

IoT-Based Server Room Environmental Monitoring Using ASHRAE TC 9.9 and RoC Alarm

Rachmat Farizky¹⁾, Sahirul Alam^{2)*}

^{1,2)}Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Yogyakarta, Indonesia

¹⁾rachmatfarizky@mail.ugm.ac.id, ²⁾sahirul.alam@ugm.ac.id

Submitted : Aug 1, 2026 | Accepted : Aug 15, 2026 | Published : Oct 5, 2026

Abstract: Server infrastructure is a vital asset for PT Hino Motors Manufacturing Indonesia (HMMI), whose performance is significantly influenced by the stability of temperature and humidity. Previously, environmental monitoring in the server room was still conducted manually and conventionally, leading to risks where hardware failures due to temperature anomalies were often detected too late. This research aims to design and implement a centralized IoT based monitoring system with the integration of ASHRAE TC 9.9 standards and the Rate of Change Alarm algorithm to overcome manual supervision constraints and minimize the potential for hardware damage. The system is built using ESP32 and ESP8266 microcontrollers as sensor nodes, the MQTT protocol for data transmission, and an interactive Next.js-based web dashboard. Frontend implementation includes real-time data visualization via WebSocket and dynamic sensor location maps. Sensor accuracy testing resulted in a temperature MAE of 0.22°C and a relative humidity MAE of 8.1 percentage points. These results indicate that the DHT11 provided adequate temperature measurements for the proposed monitoring application, although its relatively high humidity deviation remains a limitation of the current implementation. QoS testing using Wireshark proves the high reliability of the communication channel, yielding an average latency of 20.19 ms for the ESP32 and 50.85 ms for the ESP8266, packet loss ratio of 0%, and efficient bandwidth usage ranging from 200-450 bytes/s. The system has been successfully deployed on-premise at HMMI's local server, providing a responsive monitoring solution for environmental anomalies and enhancing the IT team's operational efficiency through systematic historical data reporting features.

Keywords: ASHRAE TC 9.9; IoT; Rate of Change; Server Room Monitoring; Web Application

INTRODUCTION

Reliable and secure server infrastructure has become a necessity for companies such as PT Hino Motors Manufacturing Indonesia (HMMI). A server room constitutes a critical component of an organization's information technology infrastructure, functioning as the central location for servers, database storage, applications, and various supporting network devices (Ardiyanto et al., 2024). However, the performance and reliability of a server room are highly influenced by environmental factors, particularly ambient temperature. The recommended operating temperature for server rooms generally ranges from 18°C to 27°C, with a relative humidity level between 40% and 60% (Rizkiawan et al., 2024). Failure to maintain these conditions properly may increase the risk of hardware failure and critical data loss, posing a significant threat to business operations. Therefore, real-time monitoring of server room temperature is essential to prevent operational disruptions and ensure optimal system stability and performance (Kusumah et al., 2023).

Currently, server room management at PT HMMI faces several challenges related to environmental monitoring. The existing monitoring process is conducted manually and conventionally, requiring IT personnel to physically visit each server room location to inspect temperature measurement devices installed within the facility. This approach suffers from major limitations, including inefficiencies in terms of time and labor, as well as the absence of systematically recorded historical data logs for long-term analysis. Furthermore, the lack of a

*Sahirul Alam



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

centralized alarm system often results in delayed detection of temperature anomalies occurring outside working hours or in remote locations, thereby increasing the risk of infrastructure downtime.

The Internet of Things (IoT) has emerged as a promising solution to address these challenges. This technology enables integrated communication and connectivity among various devices through the internet (Junaidi & Ramadhani, 2024). By utilizing temperature and humidity sensors integrated with microcontrollers, the system can continuously monitor and record environmental data. Moreover, the system is capable of providing early warning notifications whenever abnormal conditions are detected within the server room environment (Rifqi et al., 2025). The collected data are stored in a database and can subsequently be processed by the system, resulting in more effective and efficient management.

The RoC approach was selected because the primary objective of the proposed system is to identify rapid short-term temperature changes rather than to perform long-term environmental forecasting. Compared with more complex time-series analysis or machine-learning methods, RoC can be implemented without a large historical training dataset, model-training procedures, or periodic model retraining. Its calculation is also computationally lightweight and directly interpretable because the alarm is associated with an observable temperature change within a defined time interval. These characteristics make RoC suitable for an initial on-premise industrial monitoring system that requires simple, real-time, and transparent early-warning logic.

This study designs and implements a centralized IoT-based monitoring system at PT HMMI in response to the critical need for maintaining appropriate temperature and humidity levels within server rooms. The primary objective of this IoT implementation is to transform the existing manual monitoring process into an automated and centralized digital system. By integrating the ASHRAE TC 9.9 standard and the RoC algorithm, the proposed system is designed to provide comprehensive real-time visibility of environmental conditions across all server rooms, accelerate IT team response times through automated notifications, and establish a historical database to support preventive maintenance of the company's infrastructure.

LITERATURE REVIEW

Previous studies on server-room environmental monitoring can be grouped into several main approaches based on their computing platforms, communication architectures, monitoring interfaces, and environmental assessment methods. The first group employs local or single-board-computer-based monitoring systems. Widiyatmoko and Suharjo (2025) used a Raspberry Pi 5 with a DHT22 sensor, MariaDB storage, and Discord-based visualization, while Vinola et al. (2020) combined Raspberry Pi, MQTT, OpenHAB, and infrared-based air-conditioning control. Ramawardana and Iskandar (2025) also implemented a Raspberry Pi and Flask-based system with MQTT transmission and Telegram notifications. These studies demonstrate that local computing platforms can support data acquisition, storage, visualization, and notification within an integrated architecture. However, their main emphasis is system connectivity and local monitoring functionality rather than standards-based environmental assessment or short-term temperature-trend detection.

The second group utilizes cloud-based IoT platforms and mobile notification services. Assegaf et al. (2024) integrated ESP32, ThingSpeak, and Telegram to provide historical monitoring and on-demand environmental information. Maulana et al. (2023) used Thinger.io and Telegram notifications to reduce the need for manual inspection, whereas Hidayat and Sari (2021) implemented NodeMCU ESP8266, Blynk, and a buzzer activated by a fixed temperature threshold. Related implementations using ThingSpeak and Blynk were also reported by Sholid et al. (2025) and Ilamsyah et al. (2025). Cloud-based approaches provide convenient remote access and centralized visualization, but they depend on external internet connectivity and third-party platforms. In addition, their early-warning mechanisms generally rely on predefined threshold values and do not analyze how rapidly environmental conditions change.

The third group focuses on standalone web-based monitoring and local data management. Khobariah et al. (2022) developed a Wemos D1-based system with PHP and MySQL for monitoring overheating conditions. Wantudi et al. (2023) extended web-based monitoring by integrating temperature, smoke, motion, and electrical sensors, while Budianto (2025) developed a Laravel-based dashboard with an RTC module and auxiliary cooling control. Wibowo et al. (2023) addressed communication limitations by employing LoRa for long-range data transmission without relying on internet access. These studies improve local data ownership, historical archiving, and independence from cloud services. Nevertheless, most systems remain focused on visualization, data logging, and fixed-condition alarms, without incorporating internationally recognized operating standards or temporal change analysis.

The fourth group applies computational intelligence or automated control to improve environmental management. JB et al. (2024) implemented the Fuzzy Tsukamoto method to process temperature and electrical measurements, while Ilmi (2024) used relay-based control to activate cooling devices. Syafii et al. (2024) integrated NodeMCU with infrared-based air-conditioning control, and Pratama et al. (2025) employed fuzzy logic to classify environmental conditions into Normal, Cold, and Hot categories. Nazwansyah et al. (2025) focused on evaluating the measurement accuracy of low-cost sensors. These approaches demonstrate the potential of

intelligent classification and automated response. However, their environmental evaluations are generally based on locally defined categories, control rules, or static thresholds rather than formal compliance with ASHRAE TC 9.9 recommendations. Furthermore, complex intelligent methods may require additional rules, training data, or configuration that are not always necessary for a transparent early-warning mechanism.

Overall, previous studies have successfully addressed several aspects of server-room monitoring, including remote sensing, cloud visualization, local data storage, automated notification, and cooling control. However, three limitations remain evident. First, most systems evaluate environmental conditions using fixed thresholds that only represent the current sensor value. Second, few studies formally associate their warning categories with internationally recognized server-room environmental recommendations. Third, previous systems rarely combine absolute threshold evaluation with temporal analysis of rapid temperature changes. Consequently, a temperature value may still appear acceptable while its rapid upward trend already indicates a developing cooling anomaly.

Existing IoT-based server room monitoring systems generally rely on fixed threshold mechanisms that assume environmental changes occur gradually. In industrial server rooms, however, abrupt thermal fluctuations often precede cooling system failures. Consequently, threshold-based monitoring may generate delayed alerts after safe operating limits have already been exceeded. Furthermore, previous studies rarely integrate internationally recognized environmental standards with temporal trend analysis, limiting their capability for proactive anomaly detection. The novelty of this study lies in the development of a dual-layer environmental assessment framework that combines compliance-based evaluation using ASHRAE TC 9.9 with temporal anomaly detection through the Rate-of-Change algorithm, enabling both threshold validation and early trend detection within a single monitoring architecture.

METHOD

The methodology was structured into five main stages: system architecture and IoT network topology design, IoT device design, database design, web dashboard development along with early warning logic implementation, and IoT network Quality of Service (QoS) testing.

System Architecture

This research infrastructure was implemented on-premise using the company's local server environment. To ensure seamless integration with minimal transmission latency, the infrastructure design was divided into two primary aspects, namely the system architecture and the network topology.

The proposed system operates based on a publish-subscribe communication model. The data flow, device interconnections, and information distribution mechanisms were systematically designed to ensure efficient and reliable operation. The comprehensive integration flow from IoT devices to the local server infrastructure is illustrated in Fig. 1.

At the sensor layer, ESP32 and ESP8266 microcontrollers are utilized to acquire temperature and humidity data from DHT11 sensors. The ESP32 was specifically selected due to its dual-core processing capability, which enables simultaneous management of network connectivity and data processing tasks. The acquired raw data are transmitted to the MQTT broker (Eclipse Mosquitto) through a publish mechanism.

On the server side, a Node.js-based backend functions as an active subscriber that receives and processes incoming sensor data. The backend executes the early warning algorithm, stores the processed data in a MySQL relational database, and manages historical data retention through a cron job mechanism to optimize storage efficiency. Furthermore, real-time data visualization is delivered to a Next.js-based web dashboard through WebSocket integration, enabling continuous monitoring of environmental conditions.

IoT Device Design

The IoT device design focused on the pin interconnection configuration between the microcontroller units and the sensing components. The sensor nodes utilized ESP32 and ESP8266 microcontrollers, both of which functioned as the central control units for integrating the DHT11 temperature and humidity sensors. To provide a clear representation of the physical implementation of component interconnections in the deployment environment, the IoT device wiring configuration is illustrated in detail in Fig. 2.

The DHT11 was selected for the initial PT HMMI implementation due to its low cost, simple integration with ESP32 and ESP8266 devices, and suitability for deploying multiple preliminary monitoring nodes. The system was intended primarily for centralized temperature monitoring and early warning rather than laboratory-grade measurement. Although the DHT22 and SHT-series sensors provide better resolution and humidity accuracy, the DHT11 offered a practical balance between cost, availability, and implementation simplicity for the current deployment.

To ensure stable power delivery during continuous operation within the server room environment, the sensor VCC pin was connected to the 3.3 V output pin of the microcontroller, while the GND pin was connected through

a common grounding configuration. The sensor data pin was interfaced with the respective General Purpose Input/Output (GPIO) pin of each microcontroller.

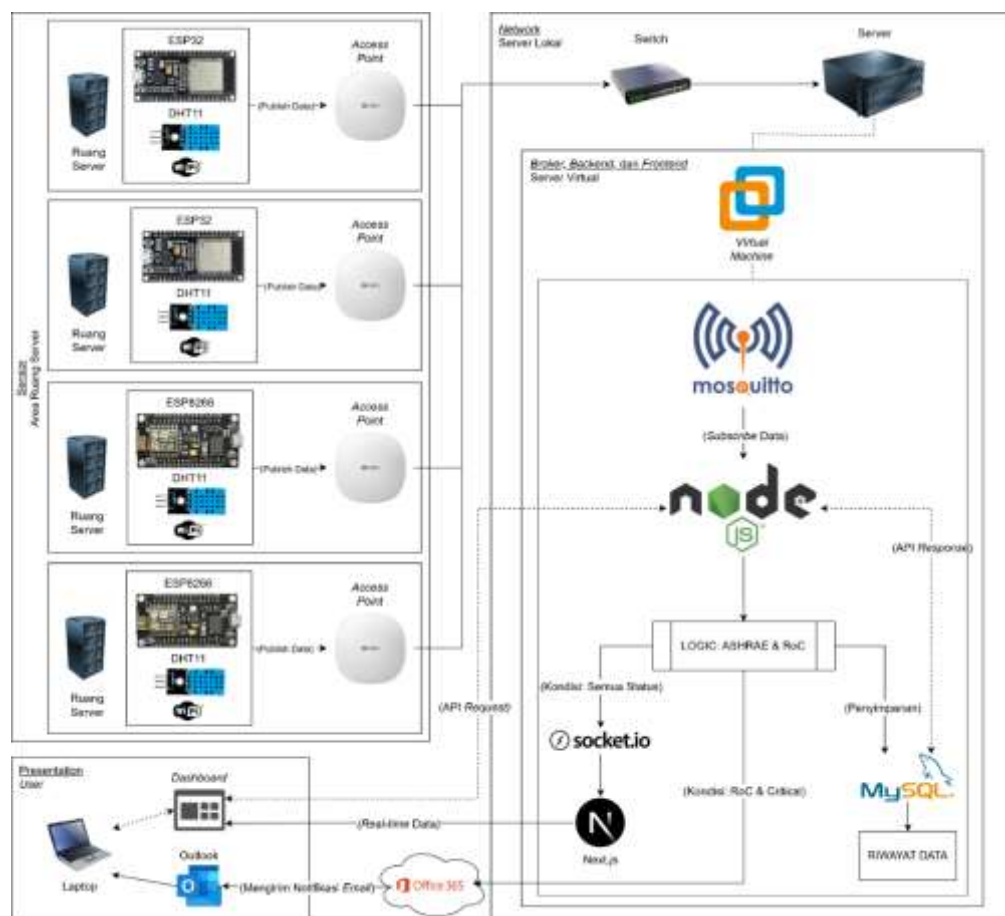


Fig. 1. System architecture of the server room temperature and humidity monitoring system

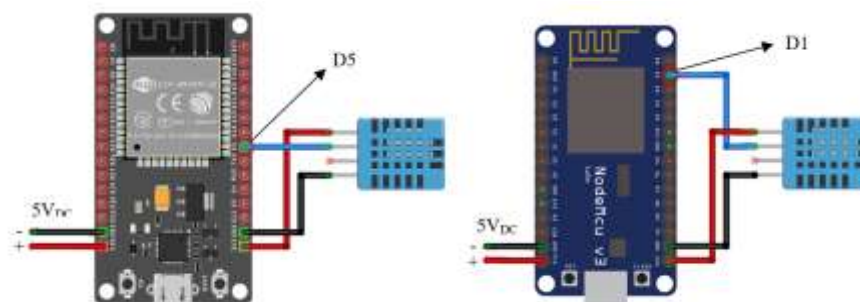


Fig. 2. IoT device wiring diagram

Database Design

Historical data storage and user session management were designed using the MySQL relational database management system (RDBMS). The database schema was normalized to maintain data integrity and optimize query execution performance within the backend application. The structure of the monitoring database is illustrated in Fig. 3.

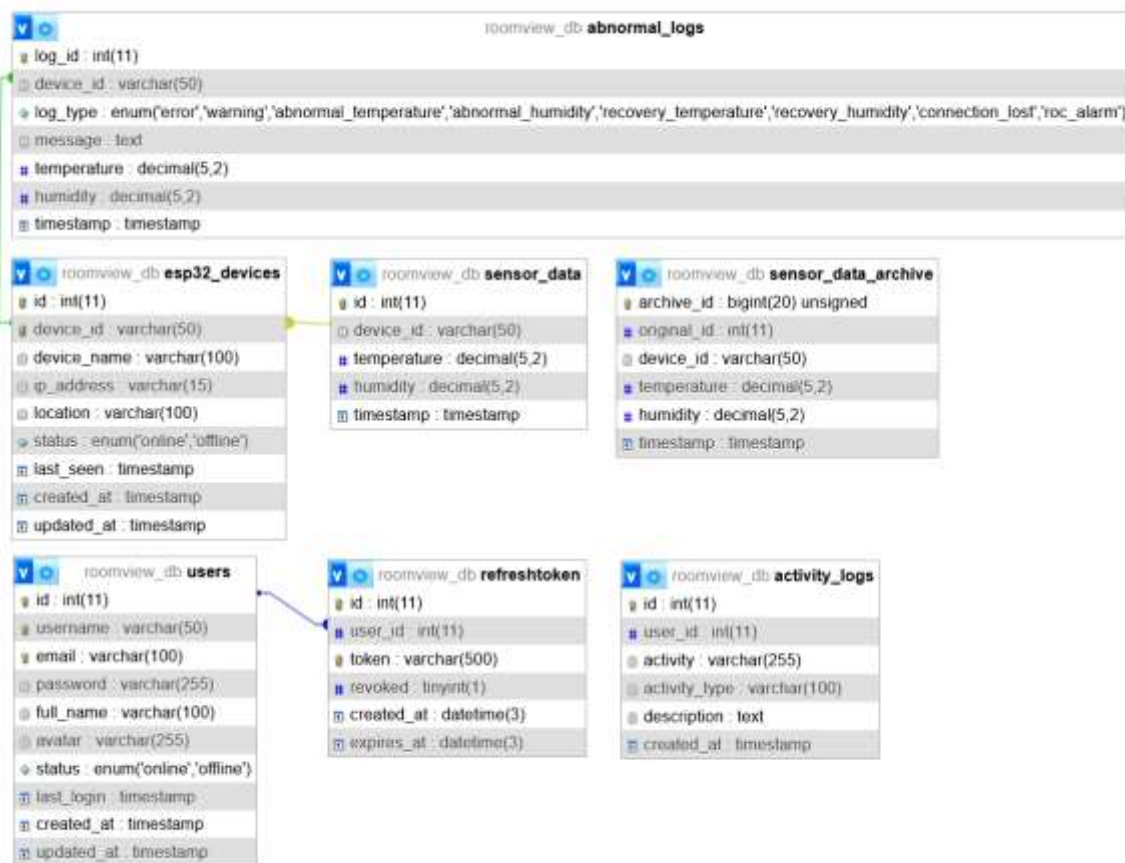


Fig. 3. Database schema of the monitoring system

Web Dashboard Development

The user interface layer was developed using the React.js-based Next.js framework to provide a responsive web application. The dashboard was designed to present centralized visualization of environmental data collected from multiple sensor nodes. A wireframe design approach was employed to define the layout and organization of interface components prior to implementation.

As the primary security gateway of the system, an authentication page was designed to ensure that only authorized IT team can access the monitoring platform. The layout of the login page is illustrated in Fig. 4. The main dashboard page, which is displayed after successful authentication, was designed to provide dynamic data trend visualization through interactive charts and environmental status cards. The structural layout of the dashboard page is presented in Fig. 5.



Fig. 4. Login page wireframe

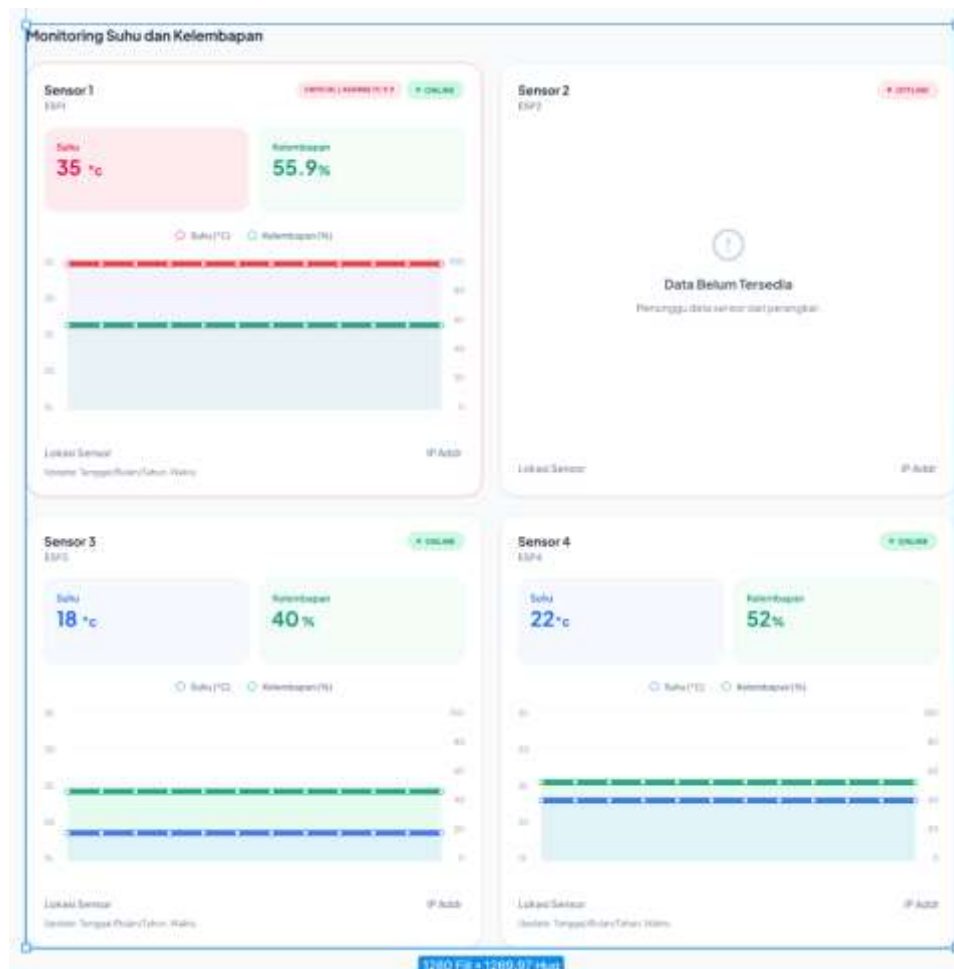


Fig. 5. Dashboard page wireframe

To facilitate device management, inventory tracking, and the visualization of microcontroller deployment within the company's local network infrastructure, the device management page was designed as shown in Fig. 6. The interface structure dedicated to displaying system activity records, historical sensor data, and anomaly-related early warning logs is illustrated in Fig. 7. A dedicated user management page was also designed to support profile administration, user role management, and account configuration for authenticated IT team. The layout of the user management page is presented in Fig. 8.

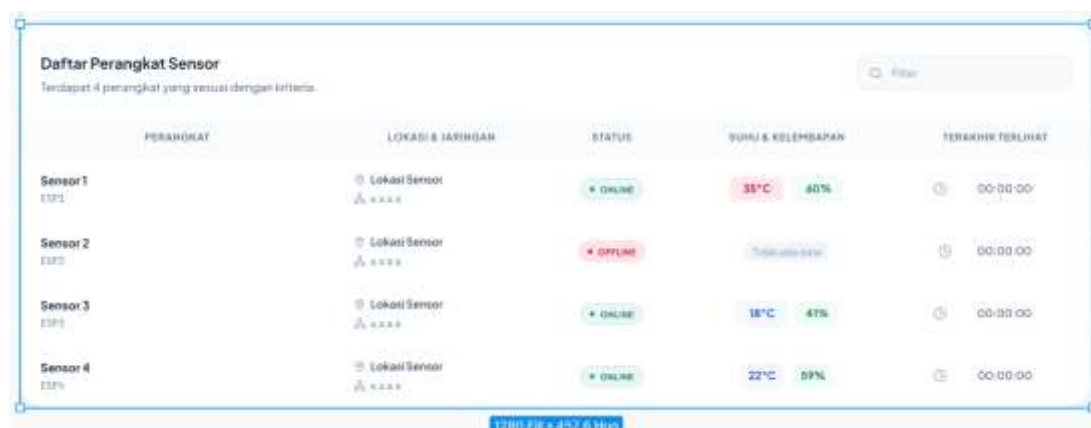


Fig. 6. Device page wireframe

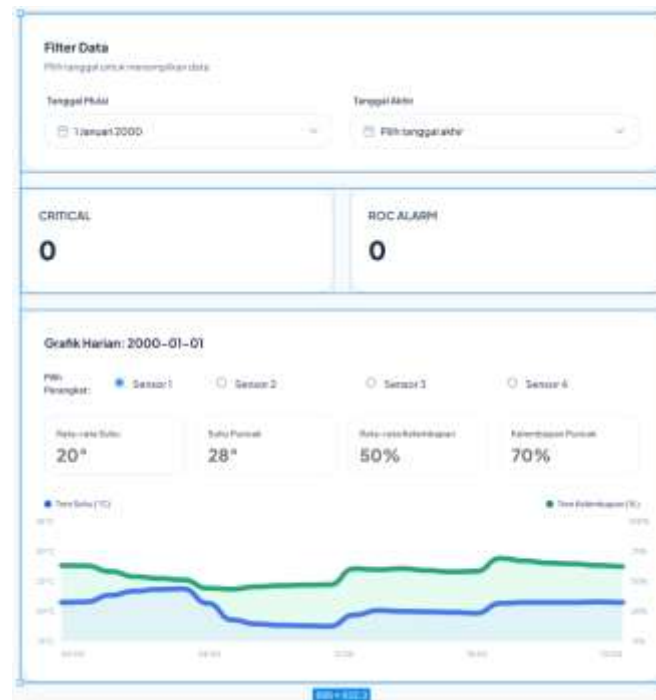


Fig. 7. Log page wireframe

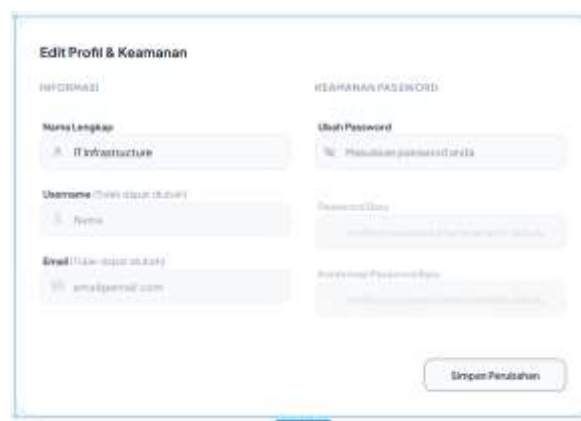


Fig. 8. User page wireframe

Monitoring and Early Warning Logic Development

The anomaly detection logic was implemented on the backend side and executed in real time to prevent potential degradation of hardware performance. The monitoring mechanism was designed based on two primary parameters, namely the ASHRAE TC 9.9 environmental standard and the RoC algorithm.

1. ASHRAE TC 9.9-Based Environmental Assessment

The system automatically validates incoming sensor data against the environmental requirements specified in the ASHRAE TC 9.9 standard for server room operation. Based on the measured temperature and humidity values, the monitored environment is classified into three operational status levels: Normal, Warning, and Critical. A warning or critical condition is triggered whenever the environmental parameters exceed the recommended operating thresholds. RoC was adopted as a lightweight trend-analysis mechanism because it can detect short-term thermal changes directly from consecutive sensor measurements without requiring a trained predictive model.

2. RoC Algorithm

In addition to threshold-based monitoring, a temporal trend-analysis mechanism was implemented using the RoC algorithm to identify rapid temperature increases. The temperature RoC at time t is calculated as:

$$RoC_t = \frac{T_t - T_{t-\Delta t}}{\Delta t} \quad (1)$$

where T_t is the current temperature, $T_{t-\Delta t}$ is the temperature recorded at the beginning of the observation interval, and Δt is the observation duration. In the implemented system, Δt was set to five minutes. Because the alarm logic is expressed as an accumulated temperature increase during this interval, the RoC alarm condition is defined as:

$$A_{RoC}(t) = \begin{cases} 1, & T_t - T_{t-\Delta t} \geq 2^\circ\text{C} \\ 0, & T_t - T_{t-\Delta t} < 2^\circ\text{C} \end{cases} \quad (2)$$

where $A_{RoC}(t) = 1$ indicates that a rapid temperature-rise alarm is activated. Thus, the system generates an anomaly alert when the temperature increases by at least 2°C within the five-minute observation window, even if the current temperature has not yet exceeded the upper limit defined by the ASHRAE-based assessment.

Any anomaly identified by either the ASHRAE TC 9.9 assessment or the RoC algorithm activates the early warning mechanism. Upon detection, the system delivers real-time in-application notifications through WebSocket integration on the monitoring dashboard and simultaneously sends automated email alerts to the IT team via the Office 365 SMTP service.

Testing Scenarios

The developed system was subsequently validated through a series of empirical and quantitative evaluations to assess its functionality, accuracy, and reliability.

1. Functional Testing

Functional testing was conducted to verify the reliability of the software layer and ensure that all system components operated as intended. The scope of this evaluation included validation of the JSON Web Token (JWT)-based authentication mechanism during user login sessions, verification of cron job execution scheduling within the MySQL database environment, and assessment of the success rate of automated email delivery when ASHRAE TC 9.9 or RoC alarm conditions were triggered.

2. Sensor Accuracy Testing

The accuracy of temperature and humidity measurements obtained from the DHT11 sensor was evaluated by comparing the readings against those recorded by an HTC-2 reference instrument. Measurement accuracy was quantified using the Mean Absolute Error (MAE), calculated according to (1).

$$MAE = \frac{\sum_{i=1}^n |x_i - y_i|}{n} \quad (3)$$

Where x_i represents the measurement value obtained from the HTC-2 reference device, y_i denotes the corresponding reading from the DHT11 sensor, and n indicates the total number of test samples.

3. IoT Network Quality of Service (QoS) Testing

The communication performance of the IoT monitoring system within the local network was evaluated according to the Telemetry service category specified in ITU-T Recommendation G.1010 (Union, 2004). Wireshark was deployed on the server side to capture and isolate JSON packet traffic originating from the microcontroller IP addresses.

The QoS evaluation focused on three performance metrics: latency, packet loss percentage, and throughput measured through TCP stream analysis. Packet loss was calculated using (2).

$$PL = \left(\frac{\sum P_t - \sum P_r}{\sum P_t} \right) \times 100\% \quad (4)$$

Where PL denotes the packet loss percentage, $\sum P_t$ represents the total number of packets transmitted by the microcontroller, and $\sum P_r$ indicates the total number of packets successfully received by the server.

RESULT

The IoT-based server room monitoring system implemented at PT HMMI was validated through a series of structured experiments. The evaluation focused on three primary aspects: software functionality, sensor measurement accuracy, and communication network performance.

Functional Testing Results

Functional testing was conducted to verify the integration and reliability of the entire system workflow, ranging from security mechanisms to backend automation processes.

1. Access Security and Authentication

Testing of the user session management module based on a JWT architecture demonstrated that the system successfully enforced user access control in real time and validated token expiration through the management of refresh tokens stored in the database.

2. Automated Data Migration

Evaluation of the MySQL cron job functionality showed that the automated scheduling mechanism successfully migrated large volumes of real-time monitoring data from the primary table to a long-term archival table at predefined intervals without generating significant increases in server computational workload.

3. Early Warning Notification System

To validate the early warning mechanism, sensor data were intentionally manipulated to exceed the operational thresholds defined by the ASHRAE TC 9.9 standard and the RoC algorithm. Upon detecting these anomaly conditions, the Node.js backend immediately triggered the notification workflow. The testing results confirmed that the status cards displayed on the dashboard dynamically changed according to the detected environmental condition, while warning emails were successfully delivered to the IT team through the Office 365 SMTP service. These results demonstrate the effectiveness of the proposed notification mechanism in providing timely alerts for potential environmental risks within the server room.

The functional test also confirmed that the RoC-based warning logic was successfully executed when the recorded temperature increase satisfied the predefined condition of $\Delta T \geq 2^{\circ}\text{C}$ within five minutes. Once this condition was detected, the backend generated an anomaly status and transmitted the warning through both the dashboard notification and automated email channels. This result verifies the functional integration of temperature-trend analysis with the system's notification mechanism. However, the test was intended to verify correct system behavior rather than to quantify classification performance using metrics such as precision, recall, or false alarm rate.

Sensor Accuracy Testing

The integrity of the server room environmental data was validated through an accuracy assessment of the DHT11 sensor by directly comparing its measurements with those obtained from a Thermo Hygrometer Clock HTC-2 reference instrument. The experiment was conducted in the First Floor Office Server Room, where environmental conditions were regulated using a Daikin FTC45NV14 air-conditioning unit. The comparative temperature measurement results are presented in Table 1. Similarly, the comparative results for relative humidity measurements obtained from the HTC-2 instrument and the DHT11 sensor are presented in Table 2.

Table 1. Temperature measurement comparison between the DHT11 sensor and the HTC-2

Time	DHT11 ($^{\circ}\text{C}$)	HTC-2 ($^{\circ}\text{C}$)	Absolute Error
13:17	21.4	21.5	0.1
13:18	21.4	21.4	0
13:19	21.4	21.4	0
13:20	21	21.3	0.3
13:21	21	21.3	0.3
13:22	21	21.3	0.3
13:23	21	21.3	0.3
13:24	21	21.3	0.3
13:25	21	21.3	0.3
13:26	21	21.3	0.3
MAE			0.22

Table 2. Relative humidity measurement comparison between the DHT11 sensor and the HTC-2

Time	DHT11 (%)	HTC-2 (%)	Absolute Error
13:17	58	51	7
13:18	58	50	8
13:19	58	51	7
13:20	59	51	8
13:21	59	50	9
13:22	59	50	9
13:23	59	51	8
13:24	59	50	9
13:25	59	51	8
13:26	59	51	8
MAE			8.1

To provide a visual representation of the continuous environmental data variations throughout the experimental period, comparative trend graphs were generated for each monitored parameter. Fig. 9 illustrates the temperature

measurement trends recorded by the DHT11 sensor and the HTC-2 reference instrument, while Fig. 10 presents the corresponding comparison for relative humidity measurements.

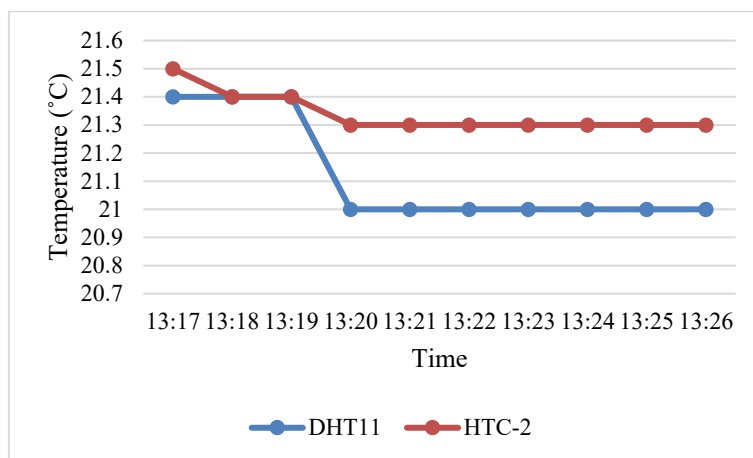


Fig. 9. Comparative temperature measurement trends between the DHT11 sensor and the HTC-2

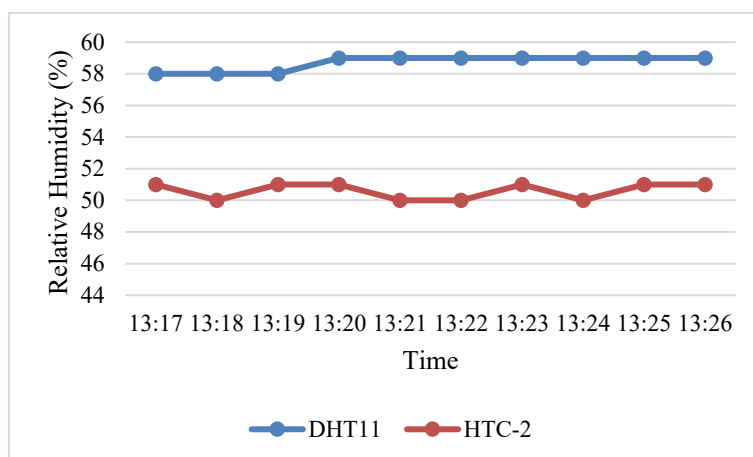


Fig. 10. Comparative relative humidity measurement trends between the DHT11 sensor and the HTC-2

IoT Network QoS Evaluation

The performance of wireless data transmission from the sensor nodes to the company's local server was quantitatively evaluated through packet capturing using Wireshark to assess the reliability of the MQTT communication protocol. The analysis was conducted for each Quality of Service (QoS) parameter as follows.

1. Latency Metric

This test measured the time interval between the transmission of an MQTT Publish packet and its successful reception by the broker, followed by the transmission of a TCP Acknowledgment (ACK) packet from the server. Latency measurements were collected from 12 data samples over a one-minute observation period. The results are summarized in Table 3.

Table 3. IoT network latency measurement results

Parameter	ESP32	ESP8266
Number of Samples	12 Samples	12 Samples
Minimum Latency (ms)	13.29 ms	41.84 ms
Maximum Latency (ms)	26.68 ms	59.53 ms
Average Latency (ms)	20.19 ms	50.85 ms
ITU-T G.1010 Requirement	< 250 ms	< 250 ms
Performance Category	Excellent	Excellent

2. Packet Loss Evaluation

The reliability of packet delivery was assessed by comparing the total number of packets successfully received by the server against the total number of packets transmitted by both microcontrollers. The detailed packet loss calculations are presented in Table 4.

Table 4. IoT network packet loss calculation results

Parameter	ESP32	ESP8266
Total Transmitted Packets	500 Packets	501 Packets
Total Received Packets	500 Packets	501 Packets
Total Packet Loss	0 Packet	0 Packet
Packet Loss Ratio (%)	0%	0%
Performance Category	Excellent	Excellent

3. Throughput Evaluation

Throughput testing was conducted to measure the network bandwidth consumption per second while the microcontrollers actively transmitted MQTT JSON payloads. The bandwidth utilization observed on the server side is summarized in Table 5.

Table 5. IoT network throughput measurement results

Parameter	ESP32	ESP8266
Average Throughput	~230 bytes/s	~250 bytes/s
Peak traffic	~420 bytes/s	~450 bytes/s
Load Classification	Very Low	Very Low
Performance Category	Excellent	Excellent

DISCUSSION

The performance characteristics of the integrated server room environmental monitoring system can be comprehensively assessed based on the quantitative data obtained from the experimental evaluations. The functional testing results demonstrated that the implementation of JWT-based security mechanisms and automated database cron jobs contributed significantly to the optimization of the company's local server infrastructure. The periodic migration of dynamic log data to long-term archival tables effectively reduced the index size of the primary MySQL tables. This strategy is essential for preventing degradation of server I/O performance, thereby ensuring that daily visualization queries remain responsive without generating computational resource spikes on the main system.

The analysis of measurement deviations showed that the DHT11 provided consistent temperature readings under the evaluated conditions. The MAE value of 0.22 reported in Table 2 is substantially lower than the tolerance limits commonly observed in commercial environmental monitoring instruments. This level of measurement consistency indicates that the sensor can support preliminary temperature monitoring and early warning under the evaluated operating conditions. Furthermore, the accurate temperature measurements enable the RoC alarm algorithm to operate with high sensitivity, allowing proactive detection of sudden temperature increases ($\Delta T \geq 2^{\circ}\text{C}$ within 5 minutes). The integration of WebSocket technology within the Next.js dashboard architecture further minimizes data update delays to less than three seconds, providing IT team with near real-time visibility of critical environmental conditions and facilitating preventive maintenance actions before overheating-related hardware failures occur.

Despite the relatively high humidity deviation, the DHT11 remained useful for the current implementation because the primary operational objective was to detect temperature anomalies and provide centralized early warnings. Its temperature MAE of 0.22°C indicates sufficient consistency for observing thermal conditions and supporting the RoC-based alarm within the evaluated environment. Nevertheless, the humidity MAE of 8.1 percentage points limits its suitability for applications requiring precise humidity compliance assessment. Therefore, the current results should be interpreted as evidence of suitability for preliminary temperature-oriented monitoring rather than high-precision environmental measurement. Future deployments should consider DHT22, SHT31, or industrial-grade sensors when more accurate humidity monitoring is required.

The results presented in Table 3 indicate a notable limitation in relative humidity measurements, where the MAE reached 8.1. This relatively high deviation can be attributed to the inherent technical limitations of the DHT11 sensor, which exhibits lower sensitivity to dynamic humidity variations in continuously air-conditioned environments. These findings provide valuable insights for future system enhancements. Potential improvements include the implementation of calibration offset compensation within the microcontroller firmware or the adoption

of industrial-grade sensing components, such as SHT-series sensors or the DHT22, to achieve more stable and accurate humidity measurements.

The implementation of a publish-subscribe architecture based on the MQTT protocol demonstrated highly efficient utilization of local network bandwidth within the PT HMMI infrastructure from the perspective of communication network performance. As shown in Table 6, the average bandwidth consumption of both microcontrollers remained extremely low, below 1 KB/s (approximately 230 bytes/s and 250 bytes/s, respectively). The minimal throughput deviation between devices, approximately 20 bytes/s, suggests that differences in internal hardware architectures between the ESP32 (dual-core) and ESP8266 (single-core) have negligible impact on JSON payload transmission at the application layer. The lightweight nature of the telemetry data confirms the long-term scalability of the proposed system, minimizing the risk of throughput bottlenecks or network saturation even when a substantially larger number of sensor nodes are deployed in the future.

The reliability of data transmission was further validated through the latency and packet loss evaluations. According to the ITU-T Recommendation G.1010 requirements for telemetry applications, end-to-end latency should remain below 250 ms. The average latency values obtained during testing, namely 20.19 ms for the ESP32 and 50.85 ms for the ESP8266, are significantly below this threshold and therefore indicate excellent communication performance. These low latency values ensure timely delivery of environmental monitoring data and support real-time visualization on the dashboard.

The packet loss evaluation yielded a value of 0%, indicating that all transmitted packets were successfully received throughout the testing period. This result suggests the absence of intermittent packet-dropping events across both the Wi-Fi Access Layer and the Ethernet-based network infrastructure. The achieved reliability can be attributed to the stability of the local network environment, the robustness of the MQTT communication architecture, and the effective operation of the underlying network infrastructure. Consequently, the proposed system is capable of maintaining dependable and uninterrupted environmental monitoring under normal operational conditions.

The average latency of the proposed ESP32 node was 20.19 ms, which was numerically lower than the 1.2 s average upload latency reported by Sholid et al. (2025) for an ESP32-based monitoring system connected to the ThingSpeak cloud platform through HTTP. The lower latency in the present study is likely associated with its on-premise MQTT architecture, where data are transmitted and processed within the PT HMMI local network without passing through an external cloud platform. Nevertheless, this comparison should be interpreted cautiously because the studies used different protocols, network conditions, latency measurement points, sampling intervals, and test durations. Therefore, the result demonstrates the low communication delay of the proposed local architecture but does not constitute a controlled benchmark against the ThingSpeak system.

The temperature MAE of 0.22°C in the present study was also lower than the 0.6°C temperature MAE reported by Sholid et al. (2025) using a DHT22 sensor. However, the DHT22 produced a substantially lower humidity error of 2.1% RH compared with the 8.1-percentage-point humidity MAE obtained from the DHT11 in this study. Nazwansyah et al. (2025) reported a temperature accuracy of 98.9% for a DHT11-based monitoring system, supporting the general suitability of this low-cost sensor for temperature monitoring. A direct numerical comparison with the present MAE cannot be made because Nazwansyah et al. reported percentage accuracy rather than MAE and used different reference instruments and testing procedures. Overall, these findings indicate that the proposed system achieved favorable temperature accuracy, while humidity precision remains a limitation of the DHT11 sensor.

Table 6 summarizes the differences between the proposed system and two closely related studies. The comparison highlights the communication advantage of the local MQTT architecture, the favorable temperature measurement deviation, and the remaining limitation in DHT11 humidity accuracy. It also shows that the main distinction of the proposed system lies in combining ASHRAE-based environmental assessment with RoC-based temperature-trend detection.

From a scalability perspective, the MQTT publish-subscribe architecture allows additional sensor nodes to be integrated using separate topics without substantially modifying the monitoring workflow. The relatively small bandwidth consumption of each node also indicates the potential to support a larger deployment. However, increasing the number of nodes may raise the processing and storage load on the MQTT broker, backend server, database, and WebSocket service. Therefore, broker capacity, data-retention intervals, and database performance should be evaluated before deployment across a larger number of server rooms.

From an operational cost-benefit perspective, the proposed system replaces repeated manual inspections with centralized real-time monitoring, automated historical data recording, and email-based anomaly notifications. The main implementation costs include sensor nodes, network connectivity, and local server resources, while the expected benefits include reduced inspection effort, faster anomaly awareness, and improved availability of environmental records for preventive maintenance. Since detailed labor and infrastructure cost data were not available, the present study provides a qualitative rather than monetary cost-benefit assessment.

Table 6. Comparison with Related IoT-Based Environmental Monitoring Studies

Study	Architecture and protocol	Sensor	Communication performance	Temperature accuracy	Humidity accuracy	Warning mechanism
Sholid et al. (2025)	ThingSpeak cloud, HTTP	DHT22 and ESP32	Average latency: 1.2 s; transmission success: 99%	MAE: 0.6°C	MAE: 2.1% RH	Configurable threshold
Nazwansyah et al. (2025)	ThingSpeak web platform	DHT11, LM35, and Arduino	Not reported	DHT11 accuracy: 98.9%; LM35 accuracy: 98.3%	Average DHT11 reading reported; error not reported	Monitoring based on recommended range
This study	On-premise server, MQTT and WebSocket	DHT11 with ESP32/ESP8266	ESP32 latency: 20.19 ms; ESP8266 latency: 50.85 ms; packet loss: 0%	MAE: 0.22°C	MAE: 8.1 percentage points	ASHRAE TC 9.9 assessment and RoC alarm

CONCLUSION

The design, implementation, and evaluation stages conducted in this study successfully addressed the limitations of the manual server room monitoring process. The results demonstrate that the digitalization of environmental monitoring infrastructure can significantly improve operational efficiency and data reliability within industrial environments.

The proposed IoT-based server room monitoring architecture successfully integrated ESP32 and ESP8266 sensor nodes into a centralized web dashboard. Through the utilization of the MQTT communication protocol and the Next.js framework, the system provides real-time environmental monitoring capabilities that effectively replace the conventional manual monitoring approach previously employed at PT HMMI.

The implementation of the ASHRAE TC 9.9 standard proved effective as a reference framework for defining temperature and humidity thresholds to ensure the reliability of server room infrastructure. Sensor accuracy evaluation yielded a MAE of 0.22 for temperature measurements, indicating that the proposed system can serve as a reliable environmental monitoring instrument for server room applications.

The automated evaluation mechanism based on the RoC Alarm algorithm successfully detected temperature anomalies through continuous trend analysis during both working hours and non-working hours. The system was capable of identifying abnormal temperature increases ($\Delta T \geq 2^{\circ}\text{C}$ within 5 minutes) and distributing notifications through both email and in-application alert mechanisms. This capability enables faster technical response times and supports preventive maintenance activities before overheating conditions can result in infrastructure failures.

REFERENCES

- Ardiyanto, T., Supriadi, C., & Priyadi. (2024). Rancang Bangun Sistem Monitoring Deteksi Dini Kebakaran Ruang Server Berbasis Iot. *Jurnal Ilmiah Sistem Informasi*, 3(3), 35–45. <https://doi.org/10.51903/432zjf74>
- Assegaf, M., Soetedjo, A., & Sotyohadi. (2024). SISTEM MONITORING RUANG SERVER BERBASIS IOT (INTERNET OF THING) DI PT. RADNET DIGITAL INDONESIA. *Magnetika*, 08(1), 1–9.
- Budianto. (2025). Implementasi Sistem Monitoring Suhu dan Kelembaban Ruang Server Berbasis Internet Of Things Menggunakan Nodemcu ESP8266 dengan Antarmuka Web Laravel. *SAINSTECH: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, 35(3), 23–36.
- Hidayat, D., & Sari, I. (2021). MONITORING SUHU DAN KELEMBABAN BERBASIS INTERNET of THINGS (IoT). *Jurnal Penelitian Teknik Informatika*, 4(1), 525–530.
- Ilamsyah, Tandilintin, A., Nullah, M. R. F., & Nulhakim, L. (2025). PERANCANGAN ALAT MONITORING SUHU DAN KELEMBABAN DATA CENTER BERBASIS MICROCONTROLLER DI PT TELKOM DATA EKOSISTEM. *JURNAL SISTEM INFORMASI DAN TEKNOLOGI (SINTEK)*, 1(2), 108–114.
- Ilmi, F. A. (2024). SISTEM MONITORING SUHU DAN KELEMBABAN PADA RUANG SERVER BERBASIS INTERNET OF THINGS (STUDI PADA RUMAH SAKIT ROEMANI MUHAMMADIYAH SEMARANG). *Jurnal Mahasiswa*, 6(3), 14–24. <https://doi.org/10.51903/jurnalmahasiswa.v6i3.229>
- JB, S. N., Huzaeni, & Salahuddin. (2024). Air Temperature and Humidity Monitoring System for Server Rooms and Data Centers Using the Fuzzy Tsukamoto Method with IoT. *Journal of Artificial Intelligence and Software Engineering*, 4(2), 104–115. <https://doi.org/10.30811/jaise.v4i2.6132>
- Junaidi, & Ramadhani, K. (2024). EFEKTIVITAS INTERNET OF THINGS (IOT) PADA SEKTOR PERTANIAN. *Jurnal Teknologi Komputer Dan Sistem Informasi*, 4(1), 12–15.
- Khobariah, N. F., Hermawan, P. D. S., & Kusumadiarti, R. S. (2022). SISTEM MONITORING SUHU DAN KELEMBAPAN RUANG SERVER BERBASIS WEMOS D1. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 07(01), 32–42.

- Kusumah, R., Islam, H. I., & Sobur, S. (2023). Sistem Monitoring Suhu dan Kelembaban Berbasis Internet of Things (IoT) Pada Ruang Data Center. *Journal of Applied Informatics and Computing*, 7(1), 82–88. <https://doi.org/10.30871/jaic.v7i1.5199>
- Maulana, R. F., Ramadhan, M. A., Maharani, W., & Maulana, M. I. (2023). Rancang Bangun Sistem Monitoring Suhu dan Kelembaban Berbasis IOT Studi Kasus Ruang Server IT Telkom Surabaya. *Indonesian Journal of Multidisciplinary on Social and Technology*, 1(3), 224–231.
- Nazwansyah, M. A., Usman, D., & Ningtias, W. A. (2025). SISTEM MONITORING SUHU DAN KELEMBABAN BERBASIS WEB SEBAGAI IMPLEMENTASI INTERNET OF THINGS. *Seminar Nasional Pendidikan, Teknologi, dan Kesehatan (TEKAD)*
- Pratama, G. W. P., Yulistiyanti, D., & Suaedah, S. (2025). PENERAPAN METODE FUZZY LOGIC UNTUK MONITORING SUHU DAN KELEMBABAN RUANGAN DATA CENTER DI PT LION AIR. *Jurnal Rekayasa Komputasi Terapan*, 05(02), 120–128.
- Ramawardana, M. S., & Iskandar, J. (2025). Perancangan Sistem Monitoring Suhu dan Kelembaban Menggunakan Sensor DHT22. *Journal of Informatics and Computer Science (JOINCOS)*, 2(1), 1–5.
- Ridla, M. A., & Rahman, M. F. (2024). Perancangan Prototype Monitoring Suhu Berbasis Internet Of Things (IoT). *JUSIFOR: Jurnal Sistem Informasi Dan Informatika*, 3(1), 72–79. <https://doi.org/10.33379/jusifor.v3i1.4367>
- Rifqi, K. S., Irawan, I., Irawan, H., Novita, I., & Chandra, J. C. (2025). Sistem Monitoring Dan Early Warning Suhu Serta Kelembaban Ruang Server Berbasis IOT Dengan Ambang Batas Real-Time Yang Dapat Disesuaikan Melalui Aplikasi Seluler. *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, 22(1), 88–96. <https://doi.org/10.36080/bit.v22i1.4021>
- Rizkiawan, M. A., Ramza, H., Nuroji, & Sofwan, A. (2024). Pemantauan Ruang Data Center Berdasarkan Suhu dan Kelembaban Dengan Internet of Things Data Center Room Monitoring Based on Temperature and Humidity with Internet of Things. *Jambura Journal of Electrical and Electronics Engineering*, 6(2), 115–123.
- Sholid, R. G., Yanto, H., Alhaq, H., & Wijanarko, S. D. (2025). Rancang Bangun Sistem Digital Pemantau Suhu dan Kelembaban Ruang Server Berbasis IoT. *Jurnal Sains Informatika Terapan (JSIT)*, 04(02), 314–319.
- Syafii, O. M. N., Astutik, R. I. I., & Hindarto, H. (2024). Penerapan Monitoring dan Controlling Suhu Ruang Server Berbasis Internet of Things (IoT). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 8(1), 282–291. <https://doi.org/10.30865/mib.v8i1.7141>
- Union, I. T. (2004). *International Telecommunication Union*. <https://www.itu.int/itudoc/itu-t/86097.pdf>
- Vinola, F., Rakhman, A., & Sarjana. (2020). Sistem Monitoring dan Controlling Suhu Ruangan Berbasis Internet of Things. *Jurnal Teknik Elektro Dan Komputer*, 9(2), 117–126.
- Wantudi, A., Triayudi, A., & Benrahman. (2023). SYSTEM MONITORING MOTION, SMOKE, LISTRIK, SUHU DAN KELEMBABAN PADA DATA CENTER MENGGUNAKAN NODEMCU ESP8266. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 8(1), 106–114.
- Wibowo, I. S., Firdaus, M. A., & Laksana, T. T. (2023). Sistem Monitoring Ruang Server Berbasis Internet of Things Menggunakan Komunikasi Wireless LoRa Ebyte E32. *Jurnal Sistem Cerdas*, 6(3), 222–231.
- Widiyatmoko, A., & Suharjo, I. (2025). Smart Temperature and Humidity Monitoring pada Data center Dinas Komunikasi dan Informatika Daerah Istimewa Yogyakarta Berbasis Raspberry Pi 5 dan Notifikasi Discord. *JURNAL KRIDATAMA SAINS DAN TEKNOLOGI*, 7(02), 613–634.