DESIGN AND DEVELOPMENT OF AN AC MOTOR SPEED CONTROL AND MONITORING SYSTEM BASED ON WHATSAPP API AND SUGENO FUZZY LOGIC

Satria Ego Vania[1], Ayusta Lukita Wardani[2*], Widi Aribowo[3], Mahendra Widyartono[4]
[1,2*,3,4] State University of Surabaya,, Surabaya, Indonesia
Corresponding author: ayustawardani@unesa.ac.id

Keywords:
WhatsApp API
Fuzzy logic
Temperature
Occupants
Motor control
IoT

ABSTRACT

As the demand for efficient and accessible automation systems increases, integrating widely used communication platforms such as WhatsApp offers a promising solution for control and monitoring applications. This study proposes the design and implementation of a fan speed control system for an AC motor using the WhatsApp API integrated with Sugeno fuzzy logic. The system automatically adjusts fan speed based on two key parameters: ambient temperature and the estimated number of occupants in the room, enabling adaptive and intelligent control. An experimental approach was employed to evaluate system performance across various operational scenarios, focusing on command execution accuracy, the effectiveness of fuzzy logic rule implementation, and the reliability of anomaly detection through alert generation. Test results demonstrate that the system can execute user commands with high precision and deliver responsive automated actions. Furthermore, the integration of fuzzy logic ensures appropriate fan speed adjustments in dynamic environments. This approach not only improves energy efficiency but also enhances user convenience by utilizing WhatsApp a platform that users are already familiar with as the primary communication interface.

INTRODUCTION

The development of smart control systems in the era of digital transformation has become increasingly relevant, especially in improving energy efficiency and remote accessibility. Traditional electric fan control systems often lack adaptability to changing environmental conditions, making automation a crucial feature for modern systems. Recent studies have shown the effectiveness of fuzzy logic in nonlinear control systems such as fan speed regulators, where temperature and humidity are key parameters (Taukid et al., 2023; Aldama et al., 2023). At the same time, lightweight communication protocols like MQTT are widely adopted in IoT applications for efficient data exchange between devices (Athallah Aditya & Setia Budi, 2023). Despite these advances, the use of widely accessible messaging platforms such as WhatsApp for smart control interfaces remains limited. With over 2 billion active users (Social et al., 2025), WhatsApp presents a promising medium for real-time remote control. Previous works like (Alamsyah et al., 2025) have demonstrated the basic use of WhatsApp API in device control, but lacked integration with fuzzy logic or adaptive environmental sensing. Therefore, this research aims to design and implement a smart fan speed control and monitoring system based on WhatsApp API and Sugeno fuzzy logic, offering a more responsive and accessible solution for remote electrical system management.

RESEARCH METHOD

This section describes the research methodology employed in this study, which includes the development of the research flowchart, device block diagram, system workflow, and the configuration of the wiring system to integrate all elements into a functional prototype.

1.1. Research Steps

This study involves several stages in designing the research, starting from literature review to results and conclusions, as illustrated in the flowchart shown in Figure 1.

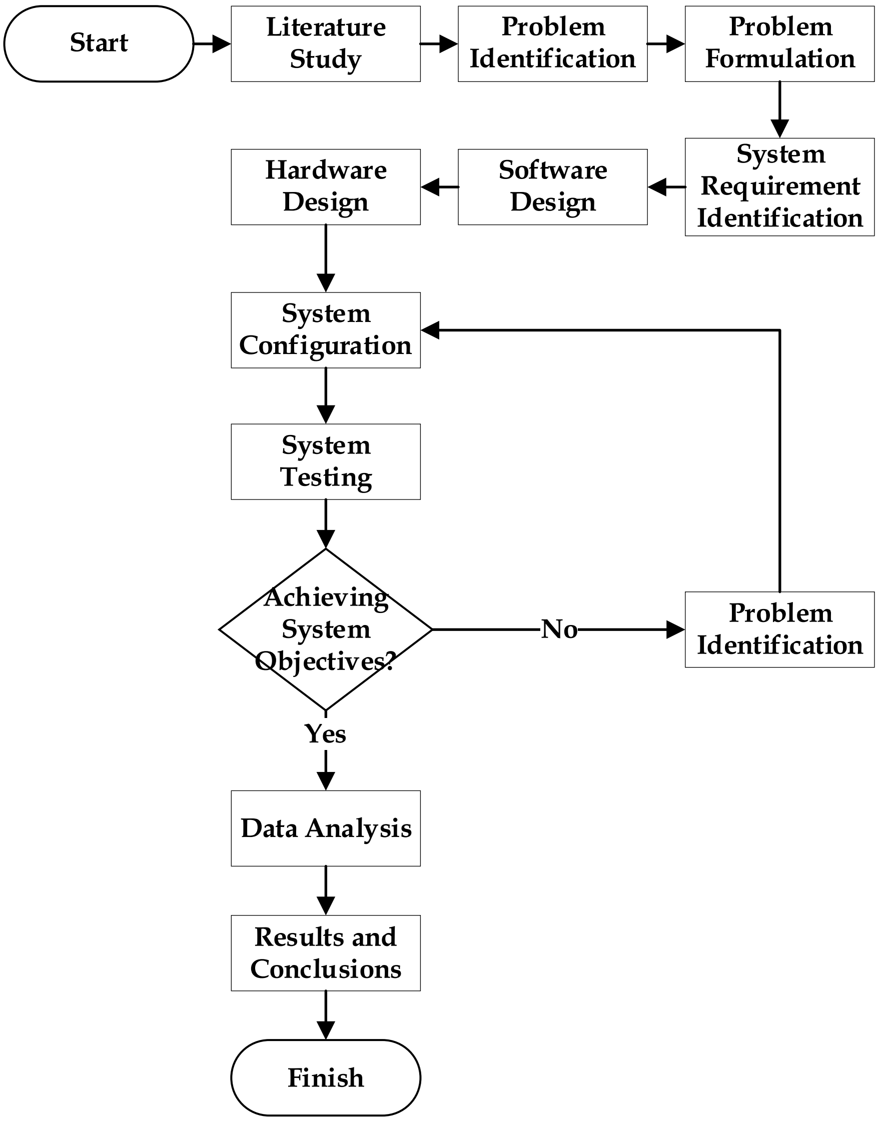

Figure 1. Research Flow Chart
Source : Personal Documentation

The following is an explanation of the research workflow diagram:

1. Literature Study: The process begins with reviewing relevant literature to gather foundational knowledge and understand existing research related to the topic.
2. Problem Identification: Based on the literature, specific problems are identified that require further investigation or a proposed solution.
3. Problem Formulation: The identified problems are then formulated into clear research questions or objectives.
4. System Requirement Identification: From the formulated problem, detailed system requirements both hardware and software are defined to guide the design process.
5. Hardware and Software Design: The system is then designed according to the requirements. This includes developing the hardware architecture and writing the necessary software code.
6. System Configuration: After the design stage, the components are integrated and configured into a working system.
7. System Testing: The configured system is tested to evaluate whether it functions as expected and meets the objectives.
8. Decision Point, Achieving System Objectives?: If the system does not achieve the intended objectives, the process loops back to the problem identification stage for re-evaluation and refinement. If the objectives are met, the process continues.
9. Data Analysis: Collected data from the testing phase is analyzed to assess performance, efficiency, or other relevant metrics.
10. Results and Conclusions: Finally, the research concludes by summarizing the findings and presenting conclusions based on the data analysis.

1.2 Creating A Device Block Diagram

At this stage, the flow rate of the components used is determined. Figure 2 below shows the tool path diagram designed in the system planning stage.

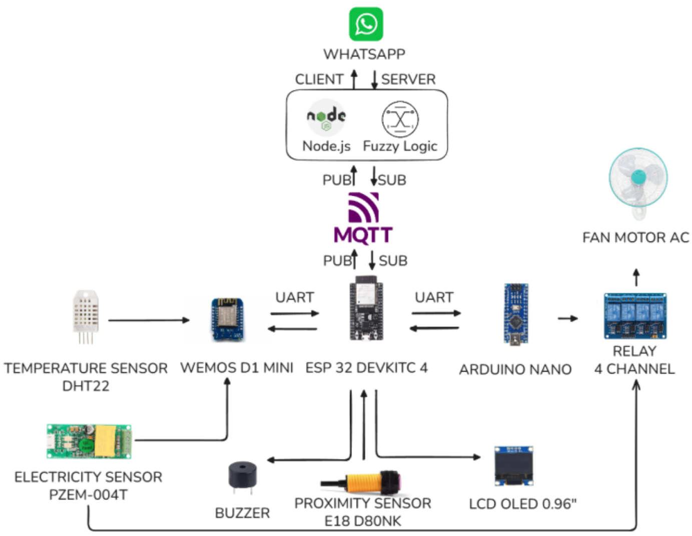

Figure 2 System Workflow Block Design
Source : Personal Documentation

1.3. System Workflow

The system begins by reading data from all connected sensors, with particular focus on temperature and occupancy detection. These primary inputs are then processed on the server to determine the appropriate motor speed control using a fuzzy logic algorithm. This automatic control mechanism can be interrupted by a manual command sent via WhatsApp, which is interpreted on the server side as a manual override and subsequently forwarded to the hardware system for execution.

In addition to the control system, the platform also integrates a voice call-based alert mechanism. This feature is triggered by the electricity sensor, which monitors the load conditions of the system. If the load is detected as active but the electrical readings fall outside of expected parameters, the system initiates an automated voice call to notify the client. This provides an added layer of safety and ensures timely response in the event of abnormal power conditions.

1.4 Microcontroller Design Diagram

The microcontroller design diagram provides a detailed overview of the interconnections between all hardware components and the microcontrollers. It illustrates the input and output (I/O) pin configurations, communication interfaces (such as UART), and the roles of each device in the system. Each sensor and actuator is connected to a specific microcontroller based on its function and data requirements.

For instance, the DHT22 temperature sensor(Siswoyo, 2025) and the PZEM-004T electricity sensor(Afandi et al., 2025) are interfaced with the Wemos D1 Mini, which acts as a data acquisition node. The collected data is then transmitted via UART to the ESP32 DevKitC V4, which serves as the central processing unit of the system. The ESP32 also receives input from a proximity sensor, drives an OLED display, and controls a buzzer. It communicates further with the Arduino Nano, which handles actuator control through a 4 channel relay module responsible for regulating the speed of a three-level fan.

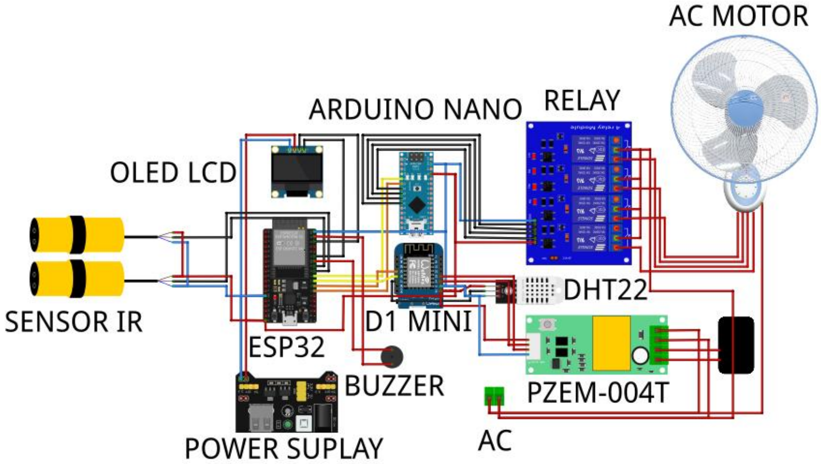

Figure 3. Microcontroller Circuit Schematic
Source : Personal Documentation

RESULTS AND DISCUSSION

This research aims to assess the feasibility and reliability of a smart control system capable of determining appropriate fan speed levels using environmental inputs, while also enabling manual control via WhatsApp. Experimental results demonstrate that the system successfully interprets ambient temperature and estimated occupancy data through Sugeno fuzzy logic to produce accurate and adaptive fan speed outputs.

1.5 Hardware Results

Figure 4 shows the hardware implementation of the system in operation. The setup demonstrates its ability to control the AC fan motor load while simultaneously reading ambient temperature and detecting the presence or movement of individuals passing through the monitored area. This physical configuration validates the integration of sensors and actuators, highlighting the system's functionality in a real world environment.

Figure 4. Hardware System in Operation
Source : Personal Documentation

1.6. Software Results

Figure 5 illustrates the software interface developed as part of the system implementation. This figure showcases the primary features integrated into the application, including the user interface and the emergency notification system. One of the key innovations presented in this work is the integration of WhatsApp voice calls as a method for emergency alerts. This allows the system to automatically initiate a voice call to a registered contact when specific emergency conditions are detected, enhancing the responsiveness and practicality of the control system. The interface is designed to be intuitive, allowing users to monitor system conditions in real time and ensuring that critical alerts are delivered promptly and reliably.

Figure 5. Software User Interface
Source : Personal Documentation

1.7. Testing Fan Speed Level Control via WhatsApp Commands

This testing phase was conducted to evaluate the system's ability to adjust fan speed levels in response to manual control commands transmitted via the WhatsApp API interface. The primary objective of this test is to assess the system's responsiveness and accuracy in processing control instructions received through WhatsApp. By simulating manual input from the user through the messaging platform, the system's ability to interpret, process, and execute the command to change the fan speed can be effectively measured. The results from this test provide insights into the reliability and performance of the integrated communication-control mechanism, particularly in terms of latency, command recognition, and actuation time.

Table 1. Testing Fan Speed Level Control via WhatsApp Commands

No	Fan Speed Command via WhatsApp (Stop/Low/Medium/High)	Actual Fan Speed Result (Stop/Low/Medium/High)	Response Time (ms)
1	Low	Low	640
2	Medium	Medium	570
3	High	High	650
4	Stop	Stop	590
5	Low	Low	610
6	Medium	Medium	890
7	High	High	840
8	Stop	Stop	530
9	Low	Low	10000
10	Medium	Medium	870
11	High	High	630
12	Stop	Stop	520

No	Fan Speed Command via WhatsApp (Stop/Low/Medium/High)	Actual Fan Speed Result (Stop/Low/Medium/High)	Response Time (ms)
13	Low	Low	550
14	Medium	Medium	840
15	High	High	680
16	Stop	Stop	520
17	Low	Low	780
18	Medium	Medium	710
19	High	High	770
20	Stop	Stop	570
Average Response Time (Excluding Data No. 9)			672

Source : Personal Documentation

1.8. Testing Fuzzy Inference

This test aims to validate the accuracy and consistency of the system's fuzzy logic inference results by comparing them to the reference outputs generated by the MATLAB Fuzzy Toolbox. The evaluation focuses on ensuring that the implemented fuzzy inference engine produces outputs that align with standard MATLAB-based fuzzy logic processing. To achieve this, each possible combination of temperature membership levels and the number of detected people is tested. By systematically analyzing the output from these combinations, the test assesses whether the system's fuzzy reasoning matches the expected behavior defined in the reference model. This verification is essential for confirming the reliability of the fuzzy logic controller in real-world operation.

Table 2. Testing Fuzzy Inference

No	Temperature (°C)	Number of People	Fuzzy Inference (MATLAB)	Fuzzy Inference (System)	Error (%)
1	15	0	0	0.01	–
2	18	1	1	1.01	1.00%
3	19	2	1	1.01	1.00%
4	20	3	1	1.01	1.00%
5	21	4	1	1.01	1.00%
6	22	5	2	2.01	0.50%
7	23	6	2.2	2.19	0.45%
8	24	7	2.67	2.66	0.37%
9	25	1	1	1.01	1.00%
10	26	2	1	1.01	1.00%
11	27	3	2	2.01	0.50%
12	28	4	2.25	2.26	0.44%
13	29	5	3	3.01	0.33%
14	30	6	3	3.01	0.33%
15	31	7	3	3.01	0.33%
16	32	1	2	2.01	0.50%

No	Temperature (°C)	Number of People	Fuzzy Inference (MATLAB)	Fuzzy Inference (System)	Error (%)
17	33	2	2	2.01	0.50%
18	34	3	3	3.01	0.33%
19	35	4	3	3.01	0.33%
20	40	10	3	3.01	0.33%
Average Error (%)					0.59%

Source : Personal Documentation

1.9. System Alerting Testing

Specifically, the alert mechanism is tested in a fault scenario where the relay is active, yet the measured power consumption is zero. This condition is used to simulate a failure in the fan system such as a motor malfunction or power disconnection where the system falsely detects the fan as running, while in reality, the load is not operating. To simulate this scenario, the AC power supply is deliberately disconnected while the relay remains in an active, load-bearing state. This creates a mismatch between the expected and actual power status. Upon detecting this discrepancy, the system automatically triggers its alert function by sending a voice call notification via WhatsApp, thereby ensuring prompt awareness of potential system faults.

Table 3. System Alerting Testing

No.	Relay Status (1, 2, 3)	Response Time (ms)
1	1	8420
2	2	7420
3	3	7110
4	1	9520
5	2	8580
6	3	8750
7	1	8910
8	2	7480
9	3	7420
10	1	8160
Average Response Time		8177

Source : Personal Documentation

1.10. explanation of the data results

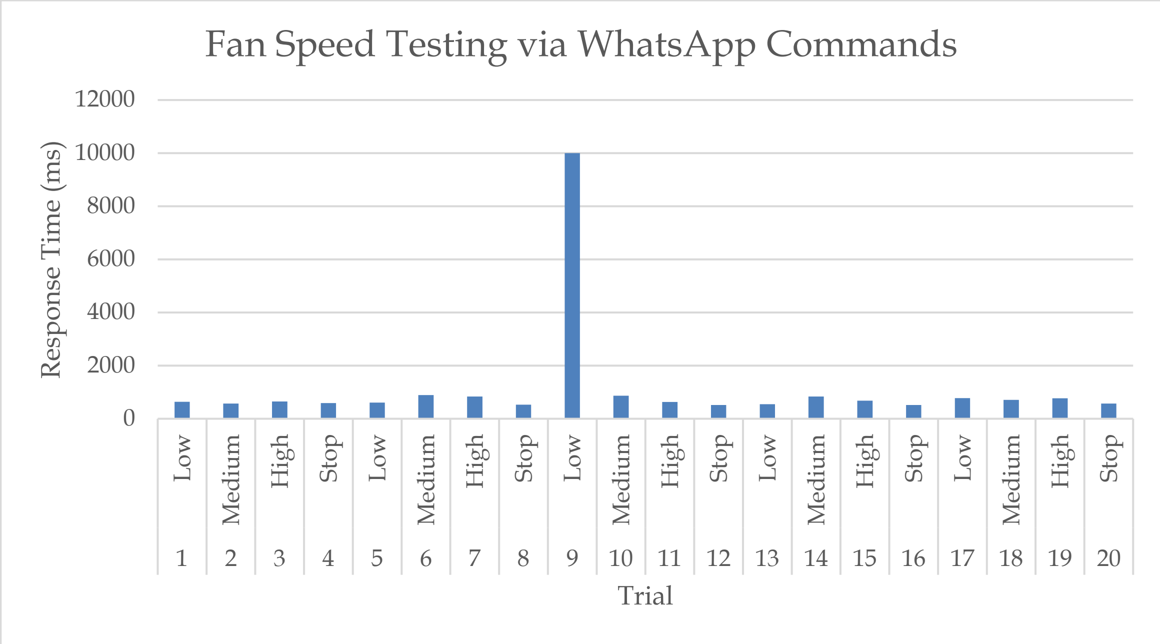

Figure 6. Bar Diagram: Fan Speed Testing via WhatsApp Commands
Source : Personal Documentation

Based on Figure 6 that includes system response time to fan speed change commands via WhatsApp, results are obtained that show variations or fluctuations in the duration of response time between tests. On average, the system is able to provide responses in the range of 520 ms to 890 ms. However, there is one data that shows response times reaching 10,000 ms in the 9th test. This value cannot be categorized as a normal response because it has passed or is equal to the set test interval limit. Thus, this condition can be considered as a command execution failure in the test.

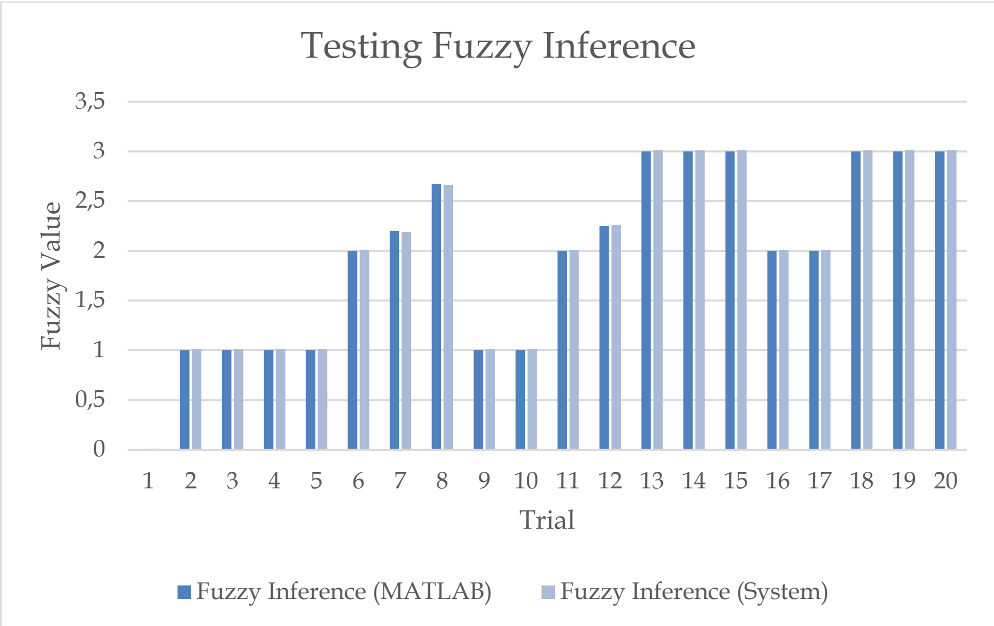

Figure 7. Bar Diagram: Testing Fuzzy Inference
Source : Personal Documentation

Based on Figure 6 the majority of the fuzzy inference outputs generated by the system exhibit a consistent pattern, with a deviation of only 0.01 from the MATLAB reference values typically appearing as 0.01, 1.01, 2.01, or 3.01. This consistency indicates that the system reliably replicates the behavior of the Sugeno fuzzy logic model. However, three data points showed unique characteristics: data point 7 (2.19 vs. 2.20), data point 8 (2.66 vs. 2.67), and data point 12 (2.26 vs. 2.25), each displaying a deviation of 0.01 in the opposite direction. These specific cases occurred because the input values (temperature and number of people) were positioned within the transitional or "shoulder" regions of the fuzzy membership functions zones where membership degrees overlap between two categories. As a result, the inference process produced more granular decimal outputs. In contrast, the majority of other data points involved inputs located at the peak of the trapezoidal membership functions (full membership) or outside the trapezoid (zero membership), which led to more stable and predictable defuzzification results.

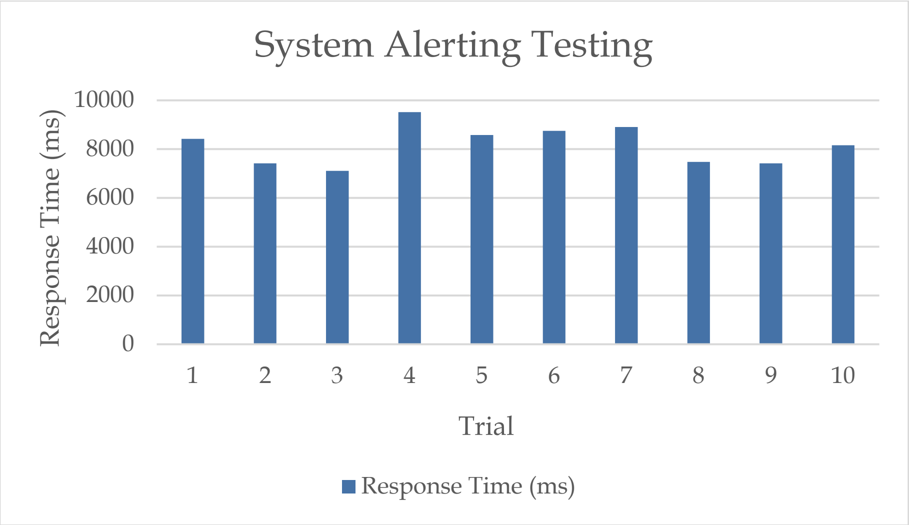

Figure 8. Bar Diagram: System Alerting Testing
Source : Personal Documentation

The alerting feature test demonstrated that the system successfully responded to anomaly conditions by sending voice call notifications via WhatsApp. Based on ten test iterations, the average response time was recorded at 8177 ms, or approximately 8 seconds. Compared to conventional alerting systems such as automatic protection mechanisms or local alarms which can respond in under one second, the WhatsApp voice call approach does exhibit a slower response time. However, this delay is considered acceptable given that the system is not designed for critical or time-sensitive applications. Instead, using WhatsApp voice calls offers a key advantage: the ability to reach users remotely while delivering alerts in a personal and attention-grabbing manner that is less likely to be ignored. Therefore, despite the slightly longer trigger time, this approach remains relevant and effective for comfort-oriented systems, aligning with the primary objectives of the system developed in this research.

CONCLUSION

Based on the results of the design, implementation, and testing of the system that has been carried out, it can be concluded that:

1. The system shows a response time to WhatsApp commands, with normal response times in the range of 520 ms to 890 ms, with an average response time of 671 ms.
2. Sugeno fuzzy logic is successfully implemented with accuracy in testing showing a maximum error of 1.00% and an average error of 0.59% against the MATLAB reference.
3. The alerting feature of the system successfully responded to all anomalous conditions tested, with an average notification delivery time of 8177 ms (8.17 seconds).

REFERENCES

Afandi, R., Satria, B., & Aryza, S. (2025). Analysis of IoT Design for Monitoring Electrical Energy Use in Homes. *Instal: Jurnal Komputer*, *17*(02), 72–85.

Ahmad Qomarudin, Q. (n.d.). SISTEM INFORMASI PRESENSI SISWA BERBASIS RFID TERINTEGRASI WEB DENGAN NOTIFIKASI API WHATSAPP. *SISTEM INFORMASI PRESENSI SISWA BERBASIS RFID TERINTEGRASI WEB DENGAN NOTIFIKASI API WHATSAPP*.

Alamsyah, M. R., Ikorasaki, F., & Sitinjak, N. M. (2025). *Perancangan Alat Penerangan Lampu Jalan Berbasis Internet Of Things Menggunakan Aplikasi Whatsapp Design of Street Lighting Devices Based on Internet of Things Using Whatsapp Application*. http://kti.potensi-utama.ac.id/index.php/JID

Aldama, M. F., Isworo, M. R. R., Ruwa'im Nafie, R., Agnesya, P. D., & Sari, A. P. (2023). Penerapan Fuzzy Logic Menggunakan Metode Sugeno Dan Tsukamoto Untuk Mengontrol Suhu AC. *Prosiding Seminar Nasional Informatika Bela Negara*, *3*, 117–121.

Ardila, Y. (2025). Design of an Inclined Plane Experiment Tool Using E18-D80NK Infrared Proximity and MPU6050 Accelerometer Sensors: An IoT-Based Approach. *Risalah Fisika*, *5*(1), 1–13.

Athallah Aditya, R., & Setia Budi, A. (2023). *Prototipe Sistem Keamanan Parkir berbasis RFID dengan Protokol MQTT* (Vol. 7, Issue 7). http://j-ptiik.ub.ac.id

Chang, Y.-H., Wu, F.-C., & Lin, H.-W. (2025). Design and implementation of esp32-based edge computing for object detection. *Sensors*, *25*(6), 1656.

Hoàng, T. T. (2025). *Smart Sensor Technologies and Communication Interfaces for Environmental Monitoring at Schools*.

Maji, S., & Ganguli, S. (2025). Fuzzy Logic Control for Industrial Applications. *Controller Design for Industrial Applications*, 1–20.

McCarthy, M., Burd, B., & Pollock, I. (2025). Publish Subscribe Using MQTT. In *Concise Guide to the Internet of Things: A Hands-On Introduction to Technologies, Procedures, and Architectures* (pp. 159–189). Springer.

Penica, M., Bhattacharya, M., O'Brien, W., McGrath, S., Hayes, M., & O'Connell, E. (2025). Advancing Interoperable IoT-Based Access Control Systems: A Unified Security Approach in Diverse Environments. *IEEE Access*.

Pradesh, A., & India, J. (2025). IoT Based Transformer Load Monitoring and Controlling With Alerting System Using Arduino Nano and GSM. *Journal of Computer and Electrical Sciences (JCES)-Vol*, *2*(01).

Siswoyo, A. (2025). Optimization of Temperature Sensor Selection for Incubators: Real—Time Accuracy Analysis of DHT22, LM35, and DS18B20 in Controlled Environment Simulations. *Internet Things*, *5*.

Social, W. A., DataReportal, & Meltwater. (2025). *Digital 2025: Global Overview Report*. We Are Social; DataReportal; Meltwater. https://www.statista.com/

Taukid, M. N., Elzandy, I., Adyani, A. P., & Sari, A. P. (2023). Program Pengendali Kipas Angin Berdasarkan Suhu dan Kelembaban Menggunakan Logika Fuzzy. *Prosiding Seminar Nasional Informatika Bela Negara*, *3*, 42–52