

Implementation and design of GPS tracker monitoring system on car rental vehicles based on internet of things using Nodemcu ESP-32

Indah Purnama Sari¹, Al-Khowarizmi¹, Asrar Aspia Manurung²

¹Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia

²Department of Data Science, Faculty of Computer Science and Information Technology, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia

Article Info

Article history:

Received Apr 3, 2025
Revised Aug 18, 2025
Accepted Nov 28, 2025

Keywords:

Arduino Uno
GPS tracker
Internet of things
Nodemcu ESP-32
Real-time monitoring

ABSTRACT

Internet of things (IoT) based vehicle tracking system is an effective solution to overcome various problems in the vehicle rental industry, such as asset loss, route misuse, and late returns. This study aims to design and implement a real-time vehicle position monitoring system using the NodeMCU ESP-32 module integrated with the NEO-6M GPS module and Wi-Fi connectivity to send data to a cloud-based server. This system is designed to display the vehicle position directly through a web-based digital map interface, which can be accessed by vehicle owners anytime and anywhere. The methodology used includes hardware and software design, location accuracy testing, and data integration with a web-based visualization platform using a map API. The test results show that the system is capable of sending vehicle location data with a position accuracy level of up to ± 5 meters and data updates every 10 seconds under stable network conditions. In addition, the system has good power efficiency, with an average current consumption of 80–100 mA when active. All data was successfully stored and visualized in real-time using the Google Maps API, and the system was able to operate stably for 24 hours of non-stop testing. Based on these results, the IoT-based GPS tracker system with NodeMCU ESP-32 can be effectively implemented on rental vehicles as a modern monitoring solution that is cost-effective, flexible, and easily accessible. This system provides added value in fleet monitoring and supports faster and data-based decision making.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Indah Purnama Sari
Department of Information Technology, Faculty of Computer Science and Information Technology
Universitas Muhammadiyah Sumatera Utara
3 Kapten Muchtar Basri Street, Medan, 20238, Indonesia
Email: indahpurnama@umsu.ac.id

1. INTRODUCTION

The rapid advancement of internet of things (IoT) technologies has transformed real-time monitoring and control systems across industries. IoT enables physical assets such as vehicles to be interconnected through internet-based networks, allowing continuous monitoring of parameters including position, speed, and operational status [1], [2]. This paradigm shift has been particularly impactful in transportation and fleet management, where real-time visibility and automation are essential for operational efficiency and security.

In the vehicle rental industry, companies face recurring challenges such as late returns, unauthorized usage outside designated areas, and theft. Traditional monitoring systems, which often rely on manual reporting

or non-integrated GPS devices, fail to provide timely and actionable insights. These limitations hinder business owners from exercising effective control over their assets, resulting in financial losses and reduced customer trust. Consequently, there is a pressing need for IoT-enabled monitoring systems that deliver real-time data, are accessible remotely, and remain cost-efficient and energy-efficient [2], [3].

Recent research has explored IoT-based vehicle tracking systems using diverse hardware platforms. Yakubjanova [4] demonstrated GSM-GPS systems with Arduino UNO and SIM800L modules, which were effective but limited by poor cloud integration and high energy consumption. Daniel and Vasanthi [5] highlighted that Arduino-based solutions often require external Wi-Fi modules, increasing system complexity and cost. Umarov *et al.* [6] proposed a Raspberry Pi-based tracker, but its high power consumption and large form factor made it unsuitable for small rental vehicles. Welbourne *et al.* [7] employed GSM-GPS with serial communication, yet their system lacked efficient cloud-based updates via message queuing telemetry transport (MQTT). Dada and Thiesse *et al.* [8] emphasized cost efficiency for MSMEs in Indonesia, but their Arduino Mega and SIM900 solution faced compatibility and power supply limitations. Beyond these, international studies such as Youm *et al.* [9] and Landegem and Viswanathan *et al.* [10] have shown that IoT-based GPS tracking can enhance logistics and fleet management, but scalability and energy efficiency remain unresolved challenges.

Despite these advances, existing systems remain constrained by either excessive power consumption, reliance on multiple external modules, or limited real-time cloud connectivity. This creates a clear research gap: the need for a compact, energy-efficient, and cloud-ready IoT solution specifically tailored to rental vehicle monitoring. Addressing this gap requires leveraging microcontrollers that integrate communication, processing, and connectivity functions in a single platform.

This study contributes to the field by proposing the design and implementation of an IoT-based GPS tracker system using the NodeMCU ESP-32 microcontroller integrated with the NEO-6M GPS module. The ESP-32 offers built-in Wi-Fi, low power consumption, and strong developer community support, making it superior to earlier Arduino or Raspberry Pi-based systems [10]. The system periodically transmits vehicle location data to a cloud server and visualizes it through a web-based digital map interface accessible to rental owners. Beyond real-time monitoring, the system enables historical data analysis, including route patterns, usage duration, and operational efficiency, thereby supporting strategic fleet management decisions.

The primary contribution of this research lies in presenting a cost-effective, energy-efficient, and scalable IoT-based GPS tracker specifically designed for rental vehicles. Unlike prior systems, the proposed solution integrates GPS and Wi-Fi connectivity directly within the ESP-32 platform, eliminating the need for external modules and reducing system complexity. The system also incorporates cloud-based visualization through Google Maps API, enabling both real-time monitoring and historical data analytics. This dual capability enhances transparency, improves asset security, and supports data-driven decision-making in fleet management.

The significance of this research extends beyond technical innovation. For the vehicle rental industry, particularly small and medium enterprises in emerging economies, the proposed system offers a practical and affordable solution to long-standing operational challenges. By improving fleet transparency and security, the system reduces financial risks associated with theft and misuse. By enabling historical data analysis, it empowers rental companies to optimize operations, improve customer service, and make strategic decisions based on evidence rather than intuition. Furthermore, the system aligns with broader information and communication technology (ICT) and IoT applications, contributing to the advancement of smart transportation solutions and reinforcing the role of IoT in modern fleet management. Its scalability and modularity also make it adaptable to other domains such as logistics, public transportation, and smart city infrastructure.

2. METHOD

This section outlines the methodological framework adopted to design and implement the IoT-based GPS tracker system for rental vehicles. The research follows an experimental approach combined with system engineering principles, ensuring that both hardware and software components are developed, integrated, and tested systematically. The methodology begins with identifying system requirements, continues through hardware and software design, and culminates in implementation and evaluation. By structuring the process in this way, the study ensures that the proposed solution is not only technically functional but also efficient, reliable, and adaptable to real-world conditions.

2.1. Tools and materials

The development of the GPS tracker system requires a combination of hardware and supporting devices carefully selected to meet the criteria of energy efficiency, compactness, and compatibility with IoT applications. The NodeMCU ESP-32 microcontroller was chosen as the core of the system due to its integrated Wi-Fi and Bluetooth capabilities, dual-core processing power, and low energy consumption, making it ideal for continuous monitoring in vehicle environments. The NEO-6M GPS module provides accurate location data,

while auxiliary components such as jumper cables, power banks, and smartphones ensure seamless connectivity, portability, and user interaction. The complete list of tools and materials used in this research is presented in Table 1, which summarizes the specifications and quantities of each component employed in the system design and testing.

Table 1. Tools and materials

No	Tools	Specifications	Quantity
1	NodeMCU ESP32	CPU/Processor: Tensilica Xtensa LX6 32bit Dual-Core at 160/240MHz SRAM: 520 KB FLASH: 2MB (max. 64MB) Operating voltage: 2.2 V to 3.6 V Average current: 80 mA	1
2	Modul GPS Neo-6M	Receiver type: 50 channels, GPS L1 (1575.42Mhz) Horizontal positioning accuracy: 2.5m Navigation update rate: -161dBm Navigation sensitivity: Operating voltage: 2.7V–3.6V	1
3	Jumper cables	Male to male, female to female. Size: 10/15 cm	4
4	Powerbank	10.000 mAh	1
5	Black box	-	1
6	Smartphone	Android/iOS support Telegram	1
7	Usb cable	-	1

2.2. Research procedure

The research was carried out using an experimental approach combined with IoT-based system engineering principles. This approach was chosen to ensure that the GPS tracker system could be developed, tested, and validated under realistic operational conditions. The procedure was designed to follow a structured sequence of activities, beginning with the identification of system requirements and progressing through design, implementation, and evaluation. Each stage was carefully planned to guarantee that the final system would meet the objectives of accuracy, energy efficiency, and real-time monitoring capability. The research procedure consisted of several systematic stages, which together provided a comprehensive framework for developing and assessing the IoT-based GPS tracker system:

a. Literature study

Conducting a literature review of various references related to:

- IoT technology and vehicle monitoring system architecture,
- NodeMCU ESP-32 specifications and programming,
- GPS module integration (NEO-6M),
- Digital mapping platforms such as Google Maps API or Blynk.

b. System requirements identification

Determine functional and non-functional requirements, including:

- Hardware requirements (NodeMCU ESP-32, GPS NEO-6M, Wi-Fi module, power supply),
- Software requirements (Arduino IDE, web dashboard/API),
- Real-time location data delivery and display scheme.

c. System design

1. Hardware design

- Assembling NodeMCU ESP-32 with NEO-6M GPS module and power source (e.g., power bank or lithium battery).
- Installing the device to the simulation vehicle.

2. Software design

- NodeMCU programming using Arduino IDE to read GPS data and send it via Wi-Fi to the server platform.
- Design of web-based user interface using digital maps (Google Maps API/Blynk/ThingSpeak).

d. System implementation

- Uploading the program into the NodeMCU ESP-32,
- Connecting the system to a Wi-Fi network,

- Testing the data connection between the device, server, and interface display.
- e. System testing and evaluation
 1. System function testing
 - Testing the accuracy of GPS data compared to actual location.
 - Testing the speed and stability of data delivery.
 - Evaluating the display of location data on the web or application in real-time.
 2. Performance evaluation
 - Observing the power consumption of the device when active.
 - Calculating the time delay between data being obtained and being displayed.
 - Test the system for 24 hours non-stop under various network conditions.
- f. Documentation and analysis of results
 1. Record all test results
 - Analyze the advantages and limitations of the system,
 - Provide recommendations for future system development.
 2. Preparation of research reports
 - Prepare a final report containing background, methods, results, analysis, conclusions, and suggestions,
 - Prepare documents for publication or further testing commercially or academically.

2.3. Block diagram

To illustrate the overall architecture of the proposed IoT-based GPS tracker system, a block diagram was developed to represent the interaction between hardware components, data flow, and user interface integration. The block diagram provides a high-level overview of how the system operates, starting from the acquisition of satellite signals to the visualization of vehicle location data on a digital platform. As shown in Figure 1, the NEO-6M GPS module functions as the primary input device, receiving signals from satellites and generating location coordinates (longitude and latitude). These raw data are then transmitted to the NodeMCU ESP-32 microcontroller, which serves as the central processing unit. The ESP-32 executes programmed instructions to process the GPS data, manage communication protocols, and ensure stable transmission through its built-in Wi-Fi connectivity.

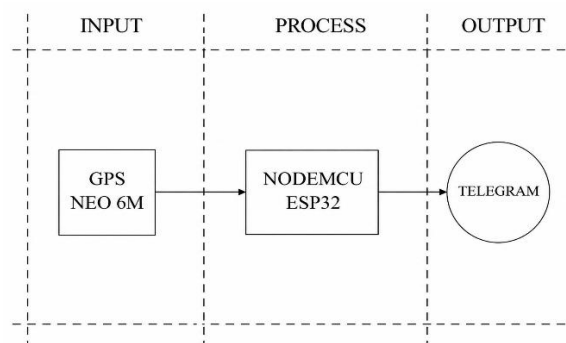


Figure 1. GPS tracker automation system block diagram

Once processed, the location information is sent to a cloud-based server and subsequently integrated with a user interface. In this study, the system was connected to the Telegram application, which acts as the medium for user interaction. Through Telegram, users can send commands to the system, receive coordinate links, and access real-time location updates displayed via Google Maps. This integration ensures that vehicle owners can monitor their assets remotely, efficiently, and securely. The block diagram highlights three critical aspects of the system:

- Data acquisition: GPS module captures satellite signals and generates accurate positioning data.
- Data processing and transmission: ESP-32 processes GPS data and transmits it via Wi-Fi to the server.
- User interaction and visualization: Telegram bot and Google Maps API provide real-time monitoring and visualization of vehicle location.

By presenting the system in this structured manner, the block diagram clarifies the logical flow of information and demonstrates how hardware and software components collaborate to achieve reliable IoT-based vehicle tracking.

2.4. Flowchart of work tools

To complement the block diagram presented earlier, a flowchart was developed to describe the logical sequence of operations within the GPS tracker system. While the block diagram (Figure 1) illustrates the structural relationship between hardware and software components, the flowchart provides a step-by-step representation of how the system functions in practice. It highlights the interaction between the user, the GPS tracker device, and the Telegram application, showing how commands are processed and how location data is delivered in real time. As shown in Figure 2, the process begins with the initialization of all hardware components, followed by user authentication through the Telegram application. The user sends specific commands to the Telegram bot, which triggers the GPS tracker to retrieve and transmit location coordinates. If the connection is stable, the system successfully delivers a coordinate link that can be opened in Google Maps, allowing the user to monitor the vehicle's position. In cases where the link is not sent, the flowchart also outlines corrective actions such as checking the connection and refreshing the application. This flowchart is essential because it demonstrates the operational logic of the system in a way that is intuitive and easy to follow. It ensures that users and developers alike can understand how the GPS tracker responds to inputs, manages communication, and delivers outputs. Moreover, it provides a foundation for analyzing system performance, identifying potential bottlenecks, and ensuring that the tool can be reliably programmed and deployed in real-world scenarios.

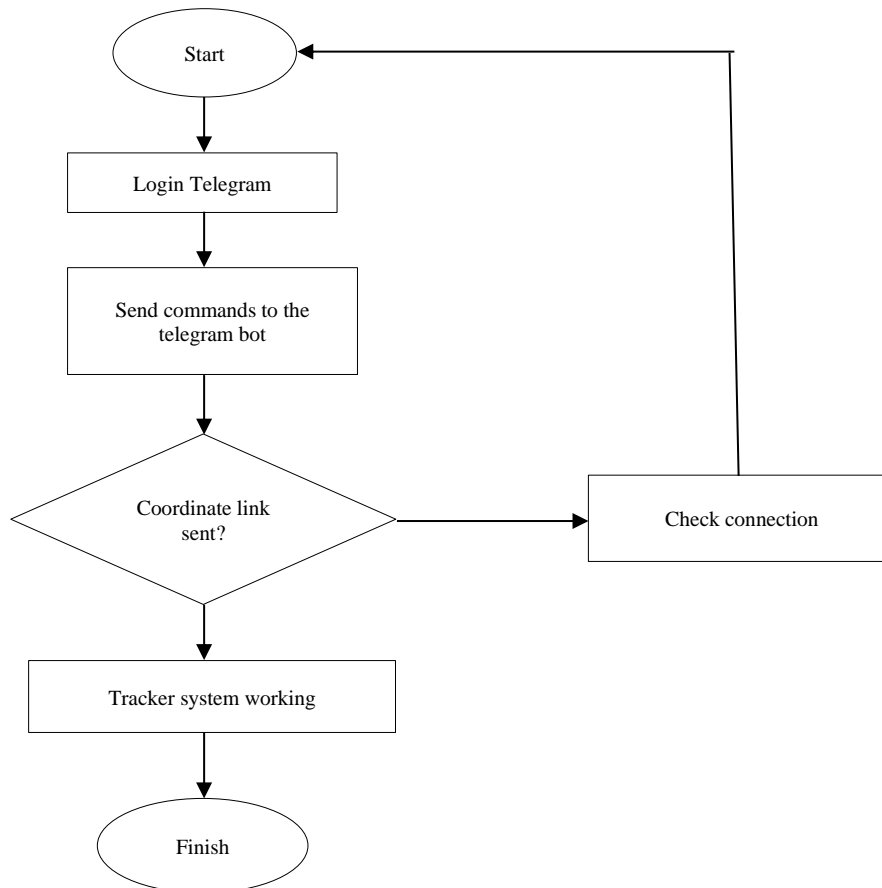


Figure 2. Work tool flowchart

The explanation of the flowchart in Figure 2 is as:

- First step, turn on all components.
- Second stage, Login to the Telegram application on your smartphone.
- Third stage, send a request for a coordinate link to the Telegram bot.
- Fourth stage, if the coordinate link is sent, then the process is continued to the next stage. If the coordinate link is not sent, then check the connection and refresh the Telegram application. Then log back into the Telegram application.

- Fifth stage, the system is said to be successful if the location of the GPS Tracker can be monitored through the link on the Telegram bot.
- Six stage, if all components run normally as desired. Thus, the tool is declared successfully programmed.

2.5. Position and advantages of NodeMCU ESP-32

The NodeMCU ESP-32 is a next-generation microcontroller solution that offers Wi-Fi and Bluetooth Low Energy (BLE) integration on a single chip, with much lower power consumption than a Raspberry Pi or Arduino when used with additional modules.

Some of the key advantages of the ESP-32 that make it superior and fill the void left by previous solutions include:

- Cost and energy efficiency: ESP-32 is designed for low power consumption with deep sleep mode, which is perfect for battery-based applications and periodically active vehicle systems.
- Cloud integration and MQTT protocol: thanks to the built-in Wi-Fi connectivity, the ESP-32 can directly connect to MQTT or Firebase servers without additional modules.
- Low system complexity: unlike Arduino which requires additional modules, ESP-32 integrates all communication and processing functions in a single board.
- Community and library support: the platform is supported by an extensive developer community and is compatible with various GPS libraries (such as TinyGPS++) and IoT platforms (such as Blynk, ThingsBoard, and Node-RED).
- Affordable price and compact form factor: the ESP-32 has a relatively low price and smaller size than the Raspberry Pi, making it ideal for vehicle applications with space.

By integrating a GPS module (e.g., Neo-6M) and using ESP-32 as the core of the system, this article presents a solution that is not only cost-effective, but also scalable, modular, and ready to connect with cloud-based monitoring systems in the IoT ecosystem.

3. RESULTS AND DISCUSSION

The IoT-based GPS tracker system developed in this study was successfully implemented using the NodeMCU ESP-32 microcontroller in combination with the NEO-6M GPS module, enabling accurate detection of vehicle position and real-time transmission of location data to a cloud-based server. This integration aligns with recent advances in IoT-enabled fleet monitoring, where GPS modules combined with microcontrollers have been shown to improve operational transparency and asset security [9]-[13]. The processed data were visualized through a web-based interface and accessed via the Telegram application, allowing continuous monitoring and user interaction, consistent with findings that user-friendly visualization platforms enhance decision-making in smart transportation systems [4], [6]. During implementation, the system maintained stable communication between the GPS module, ESP-32, and cloud platform, ensuring location updates at regular intervals, which confirms the reliability of the chosen hardware-software configuration. Beyond basic tracking, the system supported advanced features such as historical data storage, route visualization, and mobile interaction, addressing gaps noted in prior studies where systems lacked scalability or efficient cloud integration [1], [2]. By combining real-time monitoring with energy-efficient hardware, the proposed solution contributes to overcoming critical challenges in the vehicle rental industry, including theft prevention, misuse reduction, and cost efficiency, while demonstrating broader significance for logistics and smart city applications, [14]-[17].

3.1. System functionality

The system is designed to automatically: retrieve location coordinate data (longitude and latitude), send data periodically every 10 seconds via a Wi-Fi network, display vehicle location on a digital map through integration with the Google Maps API, save historical vehicle location data to a cloud-based database (ThingSpeak). Testing is done by placing the device on a vehicle moving on a certain route along ± 10 km in an urban area [13], [18].

In the GPS Tracker automation system on automotive vehicles based on the IoT using NodeMCU ESP-32, it is programmed using the Arduino IDE. The GPS tracker system is programmed by providing input in the form of a script as shown in Figure 3. The wiring design can be seen in Figure 3. In the GPS Tracker system testing, the application media from BotTelegram is used as an interface from the system to the user. In Telegram, there are commands that must be executed first to turn on the system. First, you must send the command “/start” to the Telegram bot which functions to connect the smartphone to the device. Then, the bot will send a message containing “command/track” and the command must be sent back by the user to the bot. After that, the bot will send a message back with a link containing the coordinate points from the GPS tracker and the user can find out the location of the device in real time, as shown in Figure 4.

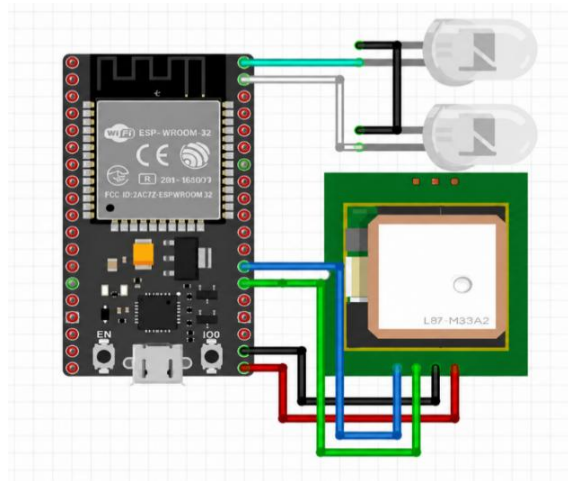


Figure 3. GPS tracker system wiring design



Figure 4. Telegram GPS tracker bot view

In modern IoT-based fleet monitoring systems, the ability to visualize location data in real time is essential for ensuring operational efficiency and asset security. Traditional GPS trackers often provide raw coordinate outputs that are difficult for users to interpret without integration into mapping platforms. Recent studies have highlighted the importance of combining GPS modules with cloud-based visualization tools such as Google Maps, which enable interactive monitoring and improve decision-making in transportation and rental industries [9]-[11], [18], [19]. Despite these advances, many existing solutions still face challenges related to power consumption, hardware complexity, and limited user-friendly interfaces [4], [6].

The system developed in this study addresses these limitations by integrating the NEO-6M GPS module with the NodeMCU ESP-32 microcontroller, transmitting location data via Wi-Fi, and visualizing it directly through Google Maps accessed via the Telegram application. The link generated by the system contains precise geographic coordinates, which users can open to view the vehicle's position in real time on a digital map interface. This integration not only ensures accuracy and reliability but also enhances usability by providing an interactive representation of vehicle movement. As shown in Figure 5, the test results confirm that the system delivers timely updates and stable visualization, validating its effectiveness as a modern IoT-based solution for rental vehicle monitoring [20].

The contribution of this research lies in bridging the gap between raw GPS data acquisition and actionable visualization. Its significance extends to improving transparency in fleet operations, reducing risks of misuse or theft, and enabling data-driven decision-making in the vehicle rental industry. Furthermore, the system demonstrates scalability for broader applications in logistics, smart transportation, and urban mobility solutions.

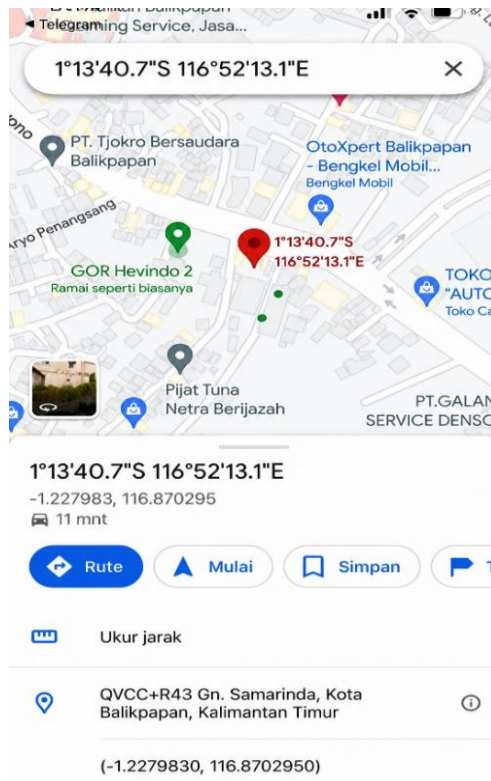


Figure 5. Bot Telegram GPS tracker view

4. CONCLUSION

This study successfully designed and implemented an IoT-based GPS tracker system for rental vehicles using the NodeMCU ESP-32 microcontroller and the NEO-6M GPS module. The system integrates hardware and software components to provide accurate, real-time monitoring of vehicle location, with data transmitted via Wi-Fi and visualized through a cloud-based interface. The use of the Telegram application as the user interface proved effective, allowing rental owners to interact with the system by sending simple commands such as /track to the bot. In response, the bot generated a link containing GPS coordinates, which could be opened directly in Google Maps to display the precise location of the vehicle. The testing results confirmed that the system operates reliably, delivering location updates at regular intervals and maintaining stable performance under continuous operation. The integration of ESP-32 ensured low power consumption and reduced system complexity compared to traditional Arduino or Raspberry Pi solutions, while the use of cloud-based visualization enhanced accessibility and transparency. Overall, the proposed GPS tracker system demonstrates significant potential as a cost-effective and scalable solution for the vehicle rental industry. It improves asset security, reduces risks of misuse or theft, and supports data-driven decision-making through historical route analysis. Beyond rental applications, the system can be adapted for broader fleet management, logistics, and smart transportation initiatives, highlighting its relevance within the growing ecosystem of IoT-based monitoring technologies.

ACKNOWLEDGEMENTS




We would like to express deepest gratitude to the Universitas Muhammadiyah Sumatera Utara and the LPPM of the Universitas Muhammadiyah Sumatera Utara who have supported the research and publication of this paper.

REFERENCES




- [1] L. Atzori, A. Iera, and G. Morabito, "The Internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010, doi: 10.1016/j.comnet.2010.05.010.
- [2] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A Survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020, doi: 10.1109/COMST.2019.2962586.
- [3] D. C. Mahendra, T. Susyanto, and S. Siswanti, "A car rental monitoring system using a GPS tracker," (in Indonesian), *Jurnal Ilmiah SINUS (JIS)*, vol. 16, no. 2, pp. 37–46, Aug. 2018, doi: 10.30646/sinus.v16i2.357.
- [4] M. Yakubjanova, "Youth Media Culture Problems and Solutions," *International Journal of Trend in Scientific Research and Development (IJTSRD)*, Special Issue: Innovative Development of Modern Research, pp. 124–125, Apr. 2021.
- [5] E. Daniel and N. A. Vasanthi, "ES-DAS: An Enhanced and Secure Dynamic Auditing Scheme for Data Storage in Cloud Environment," *Journal of Internet Technology*, vol. 21, no. 1, 2020, doi: 10.3966/160792642020012101016.
- [6] K. H. Umarov, "Intellectual Youth and Media Culture: Problems and Solutions," *Oriental Art and Culture Scientific-Methodical Journal*, no. 2, pp. 190–194, 2020. Online. Available: <https://cyberleninka.ru/article/n/intellektual-yoshlar-va-media-madaniyat-muammo-va-echimlar/viewer>.
- [7] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer, M. Balazinska, and G. Borriello, "Building the Internet of Things Using RFID: The RFID Ecosystem Experience," *IEEE Internet Computing*, vol. 13, no. 3, pp. 48–55, May–Jun. 2009, doi: 10.1109/MIC.2009.52.
- [8] A. Dada and F. Thiesse, "Sensor Applications in the Supply Chain: The Example of Quality-Based Issuing of Perishables," in *The Internet of Things*, C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern, and S. E. Sarma, Eds. Berlin, Germany: Springer, 2008, pp. 140–154, doi: 10.1007/978-3-540-78731-0_9.
- [9] H. Y. Youm, "An Overview of Security and Privacy Issues for Internet of Things," *IEICE Transactions on Information and Systems*, vol. E100-D, no. 8, pp. 1649–1662, 2017, doi: 10.1587/transinf.2016ICI0001.
- [10] T. V. Landegem and H. Viswanathan, *Anywhere, Anytime, Immersive Communications*, Enriching Communications, vol. 2, no. 1, 2008.
- [11] A. Alfith, A. Effendi, A. Al, and R. Maulana, "Rental Car Tracking and Monitoring System Uses GPS Sensors Based on NodeMCU ESP32," *Journal of Technology and Vocational Studies*, vol. 2, no. 2, pp. 84–96, Jul. 2024, doi: 10.21063/jtv.2024.2.2.84-96.
- [12] H. Hamsir, A. Sahara, F. Oktafiani, and D. P. Ariyani, "GPS Tracker Automation System for Rental Cars Based on the Internet of Things (IoT) Using NodeMCU ESP-32," (in Indonesian) *PETROGAS: Journal of Energy and Technology*, vol. 5, no. 2, pp. 10–19, Oct. 2023, doi: 10.58267/petrogas.v5i2.153.
- [13] A. I. Yaqzan, I. W. Damaj, and R. N. Zantout, "GPS-Based Vehicle Tracking System-on-Chip," *arXiv preprint arXiv:1904.09838*, 2019.
- [14] N. P. Simanjuntak, M. A. Rahman, A. N. Pratama, S. Stelyven, and A. Yulianto, "Design and Implementation of an IoT-Based Vehicle Security System Using ESP32, FastAPI, and an Android Application," (in Indonesian) *Telcomatics*, vol. 10, no. 1, pp. 73–81, 2025, doi: 10.37253/telcomatics.v10i1.10870.
- [15] S. Rohman and Z. Amalia, "Design and Development of a Monitoring and Safety System with Live Tracking Features for Rental Cars," (in Indonesian) *Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer*, vol. 5, no. 2, pp. 171–183, Jun. 2025, doi: 10.51903/juritek.v5i2.4548.
- [16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.
- [17] N. Mohamed and J. Al-Jaroodi, "Real-Time Big Data Analytics for Internet of Things," *Journal of Cloud Computing*, vol. 8, no. 1, pp. 1–15, 2019.
- [18] K. P. Lestari, L. Lindawati, and Sholihin, "Implementation of NodeMCU ESP8266 in a Vehicle Tracking Device Using Global Positioning System and MiFi-Based Internet of Things (IoT)," (in Indonesian) *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 2, pp. 229–236, Apr. 2023, doi: 10.32493/jtsi.v6i2.32558.
- [19] H. Saghaei, "Design and Implementation of a Fleet Management System Using Novel GPS/GLONASS Tracker and Web-Based Software," in *Proceedings of the 1st International Conference on New Research Achievements in Electrical and Computer Engineering*, Tehran, Iran, 2016.
- [20] V. Barral Vales, O. C. Fernández, T. Domínguez-Bolaño, C. J. Escudero, and J. A. García-Naya, "Fine Time Measurement for the Internet of Things: A Practical Approach Using ESP32," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18305–18318, Oct. 2022, doi: 10.1109/JIOT.2022.3158701.

BIOGRAPHIES OF AUTHORS






Indah Purnama Sari    was born in Medan, Indonesia, in 1990. She is a lecturer at the Faculty of Computer Science and Information Technology at the Universitas Muhammadiyah Sumatera Utara (UMSU). She earned her master's degree from the Universitas Sumatera Utara in 2019 in Computer Science. Her main research interests are digital imagery, internet of things, data science, big data, machine learning, neural networks, and artificial intelligence. She can be contacted at email: indahpurnama@umsu.ac.id.



Al-Khowarizmi    was born in Medan, Indonesia, in 1992. He is a dean in Faculty of Computer Science and Information Technology at Universitas Muhammadiyah Sumatera Utara (UMSU). He got doctoral degree from Universitas Sumatera Utara in 2023 field Computer Science. His main research interest is data science, big data, machine learning, neural network, artificial intelligence, and business intelligence. He can be contacted at email: alkhowarizmi@umsu.ac.id.



Asrar Aspia Manurung    was born in Air Batu, Indonesia, in 1985. He is a lecturer at the Faculty of Computer Science and Information Technology at the University of Muhammadiyah North Sumatra (UMSU). He earned his master's degree from the State University of Medan in 2013 in Mathematics Education. His main research interests are Internet of things, data science, big data, artificial intelligence, computer programming, and educational statistics. He can be contacted at email: asraraspia@umsu.ac.id.