Prototype Of an IoT-Based Generator Fuel Level Monitoring and Control System at The Electricity Unit of PERUM LPPNPI JATSC Branch

Niken Ayu Dwi Andini*[1], Ade Irfansyah[2], Willy Artha Wirawan[3]

[1&2] Aviation Polytechnic of Surabaya, Air Navigation Engineering
[3] Aviation Polytechnic of Surabaya, Aircraft Engineering
**Corresponding author. Email: nikenandini@gmail.com*

ABSTRACT

Generators are a critical backup power source for maintaining operational continuity in various sectors. However, manually monitoring the fuel level of a generator results in a lack of efficiency and can lead to delays in refueling, even leading to sudden fuel outages. This situation can affect the availability of backup power when needed. Therefore, this study aims to design and implement an Internet of Things (IoT)-based fuel level monitoring system that can monitor directly, provide automatic notifications when the fuel level is low, and control the fuel filling pump automatically. The development method chosen is the Waterfall model, which includes the stages of analysis, design, implementation, and system testing. The hardware used includes a NodeMCU ESP32 as a microcontroller, a HY-SRF05 ultrasonic sensor to detect fuel level, a relay for DC pump control, and a buzzer as an alarm signal. All data is sent directly to the Blynk application as a monitoring medium. Test results show that the system can read fuel level with an error rate between 0.01% and 3%, while the conversion into liters shows a maximum deviation of 2.6%. This system also successfully controls pumps and alarms automatically according to existing logic. Thus, this system can improve efficiency and accuracy in generator fuel monitoring.

Keywords: *IoT, ESP32, Fuel Monitoring, Blynk, Ultrasonic Sensor, Generator*

1. INTRODUCTION

Generator sets (gensets) are critical components in ensuring the continuity of power supply, especially in high-dependency environments such as hospitals, airports, data centers, and aviation navigation service facilities. In contexts like the Jakarta Air Traffic Services Center (JATSC), the reliability of electrical systems is paramount, as it directly affects flight navigation services. The operational reliability of gensets depends heavily on the availability of fuel in the tank. Consequently, a real-time monitoring system that can accurately measure and report fuel levels becomes essential to prevent unexpected operational disruptions According to Aribowo et al. [1].

However, based on field observations and literature reviews, fuel level monitoring in many gensets is still performed manually. This process requires technicians to open the tank cap and measure the fuel level using rudimentary tools, often involving climbing ladders—posing safety risks and inefficiencies. Manual methods do not support remote monitoring and fail to provide early warnings when fuel levels are low. As a result, decisions regarding fuel refilling are often delayed, increasing the risk of sudden power loss According to Aribowo et al. [2].

The urgency to develop automated monitoring systems is growing due to the inefficiencies and inaccuracies inherent in manual methods. Previous studies have explored IoT-based water level monitoring in storage tanks using ultrasonic sensors, SMS notifications, web interfaces, and Telegram integration as demonstrated in prior research [3], [4]. However, these studies primarily focus on water rather than fuel, which involves a higher safety risk and stricter operational requirements. Moreover, most of the existing systems lack automated control mechanisms for refueling pumps, limiting their practical utility in critical applications.

This study aims to design and implement an Internet of Things (IoT)-based system for real-time fuel level monitoring and automatic control of the refueling pump in gensets. The system is built using NodeMCU ESP32 as the microcontroller, an ultrasonic sensor for level detection, a relay module for pump control, a buzzer for alerting, and the Blynk IoT platform for remote monitoring. The system development follows the Waterfall model, encompassing analysis, design, implementation, and testing phases. This research is expected to enhance operational efficiency, provide timely alerts, and ensure safer fuel management in critical infrastructure like JATSC [5].

The main contribution of this article lies in integrating real-time monitoring with automatic control in a fuel-based IoT application—an approach rarely implemented in existing research. In addition to offering a practical and technical solution, the proposed system is empirically tested for sensor accuracy and pump control reliability. Thus, this research not only expands the application of IoT beyond passive data monitoring but also establishes a foundation for intelligent fuel management systems in backup power applications, contributing to the advancement of smart infrastructure technologies.

2. METHODS

This research adopted a quantitative experimental approach to design, develop, and evaluate an Internet of Things (IoT)-based monitoring and control system for fuel levels in genset tanks. The development process followed the Waterfall software development model, which consists of five sequential phases: requirement analysis, system design, implementation, testing, and maintenance. This structured model was selected to ensure each stage of system development was completed systematically and verified before progressing to the next phase, reducing the likelihood of integration errors.

1. Requirement analysis

During the requirement analysis phase, both hardware and software components were identified. The hardware components included a NodeMCU ESP32 microcontroller, an ultrasonic sensor HC-SR04 for distance measurement, a relay module to control the fuel pump, and a buzzer to provide local alerts. For real-time monitoring and remote control, the Blynk IoT platform was selected as the user interface. In the design phase, system architecture was drafted, including sensor-to-microcontroller communication, data transmission over Wi-Fi, and actuator response mechanisms.

2. System design

The system design phase focused on translating the functional requirements into technical architecture. This included the design of the sensor integration (ultrasonic sensor HC-SR04), microcontroller selection (NodeMCU ESP32), data communication flow via Wi-Fi, and user interface layout using the Blynk IoT platform. Relay logic was also designed for pump automation, supported by a buzzer for local alert notifications.

3. Implementation

During the implementation stage, hardware components were assembled and the microcontroller was programmed using the Arduino IDE. The sensor was calibrated to convert distance measurements into volume estimations based on the dimensions of the fuel tank. All input-output logic, including threshold settings, relay activation, and real-time data transmission to Blynk, was coded and deployed onto the ESP32 board.

4. Testing

In the testing phase, the system was evaluated for accuracy and functional integrity. Five test points were established to compare ultrasonic sensor readings with manual measurements. Additionally, system responsiveness to fuel-level thresholds, buzzer alerts, and automatic pump activation was assessed. The performance was evaluated based on error percentage, system latency, and reliability.

5. Maintenance

Although the maintenance phase was not a focus in this prototype development, future iterations may involve enhancements such as OTA (Over-the-Air) updates, integration with solar backup, or expansion to multi-tank systems. The Waterfall model provided a structured path from conceptualization to validation, allowing for controlled experimentation and ensuring that the resulting prototype met both technical and operational objectives.

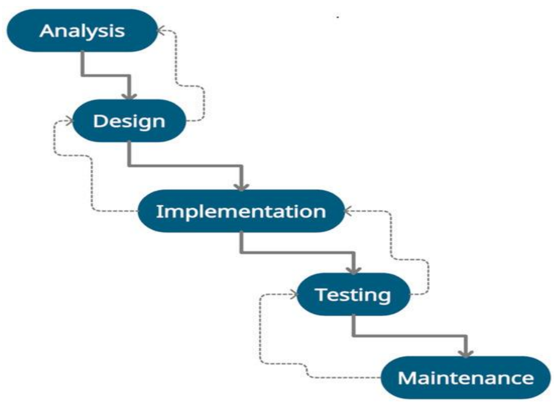

Figure 1 Research Flowchart

3. RESULTS AND DISCUSSION

This section presents the results of the system development process following the Waterfall model, covering the stages of requirement analysis, system design, implementation, testing, and evaluation. The discussion includes technical elaboration of each stage, sensor performance analysis, system response evaluation, and quantitative error calculation.

1. Requirement Analysis

Initial observations at the Power Unit of Perum LPPNPI Cabang JATSC indicated that the existing method for monitoring fuel levels in genset tanks relied on manual inspection. This involved opening the tank and visually measuring fuel depth, which was time-consuming, potentially hazardous, and prone to error. Based on this field analysis, the functional requirements defined were:

- Real-time fuel level monitoring
- Early warning system when the level drops below a threshold
- Remote access via a mobile application
- Automatic control of the fuel pump
- Local alert system (buzzer)

These requirements were established to improve safety, efficiency, and operational continuity, particularly in critical infrastructure reliant on backup power systems.

2. Design

The system design stage is a crucial phase that bridges user needs with technical implementation through the development of hardware and software architecture. This system design is intended to enable real-time fuel level data acquisition, local data processing on a microcontroller, data transmission via a Wi-Fi network, and automatic responses through actuators (relays and buzzers). Functionally, the system is divided into four main subsystems: (1) data acquisition, (2) data processing, (3) data transmission and remote control, and (4) actuators for pump control and local alarms. Data acquisition is performed by the HC-SR04 ultrasonic sensor, which operates based on the principle of echo time. This sensor emits an ultrasonic signal and measures the time it takes for the signal to be reflected back by the fuel surface, thereby obtaining the distance value.

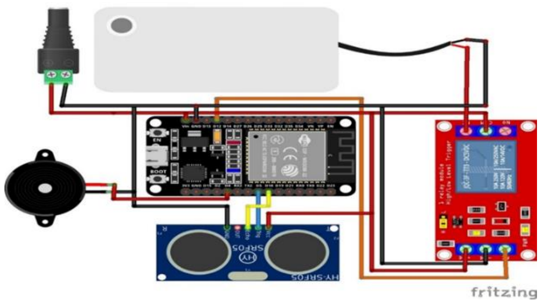

Figure 2 Integration of Components in Tool

The ESP32 microcontroller was chosen because it has integrated Wi-Fi, large memory capacity, and sufficient processor speed for data processing and real-time control. For data transmission, the ESP32 uses Wi-Fi connectivity to the internet and periodically sends data to the Blynk IoT platform, where a virtual dashboard has been designed using widgets such as “Gauge” (to display fuel level), ‘Notification’ (for alerts), and “Button” (for manual pump control). On the actuator side, a single-channel relay module is directly connected to the fuel pump control line. When the fuel level drops below the threshold (e.g., <10 cm), the system activates the relay to turn on the fuel pump.

As an additional local reminder, a buzzer is activated simultaneously as an audible warning. The digital signals for the relay and buzzer are controlled via logic commands in the Arduino IDE code using if-else conditional logic. With this approach, the system can operate in automatic mode while also providing flexibility for manual remote control by the operator, making this design ideal for operational environments such as generator rooms that are not always directly supervised.

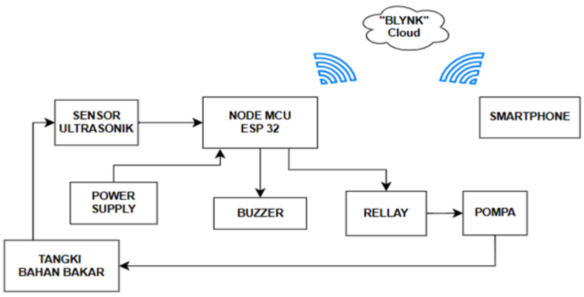

Figure 3 Equipment Design and Construction

3. Implementation

All hardware components are assembled on a breadboard, including the Node MCU ESP32, HC-SR04 ultrasonic sensor, relay module, and buzzer. The circuit is designed with consideration for the connection topology between input components (sensors), processing units (microcontrollers), and output components (actuators and monitoring platforms). To prevent voltage fluctuations and interference, pull-up resistors and integrated ground connections are used. The system operating voltage is set to 5V for the sensor and buzzer, and 3.3V for the ESP32's input/output logic.

The ESP32 microcontroller is programmed using the Arduino IDE version 2.3.6, utilizing libraries such as NewPing.h for the ultrasonic sensor and BlynkSimpleEsp32.h for communication with the Blynk cloud platform. The programming steps include: (1) configuring the digital input/output pins, (2) reading the distance value from the HC-SR04 sensor, (3) converting the data into fuel level and volume estimation, (4) sending the data to the Blynk app via Wi-Fi, and (5) control logic to activate the buzzer and relay when the fuel level reaches the threshold.

The system was assembled on a prototyping board. The ultrasonic sensor was positioned vertically to ensure accurate downward measurement. The system was programmed via Arduino IDE 2.3.6, and the Blynk dashboard was customized with data widgets for:

- Fuel level (in cm)
- Alert status (buzzer ON/OFF)
- Pump control button (manual override)

A mathematical formula was implemented in the ESP32 firmware to convert distance to fuel volume.

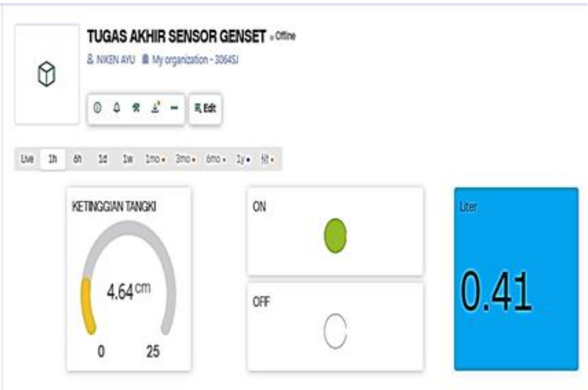

Figure 4 Display in Blynk

For initial testing, the system was placed in a simulated tank in the form of a transparent tube or box so that the process of fuel decrease and increase could be observed visually. Measurements were taken by adding liquid to the tank and recording the sensor readings and system responses.

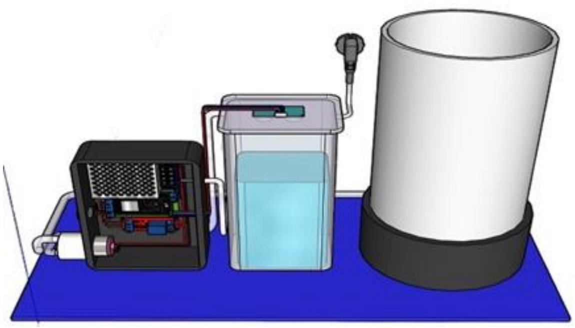

Figure 5 tank simulation media

4. Testing

The testing phase was conducted to evaluate the overall performance of the system based on two main aspects, namely the accuracy of the ultrasonic sensor measurements and the system's logical response to changes in fuel level, both automatically and manually. The testing was conducted in a controlled environment using a calibrated simulation tank and a liquid substitute for fuel. Testing was performed at five fuel level reference points: 5 cm, 10 cm, 15 cm, 20 cm, and 25 cm. Each point was tested three times to obtain stable and consistent data in Table 1.

The sensor measurements were compared with manual measurements using a ruler, then analyzed using the percentage error formula as follows:

$$System\ error\ percentage = \frac{difference}{ruler\ reading}\ X\ 100\%$$

In addition to sensor accuracy, testing also focused on the system's response to fuel level limits. The threshold was set at a distance of 20 cm from the sensor, which represents an almost empty condition. When the fluid level is below this point, the system successfully activates the buzzer and pump relay automatically. The buzzer and pump response were verified using a stopwatch, showing activation time of less than 1 second after the sensor detected the critical condition, indicating a rapid response.

Manual control functionality testing via the Blynk app was also conducted. Users can activate and deactivate the pump via virtual buttons on the dashboard. Communication runs smoothly with an average latency of <1 second, provided the Wi-Fi connection is stable. The system also successfully sends real-time notifications to the user's phone in the form of warnings when fuel is low, using the Blynk. Loge vent () features active in the firmware.

Overall, the testing phase demonstrated that the system has acceptable sensor accuracy, high response speed, and effective automatic and manual control capabilities. This indicates that the system meets the technical specifications required for an IoT-based generator fuel monitoring application.

Table 1 Ultrasonic Sensor Testing

No.	Meter Reading Distance (cm)	Sensor Reading Distance (cm)			Percentage Error (%)			Eror Level
		Data 1	Data 2	Data 3	Data 1	Data 2	Data 3	
1.	5 cm	4,90	4,95	4,80	0,02%	0,01%	0,04%	0,02%
2.	10 cm	9,97	10,1	10	0,3 %	1%	0%	0,43%
3.	15 cm	14,40	14,25	14,70	0,04%	0,05%	0,02%	0,04%
4.	20 cm	19,34	19,39	19,72	1,30%	3,05%	1,4%	1,91%
5.	23 cm	22.7	22,72	22,38	1,30%	1,22%	2,70%	1,74%

Ultrasonic sensor testing was conducted at five reference distances (5 cm, 10 cm, 15 cm, 20 cm, and 23 cm) with three measurements at each point to test consistency and accuracy. At close distances (5 cm and 10 cm), the readings were highly accurate, with average errors of approximately 0.02% and 0.43%, respectively. At medium distances (15 cm), the error remained low at around 0.04%. At longer distances, such as 20 cm and 23 cm, the average error slightly increased to 1.91% and 1.74%, respectively, but remained within acceptable limits. Overall, the average error across all distances was small, indicating that the ultrasonic sensor has high accuracy, fast response, and is suitable for automatic fuel level monitoring in Table 1.

5. Evaluation and Discussion

The system operated reliably throughout all test scenarios. When the fuel level fell below a threshold (set at 10 cm), the buzzer activated, and the relay module powered the refuelling pump. Simultaneously, the Blynk app displayed updated sensor readings and triggered an alert notification. Manual pump control via the app was also functional, providing a user-friendly interface for remote operation.

Compared to previous studies—such as [6], who used Blynk for water tank automation, and according to [7], who employed web-based visualization—this system adds value by addressing a more critical application involving fuel, which carries safety implications. Additionally, the system integrates sensor readings, remote notifications, and actuator control, which are often implemented separately in past works.

The identified contribution of this research lies in the creation of an integrated, closed-loop IoT-based system that not only informs the user but also responds autonomously. This improves both operational efficiency and safety, particularly in backup power scenarios. The ability to replicate this design in similar environments highlights its practical applicability.

4. CONCLUSION

Based on the above research, the following conclusions can be drawn:

1. Based on the design and implementation results, the fuel level monitoring system in the generator tank using Internet of Things (IoT) technology has been successfully designed using NodeMCU ESP32, HY-SRF05 ultrasonic sensor, buzzer, DC pump, and Blynk platform. This system is capable of directly monitoring fuel levels, issuing warning notifications when fuel levels reach the minimum point, and automatically controlling the pump for refueling.

2. The validity and accuracy of fuel level measurements using the IoT-based system have been proven through a series of tests comparing sensor readings with manual methods. The HY-SRF05 ultrasonic sensor is capable of stably detecting the fluid surface inside the tank, and the data is successfully converted by the ESP32 into an estimate of fuel volume through geometric formula-based calculations. Test results show that the deviation of readings from actual volume is within technical tolerance limits, and demonstrate consistency across experiments, indicating that this system is reliable for real-time monitoring applications.

REFERENCES

[1] D. Aribowo, desmira, and D. A. F., "Maintenance System of Genset Engine at PT (Persero) Pelabuhan Indonesia II," Journal of Economic Perspectives, vol. 2, no. 1, pp. 1–4, 2022.

[2] K. Rahmawati, S. Hariyadi, W. Suryono, J. Teknik, L. Bandara, T. Penerbangan, P. Surabaya, and J. Jemur Andayani, "Design and Development of Recording," 2019.

[3] K. Swasti, "Volume Monitoring System Using Ultrasonic Sensor via Internet and SMS Access," 2008.

[4] I. Gunawan and T. Akbar, "Prototype Implementation of Internet of Things (IoT) on Water Tank Level Monitoring Using NodeMCU ESP8266 and Blynk," Infotek: Journal of Informatics and Technology, vol. 3, no. 1, pp. 1–7, 2020.

[5] A. Irfansyah, Y. T. K. Yunior, and B. B. Harianto, "Internet of Things (IoT) Based Trash Can Monitoring System Using Web Services with the CodeIgniter Framework in the Surabaya Aviation Polytechnic Environment," in Proc. ICATEAS, 2023, pp. 1050–1053.

[6] G. P. H. Romadhoni, "Design and Development of Genset Monitoring System via Telegram Application Notification," 2023.

[7] I. Gunawan and T. Akbar, "Prototype Implementation of Internet of Things (IoT) on Water Tank Level Monitoring Using NodeMCU ESP8266 and Blynk," Infotek: Journal of Informatics and Technology, vol. 3, no. 1, pp. 1–7, 2020.