Available *online* at: http://jarpet.ft.unand.ac.id/

Jurnal Andalas: Rekayasa dan Penerapan Teknologi

ISSN (Online) 2797-9024

Development and Performance Evaluation of a Real-Time Environmental Monitoring System Using ESP32, MQTT, Node-RED, and MySQL Database

Micko Tomas [1], Refki Budiman [1], Baharuddin [1], Riko Nofendra [1], Hanalde Andre [1], Elsi Alfionita Syawal [2]

[1] *Department of Electrical Engineering, Universitas Andalas, Padang, Indonesia*
[2] *Department of Electrical Engineering, Universitas Jambi, Jambi, Indonesia*

ARTICLE INFORMATION

Received: June 24,26
Revised: June 25, 26
Available online: June 26, 26

KEYWORDS

Environmental Monitoring, ESP32, Internet of Things, MQTT, Node-RED

CORRESPONDENCE

Phone:

E-mail: mickotomas@eng.unand.ac.id

A B S T R A C T

Real-time environmental monitoring is an important requirement for managing facilities, laboratories, server rooms, and work environments that require temperature and humidity control. This study aims to develop and evaluate the performance of an Internet of Things (IoT)-based environmental monitoring system using ESP32 microcontrollers, DHT11 sensors, the Message Queuing Telemetry Transport (MQTT) protocol, Node-RED, and a MySQL database. The system is designed to acquire temperature and humidity data, transmit data via Wi-Fi using MQTT, display real-time measurement results on the Node-RED dashboard, and store historical data into a MySQL database. Testing was carried out with a data transmission interval of every 5 seconds using MQTT QoS 0. The test results showed that the system was capable of continuous monitoring, with a measured temperature range of 27.1–28.5 °C and a relative humidity range of 44–51% RH. Sensor data was successfully transmitted, visualized, and stored in real-time without any loss of observed data during the test period. The Node-RED dashboard can display information as gauges, trend graphs, and historical tables, facilitating the monitoring and analysis of environmental data. The integration of ESP32, MQTT, Node-RED, and MySQL produces a stable, low-cost, easy-to-develop monitoring system with potential for implementation in various IoT-based environmental monitoring applications. The results of the study indicate that the proposed architecture can provide a reliable environmental monitoring solution and support real-time, data-driven decision-making.

INTRODUCTION

The rapid advancement of the Internet of Things (IoT) has accelerated the digital transformation of conventional monitoring systems into intelligent, interconnected platforms that collect, transmit, process, and visualize data in real time. The increasing demand for environmental monitoring has emerged across sectors such as industrial facilities, laboratories, data centers, healthcare environments, smart buildings, and agricultural systems. Temperature and humidity are among the most critical environmental parameters because fluctuations beyond acceptable thresholds may adversely affect equipment performance, product quality, operational safety, and energy efficiency [1], [2].Consequently, the development of reliable and cost-effective monitoring systems has become an important research focus in recent years.

IoT technology enables seamless integration between sensing devices, communication networks, data processing platforms, and decision-support systems. The adoption of IoT-based monitoring systems has significantly improved organizations' ability to monitor environmental conditions remotely and continuously while reducing manual intervention [2], [3]. Furthermore, the emergence of cyber-physical systems has strengthened the interaction between physical environments and digital infrastructures, enabling real-time monitoring and intelligent responses to changing environmental conditions. However, ensuring reliable communication, secure data transmission, and scalable system architecture remains a major challenge in IoT deployment [4], [5].

Among various IoT hardware platforms, the ESP32 microcontroller has become one of the most widely adopted solutions due to its integrated Wi-Fi and Bluetooth capabilities, low power consumption, affordability, and sufficient processing performance for real-time applications [6]. These characteristics make ESP32 particularly suitable for environmental monitoring systems that require continuous

https://doi.org/10.25077/jarpet.v6i1.143

sensing and wireless communication. Previous studies have demonstrated the successful implementation of ESP32 in remote monitoring applications, including temperature monitoring systems and smart home environments [7], [8]. Additionally, ESP32-based architectures have shown promising results in supporting scalable IoT deployments through efficient data acquisition and communication mechanisms [9].

The effectiveness of an IoT monitoring system depends heavily on the communication protocol used for data transmission. MQTT (Message Queuing Telemetry Transport) has become one of the most popular messaging protocols for IoT applications due to its lightweight architecture, low bandwidth requirements, and publish–subscribe communication model [10], [11]. Compared with conventional communication protocols, MQTT provides efficient message delivery while minimizing network overhead, making it suitable for resource-constrained embedded devices [7]. Recent environmental monitoring studies have demonstrated that MQTT can support reliable real-time data transmission while enabling seamless integration with cloud and edge computing platforms [12], [13]. Moreover, MQTT has been successfully integrated into various monitoring systems to facilitate data exchange between sensor nodes, servers, and visualization platforms [9].

In addition to communication reliability, effective visualization and data management are essential components of environmental monitoring systems. Node-RED has gained considerable attention as a flow-based development platform that simplifies IoT integration, dashboard creation, and data processing through a graphical programming environment [14], [15]. The flexibility of Node-RED enables rapid development of monitoring dashboards that display environmental parameters in real time while also supporting database integration and remote access. Furthermore, database management systems such as MySQL provide structured storage mechanisms for historical data, enabling trend analysis, anomaly detection, and long-term environmental assessment [15].

Several studies have reported the successful implementation of environmental monitoring systems using combinations of ESP32, MQTT, and Node-RED technologies. De Side et al. developed a microclimate monitoring platform utilizing Node-RED for real-time visualization. At the same time, Utomo and Izzaturrahmani implemented a temperature and humidity monitoring system based on MQTT and Node-RED. Khan et al. further enhanced environmental monitoring capabilities by integrating predictive analytics and real-time alert mechanisms. Other studies have explored ESP32-MQTT architectures in smart homes and remote monitoring applications, demonstrating the feasibility and practicality of such approaches [7]. Despite these advances, many existing implementations primarily emphasize system functionality and deployment. Limited attention has been devoted to evaluating the operational performance of integrated monitoring architectures, particularly in terms of real-time data acquisition, communication reliability, dashboard responsiveness, and continuous database logging.

Therefore, this study proposes the development and performance evaluation of a real-time environmental monitoring system based on ESP32, MQTT, Node-RED, and MySQL database. The proposed system is designed to acquire temperature and humidity data from environmental sensors, transmit the data via MQTT, visualize the data in a web-based dashboard, and store historical records in a relational database. Furthermore, system performance is evaluated based on real-time monitoring capability, communication effectiveness, and data logging reliability. The results of this study are expected to contribute to the development of scalable, low-cost, and reliable environmental monitoring solutions that can support various industrial, educational, and smart-environment applications..

METHOD

System Architecture

This study developed a real-time environmental monitoring system based on the Internet of Things (IoT) to monitor temperature and humidity continuously. The proposed system integrates a DHT11 sensor, an ESP32 microcontroller, an MQTT communication protocol, a Node-RED platform, and a MySQL database. The overall architecture of the proposed system is shown in Figure 1.

As illustrated in Figure 1, the DHT11 sensor functions as a sensing device that measures ambient temperature and relative humidity. The measured data are acquired by the ESP32 microcontroller and transmitted wirelessly through a Wi-Fi network using the MQTT protocol. The Node-RED server acts as a middleware platform that receives, processes, visualizes, and stores incoming sensor data. Real-time monitoring is performed via a dashboard interface with gauges, charts, and historical tables, while all measurements are simultaneously stored in a MySQL database for long-term analysis. Using MQTT enables lightweight data exchange between sensor nodes and the monitoring server. Compared with conventional request-response protocols, MQTT employs a publish-subscribe mechanism, thereby reducing communication overhead and improving scalability. This architecture allows sensor devices to transmit data continuously without requiring direct communication with dashboard clients.

The system consists of an ESP32 microcontroller, a DHT11 temperature and humidity sensor, an MQTT communication protocol, a Node-RED platform, and a MySQL database. The DHT11 sensor continuously measures environmental temperature and humidity, while the ESP32 processes the sensor readings and transmits the data through Wi-Fi using the MQTT publish–subscribe mechanism. The Node-RED server subscribes to the MQTT topic, receives incoming data, visualizes the measurements on a dashboard, and stores

historical records in a MySQL database. The dashboard provides real-time monitoring through gauges, charts, and tabular views, while additional functions include data filtering, CSV export, and database management.
The data flow begins with environmental sensing, followed by data acquisition by ESP32, MQTT-based transmission, Node-RED processing, dashboard visualization, and finally storage in the MySQL database.

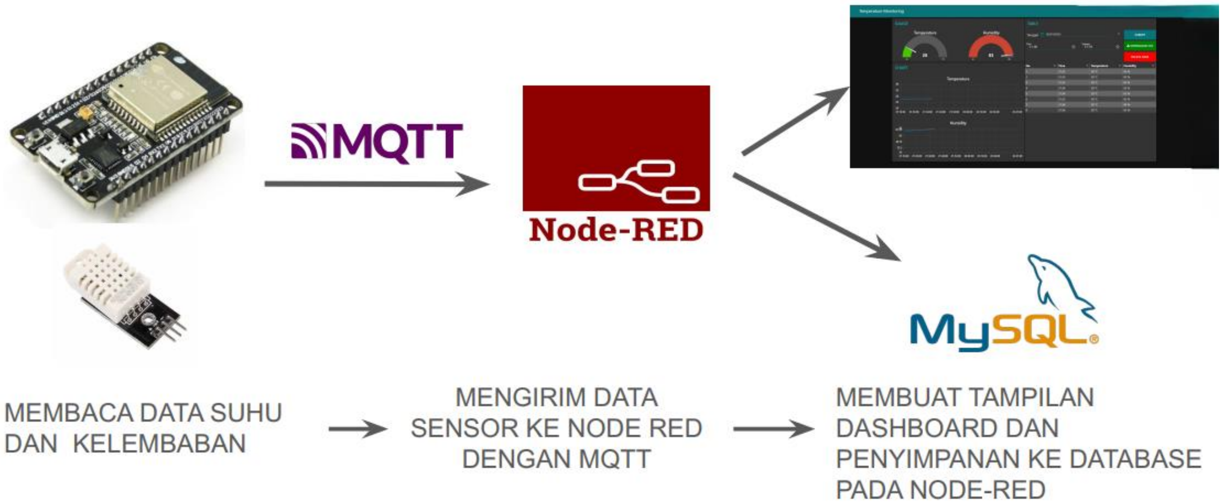

Figure 1. Overall architecture of the proposed environmental monitoring system.

Hardware Configuration

The hardware subsystem consists of an ESP32 development board and a DHT11 sensor. The ESP32 functions as the primary processing and communication unit, while the DHT11 sensor measures ambient temperature and relative humidity. The hardware subsystem consists of an ESP32 development board and a DHT11 temperature-humidity sensor, as depicted in Figure 2.

The ESP32 was selected because it provides integrated Wi-Fi communication, low power consumption, and sufficient processing capability for IoT applications. Meanwhile, the DHT11 sensor was employed due to its simplicity, affordability, and capability to measure both temperature and humidity simultaneously.

The wiring configuration is summarized as follows:
- DHT11 VCC connected to ESP32 3.3 V.
- DHT11 GND connected to ESP32 GND.
- DHT11 DATA connected to GPIO14 (D14) of ESP32.

After initialization, the ESP32 periodically reads temperature and humidity values from the DHT11 sensor and converts them into digital data. These data are then formatted as JavaScript Object Notation (JSON) payloads before being transmitted through MQTT communication.

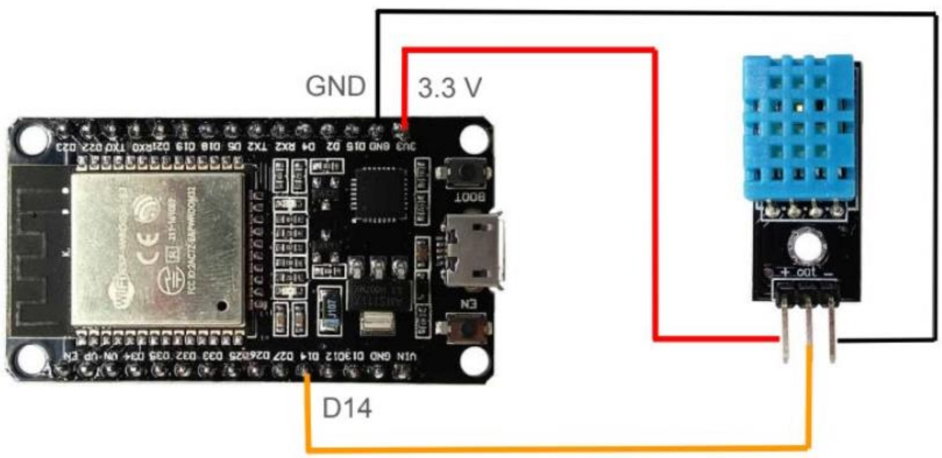

Figure 2. Wiring configuration between ESP32 and DHT11 sensor.

Software Architecture

The software architecture was implemented using Node-RED, which provides a visual programming environment for IoT applications. The complete workflow is illustrated in Figure 3. The workflow consists of four main modules: MQTT data acquisition, dashboard visualization, database storage, and data management. In the MQTT data acquisition module, the MQTT Subscriber node continuously listens to messages published by the ESP32. The received payload contains temperature and humidity measurements in JSON format, which are then parsed and converted into numerical variables for further processing.

After the data are parsed, the measurements are displayed through various dashboard components, including gauge indicators for real-time monitoring, line charts for trend analysis, and historical tables for detailed inspection. This visualization module enables users to observe environmental conditions intuitively without requiring direct access to raw sensor data. In addition, the processed measurements are automatically stored in a MySQL database. Each record includes a unique record ID, timestamp, temperature value, and humidity value. The database storage mechanism supports long-term data retention and facilitates historical analysis of environmental conditions.

To improve system usability, the platform also incorporates several data management features. Users can filter records based on specific dates and times, delete unnecessary records, and export data in CSV format for offline analysis. These functionalities enhance the practicality of the monitoring platform and support efficient data handling for various environmental monitoring applications. Figure 3 presents the complete Node-RED workflow used in the proposed system.

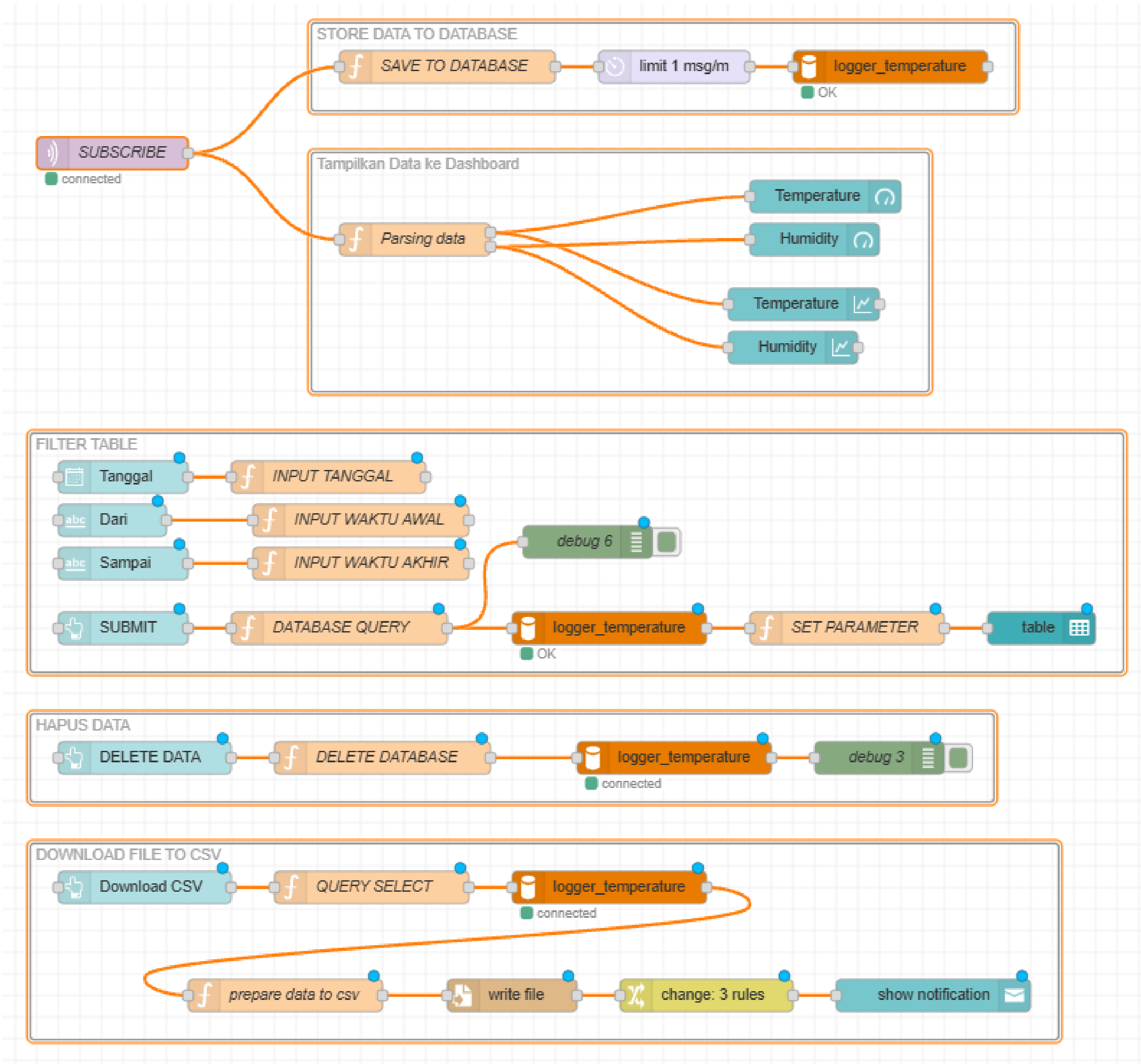

Figure 3. Node-RED workflow used in the proposed system.

 https://doi.org/10.25077/jarpet.v6i1.143

MQTT Communication Mechanism

The MQTT (Message Queuing Telemetry Transport) protocol was utilized as the primary communication mechanism between the ESP32 sensor node and the Node-RED server. MQTT was selected because it is a lightweight publish-subscribe messaging protocol specifically designed for Internet of Things (IoT) applications, where devices often operate with limited processing power, memory, and network bandwidth. Its low overhead and efficient message delivery make it suitable for real-time environmental monitoring systems.

In the proposed architecture, the ESP32 functions as an MQTT client that periodically collects temperature and humidity data from the DHT11 sensor. After each measurement cycle, the ESP32 establishes communication with the MQTT broker and publishes the acquired sensor readings to a predefined topic. The sensor data are transmitted in JavaScript Object Notation (JSON) format to ensure compatibility with Node-RED and other IoT platforms. The payload structure is expressed as:

$$["\text{temperature}","\text{humidity}"] \tag{1}$$

where:
(T) denotes temperature measurement (°C),
(H) denotes relative humidity (%RH).

An example of the transmitted payload is shown below:

$$["\text{temperature}": 29.5,\ "\text{humidity}": 72] \tag{2}$$

On the server side, Node-RED acts as an MQTT subscriber by listening to the same topic. Whenever the ESP32 publishes a new message, Node-RED automatically receives and processes the data without requiring continuous polling. The received payload is then parsed and forwarded to several processing modules, including real-time dashboard visualization, MySQL database storage, data filtering, and CSV export.

The publish-subscribe architecture provided by MQTT offers several advantages for the proposed monitoring system. First, it enables asynchronous communication between devices, allowing the ESP32 and Node-RED to operate independently. Second, it reduces network traffic because data is transmitted only when new measurements are available. Third, it improves system scalability, as multiple sensor nodes and monitoring applications can subscribe to the same broker without significant modifications to the existing infrastructure. Node-RED subscribes to the same topic and processes incoming messages immediately upon reception. This mechanism minimizes communication latency, ensures efficient data transmission, and supports real-time monitoring of environmental conditions.

Performance Evaluation

To evaluate the effectiveness of the proposed monitoring system, several statistical and communication performance metrics were employed to assess both the accuracy of environmental data acquisition and the reliability of data transmission. Statistical evaluation was conducted by analyzing the collected temperature and humidity data using descriptive measures such as minimum, maximum, average, and standard deviation values to determine data consistency and variability over the monitoring period. In addition, communication performance was assessed through MQTT message delivery observations, including transmission success rate, data update frequency, and system responsiveness between the ESP32 device, Node-RED platform, and MySQL database. These metrics provide a comprehensive evaluation of the system's capability to collect, transmit, process, and store environmental data accurately and efficiently in real-time monitoring applications.

Average Temperature

The average temperature represents the central tendency of temperature measurements during the observation period and is calculated using:

$$\bar{T} = \frac{1}{n}\sum_{i=1}^{n} T_i \tag{3}$$

where:
$(\bar{T})$ = average temperature (°C),
(T_i) = temperature measurement at sample (i),
(n) = total number of observations.
This parameter provides an overall representation of environmental thermal conditions.

Average Humidity

The average relative humidity is determined by:

$$\bar{H} = \frac{1}{n}\sum_{i=1}^{n} H_i \tag{4}$$

where:
$(\bar{H})$ = average humidity (%RH),
(H_i) = humidity measurement at sample (i),
(n) = number of observations.
The average humidity indicates the overall moisture level of the monitored environment.

Standart Deviation

Measurement stability was evaluated using standard deviation:

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{5}$$

where:
(SD) = standard deviation,
(x_i) = individual measurement,
$(\bar{x})$ = average measurement value.
A smaller standard deviation indicates that measurements are more stable and less dispersed around the mean.

Packet Delivery Ratio (PDR)

Communication reliability was assessed using Packet Delivery Ratio:

$$PDR = \frac{N_r}{N_s} \times 100\% \tag{6}$$

where:
(N_r) = total packets successfully received,
(N_s) = total packets transmitted.
A PDR value approaching 100% indicates excellent communication reliability.

Average End-to-End Delay

The communication delay was calculated using:

$$D = \frac{\sum_{i=1}^{n}(t_r - t_s)}{n} \tag{7}$$

where:
(D) = average delay (ms),
(t_s) = packet transmission time,
(t_r) = packet reception time.
Lower delay values indicate faster communication between the ESP32 and Node-RED server.

Throughput

The throughput reflects the effective data transmission rate:

 https://doi.org/10.25077/jarpet.v6i1.143

$$Throughput = \frac{\text{Total Data Received (bits)}}{\text{Transmission Time (s)}} \tag{8}$$

where the result is expressed in bits per second (bps) or kilobits per second (kbps). Higher throughput values indicate more efficient network utilization and greater data transfer capability.

RESULTS AND DISCUSSION

System Implementation Results

The proposed environmental monitoring system was successfully developed and implemented using an ESP32 microcontroller, a DHT11 temperature-humidity sensor, the MQTT communication protocol, the Node-RED platform, and a MySQL database. The developed system was designed to acquire environmental measurements, transmit sensor data wirelessly, visualize information in real time, and store historical records for further analysis.

The complete Node-RED-generated dashboard interface is shown in Figure 4. The dashboard includes temperature and humidity gauges, trend charts, historical data tables, date-time filtering, CSV export, and database management features. These components provide users with comprehensive monitoring capabilities through a single web-based interface.

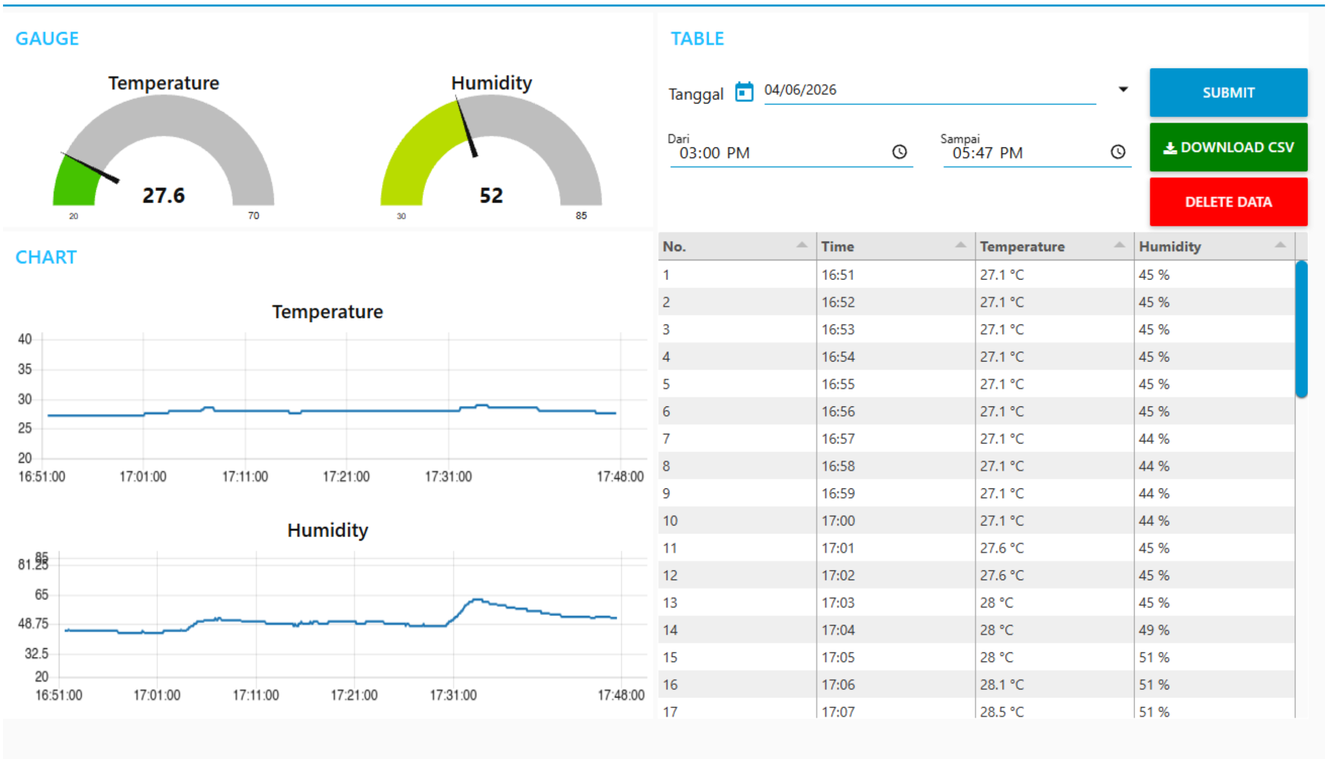

Figure 4. Real-time environmental monitoring dashboard implemented using Node-RED.

As shown in Figure 4, the dashboard displays current temperature and humidity measurements and historical records collected during the monitoring period. The gauge indicators provide immediate information regarding environmental conditions, whereas the trend charts visualize temporal changes in temperature and humidity. Furthermore, the historical table allows users to inspect previously recorded measurements and retrieve data based on selected date and time intervals. The successful implementation of these functionalities demonstrates that the proposed architecture can support continuous environmental monitoring while providing intuitive data visualization and management capabilities.

MQTT Communication Results

The communication performance of the developed system was evaluated by observing MQTT message transmission between the ESP32 sensor node and the Node-RED server. The MQTT monitoring results are presented in Figure 5.

Figure 5. MQTT message transmission and reception monitoring.

Figure 5 shows that the transmitted payload contains both temperature and humidity measurements in JSON format. The MQTT monitor confirms that messages were published and received continuously using QoS level 0. Based on the timestamps shown in Figure 5, the average data transmission interval was approximately 5 seconds. The consistent arrival of MQTT messages indicates stable communication between the ESP32 device and the MQTT broker throughout the observation period. The successful delivery of sensor measurements demonstrates that the MQTT protocol can effectively support lightweight real-time communication for environmental monitoring applications.

Temperature Monitoring Performance

Temperature monitoring constitutes one of the primary functions of the proposed environmental monitoring system because temperature is a critical parameter that influences environmental comfort, equipment operation, and overall indoor conditions. During the experimental evaluation, temperature data were continuously acquired by the DHT11 sensor and transmitted through the MQTT communication protocol to the Node-RED platform for real-time visualization and storage in the MySQL database.

The monitoring process was conducted over a predefined observation period, during which temperature measurements were recorded at regular intervals. Each measurement was automatically processed by the ESP32 microcontroller before being transmitted to the MQTT broker and displayed on the dashboard. This mechanism enabled users to observe temperature variations in real time while simultaneously maintaining a historical record of all collected data.

Table 1. Temperature Monitoring Results

Parameter	Value
Minimum Temperature	27.1 °C
Maximum Temperature	28.5 °C
Current Temperature	27.6 °C
Temperature Range	1.4 °C

To evaluate the performance of the temperature monitoring subsystem, the recorded measurements were analyzed to determine the minimum, maximum, and current temperature values observed during the experiment. These results provide an overview of the thermal conditions within the monitored environment and demonstrate the capability of the proposed system to capture environmental changes accurately and continuously. The summarized temperature measurements collected during the experiment are presented in Table 1. The temperature profile observed during the monitoring period is illustrated in Figure 6.

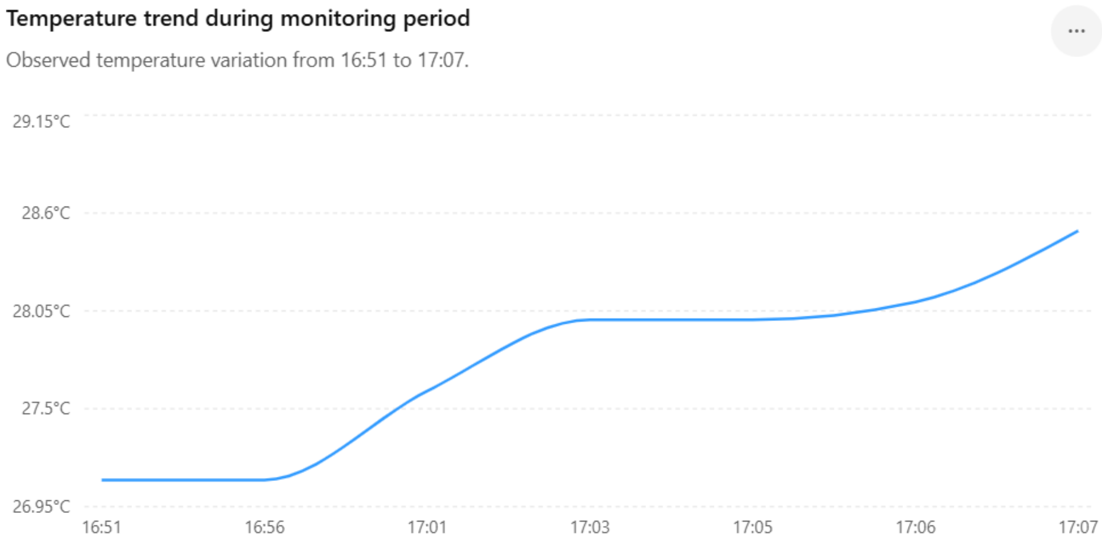

Figure 6. Temperature variation observed during the monitoring period. The temperature gradually increased from 27.1 °C to 28.5 °C, indicating that the proposed system successfully detected environmental changes and updated measurements in real time.

The recorded temperature varied from 27.1 °C to 28.5 °C, resulting in a total variation range of 1.4 °C. The relatively small fluctuation indicates that the monitored environment maintained stable thermal conditions during the experiment. A gradual increase in temperature was observed after approximately 17:00, followed by a peak value of 28.5 °C near the end of the observation period. The absence of abrupt fluctuations suggests that the ESP32-DHT11 sensing subsystem operated consistently and was capable of capturing environmental changes in real time. The temperature trend further confirms that the proposed monitoring system successfully acquired, transmitted, and visualized environmental data without observable interruptions.

Humidity Monitoring Performance

Humidity measurements obtained during the monitoring experiment are summarized in Table 2. The humidity parameter was continuously monitored using the DHT11 sensor and transmitted to the monitoring platform through the MQTT communication protocol. Monitoring humidity is important because variations in relative humidity can significantly affect indoor environmental quality, equipment performance, and human comfort. Throughout the experiment, the developed system successfully captured humidity changes in real time and displayed the measurements on the Node-RED dashboard while simultaneously storing the data in the MySQL database for historical analysis. The collected humidity data provide valuable information regarding environmental moisture conditions and demonstrate the capability of the proposed system to perform reliable and continuous humidity monitoring.

Table 2. Humidity Monitoring Results

Parameter	Value
Minimum Humidity	44 %RH
Maximum Humidity	51 %RH
Current Humidity	52 %RH
Humidity Range	7 %RH

The humidity profile observed during the experiment is presented in Figure 7. The recorded humidity values ranged between 44 %RH and 51 %RH. The humidity remained relatively stable during the initial observation period and increased gradually after approximately 17:03. Compared with temperature measurements, humidity exhibited larger fluctuations because relative humidity is highly sensitive to local environmental conditions and thermal variations. Nevertheless, the measured values remained within a reasonable operating range for indoor environmental monitoring.

The results indicate that the developed monitoring system can effectively capture dynamic changes in environmental moisture levels and display them continuously through the Node-RED dashboard.

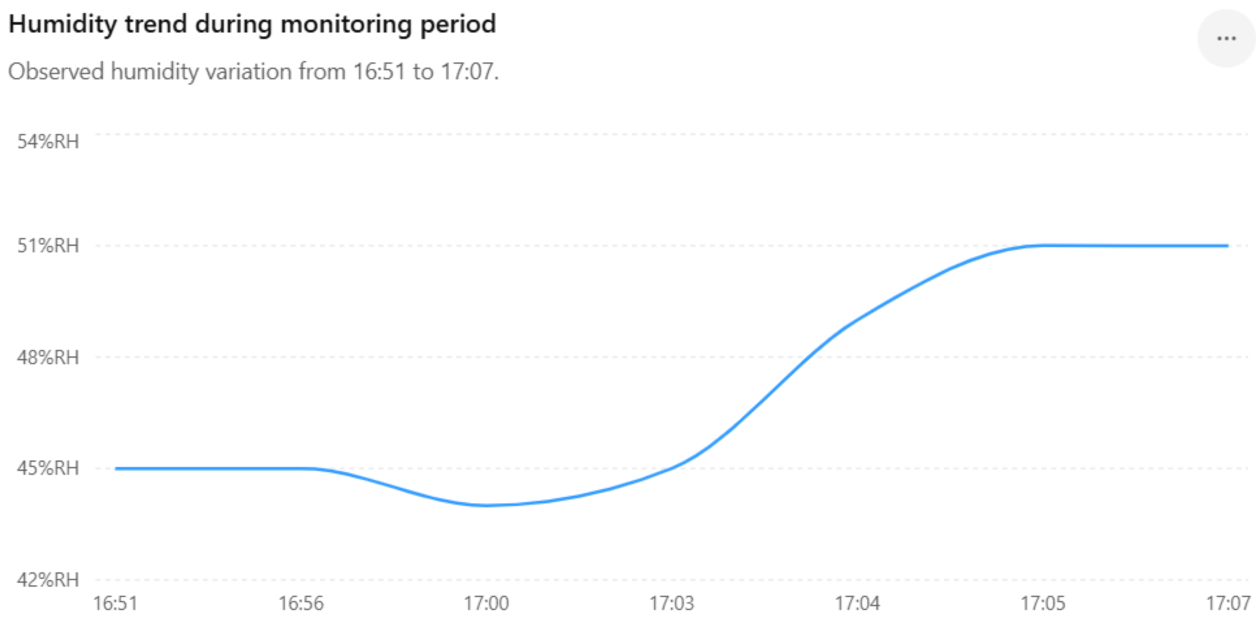

Figure 7. Relative humidity trend during the monitoring period. Humidity increased from 44 %RH to 51 %RH, demonstrating the capability of the proposed monitoring system to capture environmental moisture fluctuations continuously.

Statistical Analysis of Environmental Measurements

To obtain a more comprehensive understanding of the environmental conditions observed during the experiment, a statistical analysis was performed on the collected temperature and humidity data. Statistical evaluation is important because it provides quantitative information regarding the stability, variability, and overall behavior of the monitored environment throughout the data acquisition period. By examining the minimum, maximum, and range values of each parameter, it becomes possible to assess whether significant environmental fluctuations occurred and to evaluate the effectiveness of the proposed monitoring system in capturing these changes accurately.

The statistical summary of the collected measurements is presented in Table 3. This summary consolidates the key characteristics of the recorded data and serves as a basis for comparing the behavior of temperature and humidity during the monitoring period. The analysis also helps identify trends and variations that may not be immediately apparent from individual measurements or real-time dashboard observations.

Table 3. Statistical Summary of Environmental Measurements

Parameter	Minimum	Maximum	Range
Temperature (°C)	27.1	28.5	1.4
Humidity (%RH)	44	51	7

The results indicate that both environmental parameters remained relatively stable throughout the experiment. The temperature exhibited a narrow operating range of only 1.4 °C, while humidity varied within a range of 7 %RH. These observations suggest that the monitored environment experienced no significant disturbances during the data acquisition period. The comparison between minimum and maximum values is illustrated in Figure 8.

Figure 8 compares the minimum and maximum values of temperature and humidity recorded during the monitoring period. The graph shows that the temperature varied from 27.1 °C to 28.5 °C, resulting in a relatively small range of 1.4 °C. This narrow variation indicates that the monitored environment maintained stable thermal conditions throughout the experiment. In contrast, relative humidity showed greater variation, ranging from 44 %RH to 51 %RH, for a total fluctuation of 7 %RH. The larger humidity variation suggests that moisture levels were more sensitive to environmental changes than temperature during the observation period. Despite these fluctuations, both parameters remained within acceptable indoor environmental conditions. Furthermore, the clear distinction between minimum and maximum measurements demonstrates the proposed monitoring system's ability to detect and record environmental variations in real time accurately. The results confirm that integrating the ESP32, DHT11 sensor, MQTT communication protocol, Node-RED dashboard, and MySQL database provides reliable monitoring performance for continuous environmental observation applications.

The statistical comparison shown in Figure 8 also highlights the effectiveness of the data acquisition and transmission process. All observed variations were successfully reflected in the dashboard and database records without missing measurements, indicating stable operation of the sensing, communication, and storage subsystems. The ability to capture both small temperature fluctuations and larger humidity variations demonstrates the sensitivity of the monitoring architecture and its suitability for applications requiring continuous environmental assessment, such as laboratories, server rooms, smart buildings, and educational facilities.

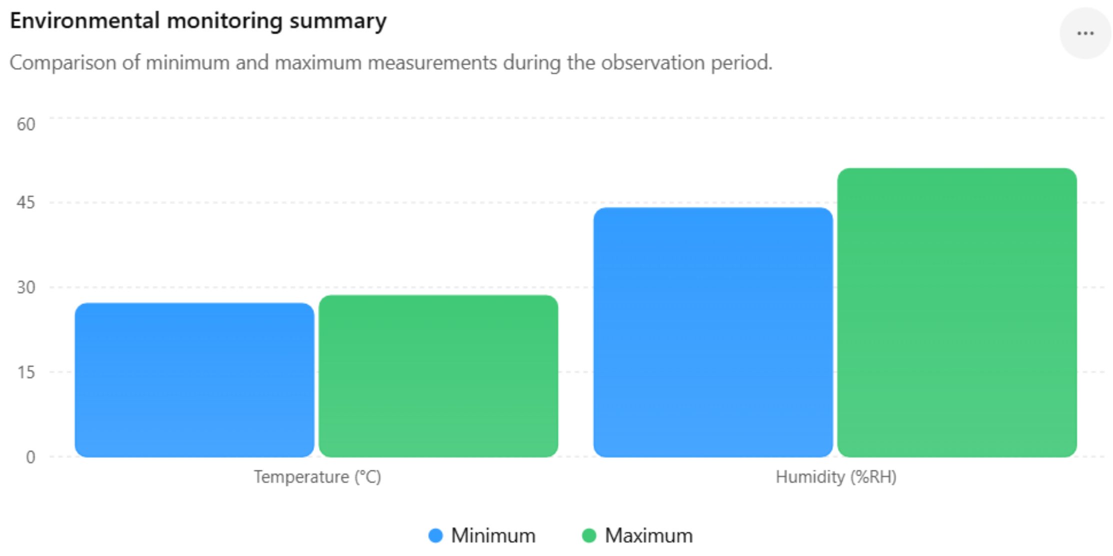

Figure 8. Comparison of minimum and maximum temperature and humidity values during the monitoring period.

Database Logging Performance

One of the primary objectives of the developed monitoring system was to provide reliable data storage for long-term environmental analysis. As shown in Figure 4, all measurements received through MQTT communication were automatically stored in the MySQL database and subsequently displayed in the historical data table.

The database logging mechanism operated continuously throughout the experiment without observable data loss. Each record was stored with its timestamp, temperature, and humidity values. This functionality ensures complete traceability of environmental conditions and supports future trend analysis. In addition, the filtering function enabled users to retrieve measurements based on selected date and time intervals, while the CSV export function facilitated offline analysis and reporting. These capabilities significantly improve the practicality of the proposed monitoring platform.

Overall System Performance Discussion

The experimental results demonstrate that the integration of ESP32, MQTT, Node-RED, and MySQL successfully provides a functional and reliable environmental monitoring platform. Throughout the testing process, the system was able to operate continuously and perform all major monitoring functions without significant interruptions. The developed platform successfully combined sensing, wireless communication, data visualization, and data storage into a unified Internet of Things (IoT) architecture capable of monitoring environmental conditions in real time.

The developed system was capable of performing four essential operations. First, the ESP32 microcontroller, together with the DHT11 sensor, successfully acquired temperature and humidity measurements from the monitored environment. The sensor readings were collected periodically and processed by the ESP32 before being transmitted to the network. The recorded measurements showed stable operation during the observation period, indicating that the sensing subsystem was able to provide consistent environmental data.

Second, the system achieved reliable MQTT-based wireless communication. The MQTT protocol served as the communication bridge between the ESP32 device and the Node-RED platform. During testing, sensor data were transmitted successfully through the MQTT broker with a consistent update interval of approximately 5 seconds. This communication mechanism enabled efficient data exchange while maintaining low network overhead, which is one of the main advantages of MQTT for IoT applications. The successful delivery of messages without noticeable data loss demonstrates the suitability of MQTT for real-time environmental monitoring systems.

Third, the Node-RED dashboard provided effective real-time visualization of environmental conditions. The dashboard displayed temperature and humidity values using gauges, trend charts, and historical data tables, allowing users to observe current conditions as well as changes over time. The graphical representation of sensor readings improved data accessibility and enabled users to quickly identify environmental fluctuations. The dashboard interface also contributed to the usability of the system by presenting information in a clear and intuitive manner.

Fourth, the MySQL database successfully performed continuous historical data storage. Every measurement received through MQTT was automatically recorded in the database, creating a structured repository of environmental information. This storage capability is particularly important for long-term monitoring applications because it enables historical analysis, trend evaluation, performance assessment, and future data processing activities. The successful integration between Node-RED and MySQL ensured that environmental measurements were archived systematically and could be retrieved whenever needed.

The overall performance of the system indicates that the proposed architecture is capable of supporting real-time environmental monitoring requirements. The combination of periodic sensing, reliable wireless communication, responsive visualization, and persistent data storage creates a comprehensive monitoring solution that can be deployed in various environments. In addition, the use of widely available hardware and open-source software components contributes to the affordability and scalability of the system.

Overall, the proposed system offers a low-cost, scalable, and user-friendly solution for environmental monitoring applications. The successful integration of sensing, communication, visualization, and database components demonstrates the feasibility of implementing ESP32-based IoT architectures for real-time environmental monitoring. The developed platform can be applied in laboratories, offices, classrooms, server rooms, smart-building environments, and other facilities that require continuous observation of temperature and humidity conditions. Furthermore, the modular design of the system allows future enhancements, such as the integration of additional sensors, cloud-based analytics, automated alert mechanisms, and remote monitoring capabilities, thereby increasing its potential for broader IoT applications.

CONCLUSIONS

This study successfully developed and evaluated a real-time environmental monitoring system based on ESP32, DHT11, MQTT, Node-RED, and MySQL database technologies. The proposed system was capable of continuously acquiring temperature and humidity measurements, transmitting sensor data wirelessly through the MQTT protocol, visualizing environmental conditions in real time using a Node-RED dashboard, and storing historical records automatically in a MySQL database. Experimental results demonstrated that the system operated reliably throughout the monitoring period, with MQTT communication successfully delivering sensor data at approximately 5-second intervals without observable interruptions. The recorded temperature values ranged from 27.1 °C to 28.5 °C, resulting in a temperature variation of 1.4 °C, while humidity measurements varied between 44 %RH and 51 %RH, corresponding to a humidity range of 7 %RH. These results indicate that the monitored environment remained relatively stable and that the developed system was capable of accurately capturing environmental changes in real time. Furthermore, the Node-RED dashboard effectively displayed environmental information through gauges, trend charts, and historical tables, thereby improving data accessibility and monitoring efficiency. The integration of MySQL database storage ensured complete traceability of environmental measurements and enabled long-term data management for future analysis. Overall, the successful integration of sensing, communication, visualization, and database components confirms that the proposed architecture provides a reliable, scalable, low-cost, and user-friendly solution for environmental monitoring applications. The developed platform demonstrates significant potential for deployment in laboratories, classrooms, offices, server rooms, smart-building environments, and other facilities requiring continuous temperature and humidity monitoring. Future work may focus on integrating additional environmental sensors, implementing cloud-based monitoring services, incorporating automated alert mechanisms, and evaluating communication performance using Quality of Service (QoS), Packet Delivery Ratio (PDR), delay, and throughput metrics to further enhance system capability and reliability.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Electrical Engineering, Universitas Andalas, for providing academic support, research facilities, laboratory resources, and technical assistance throughout the completion of this study. The authors also appreciate the valuable guidance and encouragement from the faculty members, laboratory staff, and colleagues who contributed to the successful development and evaluation of the proposed environmental monitoring system. Their continuous support and constructive feedback have been instrumental in facilitating this research and advancing the implementation of Internet of Things (IoT) technologies in environmental monitoring applications..

REFERENCES

[1] A. H. Ismail, M. A. Hannan, A. Mohamed, and P. J. Ker, "An Internet of Things (IoT) Framework for Smart Monitoring and Energy Management in Buildings," *Sensors*, vol. 21, no. 2, p. 518, 2021, doi: 10.3390/s21020518.

[2] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on Cyber-Physical Systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 27–40, 2020, doi: 10.1109/JAS.2019.1911533.

[3] A. Banks and R. Gupta, "MQTT Version 5.0," Dec. 2019, *OASIS Standard*. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[4] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12918–12940, 2021, doi: 10.1109/JIOT.2021.3055529.

[5] M. A. Jan, F. Khan, M. Alam, and M. Usman, "A Payload-Based Mutual Authentication Scheme for Internet of Things," *Future Generation Computer Systems*, vol. 92, pp. 1028–1039, 2019, doi: 10.1016/j.future.2018.08.057.

[6] "ESP32 Series Datasheet," 2024, *Shanghai, China*.

[7] C. Austin, "Implementation of IoT-Based Remote Temperature Monitoring Using ESP32 and MQTT on AWS," *Journal of Electrical Engineering*, vol. 15, no. 2, pp. 85–93, 2022.

[8] M. K. Zein, R. Hidayat, A. Nugroho, and S. Prasetyo, "Implementation of ESP32-Based Smart Home Integrated with MQTT and Node-RED," *InHardware Journal*, vol. 4, no. 1, pp. 1–10, 2024.

[9] A. Kumar, C. Banapuram, and N. Venu, "Remote-Based Home Automation with MQTT: ESP32 Nodes and Node-RED on Raspberry Pi," in *Proceedings of the 8th International Conference on Internet of Things in Social, Mobile, Analytics and Cloud (I-SMAC)*, Chennai, India, 2024. doi: 10.1109/I-SMAC61858.2024.10714655.

[10] N. Naik, "Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP," in *Proceedings of the IEEE International Systems Engineering Symposium (ISSE)*, Vienna, Austria, 2017, pp. 1–7. doi: 10.1109/SysEng.2017.8088251.

[11] J. S. Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. Aquino, "LoRaWAN in the Internet of Things: A Survey," *Computer Networks*, vol. 201, p. 108434, 2021, doi: 10.1016/j.comnet.2021.108434.

[12] E. B. Utomo and N. Izzaturrahmani, "Temperature and Humidity Monitoring System Using Node-RED Based on MQTT Protocol," *Journal of Mechatronics and Intelligent Systems*, vol. 4, no. 1, pp. 1–10, 2025.

[13] S. Khan, M. Ahmed, A. Rehman, and T. Ali, "Enhanced IoT-Based Environmental Monitoring System Using MQTT with Real-Time Alerts and Predictive Analytics," *IAR Journal of Engineering and Technology*, vol. 6, no. 3, pp. 112–120, 2025.

[14] G. N. De Side, A. S. Abdullah, M. A. Rahman, and A. A. Karim, "IoT Microclimate Monitoring System Using Node-RED Platform in Plant Factory Environment," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1441, p. 12012, 2025.

[15] H. Zidni, M. A. Pratama, A. Nugraha, and R. Setiawan, "Developing a Scalable IoT-Based Platform for Enhancing Air Quality Monitoring Using Node-RED," *Journal of Engineering and Technological Sciences*, vol. 56, no. 2, pp. 145–158, 2024.