Journal of Engineering, Electrical and Informatics

E- ISSN: 2809-8706
P- ISSN: 2810-0557

Research Article

Design of an IoT-Based Airside Vehicle Tracking System Using GPS–LoRa with an Early Warning Feature at Class III Maimun Saleh Airport

Lulu Salsitiara[1*], Muh Wildan[2], Johan Wahyudi[3]

[1-3] Program Studi Teknik Navigasi Udara, Politeknik Penerbangan Indonesia Curug, Indonesia.
*Corresponden email: lulusalsitiara22@gmail.com

Abstract: This research aims to design and implement an Internet of Things (IoT)-based airside vehicle mon-itoring system at UPBU Maimun Saleh Sabang. The main problem addressed in this study is the absence of a digital monitoring system for operational vehicles, which causes airside supervision to rely solely on manual observation and increases the risk of safety area violations. The proposed system utilizes an ESP32 microcontroller as the main controller, a GPS module for determining vehicle positions, and LoRa communication as an efficient long-range data transmission medium. Vehicle position data are transmitted using the MQTT protocol and visualized in real time through a web-based dashboard. In addition, LED and buzzer indicators function as early warning systems to alert operators when vehicles exceed operational boundaries (geofencing) or predefined speed limits. This study adopts the Research and Development (R&D) method, which includes stages of problem identification, data collection, product design, design validation, design revision, and product testing. The test results show that the system is capable of displaying vehicle positions with an accuracy of approximately ±2 meters and an average latency of 1.5 seconds. Furthermore, the early warning indicators operate effectively in detecting violations of operational area boundaries and speed limits, making the system suitable as an initial prototype to improve safety and efficiency in monitoring airside vehicle movements at Maimun Saleh Airport Sabang.

Keywords: Airside; Aviation Safety; ESP32; Geofencing; GPS.

Received: January 17, 2026
Revised: February 12, 2026
Accepted: February 25, 2026
Published: February 28, 2026
Curr. Ver.: February 28, 2026

1. Introduction

The growth of air traffic demands increasingly efficient and safety-oriented airport operational management. One of the areas with a high level of operational risk is the airside (Riska & Maulana, 2023), which includes the Runway, Taxiway, Apron, and operational vehicle movement routes. In this area, various vehicles and Ground Support Equipment (GSE) operate to support flight activities, therefore an adequate management and monitoring system is required to ensure safety and operational continuity. Safety in the airside area must be maintained through the control of vehicle and personnel movements so that the potential for incidents that could disrupt aircraft operations can be minimized (Bhoka & Sutarwati, 2024).

The ideal condition recommended in airport management is the availability of an integrated and real-time monitoring system for airside operational vehicles (Hubbard &

Hubbard, 2022). Such a system enables supervision of vehicle movement, speed, and compliance with designated operational zones. An approach based on the Internet of Things (IoT) through the utilization of global positioning sensors (Global Positioning System/GPS), geofencing, and digital vehicle identification allows vehicle movement data to be transmitted continuously to the apron control center. Thus, supervision is not only administrative in nature but also supports early detection of potential violations or unsafe conditions. The implementation of this system is in line with the concept of the Safety Management System (SMS), which requires airport operators to ensure operational safety throughout the airside area (Law Number 1 of 2009 concerning Aviation), and every vehicle operating in the movement area must be under the supervision of the airport operator in order to guarantee security, safety, and traffic smoothness on the airside (Directorate General of Civil Aviation, 1999).

Based on observations and operational conditions in the field, it was found that UPBU Class III Maimun Saleh has not yet been equipped with an integrated IoT-based monitoring or tracking system for apron vehicles. The supervision of vehicle movements is still conducted through visual observation and manual recording by field officers. This monitoring pattern causes limitations in obtaining real-time vehicle position information, thereby increasing the potential for operational violations and delays in detecting safety risks.

These monitoring limitations are reflected in an incident involving the collision with a taxiway edge light facility at the taxiway turn toward runway 28 on June 5, 2024, as reported through a letter from the Head of UPBU Class III Maimun Saleh to the Head of the Aviation Technical Center Number AU.107/01/VII/MUS-2024 dated July 2, 2024, regarding damage to the Airfield Control and Operation System (ACOS), Runway and Taxiway Indicator Light (RTIL), and Airfield Lighting Facilities (AFL). The incident was classified as a minor operational incident because it did not cause damage to aircraft or flight operations; however, it resulted in damage to airside facilities that required repair actions. Based on the report, this incident caused a temporary disruption to the function of visual navigation facilities. Although no flight cancellations were reported, the sequence of events indicates the potential for safety risks and operational losses if similar incidents occur at a higher level.

In addition, airport operational data as stated in the Airport Capacity Determination Report or Notice of Airport Capacity (NAC) of UPBU Maimun Saleh for the Winter 2024/2025 and Summer 2025 periods indicates that the utilization level of the apron during certain hours has approached the established capacity limit. This condition indicates an increase in the intensity of aircraft and supporting vehicle movements on the airside. This situation shows a gap between the ideal conditions expected by regulations and the operational practices occurring in the field (Creswell, 2014), therefore a technology-based solution is required to bridge this gap.

Based on these problems, this study designs an Internet of Things (IoT)-based airside operational vehicle monitoring system capable of tracking vehicle positions in real time using a combination of Global Positioning System (GPS) and LoRa technology. Vehicle position data are transmitted wirelessly to a local server or digital mapping system to be displayed at the control center. Each vehicle is equipped with an IoT prototype device that functions both as a monitoring tool and as a provider of direct feedback to the driver.

The designed system is equipped with an early warning feature for potential operational violations, namely vehicle speeds exceeding the established limits and violations of operational area boundaries through a geofencing mechanism. If a vehicle is detected leaving the designated area, the system will trigger a warning in the form of visual indicators (lights) and audio indicators (buzzers) as direct alerts to the driver to take corrective action. This warning feature acts as an initial preventive barrier within the framework of the Safety Management System (SMS). Therefore, the implementation of an IoT-based airside vehicle monitoring system is expected to improve safety levels, reduce the potential for human error, and support airport operational efficiency (Ministry of Transportation of the Republic of Indonesia, 2023).

Research conducted by Ramli et al., 2019 developed a LoRa and GPS-based vehicle tracking system using an open-source platform capable of performing long-distance vehicle position tracking with low power consumption. However, the system still focuses on position tracking without being equipped with early warning features and has not been applied in restricted areas such as airport airside. Furthermore, Murillo et al., 2020 evaluated the performance of LoRa networks in Internet of Things (IoT)-based vehicle tracking services within intelligent transportation systems. The results showed reliable RSSI and Packet Delivery Ratio (PDR) values for vehicle tracking services, but the study focused more on network performance evaluation and did not integrate aspects of vehicle operational safety.

Another study conducted by Pratama, 2025 implemented a LoRa system using the GNSS Neo-8M module and ESP32 for real-time mountain climber tracking with good power efficiency in difficult terrain. Fajaryasa, 2022 designed a vehicle location data transmission system on the apron based on GPS and ESP32 capable of displaying vehicle position information to support coordination between Apron Movement Control and Air Traffic Control; however, the system was still limited to presenting location information without early warning features or geofencing. Meanwhile, Wisnuardana, 2025 developed a prototype airside vehicle movement tracking system based on GPS and LoRa integrated with a web dashboard, capable of displaying vehicle movement in real time with high coordinate accuracy and stable communication. Therefore, this study was conducted to fill the gap in previous research by designing an IoT-based airside vehicle tracking system using GPS–LoRa that not only performs real-time position tracking but also integrates early warning features and geofencing-based area restrictions as a preventive barrier in accordance with the Safety Management System (SMS) concept within the airport airside environment.

Therefore, this study was prepared to fill the research gap identified in previous studies, particularly regarding the lack of integration of airside vehicle monitoring systems with early warning features and operational area restrictions as an effort to prevent safety risks. This study is entitled "Design of an IoT-Based Airside Vehicle Tracking System Using GPS-LoRa and Early Warning Features at UPBU Class III Maimun Saleh," which aims to design a prototype of an Internet of Things (IoT)-based airside vehicle monitoring system utilizing GPS–LoRa technology operating locally.

2. Literature Review

Studies on airport operations show that airports are important nodes in the air transportation system, functioning as connectors for the mobility of people and goods (Akbar et al., 2024). According to international standards established by ICAO through the Chicago Convention of 1944, airport areas are divided into two main zones, namely landside and airside (Saputra, 2024). The airside area is a restricted zone with the highest level of safety risk because it involves the simultaneous movement of aircraft, operational vehicles, and personnel (Riska & Maulana, 2023). Therefore, the movement of operational vehicles in this area must be strictly controlled through operational area restrictions, speed regulation, and continuous monitoring to prevent movement conflicts and operational accidents (ICAO, 2021).

In the context of aviation security, controlling vehicles and personnel operating within the security restricted area is an important part of the aviation security system recommended by ICAO (2017). Vehicles that move without authorization or violate operational area boundaries have the potential to cause security disturbances as well as aviation safety risks. Manual monitoring by security personnel often has limitations, especially at airports with limited resources. Therefore, the utilization of technology-based monitoring systems such as the Internet of Things (IoT) becomes a relevant solution to improve real-time monitoring capabilities and provide early warnings for potential operational violations in the airside area (ICAO, 2023).

Several studies also emphasize the importance of surface movement monitoring systems at airports through the implementation of the Surface Movement Guidance and Control System (SMGCS) recommended by ICAO (Davis, 2000; Ashton, 2010). This system integrates surveillance, guidance, and control functions for the movement of aircraft and operational vehicles on the apron, taxiway, and runway. Although SMGCS is a complex and high-standard safety system, academic research often develops alternative approaches based on the Internet of Things to enhance situational awareness of operational vehicle movements in real time, especially in airport environments with limited infrastructure.

In the development of IoT-based monitoring systems, several key technologies are commonly used, including microcontrollers, GPS modules, wireless communication, and database systems for data storage. Microcontrollers such as the ESP32 are widely used in IoT systems because they have sufficient processing capabilities and support network connectivity (Moumen et al., 2023). Vehicle position data can be obtained using Global Positioning System (GPS) or GNSS technology, which can provide accurate location information. Meanwhile, long-distance data communication can utilize LoRa technology, which belongs to the Low Power Wide Area Network (LPWAN) category and is capable of transmitting data with low power consumption and wide coverage (Wickens & Hollands, 2000; Madugalla et al., 2025).

3. Research Method

This study employed a laboratory experimental method to analyze the effect of liquid latex addition on the Marshall characteristics of Asphalt Concrete–Wearing Course (AC-WC) mixtures. The mix design and all testing procedures were conducted in accordance with the 2018 General Specifications for Road Construction and Bridges (Revision 2 of 2020) issued by the Directorate General of Highways. Five variations of liquid latex content were used, namely 0%, 2.5%, 5%, 7.5%, and 10% by weight of asphalt, with three specimens prepared for each variation, resulting in a total of 15 test specimens. The Job Mix Formula (JMF) applied in this research was obtained from the laboratory of PT. Perwita Konstruksi Cirebon.

This research uses the Research and Development (R&D) method which focuses on the design and testing of a prototype airside vehicle monitoring system at UPBU Class III Maimun Saleh. This method was chosen because the research aims to produce a product in the form of a vehicle tracking device that can be used to monitor vehicle positions in real time. The system developed uses an ESP32 microcontroller, a Neo-6M GPS module as the source of coordinate data, and LoRa communication for long-distance data transmission.

The research process consists of several main stages, namely system requirements analysis, hardware and software design, design validation, design improvement, and prototype testing. The designed system functions to send vehicle position data from the device installed on the vehicle to the server through LoRa communication. The data is then stored and displayed in the monitoring system so that vehicle movements in the airside area can be monitored more effectively.

To evaluate the system performance, design criteria were established as a reference for assessing the success of the developed prototype. Testing was conducted on the system functionality, the reliability of data communication, and the system's ability to display vehicle position information in real time. The test results were then analyzed to assess the level of system functionality fulfillment, and a summary of the system performance achievements is presented in table form as an evaluation of the effectiveness of the developed airside vehicle monitoring system.

4. Results and Discussion

Overview of the System Design

The airside vehicle tracking system based on the Internet of Things (IoT) is designed to monitor the movement of operational vehicles at UPBU Class III Maimun Saleh in real-time. The system consists of a transmitter node installed on the vehicle using an ESP32 microcontroller and a GPS BN-220 module to obtain position, speed, and time data, and is equipped with a buzzer as an early warning when the vehicle speed exceeds the predetermined limit. The data is then transmitted through LoRa E220 communication to a receiver node using ESP32 in the airport control room, and subsequently forwarded to a local server via a Wi-Fi network to be stored in an SQL database. The stored data is visualized through a digital map-based monitoring dashboard so that operators can monitor vehicle position, speed, and movement history in real-time. This system is expected to improve monitoring effectiveness and support operational safety of vehicles in the airside area.

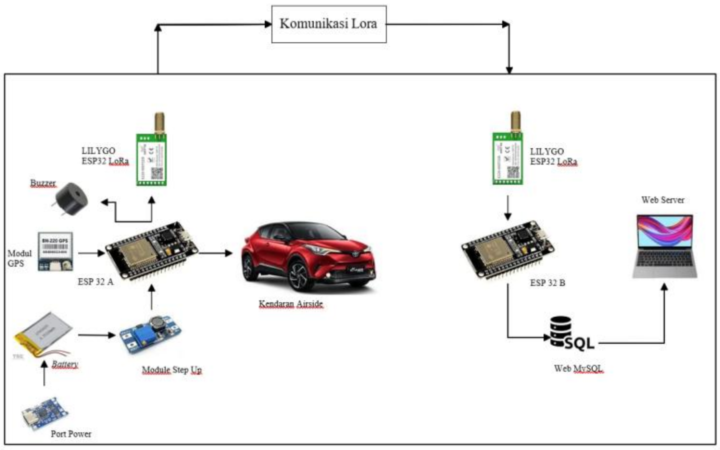

Figure 1. Overview of the System Design.

Design Stages

1. Requirement Analysis

At the requirement analysis stage, system requirements were identified in the development of the prototype vehicle tracking system in the airside area. The system is designed to detect vehicle positions in real-time with high accuracy and reliable communication connectivity, considering the operational conditions in the airside area which is wide and open. System requirements include hardware and software components that are integrated with each other to support the processes of data acquisition, transmission, processing, and visualization of vehicle position information.

On the hardware side, the system uses the LILYGO LoRa module as a long-range communication medium to send and receive vehicle location data with low power consumption. Vehicle position data acquisition is performed using the GPS BN-220 module, which is a Global Navigation Satellite System (GNSS) module supporting satellite systems such as GPS, Galileo, GLONASS, and BeiDou so that it can produce coordinate data in the form of latitude and longitude in real-time. The system power source comes from a portable battery installed on the LILYGO LoRa module so that the device can operate without dependence on an external power source. In addition, the ESP32 microcontroller is used as a data processing unit on the receiver side (gateway), which functions to receive vehicle position data via LoRa communication and forward it to the storage system through a Wi-Fi connection. To adjust the device voltage requirements, a step-up module is used which increases the battery voltage from around 3.5 volts to 5 volts to match the operating requirements of the ESP32. The system is also equipped with a buzzer as an early warning device that will activate when the vehicle exceeds the predetermined speed limit.

On the software side, the system uses MySQL as the database system to store all vehicle position data sent from devices in the field. The stored data is then used by the web dashboard system to display vehicle movement information in the form of a visual map in real-time. In addition, Notepad++ is used as a text editor to write and manage program code because it has a lightweight interface and supports various programming languages, thus facilitating the software development process in the designed vehicle tracking system.

2. System Design

At the system design stage, the technical design of the vehicle tracking system is carried out which includes preparing the hardware connection scheme and designing the data storage system in the MySQL database. This design aims to ensure that all device components can be well integrated so that the process of location data acquisition, data transmission, storage, and visualization of information can run systematically.

The hardware design on the transmitter side is carried out by connecting the GPS BN-220 module and the LoRa E220 communication module to the ESP32 microcontroller. The GPS BN-220 module functions to obtain vehicle coordinate data in the form of latitude, longitude, speed, and time in real-time, which is then processed by the ESP32. The data is then sent through LoRa communication to the receiving device. The system is also equipped with a buzzer connected to the ESP32 as an early warning device when the vehicle exceeds the predetermined speed limit. The system power source comes from a battery whose voltage is increased using a step-up module to match the voltage requirements of the ESP32.

On the receiver side, the device is designed using ESP32 connected to the LoRa E220 module to receive data sent from the vehicle unit. After the data is received through LoRa communication, the ESP32 performs the process of reading and forwarding the data to the server via Wi-Fi connection. The received data is then stored in a MySQL database so that it can be used for monitoring and analysis processes. With this design, the system is able to support the process of monitoring airside vehicle positions in real-time through a monitoring dashboard system.

Implementation

1. Hardware Assembly

Hardware assembly is carried out by arranging all electronic components used in the system according to the circuit design that has been prepared. This system consists of two main parts, namely the transmitter device installed on the vehicle side and the receiver device which functions as a data receiving gateway.

The transmitter device is designed to retrieve position data from the GPS module and send the data through the LoRa network. The main components used in this section include the ESP32 microcontroller integrated with the LoRa module, the GPS module, and several other supporting components assembled in a hardware box. The assembly process is carried out by connecting each component pin according to the predetermined configuration so that the system can work optimally. The results of the transmitter hardware assembly are shown in Figure IV.10.

Next, the hardware assembly is carried out on the receiver side. This device functions as a data receiver sent by the transmitter through LoRa communication. The receiver uses an ESP32 microcontroller that is also integrated with a LoRa module to receive data packets containing GPS information. The received data will then be processed and forwarded to the server via the internet network. The receiver hardware assembly process is carried out in the same way, namely by connecting all components according to the previously designed circuit scheme. The results of the receiver hardware assembly are shown in Figure IV.11.

2. Preparing Software

After the hardware assembly process is completed, the next stage is preparing the software used to run the system. The software design aims to program the microcontroller so that it can perform the main functions of the system, such as retrieving GPS data, transmitting and receiving data through the LoRa network, storing data in the database, and displaying the data on the web dashboard.

3. Arduino IDE Installation

Arduino IDE is used as the main software for programming the ESP32 microcontroller which functions as the main controller of the system. This software can be downloaded through the official Arduino website. After the download process is complete, the installation file is executed to begin installing the application on the computer.

After Arduino IDE is successfully installed, the next step is adding ESP32 board support so that the microcontroller can be recognized by Arduino IDE. This process is carried out by opening the Preferences menu in Arduino IDE and entering the ESP32 board repository address in the Additional Board Manager URLs field. After the address is entered, the process continues by accessing the Board Manager menu to search for the ESP32 by Espressif Systems package, then installing it until the process is complete.

In addition to board installation, the system also requires several additional libraries to support the programming process. Libraries in Arduino IDE are collections of functions that make it easier for developers to control hardware components. The library installation process is carried out through the Sketch menu, then selecting Include Library and Add .ZIP Library. Libraries used in this system include ArduinoJson, EByte LoRa, EEPROM, FS, HTTPClient, WebServer, WiFi, and WiFiClient.

4. XAMPP Installation

XAMPP is used as a local server environment that functions to manage Apache and MySQL services. In this system, MySQL is used as a database to store GPS data received from the receiver device before being displayed on the web dashboard.

The XAMPP installation process begins by downloading the installer file from the official Apache Friends website. After the installer file is executed, the user will be directed through several installation configuration stages such as selecting components to be installed, determining the installation directory location, and installing the program until completion. After the installation is successfully completed, the XAMPP Control Panel application can be run to activate the Apache and MySQL services required in the system.

5. Notepad++ Installation

Notepad++ is used as a text editor in developing program code for the server backend and web dashboard. This application is chosen because it is lightweight and supports various programming languages such as HTML, PHP, and JavaScript.

The Notepad++ installation process is carried out by downloading the installer file from the official Notepad++ website. After the installer is executed, the user follows the installation steps such as selecting the storage location and configuring components until the installation process is complete. After that, the application is ready to be used as an editor for web-based system development.

6. Microcontroller Programming

In order for the system to work as expected, the microcontroller must be programmed using the Arduino programming language. The program code is developed using Arduino IDE and then uploaded to the ESP32 board used in the system.

Before programming begins, the ESP32 board is first connected to the computer using a USB cable. After the connection is successful, the detected serial port is selected through the Tools > Port menu in Arduino IDE.

Programming on the transmitter side aims to configure the microcontroller to read data from the GPS module and send the data through the LoRa module to the receiver device. The transmitted data includes location coordinates, speed, altitude, and other parameters related to the vehicle position.

Meanwhile, programming on the receiver side aims to receive data sent by the transmitter through the LoRa network. After the data is received, the microcontroller processes the data and sends it to the server using the HTTP protocol via the WiFi network.

In addition, an additional program is also created to configure the WiFi connection and the server IP address. This configuration is required so that the receiver device can connect to the local network and send data to the prepared server.

7. MySQL Database Creation

The MySQL database is used to store vehicle tracking data sent by the receiver device. Database creation is performed through the phpMyAdmin application available in XAMPP.

The first step is running Apache and MySQL services through the XAMPP Control Panel. After the services are active, phpMyAdmin can be accessed through the address http://localhost/phpmyadmin/ in the browser. Through this interface, a new database is created with the name Tracking.

After the database is successfully created, the next step is creating a table that will be used to store GPS data. The table creation is performed using SQL commands through the SQL menu in phpMyAdmin. The created table contains several fields, including id, device_id, latitude, longitude, speed, heading, gps_time, and created_at. This table structure is designed to store all information related to vehicle tracking data.

8. Backend Server Development (PHP)

The backend server is developed using the PHP programming language which functions as a bridge between the receiver device, the database, and the web dashboard.

The first PHP file is created with the name receive.php which functions to receive GPS data sent by the receiver device via the HTTP protocol. This file then stores the received data into the previously created MySQL database.

In addition, a second PHP file is created with the name get_point.php which functions to retrieve GPS data from the database and send it to the web dashboard. This file uses the Server-Sent Events mechanism so that the dashboard can receive real-time data updates.

9. Web Dashboard Development

The final stage in system implementation is the development of a web dashboard that functions to display vehicle tracking data in the form of a visual map. The web dashboard is developed using the HTML programming language and utilizes the OpenStreetMap service as the visualization medium.

This dashboard is designed to display vehicle positions based on coordinates received from the database. The displayed data is updated periodically through the backend server so that users can monitor vehicle positions in real-time.

The dashboard file is saved with the name maps.html and placed in the local server directory of XAMPP. Dashboard testing is carried out by accessing the local server URL through a browser. If the system runs properly, the map will be displayed along with a marker indicating the vehicle position based on GPS data sent by the transmitter device.

Testing

1. Hardware Testing

Hardware testing was conducted to ensure that all electronic components used in the system function properly. This testing includes verification of the functionality of the GPS module and microcontroller on both the transmitter and receiver sides. After the GPS module was connected to the ESP32 on the transmitter side according to the circuit configuration, testing was performed to ensure that the device received sufficient power and was capable of acquiring GPS signals. The LED indicator on the GPS module lighting up indicates that the microcontroller has successfully completed the boot-up process and that the GPS module is active and ready to obtain location coordinate data.

2. Transmitter Testing

Testing of the transmitter unit was conducted to ensure that the assembled device could operate optimally before the data communication testing process was carried out. The device installed inside the protective box indicates that the hardware integration process has been completed. The external antenna connected via an SMA connector functions as a supporting component for wireless signal transmission in the LoRa module. In addition, the LED indicator on the GPS module lighting up indicates that the ESP32 microcontroller has received power and started running the uploaded program, so the device is ready to perform the GPS data acquisition process and data transmission.

3. Receiver Testing

Testing of the receiver unit was carried out by verifying the pinout connection of the LoRa module with the ESP32 board to ensure that each pin had been connected according to the circuit configuration. Device activation was performed by connecting the ESP32 board to the computer using a USB cable, which was indicated by the LED indicator lighting up as a sign that the device had received power. Next, the receiver program was run through Arduino IDE to verify the initialization process of the LoRa module and ensure that the receiving unit was ready to receive data packets sent from the transmitter unit before the data was forwarded to the server.

4. Software Testing

Software testing was conducted to ensure that all developed programs function according to the designed objectives. This testing includes verification of the GPS data acquisition process, data transmission through the LoRa network, sending data to the server, and storing data in the database. Successful software testing is indicated by the appearance of output data on the serial monitor and the system's ability to process and send data continuously.

5. Transmitter Program Testing

Testing of the program on the transmitter unit aims to ensure that the microcontroller can perform the function of GPS data acquisition and transmit the data through the LoRa network. The test results are shown through the output on the serial monitor displaying position coordinate data including latitude, longitude, number of satellites, altitude, recording time, and vehicle speed value. In addition, the appearance of the confirmation message “Data sent via LoRa" indicates that the GPS data has been successfully transmitted by the transmitter device in real-time.

6. Receiver Program Testing

Testing of the program on the receiver unit was conducted to ensure that the LoRa module can receive data sent by the transmitter unit and process the data before sending it to the server. The success of this process is indicated by the appearance of coordinate data on the serial monitor including latitude, longitude, number of satellites, altitude, recording time, RSSI value, and vehicle speed. In addition, the system also displays the message "Sending JSON to Server" with the server response status "200", which indicates that the process of sending data to the database through the backend server has been successfully completed.

7. Database Server Testing

Database server testing was conducted to verify the success of the data storage process sent by the receiver device. The test results show that the data sent in JSON format was successfully processed by the backend server and stored in the gps_data table in the MySQL database. Each row of data represents a record of vehicle position which includes parameters such as latitude, longitude, altitude, speed, and the number of GPS satellites. The correspondence between GPS reading time and data storage time on the server indicates that the data transmission and storage process occurs in real-time with low latency.

8. Web Dashboard Testing

Web dashboard testing was conducted to ensure that the system is able to display vehicle tracking data visually and informatively. The test results show that the dashboard successfully displays data through two main views, namely Map View and Table View. In the Map View display, GPS coordinate data is visualized using the OpenStreetMap map in the form of vehicle movement tracks and markers indicating the latest vehicle position. The information displayed includes recording time, latitude and longitude values, altitude, vehicle speed, and the number of connected satellites.

As a complement to the map display, Table View presents tracking data in the form of a structured table which includes GPS time, location coordinates, altitude, speed, and the number of satellites. In addition, the dashboard is equipped with a search feature, a refresh button to update data, and a CSV export feature that facilitates the process of data analysis and reporting of vehicle tracking data.

9. GPS Accuracy Testing

GPS accuracy testing was conducted by comparing the latitude and longitude coordinate data obtained from the GPS module on the device with coordinate data from the GPS on a smartphone device. The test was conducted at several location points by calculating the difference in coordinate distance between the two devices. The calculation results show that the obtained distance difference is approximately 0.712 meters. This value indicates that the system has a good level of accuracy because it is still within the GPS position error limit which is generally within a range of less than 5 meters.

10. LoRa Range Testing

LoRa communication range testing was conducted to determine the system's ability to transmit data at various distances and environmental conditions. The test results show that at close distances under Line of Sight (LOS) conditions, the RSSI value obtained indicates a very strong signal so that GPS data can be received stably. At farther distances up to several kilometers, the RSSI value decreases but the system is still able to receive and process data properly.

These results indicate that the LoRa module used has good reception sensitivity and is capable of performing data communication over quite long distances even under relatively weak signal conditions. GPS data can still be transmitted in real-time, stored in the database, and visualized on the web dashboard without significant interference.

11. Geofencing Testing

Geofencing feature testing was conducted to verify the system's ability to detect whether the vehicle is inside or outside the predetermined operational area. The system determines area boundaries based on certain geographical coordinates and calculates the vehicle's distance from the center point of the area in real-time. When the vehicle crosses the predetermined area boundary, the system automatically displays a warning on the serial monitor and activates the warning indicator.

In addition, the system also displays the vehicle status on the web dashboard through a marker color change. Vehicles inside the area are marked with green markers, while vehicles outside the area are marked with red markers. The test results show that the geofencing feature can detect area violations accurately without disrupting the GPS data transmission process or data storage in the database.

12. Buzzer Testing

Buzzer testing was conducted to validate the Early Warning System (EWS) function designed as a vehicle speed warning system. This system works based on speed data obtained from the GPS module in real-time. When the vehicle speed is still at a low level, the buzzer remains inactive. However, when the vehicle speed exceeds the predetermined limit, the system automatically activates the buzzer as a warning for the driver.

The test results show that the system is able to distinguish speed conditions accurately and activate the buzzer responsively. In addition, buzzer activation does not affect system stability because the data transmission process in JSON format continues to run properly and all GPS data remains stored in the MySQL database. Thus, the designed system not only functions as a vehicle tracking device but also as an active monitoring system that supports vehicle operational safety.

Analysis of Results

1. GPS Position Acquisition

Based on the test results, the system is able to obtain vehicle position coordinates in real-time with an average coordinate difference of 0.712 meters compared to the comparison device. This value is far below the GPS position accuracy tolerance limit which is $\leq$ 5 meters. Therefore, the GPS position acquisition aspect is considered to have been achieved very well, making the system suitable for monitoring vehicle positions in airport areas that require precise location accuracy.

2. GPS Reliability

The test results show that the GPS module is able to obtain satellite fixes stably with an average detection of 12 satellites during testing. This number exceeds the minimum standard requirement of detecting $\geq$ 5 satellites to obtain reliable position accuracy. Therefore, the GPS reliability in this system can be categorized as meeting the standard and capable of supporting continuous monitoring operations.

3. LoRa Data Transmission

LoRa data transmission testing shows that position data was successfully transmitted wirelessly with an RSSI value of –23 dBm at a distance of 10 meters and –101 dBm at a distance of 5 km. These values are still within the required LoRa sensitivity limit of ≥ –120 dBm. This indicates that the LoRa transmission system works optimally and meets the standards, particularly for long-distance communication needs in airport environments.

4. Communication Range

In terms of communication range, the system can still operate properly up to a distance of 5 km under line-of-sight (LOS) conditions with an RSSI value of –101 dBm. This distance exceeds the minimum communication range standard which generally lies within the range of 300–1000 meters. Therefore, the communication range aspect is considered achieved and highly adequate for vehicle monitoring in airport operational areas.

5. Data Integrity

The test results show that the transmitted position data is consistent and no data loss was found during the monitoring process. This condition meets the data integrity standard requiring ≥ 99% valid data. Therefore, the data integrity of this system can be categorized as achieved, ensuring that the received information can be trusted for operational monitoring purposes.

6. System Latency

Latency testing shows that the system is able to display data without significant time differences between data transmission and reception. This condition meets the near real-time system criteria with a latency limit of ≤ 10 seconds. Therefore, the system latency aspect is considered to meet the standard, supporting quick decision-making in the airside area.

7. Speed Warning System

In the speed warning system test, the buzzer was activated when the vehicle speed was within the range of 29.5–32.7 km/h. This value exceeds the recommended operational speed limit for airside vehicles which is ≥ 25 km/h as the control threshold. Therefore, the speed warning system is considered to function properly and in accordance with operational safety principles.

8. Warning System Stability

The test results show that buzzer activation does not interfere with the GPS data recording process and does not cause packet loss. This condition meets the standard requiring that no packet loss occurs when the warning system is active. Therefore, the stability of the warning system can be considered achieved.

9. Geofencing Operational Area

Geofencing testing shows that the system is able to detect operational area violations when the vehicle is more than 3 meters outside the predetermined area boundary. This value is still within the detection deviation tolerance of ≤ 5 meters. Therefore, the operational area geofencing aspect is considered to meet the standard and is effective in detecting area violations.

10. Geofencing Response

The system response to geofencing violations is indicated by buzzer activation when the vehicle leaves the designated area. The response time is within the required limit of ≤ 3 seconds. Therefore, the geofencing response of this system can be categorized as fast and in accordance with the standard.

11. Data Visualization

The results of data visualization testing show that the Map View and Table View displays run synchronously with a synchronization level of 100% and a data update time of ≤ 5 seconds. This condition meets the required usability aspect, therefore the data visualization is considered effective and informative for users.

12. Monitoring Continuity

In terms of monitoring continuity, the system is able to record data continuously with a number of 71–102 points per minute, exceeding the minimum standard of ≥ 60 points per minute. This indicates that the system is capable of performing continuous monitoring without interruption, so the monitoring continuity aspect is considered achieved.

5. Conclusion

Based on the problem formulation and test results, the Internet of Things (IoT)-based apron vehicle tracking system using GPS and LoRa at UPBU Class III Maimun Saleh has been successfully designed and implemented. The system is able to integrate the GPS module to obtain vehicle positions in real-time and transmit data through LoRa stably up to a distance of 5 km with an RSSI value of –101 dBm, position accuracy of 0.712 meters, and an average detection of 12 satellites, making it suitable for implementation in small airport airside environments.

In addition, the system effectively improves monitoring and safety through a speed warning feature (≥25 km/h) using a buzzer and the implementation of geofencing which is able to detect area violations with a deviation of ≤3 meters and a response time of ≤3 seconds. Data visualization through Map View and Table View displays shows 100% synchronization with data updates of ≤10 seconds, while monitoring continuity is able to record 71–102 coordinate points per minute, exceeding the minimum standard of 60 points per minute.

References

Akbar, M. C., Iqbal, M. Y., & Agiska, T. (2024). Manajemen prinsip Bandar Udara. *MODELING: Jurnal Program Studi PGMI*, 11(2).

Ashton, K. (2010). RFID-powered handhelds guide visitors at Shanghai Expo. *RFID Journal.* [Online]. Available: **http://www.rfidjournal.com/article/print/4986**

Bhoka, R. E. M., & Sutarwati, S. (2024). Pengawasan dan ketersediaan peralatan kerja unit apron movement control di area airside Bandar Udara H. Hasan Aroeboesman Ende. *Jurnal Ilmiah dan Karya Mahasiswa*, 2(2), 127–138. [Online]. Available: **https://doi.org/10.54066/jikma.v2i2.1640**

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). Thousand Oaks, CA: SAGE Publications.

Davis, G. B. (2000). Information systems conceptual foundations: Looking backward and forward. In R. Baskerville, J. Stage, & J. I. DeGross (Eds.), *Organizational and social perspectives on information technology* (pp. 61–82). Boston, MA: Springer.

Direktorat Jenderal Perhubungan Udara. (1999). *Keputusan Direktur Jenderal Perhubungan Udara Nomor SKEP/140/VI/1999 tentang persyaratan dan prosedur pengoperasian kendaraan di sisi udara.*

Fajaryasa, I. M. (2022). *Rancangan pengirim data lokasi kendaraan di apron berbasis mikrokontroler* (Tugas Akhir).

Hubbard, B. J., & Hubbard, S. M. (2022). Tracking and monitoring technologies to support airport construction safety. *IOP Conference Series: Materials Science and Engineering*, 1218(1), 012015. [Online]. Available: **https://doi.org/10.1088/1757-899x/1218/1/012015**

International Civil Aviation Organization. (2013). *Annex 10 – Aeronautical Telecommunications.*

International Civil Aviation Organization. (2017). *Safeguarding international civil aviation against acts of unlawful interference* (10th ed.).

International Civil Aviation Organization. (2021). *Doc 9157 – Aerodrome Design Manual, Part 4: Visual Aids* (5th ed.).

Moumen, I., et al. (2023). Real-time GPS tracking system for IoT-enabled connected vehicles. *E3S Web of Conferences*. [Online]. Available: **https://doi.org/10.1051/e3sconf/202341201095**

Murillo, F. J., et al. (2020). Experimental evaluation of LoRa in transit vehicle tracking service based on intelligent transportation systems and IoT. *Electronics*, 9(11), 1950. [Online]. Available: **https://doi.org/10.3390/electronics9111950**

Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 41 Tahun 2023 tentang Pelayanan Jasa Kebandarudaraan di Bandar Udara. (2023). [Online]. Available: **https://jdih.dephub.go.id**

Ramli, N., Zabidi, M. M., Ahmad, A., & Musliman, I. A. (2019). An open source LoRa-based vehicle tracking system. *Indonesian Journal of Electrical Engineering and Informatics*, 7(2), 221–228. [Online]. Available: **https://doi.org/10.11591/ijeei.v7i2.1174**

Republik Indonesia. (2009). *Undang-Undang Nomor 1 Tahun 2009 tentang Penerbangan*. Jakarta: Sekretariat Negara.

Riska, R., & Maulana, A. D. (2023). Analisis risiko bahaya penerbangan di sisi airside sebagai upaya meminimalisir risiko kecelakaan dengan metode HIRA di Bandar Udara Sultan Muhammad Salahuddin Bima. *Ocean Engineering: Jurnal Ilmu Teknik dan Teknologi Maritim*, 2(4), 37–52. [Online]. Available: **https://doi.org/10.58192/ocean.v2i4.1482**

Rizky, A., et al. (2025). Implementasi sistem LoRa untuk deteksi dan pelacakan pendaki gunung. *Jurnal Media Akademik (JMA)*, 3(7). **https://doi.org/10.62281**

Saputra, A. N. M. R. (2024). Pengaruh prosedur pelaksanaan sweeping bagasi terhadap kepuasan penumpang maskapai Wings Air di Bandar Udara Rahadi Oesman Ketapang. *Skripsi*.

Wickens, C. D., & Hollands, J. G. (2000). *Engineering psychology and human performance* (3rd ed.). Prentice Hall.

Wisnuardana, C. D. (2025). Prototype sistem tracking ground movement airside berbasis GPS dan LoRa dengan integrasi web dashboard. *Tugas Akhir*.