Journal of Mechanical Engineering Education
(JURNAL PENDIDIKAN TEKNIK MESIN)
Available online at https://ejournal.upi.edu/index.php/jmee
(e.ISSN: 2715-4734, p.ISSN: 2356-4997)

Developing an Industry-Based DevOps Learning Framework Through Microservices, CI/CD, and AI-Driven Bus Scheduling Optimization for Vocational Education

Pengembangan Kerangka Pembelajaran DevOps Berbasis Industri melalui Microservices, CI/CD, dan Optimasi Penjadwalan Bus Berbasis Kecerdasan Buatan untuk Pendidikan Vokasi

Roni Andarsyah[1], Cahyo Prianto[2*], Dodi Permadi[3]

[1,2]D4 Teknik Informatika, Sekolah Teknologi dan Informasi, Universitas Logistik dan Bisnis Internasional, Jl. Sariasih No.54, Sarijadi, Kec. Sukasari, Kota Bandung, Jawa Barat 40151
[3]D4 Logistik Bisnis, Sekolah Logistik dan Transportasi, Universitas Logistik dan Bisnis Internasional, Jl. Sariasih No.54, Sarijadi, Kec. Sukasari, Kota Bandung, Jawa Barat 40151
cahyo@ulbi.ac.id*

ABSTRACT/ABSTRAK

The digital transformation within the public transportation industry demands information systems that are responsive, modular, and capable of supporting data-driven decision-making. This research integrates two previously independent strategic approaches: (1) a DevOps architecture utilizing a CI/CD Pipeline based on Microservices, enhanced with API Logging and centralized monitoring to support the ticketing Dashboard system for bus operators, and (2) a hybrid optimization approach combining Machine Learning and Genetic Algorithms (GA) to optimize airport bus scheduling. Beyond these technical contributions, this research develops an Industry-Based DevOps Learning Framework, a training-oriented framework that can be adapted for vocational education and industry-based learning in DevOps and data-driven optimization. The implementation of CI/CD and Microservices enables rapid, scalable, and zero-downtime deployment, while centralized monitoring using Elasticsearch, Logstash, Kibana (ELK) Stack and Prometheus–Grafana improves Observability and early anomaly detection. On the operational side, a LightGBM model is employed to predict the probability of passenger occupancy (load factor ≥ 70%), and these predictions are used as a fitness function within the GA to generate optimal schedules. Experimental results show significant improvements: total estimated passengers increased from 478 to 841, and the average load factor improved from 82% to 87%. Furthermore, the proposed integrated system was validated as a project-based training platform through a pilot training involving 22 vocational students, who acquired hands-on competencies in Microservices development, CI/CD Pipeline, and intelligent optimization within a real-world transportation context, showing statistically significant improvement across all six competency domains (Wilcoxon signed-rank test, $p < 0.001$) with large effect sizes ($r = 0.78$-0.94). The integration of these two approaches results

ARTICLE INFO

Article History:
Submitted/Received
03 April 2026

First Revised
12 Augustus 2026

Accepted
24 September 2026

Online Date
28 September 2026

Publication Date
01 December 2026

Keywords:
Genetic Algorithm; Machine Learning; Microservices; Ticketing System, Vocational Training

Kata kunci:
Genetic Algorithm; Machine Learning; Layanan Mikro; Sistem Tiket, Pelatihan Vokasi

Copyright © [2026] Universitas Pendidikan Indonesia. This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

DOI: https://doi.org/10.17509/jmee.v13i2.98622
P-ISSN 2356-4997 | E-ISSN 2715-4734

in a reliable, adaptive ticketing and scheduling system capable of supporting real-time operational decision-making for bus companies.

Transformasi digital dalam industri transportasi publik menuntut sistem informasi yang responsif, modular, dan mampu mendukung pengambilan keputusan berbasis data. Penelitian ini mengintegrasikan dua pendekatan strategis yang sebelumnya independen: (1) arsitektur *DevOps* yang memanfaatkan *Pipeline* CI/CD berbasis *Microservices*, ditingkatkan dengan pencatatan API dan pemantauan terpusat untuk mendukung sistem *Dashboard* tiket untuk operator bus, dan (2) pendekatan optimasi hibrida yang menggabungkan *Machine Learning* dan *Genetic Algorithm* (GA) untuk mengoptimalkan penjadwalan bus bandara. Selain kedua kontribusi teknis tersebut, penelitian ini mengembangkan Kerangka Pembelajaran *DevOps* Berbasis Industri (*Industry-Based DevOps Learning Framework*), sebuah kerangka berbasis pelatihan yang dapat diadaptasi dalam pendidikan vokasi dan pembelajaran berbasis industri, khususnya pada bidang *DevOps* dan optimasi berbasis data. Implementasi CI/CD dan *Microservices* memungkinkan penyebaran yang cepat, terukur, dan tanpa *downtime*, sementara pemantauan terpusat menggunakan *Elasticsearch, Logstash, Kibana* (ELK) *Stack* dan *Prometheus Grafana* meningkatkan observabilitas dan deteksi anomali dini. Di sisi operasional, model *LightGBM* digunakan untuk memprediksi probabilitas okupansi penumpang (faktor muat ≥ 70%), dan prediksi ini digunakan sebagai fungsi *fitness* dalam GA untuk menghasilkan jadwal optimal. Hasil eksperimen menunjukkan peningkatan yang signifikan: total perkiraan penumpang meningkat dari 478 menjadi 841, dan faktor muatan rata-rata meningkat dari 82% menjadi 87%. Lebih lanjut, sistem terintegrasi yang diusulkan divalidasi sebagai platform pelatihan berbasis proyek (*project-based training*), yang memungkinkan peserta didik memperoleh kompetensi praktis dalam pengembangan *Microservices*, *Pipeline*, serta optimasi cerdas dalam konteks transportasi nyata, menunjukkan peningkatan kompetensi yang signifikan secara statistik pada seluruh enam domain (uji *Wilcoxon* signed-rank, $p < 0,001$) dengan *effect size* besar ($r = 0,78$-$0,94$). Integrasi kedua pendekatan ini menghasilkan sistem tiket dan penjadwalan yang andal dan adaptif yang mampu mendukung pengambilan keputusan operasional secara *real-time* bagi perusahaan bus.

1. PENDAHULUAN

Bus sebagai salah satu transportasi publik memiliki peran yang sangat strategis, tidak hanya sebagai sarana perpindahan orang dalam jumlah yang massal, tetapi juga sebagai tulang punggung mobilitas orang baik antar kota maupun antar provinsi, serta ikut membantu menjadi penggerak roda ekonomi masyarakat (Prianto *et al*., 2026). Kenyamanan merupakan salah satu faktor utama pemilihan transportasi publik yang akan digunakan (Mardikawati *et al*., 2023; Sugiyanto *et al*., 2021). Salah satu upaya untuk mewujudkan kenyamanan tersebut adalah dengan melakukan transformasi digital dalam pelayanan transportasi publik. Transformasi digital menuntut perusahaan transportasi publik memiliki sistem informasi yang responsif, terintegrasi, dan berbasis data (Herawati dan Yenita, 2023; Putri *et al*., 2025; Ariska *et al*., 2023). Kebutuhan ini melahirkan dua tantangan teknis: bagaimana sistem tiket dikembangkan secara cepat dan andal, dan bagaimana operasional armada dijadwalkan secara efisien. Pada aspek pertama, *Continuous Integration* dan *Continuous Delivery*

(CI/CD) bersama arsitektur *Microservices* mempercepat pengembangan, sementara *Observability* melalui *ELK Stack*, *Prometheus*, dan *Grafana* mendukung pemantauan *real-time* (Amrullah & Marthasari, 2024).

Pendekatan *Continuous Integration* (CI) dan *Continuous Deployment* (CD). CI merupakan proses otomatisasi penggabungan perubahan kode ke dalam repositori utama, sedangkan CD adalah proses otomatisasi *build, test*, dan *deployment* aplikasi ke *server* produksi (Bakti *et al*., 2026; Railić dan Savić, 2021). Sedangkan *Microservices* memungkinkan adanya skalabilitas individual pada tiap-tiap *service* yang di gunakan, jika pada suatu *service* mengalami lonjakan *traffic* maka hanya *service* tersebut yang perlu ditingkatkan kapasitasnya, tanpa mempengaruhi layanan yang lain (Arief dan Chaniago, 2025).

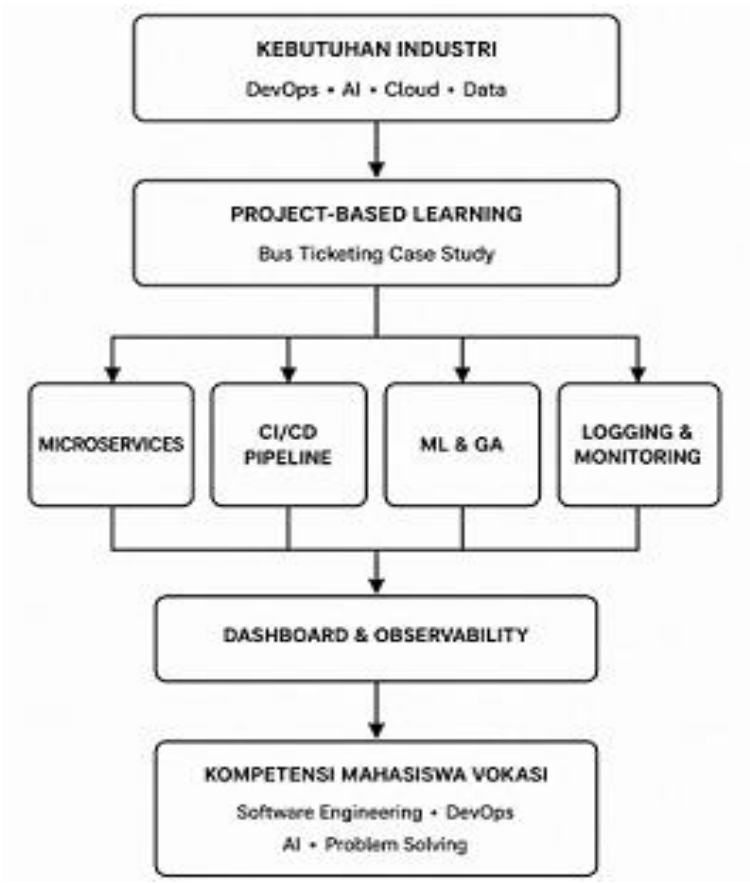
Pada aspek kedua, penjadwalan armada konvensional sering menyebabkan rendahnya okupansi, salah satu penerapan utama *Machine Learning* dalam rantai pasokan adalah *demand forecasting* atau peramalan permintaan (Manik dan Panjaitan, 2025). *Machine Learning* untuk memprediksi permintaan penumpang, dikombinasikan dengan *Genetic Algorithm* sebagai metode optimasi, terbukti meningkatkan efisiensi operasional pada berbagai konteks transportasi (Mutmainah *et al*., 2025)

Meskipun kedua aspek ini banyak diteliti, penelitian sistem tiket bus umumnya berfokus pada aplikasi berbasis *web*, sedangkan penelitian *DevOps* menitik beratkan pada *deployment* dan *monitoring* semata. Penelitian yang mengintegrasikan seluruh komponen ini sekaligus memanfaatkannya sebagai media pembelajaran berbasis proyek pada pendidikan vokasi masih sangat terbatas (Dwiputra *et al*., 2024; Lubis *et al*., 2024; Minarni *et al*., 2024; Suminten *et al*., 2025; Zuhri *et al*., 2022).

Berdasarkan kesenjangan tersebut, penelitian ini merumuskan tiga pertanyaan penelitian: (1) bagaimana merancang arsitektur *Microservices* dengan *Pipeline* CI/CD, API *Logging*, dan pemantauan terpusat yang layak digunakan pada sistem tiket dan penjadwalan bus; (2) sejauh mana integrasi model *LightGBM* dan *Genetic Algorithm* mampu meningkatkan estimasi penumpang dan faktor muat dibandingkan skema penjadwalan awal; dan (3) sejauh mana arsitektur dan alur kerja yang dihasilkan dapat diadaptasi menjadi kerangka pelatihan berbasis proyek untuk pendidikan vokasi.

Untuk menjawab pertanyaan tersebut, penelitian ini mengintegrasikan arsitektur *Microservices*, *Pipeline* CI/CD, API *Logging*, dan *centralized Observability* dengan pendekatan *Machine Learning–Genetic Algorithm* pada satu sistem tiket dan penjadwalan bus, sekaligus dirancang agar dapat direplikasi sebagai media pembelajaran berbasis proyek.

Kebaruannya terletak pada tiga hal: integrasi teknis yang menyatukan CI/CD, *Microservices*, *Observability*, dan ML-GA yang sebelumnya diteliti terpisah pada literatur *DevOps* maupun sistem tiket bus, penggunaan prediksi *LightGBM* sebagai fungsi *fitness* pada *Genetic Algorithm* untuk penjadwalan bus bandara, yang belum banyak dibahas pada konteks transportasi domestik; dan perancangan arsitektur serta alur kerja ini agar langsung diadaptasi menjadi modul pelatihan vokasi berbasis proyek, bukan sekadar catatan implikasi di akhir penelitian. Pada Gambar 1 ditunjukkan kerangka pembelajaran DevOps berbasis industri untuk pengembangan kompetensi mahasiswa vokasi.



Gambar 1. Kerangka pembelajaran *DevOps* berbasis industri untuk pengembangan kompetensi mahasiswa vokasi

Kontribusi penelitian ini bersifat ganda dan setara sejak awal: Pada sisi rekayasa sistem berupa arsitektur tiket dan penjadwalan yang lebih adaptif dan efisien, dan pada sisi pendidikan vokasi berupa Kerangka Pembelajaran *DevOps* Berbasis Industri (*Industry-Based DevOps Learning Framework*) yang memanfaatkan seluruh proses pengembangan sistem ini sebagai media pembelajaran berbasis proyek untuk memperkuat kompetensi mahasiswa sesuai kebutuhan dunia kerja pada era transformasi digital. Pendidikan vokasi berorientasi pada pengembangan kompetensi kerja melalui pembelajaran yang mengintegrasikan teori dengan praktik. *Project-Based Learning* (PjBL) menempatkan

peserta didik sebagai pusat pembelajaran melalui penyelesaian proyek nyata, sehingga mahasiswa mengembangkan pemahaman konseptual sekaligus kemampuan berpikir kritis, pemecahan masalah, dan kolaborasi (Nurzakiyah, 2025). Pendekatan ini semakin relevan ketika lingkungan belajar mengadopsi praktik industri seperti CI/CD, *Microservices*, dan *Observability*, serta kompetensi kecerdasan buatan seperti *Machine Learning* dan *Genetic Algorithm*, karena mahasiswa memperoleh pengalaman langsung menerapkan metode ini pada permasalahan operasional nyata (Amrullah dan Marthasari, 2024).

Penelitian ini memanfaatkan pengembangan sistem tiket dan penjadwalan bus sebagai media pembelajaran berbasis proyek: Mahasiswa mempelajari pembangunan sistem *Microservices*, *Pipeline* CI/CD, *API Logging*, *Observability*, hingga ML-GA, sehingga sistem yang dikembangkan berfungsi ganda sebagai kontribusi teknis sekaligus sarana pembelajaran yang mendukung kompetensi lulusan pendidikan vokasi. Untuk mengoperasionalkan kerangka pembelajaran *DevOps* berbasis industri pada gambar 1 dan tabel 1 memetakan setiap komponen teknis penelitian ini terhadap kompetensi vokasi yang dilatih beserta indikator capaiannya.

Tabel 1. Pemetaan komponen teknis terhadap kompetensi vokasi

Komponen Teknis	Kompetensi Vokasi yang Dilatih	Indikator Capaian
Arsitektur *Microservices*	Perancangan sistem terdistribusi	Mampu memisahkan fungsi aplikasi menjadi layanan independen dan menjelaskan alasan pemisahannya
Pipeline CI/CD	Otomasi *build*, *test*, dan *deployment*	Mampu menyusun *Pipeline build-test*-deploy tanpa bantuan instruktur
API *Logging* & *Observability*	Pemantauan dan diagnosis sistem	Mampu membaca *dashboard monitoring* untuk mendeteksi anomali layanan
Model Prediksi *LightGBM*	Pemodelan prediktif berbasis data	Mampu menjelaskan alur pelatihan model dan menginterpretasikan metrik evaluasi
Genetic Algorithm	Optimasi berbasis algoritma evolusioner	Mampu menjelaskan representasi individu, fungsi *fitness*, dan proses seleksi-*crossover*-mutasi
Integrasi End-to-End	Pemecahan masalah lintas komponen	Mampu mengintegrasikan hasil prediksi ke dalam proses optimasi dan menjelaskan dampaknya terhadap keputusan operasional

2. METODE PENELITIAN

2.1 Desain Penelitian

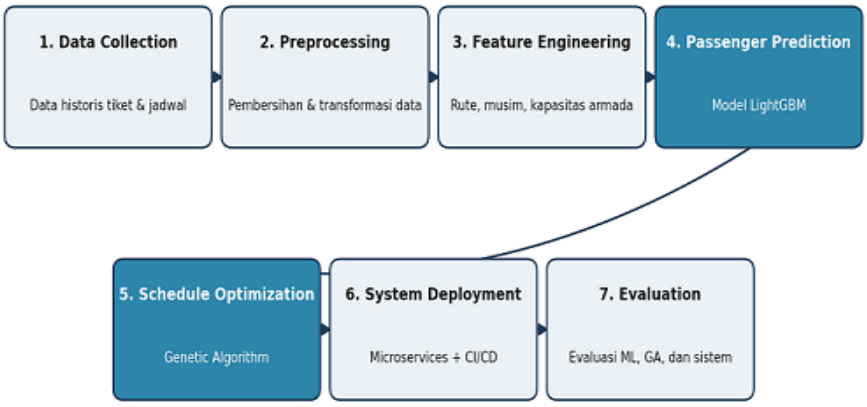
Penelitian ini menggunakan pendekatan *Design Science Research* (DSR) yang dikombinasikan dengan metode eksperimen kuantitatif (*quantitative experimental research*). Pendekatan DSR dipilih karena penelitian berfokus pada perancangan, implementasi, dan evaluasi sebuah artefak berupa sistem yang dikembangkan dan digunakan oleh manusia untuk memecahkan masalah praktis yang menjadi kepentingan umum (Haryanti *et al*., 2022, Adjie *et al*. 2026), dalam hal ini adalah sistem *ticketing* dan penjadwalan bus yang mengintegrasikan arsitektur *Microservices*, *Continuous Integration and Continuous Delivery* (CI/CD), API *logging*, *centralized observability*, *machine learning*, dan *genetic algorithm*. Selanjutnya dilakukan eksperimen kuantitatif untuk mengevaluasi performa model prediksi penumpang dan optimasi penjadwalan berdasarkan metrik evaluasi yang telah ditentukan.

Tahapan penelitian meliputi pengumpulan data operasional bus, *preprocessing* data, pembangunan model prediksi menggunakan *LightGBM*, optimasi penjadwalan menggunakan *Genetic Algorithm*, implementasi sistem berbasis *Microservices*, serta evaluasi performa sistem dan model prediksi.

2.2 Alur Kerja Penelitian

Tahapan penelitian ditunjukkan pada Gambar 2 dengan penjelasan sebagai berikut:

1. *Data Collection*: Mengumpulkan data historis pemesanan tiket, jadwal keberangkatan, okupansi bus, serta data operasional perusahaan.
2. *Data Preprocessing*: Melakukan pembersihan data, menghapus data duplikat, menangani missing value, serta melakukan transformasi atribut.
3. *Feature Engineering*: Membangun variabel prediktor seperti hari keberangkatan, musim liburan, rute, kapasitas armada, dan histori jumlah penumpang.
4. *Passenger Prediction*: Melatih model *LightGBM* untuk memprediksi jumlah penumpang pada setiap jadwal keberangkatan.
5. *Schedule Optimization*: Menggunakan hasil prediksi sebagai fungsi *fitness* pada *Genetic Algorithm* untuk menghasilkan jadwal operasional yang optimal.
6. *System Deployment*: Mengimplementasikan sistem menggunakan arsitektur *Microservices* dengan *Pipeline* CI/CD berbasis Docker.
7. *Evaluation*: Mengevaluasi performa model *Machine Learning*, optimasi *Genetic Algorithm*, serta performa sistem menggunakan metrik yang telah ditentukan.



Gambar 2. Diagram alur kerja penelitian (*research workflow*)

2.3 Arsitektur Sistem

Secara umum arsitektur sistem terdiri atas lima lapisan utama, yaitu *Presentation Layer, API Gateway, Microservices Layer, Database Layer,* dan *Monitoring Layer.* Alur kerja sistem berjalan sebagai berikut: permintaan pengguna pertama diterima pada *Presentation Layer*, kemudian diteruskan melalui *API Gateway* yang berfungsi sebagai titik masuk tunggal untuk *routing*, autentikasi, dan pembatasan laju permintaan. Selanjutnya, *API Gateway* mendistribusikan permintaan ke layanan yang relevan pada *Microservices Layer*, yang masing-masing berinteraksi dengan basis datanya sendiri pada *Database Layer*. Sepanjang alur ini, *Monitoring Layer* secara *paralel* mencatat *log* dan *metrik* dari setiap layanan untuk mendukung observabilitas sistem secara menyeluruh.

2.4 Model *Machine Learning*

Penelitian ini menggunakan algoritma *LightGBM (Light Gradient Boosting Machine)* untuk memprediksi jumlah penumpang berdasarkan data historis operasional bus. Tabel 2 menampilkan parameter LightGBM yang dipakai dalam penelitian ini. *LightGBM* dipilih karena memiliki keunggulan dalam menangani *dataset* berukuran besar, waktu pelatihan yang lebih cepat, serta efisiensi penggunaan memori dibandingkan algoritma *boosting* lainnya. Tahapan pembangunan model meliputi *preprocessing dataset, feature engineering, train-test split, training, prediction,* dan *evaluation.*

Tabel 2. Parameter *LightGBM*

Parameter	Nilai
Objective	*Regression*
Learning Rate	0.05
Num Leaves	31
Max Depth	7
Feature Fraction	0.8
Bagging Fraction	0.8

Parameter	Nilai
Boosting Type	GBDT
Metric	RMSE

2.5 Optimasi *Genetic Algorithm*

Genetic Algorithm digunakan untuk menentukan jadwal operasional bus berdasarkan hasil prediksi jumlah penumpang. Pada Tabel 3 ditunjukkan parameter Genetic Algorithm yang dipakai dalam penelitian. Tahapan *Genetic Algorithm* meliputi inisialisasi populasi, evaluasi *fitness*, seleksi, *crossover*, mutasi, pembentukan populasi baru, dan penghentian algoritma.

Tabel 3. Parameter *Genetic Algorithm*

Parameter	Nilai
Population Size	100
Max Generation	50
Crossover Rate	0.8
Mutation Rate	0.1
Selection	*Tournament*
Elitism	*Yes*

2.6 Pengaturan Eksperimen

Eksperimen dilakukan menggunakan lingkungan pengembangan berbasis *docker* dengan implementasi *Microservices* (Tabel 4). *Dataset* dibagi menjadi data pelatihan sebesar 80% dan data pengujian sebesar 20% menggunakan teknik *hold-out validation*. Model kemudian dievaluasi berdasarkan hasil prediksi terhadap data pengujian.

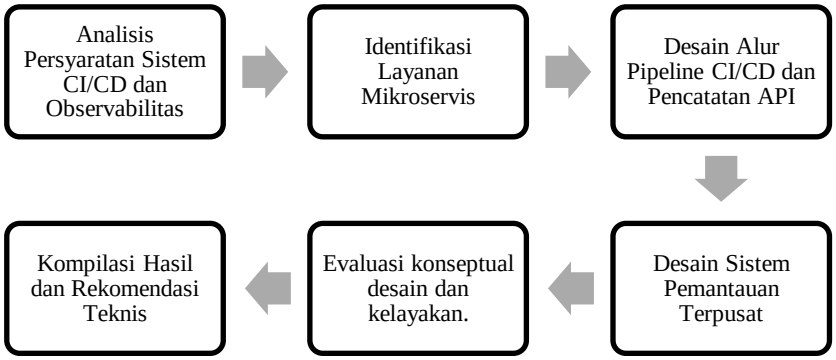
Tabel 4. Spesifikasi lingkungan eksperimen

Komponen	Spesifikasi
Processor	*Intel Core i7*
RAM	32 GB
OS	*Ubuntu Server* 24.04
Docker	27.x
PostgreSQL	16
Python	3.11
LightGBM	4.x
Scikit-learn	1.5

Bagian berikut menjelaskan secara rinci pelaksanaan dua komponen metodologi yang telah diperkenalkan pada bagian 2.1, yaitu desain dan implementasi arsitektur sistem tiket

(Bagian A) serta pengembangan dan evaluasi modul optimasi penjadwalan berbasis ML-GA (Bagian B). Kedua bagian ini menggabungkan rekayasa desain (*design and engineering*) untuk arsitektur sistem dengan eksperimen kuantitatif untuk model *Machine Learning* (ML) dan *Genetic Algorithm* (GA), sesuai pendekatan *design science research* yang dipilih.

A. Merancang Arsitektur Sistem Tiket Berbasis CI/CD, *Microservices*, Pencatatan API, dan Pemantauan Terpusat



Gambar 1. Metodologi penelitian untuk desain arsitektur sistem tiket

Berdasarkan gambar 3, analisis dimulai dari identifikasi persyaratan sistem CI/CD dan observabilitas, dilanjutkan pemodelan layanan mikro inti (pemesanan tiket, validasi pembayaran, pelaporan, penjadwalan) yang dirancang modular dan independen. *Pipeline* CI/CD dan pencatatan API dibangun menggunakan *GitLab CI* dan *Docker*, mencakup *build*, pengujian otomatis, dan *deployment* ke *staging* maupun produksi, sementara pemantauan terpusat menggunakan *ELK Stack*, *Prometheus*, *Grafana*, dan *OpenTelemetry* memberikan visibilitas *real-time* terhadap kinerja dan potensi anomali. Tahap akhir adalah evaluasi konseptual melalui studi literatur dan perbandingan dengan praktik *DevOps* industri transportasi, menghasilkan *blueprint* arsitektur CI/CD dan pedoman implementasi pemantauan yang menjadi dasar pengembangan *dashboard* tiket.

B. Pengembangan Modul Optimasi Jadwal Bus Berbasis *Machine Learning* dan *Genetic Algorithm*



Gambar 2. Metode penelitian untuk modul optimasi jadwal berbasis ML–GA

Penjelasan gambar 4 dimulai dari pengumpulan data operasional bus (rute, waktu keberangkatan, kapasitas, faktor muatan, atribut temporal), dilanjutkan analisis eksplorasi untuk mengidentifikasi pola permintaan dan ketidakseimbangan kelas, serta rekayasa fitur (faktor muatan tertunda, rata-rata bergulir, kategori waktu, penanda hari libur) dengan target *biner* (faktor muatan ≥70%). Model *LightGBM* dilatih dengan pembagian data 80:20 dan penanganan ketidakseimbangan data menggunakan *SMOTE*. Hasil prediksi digunakan sebagai fungsi *fitness* (*Fitness* = $\Sigma(P_i \times C_i)$) pada *Genetic Algorithm*, dengan tahapan seleksi, persilangan, dan mutasi, ditambah penalti konflik armada dan penyemaian hibrida untuk mempercepat konvergensi. Tahap akhir adalah *dashboard* visual berbasis *Streamlit* untuk menampilkan hasil prediksi, menjalankan optimasi GA, dan membandingkan KPI jadwal awal dengan jadwal hasil optimasi.

2.7 Evaluasi Implementasi Pelatihan Vokasi

Evaluasi implementasi kerangka pembelajaran dilaksanakan melalui pelatihan pilot berbasis proyek terhadap 22 mahasiswa vokasi semester 6 yang telah memiliki pengalaman pemrograman sebelumnya, menggunakan desain *one-group pretest-posttest* pada enam domain kompetensi: perancangan arsitektur *Microservices*, implementasi *Pipeline* CI/CD, API *Logging* dan *Observability*, pemodelan prediktif dengan *LightGBM*, optimasi dengan *Genetic Algorithm*, dan *Integrasi End-to-End*. Kompetensi dinilai menggunakan rubrik unjuk kerja skala 1–4 (1 = Belum Mampu, 2 = Cukup Mampu, 3 = Mampu, 4 = Sangat Mampu). Uji normalitas *Shapiro-Wilk* pada selisih skor pre*test*-post*test* menunjukkan seluruh domain tidak berdistribusi normal ($p < 0,05$), sehingga analisis menggunakan uji *Wilcoxon signed-rank* dengan tingkat signifikansi $\alpha = 0,05$.

3. HASIL PENELITIAN

A. Hasil Desain *Pipeline* CI/CD

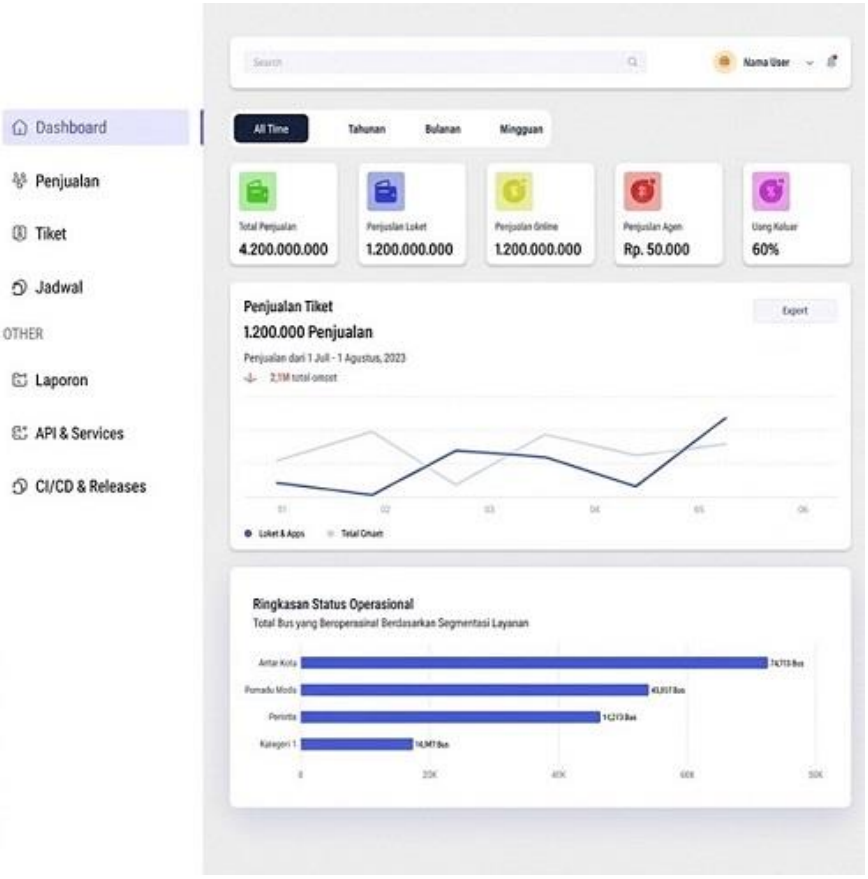
Sebelum implementasi, Perusahaan Bus mengoperasikan sistem *E-Ticketing* monolitik dengan seluruh fitur utama terintegrasi dalam satu aplikasi. Tabel 5 merangkum perbedaan karakteristiknya dengan *Pipeline* CI/CD hasil penelitian ini, yang dibangun menggunakan *GitLab CI/CD* dan *Docker Compose*.

Penelitian ini menghasilkan prototipe *Dashboard* Tiket Perusahaan Bus berbasis arsitektur *Microservices* yang terintegrasi dengan *Pipeline* CI/CD, API *Logging*, dan pemantauan terpusat. Sistem dibagi menjadi tujuh modul, yaitu *Dashboard* utama, penjualan, tiket, jadwal, laporan, API & Layanan, serta CI/CD & Rilis, masing-masing mendukung fungsi operasional dan pemantauan berbasis *DevOps*.

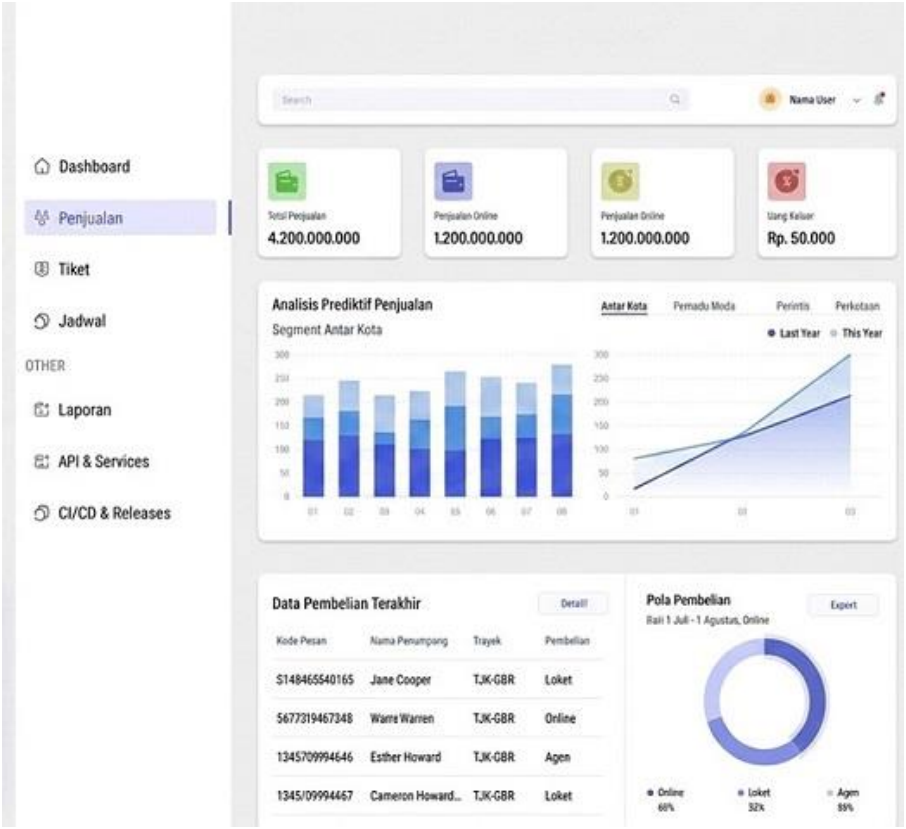
Tabel 5. Membandingkan karakteristik arsitektural, bukan hasil *benchmarking performa*.

Aspek	Sebelum (Monolitik)	Sesudah (CI/CD *Pipeline* + *Microservices*)
Arsitektur	Monolitik: seluruh fitur aplikasi berada dalam satu basis kode	*Microservices*: fungsi aplikasi dipisah menjadi layanan independen
Proses *Deployment*	Manual, seluruh aplikasi harus di-*deploy* ulang untuk setiap perubahan	Otomatis melalui *Pipeline build-test-deploy* per layanan (GitLab CI/CD, *Docker Compose*)
Pengujian Rilis	Tidak terintegrasi secara otomatis dalam alur rilis	Pengujian unit dan integrasi berjalan otomatis pada setiap tahap *Pipeline*
Skalabilitas	Sulit diskalakan karena antar-modul saling terikat	Setiap layanan dapat dikembangkan dan diskalakan secara independen
Pemantauan (*Observability*)	Tidak memiliki mekanisme pencatatan dan pemantauan terpusat	Dilengkapi API *Logging* dan pemantauan terpusat (*ELK Stack, Prometheus, Grafana*)
Ketertelusuran Insiden	Sulit melacak sumber masalah karena log tersebar dan tidak terstruktur	*Log* terstruktur dan dapat dikorelasikan dengan peristiwa *deployment* melalui *webhook*
Lingkungan Rilis	Tidak ada pemisahan eksplisit antara staging dan produksi	Mendukung *deployment* terpisah ke lingkungan staging dan produksi melalui *registry kontainer*

Dashboard Utama dan *Dashboard* Penjualan pada gambar 5 masing-masing menampilkan ringkasan kinerja operasional per periode (total penjualan, distribusi saluran, faktor muatan, status armada) dan analisis prediktif penjualan per segmentasi rute beserta tabel transaksi terbaru dan proporsi penjualan per saluran. Modul *Dashboard* Tiket memantau status pemesanan dan validasi tiket secara *real-time*, menjadi antarmuka antara sistem tiket dan *Dashboard* pemantauan untuk transaksi harian dan kapasitas bus.

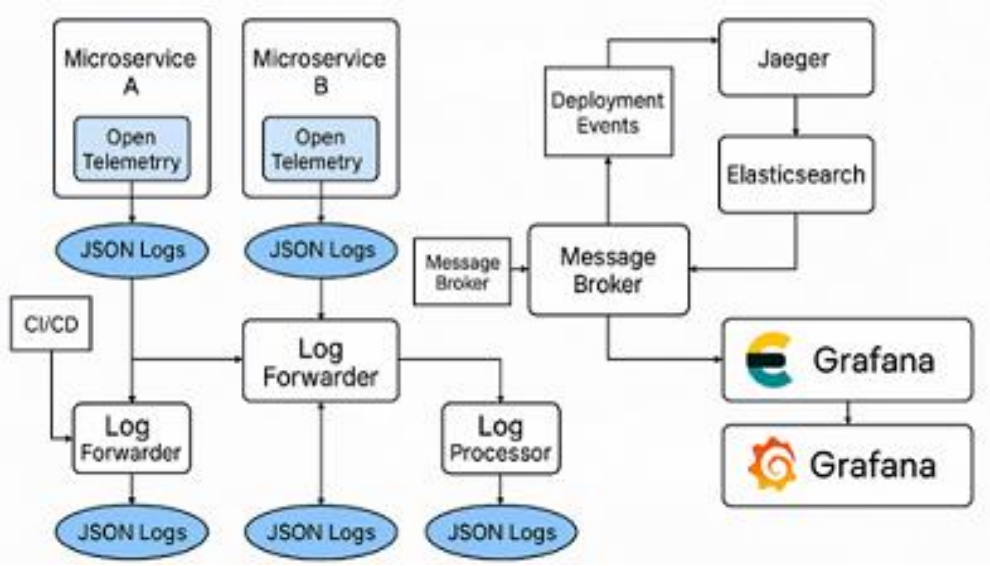


Gambar 5(a). *Dashboard* utama



Gambar 5(b). *Dashboard* penjualan

B. Skema Integrasi Pencatatan API



Gambar 3. *API Logging integration*

Skema pada gambar 6 menunjukkan alur pencatatan API: Setiap layanan *Microservices* menghasilkan *log* terstruktur (format JSON) yang dikirim melalui *log forwarder* (*Fluentd*/*Filebeat*) menuju *Logstash* untuk *parsing* dan pengayaan metadata, sebelum diindeks di *Elasticsearch* dan divisualisasikan melalui *Kibana*. Metrik layanan dikumpulkan *Prometheus* dan divisualisasikan pada *Grafana*, sedangkan penelusuran permintaan lintas layanan menggunakan *OpenTelemetry*.

Integrasi dengan *Pipeline* CI/CD memungkinkan peristiwa *deployment* dikirim ke sistem pencatatan dan pemantauan melalui *webhook*, sehingga rilis dan insiden dapat dikorelasikan secara langsung. Prototipe *Dashboard* dan modul pemantauan yang dihasilkan pada bagian ini dianalisis lebih lanjut sebagai media pelatihan pada bagian Pembahasan.

4. PEMBAHASAN

A. Analisis Hasil Eksperimen Integrasi *LightGBM* dan GA

Analisis Data Eksplorasi dan Rekayasa Fitur mengungkap struktur, anomali, dan pola tersembunyi dalam *dataset* operasional bus. Distribusi variabel target, *target_achieved* (faktor_beban di atas 70%), menunjukkan ketidakseimbangan kelas yang signifikan. Perjalanan yang tidak mencapai target (kelas 0) tercatat 10.015 kasus, sedangkan yang mencapai target (kelas 1) berjumlah 5.467 kasus, rasio sekitar 1,8:1. Ketidakseimbangan ini menjadi dasar penggunaan teknik *oversampling SMOTE* untuk menyeimbangkan data pelatihan dan metrik *F1-Score* untuk evaluasi yang lebih seimbang.

Hasil eksperimen pengembangan model menunjukkan bahwa algoritma *LightGBM*, yang ditingkatkan dengan teknik *oversampling SMOTE* dan penyetelan *hyperparameter*,

mampu menghasilkan model prediksi yang kuat. Pengujian dilakukan pada data uji yang terdiri dari 3.097 baris data.

Tabel 6. Matriks evaluasi

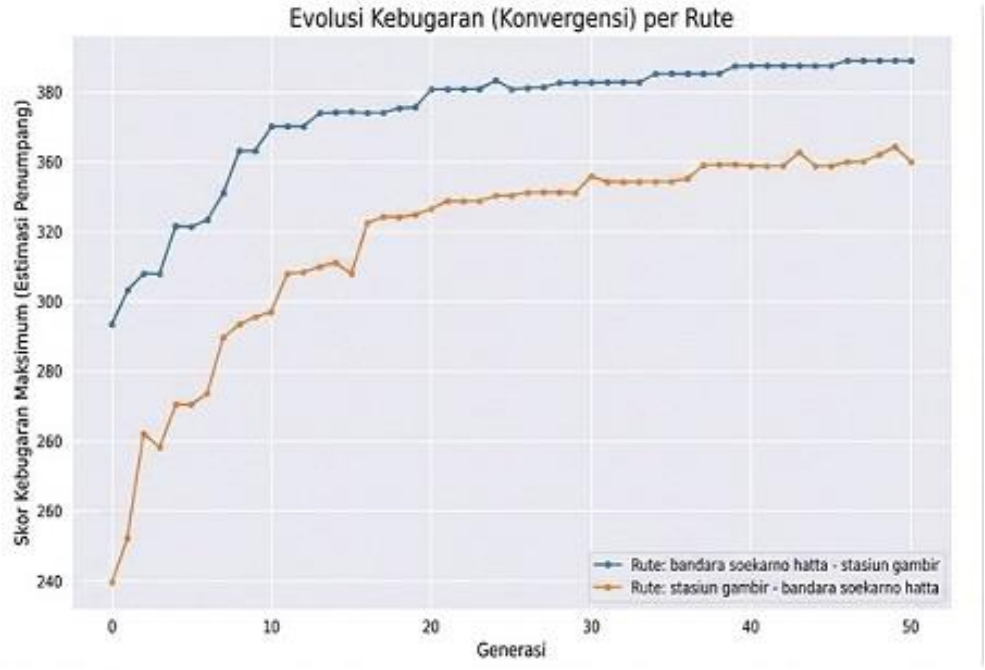
No	Matrik	Presisi	*Recall*	*F1-Score*	*Support*
1	Tidak Tercapai(0)	0.85	0.84	0.85	2003
2	Tercapai(1)	0.72	0.74	0.73	1094
3	Akurasi			0.81	3097
4	*Macro Avg*	0.79	0.79	0.79	3097
5	*Weighted Avg*	0.81	0.81	0.81	3097

Matrik evaluasi lengkap model *LightGBM* disajikan pada Tabel 6. Model ini mencapai akurasi keseluruhan sebesar 80,63%. Untuk kelas target utama, "Tercapai" (kelas 1), model menunjukkan kinerja yang baik dengan *Presisi* 0,72 dan *Recall* 0,74. Nilai *F1 Score* yang seimbang untuk kedua kelas menunjukkan bahwa model mampu memberikan prediksi yang andal tanpa bias berlebihan terhadap kelas mayoritas, membuktikan efektivitas penggunaan *SMOTE*.

Pengujian *Genetic Algorithm* (GA) murni dilakukan sebagai tolok ukur untuk dua rute utama, menunjukkan peningkatan skor *Fitness* yang konsisten (perkiraan total penumpang) pada generasi awal; kurva konvergensi ditunjukkan bersama gambar 7(a). Kurva kemudian mendatar dan konvergen setelah sekitar generasi ke-25, menunjukkan algoritma berhasil menemukan solusi mendekati optimal, membuktikan bahwa proses evolusi (seleksi, *crossover*, dan mutasi) efektif mengarahkan pencarian menuju jadwal yang lebih baik. Terdapat perbedaan yang jelas dalam skor *Fitness* maksimum: rute "Bandara Soekarno-Hatta - Stasiun Gambir" mencapai skor lebih tinggi (sekitar 378) daripada rute sebaliknya (sekitar 354), menunjukkan model menilai potensi permintaan penumpang yang lebih besar menuju Stasiun Gambir.

B. Analisis Hasil Akhir Simulasi *Dashboard*

Pengujian prototipe aplikasi *Dashboard* interaktif sebagai alat pengambilan keputusan bagi perencana jadwal mensimulasikan pembuatan jadwal untuk rute "Bandara Soekarno-Hatta - Stasiun Gambir" pada tanggal 22 Juli 2025. Gambar 7 menyandingkan kurva konvergensi GA dari eksperimen sebelumnya dengan jadwal awal hasil prediksi model untuk simulasi ini.



Gambar 7(a). Konvergensi GA



jam keberangkatan	armada	kapasitas	Load Factor	Rekomendasi
07:00	G-JK-J08	19	100%	Buk
09:00	G-JK-J01	19	100%	Buk
11:00	G-JK-J03	19	100%	Buk
14:00	G-JK-J12	19	100%	Buk
16:00	G-JK-J07	19	100%	Buk
18:00	G-JK-J06	19	100%	Buk
20:00	G-JK-J05	19	100%	Buk
22:00	G-JK-J10	19	100%	Buk
23:00	G-JK-J02	19	100%	Buk
23:00	G-JK-J02	19	100%	Buk

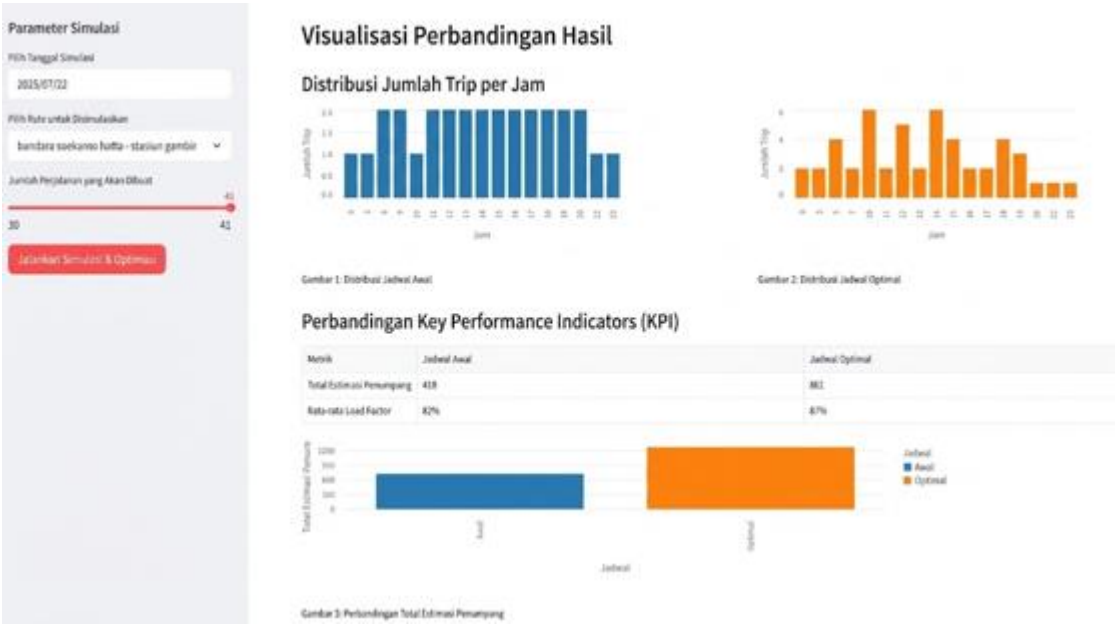
Gambar 7(b). Jadwal awal hasil prediksi model

Selanjutnya, jadwal-jadwal yang menjanjikan dari tahap pertama ini dimasukkan ke Tahap 2: Optimasi dan Penggabungan GA. *Genetic Algorithm* dijalankan untuk menemukan kombinasi jadwal terbaik, dengan jadwal dari tahap awal dimasukkan sebagai benih untuk mempercepat konvergensi. Hasil terbaik dari GA kemudian digabungkan dengan perjalanan-perjalanan terbaik (Faktor Muatan lebih besar dari 70%) dari simulasi awal. Jelas bahwa jadwal akhir jauh lebih ringkas dan efisien, dengan Faktor Muatan seragam sebesar 95% untuk hampir semua perjalanan (Gambar 8).



Gambar 8. Jadwal akhir untuk proses optimalisasi dan penggabungan GA

Halaman visualisasi perbandingan hasil pada gambar 9 menunjukkan distribusi perjalanan per jam. Jadwal optimal (oranye) lebih cerdas mengalokasikan perjalanan, menciptakan klaster pada jam permintaan puncak (misalnya pukul 10:00 pagi dan 1:00 siang), berbeda dari jadwal awal (biru) yang distribusinya lebih acak.



Gambar 9. Visualisasi perbandingan antara jadwal awal dan jadwal optimal

Perbandingan KPI menunjukkan estimasi total penumpang meningkat dari 478 menjadi 841, dan faktor muatan rata-rata meningkat dari 82% menjadi 87% (Tabel 7). Hasil ini memvalidasi bahwa pendekatan hibrida yang diusulkan dapat mengubah rangkaian perjalanan acak menjadi jadwal operasional yang lebih koheren pada skenario simulasi yang diuji.

Tabel 7. Ringkasan perbandingan kinerja sebelum dan sesudah optimasi

Metrik	Sebelum / Awal	Sesudah / Optimal	Perubahan
Akurasi Prediksi *LightGBM*	-	80,63%	-
F1-Score (kelas Tercapai)	-	0,73	-
Estimasi Total Penumpang	478	841	+76,0%
Faktor Muatan Rata-rata	82%	87%	+5 poin persentase

Dari perspektif pendidikan, temuan pelatihan pilot memberikan bukti empiris awal bahwa arsitektur dan alur kerja yang dihasilkan efektif diadaptasi menjadi kerangka pelatihan berbasis proyek. Tabel 8 menunjukkan peningkatan signifikan pada seluruh domain kompetensi yang diujikan (uji *Wilcoxon signed-rank*, $p < 0,001$), dengan *effect size* besar ($r = 0,78–0,94$), mengindikasikan kerangka pembelajaran ini secara praktis efektif meningkatkan kompetensi mahasiswa pada seluruh aspek teknis yang dilatih. Peningkatan tertinggi terjadi pada domain API *Logging* dan *Observability* (rerata 2,45 menjadi 3,50; $r = 0,94$), sedangkan peningkatan terendah terjadi pada domain *Pipeline* CI/CD (rerata 2,50 menjadi 3,45; $r = 0,78$), meski *effect size* nya tetap tergolong besar. Meski demikian, *nonequivalent control group*, sampel yang lebih heterogen, dan durasi pelatihan yang lebih panjang. Uji *Wilcoxon signed-rank* dipakai karena data tidak berdistribusi normal *(Shapiro-Wilk, $p < 0,05$). Effect size* $r = |Z|/\sqrt{N}$.

Tabel 8. Hasil pelatihan pilot: Perbandingan skor kompetensi sebelum dan sesudah pelatihan (n = 22)

Domain Kompetensi	Rerata *Pretest* *Std Deviation*	Rerata *Posttest* *Std Deviation*	*p-value*	*Effect* Size (r)
Arsitektur *Microservices*	2,50 (0,67)	3,45 (0,51)	<0,001	0,83 (besar)
Pipeline CI/CD	2,50 (0,51)	3,45 (0,51)	<0,001	0,78 (besar)
API *Logging* & *Observability*	2,45 (0,51)	3,50 (0,51)	<0,001	0,94 (besar)
Model *LightGBM*	2,41 (0,50)	3,41 (0,50)	<0,001	0,92 (besar)
Genetic Algorithm	2,41 (0,50)	3,32 (0,48)	<0,001	0,91 (besar)
Integrasi End-to-End	2,45 (0,51)	3,41 (0,50)	<0,001	0,93 (besar)

C. Keterbatasan Penelitian dan Relevansi terhadap Pendidikan Teknik Mesin

Penelitian ini masih memiliki sejumlah keterbatasan teknis yang perlu diperhitungkan sebelum diterapkan lebih luas. Pengujian arsitektur baru dilakukan pada skala data dan trafik simulasi, sehingga perilaku sistem pada beban produksi penuh, termasuk penambahan jumlah layanan *Microservices* dan volume permintaan API *Logging*, belum terukur secara empiris. Ada juga risiko keamanan yang belum ditangani: integrasi pencatatan API dan

pemantauan terpusat berpotensi menyimpan data operasional dan transaksi yang sensitif, sehingga penerapan di lapangan membutuhkan mekanisme *masking* data, kontrol akses, dan audit keamanan yang belum menjadi fokus penelitian ini. Biaya komputasi juga menjadi pertimbangan, karena optimasi *Genetic Algorithm* dengan populasi dan jumlah generasi yang lebih besar akan memperlama waktu eksekusi, sehingga penerapan pada skala rute dan armada yang lebih luas perlu mempertimbangkan efisiensi waktu terhadap kebutuhan keputusan operasional yang mendekati *real-time*.

Terkait relevansi dengan pendidikan teknik mesin, penelitian ini secara eksplisit didominasi oleh komponen rekayasa perangkat lunak dan analitik data. Keterkaitannya dengan bidang teknik mesin dapat diperkuat apabila hasil prediksi faktor muat dan pola operasional armada pada penelitian ini dihubungkan dengan pengambilan keputusan pemeliharaan mekanis, misalnya penjadwalan perawatan berkala kendaraan berdasarkan intensitas penggunaan dan rute yang diprediksi memiliki beban tinggi. Arah pengembangan tersebut belum dieksplorasi pada penelitian ini dan menjadi salah satu peluang penelitian lanjutan agar kerangka yang diusulkan lebih selaras dengan kompetensi vokasi teknik mesin, di samping kompetensi rekayasa perangkat lunak yang sudah menjadi fokus utama saat ini.

5. KESIMPULAN

Penelitian ini menunjukkan bahwa integrasi arsitektur *microservices* dengan *pipeline* CI/CD, API *logging*, dan pemantauan terpusat berbasis *ELK stack* serta *prometheus-grafana* mendukung pengembangan dan *deployment* sistem tiket bus secara lebih cepat, terkontrol, dan dapat diamati. Pada sisi optimasi jadwal, model *LightGBM* memprediksi faktor muat penumpang dengan akurasi 80,63% dan F1-*score* 0,73, dan prediksi ini digunakan sebagai fungsi *fitness* pada *genetic algorithm* sehingga total estimasi penumpang meningkat dari 478 menjadi 841 dan faktor muat rata-rata meningkat dari 82% menjadi 87% pada skenario simulasi yang diuji.

Selain kontribusi teknis tersebut, penelitian ini menghasilkan kerangka pembelajaran *DevOps* berbasis industri (*industry-based devops learning framework*) yang divalidasi secara empiris melalui pelatihan pilot terhadap 22 mahasiswa vokasi, menunjukkan peningkatan kompetensi yang signifikan (uji *wilcoxon signed-rank*, $p < 0{,}001$) dengan *effect size* besar ($r = 0{,}78–0{,}94$) pada seluruh enam domain yang diujikan. Penelitian lanjutan perlu mereplikasi pelatihan ini pada sampel yang lebih besar dan heterogen dengan kelompok pembanding, sekaligus mengkaji keterkaitannya dengan kompetensi teknik mesin seperti pemeliharaan armada berbasis data operasional.

6. UCAPAN TERIMA KASIH DAN PERNYATAAN PENDANAAN

Terima kasih kami sampaikan kepada Universitas Logistik dan Bisnis Internasional (ULBI) atas dukungan serta pendanaan yang diberikan dalam penelitian ini.

7. PERNYATAAN PENGGUNAAN KECERDASAN BUATAN (AI)

Dalam penyusunan artikel ini, penulis menggunakan chatgpt untuk membantu proses pharafrase. Penulis telah meninjau dan memverifikasi seluruh keluaran AI serta bertanggung jawab penuh atas isi, keakuratan, dan integritas artikel.

8. KONTRIBUSI PENULIS

Penulis 1 memimpin dalam pengembangan konsep penelitian, merumuskan ide penelitian, merancang metodologi, dan memberikan supervisi penelitian. Penulis 2 melakukan pengumpulan dan analisis data, serta menyusun naskah awal dan melakukan interpretasi hasil. Penulis 3 melakukan validasi data, meninjau dan menyunting naskah. Seluruh penulis telah membaca dan menyetujui versi akhir naskah serta bertanggung jawab atas isi artikel.

9. KONFLIK KEPENTINGAN

Penulis menyatakan bahwa tidak terdapat konflik kepentingan yang berkaitan dengan penelitian, penulisan, maupun publikasi artikel ini.

REFERENSI

Adjie, Di., & Zakaria, H. (2026). Implementasi metode design science research (DSR) pada monitoring server voip berbasis web dengan pendekatan agile untuk optimalisasi kinerja sistem (studi kasus: PT. XYZ). *Jurnal Fasilkom*, *16*(1), 175–182. https://doi.org/https://doi.org/10.37859/jf.v16i1.11071

Arief, M., & Chaniago, A. U. (2025). Penerapan arsitektur microservices pada web e-commerce menggunakan metode GRPC. *Cloud*, *2*(1), 1-9.

Amrullah, M. S. A., & Marthasari, G. I. (2024). Otomatisasi proses deployment dengan metode CI/CD menggunakan jenkins dan docker pada web service i-lab. *Jurnal Repositor*, *6*(4), 313-322.

Ariska, V., Fadilla, T. A. N., & Agustin, T. (2023). Sistem informasi pemesanan tiket bus online berbasis website pada PO. Haryanto. *Jurnal Penelitian Sistem Informasi (Jpsi)*, *1*(3), 107-117. https://doi.org/10.54066/jpsi.v1i3.678

Bakti, K.W., Jati Setyadi, H., & Labib Jundillah, M. (2026). Perancangan dan pengembangan infrastruktur continuous integration / continuos deployment

menggunakan jenkins dan docker. *Jurnal Tekno Kompak*, *20*(1), 113–130. https://doi.org/https://doi.org/10.33365/jtk.v20i1.602

Dwiputra, H. K., Wibowo, A., & Proboyekti, U. (2024). Penerapan web api untuk sentralisasi data sistem pembukuan. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *6*(1), 1–10. https://doi.org/10.47233/jteksis.v6i1.1073

Haryanti, T., Rakhmawati, N. A., Subriadi, A. P., & Tjahyanto, A. (2022, November 5). The design science research methodology (DSRM) for self-assessing digital transformation maturity index in Indonesia. *IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)* (pp. 1-7). IEEE.

Herawati, D., & Yenita, Y. (2023). Pelayanan perusahaan otobus sinar jaya terhadap kepuasan dan loyalitas penumpang. *Widya Cipta: Jurnal Sekretari Dan Manajemen*, *7*(1), 59–64. https://doi.org/10.31294/widyacipta.v7i1.15103

Lubis, F. A., Irwan, D., & Lubis, I. (2024). Sistem informasi untuk menentukan kelayakan operasional bus pada PT. ALS (antar lintas sumatera) menggunakan metode smart berbasis web. *CompTech: Jurnal Ilmu Komputer dan Teknologi*, *1*(1), 89-103. https://doi.org/https://doi.org/10.63854/comptech.v1i1.29

Manik, D. S., & Panjaitan, N. (2025, July). Literatur review: Penggunaan algoritma machine learning untuk optimalisasi rantai pasokan. In *Talenta Conference Series: Energy and Engineering (EE)*, *8*(1), 1038-1047.

Minarni, M., Daud, M. I., Yulianti, E., Swara, G. Y., & Putra, E. K. (2024). Perancangan sistem informasi pemesanan tiket bus berbasis website. *Jurnal Minfo Polgan*, *13*(2), 1302–1315. https://doi.org/10.33395/jmp.v13i2.13987

Mardikawati, B., Masyuni, I. A., & Nugraha, A. E. (2023). Kenyamanan sebagai faktor dominan pemilihan moda transportasi hasil kuesioner penumpang bus antar propinsi dari terminal Mengwi Bali. *Jurnal Teknologi Transportasi Dan Logistik*, *4*(1). https://doi.org/https://doi.org/10.52920/jttl.v4i1.129

Mutmainah, S., Lorosae, T. A., & Citra, E. E. (2025). Genetic algorithm optimization for solving the traveling salesman problem in the indonesian business environment. *Journix: Journal of Informatics and Computing*, *1*(2), 88–98. https://doi.org/10.63866/journix.v1i2.14

Nurzakiyah, A. (2025). Implementasi model project-based learning (PjBL) sebagai pembelajaran inovatif di Sekolah. *Jurnal Bima : Pusat Publikasi Ilmu Pendidikan Bahasa Dan Sastra*, *3*(4), 94–103. https://doi.org/10.61132/bima.v3i4.2553

Prianto, C., Suarsa, S. H., Permadi, D., Anggraeni, A. D., Guslan, D., & Fatmawati, A. P. (2026). Pengembangan smart market positioning system berbasis machine learning dan big data untuk analisis posisi pasar (studi kasus: perusahaan otobus XYZ). *Simposium Nasional Teknologi Infrastruktur,* 1-10.

Putri, M. Z., Shafiq, M. F., Rafie, R. A., Arribe, E., Informasi, S., Muhammadiyah, U., Jl, R., Tambusai, T., Pekanbaru, K., & -Indonesia, R. (2025). Perancangan sistem

informasi pemesanan tiket bus berbasis website pada PT. Fajar Riau Wisata. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(6), 10024-10031.

Railić, N., & Savić, M. (2021, March). Architecting continuous integration and continuous deployment for microservice architecture. In *2021 20th International Symposium Infoteh-Jahorina (INFOTEH)* (pp. 1-5). IEEE.

Sugiyanto, Wayan Arnaya, I., Sylvan Ryanto, S., & Oka Khrisna Surya, Aab. (2021). Analisa faktor pemilihan moda transportasi menggunakan metode analytic hierarchy process. *Jurnal Teknologi Transportasi Dan Logistik*, *2*(1), 11–18. https://doi.org/https://doi.org/10.52920/jttl.v2i`1.18

Suminten, Saridawati, & Setyawati, N. A. (2025). Perancangan sistem informasi penjualan tiket bus online pada PO garuda mas Blora berbasis web. *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)*, *8*(3), 1652–1657. https://doi.org/https://doi.org/10.32672/jnkti.v8i3.9232

Zuhri, S., Syahriza, & Syah Putra, A. (2022). Evaluasi sistem pemesanan tiket bus menggunakan business process reengineering. *Jurnal Komputer, Informasi Teknologi, Dan Elektro*, *7*(3), 2022. https://doi.org/https://doi.org/10.24815/kitektro.v7i3.31616