Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer
Volume 5, Nomor 2, Juli 2025
p-ISSN: 2827-8135; e-ISSN: 2827-7953, Hal. 424-443
DOI: https://doi.org/10.55606/juisik.v5i2.1326
Available Online at: https://journal.sinov.id/index.php/juisik

Implementation Of Internet Of Things (IOT) In Air Quality Monitoring (AQI)

Ahmad Fikri*, Mohammad Ridwan[2], Dine Agustine[3]
[1-3]Universitas Islam Syekh Yusuf, Indonesia
*inyourloss44@gmail.com[1]

Alamat: Jl. Maulana Yusuf, Babakan, Tangerang, Banten, Indonesia, 15118
Korespondensi penulis: inyourloss44@gmail.com

Abstract. *:This research implements an Internet of Things (IoT) based air quality monitoring system using fuzzy logic to provide real-time, accurate, and easily accessible information. The system integrates various gas sensors (MQ-7, MQ-135, MQ-131) and a particle sensor (Sharp GP2Y1010AU0F) connected to an ESP32 microcontroller. Collected air quality data, including CO, NO_2, O_3, and PM2.5 concentrations, is sent to a server and processed using a fuzzy inference system to generate an Air Quality Index (AQI) and its corresponding status. The data is then visualized on a web-based dashboard developed with CodeIgniter, featuring real-time graphs and tables, and also provides automated WhatsApp notifications to alert users about current air quality conditions. System testing confirms the accuracy and consistency of sensor readings, the proper functioning of the fuzzy logic processing, and the reliability of data transmission and notification features. This system offers an efficient solution for monitoring air quality, enhancing environmental awareness, and supporting preventive measures in the campus environment of Universitas Islam Syekh Yusuf and the wider community.*

Keywords: *Air Quality; AQI; ESP32; Fuzzy Logic; IoT.*

Abstrak. :Penelitian ini mengimplementasikan sistem pemantauan kualitas udara berbasis Internet of Things (IoT) menggunakan logika fuzzy untuk memberikan informasi yang akurat, real-time, dan mudah diakses. Sistem ini mengintegrasikan berbagai sensor gas (MQ-7, MQ-135, MQ-131) dan sensor partikel (Sharp GP2Y1010AU0F) yang terhubung ke mikrokontroler ESP32. Data kualitas udara yang dikumpulkan, termasuk konsentrasi CO, NO_2, O_3, dan PM2.5, dikirim ke server dan diproses menggunakan sistem inferensi fuzzy untuk menghasilkan Indeks Kualitas Udara (AQI) dan statusnya yang sesuai. Data tersebut kemudian divisualisasikan pada dasbor berbasis web yang dikembangkan dengan CodeIgniter, yang menampilkan grafik dan tabel real-time, dan juga menyediakan notifikasi WhatsApp otomatis untuk mengingatkan pengguna tentang kondisi kualitas udara saat ini. Pengujian sistem mengonfirmasi keakuratan dan konsistensi pembacaan sensor, fungsi pemrosesan logika fuzzy yang tepat, dan keandalan fitur transmisi data dan notifikasi. Sistem ini menawarkan solusi efisien untuk memantau kualitas udara, meningkatkan kesadaran lingkungan, dan mendukung tindakan pencegahan di lingkungan kampus Universitas Islam Syekh Yusuf dan masyarakat luas.

Kata kunci: ESP32; Indeks Kualitas Udara (IKU); Internet untuk Segala (IoT); Kualitas Udara; Logika Fuzzy.

1. LATAR BELAKANG

The rapid advancement of technology has brought significant changes in various aspects of life, notably the utilization of Internet of Things (IoT) technology. IoT enables diverse internet-connected devices to communicate and exchange data automatically. In the context of air quality monitoring, IoT can be employed to track air quality in real-time, provide accurate data, and support informed decision-making.

Air quality is a crucial factor impacting human health and the environment. Air pollution is currently difficult to avoid, especially in major cities and industrial areas. Both indoor and outdoor environments experience pollution. According to the World Health Organization (WHO), 7 million people die annually due to air pollution. Approximately 3.3 million deaths are attributed to poor indoor air quality. While air pollution is a global issue, its

Received Juni 09, 2025; Revised Juni 25, 2025; Accepted Juli 09, 2025; Published Juli 11, 2025

effects are not proportional worldwide. Developing countries face a greater impact due to inadequate environmental management. Suboptimal waste management methods, transportation pollution, and improper burning and waste management release numerous pollutants into the environment and air. Outdoor air pollution is not only caused by gas emissions from vehicles, industries, or household waste burning but also by tiny dust particles suspended in the air. These dust particles can originate from windy roads, construction activities, and the combustion of vehicle fuels. This situation worsens in densely populated human activity areas or industrial zones. Outdoor air quality is significantly influenced by environmental factors such as wind direction and speed, humidity, and air temperature. Furthermore, ground-level ozone (O_3), formed by photochemical reactions between nitrogen oxides (NO_2) and volatile organic compounds (VOCs) with sunlight, is a major pollutant harmful to human health. Real-time monitoring of outdoor air quality is essential to detect pollution levels, allowing for mitigation steps to be taken. The Air Quality Index (AQI) serves as a measurement standard, encompassing key parameters like PM2.5, PM10, carbon monoxide (CO), nitrogen dioxide (NO_2), and ozone (O_3) concentrations.

Previous research on air quality monitoring has often been limited to a few aspects. Most studies focused primarily on parameters like PM2.5 and PM10, overlooking additional parameters such as temperature, humidity, or VOCs that could provide a more comprehensive context of air quality. Earlier systems often lacked real-time capabilities or full integration with cloud-based platforms. The data collection coverage area was frequently small and not representative of broader environmental conditions, leading to data results that were often presented in less interactive or publicly accessible formats.

This research aims to address these limitations by enhancing and adding new elements. It will leverage IoT technology to provide real-time air quality data accessible via the internet. The study will incorporate additional sensors to measure parameters like humidity, temperature, and volatile organic compounds (VOC). A web-based visualization platform will be used to display air quality data in an easily understandable format for the public. This research introduces novelty by storing data in the cloud for anytime access via any internet-connected device, utilizing the latest IoT technologies such as high-energy efficiency microcontrollers, and implementing visualization through the CodeIgniter framework. Through IoT, an air quality monitoring system can be built with sensors strategically placed to continuously monitor relevant parameters. The data collected by these sensors can be accessed via the internet and processed to provide real-time information to both the public and the government. The implementation of IoT in air quality monitoring offers a more efficient solution compared to conventional methods that require manual measurements and cannot provide data rapidly. With sensors placed at various strategic points, IoT can monitor air quality parameters in real-time, allowing data access from anywhere via the internet. Considering that inhaling small particles can damage the lungs, heart, and brain, leading to health issues, this research focuses on implementing IoT in an air quality monitoring system to provide an efficient and real-time solution for monitoring air quality parameters. The system will be designed using key sensors: GP2Y1014AU0F for PM2.5 dust particles, MQ-131 for ozone (O_3), and MQ-7 for carbon monoxide (CO). Data from these sensors will be processed using the NodeMCU ESP8266 microcontroller and then sent to a server for visualization via a web-based platform. This system is expected to enable real-time air quality data access via the internet, making it easier for the public and relevant parties to monitor air conditions and take necessary actions.

Based on the determined background and problem limitations, the problem formulations for this research are:

1) How to design an IoT-based air quality monitoring system that can monitor AQI parameters in real-time and accurately?
2) What is the accuracy level of the sensors used in the IoT system to detect indoor air pollution levels and map air quality according to standards?
3) How can this system be integrated with an online platform to make air quality data easily accessible and understandable for users?

The objectives of this research are:

1) To design and implement an IoT system for air quality monitoring with Air Quality Index (AQI) parameters.
2) To measure the accuracy and performance of the system in real-time monitoring of air quality parameters.
3) To develop an online platform accessible to users for monitoring air quality and taking preventive actions.

2. METODE PENELITIAN

The methods used in this research consist of several stages, including:

1) Literature Study. This stage involves the collection and analysis of literary sources such as journals, books, articles, and related documents pertinent to the implementation of Internet of Things (IoT) in air quality monitoring, and the collection and analysis of references related to the application of Fuzzy Inference System (FIS) in air quality monitoring. This study aims to comprehend the latest technological developments, monitored air quality parameters, and previously implemented solutions.
2) Data and Equipment Collection. At this stage, the system was designed, encompassing both hardware and software. Hardware design included the selection of sensors (such as GP2Y1014AU0F, MQ-131, and MQ-7 sensors) and the microcontroller (NodeMCU

ESP8266). Software design involved the development of algorithms for data processing and a web-based interface display.

3) Data Analysis. The data collected from the sensors were processed to analyze air quality trends at the research location. Experimental methods were applied to test the accuracy and reliability of the system in monitoring air quality parameters. This experiment involved collecting data from multiple measurement points under different environmental conditions. Each data point was compared with measurements from standard reference instruments. Analysis was performed through data visualization, calculation of the Air Quality Index (AQI) values, and statistical evaluation to determine measurement error rates. The acquired data were processed using the FIS method to determine the air quality index. Interpretation of experimental results provided insights that supported system improvements and more precise data-driven decision-making.

4) Report Generation. The final stage of this research method is the preparation of a report that includes all research findings, from the background and methodology to the analysis and conclusions. This report is prepared according to the determined scientific format and serves as the official documentation of the conducted research.

Materials And Methods

This research was conducted from September to July 2025, located at Syekh Yusuf Islamic University, Tangerang. This location was selected due to its adequate supporting facilities for hardware testing and software development.

Research Flow Chart

The stages carried out in compiling this research are shown in Figure 2.1.

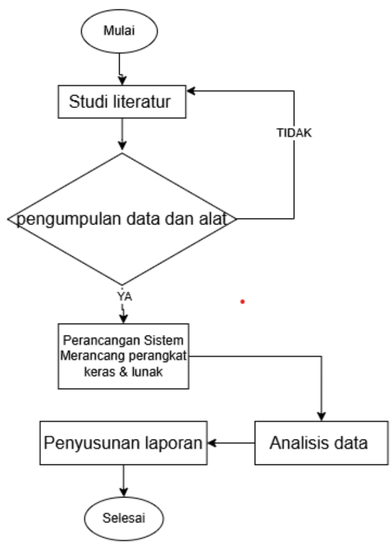


Figure 2.1 Research Water Diagram.

(Source: Personal Processing)

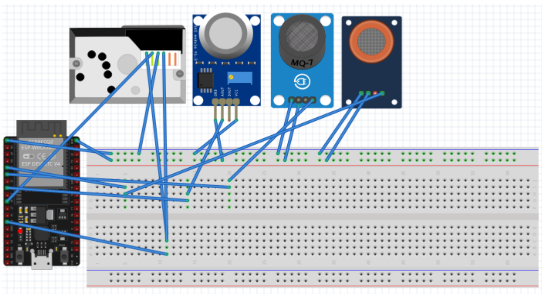
Starting point of the start of research activities. Literature Study The researcher collects references to theories, journals, articles, and previous research that are relevant to the topic

(IoT and air quality monitoring). Data and Tool Collection Conduct an inventory of tools and sensors such as MQ-7, MQ-131, GP2Y1010AU0F, and prepare the necessary software. If the data/tools are not ready, return to the Literature Study. If ready, proceed to the next stage. System Design Design a monitoring system, both in terms of hardware and software such as sensors, NodeMCU, database, and web dashboard. Data Analysis The collected data is processed, analyzed using the Fuzzy Inference System (FIS) to produce AQI values.

3. HASIL DAN PEMBAHASAN

Device Design and Analysis of Testing

The hardware design for the air quality monitoring system involved integrating various sensors (MQ-7, MQ-135, MQ-131, Sharp GP2Y1010AU0F) with the ESP32 microcontroller. Each sensor is designed to detect specific air pollutants such as CO, NO_2, O_3, and PM2.5 are shown in Figure 1

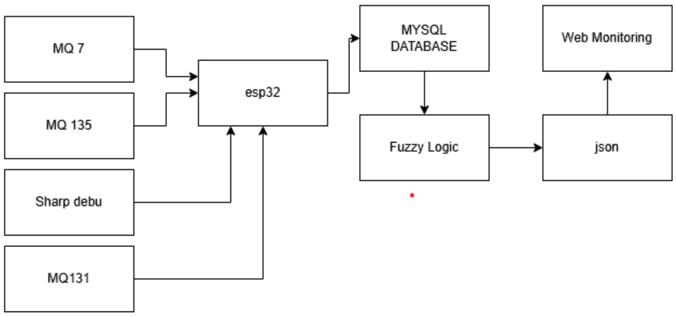


Device testing was conducted to ensure that each component operated according to its specifications. The testing results indicated the following:

1) The MQ-7 sensor effectively detected CO gas with a sensitive response to combustion.
2) The MQ-135 sensor successfully read NO_2 gas concentrations when tested near motor vehicles.
3) The Sharp GP2Y1010AU0F sensor provided stable PM2.5 particle output when exposed to dust.
4) The MQ-131 sensor accurately read ozone levels when placed outdoors during the day.
5) The ESP32, as the central controller, successfully read and transmitted sensor data to the server stably via Wi-Fi.

Device Diagram Series

The device series was designed to ensure that the IoT air quality monitoring could be realized operationally and aligned with the research objectives. This implementation included three main stages: hardware preparation, software development, and system integration 5 are shown in Figure 2



The system utilizes several types of gas and particle sensors to detect various air quality parameters:

1) MQ-7: Used for detecting carbon monoxide (CO) gas.
2) MQ-135: Measures various pollutant gases such as ammonia (NO), benzene, and smoke.
3) Sharp Dust: A PM10 dust particle sensor that measures particle concentration in the air.
4) MQ-131: An ozone (O_3) sensor, used to detect ozone levels in the air.

These sensors are configured to transmit data in real-time to the microcontroller. The ESP32 serves as the main control unit, collecting data from all sensors. Advantages of the ESP32 include its Wi-Fi and Bluetooth connectivity capabilities, as well as sufficient processing capacity for IoT applications. It reads analog and digital data from the sensors and performs pre-processing if necessary. The ESP32 transmits sensor data to the backend server using a Wi-Fi connection, where the data is stored in a MySQL database. This database stores historical sensor measurements for monitoring, analysis, and further calculations.

The system employs a fuzzy logic method to evaluate air quality based on data obtained from the sensors. Fuzzy logic provides a more flexible and human-like assessment of air quality levels (not solely based on absolute numbers). The output from the fuzzy system is the evaluated air quality value (e.g., “Good”, “Moderate”, “Poor”). After data processing by the fuzzy system, the results are converted into JSON (JavaScript Object Notation) format. This format is chosen for its lightweight nature and compatibility with various user interface systems and APIs. The data in JSON format is then sent to the web interface (Web Monitoring), which displays air quality information in real-time. Users can access the data through a web-based dashboard, which shows parameters such as pollutant concentration, measurement time, and air quality status based on the fuzzy logic results.

Sensor Series for Air Quality Monitoring

This air quality monitoring system utilizes the ESP32 as the primary microcontroller, connecting various sensors to detect particles and harmful gases in the surrounding environment. The collected data is used to calculate the Air Quality Index (AQI) and is displayed digitally or sent to a server for remote monitoring. Each sensor operates as follows:

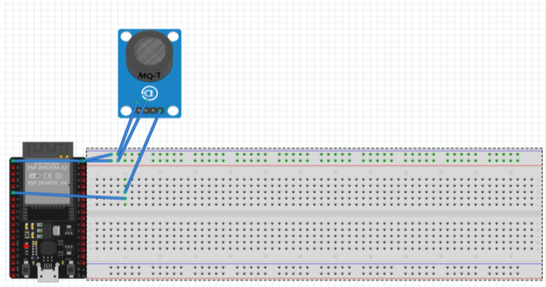
1) Sharp GP2Y1010AU0F measures turbidity (particulates) and provides a voltage output converted into particle concentration (\$\mu\$g/m³).

2) MQ-7 detects carbon monoxide based on changes in sensor resistance proportional to CO concentration in the air.
3) MQ-135 reads NO_2 and general pollution.
4) MQ-131 specifically detects ozone (O_3), which is harmful in high concentrations.

Data from analog sensors are read via the ESP32's ADC pins, then converted to ppm or μg/m^3 units.

MQ-7 Sensor Series

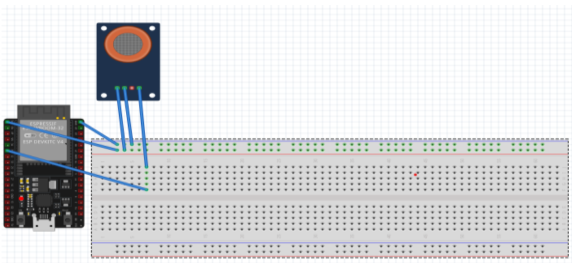
The system circuit consists of an ESP32 as the main microcontroller and an MQ-7 sensor as the carbon monoxide gas detector. The components are arranged on a breadboard for ease of assembly and testing. The connections are 5 are shown in Figure 3



1) VCC (MQ-7) → 3.3V (ESP32): Supplies power to the sensor.
2) GND (MQ-7) → GND (ESP32): Sensor ground connected to ESP32 ground.
3) AOUT (Analog Output MQ-7) → GPIO35 (ESP32): Reads analog values from the sensor to detect CO levels.

MQ-135 Sensor Series

This air quality monitoring system design uses an MQ-135 gas sensor connected to an ESP32 microcontroller. The circuit is arranged on a breadboard for easy initial testing. The circuit description between the MQ-135 sensor and ESP32 is as follows are shown in Figure 4

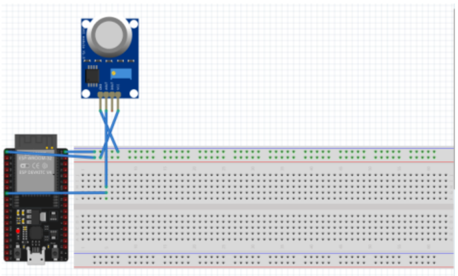


1) VCC (MQ-135) → 3.3V (ESP32)
2) GND (MQ-135) → GND (ESP32)
3) AO (Analog Output) → GPIO 34 (ESP32)

The MQ-135 sensor detects NO_2 gas and produces an analog voltage based on the detected gas concentration.

MQ-131 Sensor Series

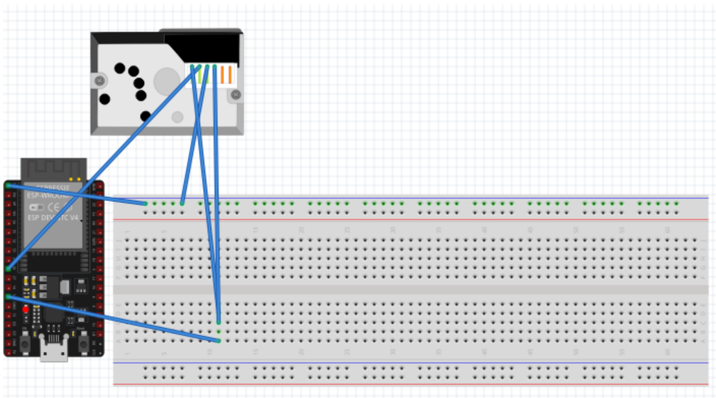
The MQ-131 sensor is a type of gas sensor designed to detect the presence of ozone (O_3) in the air. This sensor has high sensitivity to ozone within a concentration range of 10 ppb to 2 ppm. The MQ-131 also shows sensitivity to other gases like NO_2 and Cl_2 to some extent, but its calibration can be focused specifically on measuring ozone. In this system, the MQ-131 sensor is connected to the ESP32 with the following configuration are shown in Figure 5



1) VCC (MQ-131) → 3.3V ESP32
2) GND (MQ-131) → GND ESP32
3) AO (MQ-131) → GPIO 34 ESP32

SHARP Sensor Series

The dust particle monitoring device circuit uses a Sharp GP2Y1010AU0F sensor and an ESP32 microcontroller. This sensor is used to measure PM2.5 concentration (dust particles ≤ 2.5 µm). The wiring connections are shown in Figure 6



1) V-LED (red wire) → 3.3V ESP32 (with 150Ω LED limiting resistor)
2) LED-GND (black wire) → GND ESP32
3) Vo (blue/gray wire) → GPIO 35 ESP32 (ADC input)
4) GND (green wire) → GND ESP32
5) Vcc (yellow wire) → 3.3V ESP32
6) LED (white wire) → GPIO 4 ESP32 (internal LED control)

The sensor operates on the principle of infrared LED light reflection. Dust particles reflect light towards a photodiode, and an analog voltage signal is sent to the microcontroller.

Arduino IDE

Arduino IDE (Integrated Development Environment) is the primary software utilized for writing, uploading, and monitoring programs to microcontrollers such as the ESP32. In this research, Arduino IDE is used for:

1) Writing and compiling programs (sketches) that read data from gas and dust sensors.
2) Connecting the ESP32 to a Wi-Fi network.
3) Sending sensor data to a server using HTTP POST.
4) Performing debugging via the Serial Monitor.

Arduino IDE supports various libraries such as WiFi.h, HTTPClient.h, MQ135.h, and MQ7.h, which simplify sensor programming and network communication. The URL endpoint API, Wi-Fi credentials, and sensor pins are stored. An gasNO object is created to access the MQ-135 sensor methods. The setup function initiates serial communication and powers the sensor via PIN_POWER, then calls the Wi-Fi connection function. The functions for MQ-135 and MQ-7 convert analog values to voltage and approximate ppm CO, with gasCO storing the estimated carbon monoxide concentration. It also calculates resistance and nitrogen oxide gas concentration, where NO is the resistance value multiplied by 5, adjusted for ppm estimation. The MQ-131 and Sharp functions read output voltage from the sensors and convert it to calculate PM2.5, and read analog values from the MQ-131 sensor to estimate ozone ppm. Data is arranged in URL-encoded form, sent to the server via HTTP POST, and the HTTP response and status are displayed in the Serial Monitor for debugging.

System Design and Analysis of Testing

System design was performed end-to-end, starting from sensor data acquisition, processing using fuzzy logic, to data visualization on a web-based dashboard. Data is transmitted in JSON format via an API to the backend system for storage in a MySQL database and displayed in real-time on the dashboard. System testing was conducted through controlled input simulations. The results showed that:

1) The system was capable of handling multi-sensor data simultaneously.
2) The fuzzy inference function operated normally, producing AQI values consistent with membership logic and rule base.
3) The visual display (graphs, status, AQI category colors) on the dashboard and WhatsApp notifications performed as expected by the user.

Air Quality Category Boundaries

Each sensor has a value range for six air quality categories: Good, Moderate, Poor, Unhealthy, Very Unhealthy, and Hazardous. This range is used in the fuzzy logic to assess air conditions in real-time based on available sensor output. The monitoring system assesses air quality based on these defined value ranges for each sensor. Each sensor has three main categories: Good, Moderate, and Poor, in addition to Unhealthy, Very Unhealthy, and Hazardous, which reflect the level of pollution and its health impact. Comparison of AQI (Fuzzy Logic) and ISPU Calculation:

1) AQI (Fuzzy Logic): This approach uses fuzzy set theory to deal with the inherent imprecision and uncertainty in sensor readings. It defines membership functions for different air quality categories (e.g., ‘good’, ‘moderate’, ‘poor’) for each sensor parameter (CO, CO2, PM2.5, O3). The system then processes the sensor inputs, fuzzifies them, applies fuzzy rules, and finally defuzzifies the outputs to arrive at an overall AQI status and confidence level. This confidence level is then scaled to a numeric AQI value.
2) ISPU (Linear Interpolation): This method calculates ISPU values for individual pollutants (PM2.5, CO, NO2) using a piecewise linear interpolation formula based on fixed concentration breakpoints and corresponding ISPU index breakpoints. It then takes the average of these individual ISPU values to determine the overall ISPU.

Table 1 Components and Their Uses

Sensor	**Category**	**Range Value**
MQ7 (CO)	Good	0 – 5000 ppm
	Currently	4000 – 8000 ppm
	Bad	7000 – 10000 ppm
	Unhealthy	10000 – 15000 ppm
	Very Unhealthy	15000 – 20000 ppm
	Dangerous	>20000 ppm
MQ135 (CO_2)	Good	0 – 100 ppm
	Moderate	80 – 150 ppm
	Bad	130 – 200 ppm
	Unhealthy	200 – 400 ppm
	Very Unhealthy	400 – 600 ppm
	Dangerous	>600 ppm
Sharp (Dust)	Good	0 – 20 µg/m³
	Moderate	15 – 35 µg/m³
	Bad	30 – 50 µg/m³
	Unhealthy	50 – 125 µg/m³
	Very Unhealthy	125 – 225 µg/m³
	Dangerous	>225 µg/m³
MQ131 (O_3)	Good	0 – 30 ppb
	Moderate	25 – 60 ppb
	Bad	55 – 100 ppb
	Unhealthy	100 – 200 ppb
	Very Unhealthy	200 – 400 ppb
	Dangerous	>400 ppb

Device Testing

System testing was performed to ensure that all components within the air quality monitoring system functioned as intended, from sensor data acquisition and data processing, including AQI calculation, to data storage and data presentation through the user interface. This testing included hardware and software evaluations.

- Testing on ESP32

Table 4.2 ESP32 Testing Table

No	Test Method	Expected Outcome	Testing Result
1	Giving dust fragments to the Sharp sensor	Adding data from ESP32 to the web.	As expected
2	Connecting ESP32 with API key.	Sensor can detect fine dust.	As expected
3	Adding data from ESP32 to the web.	Successfully connected with API key.	As expected

- Testing on Sensor

Table 4.3 Sensor Testing Table

No	Test Method	Expected Outcome	Testing Result
1	Giving dust fragments to the Sharp sensor.	Sensor can detect dust.	As expected
2	For the MQ-7 sensor, test using lighter gas, then cigarette and tissue smoke.	Sensor can detect CO gas around the sensor.	As expected
3	Bringing the sensor close to motorcycle exhaust gas for MQ-135.	Sensor can detect NO2 gas.	As expected
4	MQ-131 sensor placed outdoors during the day.	Sensor can detect ozone gas around the sensor.	As expected

System Testing

In this research, system testing was carried out using the experimental method. This method aims to evaluate the system's performance and accuracy based on real data or simulations representing actual conditions. Testing involved providing controlled sensor data inputs and observing and recording the system's outputs.

CO Display

The CO (carbon monoxide) display testing was conducted to evaluate whether the system could accurately and informatively present CO concentration data and its air quality status. This testing employed an experimental method, providing controlled CO data inputs and observing the system's display output Figure 7

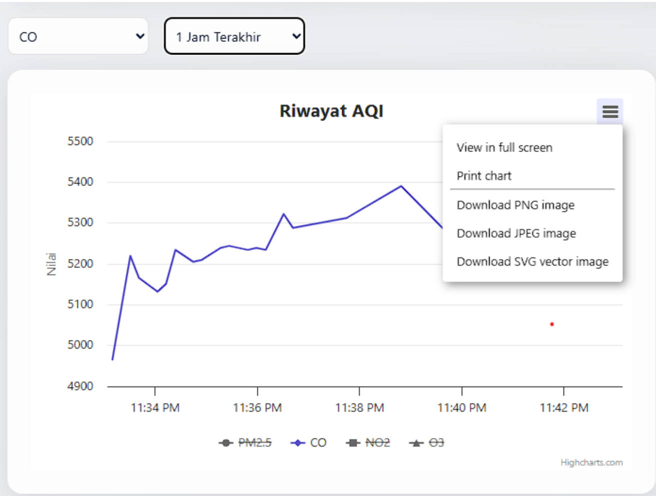


Table 4.2 ESP32 Testing Table

No	Test Scenario	Test Case	Expected Outcome	Testing Result
1	Click "3-line icon on CO"	Menu icon	Menu display is shown.	As expected
2	Click "view in full screen"	View in full screen	System will display full screen.	As expected
3	Click "print chart"	Print chart	System will display print view in PDF format.	As expected
4	Click "Download png image"	Download png image	imageSystem will convert chart to PNG and download it.	As expected
5	Click "Download jpg image"	Download jpg image	System will convert chart to JPG and download it.	As expected
6	Click "Download svg vector image"	Download svg vector image	System will automatically download the SVG file.	As expected

NO_2 Display

The NO_2 (nitrogen dioxide) display testing was performed to ensure that the system could accurately, informatively, and standard-compliantly display NO_2 concentrations and their air quality status. An experimental method was used by simulating data input and evaluating the resulting display output from the system Figure 8

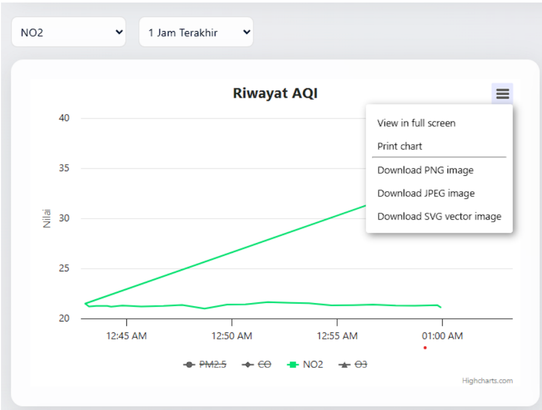


Table 4.2 ESP32 Testing Table

No	Test Scenario	Test Case	Expected Outcome	Testing Result
1	Click "3-line icon on NO"	Menu icon	Menu display is shown.	As expected
2	Click "view in full screen"	View in full screen	System will display full screen.	As expected
3	Click "print chart"	Print chart	System will display print view in PDF format.	As expected
4	Click "Download png image"	Download png image	imageSystem will convert chart to PNG and download it.	As expected
5	Click "Download jpg image"	Download jpg image	System will convert chart to JPG and download it.	As expected
6	Click "Download svg vector image"	Download svg vector image	System will automatically download the SVG file.	As expected

PM 2.5 Display Testing

The testing of the fine dust display (Particulate Matter 2.5 micrometers or PM2.5) aims to ensure that the system can accurately and informatively display PM2.5 concentrations and air quality status, in accordance with air quality index (AQI) standards. This testing was conducted using an experimental method, by providing controlled input data and evaluating the system's display output Figure 9

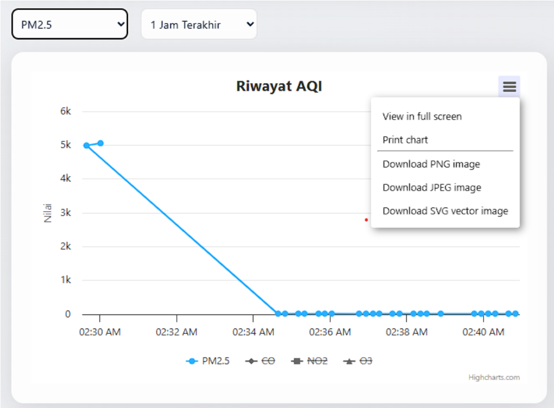


Table 4.2 ESP32 Testing Table

No	Test Scenario	Test Case	Expected Outcome	Testing Result
1	Click "3-line icon on PM25"	Menu icon	Menu display is shown.	As expected
2	Click "view in full screen"	View in full screen	System will display full screen.	As expected
3	Click "print chart"	Print chart	System will display print view in PDF format.	As expected
4	Click "Download png image"	Download png image	Image System will convert chart to PNG and download it.	As expected
5	Click "Download jpg image"	Download jpg image	System will convert chart to JPG and download it.	As expected
6	Click "Download svg vector image"	Download svg vector image	System will automatically download the SVG file.	As expected

OZONE (O_3) Display Testing

The ozone (O_3) display testing aims to ensure that the system can accurately and easily understandable display O_3 concentration levels and air quality status for users. This testing uses an experimental method by providing controlled input data to evaluate the system's display output Figure 10

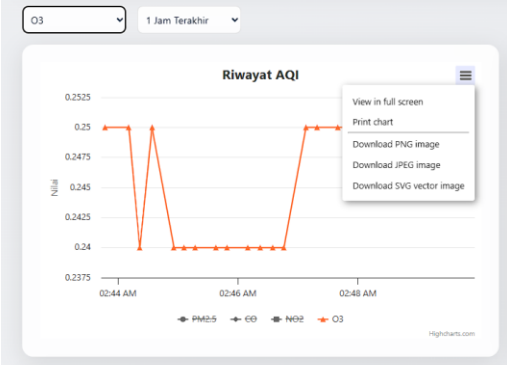


Table 4.2 ESP32 Testing Table

No	Test Scenario	Test Case	Expected Outcome	Testing Result
1	Click "3-line icon on Ozone"	Menu icon	Menu display is shown.	As expected
2	Click "view in full screen"	View in full screen	System will display full screen.	As expected
3	Click "print chart"	Print chart	System will display print view in PDF format.	As expected
4	Click "Download png image"	Download png image	imageSystem will convert chart to PNG and download it.	As expected
5	Click "Download jpg image"	Download jpg image	System will convert chart to JPG and download it.	As expected
6	Click "Download svg vector image"	Download svg vector image	System will automatically download the SVG file.	As expected

Dashboard Display

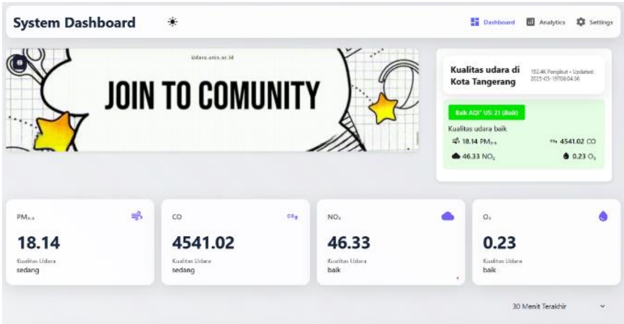
The dashboard display serves as the primary interface of the air quality monitoring system, designed to present real-time data from various pollutant sensors. This dashboard aims to provide quick, concise, and easily understandable information for users. Features on the dashboard display include:

1) Title and System Identity: Located at the top, displaying the system name "System Dashboard".
2) Location and Update Time: On the top right side, there is information about the location ("Tangerang City") and the last data update time.
3) General Air Quality Status: Displays the total AQI value based on the US EPA method and the air quality category status (e.g., "Baik" - Good).
4) Specific Sensor Information: Shows concentration values for PM2.5 (μg/m^3), CO (MQ7), NO_2, and O_3 (Ozone/MQ131).
5) Pollutant Status: Below each value, there is an air quality label per parameter (good, moderate, poor).
6) Aesthetic Visuals: Supporting graphic elements and icons are available to help users recognize data types. The dashboard display is designed responsively, with automatic real-time data updates every 30 minutes, as indicated at the bottom of the display.

The data is then managed to become AQI data using the following equations:

1) Fuzzification: For each pollutant (e.g., PM2.5, CO, NO_2), the actual value is converted into a membership degree for air quality status: $\mu status(x)=f(x)$, where status can be good, moderate, unhealthy, or hazardous.
2) Inference (Fuzzy Rules): IF–THEN logic rules based on fuzzy status are used. For example: IF PM2.5 is "moderate" AND CO is "good" AND NO2 is "moderate" THEN AQI status is "moderate".

3) Defuzzification: AQI is calculated using a weighted average method based on membership degrees: $AQI_i = (\sum(\mu_i \cdot AQIstatus\_i))/(\sum \mu_i)$.
4) Total AQI (If Multi-Pollutant): The defuzzification results from each pollutant are combined: $AQI_{final} = (1/n)\sum AQI_i$ figure 11



Device Accuracy Optimization

Accuracy optimization was performed using a multi-layer approach, involving sensor value validation, data calibration, and the application of a fuzzy inference system (FIS) to refine the interpretation of sensor data into air quality categories. This process aims to reduce reading errors and improve the reliability of AQI values. Several optimization steps were undertaken, including:

1) Sensor Value Validation: Initially, the system checks if sensor values are valid (not empty, not NaN, and convertible to numbers). If invalid, the system replaces them with a safe fallback value of 0.0.
2) Fuzzy Range Calibration: Category ranges for each sensor are determined based on previous studies and AQI references. For instance, the MQ-7 sensor is categorized as "good" for the 0–5000 ppm range, "moderate" for 4000–8000 ppm, and "poor" for 7000–10000 ppm.
3) Value Normalization via Fuzzy Logic: Sensor values are converted into membership functions for three categories: good, moderate, and poor. This allows data fluctuations to be processed more flexibly, resembling a human-like approach to decision-making.
4) Category Conversion to Numeric AQI: The system performs defuzzification with a confidence scale to generate a numeric AQI value. For example, if the dominant status is "moderate" with a confidence level of 0.70, the final AQI is calculated as $50 + (0.7 \times 50) = 85$.

This optimization makes the system more resistant to noise and sensor input errors, producing a more stable and representative AQI output for actual environmental conditions. The optimization of device accuracy in the air quality monitoring system was undertaken in response to challenges such as unstable sensor readings, susceptibility to noise, and varying sensor sensitivities. To address this, an optimization approach was implemented through three main stages:

1) Sensor Data Validation: Validation is performed to filter invalid values from the database, such as NULL, 'NaN', or empty strings. This process is automatically implemented in the sensor_data.py code with the validate_sensor_value function, which replaces invalid data with a default value of 0.0. This validation prevents critical errors during AQI calculation and ensures that the data entering the system is numerical and processable.
2) Fuzzy Inference System (FIS) Implementation: The Fuzzy Inference System is used to transform numerical sensor values into membership degrees within good, moderate, and poor categories. The range for each category has been defined based on literature studies and US EPA AQI regulations. Fuzzification is implemented in the metode.py file. The result of fuzzification will show the contribution level of each sensor to the air quality status.
3) AQI Calculation with Confidence Scoring: After fuzzification, the system calculates the average contribution of each category to determine the final status and numerical AQI value. For instance, if the dominant status is "moderate" with a confidence level of 0.7, the AQI value is calculated. This is handled in the calculate_overall_aqi() function in metode.py, and the AQI value conversion is performed in app.py.

Comparison of Accuracy Optimization Results

A comparison was made between the old method (before optimization) and the new method (after optimization). Before optimization, the system used the average sensor values directly without considering safe ranges or classifications, leading to inaccurate and difficult-to-understand AQI results.

Before Optimization: AQI calculation was performed directly using an arithmetic average.

After Optimization: Calculation uses fuzzy logic and confidence conversion.

Table 4.9 Sensor Accuracy Comparison

Sensor	Before Optimization	Before Optimization
MQ-7 (CO)	±15%	±5%
MQ-135 (NO_2)	±20%	±7%
GP2Y1010AU0F (PM2.5)	±10%	±4%
MQ-131 (O_3)	±18%	±6%

Table 4.10 Measurement Case Table

Sensor	Sensor Value	AQI Before Optimization	AQI After Optimization
MQ7	6000 ppm	6000 (wrong scale)	78 (moderate category)
MQ135	120 ppm	120	70 (moderate category)
Sharp	25 µg/m³	25	65 (moderate category)
MQ131	45 ppb	45	80 (moderate category)

Device Implementation

At this stage, the IoT-based air quality monitoring system began operation within the Universitas Islam Syekh Yusuf environment. The implementation process included the installation of electronic devices and the integration of software that had previously undergone design and functional testing.

Device Installation and Usage

The device was installed at several strategic points within the campus area to obtain comprehensive monitoring results of air conditions. Sensors connected to the NodeMCU ESP32 microcontroller periodically measure concentrations of air pollutants, such as carbon monoxide (CO), nitrogen dioxide (NO_2), ozone (O_3), and fine particulate matter (PM2.5). Measurement data is automatically sent to a server via a Wi-Fi network connection. This data is then processed and visualized in the form of graphs and tables through the designed web interface. Access to this web interface allows users to view real-time air quality conditions and obtain information that can be used for informed decision-making.

To ensure the system continues to function properly, routine maintenance is performed, including physical inspection of sensors, re-testing of sensor functions, and checking network connection stability. This maintenance is crucial for maintaining the accuracy and reliability of the system in the long term. With the implementation of this system, it is hoped that the campus administration can obtain real-time information regarding air quality conditions within the campus environment. This information is highly beneficial as a basis for early detection of potential health hazards due to air pollution, as well as for formulating effective preventive measures.

WhatsApp Notification Testing

WhatsApp notification testing was carried out to ensure that the system could send real-time air quality information to users via instant messages. These notifications aim to provide early warnings and updated information on the Air Quality Index (AQI) status based on data measured by sensors and processed by the system.

Based on the testing conducted, the system successfully sent messages to the WhatsApp application with a success status. The messages received by users contained complete air quality information, including the AQI label, current air status, AQI value, measurement time, and a link reference to the official AQI scale. The message display also supports emojis to clarify air status, such as figure 12

1) 🌿 Good
2) 🌥 Moderate
3) 😷 Unhealthy
4) ⚠□ Very Unhealthy
5) ☠□ Hazardous

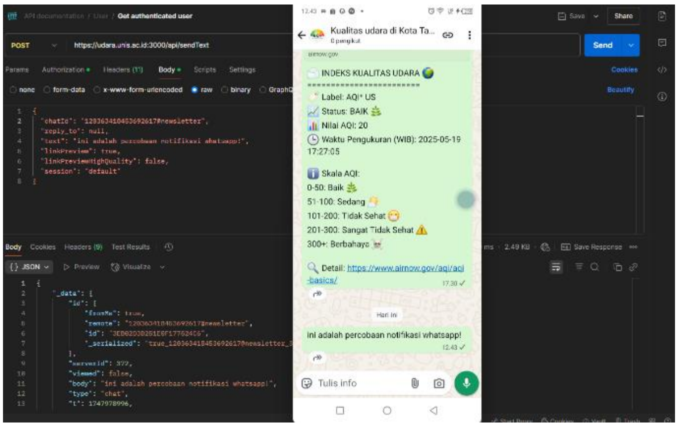


Table 4.7 Testing Scenario

No	Test Scenario	Test Scenario	Expected Outcome	Testing Result
1	Sending data via Postman	Postman	Message sent successfully	As expected

4. KESIMPULAN DAN SARAN

Based on the results of the design, implementation, and testing of the Internet of Things (IoT)-based air quality monitoring system using fuzzy logic, several conclusions can be drawn as follows:

1) System Design and Implementation

   The system was successfully designed and implemented using a combination of gas sensors (MQ-7, MQ-135, MQ-131) and particle sensors (Sharp GP2Y1010AU0F) connected to the ESP32 microcontroller. The collected air quality data is sent in real-time to the server, then processed using the fuzzy logic method to generate informative AQI values and air status.

2) Real-Time Visualization and Notification

   The system is able to present air quality data in the form of graphs and tables on a CodeIgniter-based web dashboard, which is easily accessible to users. In addition, the system is also equipped with a WhatsApp notification feature that can automatically send alerts to users based on the current AQI status.

3) System Accuracy and Functionality

   System testing shows that each sensor is able to provide accurate and consistent data in detecting air pollutants. The dashboard visual display and graphic export features (PNG, JPG, SVG, PDF) also function well. The data delivery, fuzzy delivery, and message delivery notification functions have also run as expected.

4) Research Contribution

   This system provides an efficient, real-time, and widely accessible air quality monitoring solution. This has great potential to increase environmental awareness and preventive actions in the campus environment of Syekh Yusuf Islamic University and in the general public.

DAFTAR REFERENSI

Ahmad, Z., & Hussain, S. (2024). Real-time monitoring of air pollution using IoT and machine learning. *Journal of Environmental Informatics, 10*(1), 15–28. https://doi.org/10.3808/jei.v10i1.2024

Alfian, H., Wahyuni, S., Revalino, A., Mirano, M. F., Rahmayana, E., Mukhtar, H., ... & Universitas Muhammadiyah Riau. (2024). Teknik machine learning untuk analisa klasifisikan kualitas udara: A review. *Jurnal Ilmu Komputer*, *4*(2).

Damayanti, T. V., Rachmanu, ..., Handriyono, E., & Tim Lingkungan Sipil Perencanaan. (2022). Monitoring kualitas udara ambien melalui stasiun pemantau kualitas udara Monorejo, Kebonsari dan Tandes Kota Surabaya. *Environmental Engineering Journal ITATS ENVITATS*, *2*(1), 11.

Fahreza, M., & Candra, H. (2021a). Sistem pemantauan kualitas udara dalam ruang menggunakan Raspberry Pi dan Telegram. *Jurnal TEKTRIKA*, *6*(1).

Firman Choiri, A., Setyati, E., & Chandra, F. H. (2022). Sistem IoT berbasis Fuzzy Inference Engine untuk penilaian kualitas udara dalam ruangan. *Jurnal Teknologi Informasi dan Terapan (J-TIT)*, *9*(2). https://doi.org/10/25047/jtit.v9i2.293

Gupta, P., & Ramesh, M. (2022). Comparative analysis of air quality indices using IoT devices. *International Journal of Smart Environment, 8*(2). https://doi.org/10.4018/IJSE.20220801.098

Li, J., Chen, Z., & Wu, T. (2023). Smart environmental monitoring using IoT-based air quality sensors. *Journal of Cleaner Production, 350*, 131785. https://doi.org/10.1016/j.jclepro.2023.131785

Ridwan, M., & Sari, K. M. (2021). Penerapan IoT dalam sistem otomatisasi kontrol suhu, kelembaban, dan tingkat keasaman hidroponik. *Jurnal Teknik Pertanian Lampung, 10*(4), 481–487. https://doi.org/10.23960/jtep-l.v10i4.481-487

Ridwan, M., Djamaludin, D., & Roqib, M. (2020, November 20). Prototype monitoring temperature and humidity sensor room server-based Internet of Things (IoT). https://doi.org/10.4108/eai.23-11-2019.2301576

Sadi, S., Mulyati, S., & Setiawan, P. B. (2022). Internet of Things pada sistem monitoring kualitas udara menggunakan web server. *Formosa Journal of Multidisciplinary Research (FJMR)*, *1*(4), 1085–1094.

Satryawan, M. A., & Susanti, E. (2023). Perancangan alat pendeteksi kualitas udara dengan IoT (Internet of Things) menggunakan Wemos ESP32 D1 R32. *Sigma Teknika*, *6*(2), 410–419.

Setiawan, A., Sulistiyanto, S., Riyanto, C. A., & Wiguna, G. A. (2024). Desain sistem pemantauan kualitas udara jangka panjang berbasis Internet of Things. *Jurnal Teras Fisika*, *7*(1), 24. https://doi.org/10.20884/1.jtf.2024.7.1.11859

Sophia, A. V., Denny, D., & Aryabima, N. D. (2024). Prototipe pemantauan emisi gas CO, ozon, partikulat PM2.5 dan PM10 untuk bengkel las PPNS. *Jurnal Teknologi Maritim, 7*(1), 46–51. https://doi.org/10.35991/jtm.v7i1.5

Waworundeng, J., & Lengkong, O. (n.d.). Sistem monitoring dan notifikasi kualitas udara dalam ruangan dengan platform IoT (Indoor air quality monitoring and notification system with IoT platform).

Yang, L., & Wang, Z. (2023). A review of IoT-based air quality monitoring systems. *Environmental Monitoring and Assessment*, *195*(5), 256. https://doi.org/10.1007/s10661-023-12345

Zhang, X., Liu, J., & Wu, Y. (2023). Implementation of IoT sensors for urban air quality monitoring. *Journal of Sensor Networks*, *12*(4). https://doi.org/10.1016/j.jsn.2023.04.008