

Design of an Internet Of Things-Based Air Quality Monitoring Tool Using Microcontroller and MQ-135 Sensor

Akbar Waqis Tianto¹, Raina Putri Ramandha², Reikhan Firdaus³, Zidan Inov Firmansyah⁴, Muhammad Ridwan Arif Cahyono⁵

^{1,2,3,4,5} Department of Electronics Engineering, Gajah Tunggal Polytechnic, Tangerang, Indonesia

¹ email: akbarwaqist07@gmail.com

[submitted: 07-01-2025 | reviewed: 16-10-2025 | published: 31-10-2025]

ABSTRACT: Poor air quality caused by industrial pollution poses a significant challenge for the Politeknik Gajah Tunggal campus, located in Tangerang's industrial area. To address this issue, this study developed an Internet of Things (IoT)-based air quality monitoring device utilizing an Arduino Uno microcontroller, ESP-8266, and MQ-135 sensor. The device detects various pollutants, including CO, CO₂, and other harmful gases, with real-time data displayed through a Kodular-based application. Tests conducted under various air conditions demonstrated the device's high accuracy in pollutant detection. Indoor air tests recorded an average of 42 ppm (good category), while outdoor air tests averaged 48 ppm (good category). Exposure to cigarette lighter gas produced a maximum value of 880 ppm (severe category), and burning paper smoke yielded an average of 295 ppm (poor category). Consistency between the LED P10 display and the Kodular application results reached 100%. Overall, the device delivers quick and accurate air quality information, enhancing user awareness by up to 90%, as indicated by post-implementation surveys.

KEYWORDS: ESP-8266, Internet of Things, Microcontroller, MQ-135 Sensor, Unified Modeling Language.

I. INTRODUCTION

As the human population increases, the amount of waste that pollutes the air also increases, causing pollutants in the air to increase. This happens in the city of Tangerang, especially the Gajah Tunggal Polytechnic Tangerang campus which is located in the middle of an industrial area. The large number of production activities in industrial factories makes many power plants operate, resulting in air pollution. This is called air pollution because it contains pollutants and is added to the smoke produced by factories which can greatly disturb human health and also pollute the surrounding environment.[1]

Based on data from IQAir, the air quality in the city of Tangerang is getting worse every day, which means that this cannot be avoided and still happens, and all that can be done is to prevent polluted air from being inhaled and disturbing the health of students and other campus residents. An air quality monitor is a tool that functions to measure whether the air quality is good or bad. The absence of air quality monitors makes students and campus residents unaware of this and rarely take preventive measures against air pollution such as wearing masks. Thus, the air quality monitoring tool can overcome this by providing information about air quality so that it can increase the awareness of students and campus residents to protect themselves from exposure to pollution.

Various previous studies have developed air quality monitoring tools based on *Internet of Things* (IoT) using microcontroller technology and gas sensors. In a study conducted by Rodhotul Muttaqin, an IoT

system was designed to monitor temperature, humidity, and harmful gases using DHT11 and MQ135 sensors. The data is displayed *real-time* through the *Blynk* application with accurate results[2]. In another study, Reza Ramadhan designed an IoT system to monitor air quality with NodeMCU and MQ-135 sensor, displaying *real-time* data on LCD and Thingspeak platform[3]. Meanwhile, Muhammad Abdurahman Alfari and Tiara Yasmin Arimbi developed an IoT-based air quality monitoring device using ESP32, MQ sensors, and DHT22 that functions to detect various harmful gases, temperature, and humidity, and displays data in *real-time* through the Node-RED platform for efficient environmental monitoring[4]. Muhammad Firly Akbar developed an air quality monitoring device in the hall using the MQ-135 sensor and ESP8266 which is used to detect harmful gases and display the results on the LCD and *Blynk* application in *real-time*[5]. Research conducted by Grace C. Rumampuk, et al. designed an IoT-based indoor air quality monitoring system that functions to detect and transmit *real-time* air quality data to a special platform.[6]

In addition, in research conducted by Muhammad Syahputra Novelan, namely designing an air quality monitoring tool using a microcontroller that utilizes an MQ-135 sensor to detect carbon monoxide (CO) levels and an LM35 sensor to measure temperature[7]. The results of the conducted by Muhammad Guntur Salasa, et al. show that microcontrollers and TGS-2442 gas sensors are used to transmit *real-time* air quality data on air pollution monitoring devices[8]. Meanwhile, Nogar Silitonga, et al. developed a monitoring system for LPG,

CO₂, and cigarette smoke levels indoors that uses an MQ-2 sensor to detect pollutants and an ESP8266 module for wireless data transmission[9]. Then there is also another research conducted by Egi Badar Sambani, et al. who developed a system to detect cigarette smoke in the room. This system uses the Wemos D1 ESP8266 microcontroller and MQ-135 sensor to detect cigarette smoke[10]. M. Sadeli Amli, et al. designed an accurate and *real-time* air quality monitoring system that uses electrochemical sensors (CO-B4 and SO₂-B4), Arduino microcontroller, and GSM/GPRS module to send data to MySQL *database*. The measurement results are displayed through a web-based application in the form of ISPU (Air Pollution Standard Index) .[11]

In another study, Rachmad Nur Ariefin and Suryanto developed an automated system to monitor the condition of chicken coops in *real-time* using NodeMCU ESP8266 as a microcontroller and IoT technology to monitor temperature, air quality, and cage cleanliness[12]. Khodijah Amiroh, et al. conducted *real-time* monitoring of CO and CO₂ gas to assess hospital sanitation according to Kepmenkes 1204 of 2004 which was carried out in patient rooms, parking areas, and waiting rooms[13]. In another study, Suti Kurnia Dewi, et al. developed a temperature and humidity control system in a swallow building using a Wemos D1 Mini microcontroller, DHT11 sensors for temperature and humidity, and MQ-135 sensors for air quality[14]. Meanwhile in another study, Masruhi, et al. designed a low-cost air quality monitoring device using experimental methods, this system is based on Arduino UNO and is equipped with *Grove Air Quality* and Adafruit SHT31-D sensors to measure air quality, temperature, and humidity[15]. In addition, Jacqueline M.S, et al. in their research also designed a fire detection system that can provide early warning using smoke sensors, fire sensors, microcontrollers, LEDs, and *alarm buzzers*, and integrated with the *Blynk IoT platform* .[16]

Based on some of these studies, in this research, an *Internet of Things-based* air quality monitoring device using a microcontroller and an MQ-135 sensor was designed. Not only that, this tool can also be connected to a codular-based application that is easily *installed* by anyone. This aims to make it easier for students and campus residents to find out the air quality in the campus environment, and increase awareness to use masks as a preventive measure for air pollution.

II. RESEARCH METHODS

In this research, the method used is *Research And Development* (R&D). The *Research and Development* (R&D) method is a method that can be used to produce a certain product and test the reliability of the product. The stages used in the *Research and Development* (R&D) research design include, the first stage is looking for potential and problems. In this research, the

problem found was the absence of an air quality monitoring device on the Gajah Tunggal Polytechnic campus that could build awareness of students and campus residents to take precautions (washing hands and using masks) against air pollution that can interfere with health. The second stage is data collection. At this stage, data collection is carried out by making observations and digging up references from previous research journals. Then in the third stage, several designs and designs are carried out such as *software* design, *hardware* design, tool design, codular design, and also UML design. The next stage is the fourth stage, namely the product field test, where the product will be tested to find out failures that may occur and find out what to improve. After that, the next stage is product revision, where this is the stage of repair or revision of the errors found. Finally, the application stage of the tool, at this stage the application of the product is carried out on condition that the tool can function properly.

A. CONTROL SYSTEM DESIGN AND WIRING DIAGRAM

In this research, the control system and *wiring diagram* are designed with various components such as *5 Volt Power Supply*, Arduino UNO, ESP-8266, MQ-135 Sensor, P-10 LED, *database*, and android application. The design of the control system and wiring diagram is shown in Fig. 1 and Fig. 2.

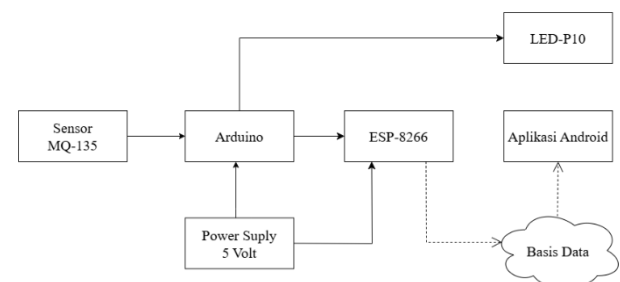


Fig. 1 Control System Design

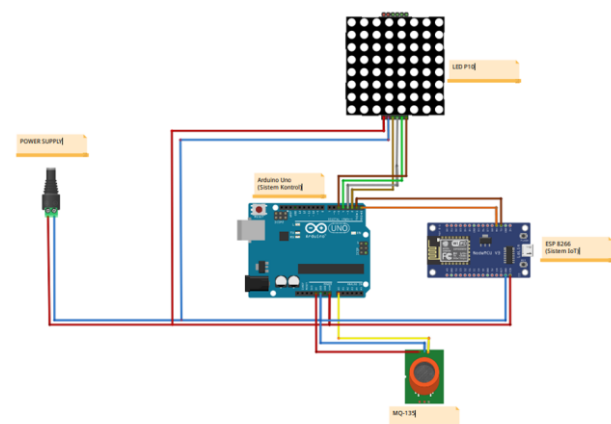


Fig. 2 Wiring Diagram

Fig.1 and Fig.2 show that the *Power Supply* sends voltage to the Arduino UNO and ESP-8266. The Arduino takes the data from the Sensor and sends it to the ESP-8266 and displays it on the P-10 LED. The

data received by the ESP is then sent to the *firebase database* which is then displayed in *real-time* on the android application.

B. TOOL DESIGN (ARDUINO, ESP-8266, MQ-135 SENSOR, LED-P10)

In this study, a tool design was made with various tool components with functions and specifications that are in accordance with the needs. This design aims to make the results of the designed tool in accordance with what is desired. The design of the tool design is shown in Figure 3.



Fig. 3 Tool Design

C. UML DESIGN DRAFTING

At this stage, the design of Android applications is carried out using the *Unified Modeling Language* (UML) method. The diagrams used include *use case diagram*, *activity diagram*, *sequence diagram*, and *class diagram*. Each diagram has a function in visualizing various aspects of the system, such as *use case* diagrams used to describe interactions between users and the system, *activity* diagrams used to show the flow of activities, *sequence* diagrams used to map the *sequence* of communication between components, and also *class* diagrams used to clarify the structure of classes and relationships between objects in the system.

D. APPLICATION DESIGN USING KODULAR

At this stage, an application is made to facilitate users in monitoring air quality. The application system used is Kodular. The application display design is shown in Figure 4.

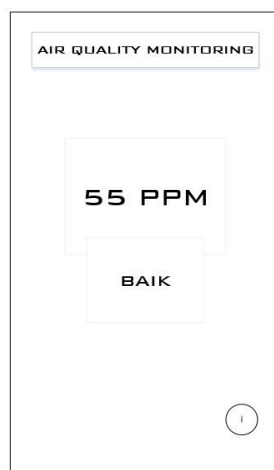


Fig. 4 App Display Design

Fig.4 shows the design of the codular-based application created by customizing some of the features to be displayed. Based on its understanding, kodular is a web-based *platform* designed to facilitate the creation of applications on Android devices. Therefore, the application can facilitate users both in *installing* and operating the application.

E. PRODUCT FIELD TEST

In the field test stage of the product to find the oddities or failures that occur from the product that has been made. The goal is to determine how the product can be improved and also how the tool works if there is a failure of the product's working system.

F. PRODUCT REVISION

At the product revision stage, improvements or improvements are made to the product that has been made, so that later the product that has been made can be operated with no damage from the work system. And also to obtain existing data whether there is progress from the previous tool or there are shortcomings from the alata that has been made.

G. PRODUCT APPLICATION

At this stage, the product that has been developed is applied in actual conditions to ensure that the system runs according to the initial design and functions properly. This test aims to ensure that there is no more damage or failure in the working system of the tool after going through the process of improvement and refinement.

III. RESULTS AND DISCUSSION

The result of this research is an *Internet of Things* (IoT)-based air quality monitoring device using Arduino Uno, ESP-8266, and MQ-135 sensors. This tool is designed to monitor air quality conditions in the Gajah Tunggal Polytechnic campus environment which is located in an industrial area with high levels of air pollution. *Real-time* air quality information is provided to users, increasing awareness to take preventive measures such as the use of masks. The results of the component circuit in this tool are shown in Figure 5.

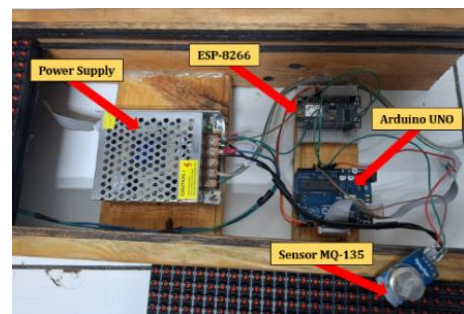


Fig. 5 Component Set

Fig. 5 shows the components used in the tool assembled in such a way by using cables as a connector.

The components contained in the circuit include a 5V Power Supply, Arduino UNO, ESP-8266, and MQ-135 Sensor. The output of the series of components is then connected to 2 P-10 LEDs which will display the results of the air quality monitored by the sensor. The circuit is neatly and simply arranged in a block-shaped wooden case. The selection of this wooden *case* aims to make the tool stronger and sturdier and show a good visual appearance. In addition, the simple size of the tool makes it easier for the tool to be placed anywhere. The results of the tool that has been made are shown in Figure 6.



Fig. 66 Tool Results

Fig. 6 shows the results of the device that has been successfully made after completing the previous stages. This tool can monitor air quality by using an MQ-135 sensor that can detect several air categories such as NH₃, NO_x, alcohol, *benzol*, smoke (CO), CO₂, and others. The data displayed on the tool is *real-time* data coming from the sensor. The data is also displayed in *real-time* in the codular application created to make it easier for users to monitor air quality when they are in locations that are not affordable to see air quality monitoring devices

Before creating a codular application, you must first design an application with a UML (*Unified Modeling Language*) diagram to describe in detail what users can access in the application that has been created. UML (*Unified Modeling Language*) is a visual language used to design and explain how a system or program works. In this study, 4 diagrams were made, namely, *use case diagram*, *activity diagram*, *sequence diagram*, and *class diagram*.

A. USE CASE DIAGRAM

Use Case Diagram is a UML (*Unified Modeling Language*) diagram used to describe the interaction between users (actors) and the system. This diagram shows the main functions that can be performed by *users* in the system, thus providing an overview of what features are provided by the system. In this research, a *use case diagram* is created which is shown in Figure 7.

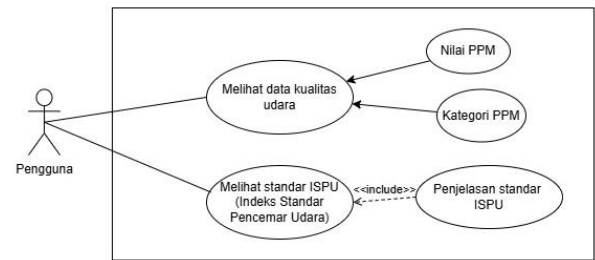


Fig 77 . Use Case Diagram

As the diagram in Fig.7 illustrates, users have access to view air quality data as well as view the ISPU (Air Pollution Standard Index) standards. The air quality data itself includes PPM values and PPM categories displayed in *real-time* from the sensor. The PPM value retrieved from the sensor through the *firebase database* is then analyzed based on ISPU to determine the category of PPM. ISPU standards can be accessed and viewed by users so that they can analyze air quality categories of good, fair, cloudy, poor, and severe.

B. ACTIVITY DIAGRAM

Activity Diagram is a UML diagram used to model *workflow* or business processes in a system. This diagram describes the steps of the activities performed, including initial conditions, transitions between activities, and end conditions. In this research, an *activity diagram* is made which is shown in Figure 8.

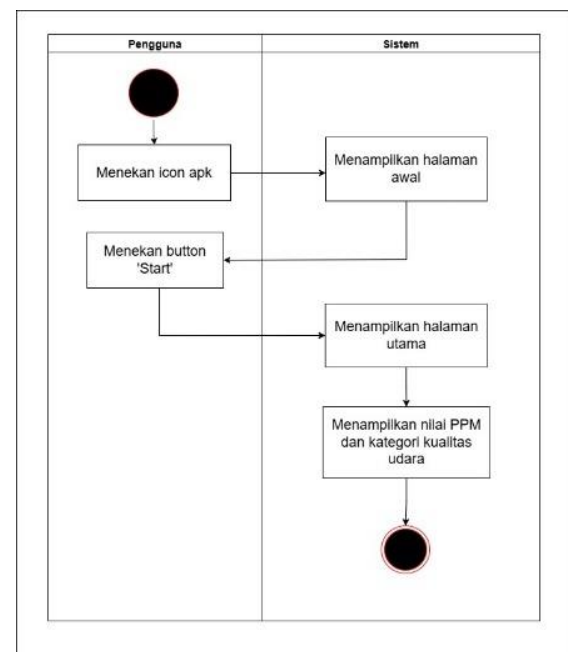


Fig. 88 Activity Diagram

The *Activity Diagram* in Fig.8 explains the work process of the air quality monitoring application that involves interaction between the user and the system. The process starts when the user presses the application *icon*, then the system displays the home page. Next, the user presses the "start" button on the home page so that

the system will display the main page. Then on the main page, the system displays PPM values and categories that can be seen directly by the user. At this stage, the system retrieves *real-time* air quality data from connected sensors. The data is then processed by the system to be analyzed and categorized into three classifications: good, fair, cloudy, poor, and severe. With limits of good category (0-50), fair category (51-100), cloudy category (101-200), bad category (201-300), and severe category (>300).

C. SEQUENCE DIAGRAM

Sequence Diagram is a UML diagram that models the time flow of interactions between objects or components in a system. This diagram illustrates how messages are sent between objects in order of time to complete a process. In this research, a *sequence diagram* is created which is shown in Figure 9.

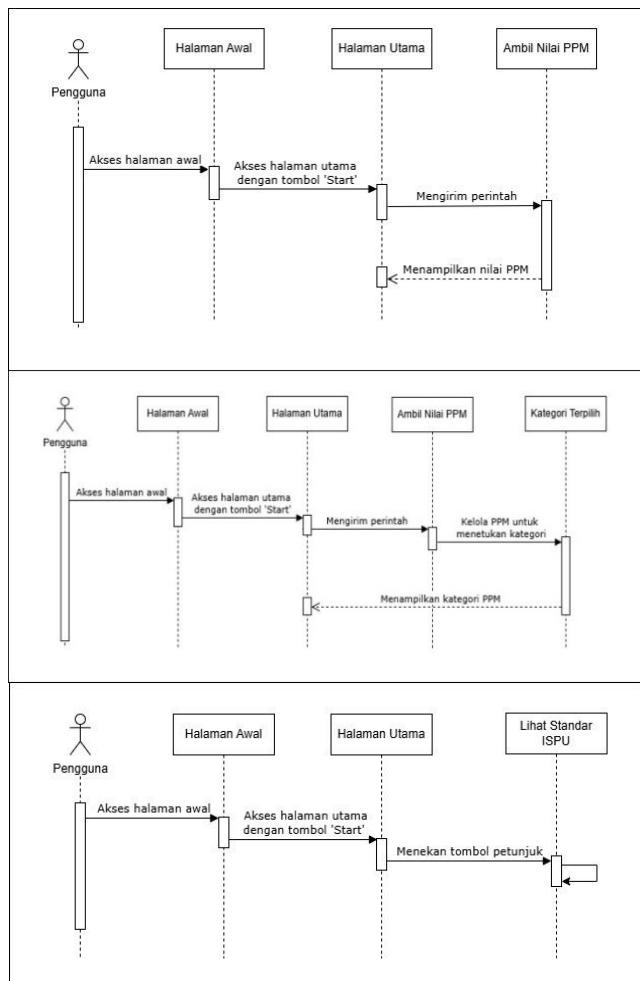


Fig 9. Sequence Diagram

The diagram in Fig.9 illustrates the flow that users do to be able to access the features of contained in the application. There are three *sequence diagrams* made according to the access that users can do in the air quality monitoring application. Access that users can do is to view PPM values, view PPM categories, and standard provisions of the ISPU (Air Pollution

Standard Index). In the first *sequence diagram*, it starts with the user opening the home page and then pressing the "start" button to open the main page. Then send a command to retrieve data from the *firebase database* and display the PPM value on the main page. *Real-time* PPM values can be seen directly by users after accessing the main page. Then, in the second *sequence diagram*, it starts with the user opening the home page then pressing the "start" button to open the main page. Then, send a command to retrieve data from the *firebase database*. Then, the PPM value data is analyzed for its category according to the category standards on ISPU and then will be displayed on the main page. Categories that appear on the main page include good, fair, cloudy, bad, and severe. Then the third *sequence diagram* starts from the user opening the home page then pressing the "start" button to open the main page. Then, the user presses the instruction button which then directs to a page containing the ISPU table. This feature is used to provide information related to actual and *valid* category standards that can be accounted for. By using a *sequence diagram*, designing the application will be easier and what can be accessed in the application can be described properly.

D. CLASS DIAGRAM

Class Diagram is a UML diagram that models the static structure of a system, including the classes, attributes, methods, and relationships between them. This diagram shows how data is organized in the system and how classes interact with each other, either through association, aggregation, composition, or generalization. In this research, the *class diagram* shown in Figure 10 was created.

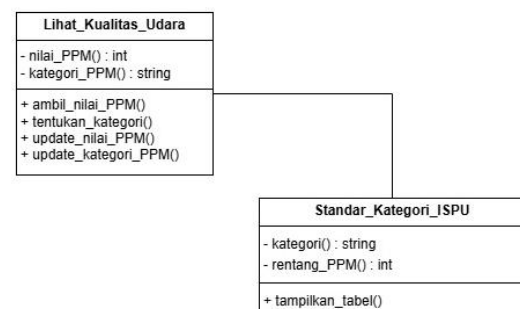


Fig. 10 Class Diagram

The diagram in Fig. 10 describes the access that can be done by users in the form of a class diagram. The *class diagram* itself includes the functions, attributes of the functions and also the activities performed by the functions. The system works in an integrated manner to provide *real-time* air quality information to users. The square-shaped *class diagram* column is divided by using lines into 3 parts. At the top there are functions that can be performed by the application system. The function can be a reference or

theme in making other parts. In the middle of the column there are attributes. Attributes are things that support the running of the function and the type of data used. Then at the bottom of the column there are methods that cover activities that can be done by the function. These activities can include retrieving data, displaying data, categorizing data, and others. The function column in the *class diagram* can be made more than one column. In the last stage, the columns are connected using a line called an association line which is a line that symbolizes that a function in the *class diagram* is interconnected or still in the same system

After the UML design is complete, a codular application is made as an implementation of the system design that has been designed. The codular application aims to make it easier for users to monitor air quality in *real-time* and can be seen at any time.

E. HOME PAGE DISPLAY

The start page of the application displays an image of the opening *icon* and also a "start" button. The display of the start page is shown in Figure 11.



Fig. 11 Kodular App Home View

The "start" button on the home page shown in Fig.11 leads to the main page containing the air quality data display.

F. MAIN PAGE DISPLAY

The main page is accessed when the user presses the "start" button presented on the home page. The main page view is shown in Fig 12.

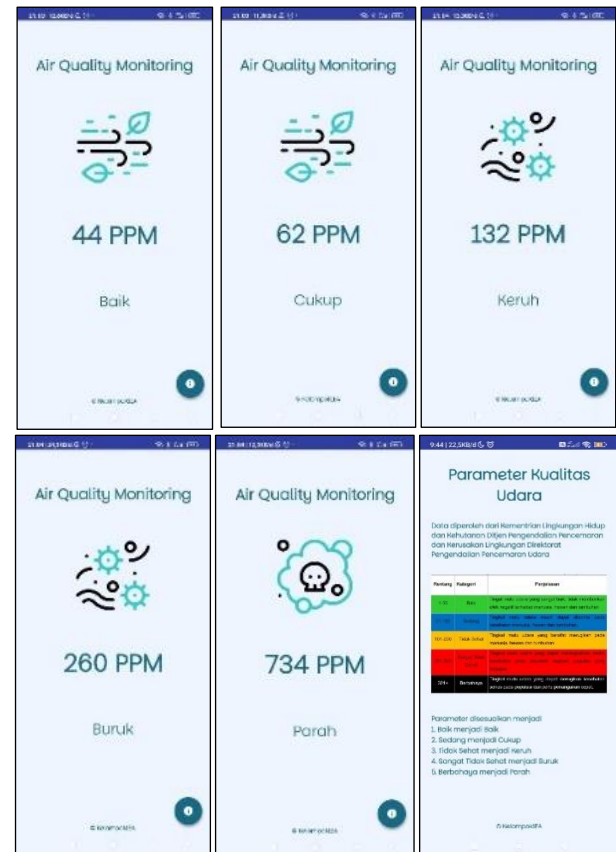


Fig. 12 Display of Air Quality in Different Categories

As can be seen from Fig.12, the main page contains PPM values, PPM categories, and air quality parameter data. The PPM value data displayed corresponds to the air quality data captured by the sensors on the device. The data is taken from the *firebase database* integrated with the device. After the PPM value is obtained, the application system will categorize the PPM value according to the ISPU (Air Pollution Standard Index) standard. These settings have been programmed when the application system was created. Air quality parameter data based on ISPU is also displayed in this application. However, the data is displayed on a different page after the user presses the instruction button on the main page. The data is presented clearly in tabular form along with its description. This aims to make it easier for users to get information about the limits of good and bad air quality. Back on the main page, the air quality value data displayed on the application matches the value data displayed on the device. For this reason, a test is made to determine the synchronization between the tool and the application. The synchronization between the device and the android application is shown in Figure 13.



Fig. 13 Tools and applications are interconnected in real-time

G. TOOL TESTING

In this study, the tool testing was carried out using various types of pollutants that could affect the sensitivity of the MQ-135 sensor used in the tool. In this test, a valid and accountable standard is needed to determine the air quality category detected by the sensor. For this reason, this test uses the ISPU (Air Pollutant Standard Index) which is a standard calculated by scientific calculation methods and refers to the Regulation of the Minister of Environment No. 45 of 1997[5]. The categories in ISPU are shown in Table 1.

Tbl. 1 Air Pollutant Standard Index

Category	Index	Description
Good	0-50	Safe for humans, animals and plants.
Simply	51-100	Harmful to sensitive plants but fairly safe for humans.
Cloudy	101-200	Harmful to humans and sensitive animals and can cause damage to living plants.
Bad	201-300	Harm to the health of a number of exposed humans.
Severe	> 301	It can cause serious harm to human health.

This test is carried out using several pollutants that affect the sensitivity of the sensor. With reference to the ISPU standard in Tbl.1, testing of the device is carried out using several pollutants including testing the device against match gas, testing the device against paper burning smoke, testing the device against outdoor air, and testing the device against indoor air. Tool testing on pollutants is shown in Fig. 14, Fig. 15, Fig. 16, and Fig. 17.

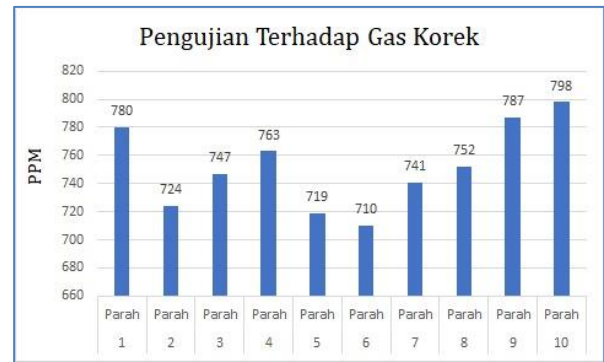


Fig. 14 Testing the tool against lighter gas

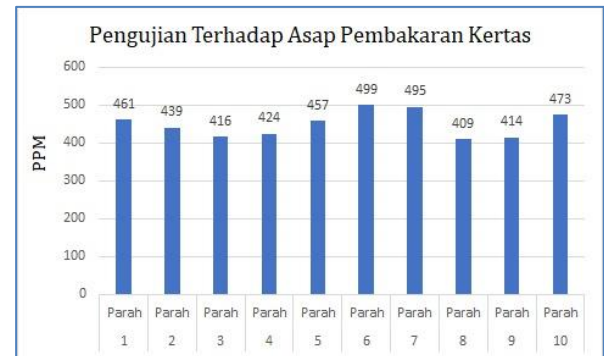


Fig. 15 Testing the appliance for paper burning smoke

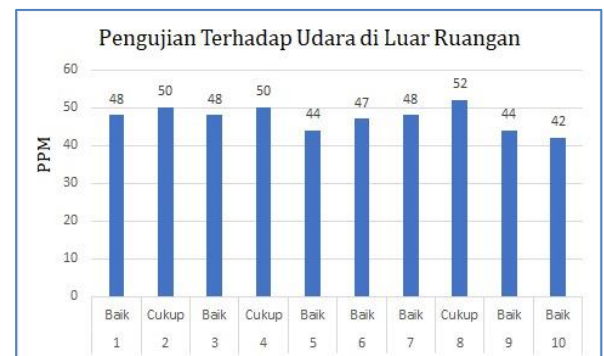


Fig. 16 Testing the Tool against Outdoor Air

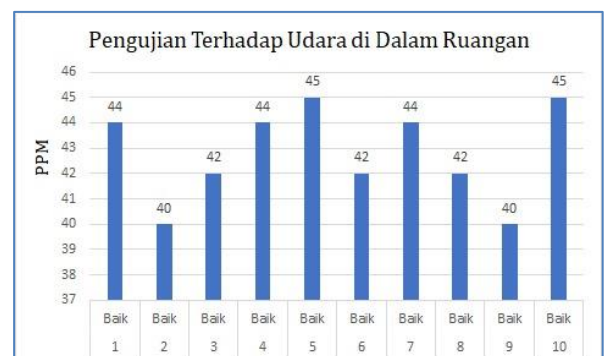


Fig. 17 Testing the Tool against Indoor Air

Fig. 14 shows the results of the device's tests on lighter gas containing butane or propane compounds. Based on the test results, the detected gas concentrations range from 710 to 798 PPM, all of which fall into the "severe" category. This category

indicates that the gas concentration is at a very dangerous level and can cause serious health problems if exposed for a long time. These results show that the device has a very good level of sensitivity to lighter gas.

Furthermore, Fig.15 shows the results of testing the device for smoke from burning paper. Burning paper produces various gaseous compounds, including carbon monoxide (CO) and carbon dioxide (CO₂). From the test results, the gas concentrations detected by the device ranged from 409 to 499 PPM, all of which fell into the "severe" category. This shows that paper burning fumes pose a threat to human health, especially if exposed in a closed room without ventilation. The success of the device in detecting high gas concentrations in this test proves that the sensor is able to work optimally in conditions filled with particles and gas compounds from combustion. This shows that the device has a very good level of sensitivity to paper burning smoke.

Then in Fig.16, the results of testing the device on outdoor air are recorded. The test results showed that the detected gas concentration ranged from 42 to

In this research, air quality PPM is not only displayed on the device but also on a codular-based application. This application can only be *installed* by Android phones. The success in *installing* the application is shown in Tbl 2.

Tbl. 2 Installation on several Android phones

No.	HP Type	Description
1.	Redmi Note 8 Pro	Successful
2.	Samsung A55 5G	Successful
3.	Samsung M23 5G	Successful
4.	Itel RS4	Successful
5.	Realme 12	Successful
6.	Poco M3	Successful
7.	Vivo Y12	Successful

The results Tbl.2 show that the application can be *installed* on various Android phones. This is an advantage of the application because not all applications can be *installed* easily on several Android phones. This affects the ease of users in monitoring the air quality around them.

But despite these advantages, the alignment between the device and application cannot be coordinated properly due to signal constraints. This greatly affects the alignment of the two even though the data will be aligned in a few moments. Data on the *delay* between the device and the application is shown in Tbl 3

52 PPM. Most of the results fell into the "good" category, although a few times the sensor recorded the "fair" category. The "good" category indicates that the gas concentration is at a level that is safe to breathe, while the "fair" category indicates a slight increase in air pollution, although still within tolerance limits. These results indicate that the outdoor air is relatively clean and not contaminated by significant amounts of harmful gases. Small numbers in PPM values can be caused by environmental factors, such as motor vehicle activity or combustion around the test site. The monitoring device's sensors proved capable of detecting small differences in gas concentrations, demonstrating a good level of accuracy.

Finally, Fig.17 shows the results of testing the device on the air in the room. The test results show that the detected gas concentrations are very low, with PPM values ranging from 40 to 45. All results are in the "good" category, indicating that the air in the room is very clean and safe to breathe. The device proves that the sensor has high sensitivity even in clean air conditions.

Tbl. 3 Tool and application *delay* range

No.	Sensor value		Delay
	LED P10	Application	
1.	48 PPM	50 PPM	0.72 s
2.	50 PPM	48 PPM	0.53 s
3.	48 PPM	50 PPM	0.70 s
4.	50 PPM	48 PPM	0.64 s
5.	44 PPM	44 PPM	0.13 s
6.	47 PPM	47 PPM	0.15 s
7.	48 PPM	48 PPM	0.18 s
8.	52 PPM	48 PPM	0.62 s
9.	44 PPM	52 PPM	0.58 s
10.	42 PPM	42 PPM	0.19 s

Tbl. 3 shows the results of testing 10 times along with the calculated *delay* time between the device and the application. From the results that have been recorded, some tests have a high *delay* time. This is due to poor signal and inhibits the alignment of the results displayed on the tool and application. Even so, this does not affect the performance of the system in the application, both in data and additional features. Testing the features of the application is shown in Table 4.

Tbl. 4 Black box testing

No.	Test Description	Expected results	Results Test	Ket
-----	------------------	------------------	--------------	-----

1.	Pressing the <i>start</i> button on the home page	The system will display the main page (ppm value, and category)	System display main page (ppm value, and category)	As per
2.	Pressing the info button on the main page	The system displays the ISPU information page	The system displays the information page of ISPU	As per
3.	Pressing the <i>back</i> button while on the home page	The system will close the application	The system closes the application	As per
4.	Pressing the <i>back</i> button while on the main page	The system will return to the home page	The system returns to the home page	As per

The results in Tbl. 4 show some testing of the features contained in the application. The first feature is the "*start*" button on the home page, after pressing the button successfully directs to the main page. Then the second feature is the ISPU (Air Pollution Standard Index) instruction button, after pressing the button successfully directs to the page where the ISPU (Air Pollution Standard Index) table is displayed. Furthermore, the third feature is the back button on the main page, after pressing the button successfully redirects back to the home page. And in the last feature, namely the back button on the home page, after pressing the button successfully closes the application. From some of these tests, it is stated that the features in the application can function properly as expected,

In the last stage, a suitability test was carried out to test the device against several pollutants. The suitability between the tool and the application is shown in Tbl. 5.

Tbl 5. Testing the suitability of the display on the P-10 LED and the application

No.	Sensor testing	LED P-10		Application		Ket.
		Sensor Value	Categor y	Sensor Value	Catego ry	
1.	Match gas	712 ppm	Severe	712 ppm	Severe	As per
2.	Paper burning smoke	296 ppm	Cloudy	296 ppm	Cloudy	As per
3.	Outdoor air	47 ppm	Good	47 ppm	Good	As per
4.	Air in the room	42 ppm	Good	42 ppm	Good	As per

IV. CONCLUSION

1. This study successfully designed an air quality monitoring device using Arduino Uno as the basis of its control system by connecting the MQ-135 sensor as a catcher of pollutant parameters in the air.
2. The design of an Internet of Things (IoT) system for air quality monitoring using ESP-8266 can be connected to the internet and firebase database.
3. UML design for air quality monitoring applications using use case diagrams, activity diagrams, sequence diagrams, class diagrams as a description of the system formed.
4. The design of an air quality monitoring application using codular as a software intermediary builds an application that is composed of block diagrams as a constituent.

ACKNOWLEDGMENTS

Thank you to all those who have helped in carrying out this research, especially Mr. Slamet Afandi ST., as a lecturer and supervisor from Gajah Tunggal Polytechnic who has directed and guided during the research. And also all related parties who have provided motivation in conducting this research.

REFERENCE

- [1] A. Masito, "Risk Assessment of Ambient Air Quality (NO₂ and SO₂) and the Respiratory Disorders to Communities in the Kalianak Area of Surabaya," *J. Kesehat. Environ.*, vol. 10, no. 4, p. 394, 2018, doi: 10.20473/jkl.v10i4.2018.394-401.
- [2] R. Muttaqin, W. S. W. Prayitno, N. E. Setyaningsih, and U. Nurbaiti, "Design of an Iot (Internet Of Things) Based Air Quality Monitoring System with DHT11 Sensor and MQ135 Sensor," *J. Lab Management. Educ.*, vol. 6, no. 2, pp. 102-115, 2024, doi: 10.14710/jlpl.6.2.102-115.
- [3] R. Ramadhan and J. C. Chandra, "Design of an Iot-Based Air Quality Monitoring System with Nodemcu," *Semin. National. Mhs. Fak. Technol. Inf. Jakarta-Indonesia*, vol. 1, no. 1, p. 1184, 2022.
- [4] P. Akhir and T. Syawal, "Bangka Belitung in 2020," 2020.
- [5] N. L. Mauliddiyah, "Utilization of MQ-135 SENSOR AS AIR QUALITY MONITORING IN THE FASILKOM GALLERY AULA," no. September, p. 6, 2021.
- [6] A. M. Rumagit *et al.*, "Design+of+System+Monitoring+Air+Quality+Indoor+Based+on+Internet+ of+ Things," pp. 1-8, 2018.

- [7] M. S. Novelan, "Indoor Air Quality Monitoring System Using Microcontroller and Android Application," *InfoTekJar J. Nas. Inform. and Technol. Jar.*, vol. 4, no. 2, pp. 50-54, 2020, [Online]. Available: <https://doi.org/10.30743/infotekjar.v4i2.2306>
- [8] M. G. Salasa, A. Rosadi, and N. Fahrani, "Design of Microcontroller-Based Air Pollution Monitoring Tool Using TGS-2442 Gas Sensor," *Comput. Insight J. Comput. Sci.*, vol. 3, no. 1, pp. 1-8, 2021, doi: 10.30651/ci:jcs.v3i1.9146.
- [9] N. Silitonga, Y. Telaumbanua, and H. G. Simanullang, "DEVELOPMENT OF IOT DEVICE FOR MONITORING AIR QUALITY IN A ROOM USING ANDROID-BASED MICROCONTROLLER," *METHOMIKA J. Manaj. Inform. and Computerized Account.*, vol. 5, no. 1, pp. 81-85, 2021, doi: 10.46880/jmika.vol5no1.pp81-85.
- [10] E. B. Sambani, D. Rohpandi, and F. A. Fauzi, "Monitoring System for Cigarette Smoke Detection Devices in Microcontroller-Based Rooms Using Mq-135 and Telegram," *e-Journal JUSITI (Journal of Inf. Systems and Technol. Information)*, vol. 10, no. 1, pp. 53-61, 2021, doi: 10.36774/jusiti.v10i1.820.
- [11] M. S. Amli, B. Yulianto, and N. Nugraha, "Design and Manufacture of Air Quality Measurement System Using Microcontroller," *J. Automation Control and Instrumentation*, vol. 7, no. 1, p. 1, 2015, doi: 10.5614/joki.2015.7.1.1.
- [12] S. Suryanto and R. Nur Ariefin, "Monitoring System for Air Quality, Temperature and Cleanliness of Automatic Chicken Cage Based on Internet of Things (IoT)," *IMTechno J. Ind. Manag. Technol.*, vol. 4, no. 2, pp. 117-123, 2023, doi: 10.31294/imtechno.v4i2.2150.
- [13] K. Amiroh, O. A. Permata, and F. Z. Rahmanti, "Air Quality Analysis for Hospital Environmental Health Monitoring," *InfoTekJar (Journal of Nas. Inform. and Technol. Network)*, vol. 4, no. 1, pp. 29-36, 2019, doi: 10.30743/infotekjar.v4i1.1549.
- [14] S. K. Dewi, R. D. Nyoto, and E. D. Marindani, "Design of Prototype Temperature and Humidity Control System in Swallow Building with Mobile-Based Microcontroller," *J. Education and Research. Inform.*, vol. 4, no. 1, p. 36, 2018, doi: 10.26418/jp.v4i1.24065.
- [15] M. Masruhi, M. Nakkir, and R. Efendi, "Design of a Low-Cost Air Quality Monitoring Tool Based on Arduino UNO Using Grove Air Quality Sensor and Adafruit SHT31-D," *J. Mekanova Mek...*, vol. 9, no. 2, 2023, [Online]. Available: <http://jurnal.utu.ac.id/jmekanova/article/view/8487>
- [16] J. M. S. Waworundeng, "Design of Smoke and Fire Detection System Based on Sensor, Microcontroller and IoT," *CogITO Smart J.*, vol. 6, no. 1, pp. 117-127, 2020, doi: 10.31154/cogito.v6i1.239.117-127.
- [17] H. Herlianus and G. Gunadi, "Development of Android-based Animal and Human Movement Organ Learning Media Using Kodular," *Inform. J. Comput. Science*, vol. 18, no. 1, p. 88, 2022, doi: 10.52958/iftk.v17i4.4605.

AUTHOR BIOGRAPHY AND CONTRIBUTIONS



Akbar Waqis Tianto, was born in East Kalimantan in February 2004. He is currently pursuing a Diploma 3 at Gajah Tunggal Polytechnic and is taking the Electronic Engineering study program.



Raina Putri Ramandha, was born in Tangerang in October 2004. Currently studying Diploma 3 at Gajah Tunggal Polytechnic and taking the Electronic Engineering study program.



Reikhan Firdaus, born in Pemalang in August 2003. He is currently pursuing a Diploma 3 at Gajah Tunggal Polytechnic and is taking the Electronic Engineering study program.



Zidan Inov Firmansyah, was born in Tangerang in November 2003. He is currently pursuing a Diploma 3 at Gajah Tunggal Polytechnic and is taking the Electronic Engineering study program.



Muhammad Ridwan Arif Cahyono, S.T., M.T., was born in Klaten in June 1992. He earned his Bachelor's degree (S-1) from the Department of Engineering Physics, Gadjah Mada University, in 3.7 years with a Coumlude predicate. He also earned his Master's degree (S-2) from the Department of Electrical Engineering, University of Indonesia, in 1.5 years with a Coumlude predicate. He is currently a permanent lecturer in the Electronic Engineering Study Program, Gajah Tunggal Polytechnic.