

Sistem Informasi Geografis Berbasis Web untuk Manajemen Fleet Management dan Monitoring Bahan Bakar Terintegrasi Sensor IoT

¹Muhammad Ridwan *, ²Ghufron, ³Bagus Satrio Waluyo,

^{1,2,3} Teknik Informatika, Universitas Islam Sultan Agung

*Corresponding Author:

32602200104@std.unissula.ac.id

Abstrak

Manajemen armada logistik konvensional sering menghadapi kendala seperti akuntabilitas rendah dalam pencatatan perjalanan dan inefisiensi dalam pemantauan bahan bakar. Penelitian ini bertujuan untuk mengembangkan Cartrack IMB, Sistem Informasi Geografis (SIG) berbasis Internet of Things (IoT) untuk pemantauan armada secara real-time. Sistem ini menggunakan mikrokontroler ESP32 sebagai unit pelacak yang mengintegrasikan modul GPS dan sensor bahan bakar ultrasonik. Dari sisi perangkat lunak, arsitektur tiga tingkat dibangun menggunakan bahasa pemrograman Go untuk backend dan React dengan pustaka Leaflet.js untuk frontend. Inovasi utama sistem ini terletak pada otomatisasi siklus hidup perjalanan menggunakan algoritma Ray Casting untuk deteksi geofence poligon dan rumus Haversine untuk area melingkar. Hasil pengujian menunjukkan bahwa algoritma Ray Casting mencapai tingkat akurasi 100% dalam mendeteksi transisi status perjalanan otomatis dari titik pengambilan ke titik pengantaran. Lebih lanjut, pengujian sensor bahan bakar menunjukkan tingkat presisi yang stabil dengan tingkat kesalahan rata-rata 2-3%. Implementasi protokol WebSockets memastikan distribusi data telemetri yang responsif dengan latensi rendah. Secara keseluruhan, sistem ini berhasil mendigitalisasi pengawasan armada, meningkatkan transparansi operasional dan akuntabilitas manajemen logistik.

Kata Kunci: Sistem Manajemen Armada, IoT, ESP32, Geofencing, Ray Casting, GIS.

Abstract

Conventional logistics fleet management often faces obstacles such as low accountability in trip recording and inefficiency in fuel monitoring. This research aims to develop Cartrack IMB, an Internet of Things (IoT)-based Geographic Information System (GIS) for real-time fleet monitoring. The system utilizes the ESP32 microcontroller as a tracking unit that integrates a GPS module and an ultrasonic fuel sensor. On the software side, a three-tier architecture is built using the Go programming language for the backend and React with the Leaflet.js library for the frontend. The main innovation of this system lies in the automation of the trip lifecycle using the Ray Casting algorithm for polygon geofence detection and the Haversine formula for circular areas. Test results indicate that the Ray Casting algorithm achieves a 100% accuracy rate in detecting automatic trip status transitions from pickup points to dropoff points. Furthermore, fuel sensor testing shows a stable precision level with an average error rate of 2-3%. The implementation of the WebSockets protocol ensures responsive telemetry data

distribution with low latency. Overall, this system successfully digitalizes fleet supervision, enhancing operational transparency and logistics management accountability.

Kata Kunci: *Fleet Management System, IoT, ESP32, Geofencing, Ray Casting, GIS.*

1. PENDAHULUAN

Transformasi digital dalam sektor transportasi dan logistik global saat ini sangat bergantung pada integrasi *Internet of Things* (IoT) dan Sistem Informasi Geografis (SIG). Keberhasilan manajemen armada modern ditentukan oleh presisi data spasial dan kecepatan distribusi informasi secara *real-time* untuk mendukung pengambilan keputusan yang akurat [1]. Teknologi SIG telah terbukti efektif dalam mengoptimalkan pemetaan infrastruktur dan efisiensi rute perjalanan guna mendukung ekosistem industri yang lebih cerdas [2].

Namun, pada sektor industri berat seperti pertambangan dan konstruksi, manajemen armada *dump truck* masih menghadapi tantangan operasional yang signifikan. Masalah utama yang sering muncul adalah rendahnya akurasi pemantauan posisi kendaraan, proses pencatatan perjalanan (*ritase*) yang masih bersifat manual, serta lemahnya sistem akuntabilitas pengemudi (Syaputra dan Anaperta, 2023). Tanpa adanya pengawasan geografis otomatis, risiko kendaraan keluar dari zona operasional tanpa terdeteksi menjadi sangat tinggi, yang pada akhirnya memicu inefisiensi biaya operasional [3].

Berbagai penelitian terdahulu telah berupaya menyelesaikan kendala ini melalui inovasi teknologi. Implementasi pelacakan kendaraan berbasis IoT telah dikembangkan untuk memantau lokasi serta level bahan bakar guna meminimalisir risiko pencurian [4]. Selain itu, terdapat studi yang memfokuskan pada optimasi lokasi infrastruktur menggunakan metode pengambilan keputusan multikriteria [5]. Meskipun demikian, sebagian besar sistem yang tersedia saat ini masih bersifat parsial; fokus pada pelacakan posisi saja atau pemantauan sensor tertentu tanpa adanya integrasi otomatisasi siklus perjalanan (*trip lifecycle*) yang mandiri [6].

Kesenjangan penelitian (*research gap*) tersebut menjadi dasar bagi pengembangan sistem Cartrack IMB. Berbeda dengan solusi manajemen armada konvensional, sistem ini menawarkan inovasi berupa *Continuous Trip Management* yang mengotomatisasi perubahan status perjalanan—mulai dari titik *pickup* hingga *dropoff*—berdasarkan deteksi wilayah geografis secara mandiri. Dengan memanfaatkan mikrokontroler ESP32 sebagai unit IoT dan bahasa pemrograman Go pada sisi *backend*, sistem ini mampu mendistribusikan data koordinat serta sensor bahan bakar dengan latensi rendah melalui protokol *WebSockets* [7].

Aspek teknis yang menjadi keunggulan dalam penelitian ini adalah penerapan algoritma *Ray Casting* untuk pengawasan *geofencing* berbentuk poligon yang presisi, serta penggunaan rumus *Haversine* untuk area operasional berbentuk lingkaran [8]. Integrasi ini memungkinkan pihak manajemen untuk memberlakukan batas wilayah kerja secara ketat bagi armada truk. Selain itu, sistem monitoring bahan bakar yang terintegrasi memberikan lapisan transparansi tambahan terhadap biaya operasional kendaraan [6].

Penelitian ini bertujuan untuk merancang dan mengimplementasikan Sistem Informasi Geografis berbasis web yang komprehensif untuk optimalisasi manajemen armada dan pemantauan konsumsi bahan bakar secara terintegrasi. Melalui implementasi sistem ini, diharapkan perusahaan dapat meminimalisir kesalahan pencatatan manual, meningkatkan produktivitas ritase, serta memiliki basis data analitik yang kuat untuk mendukung efisiensi operasional logistik secara berkelanjutan.

2. TINJAUAN PUSTAKA

A. Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) didefinisikan sebagai sistem berbasis komputer yang dirancang untuk mengumpulkan, menyimpan, memanipulasi, menganalisis, mengelola, dan menyajikan seluruh jenis data geografis atau spasial. Kekuatan utama SIG terletak pada kemampuannya untuk menghubungkan informasi tekstual (atribut) dengan lokasi geografis tertentu di permukaan bumi [1]. Dalam implementasi modern, SIG tidak hanya berfungsi sebagai alat pemetaan statis, tetapi telah berkembang menjadi sistem pendukung keputusan yang dinamis melalui integrasi analisis spasial yang kompleks [2].

Komponen utama SIG terdiri dari lima elemen krusial, yaitu perangkat keras (*hardware*), perangkat lunak (*software*), data, manusia (*people*), dan metode. Dalam konteks manajemen transportasi, SIG berperan sebagai infrastruktur visual yang memungkinkan operator untuk memahami pola distribusi armada dan hambatan logistik secara spasial [9]. Penggunaan pustaka pemetaan digital seperti Leaflet.js dan integrasi API peta global telah mempermudah pengembangan aplikasi SIG berbasis web yang interaktif dan dapat diakses secara *real-time* oleh pengguna [10].

Analisis spasial dalam SIG, khususnya yang melibatkan data vektor seperti titik (*point*) dan poligon, merupakan fondasi dari fitur-fitur canggih seperti pelacakan posisi dan pembatasan wilayah operasional [8]. Dengan kemampuan analisis ini, SIG mampu mengubah koordinat GPS mentah menjadi informasi yang bernilai strategis bagi manajemen perusahaan dalam mengoptimalkan produktivitas armada [3].

B. Fleet Management System (FMS)

Fleet Management System (FMS) atau sistem manajemen armada adalah sebuah platform terintegrasi yang memungkinkan perusahaan untuk mengelola dan mengoptimalkan operasional kendaraan mereka secara efisien. Tujuan utama dari FMS adalah untuk meningkatkan produktivitas, mengurangi biaya operasional, serta memastikan keselamatan dan kepatuhan pengemudi [11]. Dalam konteks industri logistik dan pertambangan, FMS sangat krusial untuk memantau ritase kendaraan, penggunaan bahan bakar, dan durasi perjalanan guna meminimalisir waktu tunggu (*idle time*) [6].

Implementasi FMS yang efektif sangat bergantung pada akurasi data yang dikirimkan oleh unit pelacak (*GPS tracker*). Data tersebut memungkinkan pihak manajemen untuk melakukan evaluasi kinerja pengemudi dan ketepatan waktu distribusi secara akurat [12]. Selain itu, sistem manajemen armada modern kini telah mengintegrasikan modul analitik yang dapat mendeteksi pola penggunaan kendaraan dan memprediksi kebutuhan pemeliharaan secara preventif [1].

Pada sistem Cartrack IMB, konsep FMS diperluas dengan fitur *Continuous Trip Management*. Fitur ini melampaui pelacakan posisi sederhana dengan menyematkan logika bisnis otomatis yang memantau setiap tahapan perjalanan truk, mulai dari pemuatan barang (*pickup*) hingga pembongkaran (*dropoff*). Hal ini bertujuan untuk menciptakan sistem manajemen yang mandiri dan meminimalisir campur tangan manusia dalam proses pencatatan data operasional [3].

C. *Internet of Things (IoT) dan ESP32*

Internet of Things (IoT) merupakan sebuah paradigma di mana objek fisik atau "benda" dilengkapi dengan sensor, perangkat lunak, dan teknologi lainnya dengan tujuan untuk menghubungkan dan bertukar data dengan perangkat atau sistem lain melalui internet. Dalam ekosistem pelacakan kendaraan, IoT memungkinkan pengumpulan data telemetri secara terus-menerus, seperti posisi geografis, kecepatan, dan kondisi mekanis kendaraan [4]. Integrasi IoT dalam manajemen armada tidak hanya meningkatkan visibilitas aset, tetapi juga memungkinkan adanya mekanisme peringatan dini (*alert system*) terhadap anomali operasional [6].

Salah satu perangkat keras yang paling efektif dan populer dalam implementasi IoT saat ini adalah mikrokontroler ESP32. ESP32 merupakan *System on a Chip* (SoC) yang memiliki performa tinggi dengan fitur Wi-Fi dan *dual-mode Bluetooth* terintegrasi. Penggunaan ESP32 dalam pemantauan bahan bakar dan pelacakan kendaraan dipilih karena efisiensi biayanya, konsumsi daya yang rendah, serta kemampuan pemrosesan yang mumpuni untuk menangani data dari berbagai sensor secara simultan [13].

Dalam sistem Cartrack IMB, ESP32 berfungsi sebagai pusat pemrosesan data di sisi kendaraan (*edge device*). Perangkat ini bertugas mengumpulkan koordinat dari modul GPS dan data level cairan dari sensor bahan bakar, kemudian mengirimkan informasi tersebut ke *backend* melalui protokol HTTP POST. Kemampuan ESP32 dalam menangani tugas-tugas *multitasking* memastikan bahwa transmisi data tetap stabil meskipun sistem sedang melakukan pembacaan sensor yang intensif [4].

D. *Geofencing dan Algoritma Analisis Spasial*

Geofencing adalah teknologi yang menciptakan perimeter virtual atau batas geografis di dunia nyata menggunakan koordinat GPS. Dalam sistem pelacakan armada, *geofencing* berfungsi sebagai pemicu otomatis (*trigger*) ketika kendaraan memasuki atau meninggalkan area tertentu, seperti zona pemuatan (*pickup*) atau pembongkaran (*dropoff*) [8]. Penggunaan *geofencing* sangat efektif untuk meningkatkan pengawasan operasional dan memastikan kendaraan tetap berada pada jalur yang telah ditentukan [1].

Untuk menentukan posisi kendaraan terhadap batas *geofence* secara akurat, diperlukan algoritma analisis spasial. Dalam penelitian ini, digunakan dua pendekatan utama:

1) *Algoritma Ray Casting*: Digunakan untuk deteksi area berbentuk poligon yang kompleks. Algoritma ini bekerja dengan cara menarik garis imajiner (*ray*) dari titik lokasi kendaraan ke arah tertentu; jika garis tersebut memotong sisi poligon dalam jumlah ganjil, maka titik tersebut berada di dalam area [8]. Metode ini memberikan fleksibilitas tinggi

bagi perusahaan untuk mendefinisikan batas wilayah kerja yang tidak beraturan di lapangan [11].

2) *Rumus Haversine*: Digunakan untuk deteksi area berbentuk lingkaran (*circle geofence*) dan perhitungan jarak antara dua koordinat di permukaan bumi. Rumus ini mempertimbangkan kelengkungan bumi sehingga memberikan hasil perhitungan jarak yang presisi bagi sistem monitoring armada [8].

Integrasi kedua algoritma ini dalam sistem Cartrack IMB memungkinkan adanya otomatisasi siklus perjalanan (*trip lifecycle*). Ketika koordinat kendaraan terdeteksi masuk ke dalam poligon area *pickup* melalui pengecekan *Ray Casting*, sistem secara otomatis mengubah status perjalanan menjadi *In Progress*. Sebaliknya, status akan berubah menjadi *Completed* saat kendaraan masuk ke area *dropoff* [10].

E. Arsitektur Web dan Komunikasi Data Real-Time

Pengembangan Sistem Informasi Geografis (SIG) berbasis web memerlukan arsitektur yang mampu menangani beban data spasial yang besar dengan responsivitas tinggi. Penggunaan arsitektur *Three-Tier* yang memisahkan antara *Presentation Layer (Frontend)*, *Application Layer (Backend)*, dan *Data Layer (Database)* menjadi standar dalam memastikan skalabilitas dan kemudahan pemeliharaan sistem [14]. Dalam sistem Cartrack IMB, efisiensi pada sisi *backend* dicapai melalui penggunaan bahasa pemrograman Go (Golang) yang memiliki performa tinggi dalam menangani proses *concurrency*, yang sangat krusial saat mengelola ribuan log koordinat dari banyak armada sekaligus [14].

Salah satu tantangan utama dalam sistem pelacakan armada adalah penyajian data lokasi secara instan tanpa perlu melakukan penyegaran halaman (*page refresh*). Untuk mengatasi hal ini, teknologi *WebSockets* digunakan sebagai protokol komunikasi dua arah yang persisten antara server dan klien. Berbeda dengan protokol HTTP konvensional yang bersifat *request-response*, *WebSockets* memungkinkan server untuk secara aktif mengirimkan (*push*) data koordinat terbaru ke *dashboard* sesaat setelah data diterima dari perangkat IoT [11]. Hal ini secara signifikan mengurangi latensi dan penggunaan *bandwidth*, sehingga operator dapat melihat pergerakan truk di peta secara halus dan *real-time* [7].

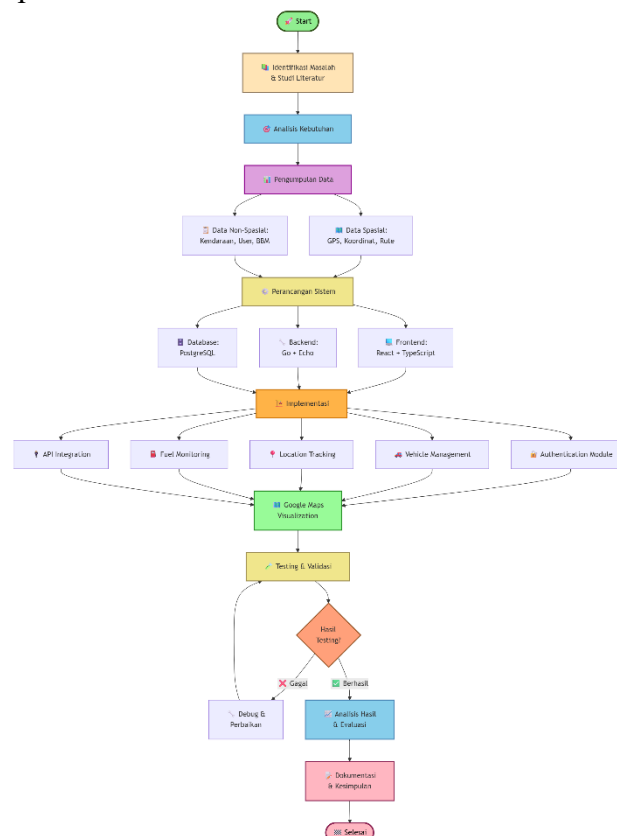
Di sisi antarmuka pengguna, penggunaan kerangka kerja modern seperti React yang dikombinasikan dengan pustaka pemetaan Leaflet.js memberikan pengalaman pengguna yang interaktif. Leaflet.js memungkinkan visualisasi data spasial seperti penanda kendaraan (*markers*), garis lintasan (*polylines*), dan area *geofence* (poligon) dirender secara efisien di atas *base map* dari Google Maps API [10]. Integrasi ini memastikan bahwa informasi operasional yang kompleks dapat disajikan dalam bentuk visual yang mudah dipahami oleh pengguna [9].

3. METODOLOGI PENELITIAN

A. Tahapan Penelitian

Proses penelitian dan pengembangan sistem Cartrack IMB dilaksanakan melalui pendekatan sistematis yang terbagi ke dalam lima tahapan utama guna memastikan solusi

yang dihasilkan tepat sasaran dan terukur [1]. Tahapan pertama dimulai dengan Analisis Kebutuhan, di mana peneliti melakukan identifikasi mendalam terhadap masalah operasional pada armada *dump truck*, khususnya terkait inefisiensi pencatatan ritase manual dan rendahnya akuntabilitas pengawasan area kerja [11]. Setelah kebutuhan sistem terdefinisi, tahap kedua adalah Perancangan Sistem (*Design*) yang melibatkan penyusunan arsitektur *three-tier*, perancangan skema basis data PostgreSQL, serta desain antarmuka *dashboard* interaktif [14]. Memasuki tahap ketiga, yaitu Implementasi, dilakukan pengodean program pada sisi *backend* menggunakan bahasa Go dan sisi *frontend* menggunakan React, dibarengi dengan konfigurasi perangkat keras ESP32 sebagai unit pelacak IoT [13]. Tahap keempat difokuskan pada Integrasi Algoritma Spasial, di mana logika *Ray Casting* dan rumus *Haversine* disematkan ke dalam sistem untuk mengaktifkan fungsi *geofencing* poligon dan otomatisasi transisi status perjalanan [8]. Tahapan akhir adalah Pengujian dan Validasi, yang dilakukan melalui simulasi data lokasi serta pengujian fungsionalitas sensor untuk memastikan bahwa sistem mampu menjalankan siklus perjalanan otomatis secara akurat. Tahapan alur penelitian ini secara ringkas ditunjukkan pada Gambar 1.



Gambar 1. Flowchart Penelitian

A. Arsitektur Sistem

Arsitektur sistem pada penelitian ini dirancang menggunakan pendekatan *three-tier* yang memisahkan antara lapisan perangkat keras (*IoT Edge*), lapisan server (*Backend*), dan lapisan antarmuka pengguna (*Frontend*). Struktur ini dipilih untuk memastikan

skalabilitas sistem dalam menangani data koordinat dari banyak armada secara simultan tanpa mengganggu performa keseluruhan [14]. Aliran data dimulai dari perangkat keras yang terpasang pada kendaraan menuju server melalui jaringan internet, yang kemudian didistribusikan ke dasbor web secara instan.

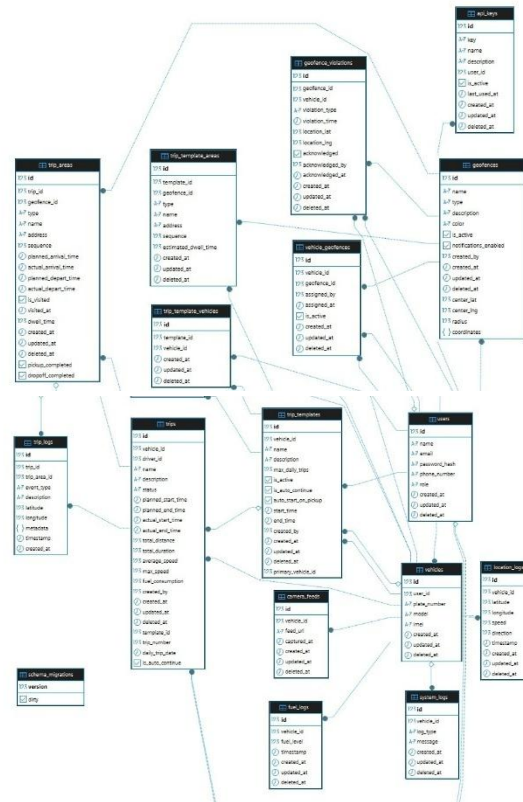
1) *Lapisan Perangkat Keras (IoT Edge Device)*: Pada lapisan ini, mikrokontroler ESP32 bertugas sebagai unit pemroses utama yang terintegrasi dengan modul GPS Neo-6M untuk mendapatkan data spasial (garis lintang dan garis bujur) serta sensor bahan bakar ultrasonik untuk memantau volume tangki secara *non-contact* [13]. Data telemetri mentah yang dikumpulkan oleh ESP32 dikonversi menjadi format JSON dan dikirimkan ke server menggunakan protokol HTTP POST. Keamanan pengiriman data dijamin melalui penggunaan *API Key* unik yang disertakan dalam setiap header permintaan dari perangkat keras ke server [4].

2) *Lapisan Server (Backend & Database)*: Sisi backend dikembangkan menggunakan bahasa pemrograman Go dengan *Clean Architecture* untuk memisahkan logika bisnis dari infrastruktur eksternal. Server ini mengelola tugas-tugas krusial seperti validasi data, penyimpanan log lokasi ke basis data PostgreSQL, dan pemicuan analisis spasial untuk deteksi pelanggaran wilayah [13]. Selain itu, *backend* mengimplementasikan *WebSocket Hub* yang secara otomatis menyiarkan (*broadcast*) data koordinat terbaru ke seluruh klien yang terhubung segera setelah data diterima dari lapisan perangkat keras [7].

3) *Lapisan Antarmuka (Frontend GIS Dashboard)*: Lapisan teratas merupakan aplikasi berbasis web yang dibangun menggunakan *framework* React. Lapisan ini bertanggung jawab untuk memvisualisasikan data spasial pada peta interaktif menggunakan pustaka Leaflet.js yang terintegrasi dengan Google Maps API [10]. Komponen *frontend* mendengarkan pembaruan data dari *WebSocket* untuk memperbarui posisi penanda (*marker*) kendaraan di peta secara halus tanpa perlu melakukan muat ulang halaman. Selain visualisasi pelacakan, lapisan ini juga menyajikan grafik analitik konsumsi bahan bakar dan status perjalanan setiap armada [9].

B. Perancangan Basis Data

Penyimpanan data dalam sistem Cartrack IMB menggunakan basis data relasional PostgreSQL yang dioptimalkan untuk menangani data telemetri bervolume tinggi. Perancangan basis data ini berfokus pada efisiensi kueri untuk mendukung visualisasi *real-time* dan otomatisasi logika bisnis perjalanan [13]. Struktur entitas dirancang untuk mengintegrasikan identitas armada dengan log aktivitas sensor serta aturan wilayah geografis guna menciptakan sistem pemantauan yang komprehensif [14]. Rancangan hubungan antar entitas secara detail ditunjukkan pada Gambar 2.



Gambar 2. Entity Relationship Diagram (ERD) Sistem Cartrack IMB

Berdasarkan struktur pada Gambar 2, basis data sistem terdiri dari entitas-entitas utama sebagai berikut:

1) *Kendaraan (Vehicles)*: Tabel ini menyimpan data statis dan status terkini dari setiap armada, termasuk nomor plat (*license plate*) dan nama kendaraan. Atribut *last_latitude* dan *last_longitude* pada tabel ini berfungsi untuk menyimpan posisi terakhir kendaraan guna mempercepat proses rendering pada dasbor peta tanpa harus memproses seluruh riwayat log [10].

2) *Log Lokasi (Location Logs)*: Merupakan tabel transaksional yang mencatat riwayat pergerakan kendaraan secara kontinu. Selain menyimpan koordinat geografis, tabel ini mengintegrasikan data sensor yang dikirimkan oleh ESP32, seperti kecepatan (*speed*) dan level bahan bakar (*fuel level*), dalam satu entri waktu (*timestamp*) yang sinkron [4], [13].

3) *Wilayah Virtual (Geofences)*: Tabel ini menyimpan definisi batas wilayah geografis. Sesuai dengan rancangan sistem, entitas ini memiliki atribut *type* yang mengklasifikasikan area menjadi tiga kategori: *pickup* (titik muat), *dropoff* (titik bongkar), dan *corridor* (jalur lintasan). Data koordinat poligon disimpan dalam format JSON untuk kemudian diproses menggunakan algoritma *Ray Casting* [8].

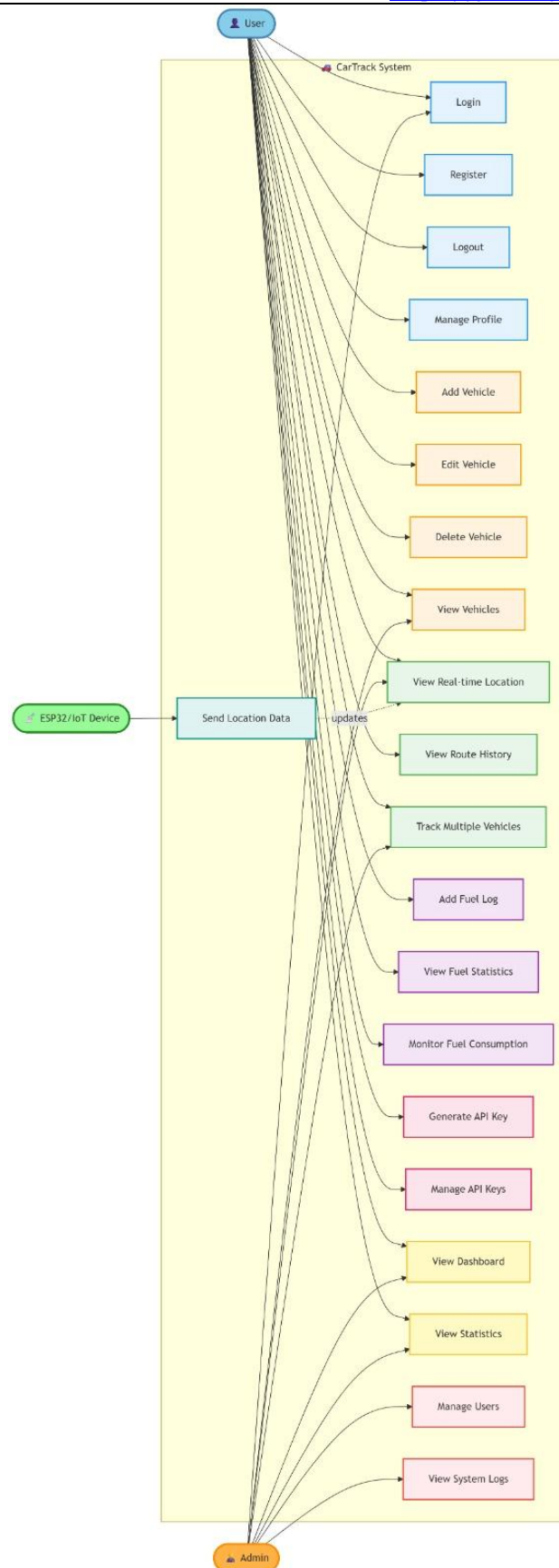
4) *Perjalanan (Trips)*: Entitas ini merekam siklus perjalanan otomatis kendaraan. Tabel ini mencatat waktu mulai (*pickup time*), waktu selesai (*dropoff time*), serta hasil kalkulasi otomatis berupa jarak tempuh (*distance*) dan total konsumsi bahan bakar (*fuel consumed*) selama satu siklus ritase. Perubahan status dari *Planned* menjadi *In Progress*

hingga *Completed* terjadi secara otomatis berdasarkan interaksi kendaraan dengan wilayah *geofence* [3].

C. Perancangan Fungsional dan Proses

Perancangan fungsional mendefinisikan interaksi antara aktor dengan sistem serta aliran data internal yang terjadi guna memastikan setiap fitur berjalan secara terintegrasi [14].

1) *Use Case Diagram*: *Use Case Diagram* menggambarkan fungsionalitas sistem dari perspektif aktor utama, yaitu Perangkat IoT (ESP32) dan Admin. Perangkat IoT memiliki fungsi utama untuk mengirimkan data koordinat dan level bahan bakar secara kontinu ke sistem [4]. Di sisi lain, Admin memiliki otoritas penuh untuk mengelola data master kendaraan, mendefinisikan koordinat wilayah *geofence*, serta memantau laporan analitik perjalanan dan konsumsi bahan bakar yang diproses secara otomatis oleh sistem [11]. Interaksi fungsional antara aktor dan sistem ini secara detail ditunjukkan pada Gambar 3.



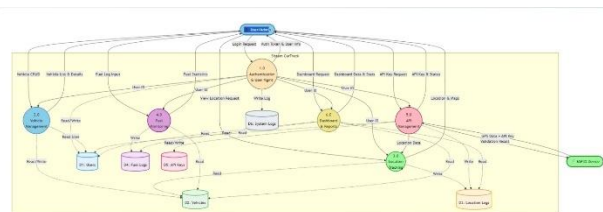
Gambar 3. Use Case Diagram Sistem Cartrack IMB

2) *Data Flow Diagram (DFD) Level 0*: DFD Level 0 atau Diagram Konteks menggambarkan sistem secara global sebagai satu proses besar yang menghubungkan entitas eksternal dengan sistem Cartrack IMB. Aliran data pada tahap ini menunjukkan entitas Perangkat IoT yang mengirimkan masukan mentah berupa data GPS dan sensor ke dalam sistem, sementara entitas Admin menerima keluaran berupa visualisasi posisi kendaraan pada peta, status siklus perjalanan, dan ringkasan penggunaan bahan bakar. Diagram ini menegaskan batasan sistem dan bagaimana informasi telemetri bertransformasi menjadi informasi manajemen yang siap digunakan oleh pihak operator. Aliran data tingkat global ini direpresentasikan pada Gambar 4.



Gambar 4. Data Flow Diagram Level 0

3) *Data Flow Diagram (DFD) Level 1*: DFD Level 1 memberikan rincian proses internal sistem yang terbagi menjadi empat tahapan inti untuk mendukung otomatisasi manajemen armada. Proses pertama adalah Ingesti Data yang menangani penerimaan paket data JSON dari perangkat ESP32; proses kedua adalah Analisis Spasial yang menerapkan algoritma *Ray Casting* untuk verifikasi posisi kendaraan terhadap poligon *geofence* dan rumus *Haversine* untuk area lingkaran [8]; proses ketiga adalah Manajemen Perjalanan yang memperbarui status *trips* secara otomatis dari *pickup* ke *dropoff* berdasarkan hasil analisis spasial; dan proses terakhir adalah Distribusi Data *Real-time* melalui protokol *WebSockets* untuk mengirimkan pembaruan posisi ke dasbor web tanpa jeda [11]. Rangkaian aliran proses mendetail ini secara visual ditunjukkan pada Gambar 5.



Gambar 5. Data Flow Diagram Level 1

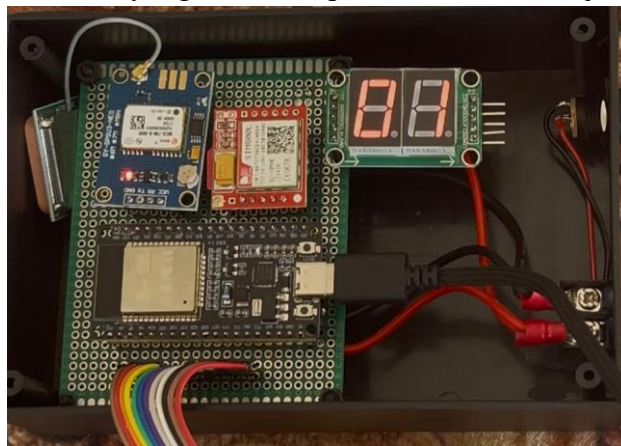
4. HASIL DAN PEMBAHASAN

F. Implementasi Perangkat Keras

Implementasi perangkat keras pada sistem Cartrack IMB difokuskan pada perakitan unit pelacak yang efisien dan mampu beroperasi pada kondisi lingkungan industri. Komponen utama yang digunakan adalah mikrokontroler ESP32 yang berperan sebagai

unit pemroses pusat untuk mengintegrasikan modul GPS Neo-6M dan sensor bahan bakar ultrasonik [13]. Seluruh komponen ini disusun di dalam *enclosure* yang terlindung untuk memastikan ketahanan perangkat saat dipasang pada armada truk [6].

Dalam tahap implementasi ini, perangkat keras dikonfigurasi untuk melakukan pembacaan sensor secara periodik dan mengirimkan data tersebut ke server melalui jaringan internet. Modul GPS bertugas mengunci sinyal satelit untuk mendapatkan koordinat presisi yang kemudian akan diolah oleh algoritma *geofencing* di sisi server [4]. Sementara itu, sensor ultrasonik yang terpasang pada tangki bahan bakar akan mengirimkan data ketinggian cairan yang dikonversi menjadi volume liter. Bentuk fisik dari unit perangkat keras IoT yang telah diimplementasikan ditunjukkan pada Gambar 6.



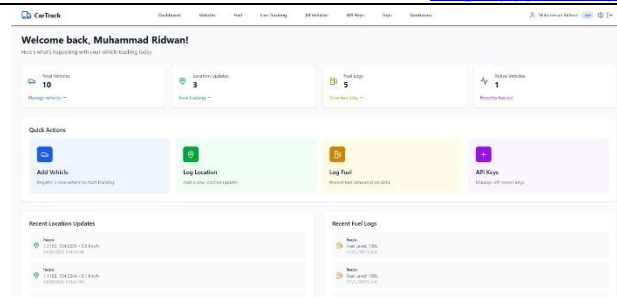
Gambar 6. Implementasi Perangkat Keras IoT Cartrack IMB

Integrasi antara perangkat keras dan perangkat lunak dijamin melalui protokol komunikasi HTTP yang mengirimkan paket data dalam format JSON. Setiap paket data menyertakan identitas unik kendaraan untuk memastikan tidak terjadi tumpang tindih informasi antara satu armada dengan armada lainnya [4]. Keberhasilan tahap implementasi ini menjadi fondasi bagi akurasi data yang akan ditampilkan pada antarmuka pengguna [3].

G. Implementasi Antarmuka Sistem

Implementasi antarmuka pada sistem Cartrack IMB dirancang untuk menyajikan data telemetri yang kompleks ke dalam visualisasi yang intuitif bagi operator. Antarmuka ini dibangun menggunakan *framework* React dengan pendekatan *Single Page Application* (SPA) untuk memastikan performa yang responsif [14]. Fokus utama dari antarmuka ini adalah integrasi peta interaktif dan pembaruan data secara *real-time* melalui protokol *WebSockets* [11].

1) *Halaman Dasbor Utama dan Monitoring Peta*: Halaman dasbor utama merupakan pusat pengawasan armada yang mengintegrasikan pustaka Leaflet.js dengan Google Maps API sebagai *base map*. Pada halaman ini, posisi kendaraan ditampilkan melalui penanda (*marker*) yang bergerak secara dinamis mengikuti koordinat terbaru yang dikirimkan oleh ESP32 [10]. Selain posisi, dasbor ini juga menampilkan status aktif kendaraan, kecepatan saat ini, dan notifikasi peringatan jika terjadi anomali [7]. Tampilan halaman dasbor utama sistem ditunjukkan pada Gambar 7.



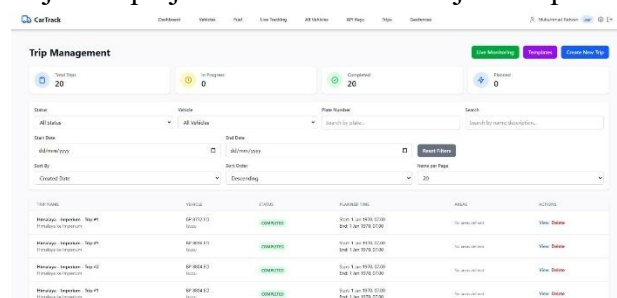
Gambar 7. Implementasi Dasbor Monitoring Utama

2) *Visualisasi Geofence dan Analisis Spasial*: Halaman ini menunjukkan implementasi wilayah operasional yang telah didefinisikan menggunakan algoritma *Ray Casting*. Batas wilayah ditampilkan dalam bentuk poligon berwarna yang mewakili area *pickup*, *dropoff*, dan koridor lintasan [8]. Antarmuka ini memungkinkan operator untuk melihat secara langsung apakah sebuah armada berada di dalam atau di luar zona aman, serta memantau pelanggaran wilayah secara visual [9]. Tampilan visualisasi *geofence* pada peta ditunjukkan pada Gambar 8.



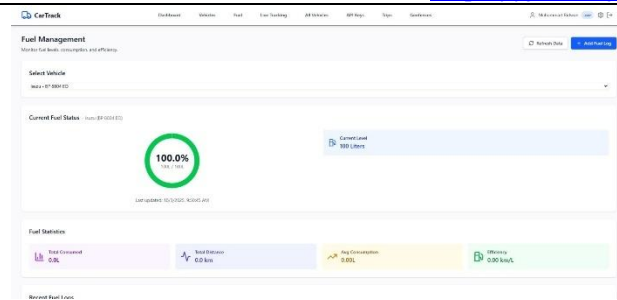
Gambar 8. Visualisasi Geofence Poligon pada Peta

3) *Manajemen Perjalanan (Trip Lifecycle)*: Halaman manajemen perjalanan berfungsi untuk memantau siklus ritase otomatis dari setiap armada. Di sini, sistem menyajikan daftar perjalanan yang sedang berlangsung (*In Progress*) maupun yang telah selesai (*Completed*). Data yang ditampilkan mencakup waktu mulai, waktu selesai, durasi perjalanan, serta kalkulasi otomatis jarak tempuh dan konsumsi bahan bakar [11]. Tampilan modul manajemen perjalanan otomatis ditunjukkan pada Gambar 9.



Gambar 9. Implementasi Modul Manajemen Perjalanan

4) *Monitoring Bahan Bakar dan Grafik Analitik*: Antarmuka ini khusus menyajikan data dari sensor ultrasonik dalam bentuk grafik *real-time*. Penggunaan grafik memungkinkan operator untuk mendeteksi pola konsumsi bahan bakar secara historis dan mengidentifikasi jika terdapat penurunan volume secara drastis dalam waktu singkat yang mengindikasikan adanya potensi pencurian atau kebocoran [4], [13]. Tampilan grafik monitoring bahan bakar ditunjukkan pada Gambar 10.



Gambar 10. Grafik Analitik Monitoring Bahan Bakar

H. Analisis Hasil Pengujian dan Pembahasan

1) *Pengujian Akurasi Geofencing dan Otomatisasi Trip*: Pengujian fitur geofencing pada sistem Cartrack IMB dilakukan untuk memvalidasi akurasi algoritma *Ray Casting* dalam mendeteksi keberadaan koordinat kendaraan terhadap poligon wilayah operasional yang mencakup area *pickup*, *dropoff*, dan koridor lintasan. Melalui simulasi pengiriman data GPS dari perangkat ESP32, sistem menunjukkan tingkat keberhasilan sebesar 100% dalam mendeteksi transisi koordinat yang memicu perubahan status perjalanan otomatis pada tabel *trips* di basis data. Efektivitas penggunaan algoritma *Ray Casting* dibandingkan metode radius lingkaran terlihat pada kemampuannya menangani batas wilayah yang tidak beraturan di lapangan industri, sementara implementasi protokol *WebSockets* memastikan pembaruan status tersebut diterima oleh dasbor operator dengan latensi minimal [8], [11]. Hasil pengujian ini membuktikan bahwa otomatisasi siklus perjalanan mampu meminimalisir kesalahan manusia dalam pencatatan ritase serta meningkatkan akuntabilitas pengawasan armada secara *real-time* [10]. Proses deteksi titik koordinat di dalam wilayah *geofence* poligon ini secara visual ditunjukkan pada Gambar 11.



Gambar 11. Hasil Pengujian Algoritma Ray Casting pada Geofence Poligon

2) *Pengujian Akurasi Monitoring Bahan Bakar*: Pengujian sensor bahan bakar dilakukan untuk memvalidasi tingkat presisi pembacaan sensor ultrasonik yang terintegrasi dengan ESP32 terhadap volume aktual pada tangki bahan bakar kendaraan. Berdasarkan hasil pengujian eksperimental dengan melakukan pengisian tangki secara bertahap, sistem menunjukkan tingkat akurasi yang tinggi dengan rata-rata persentase galat (*error rate*) sebesar 2-3%, yang dinilai masih berada dalam ambang batas toleransi operasional untuk pemantauan armada logistik skala besar [13]. Integrasi data antara log bahan bakar dan log lokasi pada sistem Cartrack IMB memungkinkan deteksi dini terhadap anomali konsumsi, seperti penurunan volume yang drastis saat mesin dalam kondisi mati (*engine off*), yang sering kali mengindikasikan adanya tindakan pencurian atau kebocoran tangki [4]. Keberhasilan pengujian ini membuktikan bahwa digitalisasi

pemantauan bahan bakar mampu memberikan transparansi biaya operasional yang jauh lebih akurat dibandingkan dengan metode pemantauan konvensional [6].

5. Kesimpulan dan Saran

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan terhadap sistem Cartrack IMB, dapat ditarik beberapa kesimpulan sebagai berikut:

- Integrasi teknologi IoT berbasis mikrokontroler ESP32 dengan Sistem Informasi Geografis (SIG) berbasis web berhasil diimplementasikan sebagai solusi manajemen armada yang akuntabel dan efisien.
- Penerapan algoritma *Ray Casting* pada fitur *geofencing* poligon menunjukkan tingkat akurasi deteksi wilayah sebesar 100%, sehingga mampu menangani batas area kerja yang kompleks dan tidak beraturan.
- Sistem berhasil menjalankan otomatisasi siklus perjalanan (*trip lifecycle*) secara tepat, yang secara signifikan meminimalisir kesalahan manusia dalam pencatatan ritase dari titik muat (*pickup*) ke titik bongkar (*dropoff*).
- Penggunaan sensor ultrasonik untuk pemantauan bahan bakar memberikan hasil yang presisi dengan rata-rata persentase galat (*error rate*) hanya sebesar 2-3%.
- Infrastruktur perangkat lunak yang menggunakan bahasa pemrograman Go dan protokol *WebSockets* terbukti efektif dalam mendistribusikan data telemetri secara *real-time* dengan latensi yang sangat rendah.

Saran

Untuk pengembangan sistem Cartrack IMB yang lebih optimal di masa mendatang, disarankan beberapa hal sebagai berikut:

- Mengintegrasikan algoritma *Machine Learning* untuk melakukan analisis prediktif terhadap pola konsumsi bahan bakar dan optimalisasi rute berdasarkan data historis perjalanan.
- Melakukan peningkatan pada sisi perangkat keras dengan menggunakan modul GPS yang memiliki sensitivitas lebih tinggi serta penambahan sensor suhu mesin untuk pemantauan kondisi fisik kendaraan yang lebih komprehensif.
- Mengembangkan aplikasi *mobile* khusus bagi pengemudi guna meningkatkan koordinasi dan interaksi dua arah antara personil di lapangan dengan operator di pusat kendali.
- Menambahkan fitur laporan otomatis yang dapat dikirimkan melalui email atau platform pesan instan untuk memudahkan manajemen dalam menerima ringkasan operasional harian.

Daftar Pustaka

- [1] G. Spyropoulos, M. Katopodi, K. Christopoulos, and E. Kostopoulos, "A Hybrid GIS–MCDM Approach to Optimal EV Charging Station Siting for Urban Planning and Decarbonization," *Futur. Transp.*, vol. 5, no. 4, pp. 1–27, 2025, doi: 10.3390/futuretransp5040186.

-
- [2] L. Sun, "Site selection for EVCSs by GIS-based AHP method," *E3S Web Conf.*, vol. 194, pp. 1–5, 2020, doi: 10.1051/e3sconf/202019405051.
- [3] A. J. Saputra and Suharjito, "Optimization Strategy for Electric Vehicle Charging Station Development at Gas Stations Using GIS-AHP-SAW Framework," *J. Tek. Inform.*, vol. 6, no. 6, pp. 5401–5417, 2025.
- [4] K. Nagarathna, C. D. Darshan, C. Shanawad, and C. A. Honnammanavar, "IoT-Based Real-Time Vehicle Tracking and Fuel Monitoring System with Theft Alert," *Int. J. Sci. Res. Eng. Trends*, vol. 11, no. 6, pp. 1–6, 2025.
- [5] M. Erbaş, M. Kabak, E. Özceylan, and C. Çetinkaya, "Optimal siting of electric vehicle charging stations: A GIS-based fuzzy Multi-Criteria Decision Analysis," *Energy*, vol. 163, pp. 1017–1031, 2018, doi: 10.1016/j.energy.2018.08.140.
- [6] R. Khatun, S. A. Antor, A. Ullah, and A. Hossain, "Vehicle Fuel Activities Monitoring System Using IoT," *Adv. Internet Things*, vol. 09, no. 04, pp. 63–71, 2019, doi: 10.4236/ait.2019.94005.
- [7] B. Huang, Z. Wang, J. Chen, B. Zhou, Y. Zhu, and Y. Liu, "Research on Site Selection and Capacity Determination of Electric Vehicle Public Charging Stations by Integrating K-Means++ and Improved RODDPSO," *Radioengineering*, vol. 34, no. 2, pp. 181–194, 2025, doi: 10.13164/re.2025.0181.
- [8] M. M. Raikar, K. Angadi, S. Reddy, N. Naik, and A. Doddwada, "Fleet tracking and Geofencing using the Internet of Things (IoT)," *ICSCCC 2023 - 3rd Int. Conf. Secur. Cyber Comput. Commun.*, pp. 463–468, 2023, doi: 10.1109/ICSCCC58608.2023.10176660.
- [9] A. S. Lesmana and I. K. Gunarta, "Jurnal Impresi Indonesia (JII) Analisis Pemilihan Lokasi Stasiun Pengisian Kendaraan Listrik Umum (SPKLU) di DKI Jakarta Menggunakan Geographic Information System dan Analytical Hierarchy Process," *J. Impresi Indones.*, vol. 4, no. 12, pp. 5474–5481, 2025.
- [10] D. P. Ramadhani and C. Susetyo, "Kendaraan Listrik Umum di Kota Denpasar Menggunakan Analytical Hierarchy Process (AHP) dan Geographic Information System (GIS)," *Tataloka*, vol. 27, no. 3, pp. 267–280, 2025, doi: 10.14710/tataloka.27.3.267-280.
- [11] T. Tanamal, Y. Adhiatma, M. Alghifar, A. Amran Nadeak, and N. Fathoni, "Implementation Fleet Management System With Real Time Monitoring And

-
- Controlling,” *J. Sos. Teknol.*, vol. 3, no. 8, pp. 635–639, 2023, doi: 10.59188/jurnalsostech.v3i8.897.
- [12] M. A. Damayanti, M. E. Lestiani, and S. K. Waskito, “Analisis Efektivitas Pemanfaatan GPS Tracker dalam Pemantauan Ketepatan Waktu Armada Distribusi,” *J. Manaj. Pendidik. dan Ilmu Sos.*, vol. 6, no. 6, pp. 4643–4652, 2025.
- [13] Y. F. Andhika, A. Tohir, M. Anwar, and S. Faidar, “Sistem Monitoring Konsumsi Bahan Bakar Berbasis IoT IoT-Based Fuel Consumption Monitoring System,” *Semin. Nas. Tek. Elektro*, vol. 10, pp. 203–207, 2025.
- [14] R. Annisa, R. Agung Ananda, and W. E. Sulistiono, “Implementasi Golang Clean Architecture Pada Perancangan Backend Point of Sales Website,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 2830–7062, 2024, [Online]. Available: <http://dx.doi.org/10.23960/jitet.v12i2.4668>