingga prioritas desain sebaiknya difokuskan pada faktor geometris fundamental.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan simulasi komputasional berbasis Monte Carlo dengan *Latin Hypercube Sampling* untuk memprediksi distribusi probabilitas beban pendingin dan pemanas bangunan. Menggunakan *Energy Efficiency Dataset* yang terdiri dari 768 sampel bangunan, beberapa temuan utama dapat disimpulkan:

1. Model regresi polinomial derajat dua menghasilkan akurasi tinggi dengan $R^2 = 0,994$ untuk *heating load* dan $R^2 = 0,968$ untuk *cooling load*, memvalidasi kualitas fungsi transfer deterministik yang digunakan dalam loop Monte Carlo.
2. Simulasi Monte Carlo dengan 10.000 iterasi LHS berhasil menghasilkan distribusi lengkap beban termal, dengan interval kepercayaan 95% *heating load* sebesar [8,83 – 41,22] kWh/m² dan *cooling load* sebesar [12,94 – 41,67] kWh/m².
3. Analisis sensitivitas mengidentifikasi Overall Height (X5) sebagai variabel paling dominan ($\rho \approx 0,86$), diikuti oleh Surface Area (X2) dan Relative Compactness (X1) dengan $|\rho| \approx 0,63–0,65$, serta Roof Area (X4) dengan korelasi negatif kuat ($\rho \approx -0,80$). Temuan ini menegaskan bahwa faktor geometris fundamental memiliki pengaruh jauh lebih besar dibanding parameter detail seperti orientasi dan distribusi kaca.
4. Pendekatan stokastik terbukti lebih informatif untuk desain sistem HVAC dibandingkan prediksi deterministik tunggal, memungkinkan perancang memperhitungkan risiko dan margin ketidakpastian secara eksplisit, terutama dalam menentukan kapasitas sistem pada kondisi ekstrem (P95).

Penelitian selanjutnya dapat mengeksplorasi penggunaan model machine learning yang lebih kompleks sebagai surrogate model, integrasi data cuaca real-time, serta eksplorasi variabel iklim mikro dan material bangunan untuk memperkaya cakupan analisis.

VI. UCAPAN TERIMAKASIH

Ucapan terima kasih kami sampaikan kepada Program Studi Teknik Informatika Universitas Mataram atas fasilitas dan bimbingan selama penelitian, serta kepada UCI Machine Learning Repository yang menyediakan Energy Efficiency Dataset. Tak lupa, kami juga berterima kasih kepada dosen pembimbing dan semua pihak atas masukan berharganya.

VII. DAFTAR PUSTAKA

[1] A. Tsanas dan A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," Energy and Buildings, vol. 49, hlm. 560–567, 2012.

[2] C. Fan, Y. Liao, G. Zhou, X. Zhou, dan Y. Ding, "Improving cooling load prediction reliability for HVAC system using Monte-Carlo simulation to deal with uncertainties in input variables," Energy and Buildings, vol. 241, art. no. 110932, 2021. doi: 10.1016/j.enbuild.2021.110932

[3] S. Alghamdi, W. Tang, S. Kanjanabootra, dan D. Alterman, "Optimising building energy and comfort predictions with intelligent computational model," Sustainability, vol. 16, no. 8, art. no. 3432, 2024. doi: 10.3390/su16083432

[4] Y. H. Wang dan Y. C. Chen, "Exploration of HVAC system sizing based on building performance simulation and Monte Carlo method," dalam Proc. The 11th International Conference on Indoor Air Quality, Ventilation & Energy Conservation in Buildings (IAQVEC 2023), E3S Web of Conferences, vol. 396, art. no. 03007, 2023. doi: 10.1051/e3sconf/202339603007

[5] F. Tahmasebinia, R. Jiang, S. Sepasgozar, J. Wei, Y. Ding, dan H. Ma, "Implementation of BIM energy analysis and Monte Carlo simulation for estimating building energy performance based on regression approach: A case study," Buildings, vol. 12, no. 4, art. no. 449, 2022. doi: 10.3390/buildings12040449

[6] J. Xiao, "The application of predictive stochastic model based on Monte Carlo method in building energy saving renovation," Archives of Civil Engineering, vol. 70, no. 1, 2024. doi: 10.24425/ace.2024.149836

[7] T. Mahmood dan M. Asif, "Prediction of energy efficiency for residential buildings using supervised machine learning algorithms," Energies, vol. 17, no. 19, art. no. 4965, 2024. doi: 10.3390/en17194965

[8] M. Irfan, F. Ramlie, Widianto, M. Lestandy, dan A. Faruq, "Prediction of residential building energy efficiency performance using deep neural network," IAENG International Journal of Computer Science, vol. 48, no. 3, 2021.

[9] M. Lim dan Y. Zhai, "Review on stochastic modeling methods for building stock energy prediction," Building Simulation, vol. 10, no. 5, hlm. 607–622, 2017. doi: 10.1007/s12273-017-0383-y

[10] M. A. Sun, Y. Wang, dan F. Xiao, "Stochastic simulation of microenvironment comfort and greenhouse gas emission of residential building using Monte-Carlo and hybrid intelligence paradigms," Environment, Development and Sustainability, 2025. doi: 10.1007/s10668-025-06635-0

[11] S. Mahmood, S. Sepasgozar, dan S. S. B. Dorji, "Comparison of deterministic, stochastic, and energy-data-driven occupancy models for building stock energy simulation," Buildings, vol. 14, no. 9, art. no. 2933, 2024. doi: 10.3390/buildings14092933

[12] M. D. McKay, R. J. Beckman, dan W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," Technometrics, vol. 21, no. 2, hlm. 239–245, 1979.

[13] J. M. Hammersley dan D. C. Handscomb, Monte Carlo Methods. London, UK: Methuen, 1964.

[14] N. Metropolis dan S. Ulam, "The Monte Carlo method," Journal of the American Statistical Association, vol. 44, no. 247, hlm. 335–341, 194.

RANCANG BANGUN GAME EDUKASI INTERAKTIF ADVENTURE BERBASIS HTML5 DAN NODE.JS MENGGUNAKAN METODE GDLC

Muhammad Nabil Fauzal Abrar[1)], Fatimah Martin[2),] Ika Arynita[3)] Suharsono[4)]
1,2,3) Program Studi Teknik Informatika Politeknik Negeri Pontianak
4) Program Studi Teknologi Informasi Politeknik Negeri Pontianak

Corresponding author e-mail: suharsono@polnep.ac.id

Abstrak

Kebutuhan akan media pembelajaran interaktif di era digital semakin mendesak untuk mengatasi kejenuhan terhadap metode pembelajaran konvensional yang sering dianggap monoton. Penelitian ini bertujuan untuk merancang dan membangun game web "Snake Fruit Adventure" sebagai media pembelajaran interaktif yang dilengkapi fitur *multiplayer*, predator, dan *power-up* guna meningkatkan keterlibatan serta motivasi belajar pengguna. Pengembangan game ini menerapkan metode Game Development Life Cycle (GDLC) yang mencakup enam tahapan sistematis, yaitu inisiasi, pra-produksi, produksi, pengujian, versi beta, dan rilis. Aplikasi dibangun menggunakan teknologi HTML5 untuk performa visual yang ringan serta Node.js untuk mendukung interaksi *multiplayer* secara *real-time*. Hasil penelitian menunjukkan bahwa game ini berhasil diimplementasikan dengan mekanik predator dan *power-up* yang berfungsi optimal sesuai rancangan sistem. Berdasarkan pengujian fungsionalitas menggunakan metode *black box*, sistem dinyatakan berjalan dengan baik tanpa adanya bug kritis. Selain itu, hasil *User Acceptance Test* (UAT) menunjukkan respon yang sangat positif dari responden, mengindikasikan bahwa Snake Fruit Adventure sangat layak digunakan sebagai sarana edukasi yang menyenangkan, menantang, dan efektif dalam meningkatkan minat belajar melalui pendekatan gamifikasi yang interaktif.

Kata kunci: Game Development Life Cycle (GDLC); HTML5 & Node.js; Media Pembelajaran Interaktif; Multiplayer; Snake Fruit Adventure

Abstract

The need for interactive learning media in the digital era has become increasingly urgent to overcome the boredom associated with conventional learning methods, which are often considered monotonous. This study aims to design and develop a web-based game entitled "Snake Fruit Adventure" as an interactive learning medium equipped with multiplayer, predator, and power-up features to enhance user engagement and learning motivation. The development of this game applies the Game Development Life Cycle (GDLC) method, which consists of six systematic stages: initiation, pre-production, production, testing, beta, and release. The application was developed using HTML5 technology to provide lightweight visual performance and Node.js to support real-time multiplayer interactions. The results of this study indicate that the game was successfully implemented, with predator and power-up mechanics functioning optimally according to the system design. Based on functionality testing using the Black Box testing method, the system was found to operate properly without any critical bugs. Furthermore, the results of the User Acceptance Test (UAT) showed highly positive responses from participants, indicating that Snake Fruit Adventure is highly suitable as an educational medium that is enjoyable, challenging, and effective in increasing learning interest through an interactive gamification approach.

Keywords: *Game Development Life Cycle (GDLC); HTML5 & Node.js; Interactive Learning Media; Multiplayer; Snake Fruit Adventure*

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi yang pesat telah membawa perubahan signifikan dalam berbagai sektor kehidupan, termasuk dunia pendidikan. Transformasi digital mendorong lahirnya berbagai inovasi media pembelajaran yang lebih interaktif, adaptif, dan menarik bagi peserta didik. Salah satu pendekatan yang semakin banyak dikaji adalah pemanfaatan game edukasi sebagai media pembelajaran alternatif yang mampu mengintegrasikan unsur hiburan (entertainment) dengan tujuan pedagogis secara sinergis [1].

Game edukasi terbukti efektif dalam meningkatkan motivasi intrinsik, keterlibatan (engagement), dan retensi pengetahuan peserta

didik dibandingkan metode pembelajaran konvensional yang bersifat satu arah [2]. Prinsip gamifikasi (gamification) yang diterapkan dalam desain pembelajaran mampu menciptakan lingkungan belajar yang kompetitif sekaligus kolaboratif, sehingga pengguna terdorong untuk terus mengeksplorasi materi secara sukarela [3]. Fenomena ini sejalan dengan teori pembelajaran berbasis pengalaman (experiential learning) yang menekankan pentingnya keterlibatan aktif peserta didik dalam proses memperoleh pengetahuan [4].

Kemajuan teknologi web modern, khususnya HTML5, telah membuka peluang baru dalam pengembangan game berbasis browser tanpa memerlukan plugin atau instalasi perangkat lunak tambahan. HTML5 menyediakan elemen Canvas dan Web Audio API yang memungkinkan rendering grafis dua dimensi serta pemrosesan audio secara langsung di dalam browser, sehingga menjadikannya platform yang ringan, lintas perangkat, dan mudah diakses oleh pengguna dari berbagai latar belakang [5]. Sementara itu, Node.js sebagai runtime JavaScript berbasis event-driven dan non-blocking I/O memberikan kemampuan untuk membangun server-side yang efisien dalam menangani komunikasi real-time, terutama untuk fitur multiplayer berbasis WebSocket [6].

Meskipun game berbasis web telah berkembang pesat, sebagian besar produk yang tersedia masih berorientasi murni pada hiburan tanpa memiliki dimensi edukatif yang terstruktur. Selain itu, fitur interaktif tingkat lanjut seperti kecerdasan buatan (Artificial Intelligence/AI) untuk mode VS AI, sistem multiplayer daring, obstacle dinamis, mekanik predator, serta power-up yang kontekstual masih jarang diimplementasikan secara terpadu dalam satu platform game edukasi berbasis web [7]. Kesenjangan ini mengindikasikan perlunya pengembangan game edukasi yang tidak hanya menghibur, tetapi juga mampu memberikan tantangan kognitif yang terukur dan pengalaman bermain yang kaya.

Berdasarkan kondisi tersebut, penelitian ini merancang dan membangun game edukasi berbasis web berjudul "Snake Fruit Adventure" yang dikembangkan menggunakan HTML5, CSS3, JavaScript, dan Node.js. Game ini mengadopsi konsep permainan klasik snake dengan penambahan fitur level progresif, mode VS AI, multiplayer real-time, obstacle, predator, dan power-up. Proses pengembangannya mengikuti metodologi Game Development Life Cycle (GDLC) yang terdiri dari enam tahapan sistematis: inisiasi, pra-produksi, produksi, pengujian, versi beta, dan rilis [8]. Penelitian ini diharapkan menghasilkan produk game edukasi yang menarik, interaktif, serta efektif sebagai sarana peningkatan minat dan motivasi belajar melalui pendekatan gamifikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun game edukasi interaktif Snake Fruit Adventure berbasis HTML5 dan Node.js dengan fitur multiplayer, predator, dan power-up?
2. Bagaimana penerapan metode Game Development Life Cycle (GDLC) dalam proses pengembangan game Snake Fruit Adventure secara sistematis dan terstruktur?
3. Bagaimana hasil pengujian fungsionalitas sistem game Snake Fruit Adventure menggunakan metode Black Box Testing?
4. Bagaimana tingkat penerimaan pengguna terhadap game Snake Fruit Adventure sebagai media pembelajaran interaktif berdasarkan hasil User Acceptance Test (UAT)?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang dan membangun game edukasi interaktif Snake Fruit Adventure berbasis HTML5 dan Node.js yang dilengkapi fitur multiplayer, predator, dan power-up.
2. Menerapkan metode Game Development Life Cycle (GDLC) sebagai kerangka pengembangan game secara sistematis dan terstruktur.
3. Menguji fungsionalitas game menggunakan metode Black Box Testing untuk memastikan sistem berjalan sesuai dengan spesifikasi yang dirancang.
4. Mengukur tingkat penerimaan pengguna terhadap game Snake Fruit Adventure sebagai media pembelajaran interaktif melalui User Acceptance Test (UAT).

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat bagi berbagai pihak sebagai berikut:

1. Bagi Pengguna Game Snake Fruit Adventure ini menyediakan media permainan interaktif yang mampu meningkatkan konsentrasi, kemampuan pengambilan keputusan, serta motivasi belajar melalui pengalaman bermain yang menarik dan menantang.
2. Bagi Pengembang game penelitian ini memberikan referensi teknis dan metodologis dalam pengembangan game edukasi berbasis web menggunakan teknologi HTML5 dan Node.js dengan pendekatan GDLC yang sistematis.
3. Bagi Penelitian Selanjutnya hasil penelitian ini dapat menjadi landasan referensi dan titik pengembangan bagi penelitian lanjutan yang berkaitan dengan game edukasi, gamifikasi, pengembangan aplikasi web interaktif, serta implementasi metode GDLC dalam konteks pendidikan berbasis teknologi.

II. TINJAUAN PUSTAKA

2.1 Game Edukasi

Game edukasi merupakan permainan digital yang dirancang untuk menggabungkan unsur hiburan dan pembelajaran sehingga dapat meningkatkan motivasi, keterlibatan, dan pemahaman pengguna terhadap materi yang dipelajari. Penggunaan game edukasi mampu menciptakan pengalaman belajar yang lebih menarik dibandingkan metode pembelajaran konvensional karena pengguna berpartisipasi secara aktif dalam proses pembelajaran [9], [10].

2.2 Media Pembelajaran Interaktif

Media pembelajaran interaktif memungkinkan terjadinya komunikasi dua arah antara pengguna dan sistem melalui pemanfaatan teknologi digital. Interaktivitas yang diberikan dapat meningkatkan perhatian, pemahaman, serta efektivitas proses belajar karena pengguna memperoleh umpan balik secara langsung terhadap aktivitas yang dilakukan [11], [12].

2.3 Game-Based Learning dan Gamifikasi

Game-Based Learning (GBL) merupakan pendekatan pembelajaran yang memanfaatkan permainan sebagai media untuk mencapai tujuan pembelajaran tertentu. Pendekatan ini didukung oleh konsep gamifikasi yang menerapkan elemen permainan seperti poin, level, tantangan, dan penghargaan pada aktivitas pembelajaran. Kombinasi keduanya terbukti mampu meningkatkan motivasi belajar, keterlibatan pengguna, serta pencapaian hasil pembelajaran [13], [14].

2.4 Teknologi HTML5 dan Browser Game

HTML5 merupakan teknologi web yang mendukung pengembangan game berbasis browser melalui elemen Canvas, CSS3, dan JavaScript. Teknologi ini memungkinkan game dijalankan secara langsung pada browser tanpa memerlukan instalasi tambahan sehingga lebih mudah diakses pada berbagai perangkat dan sistem operasi [15], [16].

2.5 Multiplayer Game Berbasis Web

Multiplayer game memungkinkan lebih dari satu pemain berinteraksi dalam lingkungan permainan yang sama secara real-time. Implementasi teknologi WebSocket pada Node.js mendukung komunikasi dua arah antara client dan server dengan latensi rendah sehingga cocok digunakan untuk pengembangan game multiplayer berbasis web [17].

2.6 Teknologi Pendidikan

Teknologi pendidikan merupakan penerapan teknologi untuk meningkatkan kualitas proses pembelajaran melalui perancangan, pengembangan, dan pemanfaatan berbagai sumber belajar. Penggunaan teknologi pendidikan dapat menciptakan pengalaman belajar yang lebih efektif, menarik, dan sesuai dengan kebutuhan peserta didik di era digital [18].

2.7 Game Development Life Cycle (GDLC)

Game Development Life Cycle (GDLC) merupakan metode pengembangan game yang terdiri dari enam tahapan yaitu Initiation, Pre-Production, Production, Testing, Beta, dan Release. Metode ini memberikan kerangka kerja yang sistematis sehingga proses pengembangan game dapat berjalan lebih terstruktur dan menghasilkan produk yang sesuai dengan kebutuhan pengguna [19], [20].

III. METODOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini menggunakan metode Research and Development (R&D) yang bertujuan untuk menghasilkan produk berupa game edukasi berbasis web berjudul Snake Fruit Adventure. Metode R&D digunakan karena tidak hanya berfokus pada pengkajian suatu fenomena, tetapi

juga menghasilkan produk yang dapat diuji dan dimanfaatkan oleh pengguna [21]. Game dikembangkan menggunakan teknologi HTML5, CSS, JavaScript, dan Node.js sehingga dapat dijalankan secara langsung melalui browser dengan dukungan fitur multiplayer secara real-time.

Metode pengembangan sistem yang digunakan adalah Game Development Life Cycle (GDLC) karena menyediakan tahapan pengembangan yang terstruktur dan sesuai untuk proses perancangan hingga implementasi game digital [19], [20].

3.2 Tahapan Penelitian

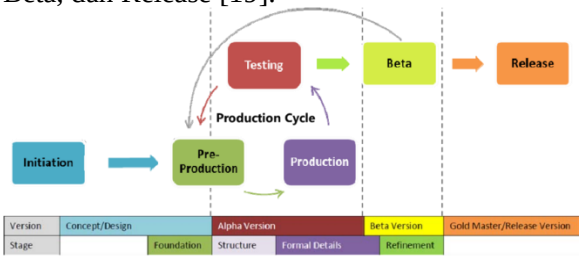
Tahapan penelitian yang dilakukan ditunjukkan pada Gambar 1.

Gambar 1. Tahapan Penelitian

Penelitian diawali dengan identifikasi masalah dan studi literatur mengenai game edukasi, HTML5, Node.js, multiplayer game, serta metode GDLC. Selanjutnya dilakukan analisis kebutuhan sistem yang menjadi dasar dalam pengembangan game. Proses pengembangan dilakukan menggunakan metode GDLC yang dilanjutkan dengan pengujian Black Box dan User Acceptance Testing (UAT). Hasil pengujian kemudian dianalisis untuk memperoleh kesimpulan mengenai kelayakan game yang dikembangkan.

3.3 Metode Pengembangan Sistem

Metode pengembangan yang digunakan adalah Game Development Life Cycle (GDLC) yang terdiri dari enam tahapan utama yaitu Initiation, Pre-Production, Production, Testing, Beta, dan Release [19].



Gambar 2. Metode GDLC

1. Initiation

Tahap initiation merupakan tahap awal yang berfokus pada identifikasi kebutuhan pengguna dan penentuan konsep permainan. Pada tahap ini ditentukan konsep game Snake Fruit Adventure sebagai game edukasi berbasis web yang mengintegrasikan fitur level, obstacle, predator, power-up, mode VS AI, dan multiplayer.

2. Pre-Production

Tahap pre-production dilakukan untuk menyusun rancangan sistem yang meliputi desain gameplay, aturan permainan, alur permainan, antarmuka pengguna (User Interface), struktur level, serta kebutuhan perangkat lunak yang digunakan dalam pengembangan game.

3. Production

Tahap production merupakan proses implementasi seluruh rancangan ke dalam bentuk aplikasi menggunakan HTML5, CSS, JavaScript, dan Node.js. Pada tahap ini dikembangkan fitur-fitur utama seperti mekanisme pergerakan karakter, sistem skor, sistem level, obstacle, predator, power-up, kecerdasan buatan sederhana pada mode VS AI, dan multiplayer berbasis WebSocket.

4. Testing

Tahap testing dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan. Pengujian menggunakan metode Black Box Testing yang berfokus pada pengujian fungsionalitas sistem tanpa memperhatikan struktur kode program [22].

5. Beta

Tahap beta dilakukan dengan melibatkan sejumlah pengguna untuk mencoba permainan secara langsung. Pengguna kemudian diminta memberikan penilaian terhadap game melalui instrumen User Acceptance Testing (UAT) guna mengetahui tingkat penerimaan dan kepuasan pengguna terhadap sistem yang dikembangkan [23].

6. Release

Tahap release merupakan tahap akhir setelah seluruh proses pengembangan dan pengujian selesai dilakukan. Pada tahap ini game dinyatakan siap digunakan dan dipublikasikan kepada pengguna.

3.4 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini meliputi:

1. Observasi, yaitu pengamatan terhadap kebutuhan pengguna terkait media pembelajaran interaktif berbasis game.
2. Studi Literatur, yaitu pengumpulan referensi yang berkaitan dengan game edukasi, HTML5, Node.js, multiplayer game, gamifikasi, dan metode GDLC.
3. Pengujian Sistem, yaitu pengumpulan data melalui pengujian Black Box Testing dan User Acceptance Testing (UAT).

3.5 Teknik Analisis Data

Analisis data dilakukan berdasarkan hasil pengujian sistem. Pengujian Black Box digunakan untuk mengevaluasi keberhasilan setiap fungsi yang terdapat pada game. Selanjutnya, hasil User Acceptance Testing (UAT) dianalisis menggunakan persentase tingkat kelayakan berdasarkan skor jawaban responden dengan rumus:

Persentase = (Skor Aktual / skor Maksimum) x 100%

Hasil persentase kemudian dikategorikan berdasarkan tingkat kelayakan sistem untuk menentukan tingkat penerimaan pengguna terhadap game Snake Fruit Adventure yang dikembangkan.

IV. HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi Sistem

Hasil penelitian ini berupa game edukasi berbasis web berjudul Snake Fruit Adventure yang dikembangkan menggunakan HTML5, CSS, JavaScript, dan Node.js. Game dapat dijalankan secara langsung melalui browser tanpa memerlukan instalasi tambahan sehingga lebih mudah diakses oleh pengguna pada berbagai perangkat. Sistem yang dikembangkan memiliki beberapa fitur utama, yaitu mode Single Player, VS AI, Multiplayer, sistem level, obstacle, predator, power-up shield, serta penyimpanan high score menggunakan Local Storage.

1. Tampilan Menu Utama

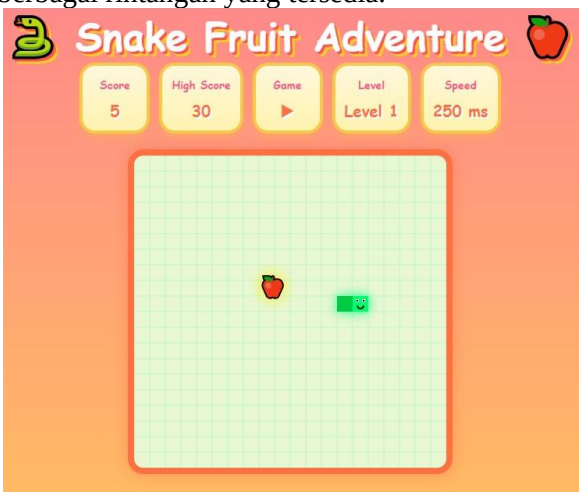
Menu utama merupakan halaman awal yang ditampilkan saat game dijalankan. Pada halaman ini pengguna dapat memilih mode permainan yang tersedia, yaitu Single Player, VS AI, dan Multiplayer. Selain itu tersedia informasi singkat mengenai aturan permainan serta tombol untuk memulai permainan.

Gambar 3. Tampilan Menu Utama

Tampilan menu dirancang sederhana dan responsif agar memudahkan pengguna dalam memilih mode permainan yang diinginkan. Penyediaan beberapa mode permainan bertujuan meningkatkan variasi pengalaman bermain sehingga pengguna tidak mudah merasa bosan.

2. Tampilan Gameplay

Gameplay merupakan bagian utama permainan yang menampilkan arena permainan, karakter ular, buah, serta informasi permainan secara real-time. Pemain mengendalikan karakter ular untuk mengumpulkan buah dan menghindari berbagai rintangan yang tersedia.



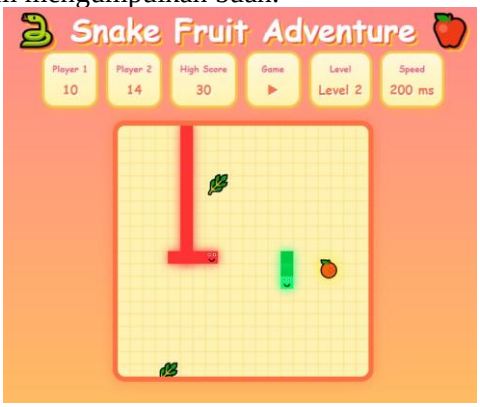
Gambar 4. Tampilan Gameplay

Setiap buah yang berhasil dikumpulkan akan menambah skor dan panjang tubuh ular. Sistem juga menampilkan informasi skor, level, kecepatan permainan, dan high score melalui Head-Up Display (HUD) sehingga pengguna dapat memantau perkembangan permainan secara langsung.

3. Implementasi Fitur Game

Game Snake Fruit Adventure memiliki lima level permainan dengan tingkat kesulitan yang meningkat secara bertahap. Peningkatan level memengaruhi kecepatan pergerakan permainan dan jumlah tantangan yang muncul sehingga memberikan pengalaman bermain yang lebih menantang.

Selain mode Single Player, sistem menyediakan mode VS AI yang memungkinkan pemain berkompetisi dengan karakter komputer dalam mengumpulkan buah.



Gambar 5. Tampilan Gameplay VS AI

Pada mode ini, kecerdasan buatan berperan sebagai lawan pemain sehingga menciptakan suasana kompetitif dan meningkatkan tingkat tantangan permainan.

Game juga menyediakan mode Multiplayer yang memungkinkan dua pemain bermain secara bersamaan menggunakan kontrol yang berbeda dalam satu arena permainan.

Gambar 6. Tampilan Gameplay Multiplayer

Implementasi mode multiplayer memberikan pengalaman bermain yang lebih interaktif karena memungkinkan terjadinya kompetisi secara langsung antar pemain dalam waktu yang sama.

Untuk meningkatkan variasi permainan, sistem dilengkapi dengan fitur obstacle, predator, dan power-up shield.

Gambar 7. Tampilan Obstacle, Predator, dan Power-Up

Obstacle mulai muncul pada Level 2 sebagai rintangan yang harus dihindari oleh pemain. Predator diaktifkan pada Level 4 dan Level 5 untuk meningkatkan tingkat kesulitan permainan karena dapat mengejar karakter pemain. Selain itu, power-up shield memberikan perlindungan sementara terhadap ancaman predator maupun tabrakan sehingga menambah strategi dalam permainan.

4.2 Pengujian

a. Pengujian Black Box

Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memastikan setiap fungsi pada game Snake Fruit Adventure berjalan sesuai dengan kebutuhan fungsional yang telah ditentukan. Pengujian dilakukan dengan memberikan input pada setiap fitur dan mengamati keluaran yang dihasilkan tanpa memperhatikan struktur kode program [22].

Tabel 1. Hasil Black Box

No	Fitur	Hasil
1	Pergerakan Ular	Berhasil
2	Sistem Skor	Berhasil
3	Sistem Level	Berhasil
4	Obstacle	Berhasil
5	Predator	Berhasil

No	Fitur	Hasil
6	Power-Up Shield	Berhasil
7	VS AI	Berhasil
8	Multiplayer	Berhasil
9	High Score	Berhasil
10	Game Over	Berhasil

Berdasarkan hasil pengujian pada Tabel 1, seluruh fitur utama game berhasil dijalankan sesuai dengan kebutuhan sistem. Fitur pergerakan ular, sistem skor, dan sistem level mampu beroperasi dengan baik selama permainan berlangsung. Selain itu, fitur obstacle, predator, dan power-up shield juga berfungsi sesuai mekanisme yang telah dirancang pada tahap pengembangan.

Pengujian terhadap mode VS AI menunjukkan bahwa karakter komputer dapat berinteraksi dengan lingkungan permainan dan bersaing dengan pemain dalam mengumpulkan buah. Sementara itu, fitur multiplayer berhasil memungkinkan dua pemain bermain secara bersamaan tanpa terjadi gangguan pada proses permainan. Sistem penyimpanan high score menggunakan Local Storage juga berhasil menyimpan dan menampilkan skor tertinggi secara otomatis. Dengan demikian, dapat disimpulkan bahwa seluruh fitur fungsional pada game telah berjalan dengan baik tanpa ditemukan kesalahan yang signifikan.

b. User Acceptance Testing (UAT)

Pengujian User Acceptance Testing (UAT) dilakukan untuk mengetahui tingkat penerimaan pengguna terhadap game Snake Fruit Adventure. Pengujian melibatkan 18 responden yang diminta memainkan game kemudian mengisi kuesioner yang terdiri dari 15 pernyataan menggunakan skala Likert 1–5. Aspek yang dinilai meliputi tampilan antarmuka, kemudahan penggunaan, kualitas fitur permainan, serta pengalaman bermain yang diperoleh pengguna.

Tabel 2. Hasil User Acceptance Testing

Keterangan	Nilai
Jumlah Responden	18
Jumlah Pertanyaan	15
Total Skor Diperoleh	1200
Skor Maksimum	1350
Persentase Kelayakan	88,89%
Kategori	Sangat Baik

Persentase kelayakan dihitung menggunakan rumus:

P = (Skor Diperoleh / Skor Maksimum) × 100%

P = (1200 / 1350) × 100%

P = 88,89%

Berdasarkan hasil perhitungan, diperoleh tingkat kelayakan sebesar 88,89% yang termasuk dalam kategori Sangat Baik. Hasil tersebut menunjukkan bahwa pengguna dapat menerima game dengan baik dari aspek tampilan, kemudahan penggunaan, dan fitur permainan yang tersedia.

4.3 Pembahasan

Berdasarkan hasil implementasi, seluruh fitur yang dirancang pada tahap pengembangan berhasil diterapkan ke dalam game Snake Fruit Adventure. Penggunaan HTML5 memungkinkan game dijalankan secara langsung melalui browser tanpa instalasi tambahan, sedangkan Node.js mendukung implementasi mode multiplayer secara real-time.

Penerapan sistem level, obstacle, predator, dan power-up berhasil menciptakan mekanisme permainan yang lebih dinamis dibandingkan permainan snake konvensional. Fitur-fitur tersebut memberikan variasi tantangan yang meningkat secara bertahap sehingga dapat meningkatkan keterlibatan (engagement) pengguna selama bermain.

Selain itu, keberadaan mode VS AI dan Multiplayer memberikan alternatif pengalaman bermain yang lebih interaktif. Mode VS AI memungkinkan pengguna berkompetisi dengan sistem komputer, sedangkan mode multiplayer mendukung interaksi langsung antar pemain. Kombinasi fitur tersebut mendukung konsep Game-Based Learning dan gamifikasi yang bertujuan meningkatkan motivasi serta partisipasi pengguna dalam proses pembelajaran melalui aktivitas bermain.

Secara keseluruhan, hasil implementasi menunjukkan bahwa game Snake Fruit Adventure berhasil dikembangkan sesuai dengan rancangan yang telah ditetapkan pada tahap GDLC dan mampu menyediakan media permainan edukatif yang interaktif, menarik, dan mudah diakses melalui browser web

V. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa game edukasi berbasis web Snake Fruit Adventure berhasil dirancang dan dikembangkan menggunakan metode Game Development Life Cycle (GDLC) melalui tahapan Initiation, Pre-Production, Production, Testing, Beta, dan Release. Implementasi teknologi HTML5, CSS, JavaScript, dan Node.js memungkinkan game dijalankan secara langsung melalui browser dengan dukungan berbagai fitur interaktif, meliputi mode Single Player, VS AI, Multiplayer, sistem level, obstacle, predator, power-up shield, dan penyimpanan high score. Hasil pengujian Black Box Testing menunjukkan bahwa seluruh fitur yang dikembangkan berfungsi sesuai dengan kebutuhan fungsional sistem tanpa ditemukan kesalahan yang signifikan. Selain itu, hasil User Acceptance Testing (UAT) yang melibatkan 18 responden memperoleh tingkat kelayakan sebesar 88,89% dengan kategori Sangat Baik, yang menunjukkan bahwa game memiliki tingkat penerimaan pengguna yang tinggi dari aspek tampilan, kemudahan penggunaan, kualitas fitur, dan pengalaman bermain. Temuan ini menunjukkan bahwa penerapan konsep Game-Based Learning dan gamifikasi yang diintegrasikan melalui fitur multiplayer, predator, dan power-up mampu meningkatkan keterlibatan pengguna dalam aktivitas permainan. Dengan demikian, Snake Fruit Adventure layak digunakan sebagai media permainan edukatif berbasis web yang interaktif, menarik, dan mudah diakses, serta berpotensi menjadi alternatif media pembelajaran digital yang mendukung peningkatan motivasi dan partisipasi pengguna dalam proses belajar. Adapun untuk pengembangan selanjutnya, penelitian ini merekomendasikan penambahan fitur leaderboard daring, integrasi basis data, peningkatan kecerdasan buatan pada mode VS AI, serta pengayaan konten pembelajaran yang lebih spesifik agar aspek edukatif game dapat diukur secara lebih komprehensif melalui evaluasi hasil belajar pengguna..

DAFTAR PUSTAKA

[1] Mayer, R. E. (2019). Computer Games in Education. Annual Review of Psychology, 70, 531–549.

[2] Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of Game-Based Learning. Educational Psychologist, 50(4), 258–283.

[3] Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does Gamification Work? A Literature Review. Proceedings of HICSS, 3025–3034.

[4] Kolb, D. A. (1984). Experiential Learning: Experience as the Source of Learning and Development. Prentice Hall.

[5] Pilgrim, M. (2010). HTML5: Up and Running. O'Reilly Media.

[6] Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Programs. IEEE Internet Computing, 14(6), 80–83.

[7] Dichev, C., & Dicheva, D. (2017). Gamifying Education: What Is Known, What Is Believed and What Remains Uncertain. International Journal of Educational Technology in Higher Education, 14(1), 1–17.

[8] Ramadan, R., & Widyani, Y. (2013). Game Development Life Cycle Guidelines. Proceedings of ICACSIS, 95–100.

[9] M. Prensky, Digital Game-Based Learning. New York, NY, USA: McGraw-Hill, 2001.

[10] J. P. Gee, What Video Games Have to Teach Us About Learning and Literacy. New York, NY, USA: Palgrave Macmillan, 2007.

[11] R. E. Mayer, Multimedia Learning, 2nd ed. New York, NY, USA: Cambridge University Press, 2009.

[12] H. D. Surjono, Multimedia Pembelajaran Interaktif. Yogyakarta: UNY Press, 2017.

[13] K. Kiili, “Digital game-based learning: Towards an experiential gaming model,” Internet Higher Education, vol. 8, no. 1, pp. 13–24, 2005.

[14] K. M. Kapp, The Gamification of Learning and Instruction. San Francisco, CA, USA: Pfeiffer, 2012.
[15] S. Lawson, Introducing HTML5. Berkeley, CA, USA: Apress, 2011.
[16] R. Nixon, Learning PHP, MySQL & JavaScript with jQuery, CSS & HTML5, 5th ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
[17] M. Rouse, "WebSocket Protocol for Real-Time Web Applications," TechTarget, 2022.
[18] A. Januszewski and M. Molenda, Educational Technology: A Definition with Commentary. New York, NY, USA: Routledge, 2008.
[19] R. Ramadan and Y. Widyani, "Game Development Life Cycle Guidelines," in Proc. ICACSIS, 2013, pp. 95–100.
[20] H. Hendrick, Game Development for Beginners. New York, NY, USA: Apress, 2019.
[21] W. R. Borg and M. D. Gall, Educational Research: An Introduction, 8th ed. Boston, MA, USA: Pearson Education, 2007.
[22] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 8th ed. New York, NY, USA: McGraw-Hill, 2015.
[23] J. Rubin and D. Chisnell, Handbook of Usability Testing, 2nd ed. Indianapolis, IN, USA: Wiley Publishing, 2008.

ANALISIS ARSITEKTUR PADA INTEGRASI EVENT STREAMING APACHE KAFKA DAN KETERHUBUNGANNYA DENGAN EKSEKUSI SERVERLESS

(*Architectural Analysis of Apache Kafka Event Streaming Integration and Its Interconnection with Serverless Execution*)

Jauda Avaro Zufaru[1], I Made Priyandika Adiyatma Putra Sari[2], Muhammad Asrofi Sazani[3] , Revano Januar Adiguna Prawira[4]

[1, 2, 3, 4]Program Studi Teknik Informatika, Universitas Mataram

Jl. Majapahit 62, Mataram, Nusa Tenggara Barat, Indonesia

Email: f1d02410059@student.unram.ac.id, f1d02410008@student.unram.ac.id, f1d02410123@student.unram.ac.id, f1d02410146@student.unram.ac.id

Abstract

The growing demand for real-time data processing in distributed systems has driven the adoption of event-driven architectures that integrate event streaming platforms with serverless computing models. This study aims to analyze and map the architectural characteristics of the integration between Apache Kafka as an event streaming platform and serverless execution as an event-based automated processing unit. The research employs a Systematic Literature Review (SLR) methodology applied to sixteen relevant scientific publications sourced from Google Scholar, IEEE Xplore, and ScienceDirect. The analysis is conducted based on five primary parameters: scalability, latency, fault tolerance, reliability, and resource efficiency. Results indicate that the combination of these two technologies effectively supports flexible and efficient real-time data processing within modern cloud environments. Apache Kafka provides high-throughput, fault-tolerant data distribution, while serverless computing offers elastic auto-scaling that optimizes resource utilization. Key challenges identified include cold start latency and partition-to-instance scaling mismatches between Kafka and serverless functions, which can be mitigated through prewarming mechanisms and predictive auto-scaling strategies. This study is intended to serve as a theoretical reference for developers seeking to adopt this distributed architecture without the complexity of physical implementation testing.

Keywords: *Apache Kafka, Serverless Computing, Event Streaming, Event-Driven Architecture, Distributed Systems, Cold Start Latency, Auto-Scaling*

1. PENDAHULUAN

Perkembangan teknologi *cloud computing* saat ini telah mengubah cara organisasi dalam mengelola dan memproses data berskala besar [1]. Kebutuhan akan informasi yang cepat mendorong adopsi arsitektur berbasis peristiwa atau *event-driven architecture* pada data diproses secara langsung pada saat kejadian berlangsung atau secara *real-time* [2]. Dalam ekosistem ini, platform *event streaming* berperan sangat penting untuk memastikan aliran data dari berbagai sumber dapat berjalan dengan lancar dan teratur. Di sisi lain, munculnya model eksekusi berbasis *serverless computing* memberikan pendekatan baru dalam pengolahan data yang lebih fleksibel, di mana sumber daya komputasi disediakan secara otomatis sesuai kebutuhan [1]. Kombinasi antara platform pengelolaan data dan model eksekusi otomatis ini menjadi dasar penting dalam pengembangan arsitektur sistem modern yang adaptif.

Dalam praktik di industri teknologi, integrasi antara kedua platform ini banyak diterapkan pada studi kasus aplikasi modern yang membutuhkan pengolahan data cepat dan fleksibel [3]. Sebagai contoh, pada sistem pemesanan *online* atau layanan finansial, aliran data berupa transaksi dari jutaan pengguna harus ditangkap dan dikelola secara aman. Dalam skenario tersebut, Apache Kafka diimplementasikan sebagai penampung utama yang mengatur antrean aliran data masuk dari pengguna agar tidak ada informasi yang hilang. Setelah data masuk ke dalam antrean, eksekusi *serverless computing* kemudian dihubungkan untuk mengambil dan memproses data tersebut secara otomatis, seperti

memvalidasi pembayaran atau memperbarui status pesanan [3]. Studi kasus ini menunjukkan bagaimana kedua subjek tersebut saling terhubung dalam menyelesaikan kebutuhan pemrosesan data di dunia nyata.

Meskipun integrasi ini menawarkan banyak keuntungan, penerapannya di lapangan masih menghadapi beberapa tantangan arsitektural yang signifikan. Salah satu kendala utama yang sering muncul adalah masalah *cold start* dalam fungsi *serverless* membutuhkan waktu komputasi tambahan untuk aktif pertama kali saat menerima data dari antrean Apache Kafka [4]. Selain itu, koordinasi skala antara jumlah partisi pada Apache Kafka dan jumlah *instance serverless* yang aktif secara bersamaan sering kali menimbulkan hambatan performa atau latensi jika tidak dikonfigurasi dengan tepat [3]. Tantangan koordinasi dan manajemen koneksi ini menjadi fokus penting yang perlu dianalisis secara mendalam agar interaksi antara penampung data dan pemrosesan otomatis dapat berjalan lebih selaras.

Untuk mengatasi kendala tersebut, berbagai penelitian terdahulu telah mencoba memetakan dan menganalisis pola arsitektur integrasi yang ideal. Beberapa solusi yang ditawarkan dalam literatur mencakup penerapan strategi pemicuan atau *triggering mechanism* yang lebih cerdas serta penyesuaian alokasi sumber daya secara dinamis [5]. Melalui analisis terhadap cetak biru arsitektur yang sudah ada, dapat dipelajari bagaimana penggunaan metode pemanasan awal atau *prewarming* pada fungsi *serverless* mampu meminimalkan keterlambatan pemrosesan data dari Apache Kafka [6]. Evaluasi terhadap pola-pola hubungan arsitektur ini memberikan landasan teori yang kuat bagi pengembangan sistem terdistribusi yang lebih stabil tanpa memerlukan uji coba fisik yang rumit.

Melalui penelitian ini, diharapkan pembaca akan mendapatkan pemahaman yang mendalam mengenai karakteristik arsitektur pada integrasi *event streaming* Apache Kafka dan keterhubungannya dengan eksekusi *serverless*. Dengan memanfaatkan pengetahuan mengenai berbagai model interaksi, tantangan teknis, dan solusi arsitektural yang dibahas dari berbagai literatur valid, pengembang maupun perusahaan dapat membuat keputusan yang tepat dalam mengadopsi teknologi ini. Pada akhirnya, kajian literatur sistematis ini diharapkan dapat menjadi referensi yang mudah dipahami dan diaplikasikan dalam mempercepat pengembangan infrastruktur aplikasi modern yang responsif dan terarah.

2. TINJAUAN PUSTAKA

Perkembangan sistem terdistribusi modern membutuhkan pengolahan data secara *real-time* yang terus meningkat seiring bertambahnya jumlah perangkat dan layanan digital yang saling terhubung. Sistem tidak hanya dituntut mampu memproses data dalam jumlah besar, tetapi juga harus memiliki *scalability*, *reliability*, dan *fault tolerance* yang baik. Salah satu teknologi yang banyak digunakan untuk memenuhi kebutuhan tersebut adalah Apache Kafka sebagai platform *event streaming* berbasis *publish*/*subscribe*. Penelitian oleh Obannagari dkk. menunjukkan bahwa Kafka mampu meningkatkan efisiensi pengolahan data pada lingkungan *Internet of Things* (IoT) melalui mekanisme distribusi data secara *real-time* [7]. Namun, penelitian tersebut masih lebih berfokus pada pengelolaan aliran data dan belum membahas integrasi dengan mekanisme pemrosesan yang bersifat dinamis seperti *serverless computing*.

Selain Kafka, berbagai teknologi *messaging* dan *event streaming* juga mulai berkembang untuk mendukung kebutuhan sistem modern. Penelitian mengenai perbandingan performa Apache Kafka, RabbitMQ, Apache Pulsar, dan *serverless event bus* menunjukkan bahwa setiap platform memiliki karakteristik yang berbeda [8]. Apache Kafka unggul dalam menangani *throughput* data yang besar dan stabil, sedangkan arsitektur *serverless* lebih unggul dalam efisiensi penggunaan *resource* karena proses komputasi hanya berjalan saat dibutuhkan. Perbedaan karakteristik tersebut menunjukkan bahwa integrasi antara *event streaming* dan *serverless computing* menjadi solusi yang menarik untuk mendukung sistem terdistribusi yang lebih fleksibel dan efisien.

Konsep *Event-Driven Architecture* (EDA) menjadi salah satu pendekatan utama dalam pengembangan sistem terdistribusi modern. Pada arsitektur ini, *event* berperan sebagai pemicu komunikasi antar layanan secara *asynchronous*. Penelitian mengenai *triggerflow* menunjukkan bahwa mekanisme *trigger-based orchestration* mampu mengatur *workflow serverless* secara otomatis dan adaptif pada lingkungan *cloud* [9]. Selain itu, penelitian lain mengenai *design pattern* pada Apache Kafka juga menjelaskan bahwa pola seperti *event sourcing*, Com*mand Query Responsibility Segregation* (CQRS), dan *exactly-once processing* berperan penting dalam menjaga konsistensi dan keandalan distribusi data [10]. Namun, sebagian besar penelitian tersebut masih berfokus pada desain komunikasi data dan belum membahas secara mendalam mengenai sinkronisasi performa antara Kafka dan unit eksekusi *serverless*.

Dalam implementasinya, Apache Kafka banyak digunakan pada sistem *big data analytics* karena memiliki kemampuan *throughput* tinggi dan *latency* rendah dalam memproses jutaan *event* secara *real-time* [11]. Dukungan teknologi seperti *kafka streams* dan *kafka connect* juga membuat Kafka semakin fleksibel untuk digunakan pada berbagai kebutuhan data *pipeline* modern. Penelitian lain yang menggabungkan Kafka dengan Apache Flink menunjukkan bahwa kombinasi tersebut mampu menangani pemrosesan *hybrid batch-stream* dengan baik pada lingkungan *data lake* [12]. Akan tetapi, pendekatan tersebut masih menggunakan resource komputasi yang cenderung statis sehingga kurang efisien ketika terjadi perubahan *workload* yang fluktuatif.

Untuk mengatasi masalah efisiensi *resource*, *serverless computing* hadir sebagai model komputasi yang menawarkan mekanisme *auto-scaling* dan pembayaran berbasis penggunaan *resource* aktual [13]. Pendekatan ini memungkinkan sistem melakukan *scaling* secara otomatis tanpa perlu mengelola server secara langsung. Meskipun demikian, *serverless computing* masih memiliki kendala berupa *cold start latency*, yaitu keterlambatan eksekusi fungsi akibat proses inisialisasi kontainer saat menerima *request* pertama [13]. Permasalahan ini dapat memengaruhi *Quality of Service* (QoS) terutama pada aplikasi *real-time* yang membutuhkan respons cepat.

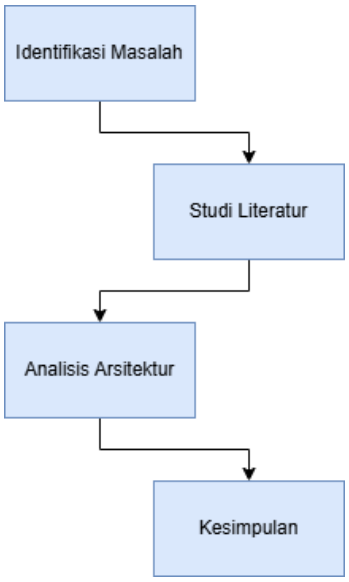
Beberapa penelitian telah mencoba mengatasi masalah *cold start* pada *serverless* platform. Salah satu pendekatan yang cukup efektif adalah penggunaan *warm container* atau *prewarming mechanism* pada platform *Function as a Service* (FaaS). Penelitian oleh Lin dan Glikson menunjukkan bahwa metode *pool-based warm container* mampu mengurangi waktu respons sistem secara signifikan karena fungsi *serverless* telah dipersiapkan sebelum menerima *request* [14]. Namun, pengujian pada penelitian tersebut masih dilakukan pada lingkungan internal platform dan belum sepenuhnya mempertimbangkan pola lonjakan *event* dari Apache Kafka yang bersifat dinamis.

Integrasi antara Apache Kafka dan *serverless computing* mulai banyak diterapkan pada lingkungan *cloud* modern. Penelitian mengenai integrasi Kafka dengan layanan *cloud* seperti AWS Lambda dan Amazon MSK menunjukkan bahwa kombinasi keduanya mampu meningkatkan fleksibilitas pengelolaan data *pipeline* secara *real-time* [15]. Kafka berperan dalam menjaga kestabilan distribusi data, sedangkan *serverless* berfungsi untuk menyediakan proses komputasi yang elastis dan efisien. Selain itu, penelitian mengenai *multi-cloud* data *synchronization* menggunakan Kafka *stream processing* juga menunjukkan bahwa Kafka mampu mempertahankan *latency* rendah dan *reliability* tinggi pada proses sinkronisasi data antar *cloud* [16]. Akan tetapi, penelitian tersebut masih belum membahas optimasi mekanisme *auto-scaling* dan sinkronisasi *workload* secara detail pada integrasi *serverless*.

Berdasarkan berbagai penelitian sebelumnya, masih terdapat beberapa keterbatasan pada integrasi Apache Kafka dan *serverless computing*, khususnya terkait *scalability*, *latency*, dan efisiensi *resource* saat menghadapi *workload* yang berubah secara dinamis. Oleh karena itu, penelitian ini difokuskan pada analisis arsitektur integrasi *event streaming* Apache Kafka dan eksekusi *serverless* dengan meninjau parameter *scalability*, *latency*, *fault tolerance*, *reliability*, dan *resource efficiency*. Analisis dilakukan untuk memahami bagaimana strategi *auto-scaling* dan *prewarming* dapat membantu meningkatkan stabilitas dan performa sistem terdistribusi modern.

3. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan yang disusun secara sistematis dalam bentuk *flowchart* untuk memudahkan pemahaman alur penelitian. *Flowchart* tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Alur Tahapan Penelitian

Berdasarkan *flowchart* tersebut, penelitian dimulai dari tahap identifikasi masalah hingga penarikan kesimpulan yang dilakukan secara bertahap dan terstruktur.

3.1. Identifikasi Masalah

Tahap identifikasi masalah dilakukan dengan meninjau kebutuhan sistem dalam pemrosesan data *real-time* pada lingkungan *distributed computing* serta mengkaji permasalahan yang sering muncul berdasarkan hasil studi literatur.

Dari proses tersebut, diperoleh beberapa permasalahan utama, antara lain *latency* yang disebabkan oleh *cold start* pada *serverless computing*, ketidaksesuaian skala antara Apache Kafka dan *serverless function*, serta efisiensi penggunaan *resource* pada kondisi *workload* yang bersifat dinamis. Permasalahan ini kemudian menjadi fokus utama dalam penelitian.

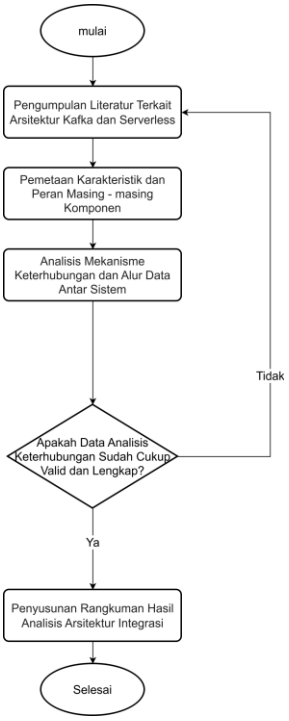
3.2. Studi Literatur

Tahap studi literatur dilakukan untuk memperoleh landasan teori yang mendukung penelitian. Proses ini dilakukan dengan mengumpulkan dan mengkaji berbagai sumber ilmiah yang relevan dengan topik penelitian, seperti jurnal, prosiding konferensi, dan artikel ilmiah yang membahas Apache Kafka, *event streaming*, *serverless computing*, serta integrasi keduanya dalam sistem terdistribusi.

Pengumpulan literatur (sumber ilmiah yang relevan) dilakukan melalui *database* ilmiah seperti Google Scholar, IEEE Xplore, ACM (Association for Computing Machinery) Digital Library, SpringerLink, DOAJ (Directory of Open Access Journals), dan ScienceDirect atau pun pusat data ilmiah sejenis dengan menggunakan kata kunci seperti "Apache Kafka", "*event-driven architecture*", "*serverless computing*", dan "*event streaming integration*". Literatur yang dipilih merupakan publikasi dalam rentang 5 hingga 10 tahun terakhir untuk memastikan kesesuaian dengan perkembangan teknologi terkini. Selanjutnya, literatur yang telah dikumpulkan dianalisis dengan cara mengidentifikasi konsep utama, arsitektur yang digunakan, serta kelebihan dan keterbatasan dari masing-masing pendekatan. Hasil dari proses ini digunakan sebagai dasar dalam melakukan analisis arsitektur pada penelitian.

3.3. Analisis Arsitektur

Tahap utama dalam penelitian ini adalah analisis arsitektur integrasi antara Apache Kafka dan *serverless computing*. Pada tahap ini, proses analisis akan dilakukan berdasarkan alur analisis yang ditunjukkan pada *flowchart* (diagram alir) analisis arsitektur integrasi Apache Kafka dan *serverless computing* sebagaimana yang tertera pada Gambar 2.



Gambar 2. Arsitektur Integrasi Apache Kafka dan *Serverless*

3.3.1. Flowchart Analisis Arsitektur

Flowchart analisis arsitektur menggambarkan alur proses mulai dari data yang dikirim oleh *producer*, kemudian diterima oleh Apache Kafka sebagai *event streaming* platform, hingga diproses oleh *serverless function*. Selanjutnya, hasil pemrosesan dikirim ke *database* atau *consumer service* untuk digunakan dalam proses lanjutan. Alur ini menunjukkan bagaimana integrasi antara Kafka dan *serverless* bekerja dalam sistem *event-driven* secara keseluruhan.

3.3.2. Deskripsi Sistem

Sistem yang dianalisis menggunakan pendekatan *event-driven architecture* dengan Apache Kafka sebagai *message broker* utama yang bertugas menerima dan mendistribusikan *event* dari *producer* menuju layanan *serverless*.

Producer berfungsi sebagai pengirim data atau *event* secara *real-time*, kemudian data tersebut diproses oleh Kafka melalui mekanisme *asynchronous communication*. Selanjutnya *event* diteruskan menuju *serverless function* untuk dilakukan pemrosesan data secara otomatis sesuai kebutuhan sistem.

Hasil pemrosesan kemudian dikirim menuju *database* atau *consumer service* untuk digunakan pada proses lanjutan. Arsitektur ini dipilih karena mampu mendukung *scalability*, *fault tolerance*, dan efisiensi *resource* pada sistem terdistribusi modern.

3.3.3. Parameter Analisis

Analisis ini bertujuan untuk memahami mekanisme interaksi antara komponen sistem serta mengevaluasi kinerja arsitektur berdasarkan parameter yang telah ditentukan.

Parameter yang digunakan dalam penelitian ini meliputi *scalability*, *latency*, *fault tolerance*, *reliability*, dan *resource efficiency*. Parameter tersebut digunakan untuk mengetahui kemampuan arsitektur sistem dalam menangani peningkatan *workload*, menjaga kestabilan sistem, serta mengoptimalkan penggunaan *resource* pada lingkungan *cloud* dan *distributed system*.

TABEL I. PARAMETER ANALISIS ARSITEKTUR

Parameter	Deskripsi
Scalability	Kemampuan sistem menangani peningkatan jumlah *event* dan *workload*
Latency	Waktu yang dibutuhkan dalam pemrosesan *event* secara *real-time*
Fault Tolerance	Kemampuan sistem tetap berjalan saat terjadi gangguan atau kegagalan *service*
Reliability	Konsistensi sistem dalam memproses dan mendistribusikan data
Resource Efficiency	Efisiensi penggunaan *resource* pada lingkungan *cloud*

4. HASIL DAN PEMBAHASAN

Berdasarkan hasil analisis dari berbagai penelitian sebelumnya, integrasi Apache Kafka dengan *serverless computing* menunjukkan peningkatan performa pada sistem *distributed computing* modern, terutama dalam aspek *scalability*, *latency*, *reliability*, *fault tolerance*, dan *resource efficiency*. Analisis dilakukan dengan membandingkan karakteristik arsitektur *event-driven* yang menggunakan Apache Kafka sebagai *event streaming* platform dan *serverless function* sebagai mekanisme pemrosesan otomatis berbasis *event*.

4.1. Hasil Studi Literatur

No.	Ulasan Literatur	
	Informasi	Deskripsi
1.	Judul	*Serverless Computing*: Kajian Literatur untuk Memahami Arsitektur dan Implikasinya di Era *Cloud Computing*
	Penulis	Andre Leto
	Temuan	Penelitian menggunakan metode *Systematic Literature Review* (SLR) untuk menganalisis perkembangan arsitektur *serverless computing* berbasis *event-driven* dan *Function-as-a-Service* (FaaS). Hasil penelitian menunjukkan bahwa integrasi *event-driven architecture* mampu meningkatkan *scalability*, efisiensi *resource*, dan fleksibilitas sistem *cloud* modern. Penelitian juga menjelaskan bahwa mekanisme *asynchronous execution* pada *serverless* sangat cocok digunakan bersama platform *event streaming* seperti *Apache Kafka* untuk pemrosesan data *real-time*.
2.	Judul	*Cloud Programming Simplified: A Berkeley View on Serverless Computing*
	Penulis	E. Jonas, Q. Pu, S. Venkataraman, I. Stoica, dan B. Recht
	Temuan	Apache Kafka dapat diintegrasikan dengan *serverless computing* untuk mendukung pemrosesan *event* secara otomatis melalui mekanisme *event-driven architecture* dan *auto-scaling* berbasis *cloud*.
3.	Judul	*A Review of Serverless Use Cases and their Characteristics*
	Penulis	S. Eismann, J. Scheuner, E. Van Eyk, M. Schwinger, J. Grohmann, N. Herbst, C. L. Abad, dan A. Iosup
	Temuan	Apache Kafka digunakan sebagai *event-streaming* platform yang mendukung *workload* dinamis pada *serverless computing* sehingga penggunaan *resource* menjadi lebih efisien dan fleksibel.
4.	Judul	*Serverless Computing for Internet of Things: A Systematic Literature Review*
	Penulis	G. Cassel, V. Rodrigues, R. Righi, M. Bez, A. Cruz, dan C. André da Costa
	Temuan	Apache Kafka digunakan pada sistem IoT berbasis *event-driven* untuk mendukung distribusi data *real-time* dan meningkatkan *scalability* pada integrasi *serverless computing*.
5.	Judul	*A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms*

	Penulis	T. Lynn, P. Rosati, A. Lejeune, dan V. Emeakaroha
	Temuan	Apache Kafka mendukung arsitektur *distributed system* pada lingkungan *enterprise* dengan menyediakan mekanisme *event streaming* yang stabil untuk *serverless execution*.
6.	Judul	*Serverless Computing Optimization Strategies Using ML-Based Auto-Scaling and Event-Stream Intelligence for Low-Latency Enterprise Workloads*
	Penulis	P. R. Nangi, C. K. R. N. Obannagari, dan S. Settipi
	Temuan	Apache Kafka digunakan sebagai *event-stream intelligence* platform untuk mendukung *predictive auto-scaling* dan mengurangi *latency* pada sistem *serverless* berbasis *real-time processing*
7.	Judul	*ESTemd: A Distributed Processing Framework for Environmental Monitoring based on Apache Kafka Streaming Engine*
	Penulis	Adeyinka Akanbi
	Temuan	Apache Kafka diimplementasikan sebagai *streaming engine* untuk memproses dan mendistribusikan data *monitoring* lingkungan secara *real-time* dengan *throughput* tinggi dan *fault tolerance* yang baik.
8.	Judul	*Triggerflow: Trigger-based Orchestration of Serverless Workflows*
	Penulis	P. G. López, A. Arjona, J. Sampé, A. Slominski, and L. Villard
	Temuan	Apache Kafka mendukung mekanisme *trigger-based orchestration* pada *serverless workflow* untuk sinkronisasi layanan secara *asynchronous* pada *cloud environment*.
9.	Judul	*Analysis of Design Patterns and Benchmark Practices in Apache Kafka Event-Streaming Systems*
	Penulis	M. Mohammad
	Temuan	Apache Kafka menggunakan *design pattern seperti event sourcing* dan *exactly-once processing* untuk meningkatkan *reliability*, *consistency*, dan *fault tolerance* pada sistem *event-streaming*.
10.	Judul	*Real-Time Data Processing Using Apache Kafka: Architecture, Implementation, and Performance Evaluation*
	Penulis	Gomathy, C. K., Nikhil, M. V. R., & Sriharsha, S.
	Temuan	Apache Kafka mampu memproses *event* secara *real-time* dengan *latency* rendah dan *throughput* tinggi sehingga efektif digunakan pada *distributed computing* dan *big data processing.*
12.	Judul	*A Tale of Many Streams: Characterizing a Hybrid Batch-Stream Production Workload in Digazu, a Data Lake Supported by Apache Kafka and Flink*
	Penulis	Schmitz, D., Berrewaerts, L., Rosinosky, G., Skhiri, S., & Rivière, E.
	Temuan	Apache Kafka digunakan bersama Apache Flink untuk mendukung *hybrid batch-stream processing* pada *data lake* sehingga distribusi *workload* menjadi lebih stabil dan efisien.
13.	Judul	*Cold Start Latency in Serverless Computing: A Systematic Review, Taxonomy, and Future Directions*
	Penulis	M. Golec, G. K. Walia, M. Kumar, F. Cuadrado, S. S. Gill, dan S. Uhlig
	Temuan	Apache Kafka pada integrasi *serverless* masih menghadapi kendala *cold start latency* yang dapat memengaruhi kecepatan pemrosesan *event* secara *real-time*.
14.	Judul	*Mitigating Cold Starts in Serverless Platforms: A Pool-Based Approach*
	Penulis	P.-M. Lin dan A. Glikson
	Temuan	Apache Kafka dapat dioptimalkan pada integrasi *serverless* menggunakan *warm container* dan *prewarming mechanism* untuk

		mengurangi *latency* saat pemrosesan *event*.
15.	Judul	*Real-Time Data Ingestion with Kafka and AWS Tools*
	Penulis	S. K. Gupta
	Temuan	Apache Kafka digunakan untuk *real-time data ingestion* pada *cloud environment* AWS sehingga mampu meningkatkan *scalability* dan efisiensi distribusi *data streaming*.
16.	Judul	*Multi-Cloud Data Synchronization Using Kafka Stream Processing*
	Penulis	V. K. Tambi
	Temuan	Apache Kafka digunakan untuk sinkronisasi data antar *cloud* melalui *stream processing* dengan *latency* rendah dan *reliability* tinggi pada sistem terdistribusi.

4.2. Analisis *Scalability* Sistem

Hasil analisis menunjukkan bahwa Apache Kafka memiliki kemampuan *scalability* yang sangat baik dalam menangani peningkatan jumlah *event* secara *real-time*. Kafka mampu mendistribusikan data melalui mekanisme *partition* sehingga proses pemrosesan dapat dilakukan secara paralel oleh beberapa *consumer* [7]. Hal ini membuat sistem tetap stabil meskipun jumlah *event* terus meningkat.

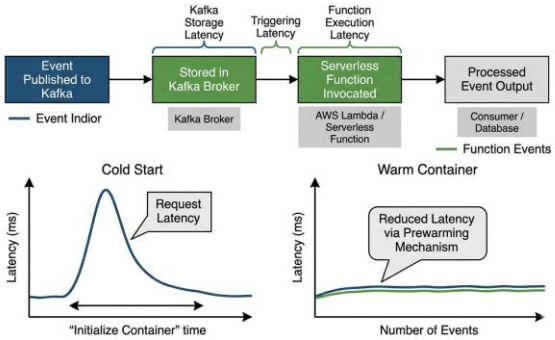
Pada integrasi dengan *serverless computing*, proses *scaling* dilakukan secara otomatis sesuai jumlah *request* yang diterima sistem. Ketika *event* pada Kafka meningkat, *serverless function* akan menambah *instance* pemrosesan secara dinamis tanpa perlu konfigurasi manual [2]. Pendekatan ini membuat sistem lebih fleksibel dibanding arsitektur tradisional yang masih menggunakan server statis.

Namun, berdasarkan penelitian Nangi (2024), *reactive auto-scaling* memiliki kelemahan ketika terjadi *bursty workload* atau lonjakan *event* secara mendadak. Sistem membutuhkan waktu untuk menambah *resource* sehingga *latency* meningkat sementara waktu. Oleh karena itu, *predictive auto-scaling* menjadi solusi yang lebih efektif karena mampu memprediksi kenaikan *workload* sebelum lonjakan *event* benar-benar terjadi.

Selain itu, penelitian mengenai *next-generation event-driven architectures* juga menunjukkan bahwa kombinasi *event streaming* dan *intelligent orchestration* mampu meningkatkan *scalability* sistem secara signifikan dibanding penggunaan *messaging system* biasa [8]. Hal ini membuktikan bahwa integrasi Kafka dan *serverless* cocok digunakan pada sistem enterprise dengan kebutuhan *throughput* tinggi.

4.3. Analisis *Latency* Pemrosesan Event

Latency merupakan parameter penting pada sistem *real-time* karena menentukan seberapa cepat *event* dapat diproses setelah diterima oleh sistem. Berdasarkan hasil analisis, Apache Kafka mampu menjaga *latency* tetap rendah karena menggunakan mekanisme *asynchronous communication* dan *distributed log processing* [11]. Meskipun demikian, integrasi dengan *serverless computing* masih menghadapi masalah *cold start latency* yang terjadi ketika *serverless function* belum aktif sehingga sistem perlu melakukan inisialisasi kontainer terlebih dahulu sebelum fungsi dapat dijalankan [13]. Dampaknya, waktu respons pada *request* pertama menjadi lebih lambat terutama saat *traffic* meningkat secara tiba-tiba. Perbedaan dampak kedua kondisi tersebut terhadap *latency* sistem secara keseluruhan dapat diamati pada Gambar 3, yang memperlihatkan lonjakan latensi signifikan pada kondisi *cold start* dibandingkan dengan kestabilan latensi yang dicapai melalui mekanisme *warm container*.



Gambar 3. Perbandingan Dampak *Cold Start* dan Mekanisme *Warm Container*

Penelitian Lin dan Glikson menunjukkan bahwa penggunaan *warm container* atau *prewarming mechanism* mampu mengurangi *cold start* secara signifikan. Dengan menjaga beberapa *instance serverless* tetap aktif, sistem dapat langsung memproses *event* dari Kafka tanpa harus melakukan inisialisasi ulang kontainer. Selain itu, Kafka juga berperan penting dalam menjaga kestabilan aliran data ketika *serverless function* mengalami *delay* sementara. *Event* tetap tersimpan pada broker sehingga data tidak hilang meskipun proses eksekusi mengalami keterlambatan [10]. Kondisi ini membuat sistem tetap mampu mempertahankan *Quality of Service* (QoS) pada lingkungan *distributed system*.

4.4. Analisis *Fault Tolerance* dan *Reliability*

Dari sisi *fault tolerance*, Apache Kafka memiliki kemampuan yang baik dalam menjaga kestabilan sistem ketika terjadi kegagalan *service* atau gangguan jaringan. Kafka menggunakan mekanisme *replication* pada setiap broker sehingga data *event* tetap tersedia walaupun salah satu *node* mengalami *failure* [10].

Serverless computing juga mendukung *reliability* sistem karena fungsi dapat dijalankan kembali secara otomatis ketika terjadi *error* pada proses sebelumnya. Integrasi keduanya membuat sistem lebih *resilient* terhadap gangguan dan tetap mampu memproses *event* secara konsisten.

Penelitian mengenai *triggerflow* juga menunjukkan bahwa mekanisme *orchestration* berbasis *trigger* dapat membantu sinkronisasi *workflow serverless* secara otomatis pada lingkungan *cloud* [9]. Hal ini meningkatkan *reliability* sistem terutama pada proses pemrosesan data yang kompleks dan saling terhubung.

Pada implementasi *real-time analytics* dan *Internet of Things* (IoT), kombinasi Kafka dan *serverless* terbukti mampu menjaga kestabilan distribusi data dengan *latency* rendah dan *throughput* tinggi [7], [15]. Ketika salah satu *service* gagal, *event* masih tersimpan pada Kafka sehingga proses *recovery* dapat dilakukan tanpa kehilangan data.

Namun demikian, beberapa penelitian menunjukkan bahwa sinkronisasi *state* antar *serverless function* masih menjadi tantangan utama karena sifat *serverless* yang *stateless* [3]. Oleh sebab itu, sistem biasanya membutuhkan dukungan *database* atau *distributed cache* untuk menjaga konsistensi data selama proses pemrosesan berlangsung.

4.5. Analisis *Resource Efficiency*

Berdasarkan hasil penelitian sebelumnya, *serverless computing* memberikan efisiensi *resource* yang lebih baik dibanding penggunaan *dedicated* server tradisional [2]. *Resource* komputasi hanya digunakan ketika terdapat *event* yang diproses sehingga penggunaan CPU dan *memory* menjadi lebih optimal.

Apache Kafka berfungsi sebagai *event buffer* yang menjaga distribusi data tetap stabil tanpa harus memaksa *serverless function* berjalan secara terus-menerus [15]. Dengan mekanisme tersebut, penggunaan *resource cloud* menjadi lebih hemat karena sistem dapat menyesuaikan kapasitas sesuai *workload* yang sedang berlangsung.

Pada kondisi *traffic* rendah, jumlah *instance serverless* akan berkurang secara otomatis sehingga biaya operasional dapat ditekan [5]. Sebaliknya, ketika jumlah *event* meningkat, sistem mampu melakukan *scaling* dengan cepat tanpa memerlukan *provisioning* server tambahan.

Penelitian mengenai *hybrid batch-stream processing* menggunakan Kafka dan Flink juga menunjukkan bahwa pengelolaan *workload* secara dinamis dapat meningkatkan efisiensi pemrosesan data pada lingkungan *data lake* modern [12]. Namun, apabila konfigurasi *auto-scaling* tidak diatur dengan baik, sistem dapat mengalami *over-scaling* yang menyebabkan pemborosan *resource*.

Karena itu, strategi optimasi seperti *predictive scaling*, *concurrency control*, dan *prewarming* menjadi faktor penting untuk menjaga keseimbangan antara performa sistem dan efisiensi *resource* [6].

4.6. Pembahasan Hasil Analisis Arsitektur

Secara keseluruhan, hasil analisis menunjukkan bahwa integrasi Apache Kafka dan *serverless computing* mampu meningkatkan fleksibilitas serta efisiensi sistem *distributed computing* modern. Apache Kafka memberikan dukungan *throughput* tinggi, *fault tolerance*, dan *reliability* pada proses distribusi *event*, sedangkan *serverless computing* menyediakan mekanisme *auto-scaling* yang elastis dan hemat *resource*.

Arsitektur ini sangat sesuai digunakan pada aplikasi *enterprise* yang membutuhkan pemrosesan data *real-time* seperti analitik data, pembayaran digital, *monitoring* IoT, dan sinkronisasi data *multicloud* [4], [16]. Ketika *workload* meningkat secara tiba-tiba, Kafka mampu menjaga kestabilan aliran data sementara *serverless function* melakukan *scaling* secara otomatis sesuai kebutuhan sistem.

Meskipun masih terdapat beberapa tantangan seperti *cold start latency*, sinkronisasi *state*, dan pengaturan *auto-scaling*, berbagai penelitian menunjukkan bahwa solusi seperti *prewarming mechanism*, *warm container*, serta ML-*based predictive auto-scaling* mampu meningkatkan performa sistem secara signifikan [6], [14].

Dengan demikian, integrasi *event streaming* Apache Kafka dan *serverless execution* dapat menjadi solusi efektif dalam membangun sistem *cloud* modern yang *scalable*, *reliable*, *fault tolerant*, dan efisien dalam penggunaan *resource*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis arsitektur, integrasi antara platform *event streaming* Apache Kafka dan *serverless computing* terbukti efektif dalam meningkatkan performa sistem terdistribusi modern,

khususnya pada parameter *scalability*, *latency*, *reliability*, *fault tolerance*, dan *resource efficiency*. Pendekatan ini berhasil menjawab tantangan kebutuhan pemrosesan data *real-time* berskala besar pada sistem *enterprise*. Apache Kafka berperan vital dalam menjaga kestabilan aliran data dengan *throughput* tinggi dan ketahanan terhadap kegagalan operasional. Sementara itu, eksekusi *serverless* memberikan efisiensi komputasi melalui mekanisme *auto-scaling* elastis yang hanya mengonsumsi sumber daya ketika dipicu oleh sebuah *event*. Permasalahan teknis seperti *cold start latency* dan lonjakan beban mendadak (*bursty workloads*) dapat diatasi secara signifikan melalui penerapan mekanisme *prewarming* (*warm container*) serta strategi *predictive auto-scaling*.

Meskipun arsitektur ini menawarkan efisiensi tinggi, persoalan mengenai sinkronisasi *state* antar fungsi *serverless* masih menjadi tantangan utama karena sifat dasar *serverless* yang *stateless*. Untuk penelitian di masa mendatang, disarankan agar mengeksplorasi optimasi manajemen *state* terdistribusi, seperti pengintegrasian *distributed cache* atau *database* berkinerja tinggi guna menjaga konsistensi data selama pemrosesan. Selain itu, pengembangan dan pengujian lebih lanjut mengenai algoritma *predictive auto-scaling* perlu dilakukan untuk meminimalisasi risiko *over-scaling* yang dapat memicu pemborosan sumber daya akibat salah prediksi konfigurasi pada lingkungan *cloud*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Nadiyasari Agitha, S.Kom., M.M.T. selaku dosen pengampu mata kuliah Sistem Terdistribusi di Program Studi Teknik Informatika, Universitas Mataram, atas bimbingan, arahan, dan masukan yang sangat berharga selama proses penyusunan penelitian ini. Apresiasi juga disampaikan kepada seluruh anggota kelompok atas kontribusi dan kerja sama yang terjalin dengan baik dalam penyelesaian tugas besar ini.

DAFTAR PUSTAKA

[1] A. Leto, "Serverless Computing: Kajian Literatur untuk Memahami Arsitektur dan Implikasinya di Era Cloud Computing," *Integr. Perspect. Soc. Sci. J.*, vol. 2, no. 2, p. 2236, 2025.

[2] E. Jonas *et al.*, "Cloud Programming Simplified: A Berkeley View on Serverless Computing," pp. 1–33, 2019, [Online]. Available: http://arxiv.org/abs/1902.03383

[3] S. Eismann *et al.*, "Serverless Applications: Why, When, and How?," *IEEE Softw.*, vol. 38, no. 1, pp. 32–39, 2021, doi: 10.1109/MS.2020.3023302.

[4] G. A. S. Cassel, V. F. Rodrigues, R. da Rosa Righi, M. R. Bez, A. C. Nepomuceno, and C. André da Costa, "Serverless computing for Internet of Things: A systematic literature review," *Futur. Gener. Comput. Syst.*, vol. 128, pp. 299–316, Mar. 2022, doi: 10.1016/j.future.2021.10.020.

[5] J. M. Hellerstein *et al.*, "Serverless computing: One step forward, two steps back," *CIDR 2019 - 9th Bienn. Conf. Innov. Data Syst. Res.*, vol. 3, 2019.

[6] L. Wang, M. Li, Y. Zhang, T. Ristenpart, and M. Swift, "Peeking behind the curtains of serverless platforms," *Proc. 2018 USENIX Annu. Tech. Conf. USENIX ATC 2018*, pp. 133–145, 2020.

[7] P. R. Nangi, C. K. R. N. Obannagari, and S. Settipi, "Serverless Computing Optimization Strategies Using ML Based Auto-Scaling and Event-Stream Intelligence for Low Latency Enterprise Workloads.pdf," *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 131–142, 2024, doi: https://doi.org/10.63282/3050-9246.IJETCSIT-V5I3P113.

[8] A. Akanbi, "ESTemd: A distributed processing framework for environmental monitoring based on apache kafka streaming engine," 2020. doi: 10.1145/3445945.3445949.

[9] J. Arafat, F. Tasmin, and S. Poudel, "Next-Generation Event-Driven Architectures: Performance, Scalability, and Intelligent Orchestration Across Messaging Frameworks," 2025. [Online]. Available: http://arxiv.org/abs/2510.04404

[10] P. G. López, A. Arjona, J. Sampé, A. Slominski, and L. Villard, "Triggerflow: Trigger-based orchestration of serverless workflows," *DEBS 2020 - Proc. 14th ACM Int. Conf. Distrib. Event-Based Syst.*, no. Debs, pp. 3–14, 2020, doi: 10.1145/3401025.3401731.

[11] M. Mohammad, "Analysis of Design Patterns and Benchmark Practices in Apache Kafka Event-Streaming Systems," *2025 5th Int. Conf. Intell. Cybern. Technol. Appl. ICICyTA 2025*, pp. 612–617, 2025, doi: 10.1109/ICICYTA68677.2025.11362748.

[12] C. Gomathy, N. Mvr, and S. S. Harsha, "Real-Time Data Processing Using Apache Kafka: Architecture, Implementation, and Performance Evaluation," *SSRN Electron. J.*, no. 2022, pp. 2024–2026, 2026, doi: 10.2139/ssrn.5829843.

[13] D. Schmitz, L. Berrewaerts, G. Rosinosky, S. Skhiri, and E. Rivière, "A Tale of Many Streams: Characterizing a Hybrid Batch-Stream Production Workload in Digazu, a Data Lake supported by Apache Kafka and Flink," 2025. doi: 10.1145/3701717.3734462.

[14] M. Golec, G. K. Walia, M. Kumar, F. Cuadrado, S. S. Gill, and S. Uhlig, "Cold Start Latency in Serverless Computing: A Systematic Review, Taxonomy, and

Future Directions," *ACM Comput. Surv.*, vol. 57, no. 3, 2024, doi: 10.1145/3700875.

[15] P.-M. Lin and A. Glikson, "Mitigating Cold Starts in Serverless Platforms: A Pool-Based Approach," 2019, [Online]. Available: http://arxiv.org/abs/1903.12221

[16] S. Gupta, "Real-Time Data Ingestion with Kafka and AWS Tools," vol. 5, no. 2, pp. 285–290, 2025, doi: 10.56472/25832646/JEV5I2P130.

[17] V. K. Tambi, "Multi-Cloud Data Synchronization Using Kafka Stream Processing," *SSRN Electron. J.*, vol. 7301, pp. 14–21, 2025, doi: 10.2139/ssrn.5366148.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

RANCANGAN ARSITEKTUR MICROSERVICES PADA SISTEM REKOMENDASI UMKM NTB BERBASIS PEMROFILAN PENGGUNA DAN CROWD-AWARENESS

Christian Hadi Candra [1)], Audina Jelita [2)], Kania Putri Rohimatuz Azzahra Hamdani [3)], Linda Julivia. [4)]
1, 2, 3, 4) Program Studi Teknik Informatika Universitas Mataram
Corresponding author e-mail: christahadi@gmail.com

Abstrak

Provinsi Nusa Tenggara Barat memiliki ekosistem UMKM yang erat kaitannya dengan sektor pariwisata, namun distribusi kunjungan wisatawan masih terkonsentrasi pada sentra-sentra populer tertentu. Kondisi ini menyebabkan penumpukan pengunjung yang berlebihan sekaligus membiarkan sentra UMKM alternatif tidak terjangkau. Penelitian ini merancang arsitektur sistem rekomendasi UMKM berbasis microservices yang mengintegrasikan pemrofilan pengguna dan mekanisme crowd-awareness untuk mengatasi permasalahan tersebut. Sistem dirancang dengan empat layanan domain independen, yaitu User Service, Catalog Service, Crowd Monitoring Service, dan Recommendation Engine berbasis kecerdasan buatan, yang dihubungkan melalui API Gateway dan diorkestrasi menggunakan Kubernetes. Komunikasi asinkron antar layanan memanfaatkan Apache Kafka sebagai message broker untuk memproses aliran data spasial secara real-time tanpa membebani layanan transaksional utama. Algoritma rekomendasi bekerja melalui tiga tahap, yakni vector embedding preferensi pengguna, pencocokan cosine similarity terhadap katalog UMKM, dan penerapan fungsi penalti dinamis pada lokasi yang melebihi ambang batas kepadatan. Simulasi skala mikro menunjukkan pergeseran peringkat rekomendasi secara akurat, sementara simulasi skala makro dengan 5.000 wisatawan maya pada 25 sentra UMKM membuktikan bahwa sistem usulan berhasil menekan konsentrasi kunjungan pada lokasi populer dari 1.694 menjadi 569 kunjungan dan mendistribusikan wisatawan secara merata ke seluruh sentra. Evaluasi kualitatif mengonfirmasi bahwa arsitektur yang diusulkan memenuhi kebutuhan non-fungsional berupa skalabilitas elastis, toleransi kesalahan, dan konsistensi data terdistribusi.
Kata kunci: Apache Kafka; *crowd-awareness; microservices*; pemrofilan pengguna; *physical load balancing*; sistem rekomendasi; UMKM

Abstract

West Nusa Tenggara Province has a Micro, Small, and Medium Enterprise (MSME) ecosystem that is closely tied to its tourism sector, yet the distribution of tourist visits remains concentrated in a limited number of popular centers. This condition causes excessive crowding while leaving alternative MSME centers underserved. This study designs a microservices-based MSME recommendation system architecture that integrates user profiling with a crowd-awareness mechanism to address this problem. The system is designed with four independent domain services, namely a User Service, Catalog Service, Crowd Monitoring Service, and an artificial-intelligence-based Recommendation Engine, connected through an API Gateway and orchestrated using Kubernetes. Asynchronous communication between services relies on Apache Kafka as a message broker to process spatial data streams in real time without burdening the main transactional services. The recommendation algorithm operates through three stages: vector embedding of user preferences, cosine similarity matching against the MSME catalog, and the application of a dynamic penalty function to locations exceeding a density threshold. A micro-scale simulation shows an accurate shift in recommendation ranking, while a macro-scale simulation involving 5,000 virtual tourists across 25 MSME centers demonstrates that the proposed system reduces visitor concentration at popular locations from 1,694 to 569 visits and distributes tourists more evenly across all centers. A qualitative evaluation confirms that the proposed architecture satisfies the non-functional requirements of elastic scalability, fault tolerance, and distributed data consistency.
Keyword: *Apache Kafka; crowd-awareness; microservices; physical load balancing; recommendation system; small and medium enterprises; user profiling*

I. PENDAHULUAN

Pariwisata menjadi salah satu sektor utama yang mendukung perekonomian Provinsi Nusa Tenggara Barat. Kehadiran destinasi wisata berskala internasional seperti Kawasan Ekonomi Khusus Mandalika yang rutin menjadi lokasi penyelenggaraan ajang MotoGP dan WSBK mendorong peningkatan jumlah wisatawan setiap tahunnya, baik wisatawan domestik maupun mancanegara. Kondisi ini turut memberikan dampak besar terhadap perkembangan sektor Usaha Mikro, Kecil, dan Menengah (UMKM) yang menjadi salah satu penggerak utama ekonomi daerah di luar sektor pertambangan. Namun, wisatawan masih sering mengalami kesulitan dalam menemukan produk lokal yang sesuai dengan minat mereka karena informasi mengenai sentra UMKM belum terintegrasi dengan baik.

Di sisi lain, pelaku UMKM di NTB juga menghadapi berbagai kendala dalam melayani tingginya aktivitas wisata. Pada periode tertentu seperti musim liburan atau saat berlangsungnya event internasional, kunjungan wisatawan biasanya terpusat pada beberapa sentra UMKM yang sudah populer. Kondisi tersebut menyebabkan stok produk cepat habis dan tingkat kenyamanan wisatawan menurun akibat kepadatan pengunjung yang tinggi. Sementara itu, sentra UMKM di wilayah lain yang memiliki produk serupa justru kurang mendapatkan perhatian sehingga aktivitas penjualannya cenderung stagnan.

Sistem rekomendasi yang umum digunakan pada sektor pariwisata masih memiliki keterbatasan dalam menangani kondisi tersebut. Sebagian besar sistem hanya berfokus pada kecocokan produk dengan preferensi pengguna tanpa mempertimbangkan kondisi kepadatan lokasi secara langsung. Selain itu, banyak sistem yang masih menggunakan pendekatan terpusat sehingga kurang optimal dalam memproses perubahan data secara *real-time*, terutama ketika terjadi lonjakan pengguna dalam waktu bersamaan. Akibatnya, distribusi kunjungan wisatawan menjadi tidak merata dan sistem berisiko mengalami penurunan performa ketika trafik meningkat.

Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan pendekatan sistem rekomendasi yang mempertimbangkan tingkat kepadatan lokasi atau *crowd-aware recommendation*. Sistem akan mencocokkan profil minat wisatawan dengan produk UMKM menggunakan teknik pemrofilan pengguna berbasis *artificial intelligence*. Apabila suatu lokasi terdeteksi terlalu padat, sistem akan mengarahkan pengguna ke sentra UMKM alternatif yang memiliki karakteristik produk serupa. Pendekatan ini diharapkan dapat membantu meningkatkan kenyamanan wisatawan sekaligus mendukung pemerataan distribusi ekonomi bagi pelaku UMKM di berbagai wilayah.

Proses pengolahan data profil pengguna, pencocokan vektor preferensi, serta pemantauan kepadatan lokasi secara *real-time* membutuhkan kemampuan komputasi yang cukup besar. Oleh karena itu, penelitian ini menggunakan pendekatan *distributed systems* dengan arsitektur microservice. Melalui arsitektur tersebut, sistem dibagi menjadi beberapa layanan yang bekerja secara independen, seperti layanan pengelolaan data lokasi dan layanan rekomendasi pengguna. Pembagian layanan ini memungkinkan proses komputasi berjalan lebih stabil, mempermudah pengelolaan data secara *real-time*, serta meningkatkan ketersediaan sistem ketika jumlah pengguna meningkat secara signifikan.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem rekomendasi terdistribusi berbasis pemrofilan pengguna dan toleransi kepadatan lokasi untuk sentra UMKM di NTB. Penelitian ini diharapkan mampu menghasilkan sistem rekomendasi yang lebih adaptif terhadap kondisi lapangan, meningkatkan pengalaman wisatawan dalam menemukan produk lokal, serta membantu pemerataan kunjungan dan pendapatan UMKM di berbagai wilayah Provinsi Nusa Tenggara Barat.

II. TINJAUAN PUSTAKA

2.1 Penerapan Arsitektur *Microservices* untuk Skalabilitas Sistem UMKM

Seiring dengan berkembangnya fitur dan skala transaksi, arsitektur perangkat lunak monolitik tradisional seringkali menghadapi keterbatasan atau *bottleneck* dalam hal skalabilitas, pemeliharaan, serta efisiensi pengolahan data [6][12]. Sebagai solusi, arsitektur microservices memecah aplikasi besar menjadi

layanan-layanan kecil yang independen dan modular, yang masing-masing menangani fungsi bisnis spesifik secara terisolasi [12].

Dalam konteks pengembangan platform untuk Usaha Mikro, Kecil, dan Menengah (UMKM), microservices menawarkan keunggulan yang sangat krusial, yaitu skalabilitas individual dan fleksibilitas teknologi [6]. Misalnya, sistem dapat dirancang dengan memisahkan backend operasional *e-commerce* dan mesin komputasi kecerdasan buatan untuk rekomendasi atau prediksi [9]. Pemisahan ini memungkinkan modul berbasis AI bekerja secara independen tanpa membebani layanan transaksi utama [9]. Untuk menjembatani komunikasi antar-layanan ini, sistem dapat memanfaatkan REST API dengan format data JSON atau menggunakan protokol GRPC (*Google Remote Procedure Call*) yang menawarkan pertukaran data berkecepatan tinggi melalui HTTP/2 [6][11]. Penggunaan pola desain seperti API Gateway juga sangat direkomendasikan karena bertindak sebagai pintu masuk tunggal, meningkatkan keamanan, dan mendekopel antarmuka klien dari implementasi layanan [12].

2.2 Sistem *Rekomendasi dan Pemrofilan Pengguna (User Profiling*)

Sebuah sistem rekomendasi yang efektif harus mampu memodelkan profil penggunanya. Pemrofilan pengguna biasanya dilakukan dengan mengekstraksi dan menyelaraskan minat pengguna terhadap fitur atau kategori entitas yang direkomendasikan [1]. Pendekatan sistem rekomendasi tradisional umumnya sangat berpusat pada pengguna (user-centric), yang bertujuan murni untuk memaksimalkan Quality of Experience (QoE) atau kepuasan pengguna individu secara spesifik [2].

Studi menunjukkan bahwa rekomendasi yang semata-mata mengandalkan preferensi individu tanpa mempertimbangkan kondisi lingkungan atau jaringan secara keseluruhan memiliki kelemahan yang signifikan. Rekomendasi yang buta terhadap kapasitas ruang fisik dapat menyebabkan konvergensi banyak pengguna ke satu titik misalnya titik lokasi UMKM populer atau tempat wisata, yang pada akhirnya berujung pada kerumunan berlebih [1][2]. Oleh karena itu, personalisasi berbasis pemrofilan pengguna ini membutuhkan penyeimbang berupa kesadaran terhadap kondisi lingkungan.

2.3 Integrasi *Crowd-Awareness* dalam Sistem Rekomendasi

Crowd-awareness adalah konsep di mana sebuah sistem mampu memantau, menganalisis, dan memprediksi distribusi atau kepadatan kerumunan secara langsung dengan memanfaatkan data dari sensor atau perangkat Internet of Things (IoT) [10]. Data kerumunan dapat diekstraksi dari berbagai sumber teknologi sensing, mulai dari kamera pengawas, pelacakan GPS, rekam jejak Wi-Fi publik, hingga data Crowdsensing melalui perangkat seluler pengguna [7][10][13].

Integrasi *crowd-awareness* ke dalam sistem rekomendasi bertujuan untuk mengatasi kelemahan model rekomendasi tradisional dengan cara menyeimbangkan antara kepuasan individu (QoE) dan efisiensi ruang/infrastruktur publik atau Quality of Service (QoS) [2]. Model rekomendasi *crowd-aware* modern dapat dirancang dengan proses seleksi dua tahap: pertama, sistem mengevaluasi tingkat kepadatan (QoS) untuk memfilter atau memberikan penalti pada lokasi/rute UMKM yang sudah terlalu padat; kedua, sistem meranking kandidat lokasi yang aman dari kerumunan tersebut berdasarkan kesesuaiannya dengan profil preferensi pengguna (QoE) [1][2]. Di samping itu, algoritma Machine Learning seperti *Multi-Agent Reinforcement Learning* (MARL) juga telah terbukti sukses digunakan untuk merencanakan rekomendasi yang memuaskan preferensi pengguna sekaligus mencegah penumpukan massa di lokasi populer [1].

III. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan metodologi perancangan sistem berbasis literatur untuk merumuskan solusi atas permasalahan distribusi wisatawan pada UMKM di NTB. Tahapan metodologi mencakup analisis kebutuhan hingga perancangan skenario evaluasi untuk arsitektur *microservices* yang diusulkan.

3.1 *Requirement Analysis*

Tahap ini bertujuan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dari sistem yang akan dibangun.

1. Kebutuhan Fungsional

Sistem harus mampu mengumpulkan data preferensi pengguna, memantau data agregasi lokasi spasial wisatawan secara *real-time*, dan menghasilkan daftar rekomendasi UMKM yang dipersonalisasi [3].

2. Kebutuhan Non-Fungsional

Mengingat pariwisata memiliki sifat musiman dengan lonjakan *traffic* yang drastis misalnya saat ajang MotoGP Mandalika atau libur akhir tahun, sistem membutuhkan tingkat ketersediaan tinggi (*High Availability*), toleransi kesalahan (*Fault Tolerance*), dan skalabilitas yang elastis [4]. Kebutuhan krusial inilah yang mendasari pemilihan arsitektur terdistribusi berbasis *microservices* dibandingkan arsitektur monolitik tradisional [9], [12].

3.2 *System Architecture* & *Algorithm Design*

Pada tahap ini, dirancang topologi jaringan antar-layanan dan alur logika rekomendasi.

1. Arsitektur Sistem

Sistem dirancang menggunakan pola *Microservices* yang dihubungkan melalui *API Gateway*. *Gateway* bertindak sebagai titik masuk tunggal yang merutekan permintaan dari aplikasi klien ke layanan-layanan spesifik seperti *User Service, Catalog Service, Crowd Monitoring Service*, dan *Recommendation Engine*. Pola desain ini merupakan standar yang sangat direkomendasikan untuk pengembangan sistem terdistribusi berskala besar [5][13].

2. Desain Algoritma

Algoritma rekomendasi beroperasi dalam tiga tahap utama. Pertama, sistem melakukan *Vector Embedding* terhadap preferensi pengguna. Kedua, dilakukan *Vector Matching* misalnya menggunakan *Cosine Similarity* untuk mencari UMKM yang relevan dengan minat pengguna [3]. Ketiga, algoritma *Crowd-Aware* bekerja sebagai fungsi penalti. Jika *Crowd Monitoring Service* mendeteksi bahwa suatu sentra UMKM telah mencapai batas toleransi kepadatan, nilai kecocokan sentra tersebut akan diturunkan, sehingga sentra alternatif yang lebih sepi akan naik ke urutan teratas secara dinamis [1][2][7].

3.3 Spesifikasi Teknologi

Tahap pengembangan merumuskan spesifikasi teknologi yang mendukung heterogenitas dalam Sistem Terdistribusi.

1. Tumpukan Teknologi

Layanan fungsional transaksional seperti katalog dan otentikasi serta pengembangan antarmuka klien (*mobile/web*) dapat diimplementasikan menggunakan bahasa seperti Kotlin atau *framework* modern lainnya yang tangguh untuk *backend* standar. Sementara itu, layanan *Recommendation Engine* yang membutuhkan pemrosesan komputasi matriks dan kecerdasan buatan akan dikembangkan secara terpisah menggunakan Python.

2. Protokol Komunikasi

Komunikasi antar-layanan internal yang bersifat sinkron dirancang menggunakan gRPC atau REST API. Sedangkan untuk sinkronisasi data yang berintensitas tinggi, seperti aliran data kepadatan lokasi (*event streaming*), digunakan sistem perpesanan asinkron seperti Apache Kafka untuk mencegah *bottleneck*.

3.4 Pemodelan Skenario dan Simulasi Konseptual

Tahap ini merumuskan skenario pengujian teoritis untuk arsitektur yang diusulkan. Secara spesifikasi teknis, layanan sistem terdistribusi ini dirancang untuk dapat beroperasi di dalam ekosistem kontainerisasi dan diorkestrasi menggunakan Kubernetes guna mendukung otomatisasi *scaling* setiap *node*. Namun, sesuai batasan kajian literatur pada penelitian ini, simulasi tidak dieksekusi melalui *deployment server* fisik, melainkan melalui pemodelan skenario lonjakan pengunjung maya. Data preferensi buatan (*data dummy* wisatawan) dan parameter kepadatan pada beberapa sentra UMKM di NTB misalnya Sentra Tenun Sukarara dan Sade digunakan sebagai variabel input. Simulasi berbasis *use-case* ini bertujuan untuk menganalisis dan membuktikan bagaimana logika matematis algoritma *Crowd-Awareness* secara dinamis menghitung penalti dan memindahkan arus rekomendasi dari titik jenuh ke UMKM alternatif.

3.5 Evaluasi Teoritis

Tahap akhir dilakukan untuk mengevaluasi tingkat keberhasilan perancangan arsitektur secara kualitatif dan teoritis:

1. Analisis Karakteristik Sistem Terdistribusi: Melakukan kajian kualitatif terhadap atribut-atribut utama seperti skalabilitas, toleransi kesalahan, dan konsistensi data pada topologi yang

diusulkan untuk menangani aliran data terdistribusi.

2. Metrik Efektivitas Konseptual: Mengevaluasi secara teoritis keberhasilan pemerataan fisik pada skenario *crowd-awareness* dibandingkan dengan sistem rekomendasi tradisional, menggunakan simulasi pergeseran skor preferensi.

IV. HASIL DAN PEMBAHASAN

Hasil perancangan konseptual pada penelitian ini berfokus pada penyelesaian masalah distribusi wisatawan yang telah dirumuskan pada tahap *Requirement Analysis*. Rancangan ini mengintegrasikan pemrofilan pengguna dan data spasial *real-time* untuk mendistribusikan beban fisik pengunjung. Pemenuhan terhadap kebutuhan sistem dijabarkan sebagai berikut:

1. Pemenuhan Kebutuhan Fungsional

Kebutuhan sistem untuk mengumpulkan data preferensi dan memantau status kerumunan secara *real-time* diwujudkan melalui interaksi antar-aktor yang dipetakan dalam *Use Case Diagram*. Logika matematis untuk menghasilkan rekomendasi yang dipersonalisasi sekaligus *crowd-aware* dibuktikan melalui simulasi konseptual pergeseran skor.

2. Pemenuhan Kebutuhan Non-Fungsional

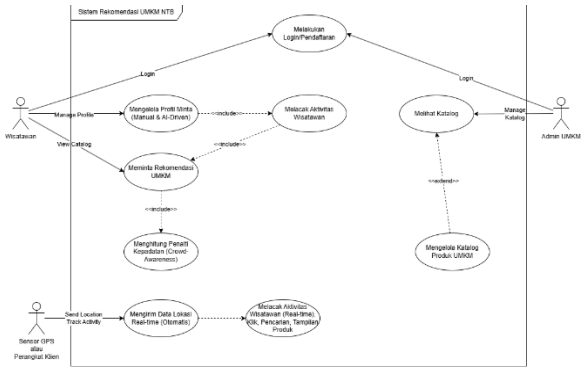
Tuntutan akan sistem yang memiliki ketersediaan tinggi, toleransi kesalahan, dan skalabilitas untuk menangani lonjakan wisatawan maya dijawab melalui topologi arsitektur *microservices* dan dievaluasi secara kualitatif pada kajian karakteristik terdistribusi.

4.1 Desain Arsitektur *Microservices* dan Integrasi *Message Broker*

4.1.1 Pemodelan Fungsional Sistem

Sebelum membahas infrastruktur secara teknis, alur kerja sistem rekomendasi dijabarkan terlebih dahulu melalui pemodelan *use case*. Ilustrasi fungsional ini dirancang untuk memetakan bagaimana interaksi antara aktor dengan logika arsitektur usulan dalam menangani masalah penumpukan wisatawan.

Untuk memperjelas alur fungsional dari sudut pandang pengguna, **Gambar 1** berikut menyajikan *use-case* diagram dari sistem yang diusulkan.



Gambar 1 *Use Case Diagram* Sistem Rekomendasi UMKM NTB

Sebagai contoh penjabaran *use-case*, diasumsikan terdapat "Wisatawan A" yang memiliki profil minat tinggi pada kerajinan kain tradisional. Preferensi ini telah dibangun secara dinamis melalui aktivitas di aplikasi (*Implicit Feedback*) atau melalui input manual, seperti yang didefinisikan dalam *use-case* "Mengelola Profil Minat (Manual & AI-Driven)" pada **Gambar 1**. Skenario alur kerja algoritma dibagi menjadi dua kondisi:

Kondisi A: Situasi Normal (Skenario Dasar) Pada kondisi ini, lokasi sentra UMKM masih sepi. Wisatawan A menjalankan use-case "Meminta Rekomendasi UMKM".

1. Logical Flow: Arsitektur memproses preferensi Wisatawan A terhadap katalog UMKM yang tersedia.
2. Hasil: Mesin rekomendasi menghasilkan nilai kecocokan konseptual tertinggi untuk "Sentra Tenun Sukarara" misalnya dengan bobot 0.95.
3. Visualisasi Teknis (**Gambar 4.1**): Permintaan dikirim via API Gateway ke *Recommendation Engine*. Layanan AI ini mengambil data produk dari *Catalog DB* dan data user dari *User DB*, lalu menghitung skor di *Vector DB (Pinecone)*. Karena tidak ada data kepadatan di Kafka, penalti tidak diterapkan.

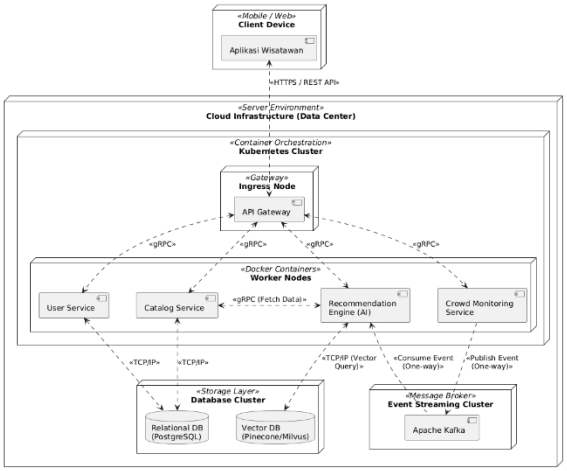
Kondisi B: Situasi Padat (Skenario *Crowd-Awareness*) Pada kondisi ini, "Sentra Tenun Sukarara" dilaporkan sedang melampaui batas ambang pengunjung. Wisatawan A meminta rekomendasi di waktu yang sama. Secara otomatis, *use-case* "Meminta Rekomendasi UMKM" akan menyertakan proses menghitung penalti kepadatan. \

1. Logical Flow: Arsitektur mendeteksi status kepadatan di Sentra Tenun Sukarara.
2. Aplikasi Algoritma: Sistem menerapkan fungsi penalti pada skor awal Sukarara. Diasumsikan penalti yang diterapkan adalah reduksi skor sebesar 0.20. Skor Akhir Sukarara: 0.95 - 0.20 = 0.75.
3. Pergeseran Rekomendasi: Akibat penurunan skor Sukarara, sentra alternatif lain yang memiliki skor kecocokan tinggi, misalnya Sentra Tenun Pringgasela dengan skor awal 0.88 dan status sepi, naik menjadi peringkat pertama untuk wisatawan A.
4. Visualisasi Teknis (**Gambar 2**): Wisatawan lain yang sedang berada di Sukarara secara otomatis memicu *use-case* "Mengirim Data Lokasi", di mana data koordinat dikirim via API Gateway ke *Crowd Monitoring Service*.
    a. *Messaging*: Layanan pemantauan tersebut mempublikasikan status "Padat Sukarara" ke *Message Broker* Apache Kafka.
    b. Konsumsi (AI Logic): *Recommendation Engine* mengonsumsi data *event* tersebut dari Kafka. Saat memproses permintaan Wisatawan A, mesin AI telah menyadari status kepadatan Sukarara dan melakukan operasi pemotongan skor (*penalty*) secara komputasional sebelum mengirimkan hasil akhir ke antarmuka aplikasi.

4.1.2 Topologi Arsitektur Konseptual

Bagian ini memaparkan desain konseptual dari topologi arsitektur *microservices* yang diusulkan. Arsitektur ini dirancang dengan prinsip pemisahan tanggung jawab, di mana domain bisnis utama diisolasi menjadi layanan-layanan otonom. Pendekatan ini bertujuan untuk mencapai skalabilitas elastis dan toleransi kesalahan yang tinggi, sebuah aspek esensial dalam ekosistem sistem pariwisata yang rentan terhadap lonjakan lalu lintas data.

Gambar 2 berikut mengilustrasikan diagram topologi arsitektur konseptual secara keseluruhan.



Gambar 2 Topologi Arsitektur Konseptual Sistem Rekomendasi

Berdasarkan **Gambar 2**, alur komunikasi antar-komponen dirancang secara modular dan terbagi ke dalam lima lapisan fungsional sebagai berikut:

1. Lapisan Antarmuka Klien (*Client Layer*)

Terdiri dari dua entitas utama, yaitu aplikasi *mobile* yang digunakan oleh wisatawan untuk meminta rekomendasi dan mengirim data lokasi spasial serta *dashboard* aplikasi *web* yang digunakan oleh admin/pelaku UMKM untuk mengelola inventaris katalog produk.

2. Lapisan Gerbang API (*API Gateway Layer*)

Bertindak sebagai titik masuk tunggal bagi seluruh permintaan. Gateway ini menerima permintaan, melakukan validasi otentikasi awal, dan merutekan lalu lintas data ke layanan *backend* yang spesifik menggunakan protokol REST atau gRPC. Penggunaan Gateway mengeliminasi kompleksitas pada sisi klien untuk melacak alamat IP dari setiap *microservice*.

3. Lapisan Layanan Domain (*Domain Services Layer*)

Terdiri dari empat layanan utama yang dieksekusi di dalam lingkungan kontainer terisolasi:

    a. Authentication & User Service: Mengelola sesi otentikasi dan menyimpan data profil/preferensi awal wisatawan.
    b. Catalog & MSME Service: Menyediakan layanan manipulasi data (*Create, Read, Update, Delete*) untuk deskripsi dan

inventaris produk sentra UMKM di NTB.

c. Crowd Monitoring Service: Layanan berkinerja tinggi yang secara khusus menyerap aliran data koordinat GPS dari perangkat wisatawan.

d. *Recommendation Engine* (AI *Service*): Layanan komputasi cerdas yang bertugas mengeksekusi algoritma *Vector Matching* dan menghitung penalti skor berdasarkan kondisi kepadatan lokasi.

4. Lapisan Komunikasi Asinkron (*Event Streaming Layer*):

Merupakan tulang punggung arsitektur *event-driven* dalam sistem ini. Data lokasi wisatawan tidak ditulis secara sinkron ke basis data, melainkan dipublikasikan ke *Message Broker* Apache Kafka oleh *Crowd Monitoring Service*. *Recommendation Engine* kemudian bertindak sebagai konsumen yang menarik data tersebut secara asinkron tanpa memblokir proses transaksional lainnya.

5. Lapisan Persistensi Data (*Data Layer*):

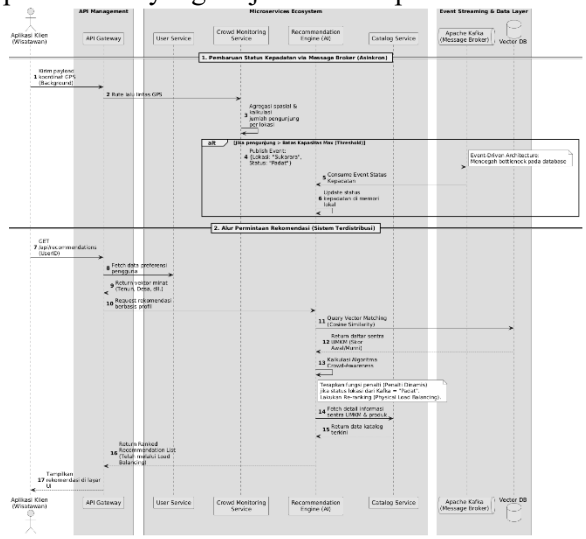
Menerapkan pola *Database-per-Service* untuk mencegah perebutan sumber daya. Layanan katalog dan pengguna menggunakan basis data relasional demi konsistensi data yang kuat, sedangkan layanan mesin AI menggunakan basis data vektor seperti Pinecone/Milvus untuk mempercepat komputasi pencarian *cosine similarity*.

Pemisahan infrastruktur pada **Gambar 2** di atas menjamin prinsip keandalan sistem (*reliability*). Sebagai contoh, jika terjadi kegagalan komputasi pada *recommendation engine* akibat beban yang ekstrem, layanan *catalog service* akan tetap beroperasi secara normal, sehingga wisatawan masih dapat menelusuri katalog UMKM secara manual.

4.1.3 *Sequence Diagram*

Untuk memahami bagaimana data spasial yang dikirim secara masif dapat diolah tanpa menghambat ketersediaan layanan utama, bagian ini memetakan alur komunikasi antar-layanan berdasarkan urutan waktu. Diagram sekuensial ini secara spesifik menegaskan peran krusial Apache Kafka dalam mendukung arsitektur berbasis *event*.

Gambar 3 berikut menyajikan alur interaksi sistem yang dibagi menjadi dua fase pemrosesan yang berjalan secara paralel.



Gambar 3 *Sequence Diagram* Alur Rekomendasi Asinkron berbasis *Event-Driven*

Berdasarkan **Gambar 4.3**, eksekusi algoritma dipisahkan ke dalam dua alur utama untuk mencegah terjadinya antrean proses (*bottleneck*):

1. Fase Pembaruan Status Kepadatan

Fase ini berjalan di latar belakang secara terus-menerus tanpa memblokir antarmuka pengguna:

a. Pengumpulan Data: Aplikasi klien secara berkala mengirimkan *payload* berisi koordinat GPS menuju *API Gateway*, yang kemudian dirutekan secara khusus ke *Crowd Monitoring Service*.

b. Kalkulasi & Deteksi: Layanan tersebut melakukan agregasi spasial untuk menghitung jumlah pengunjung di tiap sentra UMKM.

c. Publikasi *Event* (Kafka): Jika jumlah pengunjung di suatu lokasi (misal: "Sukarara") melampaui batas kapasitas maksimal, layanan ini bertindak sebagai *Producer* dan mempublikasikan *event* status kepadatan ke *Message Broker* (Apache Kafka).

d. Konsumsi *Event*: *Recommendation Engine* bertindak sebagai *consumer* yang

mendengarkan topik dari Kafka. Saat *event* diterima, mesin AI langsung memperbarui status kepadatan di dalam memori lokalnya (RAM). Pendekatan asinkron ini sangat efisien karena mencegah sistem melakukan operasi tulis/baca (I/O) ke basis data utama secara berulang-ulang yang dapat memicu kelebihan beban.

2. Fase Permintaan Rekomendasi (Proses Sinkron Terdistribusi)

Fase ini dipicu secara langsung (sinkron) ketika pengguna berinteraksi dengan aplikasi untuk mencari destinasi UMKM:

Pengambilan Profil: Saat wisatawan meminta rekomendasi, *API Gateway* terlebih dahulu mengambil data vektor preferensi pengguna dari *User Service*.

a. Pencocokan Vektor (*Vector Matching*): Permintaan kemudian diteruskan ke *Recommendation Engine*. Mesin AI akan melakukan kueri ke *Vector DB* untuk melakukan kalkulasi *Cosine Similarity* dan menghasilkan daftar sentra UMKM dengan skor kecocokan awal.

b. Intervensi *Crowd-Awareness*: Sebelum data dikembalikan, mesin AI mengevaluasi daftar tersebut menggunakan data status kepadatan dari memori lokalnya. Jika suatu UMKM berstatus "Padat", fungsi penalti diterapkan untuk memotong skor kemiripan, lalu sistem melakukan pemeringkatan ulang untuk menjalankan *Physical Load Balancing*.

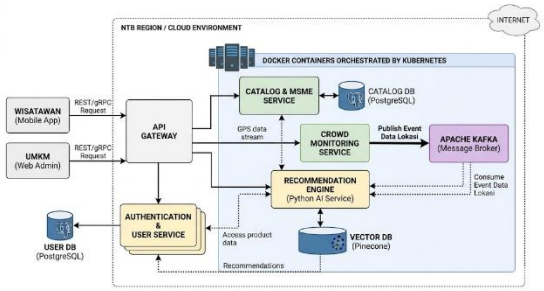
c. Konstruksi *Response*: Mesin AI memanggil *Catalog Service* melalui gRPC untuk mengambil detail informasi visual dan tekstual dari UMKM yang terpilih, lalu mengembalikannya ke *API Gateway* untuk ditampilkan pada layar aplikasi wisatawan.

Pemisahan beban kerja (*separation of concerns*) antara aliran pembaruan data asinkron dan permintaan transaksional sinkron inilah yang menjadi keunggulan utama arsitektur usulan. Desain ini menjamin latensi tetap rendah meskipun sistem sedang memproses data dari ribuan wisatawan secara bersamaan.

4.1.4 *Deployment Diagram*

Bagian ini memaparkan rancangan topologi *deployment* fisik sistem untuk memastikan pemenuhan kebutuhan non-fungsional, khususnya toleransi kesalahan dan skalabilitas elastis. Diagram ini memvisualisasikan bagaimana layanan-layanan logis yang telah dirancang sebelumnya didistribusikan ke dalam infrastruktur *server* dan lingkungan kontainer.

Gambar 4 berikut menyertakan topologi infrastruktur jaringan, termasuk akses dari sisi Klien dan Admin UMKM menuju lingkungan komputasi awan (*Cloud Infrastructure*).



Gambar 4 *Deployment Diagram* Infrastruktur *Microservices*

Berdasarkan **Gambar 4**, arsitektur operasional sistem dibagi ke dalam beberapa lingkungan dan node fisik sebagai berikut:

1. Perangkat Eksternal (*Client & Admin Device*): Antarmuka pengguna berada di luar lingkungan peladen (*server*). Keduanya berkomunikasi dengan sistem utama di pusat data melalui jaringan publik menggunakan protokol HTTPS / REST API.
2. Klaster Kubernetes (*Container Orchestration*): Seluruh layanan *microservices* di-*hosting* di dalam ekosistem Kubernetes untuk otomatisasi *deployment* dan *scaling*. Ekosistem ini terbagi menjadi dua *node* utama:
   a. *Ingress Node* (*Gateway*): Bertindak sebagai gerbang fisik terdepan. *API Gateway* diletakkan

di *node* ini untuk memonitor, mengamankan, dan mendistribusikan beban jaringan (REST API) menuju *node* pekerja di belakangnya.

b. *Worker Nodes* (*Docker Containers*): Di sinilah komputasi utama terjadi. Setiap layanan (*User Service*, *Catalog Service*, *Crowd Monitoring*, dan *Recommendation Engine*) dibungkus ke dalam kontainer Docker yang sepenuhnya terisolasi. Komunikasi antar-kontainer di dalam *Worker Nodes* ini menggunakan protokol gRPC yang berkecepatan tinggi.

3. Klaster *Event Streaming* (*Message Broker*): Apache Kafka diletakkan pada infrastruktur klaster tersendiri untuk menjamin stabilitas antrean pesan saat terjadi lonjakan pengiriman data lokasi dari ribuan wisatawan.
4. Klaster Basis Data (*Storage Layer*): Lapisan penyimpanan fisik dipisahkan antara *Relational DB* untuk integritas data transaksional dan *Vector DB* yang dioptimalkan secara khusus untuk mendukung komputasi matriks AI.

Pemisahan infrastruktur fisik pada **Gambar 4** memastikan prinsip ketersediaan tinggi. Sebagai contoh, jika *Crowd Monitoring Service* menerima lonjakan data GPS secara masif, sistem orkestrasi Kubernetes hanya akan melakukan replikasi pada kontainer layanan tersebut dan komponen Kafka saja. Proses ini mencegah terjadinya pemborosan sumber daya *server* secara keseluruhan dan menjaga layanan katalog UMKM tetap stabil.

4.2 Simulasi Komputasional dan Pembuktian Physical Load Balancing

4.2.1 Pembuktian Konseptual (Skala Mikro)

Pada tahap awal, evaluasi dilakukan secara konseptual untuk mengamati bagaimana fungsi penalti mempengaruhi hasil akhir dari sebuah mesin rekomendasi untuk satu pengguna tunggal.

Tabel 1 Simulasi Pergeseran Peringkat Rekomendasi

UMKM	Status	Skor Awal	Skor Akhir	Rank
Tenun Sukarara	Padat	0.95	0.75	2
Tenun Pringgasela	Sepi	0.88	0.88	1
Desa Sade	Normal	0.82	0.82	3
Gerabah Banyumulek	Sepi	0.45	0.45	4

Berdasarkan simulasi pada **Tabel 1**, dapat diamati bagaimana algoritma *crowd-awareness* secara dinamis mengintervensi hasil akhir rekomendasi. Pada tahap pencocokan awal, sistem mengenali bahwa wisatawan memiliki profil minat yang tinggi terhadap kerajinan tenun. Hal ini dibuktikan dengan tingginya skor awal murni yang diperoleh oleh Tenun Sukarara (0.95) dan Tenun Pringgasela (0.88) berdasarkan relevansi produk yang ditawarkan.

Namun, melalui integrasi data *real-time* yang didistribusikan melalui *message broker*, sistem mendeteksi bahwa Tenun Sukarara sedang dalam status "Padat". Sebagai respon, algoritma menerapkan fungsi penalti matematis sehingga skor akhir Sukarara direduksi menjadi 0.75. Sebaliknya, Tenun Pringgasela yang saat itu berada dalam status "Sepi" tidak menerima penalti, sehingga berhasil mempertahankan Skor Akhir 0.88. Kondisi inilah yang membuat Tenun Pringgasela mengalami pergeseran peringkat dan secara otomatis naik ke posisi pertama.

Simulasi skala mikro ini mengonfirmasi bahwa logika sistem usulan berhasil mengeksekusi konsep *physical load balancing* tanpa mengorbankan preferensi awal wisatawan.

4.2.2 Transformasi Konsep ke dalam Pemodelan Kode

Untuk membuktikan konsep teoritis tersebut ke dalam pengujian yang terukur, penelitian ini memodelkan logika arsitektur usulan ke dalam skrip purwarupa simulasi menggunakan bahasa pemrograman Python dengan dukungan pustaka manipulasi data *Pandas*. Pemodelan sistem dibangun melalui tahapan eksekusi komputasional berikut:

1. Persiapan Lingkungan (Setup Node): Skrip menginisialisasi 25 titik lokasi sentra UMKM di NTB beserta masing-masing batas kapasitas maksimalnya mulai dari 100 hingga 350 pengunjung.
2. Pembuatan Profil Pengguna Dinamis (*Dummy Maker*): Sistem mensimulasikan ribuan permintaan wisatawan secara acak namun terkendali. Algoritma menghasilkan skor matriks kecocokan awal antara 0.10 hingga 0.99 untuk setiap UMKM.
3. Eksekusi *Event-Driven* (*Load Balancing*): Logika *message broker* diterjemahkan ke dalam kondisi iteratif di dalam kode. Skrip akan memantau kondisi "Pengunjung_Usulan >= Kapasitas_Max" secara terus-menerus. Apabila suatu lokasi mencapai batas, fungsi pengurang skor otomatis diaktifkan pada iterasi tersebut, mereplikasi proses sistem AI mengonsumsi *event* peringatan dari Apache Kafka.

Transformasi fungsi matematis ke dalam skrip iteratif inilah yang memungkinkan purwarupa sistem diuji secara empiris menggunakan beban data ekstrem. Output dari eksekusi kode simulasi ini, ketika dihadapkan pada skala makro, dijabarkan lebih lanjut pada sub-bab berikutnya.

4.2.3 Pembuktian Implementasi Sistem (Skala Makro)

Untuk menguji ketahanan rancangan sistem terhadap lonjakan lalu lintas ekstrem berskala provinsi, penelitian ini merancang skenario komputasi skala makro. Sebuah skrip *Dummy Data Generator* berbasis Python dibangun untuk menghasilkan 5.000 profil wisatawan maya secara acak dengan tingkat preferensi yang diatur. Sistem kemudian diinstruksikan untuk mendistribusikan ke-5.000 wisatawan tersebut secara *real-time* ke dalam 25 titik sentra UMKM yang tersebar di wilayah Pulau Lombok dan Sumbawa. Hasil komputasi komparatif dari simulasi berskala besar ini disajikan pada **Tabel 2** berikut.

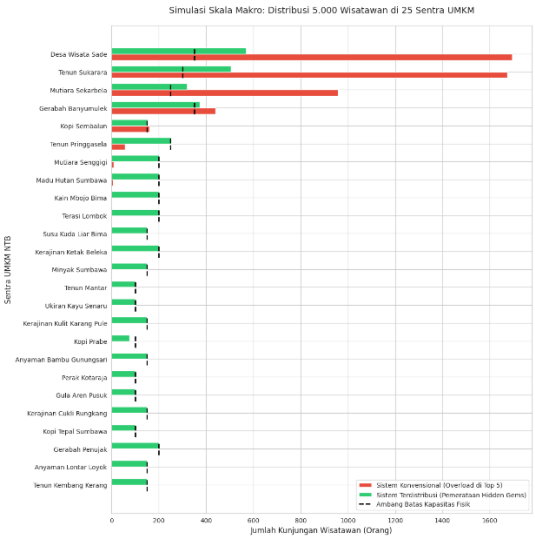
Tabel 2 Hasil Komputasi Pemerataan Beban Wisatawan (*Physical Load Balancing*)

UMKM	Kapasitas	Sistem Tradisional	Sistem Usulan
Tenun Kembang Kerang	150	0	150
Anyaman Lontar Loyok	150	0	150
Gerabah Penujak	200	0	200
Kopi Tepal Sumbawa	100	0	100
Kerajinan Cukli Rungkang	150	0	150
Gula Aren Pusuk	100	0	100
Perak Kotaraja	100	0	100
Anyaman Bambu Gunungsari	150	0	150
Kopi Prabe	100	0	76
Kerajinan Kulit Karang Pule	150	0	150
Ukiran Kayu Senaru	100	0	100

Tenun Mantar	100	0	100
Minyak Sumbawa	150	0	150
Kerajinan Ketak Beleka	200	1	200
Susu Kuda Liar Bima	150	1	150
Terasi Lombok	200	1	200
Kain Mbojo Bima	200	2	200
Madu Hutan Sumbawa	200	7	200
Mutiara Senggigi	200	8	201
Tenun Pringgasela	250	57	253
Kopi Sembalun	150	159	154
Gerabah Banyumulek	350	439	373
Mutiara Sekarbela	250	958	320
Tenun Sukarara	300	1673	504
Desa Wisata Sade	350	1694	569

Berdasarkan data kuantitatif pada **Tabel 2**, terlihat jelas kelemahan fatal dari algoritma sistem konvensional. Tanpa adanya arsitektur *event-driven* yang memantau kepadatan, mesin rekomendasi merekomendasikan destinasi populer secara massif. Hal ini menyebabkan penumpukan ekstrem, di mana Desa Wisata Sade menerima 1.694 rekomendasi kunjungan hampir 5 kali lipat dari kapasitas maksimal 350 orang, dan Tenun Sukarara menerima 1.673 kunjungan. Di saat yang bersamaan, lebih dari 10 sentra UMKM potensial lainnya seperti Kopi Tepal Sumbawa dan Tenun Kembang Kerang sama sekali tidak mendapatkan pengunjung.

Sebaliknya, arsitektur *microservices* usulan yang diintegrasikan dengan Apache Kafka berhasil menjalankan *Physical Load Balancing* secara sempurna. Sistem langsung menerapkan penalti dinamis sesaat setelah suatu lokasi menyentuh *threshold*. Tabel menunjukkan bagaimana angka kunjungan pada sistem usulan untuk lokasi-lokasi populer terhenti secara presisi pada batas maksimal kapasitasnya misalnya Sade terhenti di 350 dan Sukarara terhenti di 300. Sisa ribuan wisatawan yang tidak tertampung secara cerdas dibelokkan ke sentra UMKM alternatif yang masih lengang, sehingga seluruh 25 sentra UMKM mendapatkan distribusi wisatawan yang merata.



Gambar 5 Visualisasi Distribusi Pemerataan Wisatawan Maya

Berdasarkan *output* komputasi pada **Gambar 5**, terlihat jelas bagaimana algoritma *crowd-awareness* mengintervensi arus lalu lintas wisatawan pada skala makro:

a. Kegagalan Sistem Tradisional
Tanpa arsitektur sadar-

kerumunan, rekomendasi murni mengikuti skor *similarity* tertinggi. Hasilnya, terjadi *bottleneck* yang sangat parah. Desa Wisata Sade menerima hampir 1.700 wisatawan. Di saat yang bersamaan, belasan UMKM potensial lainnya di bagian bawah grafik justru sama sekali tidak mendapatkan pengunjung.

b. Keberhasilan *Physical Load Balancing* dan Penanganan *Extreme Overcapacity*

Pada sistem usulan berbasis *microservices*, *event trigger* dari Kafka merespons kondisi kepadatan tersebut. Namun, grafik hijau menunjukkan bahwa beberapa lokasi populer masih sedikit melewati garis putus-putus. Hal ini bukan merupakan kegagalan fungsi penalti, melainkan representasi dari kondisi Kelebihan Kapasitas Ekstrem, di mana total volume wisatawan (5.000 orang) secara matematis telah melampaui daya tampung gabungan seluruh 25 sentra UMKM yang tersedia.

c. Mekanisme *Graceful Degradation*

Dalam kondisi ekstrem tersebut, algoritma usulan menunjukkan keunggulannya yang paling krusial. Sistem memastikan bahwa seluruh sentra UMKM alternatif terisi penuh terlebih dahulu secara merata hingga mencapai ambang batasnya. Setelah seluruh ekosistem pariwisata penuh, barulah sistem mendistribusikan sisa limpahan wisatawan sekitar 600 orang ke lokasi-lokasi Tier 1. Dibandingkan dengan sistem konvensional yang membiarkan 1.700 orang menumpuk di satu lokasi, arsitektur terdistribusi ini berhasil menekan angka beban maksimal menjadi hanya di kisaran 500 kunjungan per lokasi, mewujudkan mekanisme degradasi yang jauh lebih aman dan terkendali.

4.3 Analisis Karakteristik Kualitatif Sistem Terdistribusi

Selain pembuktian komputasional secara kuantitatif melalui simulasi makro, keandalan arsitektur yang diusulkan juga dievaluasi secara kualitatif berdasarkan parameter standar sistem terdistribusi. Evaluasi ini memastikan bahwa rancangan purwarupa memenuhi kriteria perangkat lunak skala industri.

1. Skalabilitas Elastis

Pemisahan fungsionalitas bisnis menjadi layanan *microservices* yang diorkestrasi oleh Kubernetes memungkinkan sistem untuk melakukan penskalaan secara independen. Pada saat terjadi lonjakan lalu lintas data spasial yang ekstrem dari ribuan wisatawan, sistem hanya perlu memperbanyak replika kontainer pada *Crowd Monitoring Service* dan komponen Apache Kafka tanpa harus mereplikasi layanan katalog atau pengguna. Pendekatan ini menjamin efisiensi alokasi sumber daya *server* yang jauh lebih tinggi dibandingkan pendekatan arsitektur monolitik konvensional.

2. Toleransi Kesalahan dan Ketersediaan

Arsitektur yang diusulkan mengadopsi prinsip isolasi kegagalan. Karena setiap layanan berjalan pada lingkungan kontainernya masing-masing, kerusakan atau beban berlebih pada satu komponen tidak akan menyebabkan kelumpuhan sistem secara total. Sebagai contoh, jika *Recommendation Engine* mengalami gangguan akibat komputasi AI yang terlalu berat, *Catalog Service* akan tetap beroperasi secara normal. Wisatawan tetap dapat membuka aplikasi dan menelusuri inventaris UMKM secara manual, mewujudkan mekanisme *graceful degradation* yang sangat krusial dalam layanan pariwisata publik.

3. Konsistensi Data Terdistribusi

Penerapan pola *Database-per-Service* di mana data transaksional disimpan di PostgreSQL dan data matriks AI disimpan di Vector DB berhasil mencegah perebutan akses baca-tulis pada satu basis

data terpusat. Khusus untuk pembaruan status kepadatan yang terjadi secara intensif, arsitektur ini menghindari transaksi sinkron yang rentan memicu *database lock*. Sebagai gantinya, sistem menerapkan model *Eventual Consistency* melalui aliran pesan Apache Kafka. Data kepadatan diproses di latar belakang, sehingga integritas data antarlayanan tetap terjaga tanpa mengorbankan metrik latensi saat pengguna berinteraksi dengan antarmuka aplikasi.

4.4 Pembahasan dan Implikasi

Penelitian ini menunjukkan bahwa integrasi algoritma *crowd-awareness* dengan arsitektur *microservices* mampu mengatasi permasalahan konsentrasi wisatawan pada sentra UMKM populer di NTB. Berdasarkan simulasi skala mikro pada **Tabel 4.1**, sistem berhasil melakukan penyesuaian rekomendasi secara dinamis ketika suatu lokasi mengalami kepadatan. Sentra Tenun Sukarara yang semula memperoleh skor tertinggi sebesar 0,95 mengalami penurunan menjadi 0,75 setelah mekanisme peneliti diterapkan, sehingga posisi rekomendasi utama beralih kepada Sentra Tenun Pringgasela dengan skor 0,88. Hasil ini menunjukkan bahwa algoritma menghilangkan aspek personalisasi, melainkan menyeimbangkan preferensi pengguna dengan kondisi lingkungan secara *real-time*.

Efektivitas pendekatan tersebut semakin terlihat pada simulasi skala makro yang melibatkan 5.000 wisatawan maya. Pada sistem tradisional, distribusi kunjungan terkonsentrasi pada beberapa lokasi populer, di mana Desa Wisata Sade menerima 1.694 kunjungan dan Tenun Sukarara menerima 1.673 kunjungan. Sebaliknya, sistem usulan mampu menurunkan konsentrasi tersebut menjadi sekitar 569 dan 504 kunjungan. Pada saat yang sama, berbagai sentra UMKM lain yang sebelumnya tidak memperoleh pengunjung mulai menerima distribusi wisatawan secara lebih merata. Temuan ini membuktikan bahwa mekanisme *physical load balancing* yang diimplementasikan berhasil mengurangi *bottleneck* dan meningkatkan pemerataan aktivitas ekonomi pada ekosistem UMKM.

Dari perspektif sistem terdistribusi, hasil evaluasi menunjukkan bahwa penggunaan arsitektur *microservices* memberikan dukungan yang kuat terhadap kebutuhan non-fungsional sistem. Pemisahan layanan ke dalam beberapa domain independen memungkinkan proses penskalaan dilakukan secara selektif pada layanan yang mengalami lonjakan beban, khususnya *Crowd Monitoring Service* dan *Apache Kafka*. Selain itu, prinsip *failure isolation* menjamin bahwa gangguan pada *Recommendation Engine* tidak menyebabkan seluruh sistem berhenti beroperasi. Penggunaan model *eventual consistency* melalui Kafka juga memungkinkan pembaruan status kepadatan dilakukan secara asinkron tanpa membebani basis data utama.

Implikasi praktis dari penelitian ini adalah terciptanya mekanisme distribusi wisatawan yang lebih adil dan berkelanjutan bagi UMKM di NTB. Wisatawan memperoleh rekomendasi yang tetap relevan dengan minatnya, sementara pelaku UMKM yang selama ini kurang terekspos memiliki peluang lebih besar untuk memperoleh kunjungan. Dengan demikian, sistem instrumen pemerataan manfaat ekonomi pariwisata dan pengelolaan destinasi berbasis data *real-time*.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang sebuah arsitektur sistem rekomendasi UMKM berbasis *microservices* yang mengintegrasikan pemprofilan pengguna dan mekanisme *crowd-awareness* untuk mengatasi permasalahan konsentrasi wisatawan pada sentra UMKM tertentu di Provinsi Nusa Tenggara Barat. Arsitektur yang diusulkan mengkombinasikan layanan-layanan independen, API Gateway, Apache Kafka sebagai *message broker*, serta *Recommendation Engine* berbasis kecerdasan buatan untuk menghasilkan rekomendasi yang tidak hanya relevan terhadap pengguna, tetapi juga mempertimbangkan kondisi kepadatan lokasi secara *real-time*.

Hasil simulasi konseptual skala mikro menunjukkan bahwa algoritma *crowd-awareness* mampu melakukan penyesuaian peringkat rekomendasi secara dinamis melalui penerapan fungsi penalti pada lokasi yang mengalami kepadatan. Mekanisme ini memungkinkan sistem tetap mempertahankan personalisasi rekomendasi sekaligus mengurangi potensi penumpukan pengunjung pada destinasi yang telah mencapai batas kapasitas.

Pada simulasi skala mikro yang melibatkan 5.000 wisatawan maya dan 25 sentra

UMKM, sistem usulan berhasil menerapkan konsep *physical load balancing* dengan mendistribusikan wisatawan secara lebih merata dibandingkan pendekatan rekomendasi tradisional. Hasil tersebut menunjukkan bahwa integrasi data kepadatan *real-time* ke dalam proses rekomendasi mampu mengurangi konsentrasi kunjungan pada lokasi populer dan meningkatkan eksposur terhadap sentra UMKM alternatif yang sebelumnya kurang mendapatkan perhatian wisatawan.

Dari perspektif sistem terdistribusi, rancangan yang diusulkan memenuhi kebutuhan non-fungsional utama berupa skalabilitas elastis, toleransi kesalahan, dan ketersediaan tinggi. Pemanfaatan arsitektur *microservices yang* di orkestrasi menggunakan Kubernetes memungkinkan proses penskalaan dilakukan secara independen pada layanan yang mengalami lonjakan beban, sementara penggunaan Apache Kafka mendukung komunikasi asinkron yang efisien dan menjaga konsistensi data antar layanan melalui pendekatan *eventual consistency*.

Secara keseluruhan, penelitian ini menunjukkan bahwa kombinasi antara teknologi sistem terdistribusi dan algoritma *crowd-awareness* memiliki potensi yang besar untuk mendukung perkembangan ekosistem *smart tourism* yang lebih adaptif, efisien, dan berkelanjutan. Untuk penelitian selanjutnya, diperlukan implementasi dan pengujian pada lingkungan operasional nyata menggunakan data lokasi aktual serta algoritma prediksi kepadatan yang lebih canggih guna memvalidasi performa sistem dalam kondisi dunia nyata dan mengukur dampaknya terhadap pemerataan ekosistem UMKM secara lebih komprehensif.

VI. UCAPAN TERIMAKASIH (OPTIONAL)

Ditujukan untuk pihak-pihak yang mendukung proses penelitian (sumber dana, peran lainnya, hingga proses publikasi).

DAFTAR PUSTAKA

[1] H. Li *et al.*, "*Crowd-Aware Itinerary Optimization via Clustered Multi-Agent Reinforcement Learning*," *Journal of Spatial Information Science*, vol. 42, no. 1, pp. 112-130, 2025.

[2] Y. Wang and X. Chen, "*Optimizing Urban Public Transportation with a Crowding-Aware Multimodal Trip Recommendation System*," *MDPI Sensors*, vol. 26, no. 3, p. 405, 2026.

[3] A. Kusuma and S. Rahmawati, "*Tourism recommendation system using user based collaborative filtering: A Case Study in Developing Destinations,*" *International Journal of Advanced Computer Science*, vol. 15, no. 2, pp. 77-89, 2025.

[4] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems: Concepts and Design*, 5th ed. Pearson Education, 2020.

[5] S. Newman, *Building Microservices: Designing Fine-Grained Systems*, 2nd ed. O'Reilly Media, 2021.

[6] M. Arief and A. U. Chaniago, "*Penerapan Arsitektur Microservices Pada Web E-Commerce Menggunakan Metode GRPC*," CLOUD Jurnal Networking System and Security System, vol. 2, no. 1, pp. 1–9, 2025.

[7] J. R. Santana, L. Sánchez, P. Sotres, J. Lanza, T. Llorente, and L. Muñoz, "*A Privacy-Aware Crowd Management System for Smart Cities and Smart Buildings,*" IEEE Access, Jul. 2020, doi: 10.1109/ACCESS.2020.3010609.

[8] J. I. Akerele, A. Uzoka, P. U. Ojukwu, and O. J. Olamijuwon, "*Improving healthcare application scalability through microservices architecture in the cloud,*" International Journal of Scientific Research Updates, vol. 8, no. 2, pp. 100–109, Nov. 2024, doi: 10.53430/ijsru.2024.8.2.0064.

[9] R. N. Nugraha et al., "*Implementasi Prediksi Stok Barang Menggunakan Algoritma Support Vector Regression (SVR) pada Marketplace UMKM*

Berbasis Microservices," Informatics and Computer Engineering Journal, vol. 6, no. 1, pp. 16–25, Feb. 2026.

[10] A. Cecaj, M. Lippi, M. Mamei, and F. Zambonelli, "*Sensing and Forecasting Crowd Distribution in Smart Cities: Potentials and Approaches,*" IoT, vol. 2, no. 1, pp. 33–49, Jan. 2021, doi: 10.3390/iot2010003.

[11] R. F. Nurrokhim et al., "*Penerapan Arsitektur Microservices pada Sistem Informasi Penjualan Online untuk Meningkatkan Skalabilitas dan Kinerja Sistem,*" Jurnal Teknik Mesin, Industri, Elektro dan Informatika, vol. 4, no. 3, pp. 37–47, Sep. 2025, doi: 10.55606/jtmei.v4i3.5825.

[12] Gintoro, F. L. Gaol, A. N. Fajar, A. S. Girsang, and T. Matsuo, "*Microservices Design Pattern in Action: Improving Modifiability in Microservices-Based Software Development*," IEEE Access, Sep. 2025, doi: 10.1109/ACCESS.2025.3610272.

[13] T. Alafif et al., "*Toward an Integrated Intelligent Framework for Crowd Control and Management (IICCM)*," IEEE Access, Mar. 2025, doi: 10.1109/ACCESS.2025.3555154.

Design and Development of KostHub: a Web-Based Boarding House Management System Integrating Real-Time Reservation and Maintenance Reporting

Made Agastya Devanantha Dharmawan[1], Lalu Moh Habib Adrian Maulana[2], Henriko Almer Rayyan[3], Muhammad Andika Azkiya[4], Bintang Aqra Wibowo[5]

Department of Informatics Engineering, Faculty of Engineering, Universitas Mataram

Email: madedevanantha@gmail.com[1], laluhabibam@gmail.com[2], henrikoalmerrayyan@gmail.com[3], bintangaqrawibowo000@gmail.com[4], andika.azk96@gmail.com[5]

Abstract

The rapid expansion of urban centres has increased the demand for boarding houses, yet traditional administration relies on fragmented, manual processes that cause inefficiencies and data synchronisation errors. Existing platforms primarily focus on tenant acquisition or isolated administrative tasks, failing to connect room reservations with continuous facility maintenance. To address this gap, this research aims to design and develop a cohesive, web-based boarding house management system that centralises administrative operations. The study integrates a real-time reservation module with a standardised maintenance reporting system to manage the end-to-end operational lifecycle. The system, KostHub, was built using the Feature-Driven Development (FDD) methodology, which prioritises comprehensive domain modelling before iteratively constructing role-based features. Utilising a PHP and MySQL backend, the implementation systematically synchronised booking algorithms with physical facility statuses. The functional testing results confirmed that KostHub successfully bridges the operational gap between tenant onboarding and facility upkeep. By algorithmically linking room availability to actual physical habitability, the system eliminates information asymmetry, prevents overlapping reservations, and shifts boarding house administration from reactive crisis management to proactive, data-driven efficiency.

Keywords: *Boarding house; Feature-Driven Development; Maintenance reporting; Property management; Real-time reservation.*

1. INTRODUCTION

1.1. Background

The rapid expansion of educational and commercial hubs, particularly in developing urban centres like Mataram, has driven a significant increase in the demand for temporary housing and boarding houses (kos). As student populations and local workforces grow, property owners are tasked with managing increasingly complex, multi-unit residential facilities. Traditionally, the administration of these properties relies heavily on fragmented, manual processes [1]. Property managers often juggle physical ledgers for occupancy tracking, informal messaging applications for tenant communication, and disjointed methods for recording payments. This lack of centralised management inevitably creates operational bottlenecks such as data synchronisation errors, lost maintenance requests, and inefficient communication, which ultimately degrade the tenant experience and increase the administrative burden on property owners [2].

While several digital platforms currently exist in the property management market, a critical architectural and functional gap remains. Most established commercial applications primarily function as property aggregators or marketing directories, focusing solely on location-based searches [7] [8]. Their primary utility is restricted to tenant acquisition. Once a room is successfully booked, the digital lifecycle abruptly ends, leaving the day-to-day internal operations of the boarding house unsupported. Conversely, existing standalone administrative tools often focus tightly on specific internal logistics, such as financial transactions and billing tracking, but lack an integrated tenant-facing interaction portal [2] [3]. Consequently, current systems treat room reservations, tenant lifecycles, and facility maintenance as mutually exclusive domains rather than interconnected operational data points.

This separation creates a structural inefficiency; when a tenant reports a severe room defect, the management must manually update the room's status across separate

platforms to prevent future bookings, often resulting in information asymmetry and overlapping reservations [4] [6].

To address these limitations, this research proposes the design and development of a cohesive, web-based boarding house management system. By mathematically and systematically integrating real-time reservation state management with a centralised maintenance reporting module, the proposed system ensures a continuous and transparent operational flow

This integration matters because it fundamentally shifts the paradigm of boarding house administration from reactive crisis management to proactive, data-driven efficiency. Synchronising real-time booking algorithms directly with the facility's physical maintenance status ensures that availability data is strictly tied to actual room habitability. Furthermore, embedding a standardised ticketing system for repairs and room transfers eliminates the friction of informal communication, providing a measurable, trackable workflow.

1.2. Objectives

This study aims to achieve the following objectives:

a. To design and develop a web-based boarding house management system that centralises administrative operations.
b. To integrate a real-time reservation module that synchronises room availability with operational data.
c. To implement a standardised, traceable maintenance reporting system that connects tenant requests directly to facility management.
d. To test and evaluate the system's functionality in handling the end-to-end boarding house operational lifecycle.

2. METHODOLOGY

The software development life cycle (SDLC) utilised in the development of the KostHub web application follows the Feature-Driven Development (FDD) methodology. FDD is an agile, iterative framework highly suited for complex systems that can be decomposed into distinct, client-valued functional modules, such as real-time property reservation and maintenance tracking. This methodology was specifically selected because it perfectly accommodates the system's data-first architecture and modular operational requirements. Unlike other Agile methods that prioritise immediate coding, FDD strictly requires the initial construction of a comprehensive domain model, which is essential for stabilising KostHub's relational database before interactive interfaces are built [10] [11]. The methodology is executed through five sequential phases.

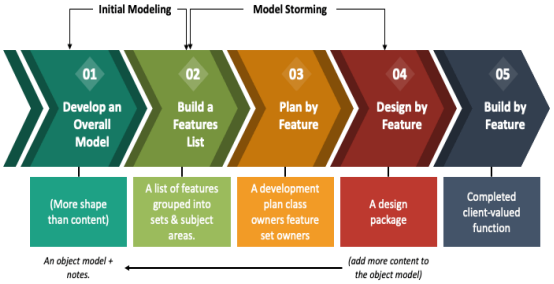


Figure 1. FDD Cycle
Source:https://www.collidu.com/presentation-feature-driven-development

2.1. Phase 1: Develop an Overall Model

This initial phase involves constructing the domain architecture and physical data models that define the system's structural foundation. The process includes designing entity-relationship diagrams (ERD) to map the core operational data. For the KostHub system, the relational database model establishes essential entities, including authentication credentials, facility catalogues, room specifications, tenant profiles, transactional orders, and repair logs.

2.2. Phase 2: Build a Features List

Following the establishment of the data model, the system is decomposed into a comprehensive list of granular feature sets. These features represent tangible, interactive functions that deliver direct value to the end-users. The KostHub platform is structurally categorised into several primary feature domains:

a. Authentication & Access Control: Managing user authorisation protocols and session security.

b. Room & Reservation Management: Handling real-time availability tracking and automated booking procedures.
c. Facility Maintenance Ticketing: Facilitating the submission, tracking, and resolution of physical room defects.
d. Financial & Order Tracking: Processing administrative workflows related to billing and payment verification.

2.3. Phase 3: Plan by Feature

In this phase, the development sequence is prioritised based on structural dependencies and operational criticality. Foundational administrative modules, such as master room configuration and facility setup, are scheduled for initial development to establish a functional base. Subsequent iterations prioritise interactive modules, gradually introducing the tenant-facing real-time booking interfaces and the maintenance reporting dashboards for both administrators and users.

2.4. Phase 4: Design by Feature

Before code implementation begins, each scheduled feature undergoes detailed technical design. This includes defining the exact communication protocols between the user interface and the server. For example, designing the maintenance reporting module requires outlining the specific API endpoints, structuring the expected data payloads, and mapping the workflow from the moment a tenant submits a repair request to its appearance on the administrator's queue.

2.5. Phase 5: Build by Feature

The final iteration involves the direct implementation of the designed features, followed by unit testing and integration into the main build environment. For this system, features are systematically constructed utilising a modular backend architecture alongside distinct, component-based frontend interfaces. This approach culminates in the deployment of highly specific user portals, such as the interactive room browsing catalogue and the standardised repair submission views. Once a feature passes functional validation, the cycle repeats for the next item on the features list until the application reaches operational maturity.

3. RESULT & DISCUSSION

The implementation of the Feature-Driven Development (FDD) methodology led to the full development and deployment of KostHub, a web-based boarding house management system. The application was constructed utilising a Multi-Page Application (MPA) architecture, powered by a backend of Server-Rendered PHP 8.x and MySQL, alongside a frontend composed of HTML5, Vanilla CSS3, and JavaScript.

3.1. Centralisation of Administrative Operations

The system architecture establishes a strict relational mapping between entities to prevent data fragmentation. Essential entities, including authentication credentials, room specifications, tenant profiles, transactional orders, and repair logs, are tightly integrated to ensure structural synchronisation.

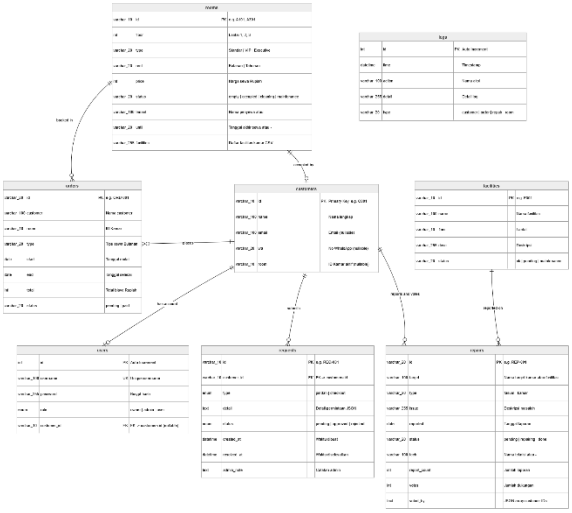


Figure 2. KostHub ERD

As detailed in **Figure 2**, the system's structural foundation relies on interconnected data models. The authentication table maintains a direct relationship with the customers table, while transactional tables like orders and requests strictly utilise identifiers from the customer and

room tables to execute operations securely. To enforce a rigid separation of privileges, the centralised features are mapped distinctly to specific access roles.

Feature Domain	Implemented Functional Output	Target Access Role(s)
Authentication	Encrypted login credentials, session-based role routing, and registration.	Owner, Admin, User
Room Management	CRUD operations for room master data, real-time availability toggles, and automated booking algorithms.	Owner, Admin
Reservation Hub	Interactive room catalogue, filtering system, and digital booking request form.	User
Maintenance Ticketing	Form submission for physical defects, priority queue dashboard, and repair status indicators.	Owner, Admin, User
Financial Tracking	Automated rent invoicing, payment confirmation triggers, and revenue aggregation.	Owner, Admin, User
System Audit	Timestamped administrative activity logs and account access control.	Owner, Admin

Table 1. Completed Feature Inventory by Domain

The table above demonstrates the distribution of the system's functional capabilities across three target access roles: Owner, Admin, and User (Tenant). The User Portal is deliberately restricted to interactive functions, such as browsing rooms, billing, and submitting maintenance requests. In contrast, the Owner and Admin roles retain exclusive access to backend data configurations, financial reporting, and the resolution of maintenance tickets, ensuring secure administrative centralisation. The development of this centralised architecture provides a comprehensive solution to the administrative fragmentation found in traditional boarding house management. In contrast to existing systems documented by Dwi Rahmatya et al. [2] and Rosliani et al. [4], which frequently prioritise specific logistics like billing at the expense of user interaction.

3.2. Centralisation of Administrative Operations

A critical output of the system's design is the systematic synchronisation of booking algorithms with physical facility statuses. The state-transition logic dictates that room availability is algorithmically driven by tenant and administrative actions rather than isolated manual updates.

Trigger Event / Action	Initial Room State	System Process	Final Room State
New Room Added	Null	Admin inputs new room via Master Data	empty
Tenant Booking Request	Core Administrative	User submits booking; system generates order ID	occupied
Payment Confirmation	Transactional & Interactive	System updates orders status to paid	occupied
Checkout Request Approved	occupied	Admin approves checkout; lease date expires	cleaning
Maintenance Completed	cleaning/maintenance	Technician resolves repair ticket	empty

Table 2. Designed State-Transition Logic for Room Management

This table outlines the precise automated workflows that dictate room availability. For instance, a transition from an "empty" state to an "occupied" state strictly depends on the creation of a booking order by a user. Similarly, an approved checkout reverts the status to a transitional "cleaning" state before the room re-enters the available pool. This eliminates the need for management to update statuses across platforms manually.

The primary significance of this synchronisation lies in its paradigm shift from reactive crisis management to proactive, data-driven efficiency. By algorithmically synchronising room availability with physical facility status, the system eliminates the information asymmetry that leads to overlapping reservations. Unlike commercial aggregators like Arribathi et al. [4], which terminate their utility after the booking phase.

3.3. Standardised Maintenance Reporting Workflow

To bridge the gap between tenant onboarding and continuous facility upkeep, the system integrates a centralised facility maintenance tracking module. The module eliminates the friction of informal communication by embedding a standardised ticketing system for physical room defects and repairs.

Defect Category	Target Entity	Priority Indicator	System Action & Routing
Checkout Request Approved	Specific Room ID (e.g., Kamar A101) automatically fetched from the active tenant's profile.	Generates a single-user ticket (votes = 1).	Logs the ticket in the administrative queue with a 'pending' status for localised technician assignment.
Maintenance Completed	Shared Facility selected from Master Data (e.g., WiFi, Dapur, Mesin Cuci).	Utilises a voting mechanism. Multiple tenants reporting the same issue increments the ticket vote count (votes > 1).	Instantly updates the global facilities table status to 'pending' to alert all users and escalates the ticket in the admin priority dashboard.

Table 3. Maintenance Categorisation and Routing Matrix

As detailed in **Table 3**, the maintenance module automatically categorises defect reports into two distinct structural workflows to optimise technician deployment. The system aggregates report submissions using a voting mechanism rather than creating duplicate tickets. This aggregation visually flags the ticket as a high-priority, multi-user disruption in the administrator's queue and simultaneously updates the facility's global status, thereby preventing redundant reports and accelerating resolution.

By automatically updating the global facility status and visually flagging multi-user disruptions, the system protects administrators from ticket fatigue while accelerating technician deployment.

3.4. End-to-End System Functionality Testing

To evaluate the system's capacity to handle the end-to-end boarding house operational lifecycle, functional testing was conducted on the core reservation and maintenance modules. The evaluation utilised Blackbox Testing to verify that the interconnected workflows operated strictly according to the established design parameters without requiring visibility into the internal code structure. The testing scenarios focused on input validation, state transitions, and concurrency handling. A summary of functional test results by module is presented in **Table 4.**

No	Operational Lifecycle Phase	Test Cases	Pass	Fail
1	New Tenant Registration	6	6	0
2	Authentication & Access Control	14	14	0
3	Room Discovery & Booking	5	5	0

4	Rent Payment Processing	6	6	0
5	Damage Reports (Tenant)	5	5	0
6	Service Requests (Tenant)	6	6	0
7	Room, Customer & Facility Management	15	15	0
8	Order & Repair Administration	8	8	0
9	Request Processing & Room Transfer	9	9	0
10	Staff Governance (Owner)	9	9	0
Total		83	83	0

Table 4. Maintenance Categorisation and Routing Matrix

The Blackbox Testing matrix above summarises the execution of 83 functional test cases spanning the entire operational lifecycle, from initial tenant onboarding to comprehensive staff and facility governance. The scenarios rigorously tested input validation, state transitions, and role-based access control.

Achieving zero failures across all 83 test cases empirically validates the system's capacity to seamlessly manage the operational lifecycle. This flawless execution demonstrates that the FDD methodology successfully produced a highly stable and structurally sound relational architecture. Crucially, the perfect execution in Phase 3 (Room Discovery & Booking) proves that the systematic synchronisation of booking algorithms effectively handles real-time concurrency, preventing overlapping reservations. Furthermore, the operational success spanning Phases 5 through 9 validates the system's ability to maintain a continuous digital lifecycle by flawlessly transitioning data from tenant acquisition into proactive physical property maintenance.

While the functional testing confirms that KostHub successfully executes its core operational lifecycle, its current iteration acknowledges specific constraints. The payment validation relies on simulated gateway inputs rather than live banking callbacks, and maintenance reporting remains dependent on manual tenant submissions. Consequently, future developments should prioritise integrating third-party banking APIs for automated financial clearing and adopting IoT-based sensors to transition toward automated facility monitoring, thereby advancing the system's operational maturity.

4. CONCLUSION

This study successfully developed KostHub, a web-based boarding house management system built using the FDD methodology, to resolve the administrative fragmentation inherent in traditional property management. By integrating real-time reservation algorithms with database row-level locking and a standardised maintenance ticketing module, the system successfully eliminates overlapping reservations and optimises repair workflows. This fundamentally shifts property administration from reactive crisis management into proactive, data-driven efficiency.

Blackbox testing across 83 scenarios achieved a 100% pass rate, empirically validating the system's ability to flawlessly execute the end-to-end operational lifecycle without data fragmentation. To further enhance operational maturity, future developments should focus on integrating third-party banking APIs for automated financial clearing and IoT-based smart sensors for automated facility monitoring.

REFERENCES

[1] M. Rivaldi, A. Sutanta, dan R. A. Kumalasanti, "Sistem Manajemen Penyewaan Kamar Kos Berbasis Web," Jurnal Media Akademik (JMA), vol. 3, no. 12, 2021.

[2] M. Dwi Rahmatya, D. Simangunsong, dan M. F. Wicaksono, "e-Kos sebagai Sistem Informasi Pengelolaan Kos pada Mazasi's House," Jurnal Teknologi dan Informasi, vol. 12, no. 2, hlm. 176-190, 2022, doi: 10.34010/jati.v12i2.8027.

[3] D. Yusma, N. Merlina, dan N. Nurajijah, "Sistem Informasi Pencarian Rumah Kost Berbasis Web," INTI Nusa Mandiri, vol. 15, no. 2, hlm. 127–134, 2021, doi: 10.33480/inti.v15i2.1702.

[4] A. H. Arribathi, A. D. A. Putra, dan A. A. Syah, "Perancangan Website Pencarian Kos Berbasis Lokasi Menggunakan Pendekatan Agile," ICIT Journal, vol. 11, no. 2, hlm. 120-243, Agst. 2025..

[5] E. R. Rosliani, C. Fahmidin, dan I. Nurul, "Sistem Informasi Pembayaran Rumah Kost Berbasis Website pada Elin Kost Garut," INTERNAL (Information System Journal), vol. 5, no. 1, hlm. 29–39, 2022, doi: 10.32627/internal.v5i1.529.

[6] Dhea Apriliyanti dan Ariyani Wardhana, "Perancangan Sistem Informasi Penyewaan Rumah Kost Berbasis Web Menggunakan Soft System Methodology (SSM) (Studi Kasus : Dhaykost)," Format: Jurnal Ilmiah Teknik Informatika, vol. 9, no. 2, hlm. 194, 2021, doi: 10.22441/format.2020.v9.i2.010.

[7] "Rancang Bangun Sistem Informasi Monitoring Kerusakan dan Perbaikan Fasilitas Berbasis Web (Studi Kasus: Pondok Pesantren Al Islamu Al Ainul Baahiroh)," J-PTIIK, vol. 9, no. 6, Jul. 2025, Diakses: 22 Mei 2026. [Daring]. Tersedia: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14977

[8] "Pengembangan Sistem Informasi Pelaporan Kerusakan Fasilitas berbasis Web. Studi Kasus: Rusunawa Universitas Brawijaya," J-PTIIK, vol. 8, no. 2, Feb. 2024, Diakses: 22 Mei 2026. [Daring]. Tersedia: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13259

[9] Y. P. Herawati, P. Sokibi, and L. Norhan, "Rancang Bangun Sistem Penjualan Kue Online Berbasis Web dengan Pendekatan Feature Driven Development (FDD) Elluffi Home Bakery," Jurnal Informasi Interaktif, vol. 10, no. 3, pp. 209-216, Sep. 2025.

[10] M. Mustika, N. Rismawati, and Y. I. Syuhardi, "Perancangan Aplikasi Sistem Pembayaran Terapi pada Siswa Inklusi Menggunakan Metode Feature Driven Development (FDD)," Journal of Information System, Applied, Management, Accounting and Research (JISAMAR), vol. 5, no. 4, pp. 868-884, Nov. 2021.

[11] I. H. Rasyad, M. D. Irawan, and Triase, "Implementasi Metode Feature Driven Development pada Sistem Informasi Pelayanan Masyarakat," Jurnal Sistem Informasi dan Teknik Komputer, vol. 10, no. 2, 2025.

PENGEMBANGAN APLIKASI HRIS BERBASIS WEB UNTUK DIGITALISASI SUMBER DAYA MANUSIA PADA PT LINTAS DATA PRIMA

Muhamad Hilal Fakhri [1], Haris Setyawan [2,*], Cahya Damarjati [3]
1) Program Studi Teknologi Informasi Universitas Muhammadiyah Yogyakarta
Corresponding author e-mail: haris.setyawan@umy.ac.id

Abstrak

Pengelolaan sumber daya manusia dan administrasi dokumen pada PT Lintas Data Prima sebelumnya masih dilakukan secara manual, sehingga menimbulkan kendala pada pencatatan data pelamar, pemantauan rekrutmen, pengelolaan data karyawan, serta pengarsipan surat. Penelitian ini bertujuan mengembangkan aplikasi Human Resource Information System (HRIS) berbasis web untuk mendukung digitalisasi proses sumber daya manusia dan administrasi dokumen secara terintegrasi. Metode pengembangan yang digunakan adalah Software Development Life Cycle model Waterfall melalui tahapan analisis, desain, implementasi, pengujian, dan deployment. Sistem dikembangkan menggunakan Golang dengan framework Gin sebagai backend, Next.js sebagai frontend, serta MySQL sebagai basis data. Fitur utama sistem meliputi pengelolaan akun, profil pelamar dan karyawan, rekrutmen, jadwal interview, onboarding, offboarding, pengaduan, pengajuan resign, pengelolaan surat, template surat, log aktivitas, serta screening CV berbasis kecerdasan buatan. Hasil pengujian menunjukkan seluruh fitur berjalan sesuai fungsi berdasarkan Black Box Testing. User Acceptance Testing memperoleh nilai rata-rata 3,93 dengan kategori sangat baik, sedangkan pengujian screening CV berbasis AI pada data CV yang mewakili beberapa kategori kesesuaian kandidat menghasilkan nilai Mean Absolute Error sebesar 5,75. Hasil tersebut menunjukkan bahwa aplikasi HRIS yang dikembangkan layak digunakan untuk mendukung digitalisasi proses sumber daya manusia dan administrasi dokumen pada PT Lintas Data Prima.
Kata kunci: AI screening CV; Digitalisasi SDM; HRIS; Rekrutmen; Waterfall

Abstract

Human resource management and document administration at PT Lintas Data Prima were previously conducted manually, causing problems in applicant recording, recruitment monitoring, employee data management, and letter archiving. This study aims to develop a web-based Human Resource Information System (HRIS) application to support the digitalization of human resource processes and integrated document administration. The system was developed using the Software Development Life Cycle with the Waterfall model through analysis, design, implementation, testing, and deployment stages. The application was built using Golang with the Gin framework as the backend, Next.js as the frontend, and MySQL as the database. The main features include account management, applicant and employee profiles, recruitment, interview scheduling, onboarding, offboarding, complaints, resignation requests, letter management, letter templates, activity logs, and AI-based CV screening. The test results show that all features worked according to their functions based on Black Box Testing. User Acceptance Testing obtained an average score of 3.93, categorized as very good, while the AI-based CV screening test produced a Mean Absolute Error value of 5.75. These results indicate that the developed HRIS application is feasible to support the digitalization of human resource processes and document administration at PT Lintas Data Prima.
Keyword: *AI screening; HRIS; Human Resources Digitalization; Recruitment; Waterfall*

I. PENDAHULUAN

Perkembangan teknologi informasi mendorong organisasi untuk mengelola data operasional secara lebih terstruktur dan terintegrasi. Salah satu bentuk penerapannya adalah Human Resource Information System (HRIS), yaitu sistem informasi yang membantu organisasi mengumpulkan, menyimpan, menganalisis, dan mengomunikasikan informasi sumber daya manusia [1]. Penerapan HRIS dinilai penting karena mampu mempermudah proses administrasi, meningkatkan akses terhadap data, serta mendukung efisiensi pengelolaan karyawan [2].

PT Lintas Data Prima merupakan perusahaan yang bergerak di bidang jasa teknologi informasi dan telekomunikasi. Berdasarkan observasi dan wawancara, proses pengelolaan data pelamar, pemantauan rekrutmen, pengelolaan data

karyawan, serta administrasi surat masuk dan surat keluar masih dilakukan secara manual. Kondisi tersebut dapat menimbulkan risiko data tidak terdokumentasi dengan baik, proses pencarian informasi menjadi lambat, serta pengarsipan dokumen tidak terpusat. Di sisi lain, HRIS dapat membantu proses rekrutmen secara digital, termasuk penyaringan kandidat dan pemantauan tahapan seleksi [3].

Selain aspek sumber daya manusia, organisasi juga membutuhkan pengelolaan surat yang sistematis. Penelitian terdahulu menunjukkan bahwa sistem pengarsipan surat berbasis web dapat mempermudah pencatatan, pencarian, dan pengarsipan dokumen [4], [5]. Oleh karena itu, penelitian ini bertujuan mengembangkan aplikasi HRIS berbasis web yang mengintegrasikan pengelolaan pelamar, karyawan, rekrutmen, screening CV berbasis AI, dan administrasi surat pada PT Lintas Data Prima.

II. TINJAUAN PUSTAKA

Penelitian mengenai HRIS telah menunjukkan bahwa sistem informasi sumber daya manusia dapat mendukung kinerja organisasi melalui pengelolaan data karyawan dan administrasi SDM yang lebih terstruktur [2]. Pada konteks rekrutmen, HRIS dapat membantu organisasi mengelola data pelamar dan proses seleksi secara lebih efisien [3]. Penelitian lain juga mengembangkan HRIS berbasis web dengan metode prototype, tetapi ruang lingkupnya lebih berfokus pada pengelolaan data karyawan dan cuti tanpa mengintegrasikan administrasi dokumen perusahaan [6].

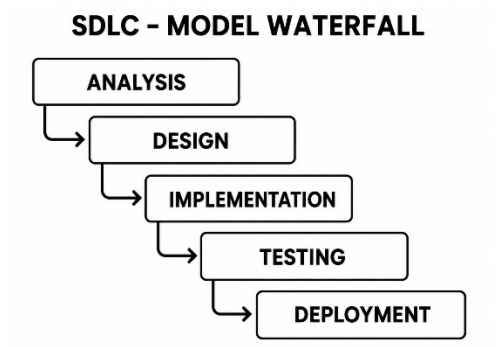
Penelitian terkait surat menyurat juga telah banyak dilakukan. Sistem informasi pengarsipan surat masuk dan surat keluar berbasis web terbukti membantu proses pencatatan dan pencarian arsip [4], sedangkan sistem pengelolaan surat pada institusi pendidikan membantu pengarsipan dokumen secara digital [5]. Namun, penelitian tersebut masih berfokus pada administrasi surat dan belum terintegrasi dengan HRIS. Selain itu, perkembangan kecerdasan buatan memungkinkan proses rekrutmen dibantu melalui resume parsing, penyaringan awal kandidat, dan rekomendasi seleksi, meskipun tetap perlu memperhatikan transparansi dan potensi bias [7].

Berdasarkan kajian tersebut, penelitian ini memiliki kontribusi pada pengembangan sistem HRIS yang menggabungkan pengelolaan data pelamar, data karyawan, rekrutmen, screening CV berbasis AI, pengaduan, resign, onboarding, offboarding, log aktivitas, serta pengelolaan surat dalam satu aplikasi web terintegrasi. Pengembangan dilakukan menggunakan model Waterfall karena tahapan analisis, desain, implementasi, pengujian, dan deployment dapat dijalankan secara sistematis [8].

III. METODOLOGI PENELITIAN

A. Metode Pengembangan Sistem

Metode penelitian yang digunakan adalah *Software Development Life Cycle* (SDLC) model *Waterfall*. Model ini dipilih karena memiliki tahapan pengembangan yang sistematis dan berurutan, mulai dari analisis, desain, implementasi, pengujian, hingga *deployment*. Tahapan pengembangan secara berurutan tersebut memudahkan proses pengembangan aplikasi karena kebutuhan sistem telah didefinisikan sejak awal. Alur tahapan SDLC model *Waterfall* yang digunakan dalam penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Tahapan SDLC Model Waterfall

Berdasarkan Gambar 1, proses pengembangan sistem dimulai dari tahap analisis kebutuhan, kemudian dilanjutkan dengan tahap desain sistem, implementasi, pengujian, dan deployment. Setiap tahapan dilakukan secara berurutan agar sistem yang dikembangkan sesuai dengan kebutuhan pengguna dan dapat diuji sebelum diterapkan.

B. Analisis Kebutuhan

Analisis kebutuhan dilakukan melalui observasi dan wawancara terhadap proses bisnis di PT Lintas Data Prima. Hasil analisis menghasilkan kebutuhan fungsional berupa autentikasi pengguna, pengelolaan profil, rekrutmen, lamaran, akun pengguna, divisi dan lowongan, data karyawan, surat, template surat, pengaduan, pengajuan resign, log aktivitas, serta pengelolaan surat oleh admin divisi. Kebutuhan nonfungsional meliputi aplikasi berbasis web, mekanisme autentikasi pengguna, kemudahan penggunaan, dan penyimpanan data secara terpusat.

C. Perancangan dan Implementasi Sistem

Pada tahap desain dilakukan perancangan arsitektur sistem, use case diagram, activity diagram, relasi antartabel, dan rancangan antarmuka sebagai acuan pengembangan aplikasi. Tahap implementasi dilakukan menggunakan Golang dengan framework Gin sebagai backend, Next.js sebagai frontend, dan MySQL sebagai basis data. Pemilihan teknologi tersebut disesuaikan dengan kebutuhan aplikasi web yang memerlukan layanan backend, antarmuka interaktif, dan penyimpanan data relasional. Golang mendukung pengembangan layanan backend berperforma tinggi [9], sedangkan Next.js mendukung pengembangan aplikasi web modern berbasis React [10].

D. Pengujian dan Deployment

Pengujian dilakukan menggunakan Black Box Testing, User Acceptance Testing (UAT), dan Mean Absolute Error (MAE). Black Box Testing digunakan untuk memeriksa kesesuaian input dan output dari setiap fitur tanpa melihat struktur kode program [11]. UAT digunakan untuk mengukur penerimaan pengguna akhir terhadap sistem [12]. Sementara itu, MAE digunakan untuk mengevaluasi fitur screening CV berbasis AI dengan membandingkan skor AI terhadap skor expert. Rumus MAE ditunjukkan pada persamaan (1).

$$MAE = \frac{\sum|SkorAI - SkorExpert|}{n} \quad (1)$$

Pada persamaan (1), SkorAI merupakan skor yang diberikan oleh AI pada data CV ke-i, SkorExpert merupakan skor yang diberikan oleh expert pada data CV ke-i, dan n merupakan jumlah data CV yang digunakan dalam pengujian. Tahapan terakhir adalah deployment, yaitu penerapan aplikasi pada lingkungan yang dapat diakses pengguna sesuai hak akses masing-masing. Aktor utama dalam sistem terdiri dari Super Admin, Admin Divisi, Staff Divisi, dan Pelamar.

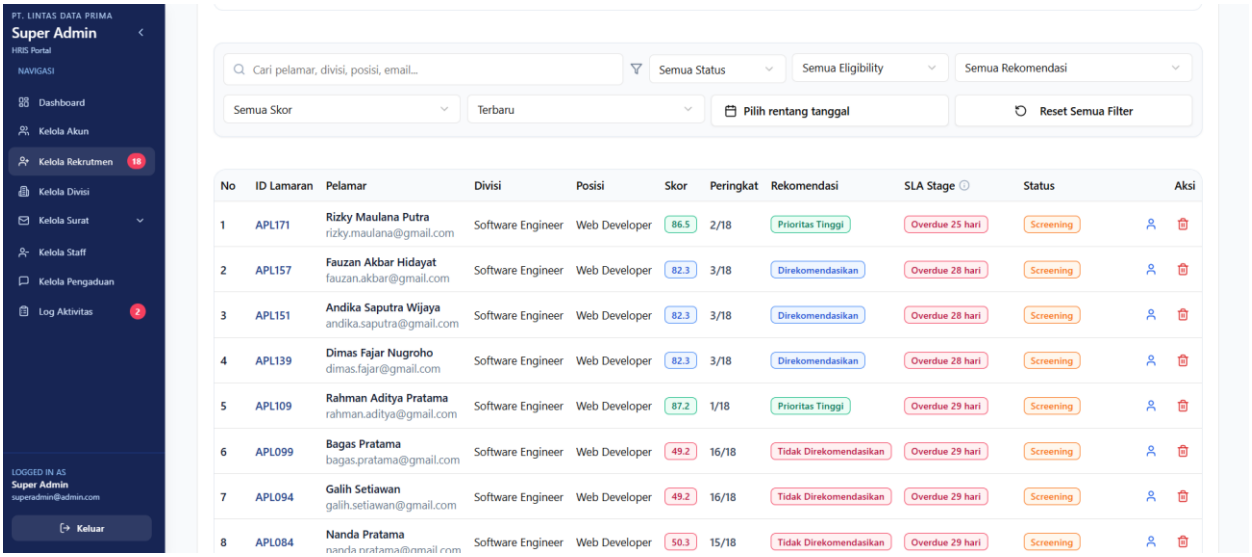
IV. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Hasil penelitian berupa aplikasi HRIS berbasis web yang menyediakan fitur utama untuk empat peran pengguna. Super Admin dapat mengelola akun, rekrutmen, data pelamar, jadwal interview, onboarding, offboarding, divisi, lowongan, surat, template surat, pengaduan, dan log aktivitas. Admin Divisi dapat mengelola surat dalam lingkup divisi, Staff Divisi dapat mengelola profil, membuat pengaduan, dan mengajukan resign, sedangkan Pelamar dapat membuat akun, melengkapi profil, mengirim lamaran, melihat status lamaran, serta melihat jadwal interview.

Gambar 2 menunjukkan tampilan halaman Kelola Rekrutmen pada akun Super Admin. Halaman ini digunakan untuk memantau data pelamar, melakukan filter kandidat berdasarkan divisi, posisi, status, eligibility, rekomendasi, dan skor, serta melihat informasi seperti skor kandidat, peringkat, status rekrutmen, dan SLA stage. Dengan adanya fitur ini, proses rekrutmen dapat dikelola secara lebih terstruktur, terpusat, dan mudah dipantau oleh Super Admin.

Sistem juga menyediakan fitur screening CV berbasis AI sebagai rekomendasi awal dalam proses seleksi kandidat. Fitur ini tidak digunakan sebagai keputusan akhir, tetapi membantu Super Admin menilai kesesuaian kandidat terhadap kriteria lowongan. Dengan adanya integrasi fitur tersebut, proses rekrutmen, pengelolaan karyawan, dan administrasi surat dapat dilakukan dalam satu sistem terpusat. Implementasi sistem ini menunjukkan bahwa aplikasi HRIS yang dikembangkan tidak hanya berfungsi sebagai sistem pencatatan data, tetapi juga mendukung proses pengambilan keputusan awal dalam rekrutmen melalui bantuan AI.



Gambar 2. Halaman Kelola Rekrutmen Super Admin

B. Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa fitur yang dikembangkan berjalan sesuai kebutuhan dan dapat diterima oleh pengguna. Ringkasan hasil pengujian sistem ditunjukkan pada Tabel 1.

Tabel 1. Ringkasan hasil pengujian sistem

Jenis Pengujian	Objek	Hasil Utama	Keterangan
Black Box Testing	Fitur utama aplikasi	Seluruh test case berstatus pass	Fungsi sistem berjalan sesuai kebutuhan
User Acceptance Testing	24 responden	Rata-rata 3,93 dari skala 4	Kategori sangat baik
Pengujian AI	Data CV representatif berdasarkan kategori kesesuaian kandidat	MAE sebesar 5,75	Skor AI mendekati penilaian expert

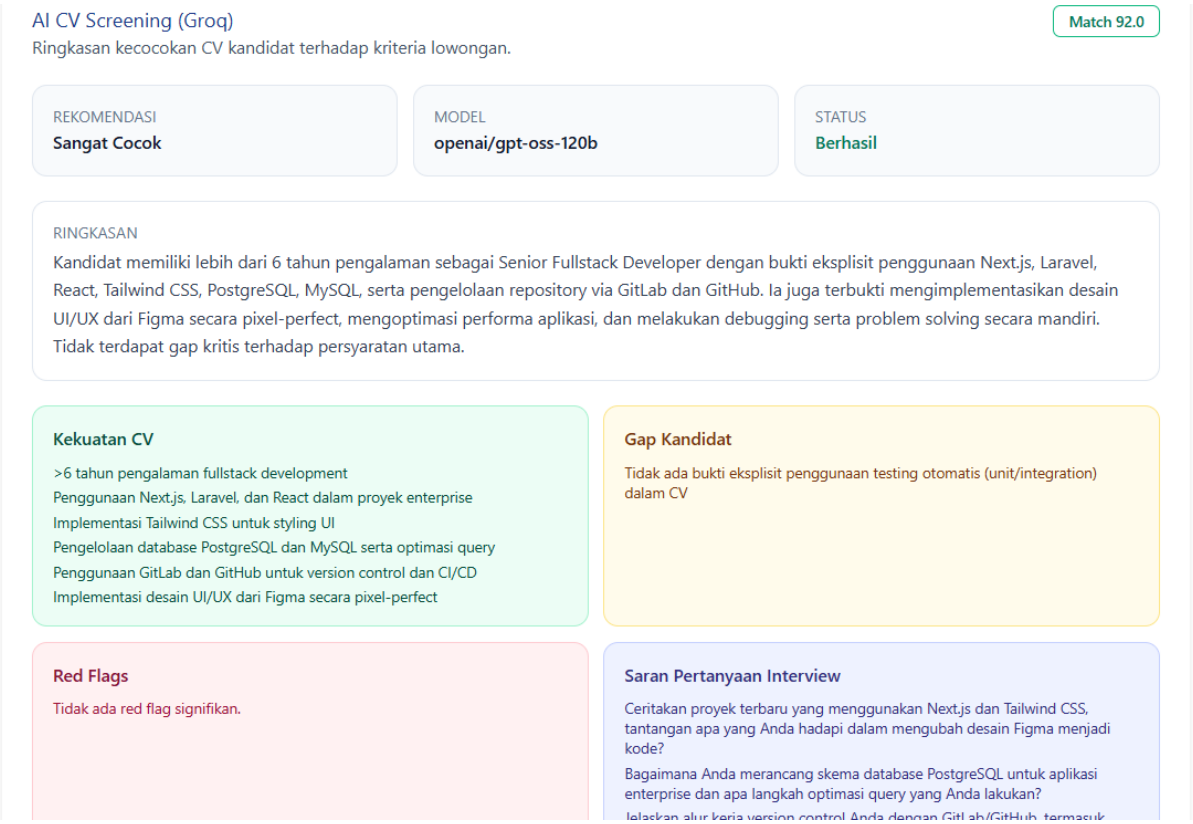
Berdasarkan Tabel 1, pengujian fungsional menunjukkan bahwa seluruh fitur utama pada aplikasi HRIS telah berjalan sesuai dengan kebutuhan pengguna. Pengujian UAT melibatkan 24 responden yang mewakili empat peran pengguna, yaitu Super Admin, Admin Divisi, Staff Divisi, dan Pelamar. Nilai rata-rata UAT sebesar 3,93 dari skala 4 menunjukkan bahwa sistem mudah digunakan, sesuai dengan kebutuhan proses bisnis, dan dapat diterima dengan sangat baik oleh pengguna. Sementara itu, pengujian fitur screening CV berbasis AI dibahas lebih lanjut pada bagian berikutnya.

C. Pengujian Screening CV Berbasis AI

Fitur screening CV berbasis AI digunakan untuk membantu Super Admin dalam menilai kesesuaian kandidat terhadap kriteria lowongan. Fitur ini menampilkan skor kecocokan kandidat, rekomendasi, ringkasan hasil analisis, kekuatan CV, gap kandidat, serta saran pertanyaan interview. Hasil analisis tersebut dapat digunakan sebagai bahan pertimbangan awal dalam proses seleksi kandidat, namun tidak dijadikan sebagai dasar keputusan akhir penerimaan karyawan.

Gambar 3 menunjukkan tampilan hasil screening CV berbasis AI pada sistem HRIS. Pada halaman tersebut, sistem menampilkan informasi kandidat, nilai match atau kecocokan, rekomendasi kandidat, model AI yang digunakan, status proses screening, ringkasan analisis, kekuatan CV, gap kandidat, red flags, serta saran pertanyaan interview. Tampilan ini membantu Super Admin dalam memahami hasil penilaian kandidat secara lebih cepat dan terstruktur.

Tabel 2 menyajikan contoh hasil perbandingan skor AI dan expert pada data CV yang mewakili beberapa kategori kesesuaian kandidat. Kategori tersebut terdiri dari sangat cocok, cocok, tidak cocok, dan sangat tidak cocok. Pemilihan data CV berdasarkan kategori ini bertujuan untuk melihat kemampuan sistem AI dalam memberikan penilaian pada variasi tingkat kesesuaian kandidat terhadap kriteria lowongan.



Gambar 3. Tampilan Hasil Screening CV Berbasis AI

Tabel 2. Perbandingan Skor AI dan Expert Berdasarkan Kategori Kesesuaian CV

CV	Kategori Expert	Skor AI	Skor Expert	Selisih
CV1	Sangat cocok	90	87	3
CV2	Cocok	68	75	7
CV3	Tidak cocok	30	35	5
CV4	Sangat tidak cocok	12	20	8

Berdasarkan Tabel 2, hasil perbandingan skor AI dan expert menunjukkan bahwa penilaian AI memiliki kecenderungan yang sejalan dengan kategori kesesuaian kandidat yang diberikan oleh expert. Data CV yang digunakan pada pengujian ini dipilih sebagai representasi dari beberapa kategori penilaian, yaitu sangat cocok, cocok, tidak cocok, dan sangat tidak cocok. Perbedaan skor antara AI dan expert terjadi karena AI cenderung menilai berdasarkan informasi yang tertulis secara eksplisit pada CV, seperti keterampilan, pengalaman, dan teknologi yang dicantumkan, sedangkan expert juga mempertimbangkan kedalaman pengalaman, relevansi proyek, serta keterkaitan peran sebelumnya dengan kebutuhan posisi. Nilai MAE sebesar 5,75 menunjukkan bahwa rata-rata selisih skor AI dan expert relatif kecil, sehingga fitur screening CV berbasis AI dapat digunakan sebagai alat bantu rekomendasi awal dalam proses seleksi kandidat, bukan sebagai dasar keputusan akhir penerimaan.

V. KESIMPULAN DAN SARAN

Aplikasi HRIS berbasis web untuk PT Lintas Data Prima berhasil dikembangkan sesuai dengan kebutuhan pengguna. Sistem mampu mengintegrasikan pengelolaan pelamar, rekrutmen, data karyawan, pengaduan, pengajuan resign, onboarding, offboarding, log aktivitas, serta administrasi surat dalam satu platform terpusat. Hasil Black Box Testing menunjukkan bahwa seluruh fitur utama berjalan sesuai

fungsinya, sedangkan pengujian User Acceptance Testing memperoleh nilai rata-rata 3,93 dari skala 4 dengan kategori sangat baik, yang menunjukkan bahwa sistem mudah digunakan, sesuai kebutuhan, dan dapat diterima oleh pengguna. Selain itu, pengujian fitur screening CV berbasis AI menghasilkan nilai Mean Absolute Error sebesar 5,75 pada data CV yang mewakili beberapa kategori kesesuaian kandidat, sehingga fitur tersebut dapat digunakan sebagai alat bantu rekomendasi awal dalam proses seleksi kandidat, bukan sebagai dasar keputusan akhir penerimaan. Dengan demikian, sistem dinyatakan layak digunakan untuk mendukung digitalisasi proses sumber daya manusia dan administrasi dokumen pada PT Lintas Data Prima. Pengembangan selanjutnya dapat diarahkan pada integrasi sistem presensi, payroll, dan pengelolaan cuti agar cakupan HRIS menjadi lebih lengkap, serta peningkatan keamanan, penyempurnaan antarmuka, dan pengujian screening CV dengan data yang lebih banyak dan bervariasi agar hasil evaluasi menjadi lebih representatif.

DAFTAR PUSTAKA

[1] G. M. A. A. Quaosar dan S. Rahman, “Human Resource Information Systems (HRIS) of Developing Countries in 21st Century: Review and Prospects,” 2021, hlm. 470-483.

[2] S. A. Saidah, D. Jhoansyah, dan R. Nurmala, “The Influence of the Human Resource Information System (HRIS), Organizational Culture, Financial Incentives, and Flexible Working Space (FWS) On Employee Performance,” 2024.

[3] E. N. Farida, M. R. Fauzi, dan S. Shaddiq, “The Role of Human Resource Information System (HRIS) in Improving the Efficiency of the Recruitment Process in the Revolutionary Era 4.0,” 2025, hlm. 852-859.

[4] R. Idrus, A. Q. Hatta, “Sistem Informasi Pengarsipan Surat Masuk dan Surat Keluar Berbasis Web pada Fakultas Ekonomi Universitas Sulawesi Barat,” Jurnal Ilmiah Ilmu Komputer, vol. 9, no. 1, hlm. 51-62, 2023.

[5] I. G. N. A. Pawana, M. W. Jayantari, dan M. D. Nugraha, “Rancang Bangun Sistem Informasi Pengelolaan Surat Berbasis Web pada Fakultas Vokasi Universitas Warmadewa,” Jurnal Ilmiah TELSINAS, vol. 7, no. 2, hlm. 128-144, 2024.

[6] N. Marbun, S. Ma, dan T. Solikhah, “Pengembangan Aplikasi Human Resource Information System (HRIS) Berbasis Web dengan Metode Prototype,” Journal of Research and Publication Innovation, vol. 3, no. 1, hlm. 2524-2526, 2025.

[7] S. Murodilla, U. Dadaboyev, J. Abdullayeva, N. Abbosova, dan A. Suleymenova, “Role of artificial intelligence in employee recruitment: systematic review and future research directions,” Discover Global Society, 2025.

[8] Y. S. Rahayu, Y. Saputra, dan D. Irawan, “Implementasi Metode Waterfall pada Pengembangan Sistem Informasi Mobile E-Disarpus,” Zonasi: Jurnal Sistem Informasi, vol. 6, no. 2, hlm. 523-534, 2024.

[9] D. E. Costa, S. Mujahid, R. Abdalkareem, dan E. Shihab, “Breaking Type Safety in Go: An Empirical Study on the Usage of the unsafe Package,” IEEE, hlm. 1-17, 2020.

[10] R. Hanafi, A. Haq, dan N. Agustin, “Comparison of Web Page Rendering Methods Based on Next.js Framework Using Page Loading Time Test,” vol. 13, no. 1, hlm. 102-108, 2024.

[11] S. D. Pratama, L. Lasimin, dan M. N. Dadaprawira, “Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence dan Boundary Value,” Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD, vol. 6, hlm. 560-569, 2023.

[12] I. Afrianto, A. Heryandi, A. Finadhita, dan S. Atin, “User Acceptance Test For Digital Signature Application In Academic Domain To Support The Covid-19 Work From Home Program,” IJISTECH, vol. 5, no. 36, hlm. 270-280, 2021.

SENIFORMA 2026
Seminar Nasional Informatika Universitas Mataram

ANALISIS OPTIMALISASI ARSITEKTUR MICROSERVICES BERBASIS DOMAIN-DRIVEN DESIGN UNTUK SCALABILITY DAN CONCURRENCY PADA FLASH SALE E-COMMERCE

Istiqomah Virginia[1], Qomari Auliyah[2], Inaz Putri Kamila[3], Annisa Makarima Ahlak[4]
1,2,3,4) Program Studi Teknik Informatika, Universitas Mataram
Corresponding author e-mail: yourizzty13@gmail.com, qomaaaulii@gmail.com, nzkmila@gmail.com, anisamakarimaahlak@gmail.com,

Abstrak

Periode flash sale pada platform e-commerce menimbulkan tantangan arsitektural karena trafik pengguna dan transaksi meningkat secara drastis dalam waktu singkat. Kondisi ini dapat menyebabkan bottleneck pada layanan kritis, terutama layanan inventaris, pesanan, dan pembayaran, yang berpengaruh langsung terhadap scalability dan concurrency sistem. Penelitian ini bertujuan untuk menganalisis optimalisasi arsitektur microservices berbasis Domain-Driven Design dalam mendukung pemetaan layanan, kepemilikan data, dan batas tanggung jawab proses pada transaksi flash sale. Metode yang digunakan adalah Systematic Literature Review dengan mengkaji penelitian terdahulu terkait microservices, Domain-Driven Design, flash sale, bottleneck, scalability, dan concurrency. Hasil kajian menunjukkan bahwa Domain-Driven Design membantu memetakan sistem e-commerce ke dalam domain bisnis yang lebih jelas, seperti produk, keranjang, inventaris, pesanan, dan pembayaran. Pemetaan tersebut memungkinkan layanan kritis diidentifikasi dan diskalakan secara lebih terarah. Selain itu, kejelasan batas layanan membantu mengurangi tumpang tindih proses dan risiko konflik data saat banyak transaksi berlangsung secara bersamaan. Penelitian ini menyimpulkan bahwa arsitektur microservices berbasis Domain-Driven Design sesuai digunakan sebagai dasar konseptual untuk mengurangi risiko bottleneck pada flash sale e-commerce, meskipun tetap membutuhkan mekanisme teknis pendukung seperti load balancing, caching, antrean, dan komunikasi tidak langsung.
Kata kunci: Concurrency; Domain-Driven Design; Flash Sale; Microservices; Scalability

Abstract

Flash sale periods in e-commerce platforms create significant architectural challenges because traffic and transactions increase sharply within a short time. This condition may cause bottlenecks in critical services, especially inventory, order, and payment services, which directly affect scalability and concurrency. This study aims to analyze the optimization of microservices architecture based on Domain-Driven Design in supporting service decomposition, data ownership, and responsibility boundaries during flash sale transactions. The method used in this study is a Systematic Literature Review by examining relevant studies on microservices, Domain-Driven Design, flash sale, bottleneck, scalability, and concurrency. The results show that Domain-Driven Design helps map e-commerce systems into business domains, clarify service boundaries, identify critical services, and support more directed scalability and concurrency management. In the flash sale context, Inventory Service, Order Service, and Payment Service require higher attention because they are directly related to stock consistency, order creation, and payment status synchronization. This study concludes that Domain-Driven Design-based microservices architecture is conceptually suitable for reducing bottleneck risks, although technical mechanisms such as load balancing, caching, queue management, and asynchronous communication are still required to support implementation.
Keyword: *Concurrency; Domain-Driven Design; Flash Sale; Microservices; Scalability*

I. PENDAHULUAN

Perkembangan teknologi informasi mendorong pertumbuhan pesat sektor *e-commerce*. Di Indonesia, nilai transaksi *e-commerce* pada tahun 2025 mencapai US$71 miliar atau meningkat 14% dibandingkan tahun sebelumnya [1]. Pertumbuhan ini menuntut sistem yang stabil, responsif, dan mampu melayani banyak pengguna secara bersamaan.

Salah satu tantangan utama dalam operasional *e-commerce* adalah periode *flash sale*, ketika jumlah pengguna dan transaksi meningkat drastis dalam waktu singkat. Studi pada *platform* berskala besar menunjukkan bahwa lonjakan trafik (*massive flash crowd*) dapat menurunkan performa infrastruktur dan meningkatkan *latency* transaksi apabila kapasitas sistem tidak mampu mengakomodasi beban yang meningkat [2], [3].

Kondisi tersebut berpotensi menimbulkan *bottleneck* pada layanan kritis, terutama inventaris, pesanan, dan pembayaran. Pada arsitektur *monolithic*, risiko *bottleneck* lebih besar karena seluruh fungsi sistem berada dalam satu aplikasi sehingga beban tinggi pada satu komponen dapat memengaruhi kinerja sistem secara keseluruhan [4].

Arsitektur *microservices* menjadi alternatif karena memungkinkan layanan dikembangkan dan diskalakan secara independen [5]. Namun, pemisahan layanan perlu diselaraskan dengan domain bisnis agar tidak menimbulkan ketergantungan dan *bottleneck* baru. *Domain-Driven Design* (DDD) digunakan untuk memetakan tanggung jawab layanan, seperti produk, keranjang, inventaris, pesanan, dan pembayaran [6], [7]. Berdasarkan hal tersebut, penelitian ini menganalisis optimalisasi arsitektur *microservices* berbasis DDD untuk mendukung *scalability* dan *concurrency* pada *flash sale e-commerce*.

II. TINJAUAN PUSTAKA

Transformasi arsitektur sistem *e-commerc*e dari model *monolithic* menuju *microservices* menjadi salah satu pendekatan untuk meningkatkan fleksibilitas dan kemampuan sistem dalam menangani pertumbuhan pengguna serta transaksi. Pada arsitektur *monolithic*, berbagai fungsi seperti katalog produk, keranjang, inventaris, pesanan, dan pembayaran berada dalam satu aplikasi sehingga peningkatan beban pada satu komponen dapat mempengaruhi keseluruhan sistem dan memicu *bottleneck* [1],[4]. Kondisi tersebut semakin terlihat pada periode *flash sale* ketika terjadi lonjakan trafik dan transaksi dalam waktu singkat yang dapat menurunkan performa sistem [2] [3].

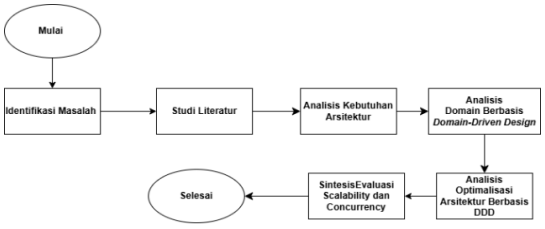
Arsitektur *microservices* memungkinkan sistem dipecah menjadi layanan-layanan independen yang dapat dikembangkan dan diskalakan sesuai kebutuhan [5]. Dalam konteks *e-commerce*, pendekatan ini memungkinkan layanan dengan beban tinggi, seperti inventaris, pesanan, dan pembayaran, memperoleh kapasitas tambahan tanpa harus meningkatkan seluruh sistem. Namun, pemisahan layanan yang hanya berorientasi teknis berisiko menimbulkan *coupling* tinggi dan *bottleneck* baru apabila tidak mempertimbangkan kebutuhan bisnis [7].

Domain-Driven Design (DDD) digunakan untuk menyelaraskan struktur layanan dengan domain bisnis melalui pemetaan fungsi seperti produk, keranjang, invetaris, pesanan, dan pembayaran [6]. Pendekatan ini membantu memperjelas batas tanggung jawab layanan, meningkatkan *cohesion*, dan mengurangi *coupling* antar layanan [8] [9]. Selain itu, DDD mendukung identifikasi layanan kritis yang memerlukan prioritas dalam pengelolaan beban dan pengembangan sistem.

Dalam konteks *flash sale*, DDD berkontribusi terhadap *scalability* dan *concurrency* melalui pemetaan domain serta kejelasan kepemilikan data dan proses bisnis. Layanan inventaris, pesanan, dan pembayaran menjadi domain yang paling kritis karena berhubungan langsung dengan konsistensi stok, pembentukan pesanan, dan sinkronisasi pembayaran. Dengan memanfaatkan batas domain dan *aggregate*, perubahan data serta logika bisnis dapat dikelola secara lebih terstruktur [10]. Oleh karena itu, DDD dipandang sebagai dasar konseptual yang mendukung optimalisasi arsitektur *microservices* untuk mengurangi risiko *bottleneck* serta meningkatkan *scalability* dan *concurrency* pada sistem *e-commerce* saat *flash sal*e.

III. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode *Systematic Literature Review* (SLR) dengan pendekatan analisis konseptual untuk mengkaji penerapan *Domain-Driven Design* (DDD) pada arsitektur *microservices* dalam sistem *flash sale e-commerce*. Tahapan penelitian dilakukan sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

3.1 Identifikasi Masalah

Pada tahap ini, penelitian mengidentifikasi permasalahan utama yang muncul pada sistem *e-commerce* saat periode *flash sale*, seperti lonjakan trafik pengguna, peningkatan transaksi secara bersamaan, potensi konflik stok, antrean pesanan, dan status pembayaran yang tidak sinkron. Permasalahan tersebut kemudian dikaitkan dengan kebutuhan *scalability* dan *concurrency*, karena sistem harus mampu menyesuaikan kapasitas layanan sekaligus memproses banyak transaksi dalam waktu bersamaan.

3.2 Studi Literatur

Pada tahap kedua, literatur yang dikaji meliputi penelitian tentang *flash sale*, *flash crowd*, arsitektur *microservices*, DDD, *scalability*, *concurrency*, dan *bottleneck*. Literatur dipilih berdasarkan keterkaitannya dengan lonjakan trafik, transaksi bersamaan, pemisahan layanan, pemetaan domain bisnis, dan pengelolaan layanan kritis pada sistem *e-commerce*. Hasil studi literatur digunakan sebagai dasar untuk memahami hubungan antara arsitektur sistem dan risiko performa pada periode *flash sale*.

3.3 Analisis Kebutuhan Arsitektur

Tahap ini menganalisis kebutuhan arsitektur yang diperlukan agar sistem *e-commerce* mampu menghadapi kondisi *flash sale*. Aspek yang dianalisis meliputi penskalaan layanan, distribusi beban kerja, pemisahan tanggung jawab layanan, pengurangan ketergantungan antar-layanan, dan kemampuan menangani transaksi secara bersamaan.

3.4 Analisis *Domain-Driven Design*

Tahap ini memetakan sistem *e-commerce* ke dalam domain utama, yaitu produk, keranjang, inventaris, pesanan, dan pembayaran. Setiap domain dianalisis berdasarkan tanggung jawab, karakteristik beban, dan keterkaitannya dengan proses transaksi saat *flash sale*. Pemetaan ini digunakan untuk menentukan batas layanan dan mengidentifikasi layanan yang berpotensi menjadi titik *bottleneck*.

3.5 Analisis Optimalisasi Arsitektur Berbasis DDD

Tahap ini difokuskan pada layanan yang berada pada jalur utama transaksi, yaitu *Inventory Service*, *Order Service*, dan *Payment Service*. Analisis dilakukan untuk menilai bagaimana pemisahan tanggung jawab layanan berbasis DDD dapat mendukung penskalaan mandiri, memperjelas kepemilikan data, dan mengurangi tumpang tindih proses saat transaksi berlangsung secara bersamaan.

3.6 Sintesis dan Evaluasi Konseptual

Tahap terakhir adalah menyintesis hasil kajian literatur dan analisis arsitektur untuk menilai peran DDD dalam mendukung *scalability*, *concurrency*, dan pengurangan risiko *bottleneck* pada *flash sale e-commerce*. Evaluasi dilakukan dengan melihat keterkaitan antara pemetaan domain, layanan kritis, batas tanggung jawab layanan, dan kebutuhan mekanisme pendukung seperti *load balancing*, *caching*, antrean, serta komunikasi tidak langsung.

IV. HASIL DAN PEMBAHASAN

4.1 Hasil Sintesis Literatur

Berdasarkan kajian literatur, *flash sale* pada sistem *e-commerce* menyebabkan lonjakan trafik dan transaksi yang berpotensi menimbulkan *bottleneck* pada layanan utama [2], [3]. *Bottleneck* berkaitan dengan *scalability* dan *concurrency* karena peningkatan pengguna dan transaksi dapat menyebabkan keterbatasan kapasitas layanan, antrean proses, konflik data, dan kegagalan transaksi [10], [11]. Dalam arsitektur *microservices*, *Domain-Driven Design* (DDD) membantu memetakan domain dan memperjelas batas tanggung jawab layanan sehingga layanan kritis, seperti *Inventory Service*, *Order Service*, dan *Payment Service*, dapat diprioritaskan untuk

penskalaan [2], [12]. Selain itu, komunikasi tidak langsung antar-layanan dapat mengurangi ketergantungan sistem dan meningkatkan kemampuan menghadapi lonjakan beban [13]. Oleh karena itu, pembahasan difokuskan pada layanan kritis serta peran DDD dalam mendukung *scalability* dan *concurrency* pada *flash sale e-commerce*.

4.2 Pemetaan Layanan Kritis pada *Flash Sale E-Commerce*

Pada periode *flash sale*, *Inventory Service*, *Order Service*, dan *Payment Service* menjadi layanan paling kritis karena berhubungan langsung dengan stok, pembuatan pesanan, dan validasi pembayaran. Melalui pendekatan DDD, layanan dipetakan berdasarkan domain bisnis sehingga batas tanggung jawab setiap layanan menjadi lebih jelas [9], [10].

Tabel 1. Pemetaan Layanan Kritis pada *Flash Sale E*-Commerce

Layanan	Risiko Utama saat *Flash Sale*	Fokus Optimalisasi
Product Service	Trafik akses produk meningkat.	Pengurangan beban akses baca.
Cart Service	Aktivitas keranjang meningkat.	Pemisahan proses keranjang.
Inventory Service	Konflik stok pada produk yang sama.	Kejelasan pengelolaan stok.
Order Service	Antrean pembuatan pesanan.	Penskalaan proses pesanan.
Payment Service	Status pembayaran tidak sinkron.	Sinkronisasi status transaksi.

Berdasarkan Tabel 1, *Inventory Service*, *Order Service*, dan *Payment Service* menjadi prioritas utama karena berada pada jalur transaksi dan memiliki risiko *bottleneck* tertinggi saat *flash sale*. Oleh karena itu, optimalisasi difokuskan pada ketiga layanan tersebut untuk mendukung keberhasilan transaksi.

4.3 Peran DDD dalam Mengurangi Risiko *Bottleneck*

DDD mengurangi risiko *bottleneck* dengan memperjelas batas tanggung jawab layanan dan memisahkan fungsi bisnis ke dalam layanan yang lebih spesifik. Pendekatan ini membantu mendistribusikan beban kerja, meningkatkan *cohesion*, dan mengurangi *coupling* antar-layanan[9].

Selain itu, batas domain dan *aggregate* memudahkan identifikasi layanan kritis serta pengelolaan perubahan data dan proses bisnis secara lebih terstruktur [10]. Dengan demikian, DDD menjadi dasar konseptual dalam mendukung *scalability* dan *concurrency* pada *flash sale e-commerce*.

4.4 Evaluasi terhadap *Scalability*

Dari sisi *scalability*, arsitektur *microservices* berbasis DDD memungkinkan layanan dipisahkan berdasarkan tanggung jawab bisnis dan diskalakan sesuai kebutuhan beban. Berbeda dengan arsitektur *monolithic* yang meningkatkan kapasitas seluruh aplikasi, *microservices* memungkinkan layanan dengan beban tinggi ditingkatkan secara mandiri [5]. Dalam konteks *flash sale*, DDD memabntu mengidentifikasi layanan prioritas, terutama *Inventory Service*, *Order Service*, dan *Payment Service*, karena ketiganya berada pada jalur utama transaksi [8], [9].

Pemetaan berbasis DDD membuat strategi penskalaan lebih terarah dengan memfokuskan kapasitas tambahan pada layanan yang berisiko mengalami *bottleneck*. Namun, *microservices* tetap dapat mengalamin ketidakseimbangan beban apabila banyak layanan bergantung pada satu titik yang sama [12]. Oleh karenanya, DDD berperan sebagai dasar pemetaan layanan, sedangkan mekanisme seperti *load balancing*, *caching*, antrean, dan komunikasi tidak langsung diperlukan untuk menjaga kestabilan sistem saat *flash sale* [13], [14].

4.5 Evaluasi terhadap *Concurrency*

Dari sisi *concurrency*, transaksi yang terjadi secara bersamaan saat flash sale berpotensi menyebabkan konflik stok, antrean pesanan, dan ketidaksinkronan pembayaran. Konflik proses yang tidak dikelola dengan baik dapat

menurunkan performa sistem *multi-microservice* [11]. Dalam konteks DDD, batas domain dan *aggregate* membantu mengelola perubahan data serta logika bisnis secara lebih terstruktur [10].

Peran DDD dalam mendukung pengelolaan *concurrency* yaitu sebagai berikut.

1. DDD memperjelas kewenangan layanan terhadap data stok, pesanan, dan pembayaran.
2. DDD mengurangi tumpang tindih proses melalui batas tanggung jawab layanan yang spesifik.
3. DDD membantu mengidentifikasi titik rawan konflik pada layanan inventaris, pesanan, dan pembayaran.

Melalui pemisahan tersebut, risiko *concurrency* tidak sepenuhnya hilang, tetapi menjadi lebih mudah dikendalikan. Sistem dapat lebih jelas menentukan layanan mana yang harus menjaga konsistensi data, layanan mana yang perlu diprioritaskan saat transaksi meningkat, dan bagian mana yang berpotensi menjadi titik *bottleneck* pada periode *flash sale*.

4.6 Sintesis Rekomendasi Arsitektur Berbasis DDD

Berdasarkan hasil pembahasan, optimalisasi arsitektur *microservices* pada *flash sale e-commerce* perlu diarahkan pada pemisahan layanan berdasarkan domain bisnis dan karakteristik beban kerja. Layanan yang berada pada jalur transaksi utama, yaitu *Inventory Service*, *Order Service*, dan *Payment Service*, memiliki risiko *bottleneck* lebih tinggi karena berkaitan langsung dengan stok, pembuatan pesanan, dan validasi pembayaran. Penyelarasan layanan dengan domain bisnis melalui DDD dapat meningkatkan *cohesion* dan menurunkan *coupling* antar-layanan, sehingga layanan lebih mudah diprioritaskan, diskalakan, dan dikendalikan saat terjadi lonjakan transaksi [9].

Tabel 2. Sintesis Rekomendasi Arsitektur Berbasis DDD

Aspek	Peran DDD
Pemetaan domain	Memisahkan layanan berdasarkan domain bisnis.
Layanan kritis	Memprioritaskan *Inventory Service*, *Order Service*, dan *Payment Service*.
Scalability	Mendukung penskalaan mandiri layanan beban tinggi.
Concurrency	Memperjelas kepemilikan data dan tanggung jawab proses.
Pengendalian *Bottleneck*	Didukung *load balancing*, *caching*, antrean, dan komunikasi tidak langsung.

Rekomendasi pada Tabel 2 menunjukkan bahwa DDD berfungsi sebagai dasar pembentukan batas layanan, bukan satu-satunya solusi teknis. *Load balancing*, *caching*, antrean, dan komunikasi tidak langsung tetap diperlukan untuk mengatasi ketidakseimbangan beban dan konflik proses [10], [13]. Arsitektur *microservices* berbasis DDD menjadi lebih optimal jika didukung mekanisme yang sesuai dengan kebutuhan layanan kritis.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil kajian, *flash sale* pada sistem *e-commerce* menimbulkan risiko *bottleneck* pada layanan kritis, terutama *Inventory Service*, *Order Service*, dan *Payment Service*. Kondisi ini berkaitan dengan *scalability* dan *concurrency* karena sistem harus mampu menyesuaikan kapasitas layanan dan menangani transaksi secara bersamaan. Arsitektur *microservices* berbasis *Domain-Driven Design* (DDD) dinilai optimal secara konseptual karena memperjelas batas layanan berdasarkan domain bisnis dan membantu mengidentifikasi layanan kritis yang perlu diprioritaskan saat terjadi lonjakan beban.

Dari sisi *scalability*, layanan dapat diskalakan secara mandiri, sedangkan dari sisi *concurrency*, transaksi bersamaan dapat dikelola lebih baik melalui kepemilikan data dan proses yang jelas. Namun, DDD tetap memerlukan dukungan mekanisme teknis seperti *load balancing*, *caching*, antrean, komunikasi tidak langsung, dan pengendalian kapasitas layanan. Penelitian selanjutnya disarankan menguji arsitektur *microservices* berbasis DDD pada skenario *flash sale* menggunakan indikator *latency*, *throughput*, keberhasilan transaksi, konsistensi stok, dan kemampuan menangani transaksi bersamaan.

VI. UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah atas arahan yang diberikan selama penyusunan artikel ini. Ucapan terima kasih juga disampaikan kepada rekan-rekan yang telah memberikan dukungan dan saran dalam proses penyusunan penelitian. Penelitian ini tidak menerima pendanaan khusus dari pihak mana pun.

DAFTAR PUSTAKA

[1] Pusat Kebijakan Perdagangan Domestik Badan Kebijakan Perdagangan Kementerian Perdagangan, *Kinerja Perdagangan Melalui Sistem Elektronik (PMSE) Indonesia Tahun 2025*. 2025.

[2] J. Ge, Y. Tian, L. Liu, R. Lan, and X. Zhang, "Understanding E-Commerce Systems under Massive Flash Crowd: Measurement, Analysis, and Implications," *IEEE Trans. Serv. Comput.*, vol. 13, no. 6, pp. 1180–1193, 2020, doi: 10.1109/TSC.2017.2767600.

[3] G. Huang *et al.*, "X-engine: An optimized storage engine for large-scale e-commerce transaction processing," *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 651–665, 2019, doi: 10.1145/3299869.3314041.

[4] Anusha Reddy Guntakandla, "Microservices Architecture: Decomposing E-Commerce Monoliths into Scalable, Independent Services," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 58s, pp. 642–650, 2025, doi: 10.52783/jisem.v10i58s.12665.

[5] I. A. Jaiswal and E. A. Jain, "Architecting Scalable Microservices for High-Traffic E-commerce Platforms," *Int. J. Res. Publ. Semin.*, vol. 16, no. 2, pp. 103–109, 2025, doi: doi.org/10.36676/jrps.v16.i1.1655.

[6] O. Özkan, Ö. Babur, and M. van den Brand, "Domain-Driven Design in software development: A systematic literature review on implementation, challenges, and effectiveness," *J. Syst. Softw.*, vol. 230, no. June, p. 112537, 2025, doi: 10.1016/j.jss.2025.112537.

[7] J. Fattah-weil, M. Schmeißner, and N. Gronau, "AIS Electronic Library (AISeL) Bridging Domain Models And Microservices : A Domain-Specific Language For Domain-Aligned System Design Bridging Domain Models And Microservices : A Domain-Specific Language For Domain-Aligned," no. June, 2026.

[8] J. A. Suthendra and M. A. I. Pakereng, "Implementation of Microservices Architecture on E-Commerce Web Service," *ComTech Comput. Math. Eng. Appl.*, vol. 11, no. 2, pp. 89–95, 2020, doi: 10.21512/comtech.v11i2.6453.

[9] S. Karuppuchamy and D. C. Sathyakumar, "AI-Driven Domainification and Configuration-First Microservices: A Framework for Monolith-to-Microservices Modernization," *Int. Conf. Artif. Intell. Comput. Data Sci. Appl.*, pp. 1–7, 2026, doi:10.1109/ACDSA67686.2026.114681.

[10] D. da P. Pereira and A. R. Silva, "A Domain-Driven Design Simulator for Business Logic-Rich Microservice Systems," pp. 1–32, 2026, [Online]. Available: http://arxiv.org/abs/2605.01159

[11] P. Fan, J. Liu, W. Yin, H. Wang, X. Chen, and H. Sun, "2PC*: a distributed transaction concurrency control protocol of multi-microservice based on cloud computing platform," *J. Cloud Comput.*, vol. 9, no. 1, 2020, doi: 10.1186/s13677-020-00183-w.

[12] R. Bhattacharya, Y. Gao, and T. Wood, "Dynamically Balancing Load with Overload Control for Microservices," *ACM Trans. Auton. Adapt. Syst.*, vol. 19, no. 4, 2024, doi: 10.1145/3676167.

[13] R. Ramdhani, A. Sujjada, and N. Nugraha, "Enhancing E-Commerce System Scalability Through Event-Driven Architecture with RabbitMQ and Docker," *bit-Tech*, vol. 8, no. 2, pp. 1437–1445, 2025, doi: 10.32877/bt.v8i2.2948.

[14] S. Duggirala and R. Mishra, Dr, "Microservices Architecture: A Comparative Analysis of Domain-Driven Design and Service-Oriented Architecture," *Int. J. Res. Publ. Semin.*, vol. 16, no. 2, pp. 56–62, 2025, [Online]. Available:https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4984335

SIMULASI MONTE CARLO MIGRASI C/C++ KE RUST TERHADAP BUG DAN BIAYA

Muhammad Fawaz Hadimatulloh[1)], Muhammad Anshar Putra[2)], Ahmad Budi Setiawan[2)]
[1)]Program Studi Teknik Informatika, Universitas Mataram
Corresponding author e-mail: mufaha2004@gmail.com, F1d02310078@unram.ac.id, setiawanahmadbudi415@gmail.com

Abstrak

C/C++ masih banyak digunakan pada perangkat lunak sistem karena performanya tinggi, tetapi pengelolaan memori manual dapat memunculkan kerentanan seperti buffer overflow, use-after-free, dan double free. Rust menawarkan pendekatan memory safety melalui ownership dan borrow checker, sehingga menarik sebagai alternatif migrasi untuk komponen yang rawan bug. Penelitian ini bertujuan mengevaluasi dampak beberapa strategi migrasi C/C++ ke Rust terhadap jumlah bug, memory-safety bug, biaya pemeliharaan, biaya migrasi, dan waktu break-even. Metode yang digunakan adalah simulasi Monte Carlo berbasis data lokal MegaVul Simple, Big-Vul, dan RustSec Advisory DB. Simulasi dilakukan pada 100 modul selama 24 bulan dengan 1.000 iterasi untuk lima skenario: tidak migrasi, migrasi lambat, migrasi sedang, migrasi agresif, dan hybrid security-critical. Hasil simulasi menunjukkan bahwa skenario migrasi agresif menghasilkan rata-rata total bug terendah sebesar 60,22 bug, memory-safety bug terendah sebesar 27,15 bug, biaya total rata-rata terendah sebesar Rp446.502.865, serta break-even tercepat pada bulan ke-15. Hasil ini menunjukkan bahwa migrasi agresif dapat menjadi pilihan paling optimal dalam asumsi simulasi, meskipun tetap membutuhkan kesiapan biaya awal, pelatihan, dan adaptasi produktivitas pengembang.
Kata kunci: Biaya Migrasi; C/C++; Memory Safety; Monte Carlo; Rust

Abstract

C/C++ remains widely used in systems software due to its high performance, yet manual memory management may introduce vulnerabilities such as buffer overflow, use-after-free, and double free. Rust provides memory-safety mechanisms through ownership and the borrow checker, making it a relevant alternative for migrating bug-prone components. This study evaluates the impact of several C/C++ to Rust migration strategies on total bugs, memory-safety bugs, maintenance cost, migration cost, and break-even time. The method uses a Monte Carlo simulation based on local MegaVul Simple, Big-Vul, and RustSec Advisory DB datasets. The simulation models 100 modules over 24 months with 1,000 iterations across five scenarios: no migration, slow migration, moderate migration, aggressive migration, and security-critical hybrid migration. The results show that aggressive migration produces the lowest average total bugs at 60.22, the lowest memory-safety bugs at 27.15, the lowest average total cost at IDR 446,502,865, and the fastest break-even point at month 15. These findings indicate that aggressive migration can be the most optimal strategy under the simulation assumptions, although it still requires initial migration cost, training, and productivity adaptation.
Keyword: C/C++; Memory Safety; Migration Cost; Monte Carlo; Rust

I. PENDAHULUAN

Perangkat lunak sistem banyak dibangun menggunakan C dan C++ karena kedua bahasa tersebut memberikan kontrol tinggi terhadap memori, performa, dan integrasi perangkat keras. Namun, fleksibilitas tersebut juga membawa risiko. Kesalahan pengelolaan memori, seperti penggunaan pointer yang tidak valid, *buffer overflow*, *use-after-free*, dan *double free*, dapat berkembang menjadi bug serius maupun celah keamanan. Pada konteks sistem yang semakin kompleks, biaya untuk memperbaiki bug dan menjaga keamanan perangkat lunak dapat meningkat secara signifikan.

Rust hadir sebagai bahasa pemrograman sistem yang menargetkan performa tinggi sekaligus *memory safety*. Rust menggunakan sistem ownership yang diperiksa oleh *compiler* untuk mengatur memori tanpa *garbage collector* [1]. Pendekatan ini menjadikan Rust menarik untuk menggantikan atau melengkapi komponen C/C++ yang memiliki risiko *memory-safety bug*. Meskipun demikian, migrasi dari C/C++ ke Rust tidak selalu sederhana karena membutuhkan biaya *rewrite*, pelatihan pengembang, dan adaptasi produktivitas.

Masalah utama dalam migrasi adalah menentukan strategi yang paling seimbang antara pengurangan *bug* dan biaya migrasi. Migrasi terlalu lambat dapat membuat manfaat

keamanan baru terasa dalam jangka panjang, sedangkan migrasi terlalu agresif dapat meningkatkan biaya awal dan menurunkan produktivitas sementara. Oleh karena itu, dibutuhkan pendekatan simulasi untuk membandingkan beberapa skenario migrasi sebelum keputusan teknis diterapkan pada proyek nyata.

Penelitian ini menggunakan simulasi Monte Carlo untuk mengevaluasi dampak migrasi C/C++ ke Rust terhadap total *bug*, *memory-safety bug*, biaya, dan waktu *break-even*. Tujuan penelitian adalah membandingkan lima skenario migrasi dan menentukan skenario yang paling optimal berdasarkan hasil simulasi.

II. TINJAUAN PUSTAKA

A. *Memory Safety* pada C/C++ dan Rust

Memory safety berkaitan dengan kemampuan program untuk mengakses memori secara valid dan terkontrol. Pada C/C++, pengelolaan memori banyak bergantung pada programmer sehingga kesalahan implementasi dapat menyebabkan *vulnerability*. Rust menggunakan *ownership*, *borrowing*, dan *lifetime* untuk memastikan aturan akses memori diperiksa pada waktu kompilasi. Dokumentasi Rust menjelaskan bahwa *ownership* adalah sekumpulan aturan yang mengatur bagaimana program Rust mengelola memori, dan aturan tersebut diperiksa oleh *compiler* tanpa memperlambat program saat *runtime* [1].

Walaupun Rust mengurangi banyak risiko *memory-safety bug*, penggunaan *unsafe code* dan kesalahan abstraksi tetap dapat menjadi sumber masalah. Xu et al. menunjukkan bahwa *memory-safety bug* pada Rust umumnya berkaitan dengan penggunaan *unsafe code* dan pola implementasi tertentu [4]. Dengan demikian, migrasi ke Rust bukan berarti menghapus seluruh risiko keamanan, tetapi dapat menurunkan peluang *bug* tertentu jika kode ditulis dengan praktik yang benar.

B. Dataset Kerentanan Perangkat Lunak

Penelitian ini memanfaatkan tiga sumber data. MegaVul merupakan dataset kerentanan C/C++ berskala besar yang mengumpulkan informasi CVE dan perubahan kode dari proyek *open-source* [5]. Big-Vul juga digunakan sebagai dataset historis kerentanan C/C++ yang sering dipakai dalam penelitian *vulnerability detection* [6]. Untuk sisi Rust, RustSec Advisory DB digunakan sebagai sumber advisory keamanan ekosistem Rust [7].

Penggunaan dataset tersebut bertujuan untuk membuat parameter simulasi lebih dekat dengan karakteristik kerentanan nyata. MegaVul dan Big-Vul membantu mengestimasi bug serta proporsi memory-related weakness pada C/C++, sedangkan RustSec digunakan untuk melihat karakteristik advisory pada ekosistem Rust.

C. Simulasi Monte Carlo

Simulasi Monte Carlo adalah pendekatan komputasi yang menggunakan pengulangan acak untuk memperkirakan perilaku suatu sistem yang memiliki ketidakpastian. Dalam konteks penelitian ini, ketidakpastian muncul dari peluang *bug*, biaya perbaikan *bug*, biaya rewrite modul, biaya pelatihan, dan perubahan produktivitas pengembang. Dengan melakukan 1.000 iterasi, simulasi dapat memberikan estimasi rata-rata untuk setiap skenario migrasi, bukan hanya satu hasil deterministik.

III. METODOLOGI PENELITIAN

Penelitian dilakukan dengan pendekatan simulasi. Objek simulasi adalah proyek perangkat lunak hipotetis yang terdiri dari 100 modul dan diamati selama 24 bulan. Setiap skenario dijalankan sebanyak 1.000 iterasi *Monte Carlo* sehingga total hasil simulasi mencapai 120.000 baris data.

A. Sumber Data

Dataset yang berhasil dimuat pada notebook terdiri dari MegaVul Simple sebanyak 50.000 baris, Big-Vul sebanyak 4.432 baris, dan RustSec Advisory DB sebanyak 1.099 advisory. MegaVul digunakan untuk mengestimasi karakteristik *bug* C/C++, Big-Vul menjadi data pendukung historis, dan RustSec digunakan untuk memperkirakan karakteristik bug pada ekosistem Rust.

B. Parameter Simulasi

Parameter utama simulasi ditunjukkan pada Tabel 1. Simulasi menggunakan peluang *bug* C/C++ sebesar 8%, peluang bug Rust sebesar 2%, rasio memory bug C/C++ sebesar 43,29%, dan rasio *memory bug* Rust sebesar 45,04%. Biaya perbaikan bug normal berada pada rentang Rp500.000 sampai Rp2.000.000, sedangkan biaya perbaikan *memory bug* berada pada rentang Rp3.000.000 sampai Rp8.000.000.

Tabel 1. Parameter utama simulasi

Parameter	Nilai
Jumlah modul awal	100
Durasi simulasi	24 bulan
Jumlah developer	5

Parameter	Nilai
Jumlah iterasi	1.000
Probabilitas *bug* C/C++	8%
Probabilitas *bug* Rust	2%
Rasio *memory bug* C/C++	43,29%
Rasio *memory bug Rust*	45,04%

C. Skenario Migrasi

Lima skenario yang dibandingkan ditunjukkan pada Tabel 2. S0 digunakan sebagai baseline tanpa migrasi. S1, S2, dan S3 merepresentasikan migrasi lambat, sedang, dan agresif. S4 merepresentasikan pendekatan *hybrid security-critical*, yaitu migrasi hanya sampai 50% modul dengan fokus pada komponen kritis.

Tabel 2. Skenario migrasi C/C++ ke Rust

Skenario	Rate	Maks. migrasi
S0 Tidak Migrasi	0,00	0%
S1 Migrasi Lambat	0,05	100%
S2 Migrasi Sedang	0,10	100%
S3 Migrasi Agresif	0,20	100%
S4 *Hybrid Security-Critical*	0,10	50%

D. Perhitungan Biaya dan *Break-even*

Biaya total dihitung dari penjumlahan biaya pemeliharaan *bug*, biaya migrasi, biaya rewrite, biaya pelatihan, dan biaya kehilangan produktivitas. *Break-even* ditentukan sebagai bulan pertama ketika akumulasi biaya suatu skenario migrasi menjadi lebih rendah dibandingkan *baseline* S0. Dengan cara ini, setiap skenario tidak hanya dinilai dari jumlah bug, tetapi juga dari kelayakan biaya selama periode simulasi.

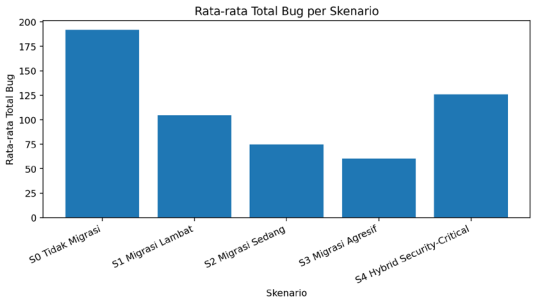
IV. HASIL DAN PEMBAHASAN

Hasil ringkasan simulasi ditunjukkan pada Tabel 3. S0 menghasilkan rata-rata total *bug* tertinggi, yaitu 191,72 *bug*, dengan rata-rata *memory bug* 83,32. Skenario migrasi menghasilkan penurunan *bug* yang berbeda-beda sesuai tingkat agresivitas migrasi. S3 Migrasi Agresif memberikan hasil terbaik dengan rata-rata total *bug* 60,22 dan rata-rata memory *bug* 27,15.

Tabel 3. Ringkasan hasil simulasi setiap skenario

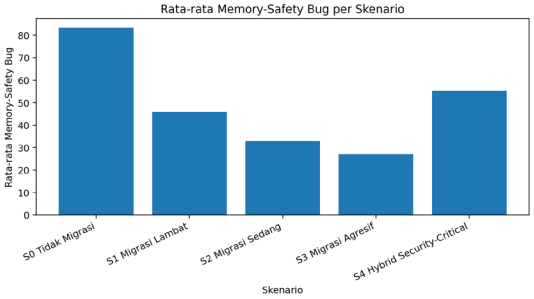
Skenario	Avg total bug	Avg memory bug	Avg total cost	Break-even
S0 Tidak Migrasi	191,72	83,32	Rp636.323.078	Baseline
S1 Migrasi Lambat	104,48	45,85	Rp590.935.801	Bulan 22
S2 Migrasi Sedang	74,74	32,91	Rp491.798.703	Bulan 17
S3 Migrasi Agresif	60,22	27,15	Rp446.502.865	Bulan 15
S4 *Hybrid Security-Critical*	125,94	55,24	Rp561.330.301	Bulan 17

Gambar 1 menunjukkan perbandingan rata-rata total *bug*. S3 memiliki nilai paling rendah, diikuti S2, S1, dan S4. Pola ini menunjukkan bahwa semakin cepat proporsi modul C/C++ digantikan oleh Rust, semakin besar penurunan total *bug* yang diperoleh dalam periode 24 bulan.



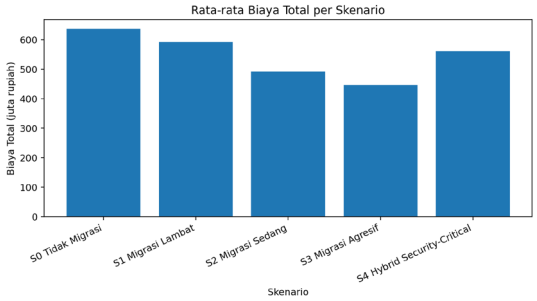
Gambar 1. Perbandingan rata-rata total *bug* pada setiap skenario.

Gambar 2 menunjukkan perbandingan *memory-safety bug*. S3 kembali menjadi skenario terbaik dengan rata-rata 27,15 *memory bug*. Dibandingkan S0, nilai ini menunjukkan penurunan sekitar 67,42%. S2 juga memberikan penurunan besar, yaitu sekitar 60,50%, tetapi masih kalah dari S3.



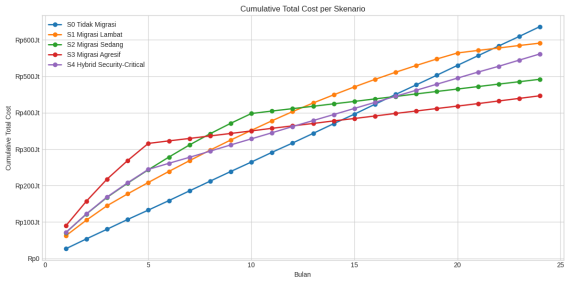
Gambar 2. Perbandingan rata-rata *memory-safety bug* pada setiap skenario.

Dari sisi biaya, Gambar 3 menunjukkan bahwa S0 memiliki biaya total rata-rata tertinggi sebesar Rp636.323.078. Hal ini terjadi karena tidak ada biaya migrasi, tetapi biaya pemeliharaan *bug* tetap tinggi sepanjang periode simulasi. S3 memiliki biaya total rata-rata terendah sebesar Rp446.502.865. Hasil ini menarik karena migrasi agresif tetap menjadi paling murah secara rata-rata, meskipun memiliki biaya migrasi dan adaptasi di awal.



Gambar 3. Perbandingan rata-rata biaya total pada setiap skenario.

Gambar 4 memperlihatkan biaya kumulatif setiap skenario. Break-even tercepat diperoleh oleh S3 pada bulan ke-15. S2 dan S4 mencapai *break-even* pada bulan ke-17, sedangkan S1 baru mencapai *break-even* pada bulan ke-22. Dengan demikian, migrasi yang terlalu lambat membutuhkan waktu lebih lama untuk menutup biaya awal migrasi.



Gambar 4. Perbandingan biaya kumulatif dan titik *break-even* setiap skenario.

Secara umum, hasil simulasi menunjukkan bahwa migrasi ke Rust dapat mengurangi *bug* dan biaya jangka menengah. Namun, hasil tersebut bergantung pada asumsi parameter, terutama probabilitas bug, biaya perbaikan *bug*, biaya *rewrite*, dan *learning curve* pengembang. S4 *Hybrid Security-Critical* lebih hemat biaya migrasi dibandingkan migrasi penuh, tetapi penurunan *bug* tidak sebesar S2 dan S3 karena hanya sebagian modul yang dimigrasikan. Strategi ini tetap relevan apabila organisasi belum siap melakukan migrasi penuh, tetapi ingin memprioritaskan komponen yang kritis terhadap keamanan.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil simulasi Monte Carlo, migrasi C/C++ ke Rust berpotensi menurunkan total *bug*, *memory-safety bug*, dan biaya total dibandingkan skenario tanpa migrasi. Dari lima skenario yang diuji, S3 Migrasi Agresif menjadi skenario paling optimal karena menghasilkan rata-rata *total bug* terendah sebesar 60,22, *memory-safety bug* terendah sebesar 27,15, biaya total rata-rata terendah sebesar Rp446.502.865, serta *break-even* tercepat pada bulan ke-15. Hasil ini menunjukkan bahwa migrasi agresif dapat memberikan manfaat teknis dan ekonomis dalam periode 24 bulan, selama organisasi mampu menanggung biaya awal dan mengelola penurunan produktivitas sementara. Untuk penelitian selanjutnya, simulasi dapat dikembangkan dengan model proyek yang lebih rinci, pemisahan modul berdasarkan tingkat kritikalitas, serta validasi menggunakan data proyek nyata agar hasil estimasi semakin akurat.

VI. UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Program Studi Teknik Informatika Universitas Mataram atas dukungan pembelajaran dan

arahan akademik yang menjadi dasar penyusunan penelitian ini.

DAFTAR PUSTAKA

[1] S. Klabnik and C. Nichols, The Rust Programming Language. No Starch Press, 2023. [Online]. Tersedia: https://doc.rust-lang.org/book/

[2] National Security Agency, “Software memory safety,” 2022. [Online]. Tersedia: https://media.defense.gov/

[3] Cybersecurity and Infrastructure Security Agency, “The case for memory safe roadmaps,” 2023. [Online]. Tersedia: https://www.cisa.gov/

[4] H. Xu, Z. Chen, M. Sun, Y. Zhou, and M. R. Lyu, “Memory-safety challenge considered solved? An in-depth study with all Rust CVEs,” arXiv:2003.03296, 2020.

[5] C. Ni, L. Shen, X. Yang, Y. Zhu, and S. Wang, “MegaVul: A C/C++ vulnerability dataset with comprehensive code representation,” arXiv:2406.12415, 2024.

[6] J. Fan, Y. Li, S. Wang, and T. N. Nguyen, “A C/C++ code vulnerability dataset with code changes and CVE summaries,” in Proc. Mining Software Repositories (MSR), 2020.

[7] RustSec, “RustSec Advisory Database,” 2026. [Online]. Tersedia: https://github.com/RustSec/advisory-db

[8] M. Coblenz, M. Mazurek, and M. Hicks, “Garbage collection makes Rust easier to use: A randomized controlled trial of the Bronze garbage collector,” arXiv:2110.01098, 2021.