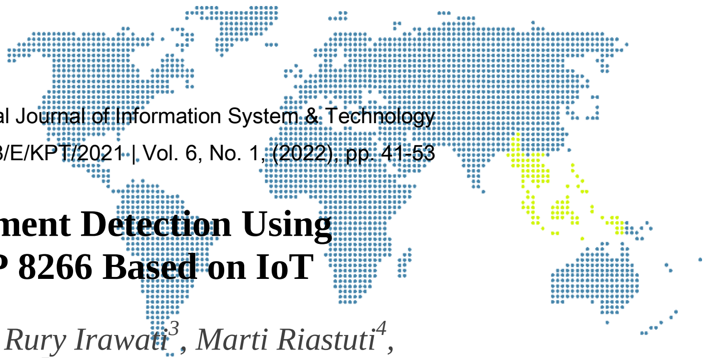


International Journal of Information System & Technology
Akreditasi No. 158/E/KPT/2021 | Vol. 6, No. 1, (2022), pp. 41-53

Design of Air Pollution Measurement Detection Using Microcontroller NodeMCU ESP 8266 Based on IoT

Yudi Irawan Chandra[1], *Fitri Sjafrina*[2], *Diyah Rury Irawati*[3], *Marti Riastuti*[4], *Winda Sari Br. Purba*[5]

[1,2,3,4,5]*STMIK Jakarta STI&K, South Jakarta, Indonesia*

[1]*yirawanc@gmail.com*, [2]*fitrisjafrina@gmail.com**, [3]*diyah.ruri@gmail.com*, [4]*tutimarti67@gmail.com*, [5]*winda@jak-stik.ac.id*

Abstract

The East Jakarta area has many residential areas, from the lower to the upper classes, and residential areas for businesses such as malls, markets, mini supermarkets, and protected tourist attractions, for example, the urban forest area. East Jakarta City is currently trying to maintain the excellent air quality in the area. Currently, the local government's prevention of poor air quality and air quality checks in East Jakarta is not going well because the budget requires a large amount of money. Therefore, a tool was made to detect air quality. The tool built in this study is based on the MQ-135, DHT-11, NodeMCU 8266 microcontroller, and Blynk application using the C++ language on the Arduino application. This tool is expected to provide information to users regarding the level of air quality in the area. This tool will automatically connect to an available wifi network. An air quality notification will immediately appear, along with the temperature and humidity values displayed on the smartphone. The tool made is expected to help view air quality. All information obtained can be helpful for follow-up by the government and the community to maintain the beauty and good air quality in the area.

Keywords: *air pollution, microcontroller, Internet of Things*

1. Introduction

The City of Jakarta, or the Special Capital Region of Jakarta (abbreviated DKI Jakarta), is the state capital and the largest city in Indonesia. Jakarta is located on the northwest coast of Java Island. Jakarta has an area of about 664.01 km² (ocean: 6,977.5 km²), based on data from the Directorate General of Population and Civil Registration (Dukcapil) of the Ministry of Home Affairs (Kemendagri), the population of DKI Jakarta is 11.25 million as of June 30, 2021. equivalent to 4.13% of the total population of Indonesia, which reached 272.23 million people. The East Jakarta City Forest has won the Adipura award for the best urban forest category for 2017-2018. This City Forest is located not far from the entrance gate of the Kopassus Command Headquarters complex and Cijantung Mall. This forest which has an area of approximately 17 hectares, is overgrown with various trees, such as Mahogany, Jati, Ketapang, Asam Landi, Trembesi, Angsana, Bungur, Flamboyant, and other trees that make the atmosphere very shady. A jogging track is also provided, so people who want to exercise can do it here. If you want to relax, this urban forest also provides several unreserved seats to find. The attractiveness of the East Jakarta City Forest has attracted the attention of the East Jakarta City Government to maintain the air quality in the area. However, currently, the government's prevention of poor air quality or air quality checks in East Jakarta is not going well or is being pursued by the local government because the budget is relatively large. Based on the description above, a prototype tool is designed with MQ-135 and DHT-11 sensors to measure air temperature, humidity, and air quality controlled by the NodeMCU ESP 8266 microcontroller through the c programming language to solve problems regarding poor air quality. The application built in this research is expected to function as a marker to

ISSN : 2580-7250

monitor air quality in the area specified by the user. So the system can provide information to users about the level of air quality in the area. The limitation of the problem in this research is a tool made using a DHT-11 sensor to detect temperature and MQ-135 to detect air quality and humidity; the data is displayed on the android application, and this tool uses the ESP8266 Wifi Module to send data via IoT (Internet of Things).

In previous research, Ramdan Satra (2010) developed a monitoring system for air pollution levels using the ZigBee protocol using a CO (Carbon Monoxide) gas sensor. This study uses the ZigBee protocol as a wireless transmission medium, then uses Arduino with an XBee socket and an MQ-9 gas sensor as a node station. The results of this study are a prototype of a carbon monoxide air pollution level monitoring system based on the ZigBee protocol. They have succeeded in sending the results of air pollution data retrieval from the CO gas sensor client node to the raspberry pi server [1]. Adam Faroqi (2017) designed a tool to detect pollution levels in the air that functions as a monitor for pollution levels and early detection of air pollution so that it becomes a reference for making programs to overcome the problem of air pollution. To detect levels of air pollution using the MQ-7 gas sensor, which is sensitive to carbon monoxide gas, is stable, and has a long life. Display using an LCD that was previously processed by the Arduino UNO microcontroller and display using a laptop previously processed by MATLAB software, which is transmitted using the HC-05 wireless technology found on the device so that the display on the LCD and laptop is the same. The system can provide a solution to the problem of air pollution because it is practical and easy to find, and the costs required are affordable compared to tools from environmental agencies. The design of an air pollution detector using the MQ-7 gas sensor with Wireless Technology HC-05 can detect carbon monoxide. The input from the gas sensor is processed via Arduino, produces output on the LCD and LED, and is sent via wireless HC-05 to the PC [2]. Tri Fidrian Arya (2018) conducted research for air quality monitoring which was applied to a wireless sensor network (WSN) by installing more than one sensor node at a specific location, and there was one sink that acted to collect data from the sensor node and send it to the server [3]. Several existing studies have not seen that an air quality detection device connected to the Internet is, therefore, a tool made using IoT connected to the Internet where the measurement results can be displayed on an Android-based smartphone.

2. Research Methodology

Before compiling a program on the Arduino Uno microcontroller, the first step that must be done is to compile a flowchart or flowchart that will be used as a reference for writing a sketch program on the Arduino microcontroller [4]. The flowchart in Figure 1 explains that this system will first initialize all WiFi devices that are active when searching for HIGH (1) APs, MQ-135 Sensors, and DHT-11 Sensors with HIGH (1) status on the Microcontroller. After the program runs, the NodeMCU ESP8266 V1 will connect to the AP (Access Point) that has been determined so that it can access the internet; if it fails to connect to the AP, it will continue to try to reconnect until it is successfully connected. If the AP is connected, the sensor sends the sensor's initial value (aqi, h, t) to the Microcontroller. Then the Microcontroller will calculate the value of the air based on the humidity, temperature, and quality of the air [5].

After the humidity (H), temperature (T), and quality (Q) of the air are obtained, the NodeMCU will send the value of the data that has been obtained to the application via pins V2, V3, V4. The application will read and display the value of these data in the android application. The Microcontroller will check whether the power is still on; if it is, the Microcontroller will continue to repeat the sensor readings to update the data periodically. Figure 1 shows the flowchart of the designed tool :

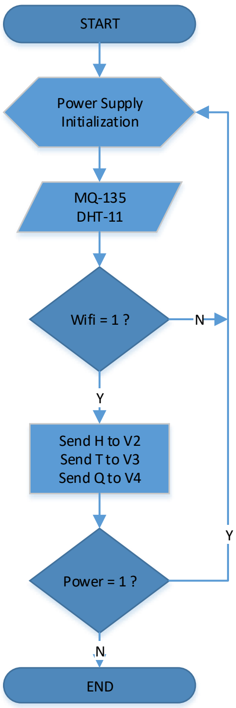


Figure 1. Flowchart of The Tool

Image Caption Figure 1:
H = Humidity
T = Temperature
Q = Quality (Air Quality Index)
V2 = Virtual 2
V3 = Virtual 3
V4 = Virtual 4

The flowchart above explains that this system will first initialize all devices consisting of active WiFi when looking for an AP with HIGH status (1), Sensor MQ-135 and Sensor DHT-11 with HIGH status (1) on the Microcontroller. After the program runs, the NodeMCU ESP8266 V1 will connect to the AP (Access Point) that has been determined so that it can access the internet; if it fails to connect to the AP, it will continue to try to reconnect until it is successfully connected. If the AP is connected, the sensor sends the sensor's initial value (aqi, h, t) to the Microcontroller. Then the Microcontroller will calculate the value of the air based on the humidity, temperature, and quality of the air. After the humidity (H), temperature (T), and quality (Q) of the air are obtained, the NodeMCU will send the value of the data that has been obtained to the application via pins V2, V3, V4. The application will read and display the value of these data in the android application. The Microcontroller will check whether the power is still on; if it is, the Microcontroller will continue to repeat the sensor readings to update the data periodically.

Figure 2 shows a flowchart of the application starting from the initial display of the application; if the data is humidity (H), temperature (T), and quality (Q) of the incoming air, the application will process the data and save it for display. If the sensor changes data, the data obtained will change once every 1 second and then be sent again to the Blynk application to be displayed again in the android application on a smartphone.[6] If the air quality value is less than 50, then the application will send a notification or notification to the smartphone that the air quality is good. If the air quality value is more than 50 and less than 100, then the application will send a notification or notification to the smartphone that the air quality is moderate. If the air quality value is more than 100, then the application will send a notification or notification to the smartphone that the air quality is poor. Otherwise, the app will continue to read the sensor and display it.

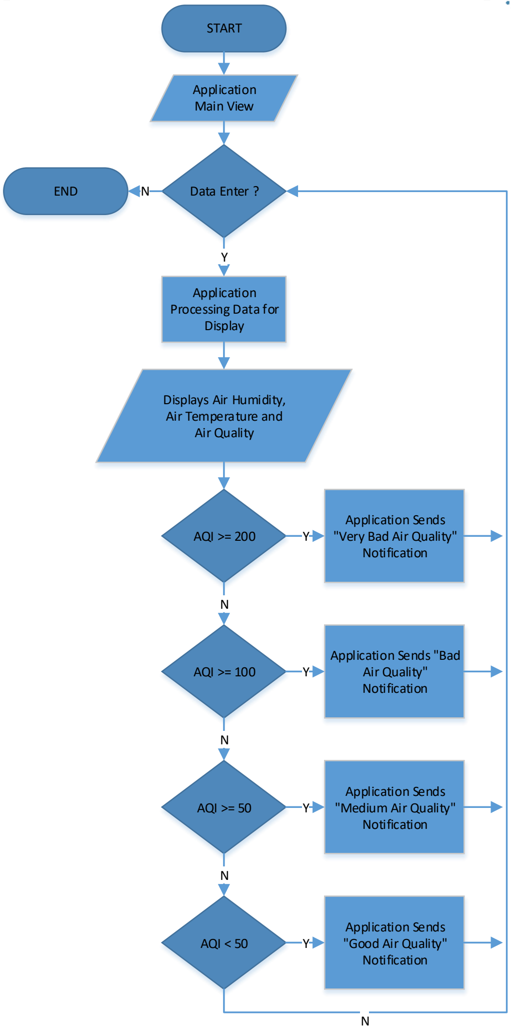

START
Application Main View
Data Enter ?
N
END
Y
Application Processing Data for Display
Displays Air Humidity, Air Temperature and Air Quality
AQI >= 200
Y
Application Sends "Very Bad Air Quality" Notification
N
AQI >= 100
Y
Application Sends "Bad Air Quality" Notification
N
AQI >= 50
Y
Application Sends "Medium Air Quality" Notification
N
AQI < 50
Y
Application Sends "Good Air Quality" Notification
N

Figure 2. Flowchart of The Circuit

The design and manufacture of tools are grouped into several sections. Circuit block diagrams describe input, process, and output blocks in general. The design of circuits and components describes the stages of tool building design, the components needed, their functions and the overall circuit. The working principle of the circuit contains a detailed description of the work process. The program flow diagram describes the tool's working steps in the form of a diagram. Making a microcontroller program contains the stages of software programming.

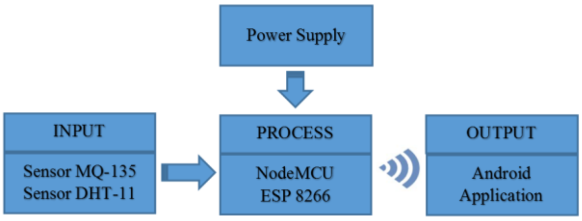


Figure 3. Block Diagram Circuit

The following is a description of the workings of each circuit block in the block diagram in Figure 3:

a) The MQ-135 gas sensor is a type of chemical sensor that is sensitive to NH3, NOx, alcohol, benzene, smoke (CO), CO2, and others. The way this sensor works is to accept changes in the resistance value (analogue) when exposed to a gas the data is sent to the NodeMCU ESP8266 [7].
b) The Digital Humidity and Temperature (DHT-11) sensor functions as an input sensor in reading data from the environment in temperature and humidity. The data obtained is sent to the NodeMCU ESP8266 [8].
c) NodeMCU ESP8266 functions to process input data given by reading data from the environment based on detection by the MQ-135 and DHT-11 sensors so that the output in the form of Air Quality, temperature and humidity are displayed on the application.
d) The application functions as an output to display the results of reading digital data from the MQ-135 and DHT-11 sensors that have been processed in the form of air quality in the form of Parts Per Million (ppm), temperature and humidity [9].

3. Results and Discussions

In this research, the IoT-based prototype was made by utilizing the NodeMCU ESP8266 V1, which is used to process data received by the MQ-135 and DHT-11 sensors. NodeMCU ESP8266 V3 is also used to control connected resources and components to perform data processing and processing. The Microcontroller can work by connecting it via the Universal Serial Bus (USB) port to obtain the required voltage to work of 3.3V. The DHT-11 sensor and the MQ-135 sensor are used as input media for the NodeMCU ESP8266 V1 so that when the sensor reads, the data will be sent to the Microcontroller, which is then processed and forwarded to blink so that the Android application can pull data from blink and display monitoring results on a smartphone.

3.1. DHT-11 Circuit

DHT-11 is a temperature and humidity sensor using a polymer capacitor detecting element. It has an output in the form of a digital signal with conversion and calculation by the sensor itself. This sensor works at 3.3 – 6 Volt Direct Current (VDC) and has a range of 0 to 100% for humidity and -40 °C to 80 °C for temperature.[10] The DHT 11 circuit is shown in Figure 4 below.

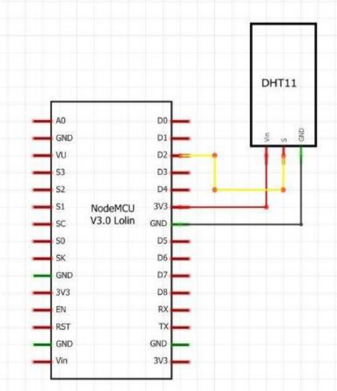


Figure 4. DHT-11 Circuit

The working principle of DHT-11 is single-wire bidirectional communication, and bidirectional communication is carried out through a reciprocal relationship (sender and receiver) between the sensor and the Microcontroller. To avoid communication errors, synchronization is needed using Serial Clock Input (SCK). This sensor system has 1 data line used for addressing and reading commands. When the sensor is active HIGH (1), the data sent from DHT-11 is digital data so that the Microcontroller can directly process it for display. When the sensor is not active at LOW (0), the digital data will stop being sent to the Microcontroller and not be processed. Voltage connected to pin 3.3V is connected to the VDD pin, and the GND pin is connected to the GND pin on the NodeMCU ESP8266 V3. The pin configuration on the DHT-11 circuit can be seen in table 1.

Table 1. DHT-11 Pin Configuration

DHT-11	NodeMCU ESP8266
DATA	Pin D2
VCC	3.3 V
GND	GND

3.2. MQ-135 circuit

The MQ-135 gas sensor is a type of chemical sensor that is sensitive to NH3 compounds, NOx, alcohol, benzene, smoke (CO), CO2, and others.[11] This sensor works by accepting changes in the resistance value (analogue) when exposed to the gas. This sensor works at 3.3 v – 5 v AC or DC and has a measurement range of 10 – 300 ppm ammonia, 10 – 1000 ppm benzene, and 10 – 300 alcohol. The MQ-135 circuit is shown in Figure 5 below.

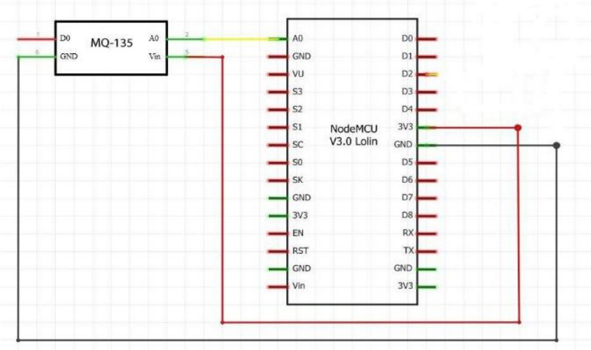


Figure 5. MQ-135 Circuit

The working principle of the MQ-135 sensor is basically to detect the presence of gases that are considered to represent NH3, NOx, alcohol, benzene, smoke (CO), and CO2 compounds. If the sensor detects the presence of these gases in the air with a particular concentration level, then the sensor will assume that there are gases in the air. When the sensor detects the presence of these gases, the sensor's electrical resistance will decrease. By utilizing the working principle of this sensor, it can detect the presence of these gases in the air. This sensor system has two data lines that can be used, namely pins A0 and D0; in this design, the ping used is pin A0, which sends an analogue signal. If the sensor detects the presence of these gases in the air, the analogue value will enlarge, which is then sent to the microcontroller. The voltage connected to the 3.3V pin is connected to the VDD pin, and the GND pin is connected to the GND pin located on the NodeMCU ESP8266 V3. The pin configuration on the DHT-11 circuit can be seen in table 2.

Table 2. MQ-135 Pin Configuration

MQ-135	NodeMCU
A0	Pin A0
D0	-
VCC	3.3 V
GND	GND

3.3. Overall Circuit

The circuit below describes the combination of all the circuits from each connected part into one in this prototype Air Quality, Temperature and Humidity Monitoring System; as shown in Figure 6, all components work and have their respective functions, namely the DHT-11 and MQ-135 sensors. As a data input medium, that will be sent to the NodeMCU ESP8266 V1 to be processed and produce output in the form of Air Quality, Temperature and Humidity.

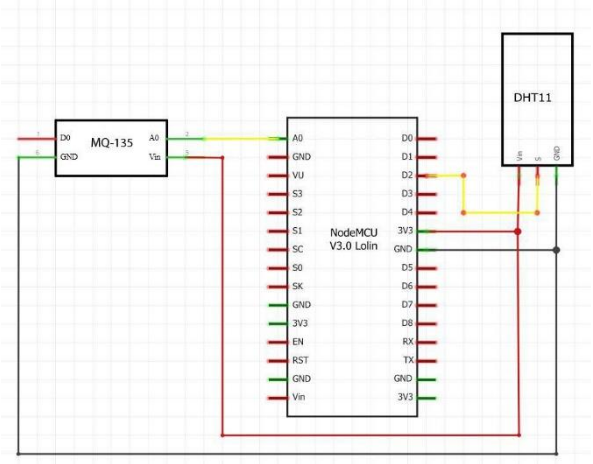
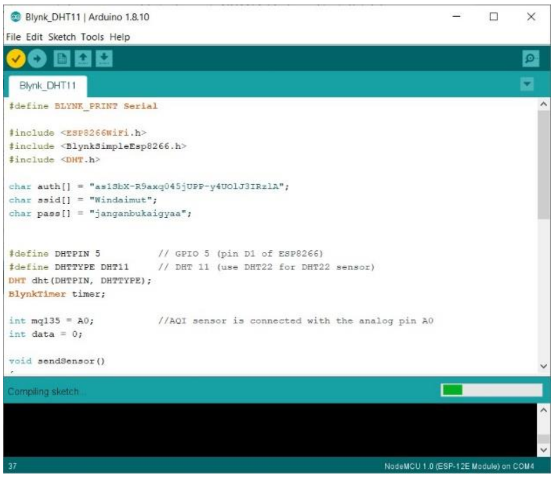


Figure 6. Overall Circuit

3.4. Arduino Programming

This tool requires Microcontroller programming to work according to its working principle. After opening the Arduino IDE application, the next step is to write the sketch program on the available worksheet.[12] After the sketch program has been created, the next step is to compile the program to check whether or not there are errors in writing program coding. After the program is compiled and no errors occur, the program can be uploaded to the Arduino Uno. Before uploading the program, first set the type of board on the Arduino by selecting tools - board - Arduino Uno, then set the port settings by selecting tools - port - COM14 (Arduino Uno). After the setup is complete, the program is ready to be uploaded. The program upload process can be seen in Figure 7.



```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

char auth[] = "as1SbX-R9axq045jUPP-y4UOlJ3IRzlA";
char ssid[] = "Windaimut";
char pass[] = "janganbukaigyaa";


#define DHTPIN 5          // GPIO 5 (pin D1 of ESP8266)
#define DHTTYPE DHT11     // DHT 11 (use DHT22 for DHT22 sensor)
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

int mq135 = A0;           //AQI sensor is connected with the analog pin A0
int data = 0;

void sendSensor()
```

```
Compiling sketch...
```

Figure 7. The process of compiling and uploading programs on the Arduino IDE

3.5. Tool Implementation

The prototype that has been produced is then implemented to detect humidity (H), temperature (T), and quality (Q) of the air in an area. The following figure illustrates the application of the prototype to the humidity (H), temperature (T), and quality (Q) of the air in an area.

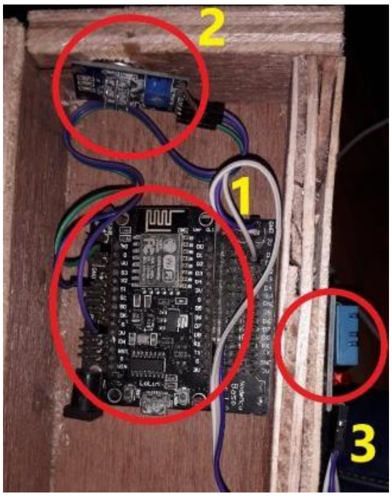


Figure 8. Illustration of Application of Prototype Detecting humidity (H), temperature (T), and quality (Q) of the air.

The placement of the components for detecting humidity (H), temperature (T), and quality (Q) of the air are shown in Figure 8. The placement of the humidity, temperature, and quality detector prototypes is in several places in the picture. NodeMCU at number 1, Sensor MQ-135 at number 2 and Sensor DHT-11 at number 3.

3.6. Testing Tool

This test aims to determine the working principle of the tools and programs that have been made under what will be achieved so that truth is obtained from each unit that is interconnected in the system.

The steps taken to perform the test are:

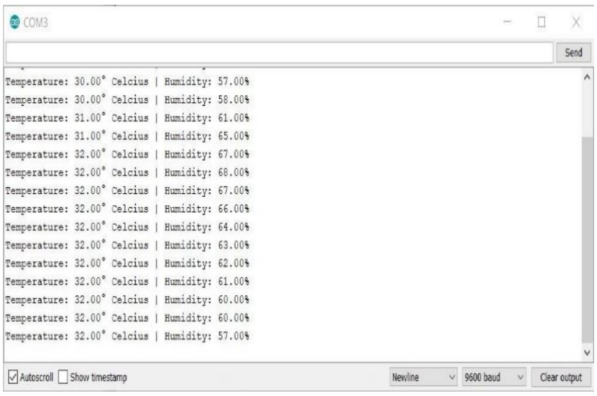
a) Prepare equipment and tools to be tested.

b) Setting up a direct current / DC 5V power supply as a voltage source for each component.

c) Check that all components are connected correctly and that there are no errors in the pins' configuration.

d) The trial is ready to be carried out.

The DHT-11 test sensor uses a serial monitor to determine the resulting output, namely temperature & humidity, by activating and programming the connectivity on the pins contained in the DHT-11 sensor with min NodeMCU ESP8266 V1.



```
Temperature: 30.00° Celcius | Humidity: 57.00%
Temperature: 30.00° Celcius | Humidity: 58.00%
Temperature: 31.00° Celcius | Humidity: 61.00%
Temperature: 31.00° Celcius | Humidity: 65.00%
Temperature: 32.00° Celcius | Humidity: 67.00%
Temperature: 32.00° Celcius | Humidity: 68.00%
Temperature: 32.00° Celcius | Humidity: 67.00%
Temperature: 32.00° Celcius | Humidity: 66.00%
Temperature: 32.00° Celcius | Humidity: 64.00%
Temperature: 32.00° Celcius | Humidity: 63.00%
Temperature: 32.00° Celcius | Humidity: 62.00%
Temperature: 32.00° Celcius | Humidity: 61.00%
Temperature: 32.00° Celcius | Humidity: 60.00%
Temperature: 32.00° Celcius | Humidity: 60.00%
Temperature: 32.00° Celcius | Humidity: 57.00%
```

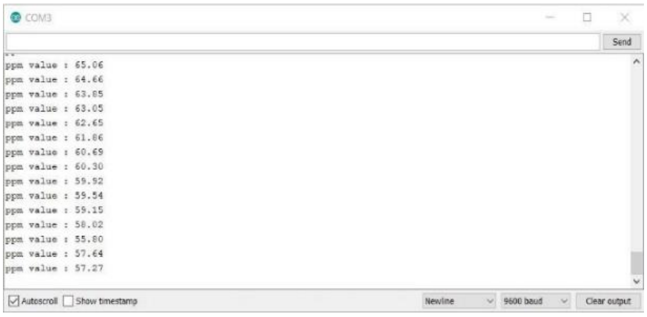
Figure 9. DHT-11 Sensor Serial Monitor Test Results

In Figure 9, the DHT-11 sensor receives input and is successfully processed by the microcontroller. The results obtained show the output in the form of temperature and humidity detected from the surrounding environment with changes every 10 minutes, which will be displayed on the serial monitor. This test takes temperature and humidity data every 10 minutes, as shown in table 3 below.

Table 3. DHT-11 Sensor Test Results

No.	Temperature (°C)	Humidity (%)	Time (seconds)
1	30,00	57,00	1
2	30,00	58,00	1
3	31,00	61,00	1
4	31,00	65,00	1
5	32,00	67,00	1
6	32,00	68,00	1
7	32,00	67,00	1
8	32,00	66,00	1
9	32,00	64,00	1
10	32,00	62,00	1

The next test is a test of the MQ-135 sensor using a serial monitor to determine the output produced, namely the air quality in the surrounding environment, by activating and programming the connectivity on the pins found on the MQ-135 sensor, the NodeMCU ESP8266 V3 pin.



```
ppm value : 65.06
ppm value : 64.66
ppm value : 63.85
ppm value : 63.05
ppm value : 62.65
ppm value : 61.86
ppm value : 60.69
ppm value : 60.30
ppm value : 59.92
ppm value : 59.54
ppm value : 59.15
ppm value : 58.02
ppm value : 55.80
ppm value : 57.64
ppm value : 57.27
```

Figure 10. Test Results Serial Monitor Sensor MQ-135

Figure 10 shows the MQ-135 sensor receiving input and being successfully processed by the microcontroller. The results obtained show the output in the form of air quality detected from the surrounding environment with changes every 10 minutes, which will be displayed on the serial monitor. This test is carried out by taking air quality data every 10 minutes, as shown at table 4 below.

Table 4. MQ-135 Sensor Test Results

No	Air Quality (PPM)	Time (seconds)
1	65.06	1
2	64,66	1
3	63,85	1
4	63,05	1
5	62,05	1
6	61,86	1
7	60,69	1
8	60,30	1
9	59,92	1
10	59,54	1

The next test is carried out to see the appearance of the Android application on a smartphone that has been successfully installed—testing whether the application has bugs or installation failures that can cause the application to crash or force close. If there are no bugs, the android application will display readings from the MQ-135 and DHT-11 sensors. The appearance of the Android application can be seen in Figure 11 below.

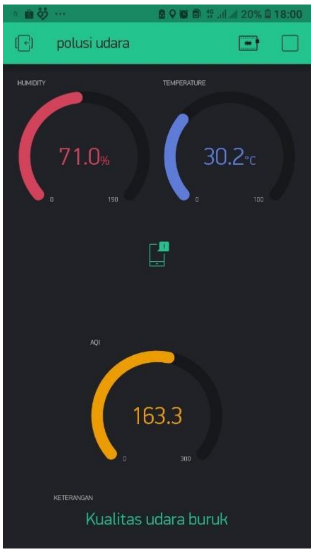
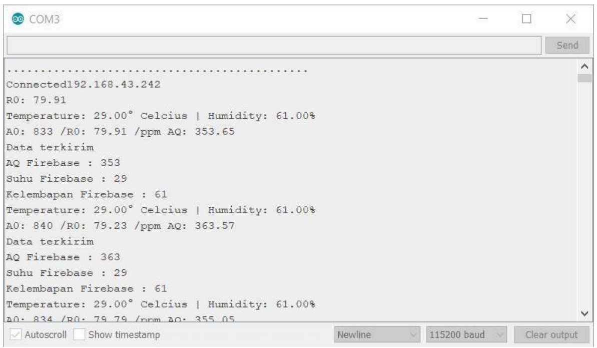


Figure 11. Display of the Air Quality Monitoring Android Application

3.7. Functional Test

In this functional test, which is carried out on a different circuit from the technical testing carried out previously, the technical test is carried out to determine the specifications of the physical size and condition of the components. In contrast, functional testing is carried out to determine the initialization of each circuit performance and whether it has been running and functioning as expected. or not. Before carrying out functional testing, the circuit is first connected to the power supply. After being connected to the power supply, the testing process is carried out Before testing, first, check whether WiFi is connected to the specified AP (Access Point) after the program is run or reset and make sure the AP is turned on. See figure 12 when WiFi has been connected to the AP.



```
............................................
Connected192.168.43.242
R0: 79.91
Temperature: 29.00° Celcius | Humidity: 61.00%
A0: 833 /R0: 79.91 /ppm AQ: 353.65
Data terkirim
AQ Firebase : 353
Suhu Firebase : 29
Kelembapan Firebase : 61
Temperature: 29.00° Celcius | Humidity: 61.00%
A0: 840 /R0: 79.23 /ppm AQ: 363.57
Data terkirim
AQ Firebase : 363
Suhu Firebase : 29
Kelembapan Firebase : 61
Temperature: 29.00° Celcius | Humidity: 61.00%
A0: 834 /R0: 79.79 /ppm AQ: 355.05
```

Figure 12. Access Point Connection Settings

Then after connecting to the AP, it will get an IP address, so the data that NodeMcu ESP8266 V3 has sent can be sent to the Realtime Database in Blynk. Equipment testing is divided into three parts: testing tools in the house, basement, and urban forest area as shown in Figure 13 below.

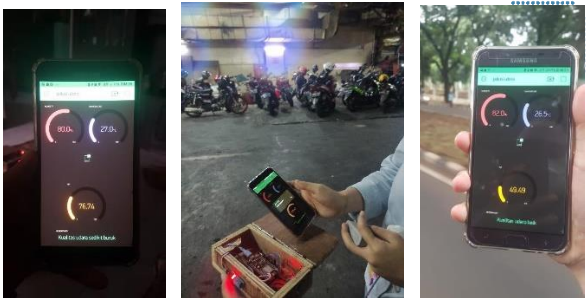


Figure 13. Testing Tools In The House, Basement and Urban Forest Area

Table 5 contains data on the measurements and sensor tests carried out inside the house. The air quality is in slightly poor condition from the data obtained from checking inside the house with an average index of 76, air temperature of 27°C and 80% humidity. Testing of the equipment was carried out from 06.26 to 07.30, the condition of the house being closed with many activities and the number of people in it affected the air quality.

Table 5. Test Results of Tools In The House

Time	Air Quality Index	Temperature (C)	Humidity (%)	Category
06.26	76.74	27	80	Medium
06.33	75.83	27	80	Medium
06.50	76.64	27	80	Medium
07.10	76.84	27	80	Medium
07.30	76.83	27	80	Medium

Table 6 shows air quality, temperature, and humidity data in the basement of a mall. Measurements were carried out from 16.35 to 17.00. Testing the equipment in the basement showed the air quality in the high average category with an average index of almost 200, an average air temperature of 31 °C, and average air humidity of 61%. Relatively high air temperature and very high humidity.

Table 6. Test Results of Tools in The Basement of A Mall

Time	Air Quality Index	Temperature (C)	Humidity (%)	Category
16.35	202.43	31	61	Very bad
16.43	201.77	31	61	Very bad
16.49	189.27	31	60	Bad
16.57	177.97	31	61	Bad
17.00	184.71	31	62	Bad

Table 7 shows air quality, temperature, and humidity data in the urban forest area. Measurements are made from hours 06.23 to 06.37, the results of the tool testing in the urban forest area show the air quality in a suitable category with an average index of 49, an average air temperature of 26 °C, and a humidity of 82%. The condition of the urban forest area is filled with trees and plants that are still maintained and rarely visited by humans because the urban forest is one of the protected places; this situation affects the air quality in the area.

Table 7. Test Results of Tools in The Urban Forest Area

Time	Air Quality Index	Temperature (C)	Humidity (%)	Category
06.23	49.49	26	82	Good
06.27	47.33	25	82	Good

Time	Air Quality Index	Temperature (C)	Humidity (%)	Category
06.30	49.35	26	82	Good
06.35	48.49	26	82	Good
06.37	46.83	26	82	Good

A tool consists of 3 parts: input, processor, and output. The input on this prototype is a DHT-11 sensor that provides input signals in the form of temperature and humidity readings, and the MQ-135 sensor, which functions to provide input signals in the form of air quality readings to the data processor. The data processor is an essential part of this circuit because it processes data from the input received to be processed into output as needed; this data processor uses the NodeMCU ESP8266 V1. The processed data obtained is then sent to the Realtime Database Blynk to be displayed on an android application created to display the output of the DHT-11 and MQ-135 sensors. Tests carried out on the DHT-11 and MQ-135 sensors through readings of the surrounding environment detected by the sensor to test the readings that can be reached DHT-11 and MQ-135 will detect when there is an input received. In the tests carried out, the data obtained will be sent from the microcontroller, and then the output will be displayed on the android application created.

4. Conclusion

Based on the above discussion of the system, starting from the design and manufacture of tools to the trials that have been carried out, the researchers have several conclusions. Namely, the system has succeeded in detecting and measuring microcontroller-based residential environmental air pollution via IOT with NodeMCU ESP8266 V1, air quality sensor MQ135, temperature sensor, and DHT-11 humidity and connected to the Blynk Realtime Database and the Blynk Android App. Based on the test results, it is proven that the air quality in the house is not necessarily good, so it would be nice to know and maintain the excellent air quality in the house. For further development of the tool, this device requires an indication device such as a buzzer to warn when the air quality is poor immediately. An LED for warnings such as a green light when the air quality is good, a yellow light when the air quality is slightly poor, and a red light when the air quality is poor. Moreover, the device requires a Dust Sensor (Dust Sensor GP2Y1010AU0F) to determine the dust density in the air to get the correct air quality value.

References

[1] R. Satra, R. Satra, and A. Rachman, “Pengembangan Sistem Monitoring Pencemaran Udara Berbasis Protokol ZIGBEE dengan Sensor CO,” *Ilk. J. Ilm.*, vol. 8, no. 1, pp. 17–22, Apr. 2016.

[2] D. K. Faroqi, Adam Hadisantoso, Eko Prabowo Halim and M. S. WS, “Perancangan Alat Pendeteksi Kadar Polusi Udara Menggunakan Sensor Gas MQ-7 Dengan Teknologi Wirelles HC-05,” *J. ISTEK*, vol. X, no. 2, pp. 33–47, 2016.

[3] T. F. Arya, M. Faiqurahman, and Y. Azhar, “Aplikasi Wireless Sensor Network untuk Sistem Monitoring dan Klasifikasi Kualitas Udara,” *J. Sist. Inf.*, vol. 14, no. 2, pp. 74–82, Oct. 2018.

[4] F. M. Rizon, “Alat Pendeteksi Udara Di Dalam Mobil Menggunakan Arduino Uno,” *Jumantaka*, vol. 02, no. 01, pp. 31–40, 2018.

[5] I. Journal, D. Srivastava, A. Kesarwani, and S. Dubey, “IRJET-Measurement of Temperature and Humidity by using Arduino Tool and DHT11 Measurement of Temperature and Humidity by using Arduino Tool and DHT11,” *Int. Res. J. Eng. Technol.*, vol. 876, 2008.

[6] “Hands-On Internet of Things with Blynk: Build on the power of Blynk to ... - Pradeeka Seneviratne - Google Buku.” [Online]. Available:

https://books.google.co.id/books?hl=id&lr=&id=ZHteDwAAQBAJ&oi=fnd&pg=PP1&dq=blynk+application&ots=K0_6c9QtiO&sig=46c3ZhLFuy36t01nI2oYXo63fN4&redir_esc=y#v=onepage&q=blynk application&f=false. [Accessed: 30-Apr-2022].

[7] D. H. Hareva and C. D. P. Marsyaf, "Air quality monitoring at Pelita Harapan University using the MQ-135 sensor," *ACM Int. Conf. Proceeding Ser.*, pp. 6–10, Nov. 2019.

[8] H. Ouldzira, A. Mouhsen, H. Lagraini, M. Chhiba, A. Tabyaoui, and S. Amrane, "Remote monitoring of an object using a wireless sensor network based on NODEMCU ESP8266," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 3, pp. 1154–1162, 2019.

[9] F. N. Abbas, I. M. Saadoon, Z. K. Abdalrdha, and E. N. Abud, "Capable of gas sensor MQ-135 to monitor the air quality with arduino uno," *Int. J. Eng. Res. Technol.*, vol. 13, no. 10, pp. 2955–2959, 2020.

[10] A. D. Hendra Saptadi Sekolah Tinggi Teknologi Telematika Telkom Purwokerto Jl I Panjaitan No, "Perbandingan Akurasi Pengukuran Suhu dan Kelembaban Antara Sensor DHT11 dan DHT22," *J. INFOTEL*, vol. 6, no. 2, pp. 49–56, Nov. 2014.

[11] R. A. Gustavia and E. Nurraharjo, "Rancang Bangun Sistem Multiple Warning Deteksi Asap Rokok," *Pros. SINTAK 2018*, pp. 278–282, 2018.

[12] A. A. Rosa, B. A. Simon, and K. S. Lieanto, "Sistem Pendeteksi Pencemaran Udara Portabel Menggunakan Sensor MQ-7 dan MQ-135," *Ultim. Comput. J. Sist. Komput.*, vol. 12, no. 1, pp. 23–28, Jul. 2020.