International Journal Of Health, Engineering And Technology (IJHET) E-ISSN 2829 - 8683
Volume 5, Number 2, July 2026, Page. 2492 - 2508
Email : editorijhess@gmail.com

Implementation Of An Internet Of Things (IoT)-Based Indoor Air Quality Monitoring System: A Case Study Of PT. Bianglala Metropolitan

Muhammad Dimas Pramudya Anugrah Pratama[1]), Rinna Rachmatika[2])
[1,2)] Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang

*Corresponding Author
E-mail: pramudyapratama1105@gmail.com

Abstract

This research aims to develop an indoor air quality monitoring system using the ESP32-S3 equipped with gas, temperature, and humidity sensors. The introduction discusses the background of indoor air quality issues and the importance of real-time monitoring. The research objective is to design and implement a prototype device capable of efficiently detecting and monitoring air quality parameters. The methodology involves developing a prototype device by integrating gas, temperature, and humidity sensors into the ESP32-S3. Subsequently, testing is conducted to evaluate the prototype's performance in measuring and monitoring air quality indoors. The results indicate that the prototype successfully measures and monitors gas concentration, temperature, and humidity accurately. Monitoring data can be accessed in real-time through applications or web-based platforms. These findings suggest that the development of this system can be an effective solution for indoor air quality monitoring in various enclosed spaces. The conclusion of this research states that the implementation of gas, temperature, and humidity sensors in the development of an indoor air quality monitoring system using the ESP32-S3 has successfully achieved its intended goals. This system can serve as a foundation for further development to enhance monitoring and control of air quality in various indoor environments.

Keywords: ESP32-S3, Indoor Air Quality, Real-Time Monitoring, Gas Sensor, Temperature and Humidity, Internet Of Things (IoT), Environmental Monitoring System

INTRODUCTION

Indoor air quality (IAQ) is one of the important factors affecting the health, comfort, and productivity of building occupants. Along with increasing urbanization, the extensive use of multi-story buildings, and the growing number of indoor human activities, the need for indoor air quality monitoring systems has become increasingly important. Various parameters such as hazardous gas concentrations, temperature, humidity, and airborne particles can affect human health if not properly controlled. The Government of Indonesia, through the Regulation of the Minister of Health of the Republic of Indonesia Number 1077/MENKES/PER/V/2011 concerning Guidelines for Indoor Air Sanitation in Residential Buildings, emphasizes that poor indoor air quality may cause health problems and therefore requires appropriate and sustainable mitigation efforts from all relevant stakeholders (Ministry of Health of the Republic of Indonesia, 2011).

The deterioration of indoor air quality is closely associated with an increased risk of respiratory disorders, particularly Acute Respiratory Infections (ARI). A systematic review conducted by Nasution et al. (2025) indicated that inadequate ventilation, improper humidity levels, and high concentrations of pollutants such as carbon monoxide (CO), sulfur dioxide (SO_2), and nitrogen dioxide (NO_2) significantly contribute to the increasing incidence of ARI among communities living in densely populated areas.

Indoor air quality issues have also been identified in the working environment of PT. Bianglala Metropolitan, a public transportation company operating bus services as one of Transjakarta's partner operators in the Greater Jakarta area (Jabodetabek). The movement of buses entering and leaving operational areas generates exhaust emissions resulting from fuel combustion processes, including carbon monoxide (CO) and carbon dioxide (CO_2). In addition, maintenance and repair workshops for bus fleets produce fuel vapors, combustion smoke, and particulate matter generated from mechanical activities, all of which potentially contribute to indoor air pollution. These conditions are further exacerbated by inadequate air circulation, high employee activity density, and the use of electronic equipment that increases indoor temperature and humidity levels.

The transportation sector itself is one of the major contributors to urban air pollution due to the high emissions produced by fossil fuel-powered vehicles. Fuel combustion processes release various pollutants such as carbon monoxide (CO), carbon dioxide (CO_2), nitrogen oxides (NOx), and hydrocarbons that may degrade environmental air quality. If left uncontrolled, these conditions may cause both short-term and long-term health impacts on workers and surrounding communities. Several studies have even shown that indoor air pollution poses greater health risks than outdoor air pollution because pollutant concentrations tend to be higher indoors and exposure durations are generally longer (A'yun & Umaroh, 2022).

Currently, air quality monitoring activities within PT. Bianglala Metropolitan are still conducted manually through direct physical observation by personnel without support from automated sensor-based measurement systems. Consequently, changes in air quality conditions cannot be detected quickly and accurately, making it difficult to anticipate potential health risks to workers and declines in work productivity at an early stage.

The rapid development of Internet of Things (IoT) technology provides significant opportunities for the development of automatic and real-time air quality monitoring systems. IoT technology enables various sensors to measure environmental parameters such as CO concentration, CO_2 concentration, temperature, and humidity, and subsequently transmit the collected data directly to servers or web-based platforms for monitoring from multiple locations. Hanum and Elfizon (2023) stated that IoT-based air quality monitoring systems are capable of providing environmental information quickly, accurately, and continuously, thereby supporting more effective decision-making processes. Recent studies have also demonstrated that integrating gas sensors with ESP32 microcontrollers and web-based platforms can produce responsive air quality monitoring systems that are easily accessible to users in real time.

Based on these issues, an indoor air quality monitoring system capable of operating automatically and in real time using Internet of Things (IoT) technology is required. Therefore, this study aims to design and implement an IoT-based indoor air quality monitoring system utilizing gas, temperature, and humidity sensors integrated with a website-based monitoring platform. The developed system is expected to contribute to the creation of a healthy, comfortable, and sustainable working environment while supporting faster and more accurate decision-making in maintaining indoor air quality.

RESEARCH METHODS

Research Design

This study employed a Research and Development (R&D) approach using the Prototype development model to design, develop, and implement an Internet of Things (IoT)-based indoor air quality monitoring system. The prototype method was selected because it enables an iterative development process through stages of design, prototype construction, evaluation, and refinement until the system meets user requirements. The prototype model is highly suitable for IoT system development as it can accommodate changes in both hardware and software requirements throughout the development process (Pressman & Maxim, 2020).

Research Location and Period

The study was conducted within the operational environment of PT. Bianglala Metropolitan, a bus-based public transportation company that serves as one of Transjakarta's partner operators in the Greater Jakarta area (Jabodetabek). The research location was selected due to the high intensity of vehicle operations and fleet maintenance activities that potentially generate exhaust emissions affecting indoor air quality. The research was carried out from January to June 2026, covering the stages of observation, system design, implementation, and device testing.

System Development Method

The system development method adopted in this study was the Prototype Model, which consists of four main stages:

Requirement Gathering

This stage aimed to identify both functional and non-functional system requirements through field observations and interviews with relevant stakeholders. The information collected included workplace air quality conditions, the required monitoring parameters, and the requirements for a website-based monitoring system.

System Design

The design stage involved both hardware and software design processes. Hardware design included determining the configuration of sensors, microcontrollers, communication media, and output devices. Meanwhile, software design involved developing sensor reading algorithms, data communication mechanisms, data storage systems, and the user interface design for the web-based monitoring platform.

Prototype Development

At this stage, the design was implemented into a prototype air quality monitoring system utilizing an ESP32-S3 microcontroller, an MQ-135 gas sensor, a DHT22 temperature and humidity sensor, a buzzer as an alert mechanism, and an LCD for local display purposes. Sensor data were transmitted to the server via the internet and displayed on a web-based monitoring dashboard in real time.

Evaluation and Refinement

The evaluation stage was conducted to identify system deficiencies from both hardware and software perspectives. If discrepancies between system performance and user requirements were identified, improvements and refinements were carried out until the final prototype fulfilled operational needs.

Data Collection Techniques

The data collection techniques employed in this study included observation, interviews, literature review, and direct measurements. Observation was conducted directly within the operational environment of PT. Bianglala Metropolitan to identify air quality conditions, operational activity patterns, and potential sources of indoor air pollution. In addition, observations were performed to evaluate the system's response to environmental changes during testing. Interviews were conducted with operational personnel and system users to obtain information regarding air quality monitoring requirements and evaluations of the developed system's performance. A literature review was carried out by collecting references from scientific journals, books, air quality standards, government regulations, and previous studies related to IoT-based air quality monitoring systems. Measurements were performed on air quality parameters, including gas concentration, temperature, and humidity, using the sensors installed within the system. Furthermore, testing was conducted to evaluate the system's performance and reliability under various operational conditions.

Research Instruments and Equipment

The research utilized both hardware and software components as follows:

Hardware Components

1. ESP32-S3 microcontroller
2. MQ-135 gas sensor
3. DHT22 temperature and humidity sensor
4. 20×4 I2C LCD display
5. 3V active buzzer
6. Hi-Link HLK-5M05 power supply module
7. Perfboard and jumper cables

Software Components
1. Arduino IDE for ESP32-S3 programming
2. PostgreSQL database management system
3. NestJS-based backend server
4. Next.js-based monitoring frontend application
5. NGINX as a web server and reverse proxy

System Testing Methods

System testing was conducted to ensure that all components operated according to the designed specifications. The testing methods included:

Black Box Testing

Black box testing was performed by evaluating the main functionalities of the system without considering the internal structure of the program. Testing covered sensor readings, data transmission, website display functionality, LCD output, and buzzer activation to ensure that system outputs matched expected results.

White Box Testing

White box testing was conducted on the program logic and algorithms implemented within the system, including air quality classification functions, sensor data processing procedures, and mechanisms for transmitting data to the server.

User Acceptance Testing (UAT)

User Acceptance Testing was conducted with a number of end users to evaluate the system's usability, reliability, and usefulness in supporting air quality monitoring activities within the workplace environment.

RESULTS AND DISCUSSION

Implementation

The implementation stage represents the realization of the analysis and system design results into an Internet of Things (IoT)-based air quality monitoring system capable of operating in the actual working environment of PT. Bianglala Metropolitan. At this stage, hardware and software components were integrated to enable the system to automatically measure air quality parameters, transmit data through the internet network, and display monitoring results in real time through a web-based dashboard. The system was implemented using an ESP32-S3 microcontroller as the central processing unit connected to an MQ-135 gas sensor, a DHT22 temperature and humidity sensor, a 20×4 LCD display, an active buzzer, MQTT communication protocol, and a website-based monitoring dashboard. All components were integrated into a unified system capable of supporting continuous air quality monitoring.

Hardware Implementation

The hardware implementation process involved assembling all electronic components according to the system design developed in the previous stage. This implementation aimed to produce an air quality monitoring device that is stable, safe, and ready for use in operational environments. The hardware components used in this study are presented in Table 1.

Table 1. Hardware Components List

Component	Component Model	Quantity
Microcontroller	ESP32-S3 DevKit	1
Gas Sensor	MQ-135	1
Temperature and Humidity Sensor	DHT22	1
LCD Display	LCD 20×4 (2004) 5V I2C	1
Buzzer	3V Active Buzzer	1
Power Supply	Hi-Link HLK-5M05 AC-DC Module	1
Assembly Medium	7×9 cm Perfboard	1

International Journal Of Health, Engineering And Technology (IJHET) E-ISSN 2829 - 8683
Volume 5, Number 2, July 2026, Page. 2492 - 2508
Email : editorijhess@gmail.com

Cables	Jumper wires and USB Type-C cable	As required

Based on Table 1, the system consists of a single microcontroller unit serving as the control center, two primary sensors for measuring air quality parameters, and supporting devices including an LCD display and buzzer for user interaction. In addition, a power supply module and permanent assembly medium were utilized to ensure stable long-term system operation.

Component Assembly

The assembly stage was conducted to connect all components into a complete electronic circuit ready for operation. The assembly process was carried out systematically while considering the technical specifications of each component as well as electrical safety aspects. The assembly procedures included:

Component Preparation

The initial stage involved ensuring that all components were available and functioning properly, including the ESP32-S3 DevKit, MQ-135 sensor, DHT22 sensor, 20×4 I2C LCD display, 3V active buzzer, Hi-Link HLK-5M05 module, perfboard, jumper wires, and device enclosure.

MQ-135 Gas Sensor Connection

The MQ-135 sensor was connected to the analog input pin of the ESP32-S3 to measure changes in gas concentration expressed in parts per million (ppm). In addition, the VCC and GND pins were connected to a 5V power supply to ensure optimal sensor performance.

DHT22 Sensor Connection

The DHT22 sensor was connected to a digital pin of the ESP32-S3 using a single-wire communication protocol to periodically transmit temperature and humidity data.

I2C-Based 20×4 LCD Connection

The LCD display was connected through the I2C communication interface using the SDA and SCL pins, thereby improving microcontroller pin efficiency compared with conventional parallel connections.

Active Buzzer Connection

The 3V active buzzer was connected to one of the ESP32-S3 digital output pins, enabling automatic activation whenever gas concentrations exceeded the predefined threshold values.

Ground (GND) Parallelization

All ground lines from individual components were connected to a common ground reference provided by the Hi-Link HLK-5M05 module to prevent potential differences that could result in unstable sensor readings.

Perfboard Assembly

All components were mounted and soldered onto a 7×9 cm perfboard to produce a permanent circuit that is neater, more robust, and easier to maintain.

Connection Path Inspection

After the soldering process was completed, all electrical connections were inspected using a multimeter to ensure that there were no short circuits or open circuits within the system.

Initial Functional Testing

The final stage involved powering on the system and verifying that the sensors, LCD display, buzzer, and network communication operated according to the design specifications.

The implementation results demonstrated that all components were successfully integrated and operated normally as a unified IoT-based air quality monitoring system.

Component Arrangement within the Device Enclosure

After the assembly process was completed, all components were placed inside a device enclosure to protect the circuitry from dust, physical impacts, and other environmental disturbances. In addition to serving as protection, the enclosure also improves installation neatness and facilitates system maintenance. The placement of each component was carefully designed by considering accessibility, heat dissipation, and sensor measurement quality.

1. The ESP32-S3 DevKit was positioned in the upper-right section of the perfboard to serve as the system control center and facilitate firmware reprogramming when required.
2. The MQ-135 gas sensor was installed at the upper part of the enclosure with an open ventilation area to ensure direct exposure to ambient air.
3. The DHT22 temperature and humidity sensor was positioned near the air ventilation openings to obtain temperature and humidity measurements that accurately represent indoor environmental conditions.
4. The 20×4 LCD display was mounted on the front side of the enclosure to allow users to directly observe air quality information.
5. The active buzzer was installed inside the enclosure while ensuring that warning sounds remained clearly audible during hazardous conditions.
6. The Hi-Link HLK-5M05 power supply module was placed at the lower section of the assembly to separate high-voltage and low-voltage circuits, thereby improving electrical safety.
7. Pin headers were installed on the left side of the circuit to simplify sensor removal and replacement during maintenance procedures.
8. The 7×9 cm perfboard was utilized as the permanent mounting medium for all electronic components.

In addition, cable management was systematically implemented by separating sensor lines, communication lines, and power supply lines to minimize signal interference and improve system stability. The implementation results demonstrated that structured component placement significantly improved maintenance accessibility and maintained stable device operation during long-term use.

Software Implementation

In addition to hardware implementation, this study also developed software components that function as an interface between IoT devices and end users through a web-based monitoring system. The software implementation included the development of ESP32-S3 firmware, backend services, MQTT communication infrastructure, and a web-based monitoring dashboard.

The software platforms utilized in this study are presented in Table 2.

Table 2. Software Components

Platform	Software Type	Functional Description
PlatformIO (VS Code)	Programming IDE	ESP32-S3 firmware development, compilation, and uploading
HiveMQ Cloud	Cloud MQTT Broker	MQTT broker for telemetry transmission and remote commands
NestJS	Backend Framework	Sensor data validation and storage
Next.js	Frontend Framework	Web-based monitoring dashboard

Based on Table 2, the software architecture was developed using a modular approach in which each component has a clearly defined responsibility and can be independently expanded or modified in future developments.

Web Interface Implementation

The web interface was developed as the primary medium for users to monitor indoor air quality conditions in real time. Sensor data collected by the ESP32-S3 were transmitted to the HiveMQ Cloud MQTT broker and subsequently processed by the backend server before being displayed on the web-based dashboard.

Backend Implementation

The backend was developed using the NestJS framework, which is responsible for receiving telemetry data from the MQTT broker, validating incoming data, storing sensor measurements in a PostgreSQL database, and providing API services for frontend applications. This approach enables the separation of business logic from the user interface, thereby improving maintainability and system scalability. The backend implementation also supports authentication processes, data validation mechanisms, and automatic historical data recording. An example of the backend log interface receiving telemetry data from the IoT device is shown in **Figure 1**.

```
[TelemetryService] Connected to MQTT Broker
[TelemetryService] Subscribed to topic: telemetry/data
[TelemetryService] Received payload: {"chipId":"ESP32-
001","temperatureC":27.5,"humidity":62.3,"mq135Ppm":430}
[TelemetryService] Data saved to PostgreSQL
```

Figure 1. NestJS Backend Server Log Receiving Telemetry Data from IoT Devices

Frontend Implementation

The frontend application was developed using the Next.js framework to provide a responsive and interactive monitoring dashboard. The dashboard presents air quality parameters in the form of status indicators, data tables, and real-time graphs representing environmental condition changes. Users can monitor gas concentration levels, temperature, humidity, and alarm status directly through a web browser without requiring physical access to the device. This approach increases system flexibility and usability across multiple devices, including desktop computers and smartphones. An example of the monitoring dashboard interface is presented in **Figure 2**.

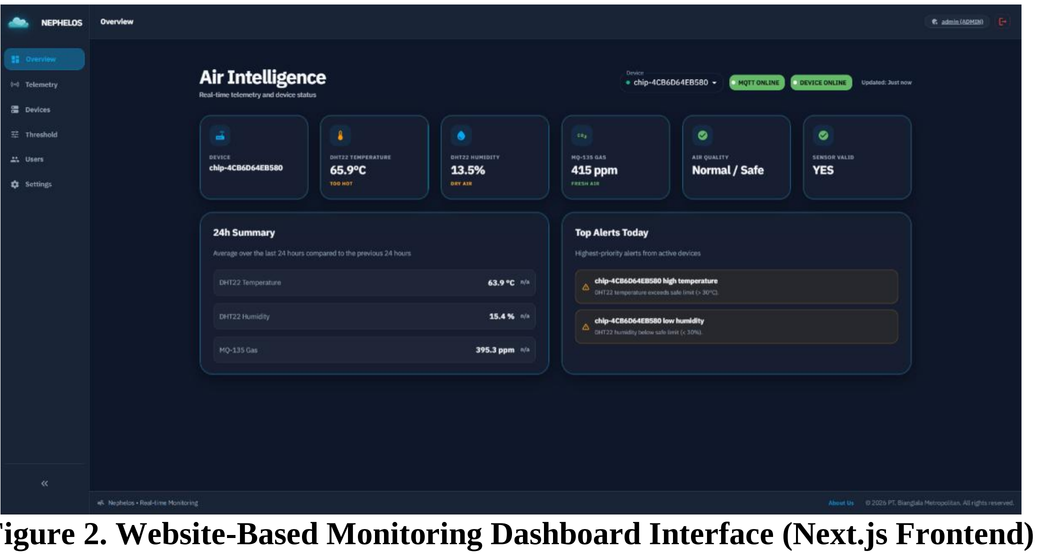

Figure 2. Website-Based Monitoring Dashboard Interface (Next.js Frontend)

Communication integration using the MQTT protocol through HiveMQ Cloud enables fast, lightweight, and efficient data exchange between IoT devices and the server infrastructure. The ESP32-S3 periodically publishes sensor data, while the backend server subscribes to the corresponding MQTT topics to receive and process incoming telemetry data before forwarding them to the monitoring dashboard in real time. Through this mechanism, the developed system is capable of providing stable, responsive, and remotely accessible indoor air quality monitoring services from various locations via the internet.

MQ-135 Gas Sensor Testing Results

Testing of the MQ-135 sensor was conducted to verify its capability to detect changes in air pollutant concentrations and to ensure that the early warning system operated according to predefined

threshold values. The tests were performed under clean air conditions as the normal operating scenario and under polluted air conditions using lighter smoke as a simulation of high-concentration gaseous pollutants.
The results of the MQ-135 sensor testing are presented in Table 3.

Table 3. MQ-135 Gas Sensor Testing Results

No.	Testing Scenario	Testing Procedure	Expected Result	Test Result	Status
1	Clean Air Measurement	Exposing the sensor to clean indoor air	Gas concentration remains below the alarm threshold	435 ppm with the status "Very Clean"	Successful
2	High Gas Pollution Measurement	Exposing the sensor to lighter smoke	PPM value exceeds the alarm threshold and activates the buzzer	1001 ppm and buzzer activated	Successful
3	Rapid Spike Filtering (Step Limits Filter)	Applying sudden exposure to dense gas pollutants	PPM increase is filtered gradually	PPM values increased smoothly and steadily	Successful
4	Automatic R0 Resistance Calibration	Powering on the device under clean air conditions	The system automatically calculates the R0 value	Calibration completed successfully and R0 stored	Successful

Based on the testing results presented in Table 3, the MQ-135 sensor successfully detected changes in gas concentration across all testing scenarios. Under clean air conditions, the sensor measured a concentration of 435 ppm, which was classified as **Very Clean**, and consequently the warning alarm remained inactive. Conversely, when the sensor was exposed to lighter smoke as a simulation of gaseous pollutants, the concentration value increased to 1001 ppm, exceeding the predefined alarm threshold. Under this condition, the system automatically activated the buzzer as an early warning mechanism to inform users of deteriorating air quality.

Testing of the step limits filtering mechanism demonstrated that the filtering algorithm successfully prevented excessively sharp measurement spikes caused by rapid environmental changes. This mechanism is important for reducing the occurrence of false alarms resulting from temporary fluctuations in sensor readings. Furthermore, the automatic calibration process for the baseline resistance value (R0) was successfully performed during the initial startup under clean air conditions. The success of this process is essential because the R0 value serves as the basis for gas concentration calculations in the MQ-135 sensor and therefore directly affects measurement accuracy. Overall, the testing results indicate that the MQ-135 sensor performed effectively in detecting air quality changes and successfully supported the early warning mechanism within the operational environment of PT. Bianglala Metropolitan.

Black Box Testing

Black box testing was conducted to verify the functionality of the system based on the relationship between inputs and outputs without considering the internal structure of the program code. The objective of this testing was to ensure that each system component responded appropriately under various operational conditions that may occur during real-world deployment.

In this study, testing focused on validating sensor measurements, network communication, notification mechanisms, and information display functions provided to users. Each component was tested independently to facilitate error identification and comprehensive system performance evaluation. The tested components included the DHT22 temperature and humidity sensor, the MQ-135 gas sensor, the 20×4 LCD display, MQTT communication services, and the multitasking and failover mechanisms implemented in the ESP32-S3-based system.

DHT22 Temperature and Humidity Sensor Testing Results

Testing of the DHT22 sensor was conducted to evaluate its capability to measure temperature and humidity accurately, responsively, and consistently under various environmental conditions. The tests were performed under air-conditioned room conditions as the normal operating scenario and under rapid environmental changes using warm breath exposure to evaluate sensor sensitivity. The results of the DHT22 sensor testing are presented in Table 4.

Table 4. DHT22 Temperature and Humidity Sensor Testing Results

No.	Testing Scenario	Testing Procedure	Expected Result	Test Result	Status
1	Room Temperature and Humidity Measurement	Exposing the sensor to an air-conditioned room	Stable temperature and humidity measurements	Temperature: 24.5°C, Humidity: 55%	Successful
2	Temperature and Humidity Change Response	Blowing warm breath near the sensor	Temperature and humidity values increase dynamically	Measurements increased rapidly and adaptively	Successful
3	Sensor Error Detection	Disconnecting the GPIO4 signal cable	LCD displays the message "SENSOR_ERROR"	LCD displayed "SENSOR_ERROR"	Successful
4	Long-Term Measurement Stability	Operating the sensor continuously for two hours	Measurements remain stable without extreme fluctuations	Measurement fluctuations < 0.2°C	Successful

Based on the testing results shown in Table 4, the DHT22 sensor successfully measured temperature and humidity under both normal environmental conditions and rapid environmental changes. The recorded temperature of 24.5°C and humidity level of 55% under air-conditioned room conditions indicate that the sensor provided stable measurements that accurately reflected actual environmental conditions. The temperature response test demonstrated that the sensor possesses good sensitivity to environmental changes, as indicated by the rapid increase in temperature and humidity values when exposed to warm air. Furthermore, the sensor error detection mechanism functioned properly when the sensor communication line was disconnected, whereby the system automatically displayed an error message and invalidated the corresponding sensor data to prevent the use of erroneous measurements. The long-term stability test further demonstrated that the sensor maintained highly consistent measurements with fluctuations of less than 0.2°C during two hours of continuous operation. These results indicate that the DHT22 sensor is suitable for continuous indoor air quality monitoring applications.

Integrated Performance and Failover Testing Results

Integrated performance and failover testing were conducted to evaluate the overall system stability when all hardware and software components operated simultaneously within a FreeRTOS-based multitasking environment. In addition, this testing aimed to ensure that the system possessed automatic recovery capabilities in the event of operational disturbances.

The testing results are presented in Table 5.

Table 5. Integrated Performance and Failover Testing Results

No.	Testing Scenario	Testing Procedure	Expected Result	Test Result	Status
1	Multitasking Integration Testing	Running sensor acquisition and MQTT communication simultaneously	Both tasks execute without blocking each other	Tasks executed in parallel successfully	Successful
2	Alarm Response Speed Testing	Introducing hazardous gas suddenly	Alarm activates within less than one second	Alarm activated almost instantly	Successful
3	Task Watchdog Timer Testing	Simulating system deadlock conditions	Watchdog automatically resets the system	System recovered automatically	Successful
4	Mutex Data Protection Testing	Accessing sensor data simultaneously from two processor cores	Memory synchronization remains safe	Data remained consistent without corruption	Successful
5	Power Recovery (Power Cycle) Testing	Disconnecting and reconnecting the power supply	System resumes normal operation	System recovered in less than five seconds	Successful

Based on the results presented in Table 5, the system demonstrated a high level of stability while executing multiple processes in parallel. The implementation of FreeRTOS enabled sensor acquisition, MQTT telemetry transmission, LCD updates, and buzzer control to operate concurrently without causing resource conflicts or execution failures. The alarm response test showed that the system was capable of detecting hazardous gas concentration increases and activating the buzzer within less than one second after the threshold value had been exceeded. This response time indicates that the system is suitable for use as an early warning mechanism against potential air quality hazards.

The Watchdog Timer mechanism also successfully recovered the system automatically when deadlock or task freeze conditions were simulated, thereby improving long-term system reliability. Furthermore, the use of mutex synchronization for sensor data access proved effective in maintaining data consistency during simultaneous access from the dual-core architecture of the ESP32-S3 processor. The final test demonstrated that the system was capable of automatically recovering after power outages or power supply interruptions. The device resumed normal operation in less than five seconds without requiring reconfiguration or user intervention. These results indicate that the system possesses robust failover capabilities and is suitable for deployment in operational environments requiring high system availability.

Network Connectivity and MQTT Telemetry Testing Results

Network and telemetry testing were performed to evaluate the capability of the ESP32-S3 microcontroller to maintain wireless network connectivity, periodically transmit telemetry data to the HiveMQ Cloud service, and receive remote control commands in real time using the MQTT protocol. The testing results are presented in Table 6.

Table 6. Network Connectivity and MQTT Telemetry Testing Results

No.	Testing Scenario	Testing Procedure	Expected Result	Test Result	Status
1	WiFi Connectivity Testing	Activating the configured access point	ESP32-S3 connects and obtains an IP address	WiFi connection established successfully	Successful
2	Telemetry Transmission (JSON Publish)	Monitoring data traffic on HiveMQ Cloud broker	JSON payload transmitted every 10 seconds	Data received by broker according to schedule	Successful
3	Remote Command Processing (Subscribe)	Sending the buzzer_on command through MQTT broker	Buzzer activates and returns acknowledgment	Buzzer activated and ACK received	Successful
4	Automatic Network Reconnection	Temporarily disconnecting and reconnecting WiFi	System reconnects automatically	Network connection restored automatically	Successful
5	Offline Telemetry Queue Transmission	Disconnecting internet connectivity during data acquisition	Data temporarily stored and transmitted after reconnection	Eight telemetry records successfully retransmitted	Successful
6	Remote Reboot Verification	Sending a reboot command via MQTT	System restarts automatically	Device rebooted and returned online	Successful

Based on the results presented in Table 6, all network communication functions operated according to the system design specifications. The ESP32-S3 successfully connected to the WiFi network and automatically obtained an IP address, allowing communication with the server without additional configuration. Telemetry testing demonstrated that the system was capable of transmitting sensor data in JSON format periodically every 10 seconds to the HiveMQ Cloud MQTT broker. This confirms that the publish-subscribe mechanism implemented by the MQTT protocol functioned properly and supported real-time air quality monitoring. Furthermore, the system's capability to receive remote commands such as buzzer activation and device reboot indicates that the platform functions not only as a monitoring tool but also as a remote device management system. Automatic reconnection testing further demonstrated that the system could detect network disconnections and re-establish connectivity without user intervention. The temporary data storage mechanism (offline queue buffering) was also successfully implemented, whereby telemetry data that failed to be transmitted due to network disruptions were temporarily stored in memory and retransmitted once internet connectivity became available. This mechanism significantly improves system reliability and ensures monitoring data continuity.

LCD 20×4 Display Testing Results

Testing of the 20×4 (2004) 5V Liquid Crystal Display (LCD) was conducted to ensure that the system was capable of displaying air quality information locally in a stable, readable, and user-friendly manner. Furthermore, the testing aimed to verify the system's capability to handle real-time data updates without introducing visual disturbances such as display flickering and to evaluate system resilience against communication failures on the I2C bus utilized by the LCD module. The testing results are presented in Table 7.

Table 7. LCD 20×4 (2004) 5V Display Testing Results

No.	Testing Scenario	Testing Procedure	Expected Result	Test Result	Status
1	Booting and Initial Display Testing	Powering on the device	Display shows "Booting..." during sensor initialization	Display behavior matched expectations	Successful
2	Dynamic Screen Update Testing	Modifying sensor data dynamically	Parameters update smoothly without flickering	Display updated without flicker	Successful
3	Special Character Testing	Displaying temperature and percentage symbols	Degree symbol (°C) and percentage symbol (%) displayed correctly	Characters displayed accurately	Successful
4	String Padding and Column Handling	Changing air quality status from long text to shorter text	Remaining characters are cleared completely	Previous characters removed successfully	Successful
5	I2C Bus Communication Failure Testing	Disconnecting LCD I2C power supply	Program continues operating without hanging or freezing	System continued operating normally	Successful

Based on the results shown in Table 7, all testing scenarios were successfully executed according to the system specifications. During the booting process, the LCD successfully displayed initialization messages, providing users with information regarding the readiness status of the system. Periodic sensor data updates were performed without causing display flickering, thereby improving user comfort when reading information from the device. Testing involving special characters also demonstrated that the LCD correctly displayed temperature and humidity symbols, making the presented information more informative and easier to interpret. In addition, the padding mechanism using blank spaces successfully removed residual characters when air quality status messages changed from longer text strings to shorter ones. The most significant result was obtained during the I2C communication failure test, where the system continued operating normally despite LCD communication disruptions or disconnections. These findings indicate that the software architecture implemented effective error handling mechanisms, ensuring that failures in the display subsystem did not cause the main system to hang or freeze.

White Box Testing

White box testing in this study focused on evaluating the internal logic structure of the firmware running on the ESP32-S3 microcontroller to ensure that all execution paths, decision branches, and data synchronization mechanisms functioned according to the system design specifications. Unlike black box testing, which evaluates system outputs based solely on provided inputs, white box testing analyzes the internal program structure directly, enabling all possible execution paths to be comprehensively examined.

The testing method employed in this study was Basis Path Testing, a technique that utilizes Cyclomatic Complexity values to determine the minimum number of independent paths that must be tested to achieve complete logic coverage. This approach was applied to three critical system functions, namely the air quality classification function classifyAirStatus(ppm), the serial command processing function handleSerialCommand(cmd), and the air quality synchronization and buzzer control function updateAirStatusAndBuzzer(data). Through this testing approach, all potential operational conditions of the system were validated to improve the reliability and stability of the Internet of Things (IoT)-based air quality monitoring system.

White Box Testing of the classifyAirStatus(ppm) Function

The first white box test was performed on the classifyAirStatus(ppm) function, which is responsible for classifying gas concentration values measured by the MQ-135 sensor into predefined air quality categories. This function is a critical component of the system because the classification results determine both the air quality status displayed on the monitoring dashboard and the activation of the warning buzzer. Control flow graph analysis revealed that the classifyAirStatus(ppm) function contains four decision nodes, resulting in a cyclomatic complexity value of **5**. Therefore, five independent execution paths were required to ensure complete logic coverage.

Table 8. Basis Path Test Cases for classifyAirStatus()

TC ID	Input (ppm)	Test Path	Expected Output	Status
TC-01	350	Path 1	Very Clean	Pass
TC-02	600	Path 2	Normal / Safe	Pass
TC-03	1000	Path 3	Slightly Polluted	Pass
TC-04	1500	Path 4	Polluted / Hazardous	Pass
TC-05	2500	Path 5	Extremely Hazardous	Pass

As shown in Table 8, all five independent paths were successfully executed and produced outputs consistent with the air quality categories defined during the system design phase. Gas concentrations of 350 ppm and 600 ppm were correctly classified as Very Clean and Normal/Safe, respectively. Concentrations of 1000 ppm, 1500 ppm, and 2500 ppm were accurately identified as Slightly Polluted, Polluted/Hazardous, and Extremely Hazardous. The successful execution of all test cases demonstrates that the classifyAirStatus(ppm) function correctly implemented all branching logic without classification errors across the tested range of values. Consequently, the function provides a reliable basis for decision-making regarding air quality status and early warning mechanisms. The testing results also confirmed that the threshold values implemented in the system operated consistently across all air quality categories.

White Box Testing of the handleSerialCommand(cmd) Function

White box testing of the handleSerialCommand(cmd) function was conducted to ensure that all logic paths involved in serial command processing executed correctly according to system specifications. This function is responsible for parsing local commands transmitted through the USB serial interface to control the buzzer in either manual or automatic mode. Control flow graph analysis indicated that the function contains six primary decision nodes, resulting in a cyclomatic complexity value of **7**, which requires seven independent paths to achieve complete testing coverage.

Table 9. Basis Path Test Cases for handleSerialCommand()

TC ID	Input	Test Path	Expected Output	Status
TC-06	Empty input	Path 1	No action	Pass
TC-07	"help"	Path 2	Display help information	Pass
TC-08	"ON"	Path 3	Buzzer ON (manual mode)	Pass
TC-09	"off"	Path 4	Buzzer OFF (manual mode)	Pass
TC-10	"beep"	Path 5	Buzzer BEEP (manual mode)	Pass
TC-11	"auto"	Path 6	Automatic buzzer mode enabled	Pass
TC-12	"restart"	Path 7	Unknown command response	Pass

The results indicate that all seven independent paths were successfully executed and produced outputs according to system specifications. The function correctly handled empty inputs, displayed help information, activated the requested buzzer mode, and safely handled invalid commands without causing system failures. These findings demonstrate that the handleSerialCommand(cmd) function possesses a high degree of reliability in handling all potential user inputs through serial communication and can support local device control safely and effectively.

International Journal Of Health, Engineering And Technology (IJHET) *E-ISSN 2829 - 8683*
Volume 5, Number 2, July 2026, *Page. 2492 - 2508*
Email : editorijhess@gmail.com

White Box Testing of the updateAirStatusAndBuzzer(data) Function

The final white box test was performed on the updateAirStatusAndBuzzer(data) function, which is one of the core functions of the system as it is responsible for synchronizing sensor status, air quality classification, and buzzer control as an early warning mechanism. Because the function operates within a FreeRTOS-based multitasking environment, it must handle mutex synchronization, sensor validity checking, manual override modes, and air quality threshold evaluation. Control flow analysis showed that the function contains six decision nodes with a cyclomatic complexity value of **7**, requiring seven independent paths for complete coverage.

Table 10. Basis Path Test Cases for updateAirStatusAndBuzzer()

TC ID	Scenario	Input (AirData)	Override	Path	Expected Buzzer Behavior	Status
TC-13	Mutex acquisition failure	Valid=True, PPM=500	False	Path 1	No change	Pass
TC-14	Sensor failure, manual mode	Valid=False, PPM=0	True	Path 2	Remains active	Pass
TC-15	Sensor failure, automatic mode	Valid=False, PPM=0	False	Path 3	Completely OFF	Pass
TC-16	Normal sensor, manual mode	Valid=True, PPM=2200	True	Path 4	Remains OFF	Pass
TC-17	Clean air, automatic mode	Valid=True, PPM=600	False	Path 5	OFF (LOW)	Pass
TC-18	Severe pollution, automatic mode	Valid=True, PPM=2400	False	Path 6	ON (HIGH)	Pass
TC-19	Sensor validity anomaly	Valid=True (Internal False)	False	Path 7	OFF (LOW)	Pass

The results demonstrate that all seven independent paths were successfully executed and produced the expected behavior. The function maintained buzzer status during mutex acquisition failures, preserved manual control when override mode was active, and automatically disabled the buzzer when sensor failures occurred under automatic mode. Furthermore, the function accurately distinguished between normal air conditions and severe pollution levels based on gas concentration values obtained from the MQ-135 sensor. Under normal conditions (600 ppm), the buzzer remained inactive, whereas under severe pollution conditions (2400 ppm), the buzzer was activated as an early warning mechanism. Testing under sensor validity anomaly conditions further demonstrated that the system adopted a fail-safe approach by disabling the buzzer to avoid false alarm activation. Overall, the results indicate that the updateAirStatusAndBuzzer(data) function operates accurately, reliably, and safely under various operational scenarios, including FreeRTOS-based multitasking environments requiring concurrent process synchronization.

User Acceptance Testing (UAT)

User Acceptance Testing (UAT) was conducted to evaluate user acceptance of the developed air quality monitoring system and to assess its alignment with the research objectives. The evaluation utilized a Google Forms questionnaire consisting of binary response options ("Yes" and "No") distributed to 20 employees of PT. Bianglala Metropolitan. The eight questions were divided into two categories. Questions 1 to 4 evaluated the success of system design and implementation, whereas Questions 5 to 8 assessed the system's capability to perform real-time air quality monitoring and its contribution to mitigating health risks in the workplace.

International Journal Of Health, Engineering And Technology (IJHET) *E-ISSN 2829 - 8683*
Volume 5, Number 2, July 2026, *Page. 2492 - 2508*
Email : editorijhess@gmail.com

Table 11. User Acceptance Test Questionnaire Results (n = 20)

No.	Question	Yes (%)	No (%)
1	The system is needed by PT. Bianglala Metropolitan	95	5
2	The gas, temperature, and humidity sensors accurately measure air conditions	85	15
3	The web monitoring dashboard is easy to understand and use	90	10
4	The ESP32-S3 device and website are well integrated	80	20
5	The system provides real-time data monitoring	90	10
6	Buzzer notifications help users quickly recognize unsafe conditions	95	5
7	The system helps reduce health risks	90	10
8	The system improves workplace comfort and environmental quality	90	10

Based on the results presented in Table 11, the average user acceptance rate reached 89.4% positive responses, indicating a very high level of acceptance among end users. For the first research objective concerning system implementation, the average positive response rate was 87.5%. The lowest score (80%) was observed for the integration between the ESP32-S3 device and the website, suggesting opportunities for future improvements in communication stability and network connectivity. For the second research objective concerning real-time monitoring and health risk mitigation, the average positive response rate reached 91.25%. These findings indicate that the system successfully provided real-time monitoring, delivered effective early warnings through the buzzer, and contributed to reducing health risks associated with poor air quality. The highest positive response rate (95%) was obtained for the perceived necessity of implementing the system at PT. Bianglala Metropolitan and the effectiveness of the buzzer as an early warning mechanism. These findings suggest that an IoT-based air quality monitoring system is highly valuable for supporting a healthier and safer workplace environment. Overall, the UAT results demonstrate that the developed Internet of Things-based air quality monitoring system successfully met user requirements and achieved the research objectives from both technological and practical perspectives within the operational environment of PT. Bianglala Metropolitan.

CONCLUSION

Based on the results of the design, implementation, and testing of the Internet of Things (IoT)-based indoor air quality monitoring system conducted within the operational environment of PT. Bianglala Metropolitan, it can be concluded that this study successfully designed and implemented an IoT-based indoor air quality monitoring system by integrating an MQ-135 gas sensor for gas concentration detection, a DHT22 sensor for temperature and humidity measurements, a 20×4 LCD as a local display interface, a 3V active buzzer as an early warning mechanism, and an ESP32-S3 microcontroller as the central system controller. The testing results demonstrated that the DHT22 sensor was capable of measuring temperature and humidity with stable readings and fluctuations of less than 0.2°C during two hours of continuous operation. Meanwhile, the MQ-135 sensor successfully performed automatic baseline resistance (R0) calibration and implemented a step limit filtering mechanism to reduce abnormal measurement spikes. From the software perspective, the system was successfully integrated with an architecture based on NestJS, PostgreSQL, Next.js, and the MQTT protocol through the HiveMQ Cloud broker, enabling real-time data transmission and visualization. White box testing using the Basis Path Testing method demonstrated that all 19 independent paths across the three critical system functions were successfully executed and produced outputs consistent with the system design specifications. These findings indicate that the developed system possesses a high level of reliability in both hardware and software aspects, thereby confirming that the first research objective concerning the design and implementation of an IoT-based air quality monitoring system was successfully achieved.

The developed system successfully performed real-time air quality monitoring and provided early warnings for air conditions that could potentially endanger user health. Black box testing results showed that the system was capable of identifying clean air conditions at gas concentrations below 800 ppm and automatically activating the buzzer when gas concentrations exceeded the threshold value of 1000 ppm, with a response time of less than one second after hazardous conditions were detected. Network communication testing demonstrated that the ESP32-S3 was capable of transmitting telemetry data periodically every 10 seconds, receiving remote control commands through MQTT, performing automatic reconnection when Wi-Fi interruptions occurred, and temporarily storing telemetry data during offline conditions using an in-memory queue mechanism with a capacity of eight records. Furthermore, the Task Watchdog Timer mechanism successfully recovered the system automatically when deadlock conditions occurred, while the power recovery mechanism restored the system to normal operational status in less than five seconds after power supply restoration. The User Acceptance Test (UAT) results also showed a user acceptance rate of 89.4%, indicating that the system was considered useful, easy to use, and capable of supporting improvements in workplace environmental quality. Therefore, the developed system has been proven to operate in a stable, responsive, and sustainable manner in performing real-time air quality monitoring and supporting the mitigation of health risks associated with deteriorating indoor air quality, thereby confirming that the second research objective was successfully achieved.

REFERENCES

Allen, J. G., MacNaughton, P., Satish, U., Santanam, S., Vallarino, J., & Spengler, J. D. (2016). *Associations of Cognitive Function Scores with Carbon Dioxide, Ventilation, and Volatile Organic Compound Exposures in Office Workers: A Controlled Exposure Study of Green and Conventional Office Environments*. *Environmental Health Perspectives*, 124(6), 805–812. https://doi.org/10.1289/ehp.1510037.

Anifam, A. (2021). *Rancang Bangun Sistem Keamanan Brankas Menggunakan Sidik Jari (Fingerprint) Berbasis Arduino Uno dengan Notifikasi Telegram*.

Andrizal, Lifwarda, Antonisfia, Zulharb, & Yuhefizar. (2020). *Sistem Kontrol Berbasis Pemrograman LabVIEW MyRIO untuk Monitoring Kualitas Udara Dalam Ruangan*.

Hanum, L., & Elfizon. (2023). *Rancang Bangun Pemantau Kualitas Udara Dalam Ruangan Berbasis Internet of Things*.

Kementerian Kesehatan Republik Indonesia. (2011). *Peraturan Menteri Kesehatan Republik Indonesia Nomor 1077/MENKES/PER/V/2011 tentang Pedoman Penyehatan Udara Dalam Ruang Rumah*. Jakarta: Kementerian Kesehatan Republik Indonesia.

Kristian. (2023). *Implementasi sebagai Tahap Operasionalisasi Gagasan dalam Sistem*.

Maharaksa. (2025). *Tahapan Implementasi dalam Pengembangan Sistem dan Evaluasi Program*.

Mokodompit, Wullur, Pasandaran, & Rotty. (2023). *Implementasi sebagai Optimalisasi Pemanfaatan Sumber Daya*.

Murad, R. F., Almasir, G., & Harahap, C. R. (2022). *Pendeteksi Gas Amonia Untuk Pembesaran Anak Ayam Pada Box Kandang Menggunakan MQ-135*. *Jurnal Ilmiah Mahasiswa Kendali dan Listrik*, 3(1), 120–130.

Naim, M. (2022). *Teori Dasar Listrik dan Elektronika*. Bojong Pekalongan: PT. Nasya Expanding Management.

Nasution, S. L., Girsang, E., Butar-Butar, E. F., Hafizah, N. Z., & Manalu, P. (2025). *Dampak kualitas udara dalam ruangan terhadap kejadian ISPA di Kecamatan Medan Denai*. *Jurnal Kesehatan Lingkungan Indonesia*, 24(3), 277–286.

Nugroho, Y. C., Mahardiko, A. A., Dhelia, S. S., Nuraini, A., & Harsadi, P. (2023). *Rancang Bangun Sistem Pemantauan Kualitas Udara Berbasis Arduino Untuk Mendeteksi Polusi Udara di Perkotaan*. *Jurnal TIKomSiN*, 11(2), 45–52. https://doi.org/10.30646/tikomsin.v11i2.795.

Oper, N., & Hasan, G. J. (2021). *Rancang Bangun Buka Tutup Tempat Sampah Otomatis Berbasis Arduino*. *JINTEKS (Jurnal Informatika Teknologi dan Sains)*, 385.

Ridho'i, A., Setyadjit, K., & Hariadi, B. (2022). *Pengaruh Suhu dan Kejernihan Air Pada Kolam Terpal Pembesaran Ikan Nila Memanfaatkan ATMEGA328*. *Jurnal Teknik Industri*, 38–51.

Salsabila, A. A. (2021). *Rancang Bangun Alat Deteksi Kadar Alkohol Pada Toner Wajah Berbasis Android*. Tugas Akhir, Politeknik Negeri Sriwijaya.

Santosa, Hasan, A., & Selvia, N. (2023). *Rancang Bangun Model Alat Penyiram Otomatis Bibit Kelapa Sawit Berbasis Arduino Uno dan Soil Moisture Sensor*. Ponorogo: Uwais Inspirasi Indonesia.

Sendari, S., Wirawan, I. M., & Nasrullah, M. (2021). *Sensor Tranduser*. Malang: Ahlimedia Press.

Sujiarta, Wirarama, & Zubaidi. (2023). *Sistem Monitoring Kualitas Udara di Ruangan Tertutup Berbasis IoT Menggunakan Sensor MQ-135 dan GP2Y1014AU0F*.

Yunardi, D. H., Misbullah, A., & Gemilang, G. (2023). *Rancang Bangun Sistem Web Monitoring Kualitas Udara di Dalam Ruangan Menggunakan Logika Fuzzy*. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 7(1), 25–34.

Zafira, Ghozali, & Sabilla. (2022). *Rancangan Bangun Prototype Monitoring Kualitas Udara dalam Ruang*.