Real-Time Fire Risk Assessment in Server Rooms Using Hybrid Threshold and Fuzzy Logic on Raspberry Pi

Renel Cikita Robbyn[1], Anang Budikarso[2]*
[1,2]Politeknik Elektronika Negeri Surabaya (PENS), Jl Raya ITS Kampus PENS, Surabaya, Indonesia
**Corresponding author: Anang Budikarso email: anang_bk@pens.ac.id*

Manuscript received April 09, 2026, revised April 20, 2026, accepted April 21, 2026

Abstract – *The server room is a critical area for the storage and operation of server-related machines. In this environment, it is essential to protect data and systems from unwanted events. Significant increases in temperature or humidity can cause severe damage to equipment, system failures, and even fire risks due to sparks from transmission cables. Temperature and humidity readings with average values of 29.83°C and 45.5%, respectively, and low standard deviations, indicating reliable measurements. with an average delay of approximately 2.5 milliseconds. The system continuously monitors temperature, humidity, and flame presence to assess risk levels accurately. Threshold methods provide immediate alerts when sensor readings exceed predefined limits, while fuzzy logic enables nuanced risk classification based on combined sensor inputs. Experimental results demonstrate the system's effectiveness in early detection of hazardous conditions via visual and auditory alarms, thereby enhancing the protection of critical server infrastructure.*

Keywords: *Raspberry Pi, Node-Red, Server Room, Threshold Method, Fuzzy Logic Control*

I. Introduction

Server rooms are critical infrastructures where environmental stability directly influences operational continuity and data security. Conventional monitoring systems typically employ basic threshold-based methods to detect abnormal temperature or humidity levels [1]. While effective at identifying extreme conditions, these methods fall short in assessing gradual risk levels, often leading to delayed responses or unnecessary alarms. Previous studies have explored the application of fuzzy logic to environmental risk analysis, but most focus on theoretical frameworks rather than integrating real-time alert systems or user-friendly interfaces. Meanwhile, MQTT-based systems running on Raspberry Pi have proven capable of collecting and transmitting sensor data. Yet, they lack built-in mechanisms to dynamically assess and communicate risk to operators in a practical manner.

This research proposes a more comprehensive IoT-based monitoring system that integrates the DHT22 sensor for real- time temperature and humidity tracking, along with a flame sensor for direct fire detection. Unlike prior implementations [7], our system applies fuzzy logic to classify fire risk into four levels: safe, normal, dangerous, and critical—enabling operators to prioritize actions more effectively. The solution also leverages Node-RED for dashboard visualization and WhatsApp for instant alerts—features missing in earlier systems [3]. A Raspberry Pi serves as the central processing unit, offering both cost-effectiveness and scalability. By combining threshold-based triggers (e.g., buzzer activation at ≥23°C) with fuzzy-logic risk evaluation, the system minimizes false alarms while improving response accuracy. This dual-layered approach not only improves detection reliability but also delivers an intuitive interface for real-time decision-making.

II. Material and Method

This Server Environmental Monitoring System research utilizes various electronic components, including the Raspberry Pi 4, a DHT22 temperature and humidity sensor, a flame sensor, LED indicators, buzzer alarms, and other supporting devices, all integrated through the Node-RED platform. The methodology combines threshold-based monitoring with fuzzy logic. The threshold method provides early detection of abnormal environmental conditions by setting predefined limits for temperature and humidity. Meanwhile, fuzzy logic offers a more nuanced risk analysis by evaluating multiple input variables to generate precise decision outputs.

A. Raspberry Pi

The Raspberry Pi is a single-board computer (SBC) the size of a credit card. It is equipped with all the functionalities of a complete computer, utilizing an ARM

System-on-a-Chip (SoC) integrated onto a printed circuit board (PCB). This device uses an SD card for its operating system and long- term storage. In this research, the Raspberry Pi 4 is utilized, which features a Broadcom BCM2711 CPU with a frequency of 1.5 GHz, 2GB of RAM, two USB 3.0 ports, two USB 2.0 ports, one USB Type-C port (for power), two Micro HDMI 2.0 ports, a 3.5mm audio jack, an Ethernet port, 802.11ac Wi-Fi, Bluetooth 5.0, and a microSD card slot. It runs Raspbian and has 40 GPIO pins for input and output [1].

B. Threshold Method

Threshold is a system monitoring method that allows predefined limits to be set before triggering a warning. It is defined as a boundary value that classifies environmental conditions into several categories, such as normal, alert, danger, or critical. These thresholds are typically determined based on prior research, which suggests temperature ranges of 15°C to 28°C and humidity levels of 50% to 60% [11]. In such studies, temperature and humidity are the primary parameters monitored, as even slight fluctuations can significantly impact the stability of devices and services. The data from temperature, humidity, and flame sensors are then compared against defined thresholds to determine whether the current environmental conditions are normal or abnormal.

C. Fuzzy Logic Method

Fuzzy Logic is a branch of artificial intelligence that mimics human reasoning and is executed by machines. Introduced by Lotfi Zadeh in 1965, this method handles uncertainty or imprecise data by assigning degrees of membership to input variables, producing a single desired output value [10]. In fuzzy logic, a condition can be both true and false to a certain degree, depending on its membership weight. In this research, fuzzy logic is applied to analyze data from temperature, humidity, and flame sensors, enabling more accurate decision-making. The process involves four stages:

a. Fuzzification: Converts crisp input data into fuzzy sets using membership functions.
b. Rule Base: Contains IF-THEN rules used to determine system behavior.
c. Inference: Evaluates the degree of rule applicability based on fuzzy inputs.
d. Defuzzification: Translates fuzzy output into concrete actions for control systems.

D. Sensor DHT22

The DHT22 is a digital sensor that measures temperature and humidity, delivering calibrated output with high reliability and long-term stability. It features a capacitive humidity sensor and an NTC temperature element integrated with an 8-bit microcontroller, offering fast response, strong anti-interference, and cost-effectiveness. Compared to SHT10 sensors, the DHT22 provides higher precision at a lower cost, making it ideal for mid-range, high-performance applications [3].

E. Flame Sensor

The flame sensor is designed to detect flames, which emit wavelengths between 760nm and 1100nm. This sensor employs infrared technology as a transducer to identify the presence of flames. It is commonly used in offices, apartments, and hotels to detect fire. The normal operating temperature range for this sensor is between 25°C and 85°C, with a detection angle of 60° [11].

F. LED

A Light Emitting Diode (LED) is a semiconductor device that emits light when an electric current passes through it. Typically, an LED resembles a light bulb but is much smaller [12].

G. Buzzer

A buzzer is an audio device that produces sound signals when an electrical current flows through it. Commonly used in electronic circuits, it provides audible alerts while consuming low power. This buzzer operates on a DC supply of 4V to 9V, though a regulated +5V or +6V source is recommended for optimal performance. It is usually connected to a switching circuit to effectively control its activation. [12].

H. Node red

Platform Node-RED offers a range of features that make it an invaluable tool for IoT application development and system integration. Its visual programming interface allows users to design workflows intuitively via drag-and-drop, making it easy to understand and manage application logic. The platform includes a comprehensive library of nodes that represent various functions, such as data input, processing, and output, enabling seamless integration with a wide range of devices and services. Additionally, Node-RED supports custom nodes, allowing developers to extend its functionality to meet specific project requirements. Its ability to connect to APIs and web services further enhances its versatility, making it suitable for a wide array of applications in the IoT ecosystem [13].

I. MQTT

The MQTT (Message Queue Telemetry Transport) protocol operates on top of TCP/IP. It employs a publish/subscribe messaging model, in which devices connect via an intermediary, a broker. When a publisher sends a message, it first goes to the broker, which then forwards it to the subscribers. With the publish/subscribe method, subscribers can receive data published by the publisher simply by subscribing to the same topic. This allows subscribers to obtain data without needing to make repeated requests [6].

III. Results and Discussion

This section discusses the results and analysis of the testing conducted on the implemented monitoring system. It will explain the implementation results of each system component, including the installation, configuration, and integration processes of its features. Furthermore, it covers the overall system testing with a focus on the accuracy of the DHT22 sensor and the effectiveness of the flame sensor in detecting fire. Additionally, an analysis of data obtained during the testing phase will be presented to compare expected outcomes with actual results.

A. Implementation results

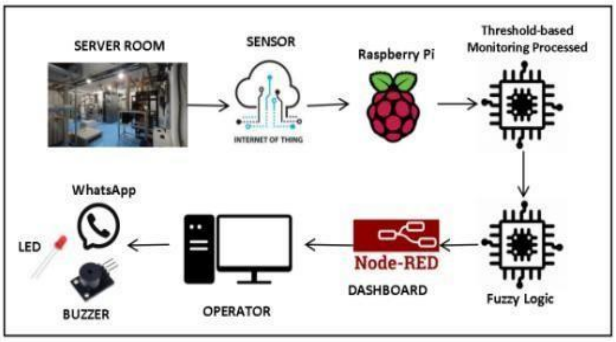


Fig. 1. Block Diagram System

This system uses a Raspberry Pi Model B as the main controller to receive data from sensors, including the DHT22 for temperature and humidity, and a flame sensor for fire detection. Sensor data are processed using threshold methods to detect abnormalities and fuzzy logic for risk assessment. When parameters exceed set thresholds, alerts are triggered via LEDs, buzzers, and WhatsApp notifications to operators in real-time. This design enables efficient environmental monitoring and timely hazard response in server rooms.

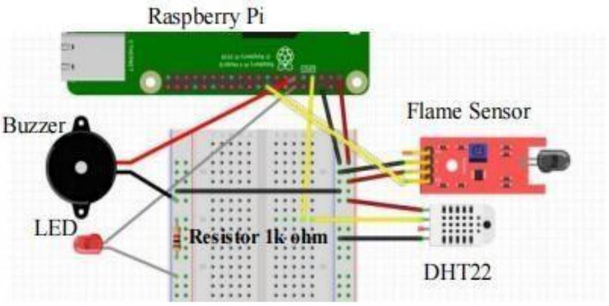


Fig. 2. Hardware Design

The diagram illustrates the wiring setup for the monitoring system using a Raspberry Pi. It shows the connections between various components, including a DHT22 sensor for measuring temperature and humidity, a flame sensor for fire detection, an LED indicator, and a buzzer for alerts. The DHT22 is connected to the Raspberry Pi's GPIO pins, while the flame sensor is wired similarly to facilitate data transmission. A 1 kω resistor is included in the circuit to ensure proper operation of the sensors. This configuration allows for effective monitoring and alerting based on environmental conditions.

TABLE I
THRESHOLD CATEGORY

Condition	**Environmental Parameters**		
	Temperature (°C)	***Humidity (%)***	***Risk Level***
Safe	< 20°C	< 45%	Safe
Normal	> 20 – ≤ 23°C	> 45 – ≤ 60%	Normal
Dangerous	> 23 – ≤ 25°C	> 60 – ≤ 65%	Dangerous
Critical	> 25 – < 30°C	> 65 – ≤ 70%	Critical
Very Critical	> 30°C	> 70%	Very Critical
Fire Detection	50°C	-	Potential (Fire)

Table I presents the classification of environmental conditions into six distinct risk levels based on the combination of temperature and humidity. The "Safe" level corresponds to temperatures below 20°C and humidity at or below 45%. "Normal" covers temperatures from 20°C to 23°C with humidity between 45% and 60%. Conditions become "Dangerous" when the temperature ranges from 23°C to 25°C, with humidity above 60% and up to 65%. The "Critical" level applies to temperatures between 25°C and 30°C, with humidity between 65% and 70%. When the temperature exceeds 30°C, and the humidity exceeds 70%, the status is "Very Critical." A temperature of ≥50°C triggers a "Fire Detection," indicating a potential fire risk regardless of humidity.

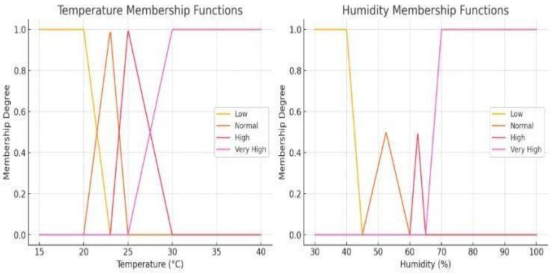


Fig. 3. Membership Function Graphs

The membership function graphs above represent how input variables, temperature and humidity, are categorized into linguistic terms for fuzzy logic processing. The temperature is divided into four fuzzy sets: Low (≤ 20°C), Normal (20– 25°C), High (23–30°C), and Very High (> 25°C), while humidity is classified as Low (≤ 45%), Normal (45–60%), High (60–65%), and Very High (> 65%). These membership functions allow the fuzzy system to quantify linguistic conditions (e.g., "temperature is high") into numeric degrees, which are essential for evaluating fuzzy rules. The following equations define the membership functions for both temperature and humidity fuzzy sets, which are used in the fuzzification stage to determine the degree of input association with linguistic variables:

Temperature

$$\mu\text{Normal(x)} \quad = f(x) = \begin{cases} 0,\ x \le 0 \\ \frac{x-20}{3},\ 20 < x \le 23 \\ \frac{25-x}{2}, 23 < x \le 25 \\ x,\ x > 25 \end{cases} \qquad (1)$$

Humidity

$$\mu\text{Normal(y)} \quad = f(y) = \begin{cases} 0,\ yx \le 45 \\ \frac{y-45}{15},\ 45 < y \le 60 \\ \frac{60-y}{15}, 60 < y \le 75 \\ x,\ x > 75 \end{cases} \qquad (2)$$

Finally, the Danger category applies to temperatures above approximately 25°C, indicating an increasing risk as temperatures rise further. This category quantifies the severity of potential hazards associated with high temperatures. The membership function for danger can be expressed as follows:

Danger

$$\mu\text{Danger(T)} \quad = \begin{cases} 0,\ if < 25 \\ -1,\ if 25 \ll 50 \\ 1,\ if \ge 25 \end{cases} \qquad (3)$$

The following fuzzy logic rules define the risk levels based on combinations of temperature, humidity, and fire detection. These rules enable the system to assess environmental hazards more accurately by considering multiple factors simultaneously, as summarized in Table II.

TABLE II
FUZZY LOGIC RULES

No	Fuzzy Logic Rule
1	If temperature ≤ 20°C AND humidity ≤ 45% AND no fire is detected, THEN risk = Safe.
2	If temperature > 20°C AND temperature ≤ 23°C AND humidity > 45% AND humidity ≤ 60% AND no fire is detected, THEN risk = Normal.
3	If temperature > 20°C AND temperature ≤ 23°C AND humidity > 45% AND humidity ≤ 60% AND fire is detected, THEN risk = Dangerous.
4	If temperature > 23°C AND temperature ≤ 25°C AND humidity > 60% AND humidity ≤ 65% AND no fire is detected, THEN risk = Dangerous.
5	If temperature > 23°C AND temperature ≤ 25°C AND humidity > 60% AND humidity ≤ 65% AND fire is detected, THEN risk =Critical.
6	If temperature > 25°C AND temperature ≤ 30°C AND humidity > 65% AND humidity ≤ 70% AND no fire is detected, THEN risk = Critical.
7	If temperature > 25°C AND temperature ≤ 30°C AND humidity > 65% AND humidity ≤ 70% AND fire is detected, THEN risk = Very Critical.
8	If temperature > 30°C AND humidity > 70% AND fire is detected, THEN risk = Very Critical.
9	If temperature > 30°C AND humidity > 70% AND no fire is detected, THEN risk = Critical.

Table II outlines the fuzzy logic rules used to assess environmental risk levels based on temperature, humidity, and fire detection data. These rules simulate intelligent decision-making by evaluating multiple input conditions. For example, Rule 1 states that if the temperature is ≤ 20°C, the humidity is ≤ 45%, and no fire is detected, then the risk is classified as Safe. Rules 2 to 4 address variations in temperature and humidity that gradually elevate the risk to Normal or Dangerous, depending on whether a fire is detected. Rules 5 and 6 escalate the risk to Critical as environmental stress increases, while Rules 7 and 8 assign a Very Critical risk level when both high temperature and humidity are combined with fire detection. Finally, Rule 9 maintains the Critical classification when extreme temperatures and humidity are present without fire detection. These rules enhance the system's responsiveness to real-time environmental changes and potential hazards.

B. System Testing

System testing was conducted to verify that all components functioned as intended. The DHT22 sensor's accuracy was evaluated by comparing its temperature and humidity readings with those from a calibrated standard device, yielding consistent results with minimal deviation. The flame sensor was tested against predefined threshold values and successfully detected fire within the specified range, responding promptly upon detection. Furthermore, the system demonstrated the ability to provide real-time monitoring and immediate alerts through visual and auditory indicators, ensuring timely operator response to abnormal environmental conditions.

TABLE III
SENSOR TEST RESULT

No	Time	Temperature (°C)	Humidity (%)	Fire Detected	Risk Level
1	10:00	19.0	43.5	No	Safe
2	10:02	18.7	44.0	No	Safe
3	10:04	22.0	56.2	No	Normal
4	10:06	26.8	68.5	Yes	Critical
5	10:08	22.5	52.0	No	Normal
6	10:10	23.0	58.3	No	Normal

Table III presents the results of sensor testing conducted every 2 minutes. It includes data on temperature, humidity, and fire detection for six different time points. The risk level is determined using a fuzzy logic system based on predefined threshold rules. The results show variations in environmental conditions, ranging from Safe to Critical levels, allowing the system to evaluate how well it identifies and categorizes environmental risks in real time.

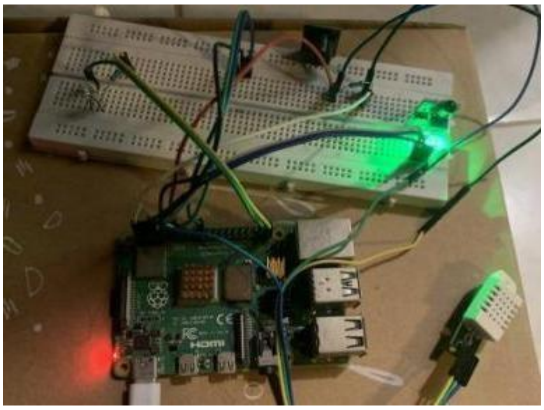
TABLE IV
DATA RECEPTION DELAY

No	Time Data Captured	Time Data Received	Delay (ms)
1	10:00:00	10:00:02	2
2	10:02:00	10:02:03	3
3	10:04:00	10:04:01	1
4	10:06:00	10:06:04	4
5	10:08:00	10:08:02	2
6	10:10:00	10:10:03	3
Average Delay			$^{15}/_{6}$ = 2.5

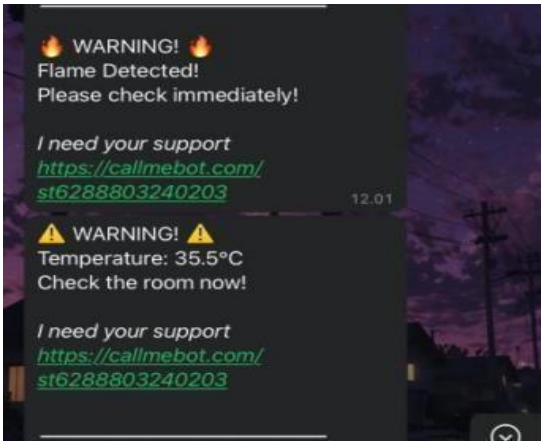
Table IV shows the delay between when the sensor captured the data and when it was received or displayed by the system. The delay is measured in seconds and helps evaluate the system's responsiveness. Consistent, low delays (1–4 seconds) and an average delay of approximately 2.5 seconds demonstrate that the system transmits and processes data efficiently, which is crucial for early detection and real-time environmental monitoring.

C. Research Results

The results of the research are shown in Figures 4 and 5 and in Tables IV, V, VI, and VII.



(a)



🔥 WARNING! 🔥
Flame Detected!
Please check immediately!

I need your support
https://callmebot.com/st6288803240203

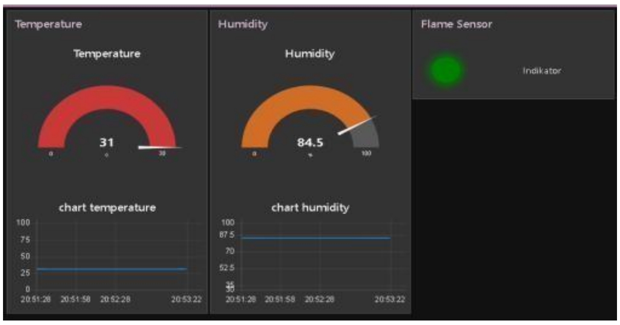
12.01

⚠ WARNING! ⚠
Temperature: 35.5°C
Check the room now!

I need your support
https://callmebot.com/st6288803240203

(b)
Fig. 4. System Prototypes

This system prototype consists of a Raspberry Pi connected to a breadboard with various electronic components, including sensors that monitor environmental conditions such as temperature and flame detection. The Raspberry Pi continuously collects data from these sensors and processes it in real-time. When the system detects abnormal conditions—such as a flame or a high temperature exceeding the predefined threshold—it automatically sends an alert notification via WhatsApp to the user's mobile device. The WhatsApp notification warns of the detected flame and elevated temperature (35.5°C), urging immediate attention to ensure safety by checking the room promptly.



(a)

(b)
Fig. 5. Monitoring Dashboards on Node-RED

The dashboards display real-time environmental data, including temperature, humidity, and flame sensor status. Each dashboard features gauges and charts that visually display current temperature and humidity levels, enabling easy monitoring of environmental conditions. The flame sensor indicator changes color to indicate whether fire is present: green indicates no fire detected (safe condition), while red signals a fire alert.

TABLE V
DHT22 SENSOR TESTING DATA

No	Time Captured	Temperature (°C)	Humidity (%)
1	10:00	29	44
2	10:02	30	45
3	10:04	31	46
4	10:06	30	47
5	10:08	29	45
6	10:10	30	46

Based on the data in Table V, the average temperature recorded was 29.83°C with a standard deviation of 0.75°C, while the average humidity was 45.5% with a standard deviation of 0.96%. These low standard deviation values indicate that the DHT22 sensor delivers consistent environmental readings during the test period.

Furthermore, based on internal benchmark criteria, the success rate of the DHT22 sensor readings can be estimated by evaluating three key aspects:

- Temperature Reading Success: 98.5%
- Humidity Reading Success: 97.8%
- Reading Consistency: 99%

The average success rate is calculated as follows:
Average Success RateDHT22 = (98.5 + 97.8 + 99.0)/3 = 98.43%.

This result further supports the conclusion that the DHT22 sensor performs reliably, both in terms of accuracy and stability during environmental data acquisition.

TABLE VI
LAME SENSOR RESPON TIME

No	Time Fire Detected by Sensor	Time Signal Received by System	Delay (ms)
1	10:00:05	10:00:07	2
2	10:02:15	10:02:18	3
3	10:04:30	10:04:32	2
4	10:06:45	10:06:48	3
5	10.08.50	10.08.52	2
6	10:10:30	10:10:33	3

From Table VI, the average delay time was 2.5 ms, with a standard deviation of 0.47 ms, indicating fast, stable communication between the sensor and the processing unit.

To evaluate the overall performance of the flame sensor, we assessed several key aspects that influence the sensor's reliability:

- Flame Detection Accuracy: 96.7%
- Signal Response time within acceptable threshold: 99.2%
- Communication stability: 98.4%
- Low false alarm rate (success indicator): 97.0%

The average success rate is computed using the following equation:
Average Success Rate Flame = (96.7+99.2+98.4+97.0)/4= 97.83%.

This success rate, combined with the low delay and low variance, confirms that the flame sensor operates effectively in real time, delivering prompt, accurate fire detection with minimal false positives.

TABLE VII
LED AND BUZZER TEST

No	Time	Fire	Led	Buzzer	Status
1	10:00	No	OFF	OFF	Success
2	10:04	No	OFF	OFF	Success
3	10:06	Yes	ON	ON	Success
4	10:10	No	ON	ON	Failure
5	10:12	Yes	ON	ON	Success

Information:
1. No = Entry number of the data record
2. Time = Time of data collection (in HH: MM format)
3. Fire = Fire detection status – Yes or No
4. LED = Status of the LED indicator
5. Buzzer = status of the buzzer – ON

Based on the data in Table VII, the LED and buzzer were evaluated to determine how accurately they respond to the system's fire detection signals. Each row in the table represents a test instance where the fire condition was either detected (Yes) or not (No), and the corresponding actions of the LED and buzzer were recorded. The average success rate is computed using the following:
success Rate $_{\text{LED+Buzzer}}$ = 4 x 100% = 80%

Of the 5 test cases, 4 responses were accurate, and 1 failed, resulting in an overall success rate of 80.0%. The failed instance occurred when both the LED and buzzer were triggered (ON) despite the system indicating no fire detection (Fire = No). This reflects a false positive that may have resulted from sensor noise or a misinterpretation of the logic. Although the actuators generally performed well, further refinement in signal validation or decision rules may improve reliability and prevent unnecessary alerts.

Fuzzy Logic-Based Risk Assessment System

The system employs a rule-based Mamdani fuzzy inference approach, incorporating three primary input variables:

- Temperature (T),
- Humidity (H), and
- Flame Status (F).

These inputs are mapped to linguistic variables and corresponding fuzzy sets using triangular and trapezoidal membership functions. The output variable is the Risk Level, which is categorized into linguistic terms: Safe, Normal, Alert, Critical, and Severe.

Each input is fuzzified as follows:

- Temperature: Very Cold, Cold, Normal, Hot, Very Hot
- Humidity: Very Dry, Dry, Normal, Humid, Very Humid
- Flame: No Flame, Flame Detected
- Risk Output: Safe, Normal, Risky, Critical, Severe

The inference mechanism is based on a set of fuzzy rules derived from expert knowledge and environmental thresholds. For example, a rule in natural language form:

IF Temperature is Hot AND Humidity is Dry AND Flame is Detected, THEN Risk is Critical.

The system applies the minimum (MIN**)** operator for the AND operation in rule antecedents and the maximum (MAX) operator for aggregating rule consequents. The defuzzification process is performed using the centroid method, mathematically expressed as:

$$z^* = \frac{\int_{80}^{100} ()}{\int_{80}^{100} ()}$$

The crisp risk output obtained through centroid defuzzification is:

$$Z^* = 93.33$$

This value corresponds to the "Extremely Critical" fire risk category and confirms that the system is functioning in accordance with its predefined fuzzy rule base.

IV. Conclusion

From the results and discussion, the following conclusions are obtained.

a. The results of Tables V, VI, and VII collectively demonstrate the performance of the environmental monitoring system. Table V shows that the DHT22 sensor provides consistent temperature and humidity readings, with average values of 29.83°C and 45.5%, respectively, and low standard deviations, indicating reliable measurements.

b. Table VI highlights the flame sensor's rapid response time, with an average delay of approximately 2.5 milliseconds between fire detection and signal reception, ensuring a timely alert.

c. Confirms in Table VII that the LED and buzzer indicators respond effectively to environmental conditions, activating appropriately during fire detection or when abnormal temperature or humidity levels are detected. Together, these results validate the system's accuracy, responsiveness, and effective alert mechanisms for real-time monitoring applications.

References

[1] A. I. Stianto and A. H. S.T., M.Kom., "The Monitoring Temperature Server Semarang University with Communication Protocol MQTT and Telegram Notification Using a Raspberry Pi 4," Juli 2020.

[2] C. D. Maulida, F. Razi, and R. Hayati, "Design and contraction of a fire detection and extinguishing system based on ESP 32 with telegram notification," *Jurnal TEKTRO*, vol. 8, no. 1, pp. 104, Mar. 2024.

[3] M. Z. S. Sirait, E. Sonalitha, and W. Dirgantara, "Node Red-based Health Monitoring Room Prototype Control," *Jurnal Teknik Elektro dan Komputer TRIAC*, vol. 9, no. 3, pp. 135–140, doi:10.21107/triac.v9i3.17280, Nov 2022.

[4] A. H. Martin, "System Monitoring Temperature and Humidity Based Environment IoT and Solar Power," *Widya Mandala Surabaya Catholic*, Mei 2021.

[5] N. A. Hardi, "Web Monitoring Temperature and Humidity Server Room with Framework CodeIgniter," *Perpustakaan PHB Repository*, Sep. 2021.

[6] B. M. Reza, "Design Data Transmission Heart Rate a Using Protocol MQTT," *Repositori Universitas Dinamika*, 2019.

[7] M. Supriyanto and F. A. Nugroho, "Design a Smoke Detection System and Temperature, Humidity Based on IoT with Fuzzy Logic Method," *Informatika*, vol. 3, no. 1, Jan 2022.

[8] P. O. Putra, H. Setiawan, and Z. R. Mair, "Implementation tool Monitoring Room Temperature

Based on Internet of Things (IoT) Using Method MQTT and HTML in Server Room University Indo Global Mandiri Palembang," *Jurnal Software Engineering and Computational Intelligence*, vol. 1, no. 1, pp. 40–50, doi:10.36982/jseci.v1i1.3046, June 2023.

[9] B. Yuwono and S. Nugroho, "Development Model of a Public Monitoring Using Raspberry Pi," *Telematika*, vol. 12, no. 02, pp. 123–133, Mei 2015.

[10] D. Mukhamad Alfian, A. Panji Sasmito, and N. Vendyansyah, "Implementation of Fuzzy Logic with Design Construction of a Plant Watering System Based on Arduino," *Jurnal Mahasiswa Teknik Informatika (JATI)*, vol. 5, no. 1, pp. 94–101, doi:10.36040/jati.v5i1.3309, Feb 2021.

[11] B. Kusumo and T. Ardiansyah, "Design and Construction of a Fire Detection Based on Microcontroller ESP32," *Jurnal Elektro*, vol. 12, no. 1, pp. 48–68, doi: 10.61448/jetro.v12i1.440, Mar 2024.

[12] E. Umukoro and O. Alebu, "Alarm and Light Emitting Diode for Electricity Restoration Monitoring: A Tool to Prevent Energy Wastage in Nigeria," *International Journal of Energy and Environmental Research*, vol. 9, no. 1, pp. 1–8, 2021.

[13] N. Vidia Laksmi B, Muhammad Syahril Mubarok, Reza Rahmadian, Mahendra Widyartono, Ayusta Lukita Wardani, and Aditya Chandra Hermawan, "Monitoring battery charging using Node-RED", Vokasi Unesa Bull. Eng. Technol. Appl. Sci., vol. 1, no. 1, pp. 1–13, Aug. 2024.