Advanced Mechanical and Mechatronic Systems

Research Article
Volume 2(1), 2026, 50–62
https://doi.org/10.53623/amms.v2i1.792

Integration of Vehicle Tracking, Control and Maintenance

Budiman Zahri, Tiara Syah Putri, Daniel Sutopo Pamungkas*

Politeknik Negeri Batam, Jl. Ahmad Yani, Tlk. Tering, Batam, Kepulauan Riau 29461, Indonesia

*Correspondence: daniel@polibatam.ac.id

SUBMITTED: 31 July 2025; REVISED: 13 September 2025; ACCEPTED: 2 October 2025

ABSTRACT: This research developed a vehicle tracking and control system aimed at improving fleet management efficiency in Batam. The problem addressed was the lack of an integrated solution that combined location monitoring, remote control, and service management for vehicles. The system integrated the ESP8266 microcontroller as the control unit, a Ublox Neo 6M GPS module for location tracking, a relay for engine control, and a SIM/GSM module for communication via HTTP. Data were stored in a PostgreSQL server and visualized using a web application developed with Node.js, Next.js, and Leaflet.js. The research objectives were to design, implement, and test a system capable of real-time vehicle monitoring, speed detection, and remote engine shutdown, accessible through a web browser on both computers and mobile devices. Testing was carried out using a motorcycle in the Batam Center area, showing that GPS readings, relay control, and data transmission were successfully executed. The highest longitude error observed was 0.000022, which remained within acceptable tolerance. The findings demonstrated that the system provided accurate and reliable vehicle tracking while offering practical solutions for fleet control. In conclusion, the developed system supported efficient vehicle management and could be further enhanced for broader fleet applications in urban areas.

KEYWORDS: GPS technology; vehicle management system; vehicle tracking.

1. Introduction

Motor vehicles were essential modes of transportation, and accurate location tracking was crucial for owners to manage their vehicles effectively. Clear location data supported various needs, such as monitoring usage, scheduling maintenance, and enabling remote control. In modern urban areas like Batam, where mobility demands were high, efficient vehicle management systems became increasingly important. However, despite the availability of various vehicle tracking and control solutions, many still faced limitations. Existing systems often focused only on one function such as GPS-based location monitoring, SMS-based control, or Google Maps-based tracking, without offering integration across key functionalities. This lack of integration resulted in gaps in real-time synchronization, limited reliability in controlling vehicle components such as relays, and the absence of structured maintenance scheduling features. These problems reduced the effectiveness of current systems in supporting comprehensive vehicle management.

Several paper has been presented a review study on vechicle tracking systems [1, 2]. A combination of literature review, system design, and impact evaluation on logistics is presnted in [1]. This literature study contributes practical solutions, strengthens data privacy, and provides insights into the role of tracking systems in transportation. Another recent review paper on vehicle tracking system is presented in [2]. The study conducted a survey study to optomise the maintainance strategy. The exisiting vehicle maintenance systems, technological advance montoring, preventive care, and safery were presented in the reivew study.

The urgency of addressing these issues arose from the increasing mobility and vehicle use in Batam. Without integrated systems, vehicle owners struggled with fragmented tools that made it difficult to ensure safety, maintain efficiency, and guarantee timely maintenance. An integrated solution was therefore necessary to provide real-time information, enhance vehicle control, and support preventive maintenance strategies.

Several tracking and control solutions had been proposed, such as IoT-based vehicle tracking system [3], GPS-based web security systems [4], SMS-based control using geographic coordinates [5], vehicle tracking applications using GPS and Google Maps [6], and motorcycle security systems with SMS-based remote features [7]. Although these studies provided important contributions, they remained limited in scope. Features such as combined tracking, speed monitoring, remote relay control, mobile and web accessibility, and maintenance scheduling had not yet been comprehensively realized. A recent study that present a combination of IoT principles and modern technologies to design a system offering real-time tracking, data collection, and analysis is available in [3]. The implementation integrates GPS, microcontrollers, wireless communication, and cloud storage, enhanced by analytics and machine learning for optimized routing and predictive maintenance. Another study that propsed an enhanced vehicle tracking system based GPS-GSM-IoT technology is presente in [8]. This paper proposed an IoT-based vehicle tracking system tailored for Somalia, leveraging GPS-enabled SIM cards and GSM networks to provide real-time monitoring despite limited internet access. The system enhances security, road safety, and fleet management, offering a practical solution to mitigate theft and optimize transportation logistics.

An efficient, low-cost vehicle tracking system integrating a smartphone application with a microcontroller is presented in [9]. The system uses GPS for real-time location data and GSM/GPRS to transmit updates to a database, displayed via Google Maps API. Users can monitor vehicles continuously, estimate arrival times, and enhance fleet management. Experimental results confirm the system's feasibility and effectiveness. Another researcher reported the vehicle tracker system based on GSM and GPS interface using Arduino platform [10]. The application of the vehicle tracking system using GPS and GSM for accident allert is presented in [11].

In a recent technology development, especially in the smart city, an IoT based vehicle traking system is necessary [12]. This study presented a smart system of vehicle traking and monitoring in real time. The proposed technology is based on the Long Range LoRa platform that enable a wireless communication infrastructure in the concept of IoE and smart cities.

Based on these considerations, this research aimed to design and implement a web-based integrated vehicle tracking system. The system combined real-time location tracking, remote relay control, vehicle speed monitoring, and calendar-based maintenance scheduling. It was developed using ESP8266, GPS Ublox Neo 6M, and GSM modules, along with PostgreSQL and supporting technologies such as Arduino IDE, Next.js, Leaflet.js, and Node.js. The research also evaluated system performance through GPS accuracy testing (Mean Absolute Error, Root Mean Square Error, and Standard Deviation), reliability of relay status control, and environmental testing (indoor, semi-indoor, and outdoor). With this approach, the study delivered a more integrated and reliable vehicle management solution suitable for the needs of urban areas like Batam.

2. Materials and Methods

This research developed a structured vehicle tracking system that integrated hardware components and web technologies to support accurate, real-time tracking and control. The methodology combined both hardware and software design with thorough testing procedures in a real-world environment. Figure 1 illustrates the flow diagram of the stages in this study. By following a structured process flow, it was expected that the implementation of the research would proceed smoothly and produce outputs consistent with the research objectives.

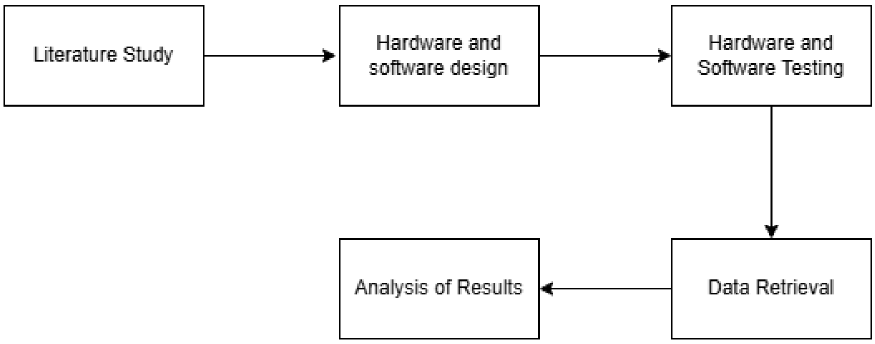

Figure 1. Diagram of the research phase.

2.1. System architecture and components.

The core of the system was the ESP8266 microcontroller, which acted as the control unit, collecting data from various sensors and transmitting it to the server [13]. For location tracking, the Ublox Neo 6M GPS module was employed due to its high sensitivity and reliable performance in diverse environments [14]. This module continuously captured latitude and longitude data in real time. The Songle SRD-05VDC-SL-C relay featured a compact design for high-density PCB mounting, supported up to 10 A switching capacity at 125 VAC/28 VDC, offered fast response times (≤10 ms operation, ≤5 ms release), and provided a long mechanical life expectancy of 10^7 operations [15]. Internet connectivity was established through the SIM900A GSM/GPRS module, which communicated with the server using the HTTP protocol [16].

All components were powered by a 2,500 mAh lithium polymer battery, ensuring stable and uninterrupted operation during testing. The devices were programmed and managed through the Arduino IDE [17]. Collected data including GPS coordinates, relay status, and vehicle speed, were transmitted to the PostgreSQL database, which was chosen for its stability, flexibility, and ability to handle complex relational data [18]. The data were then accessed through a Next.js web application, which provided real-time GPS visualization using Leaflet.js, maintenance scheduling via calendar integration, reporting, and relay control. From Figure 2, it was observed that the architecture clearly separated the hardware, server, and application layers. The ESP8266 acted as the central hub, receiving sensor data and sending it to the server through GSM/GPRS. The server then stored the information in PostgreSQL, which ensured reliable data management. Finally, the web application provided a user-friendly interface for visualization, reporting, and control.

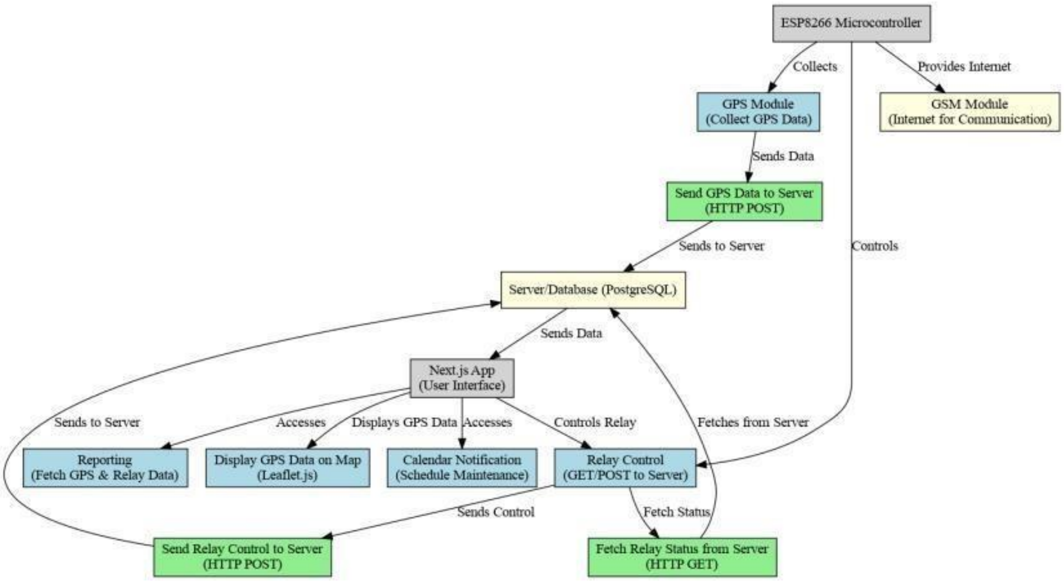

Figure 2. System architecture of vehicle tracking system.

2.2. Backend and web development.

The backend was developed using Node.js, which handled server-side operations, including API integration and database communication [19]. The frontend was built using Next.js, a React-based framework that supported real-time updates and modular web interfaces [20]. Interactive maps were implemented with Leaflet.js, a lightweight and mobile-friendly open-source JavaScript library for rendering maps. This enabled users to view vehicle location data dynamically from both web and mobile interfaces.

2.3. Research methodology.

The development process involved several stages, including observation, which focused on analyzing each component to ensure compatibility and reliable integration; a literature review to study existing vehicle tracking technologies and inform the system design; design and integration, which involved assembling the hardware and implementing the system software; and finally, field testing conducted in the Batam Center area using motorcycles, where GPS coordinates were collected across various environments (indoor, semi-indoor, and outdoor)..

2.4.Hardware pin configuration.

The table below presents the configuration of hardware components connected to the ESP8266 microcontroller. Each component was assigned specific pins on the ESP8266 to enable its designated function, including GPS tracking, GPRS communication, and remote engine control. As shown in Table 1, the ESP8266 assigned different pins to each component to ensure proper operation. The relay was connected to pin D0 for remote engine control, the SIM900A module used pins D7 and D8 for GPRS communication, and the GPS module used pins D2 and D3 for location tracking.

Table 1. ESP8266 pin configuration for Relay, SIM900A, and GPS Module.

Component	ESP8266 PIN	Function
Relay	D0	Engine Control
SIM900A	D7 (RXD5), D8 (TXD5)	GPRS communication
GPS Module	D2 (TX), D3 (RX)	Location tracking

2.5.Expected outcomes.

The system was expected to provide accurate GPS tracking, allow real-time remote control of the engine, display synchronized vehicle status data on web applications, secure user data through access control, and support responsive user interface themes along with calendar-based scheduling.

2.6.Hardware and software design.

Hardware design is carried out by integrating various components to form the Vehicle Tracking, Control, and Maintenance Integration system. The process involves defining the workflow of electrical connections and the relationships between hardware modules. Figure 3 shows the Hardware System Block Diagram, which highlights the workflow and relationships between hardware components. The diagram illustrates the flow of information between input and output components, the microcontroller, and the server or database. It demonstrates how data is transmitted, processed, and utilized within the system.

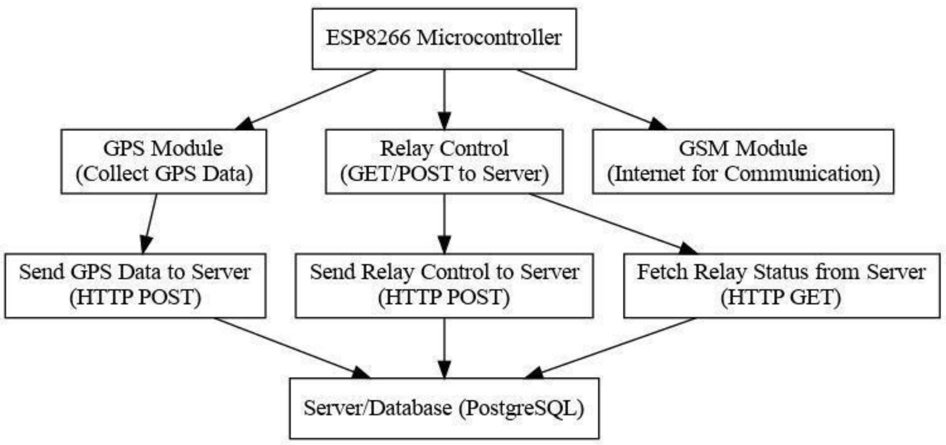

Figure 3. Hardware system block diagram.

The Hardware System Block Diagram illustrates the flow of information between input and output components, the microcontroller, and the server or database. It demonstrates how data is transmitted, processed, and utilized within the system. The software system integrates database management, API services, and web applications to enable real-time vehicle monitoring and control. Figure 4 shows the Software and Hardware System Block Diagram, which illustrates the complete integration and workflow between hardware components, the PostgreSQL server, and user applications. The diagram demonstrates how the system manages GPS data acquisition, relay control, reporting, and calendar notifications through coordinated communication between all system layers.

Figure 4 demonstrates the complete workflow, starting from the ESP8266 microcontroller and hardware modules, continuing through the server, and ending with user applications for monitoring and control. The web application provides a comprehensive user interface with structured workflows. Users begin with authentication (login or registration), then access the main dashboard for fleet overview. The system supports vehicle registration and assignment, real-time monitoring with interactive maps, maintenance scheduling through calendar integration, and detailed reporting features. This design ensures intuitive navigation and supports efficient fleet management operations from both desktop and mobile devices.

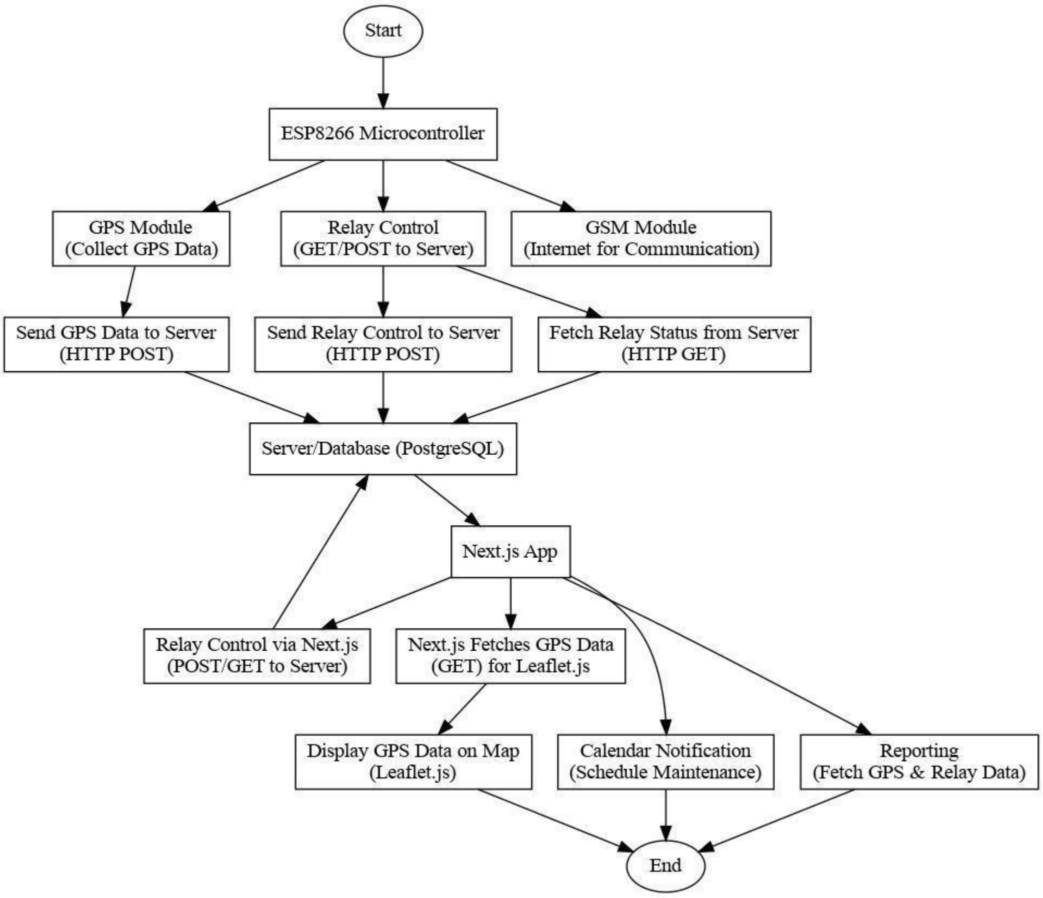

Figure 4. Software and hardware system block diagram.

2.7. Testing.

To support the implementation of the proposed vehicle tracking system, a prototype was assembled using essential hardware components. These components were connected and housed within a compact casing. The hardware setup consists of a microcontroller (ESP8266), GPS module, GPRS module (SIM900A), relay, and battery.

Figure 5. Assembled prototype of the tracking device.

As shown in Figure 5, all components are integrated into a single enclosure to facilitate data collection and transmission. This configuration enables real-time vehicle tracking, engine control through the relay, and communication with the server via GPRS. The physical prototype played a crucial role during field testing to validate system performance and data accuracy.

- Testing Techniques: Testing was conducted with black box testing to assess functionality, data accuracy, and system integration. System results are compared with actual data, such as GPS position and vehicle speed.
- Testing Objectives: The testing phase aimed to verify that the system functions according to specifications, ensure accuracy and appropriateness, and confirm that hardware and software integration works seamlessly.
- Tools, Materials, and Testing Environment: Testing uses microcontrollers with GPS and relays, database servers, web applications, and is carried out in the field for accurate tests.
- Testing Scenario: The testing scenarios included relay function verification to ensure proper command response, GPS function testing to assess location coordinate accuracy, vehicle speed verification to validate displayed readings, and data storage testing to confirm complete and lossless data retention on the server.
- Test Results and Analysis: Results showed the system worked according to specifications, GPS position and speed were accurate, relays responded appropriately, and data was stored completely on the server. Improvements were made where problems were found.

3. Results and Analysis

3.1.Research result data.

Tests were conducted to assess functionality, data accuracy, system integration, security, performance, error handling, and user experience. The password hashing process is also tested as part of the security aspect.

3.1.1. Functional testing.

Testing ensures that each feature of the system operates as intended and meets its functional requirements. During the testing phase, the Fleet Machine Creation module was validated to confirm that the system automatically generated *fleet_machine* records upon microcontroller initialization. This process effectively prevented duplication by unique identifiers. The administrator interface supported efficient machine-to-user assignment via a form-based workflow. Each machine was successfully assigned a unique ID, along with timestamps for both creation and modification tracking, ensuring accurate relationship mapping between fleet machines and users within the database.

For the Fleet Vehicle Creation module, users were able to create *fleet_vehicle* records through the web interface, seamlessly linking each vehicle to its assigned machine via a dropdown selection. The system correctly recorded essential vehicle details, including the vehicle name, license plate, and associated machine ID. Database validation confirmed that all entries were accurately stored with appropriate user associations and timestamp tracking, thereby ensuring reliable fleet management and monitoring functions.

The Monitoring Dashboard performed successfully, displaying real-time vehicle data through an interactive web interface accessible on both desktop and mobile devices. As illustrated in Figure 6, the system provided dynamic visualization of vehicle locations using interactive maps, real-time status indicators, and relay control buttons. The *fleet_status* table continuously recorded GPS data, including latitude, longitude, altitude, speed, and course, each tagged with precise timestamps. Relay control functionality was also validated through the dashboard interface, confirming that all switching commands were accurately logged in the database and executed by the hardware with 100% reliability across multiple testing cycles.

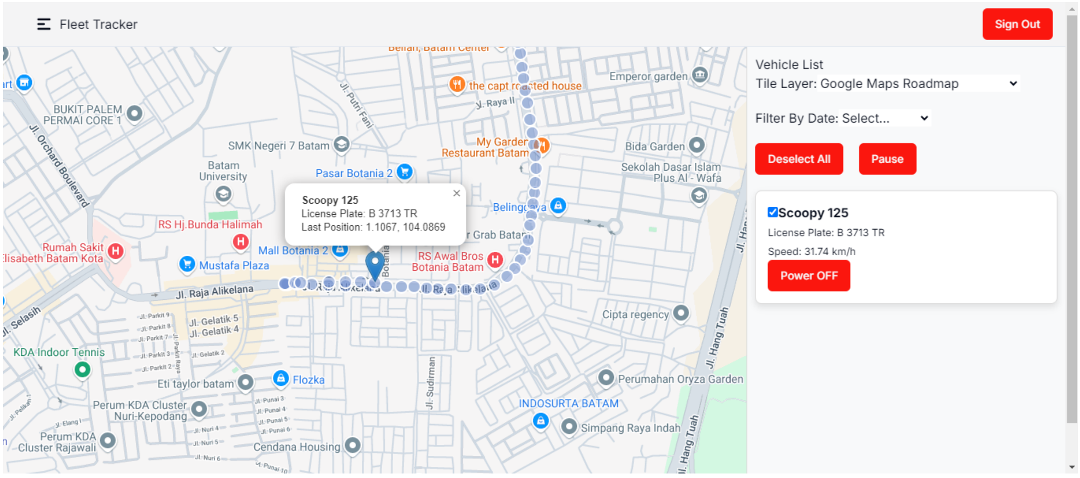

Figure 6. Dashboard UI view.

3.2.2. Data and accuracy testing.

This stage ensures that the data recorded in the system accurately represents real-world conditions and performs reliably across different operational scenarios. Vehicle speed accuracy was validated through comparative testing between system-recorded values and actual measured speeds, both at rest and during motion. As shown in Table 2, the system demonstrated high precision across various speed ranges. When stationary (0 km/h), the system correctly recorded zero velocity. During movement, recorded speeds closely matched actual values, with measurements ranging from 10.22 to 10.61 km/h at an actual speed of approximately 10 km/h, 15.04 to 15.98 km/h at 15 km/h, and 20.28 to 20.39 km/h at 20 km/h. These results confirm minimal deviation and a strong correlation between system output and real conditions, validating the reliability of the speed monitoring feature.

Table 2. Testing vehicle speed data against data on the dashboard.

No	ID	Machine Create Date	Machine ID	Speed (km/h)	Average Vehicle Speed (km/h)
1	1972	2024-10-13 21:36:04	BT-cm27of5yw00000cjvgvrpbode	0.00	0.00
2	2072	2024-10-13 21:50:16	BT-cm27of5yw00000cjvgvrpbode	0.00	0.00
3	2131	2024-10-13 21:56:31	BT-cm27of5yw00000cjvgvrpbode	10.61	10.00
4	2150	2024-10-13 21:58:06	BT-cm27of5yw00000cjvgvrpbode	10.54	10.00
5	2157	2024-10-13 21:58:41	BT-cm27of5yw00000cjvgvrpbode	10.22	10.00
6	2189	2024-10-13 22:01:23	BT-cm27of5yw00000cjvgvrpbode	15.45	15.00
7	2199	2024-10-13 22:02:13	BT-cm27of5yw00000cjvgvrpbode	15.04	15.00
8	2207	2024-10-13 22:02:53	BT-cm27of5yw00000cjvgvrpbode	15.98	15.00
9	2259	2024-10-13 22:08:23	BT-cm27of5yw00000cjvgvrpbode	20.39	20.00
10	2269	2024-10-13 22:09:12	BT-cm27of5yw00000cjvgvrpbode	20.28	20.00
11	2277	2024-10-13 22:09:55	BT-cm27of5yw00000cjvgvrpbode	20.30	20.00

GPS position testing further confirmed the system's accuracy in tracking real-time locations. Coordinates recorded by the system were compared with Google Maps reference data across multiple test points, as detailed in Table 3. The comparison showed precise alignment between recorded and reference coordinates, with deviations remaining well within acceptable tolerance levels. The system consistently provided accurate latitude and longitude values suitable for real-time vehicle tracking, maintaining very low error margins across all test environments. Slightly higher variations were observed in semi-indoor and outdoor areas compared to indoor conditions, but overall performance remained highly reliable and consistent.

Table 3. 10 Point testing.

	Point	Lat_ Avg	Lon_ Avg	MAE_ Lat	MAE_Lo n	RMSE_ Lat	RMSE_ Lon	SD_ Lat	SD_ Lon	Category
0	1_indoor.csv	1.120	104.093	0.000013	0.000013	0.000016	0.000015	0.000016	0.000015	indoor
1	2_indoor.csv	1.120	104.093	0.000015	0.000014	0.000018	0.000017	0.000018	0.000017	indoor
2	3_indoor.csv	1.121	104.094	0.000013	0.000015	0.000015	0.000016	0.000015	0.000017	indoor
3	4_indoor.csv	1.120	104.093	0.000017	0.000014	0.000018	0.000016	0.000019	0.000016	indoor
4	5_indoor.csv	1.120	104.093	0.00001	0.000011	0.000013	0.000015	0.000014	0.000015	indoor
5	6_indoor.csv	1.114	104.065	0.000014	0.000012	0.000016	0.000013	0.000016	0.000014	indoor
6	7_indoor.csv	1.115	104.065	0.000016	0.000015	0.000017	0.000018	0.000018	0.000018	indoor
7	8_indoor.csv	1.115	104.065	0.000014	0.000016	0.000017	0.000018	0.000017	0.000018	indoor
8	9_indoor.csv	1.115	104.065	0.000015	0.000014	0.000017	0.000017	0.000018	0.000017	indoor
9	10_indoor.csv	1.115	104.064	0.000012	0.000009	0.000014	0.000011	0.000014	0.000012	indoor
10	1_outdoor.csv	1.120	104.093	0.000011	0.00001	0.000013	0.000012	0.000013	0.000012	outdoor
11	2_outdoor.csv	1.121	104.093	0.000009	0.000016	0.000012	0.000018	0.000012	0.000019	outdoor
12	3_outdoor.csv	1.121	104.094	0.000013	0.000014	0.000016	0.000015	0.000016	0.000016	outdoor
13	4_outdoor.csv	1.120	104.094	0.000011	0.000009	0.000014	0.000011	0.000014	0.000011	outdoor
14	5_outdoor.csv	1.120	104.094	0.000011	0.000011	0.000013	0.000013	0.000013	0.000013	outdoor
15	6_outdoor.csv	1.120	104.094	0.000012	0.000012	0.000014	0.000014	0.000014	0.000014	outdoor
16	7_outdoor.csv	1.119	104.094	0.000015	0.000014	0.000017	0.000016	0.000018	0.000016	outdoor
17	8_outdoor.csv	1.119	104.094	0.000012	0.000012	0.000013	0.000014	0.000014	0.000014	outdoor
18	9_outdoor.csv	1.115	104.065	0.000016	0.000017	0.000019	0.000022	0.000019	0.000022	outdoor
19	10_outdoor.cs v	1.114	104.064	0.000015	0.000014	0.000017	0.000016	0.000017	0.000016	outdoor
20	1_semi_indoo r.csv	1.121	104.093	0.000013	0.000013	0.000016	0.000017	0.000016	0.000017	semi_indoor
21	2_semi_indoo r.csv	1.120	104.093	0.000012	0.000012	0.000014	0.000014	0.000015	0.000014	semi_indoor
22	3_semi_indoo r.csv	1.121	104.095	0.000013	0.000013	0.000015	0.000015	0.000015	0.000015	semi_indoor
23	4_semi_indoo r.csv	1.121	104.094	0.000013	0.000014	0.000016	0.000016	0.000016	0.000016	semi_indoor
24	5_semi_indoo r.csv	1.120	104.094	0.000011	0.00001	0.000013	0.000011	0.000013	0.000011	semi_indoor
25	6_semi_indoo r.csv	1.120	104.094	0.000012	0.000014	0.000014	0.000016	0.000014	0.000016	semi_indoor
26	7_semi_indoo r.csv	1.120	104.094	0.00001	0.000013	0.000013	0.000015	0.000013	0.000015	semi_indoor
27	8_semi_indoo r.csv	1.120	104.093	0.00001	0.000009	0.000012	0.000012	0.000012	0.000012	semi_indoor
28	9_semi_indoo r.csv	1.119	104.094	0.000016	0.000015	0.000018	0.000018	0.000018	0.000018	semi_indoor
29	10_semi_indo or.csv	1.115	104.065	0.000014	0.000014	0.000017	0.000016	0.000017	0.000016	semi_indoor

Relay testing was conducted to assess the precision of remote-control operations. A total of 37 consecutive switching cycles were performed across two fleet machines (BT-clzwxy3xy00010cjs46mv7yyj and BT-cm1de9op200020dlb1ukhgde8). The tests yielded 100% accuracy, with recorded relay states matching the actual hardware responses in every instance. Output voltage readings consistently showed 3.3V for logic HIGH and 0V for logic LOW, confirming dependable engine control functionality. The database also maintained perfect synchronization with physical relay states, with accurate timestamp logging for each switching action.

Finally, the system's GPS reliability was further evaluated across 10 indoor, 10 outdoor, and 10 semi-indoor locations to assess environmental influence on performance. Statistical analyses, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Standard Deviation (SD), were used to quantify accuracy and precision. The summarized data, presented in Table 3, indicates that the system maintained very low error margins across all conditions, with slightly higher but still acceptable deviations in outdoor and semi-indoor settings. These results, illustrated in Figure 7, confirm that the system performs consistently and accurately across diverse environmental contexts, demonstrating its robustness for real-world fleet monitoring applications.

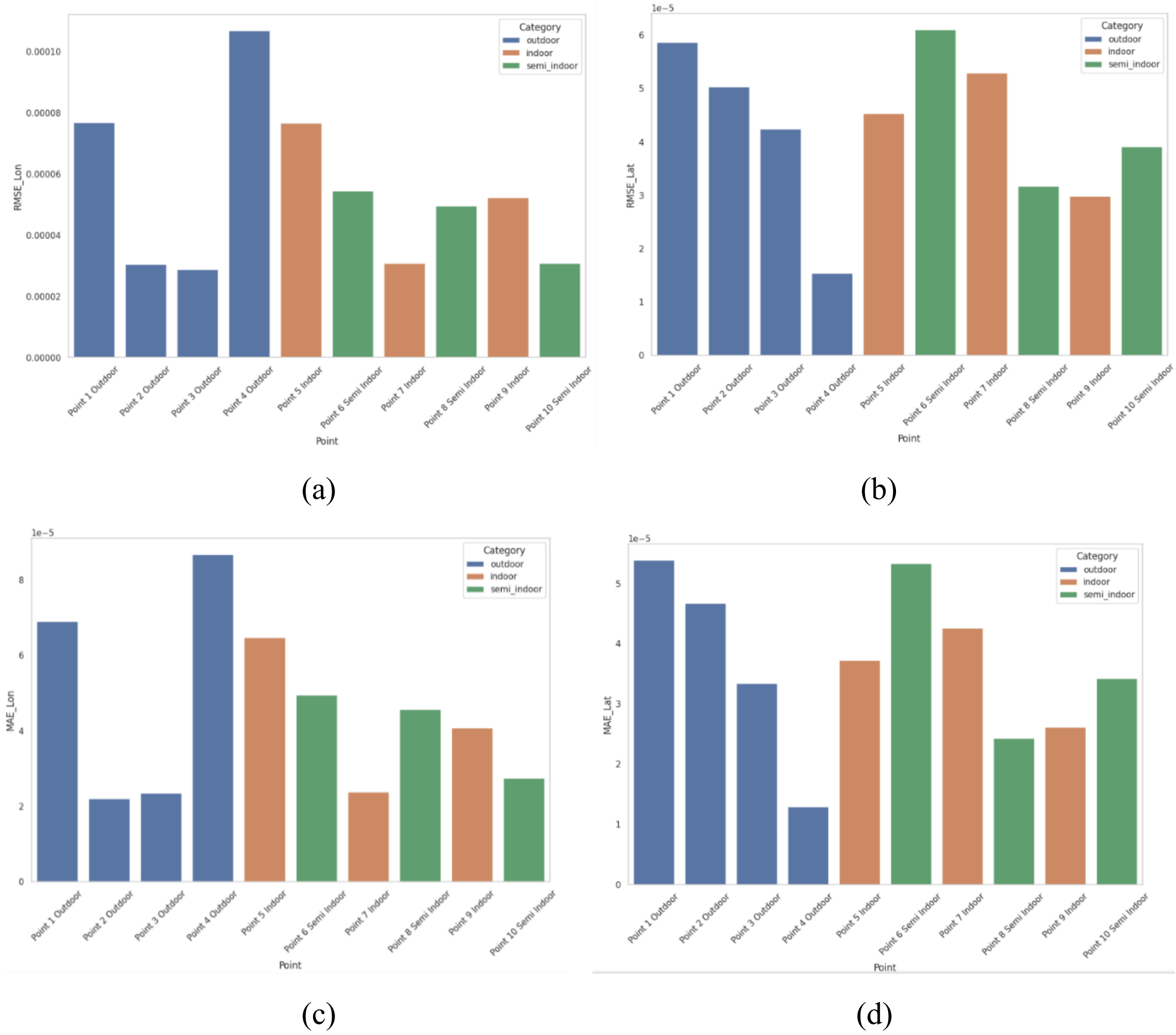

Figure 7. GPS accuracy: (a) RMSE Longitude; (b) RMSE Latitude; (c) MAE Longitude; (d) MAE Latitude.

The graphical analysis confirms that errors remain consistently low and stable across all testing conditions, demonstrating the system's high level of accuracy. Among the environments tested, outdoor conditions showed the highest precision, with Latitude RMSE = 0.000013 and Longitude RMSE = 0.000012. Indoor areas exhibited slightly higher error values (Latitude RMSE = 0.000016 and Longitude RMSE = 0.000015) due to occasional signal obstructions, while semi-indoor conditions produced intermediate results (Latitude RMSE = 0.000013 and Longitude RMSE = 0.000014). These findings collectively reinforce that the system performs reliably and maintains accurate tracking under various environmental conditions.

Security testing was conducted to ensure the system is well-protected against unauthorized access, focusing primarily on password management and user authentication. The results confirmed that user credentials are securely stored in the database through the implementation of the bcrypt hashing algorithm, effectively preventing the exposure of sensitive information in plain text. This demonstrates that the system's authentication mechanism adheres to secure coding practices and ensures data confidentiality.

Calendar feature testing validated the system's capability to display user schedules accurately and manage access permissions effectively. The system successfully identified and displayed scheduling conflicts that occurred within the same time slots. Furthermore, access control verification confirmed that only authorized users could view or modify schedules on their respective calendars, thus maintaining data integrity and privacy. Testing of the user interface themes, including Dark Mode and Light Mode, verified that the system correctly supports theme switching across all pages without functional or visual errors. The interface consistently adapted to user preferences, successfully detecting the device's default theme setting and applying the corresponding mode automatically. This ensures a seamless and responsive user experience, maintaining visual comfort and accessibility under different lighting conditions.

4. Conclusions

Based on the research findings, this study successfully developed and validated an integrated vehicle tracking and control system for fleet management applications in Batam. The system underwent comprehensive testing on GPS tracking and relay control modules, with GPS accuracy measured using Root Mean Square Error (RMSE) for latitude and longitude across 30 location points in indoor, outdoor, and semi-indoor environments. Relay functionality was rigorously tested through 37 consecutive switching operations, comparing hardware status with database records. The GPS accuracy testing demonstrated exceptional performance across all environmental conditions. Indoor environments yielded Latitude RMSE of 0.000016 and Longitude RMSE of 0.000015, while outdoor environments achieved the highest accuracy with Latitude RMSE of 0.000013 and Longitude RMSE of 0.000012. Semi-indoor conditions produced intermediate results with Latitude RMSE of 0.000013 and Longitude RMSE of 0.000014. These findings indicate that outdoor environments provide optimal accuracy, while indoor areas exhibit slightly higher error due to signal obstruction, though performance remains reliable for practical tracking applications. Relay performance testing achieved 100% accuracy across all 37 test attempts, with output voltage consistently measuring 3.3V for logic HIGH and 0V for logic LOW states. This confirms the system's reliable hardware control capability for remote engine management. The successful integration of Next.js, PostgreSQL, Arduino IDE, ESP8266, and SIM900A technologies demonstrates that all system modules performed as expected in real-time operation, effectively combining tracking, control, and maintenance functions. The developed system significantly enhances vehicle monitoring efficiency by providing accurate tracking data and reliable relay-based engine control. It contributes to improved fleet management, particularly in urban environments like Batam where real-time monitoring is critical for operational efficiency and safety.

Acknowledgments

The authors would like to express their sincere gratitude to Mr. Daniel Sutopo for serving as the supervising lecturer of this research project. The authors also wish to acknowledge the institutional support provided by Batam State Polytechnic, which greatly facilitated the successful completion of this study through access to essential resources and facilities.

Author Contribution

Budiman Zahri contributed to the conceptualization, software development, data collection, project administration, funding acquisition, as well as editing and review. Tiara Syah Putri was responsible for data analysis, writing, visualization, methodology, and hardware design and integration. Daniel Sutopo provided supervision and overall project guidance.

Competing Interest

The authors declare that there are no competing financial, personal, or professional interests that could have influenced the content or outcomes of this research.

References

[1] Alhwety, R.N.H.; Elfadil, N. (2024). Vehicle Tracking System Approaches: A Systematic Literature Review. *International Journal of Computer Science and Mobile Computing*, *13*(8), 23−31. https://doi.org/10.47760/ijcsmc.2024.v13i08.003.

[2] Zainal Abidin, M.S.; Mohd Fadzil, L. (2024). Vehicular Tracking System for Optimized Maintenance Strategy – Survey. *International Journal of Electrical and Electronics Engineering*, *11*(9), 284−293. https://doi.org/10.14445/23488379/IJEEE-V11I9P126.

[3] Khanal, A.; Shrestha, M.; (2024). Research and Appropriate Implementation on Vehicle Tracking System Using IoT. *International Journal for Multidisciplinary Research*, *6*(1), 1−45. https://doi.org/10.36948/ijfmr.2024.v06i01.12591.

[4] Kie, L. (2017). Analisis Yuridis terhadap Implementasi Pasal 362 Kitab Undang-Undang Hukum Pidana tentang Pencurian pada Kasus Kendaraan Bermotor di Kota Batam. *Judicial Review*, *XIX*(2), 133–152.

[5] Rizki Wahid, H.; Supit, Y.; Catur Sakti, S. (2023). Perancangan Aplikasi Pelacakan Kendaraan Motor Menggunakan GPS dengan Google Maps. *SIMTEK Jurnal Sistem Informasi dan Teknik Komputer*, *8*(2), 412−416. https://doi.org/10.51876/simtek.v8i2.327.

[6] Onanda, W.; Meka, S.Y.; Abadi, A.; Widia, R.; Rahman, A. (2024). Prototype Alat Pengaman Sepeda Motor Berbasis Arduino Uno Terintegrasi GPS, Notifikasi, dan Kontrol Melalui SMS. *Jurnal Teknologi dan Rekayasa*, *10*(1), 1−10. https://doi.org/10.35143/elementer.v10i1.6255.

[7] Risaldi, M.R.S.V.I.W. (2024). Rancang Bangun Alat Pelacak Posisi Kendaraan Berbasis IoT. *Teknik Informatika Unanda*, *3*(1), 29–33.

[8] Abdirahman, A.A.; Hashi, A.O.; Dahir, U.M.; Elmi, M.A.; Mohd Hashim, S.Z. (2025). Enhanced Vehicle Tracking: A GPS-GSM-IoT Approach. *International Journal of Computing and Digital Systems, 17*(1), 1−11.

[9] Le, S.J.; Tewolde, G.; Kwon, J. (2014). Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and Smartphone Application. 2014 IEEE World Forum on Internet of Things, Seoul, South Korea..

[10] Morallo, N.T. (2021). Vehicle Tracker System Design based on GSM and GPS Interface Using Arduino as Platform. *Indonesian Journal of Electrical Engineering and Computer Science, 23*(1), 258−264. http://doi.org/10.11591/ijeecs.v23.i1.pp258-264.

[11] Jammula, M.; Charanjit, N.; Saitharun, B.; Vashista, B. (2021). Accident Alert and Vehicle Traking System Using GPS and GSM. *Asian Journal of Applied Science and Technology, 5*(2), 81−89.

[12] Angelov, K.K.; Manchev, N.P.; Pasarelski, R.I.; Hristina, S. (2024). IoT Based Vehicle Tracking System in Smart City Concept. 2024 XXXIII International Scientific Conference Electronics, Sozopol, Bulgaria.

[13] ESP8266 Series of Modules. (accessed on 14 October 2024). Available online: https://www.espressif.com/en/products/modules/esp8266.

[14] NEO-6 u-blox 6 GPS Modules Data Sheet (accessed on 25 April 2024).. Available online: https://www.u-blox.com.

[15] SRD-05VDC-SL-C Relay Datasheet. (accessed on 1 October 2024). Available online: https://www.alldatasheet.com/datasheet-pdf/download/1132028/SONGLERELAY/SRD-05VDC-SL-C.html.

[16] SIM900A Datasheet. (accessed on 14 October 2024). Available online: https://www.datasheethub.com/wp-content/uploads/2022/08/SIM900A-Datasheet.pdf.

[17] Arduino Integrated Development Environment (IDE) v1. (accessed on 14 October 2024).. Available online: https://www.arduino.cc/en/software.

[18] PostgreSQL 17 Documentation. (accessed on 14 October 2024). Available online: https://www.postgresql.org/docs/17.

[19] Node.js Documentation. (accessed on 14 October 2024). Available online: https://nodejs.org/en/docs.

[20] Next.js by Vercel – The React Framework.. (accessed on 14 October 2024). Available online: https://nextjs.org/.