



# Design of Gas Leakage Monitoring System Based on Android Application and Nodemcu ESP8266

Nabil Fuadi<sup>1\*</sup>, Achmad Fauzi<sup>2</sup>, Husnul Khair<sup>3</sup>

<sup>1,2,3</sup>STMIK Kaputama

[nabilfuadfuadi@gmail.com](mailto:nabilfuadfuadi@gmail.com)<sup>1\*</sup>, [fauzyrivai88@gmail.com](mailto:fauzyrivai88@gmail.com)<sup>2</sup>, [husnul.khair@gmail.com](mailto:husnul.khair@gmail.com)<sup>3</sup>

## Abstract

Gas leakage is a serious problem that can threaten public safety and health and cause significant material losses. This research aims to design and implement a gas leak monitoring system that can be accessed remotely based on Android applications and NodeMCU ESP8266. NodeMCU ESP8266 is chosen as the main microcontroller equipped with an MQ-2 gas sensor to detect the presence of hazardous gas. This research incorporates Internet of Things (IoT) technology to allow users to remotely monitor the condition of gas leaks, thereby increasing the level of safety in the use of gas in households or small industries. The hardware design includes the use of an MQ-2 gas sensor that is sensitive to certain gas concentrations, as well as setting up an ESP8266 NodeMCU to transmit detection data to a Firebase server for later access through an Android application. The Android application was developed using Android Studio with a focus on an intuitive user interface to monitor the status of gas leaks in real-time. The research methods used include system design, hardware and software implementation, and thorough system testing by simulating gas leak scenarios to test the reliability and response of the system. The results show that the system can detect gas leaks with high accuracy. It is expected that the system developed in this research can be a practical and effective solution in reducing the risk of accidents caused by gas leaks, as well as making a positive contribution in increasing awareness of the safety of gas use in the community.

**Keywords:** gas leak, monitoring, NodeMCU ESP8266, Android application, Internet of Things

## 1. Introduction

Today, Internet of Things (IoT) technology has made a significant impact on improving safety and convenience, including gas leak detection in the home or workplace. The use of gas sensors connected to the IoT network enables real-time and responsive monitoring of gas leaks. Internet of Things (IoT) technology provides the potential to overcome the limitations that exist in conventional gas leak monitoring systems. Through the integration of gas detection sensors with internet-connected hardware, such as NodeMCU ESP8266, and mobile applications, gas leak monitoring systems can become more responsive, accessible, and smarter [1], [2].

One threat that is often faced is gas leaks, such as LPG (Liquefied Petroleum Gas) gas used for cooking. Gas leaks can be a serious hazard because they can cause fires or explosions that threaten life and property. Therefore, the development of a gas leak monitoring system is very important as a preventive measure to reduce the risk of accidents due to gas leaks. By using increasingly sophisticated technology, we can design a system that is effective in detecting gas leaks quickly and accurately. In this context, the use of Android applications and microcontroller devices such as NodeMCU ESP8266 offers an innovative and affordable solution [3].

## 2. Literature Review

### 2.1. Design

Design is a visual form that results from planned creative forms. The first step in design starts from irregular things in the form of ideas or ideas then through the process of cultivation and management will produce regular things, so that things that are already regular can fulfil their functions and uses properly. Design is the depiction, planning, sketching of several separate elements into a unified and functional whole [4].

### 2.2. Internet of Things (IOT)

Internet of Things, often known as IoT, is an embedded system that aims to expand the utilisation of continuous internet connectivity. Capabilities such as data sharing, remote control, and so on. In building the Internet of Things, engineers must pay attention to three aspects,

namely: Size, space, and time. In developing IoT, the time factor is usually an obstacle. It usually takes a long time because building a complex network in IoT is not easy and cannot be done by just anyone [5].

### **2.3. Mikrokontroler**

Control circuits are increasingly needed to control various equipment used by humans in everyday life. A control circuit or also known as a microcontroller is a circuit that is created to perform various functions according to needs. Microcontroller is a breakthrough in microprocessor technology and also microcomputers. As a new technology, semiconductor technology with more transistor content but only requires a small space. In general, microcontroller capabilities have different capacities so that the selection of the right microcontroller is very influential on the performance of the programme to be built. What needs to be known between one person and another will differ in terms of ease of learning. If you are familiar with the basic programming language you can use the basic stamp microcontroller, if you are familiar with the java programming language you can use Jstamp, if you are familiar with the C++ programming language you can use it for the MCS51 family and many more [6].

### **2.4. NodeMCU ESP8266**

NodeMCU ESP8266 is a microcontroller module designed with ESP8266 inside. ESP8266 functions for Wifi network connectivity between the microcontroller itself and the Wifi network. The term NodeMCU by default actually refers to the firmware used rather than the hardware development kit, and NodeMCU can also be interpreted as the ESP 8266 arduino board. (Setiawan, 2023) NodeMCU has 17 GPIO pins that can be integrated with other electronic components. Works at a voltage of 3.3 v - 5 v, with a power consumption of 10uA ~ 170mA. The processor speed ranges from 80~160MHZ and has 32KB+80KB of RAM and flash memory up to 16 MB making the NodeMCU V1 more efficient than previous versions [7].

### **2.5. Socket Expansion Esp8266**

This NodeMCU ESP8266 Expansion Socket is an expansion board designed to work with the NodeMCU V3 ESP8266 ESP-12E. Powered by 6V to 24V DC input, it provides support power to operate the NodeMCU V3 and provides easy connection to all its input and output pins to help you connect your NodeMCU board to other devices [8].

### **2.6. Arduino Software (IDE)**

IDE stands for Integrated Development Environment or in simple language is an integrated environment used for development. Arduino uses its own programming language, which is similar to the C language. The Arduino programming language (sketch) has been changed to make it easier for beginners to do programming from the original language. Before being sold to the market, the arduino microcontroller IC has been embedded with software called a bootloader, which acts as an intermediary between the Arduino compiler and the microcontroller [9].

### **2.7. Firebase**

Firebase is a cloud service provider with back-end as a service based in San Francisco, California. Firebase makes a number of products for mobile or web application development. Two interesting features of Firebase are Firebase Remote Config and Firebase Realtime Database. In addition, there are supporting features for applications that require notifications, namely Firebase Notification. One of the advantages offered by Firebase is that this database stores data locally when a device is not connected to internet access.

### **2.8. Android studio app**

Android studio is an integrated development environment (IDE) which is a developer environment created specifically for the development of the android application operating system. This application is created in a software called IntelliJ IDEA owned by JetBrains, commonly called android studio which is a replacement for Eclipse Android Development Tools or ADT as the most important IDE in creating android applications. Android is an open source platform that is developed by anyone. This android studio application is used to create the android application itself.

### **2.9. Sensor Mq-6**

The MQ-6 gas sensor is a sensor that can be used to detect the presence of LPG gas through the propane and butane gas content in the LPG gas. This sensor has a high sensitivity to LPG, iso-butane, propane, and LNG liquefied natural gas.

### **2.10. Lcd 16x2**

Lcd 16x2 is a 16x2 LCD (Liquid Crystal Display) is one of the most popular display media used as an interface between the microcontroller and the user. With this 16x2 LCD viewer the user can see / monitor the state of the sensor.

### **2.11. Buzzer**

Buzzer is an electronic component that functions to convert electrical vibrations into sound vibrations. Basically the working principle of a buzzer is almost the same as a loud speaker. In this research, I used a 3V-24V SFM 27 3cm DC buzzer.

### 2.12. Motor Servo

Servo motor is an electrical device that is generally used in industrial equipment with a closed feedback system that functions as a driver or rotating object with high precision control in angular position, acceleration and speed that is not owned by other motor drives.

## 3. Analysis and Design

### 3.1. System Design

In the design entitled "Design of a gas leak monitoring system based on android applications and nodemcu esp8266." whose manufacture there are several problems that must be solved these problems include:

1. How to design a gas leak monitoring system based on android application?  
Designing an android application-based gas leak monitoring system.
2. How to design a monitoring system that integrates hardware (such as gas sensors, NodeMCU ESP8266) and software (Android application) so that it can function efficiently in detecting gas leaks, sending data in real-time, and providing alerts to users?  
Designing a monitoring system that integrates hardware (such as gas sensors, NodeMCU ESP8266) and software (Android application) so that it can function efficiently in detecting gas leaks, sending data in real-time, and providing alerts to users.
3. How to develop an Android application that can interact with NodeMCU ESP8266 via Wi-Fi connection, receive gas detection data, display information clearly and easily understandable to users, and provide notifications or alarms when a gas leak is detected.  
Develop an Android application that can interact with NodeMCU ESP8266 via Wi-Fi connection, receive gas detection data, display information clearly and easily understandable to users, and provide notifications or alarms when a gas leak is detected.

### 3.2. Circuit Block Diagram

The block diagram of the designed system, as shown in the figure:

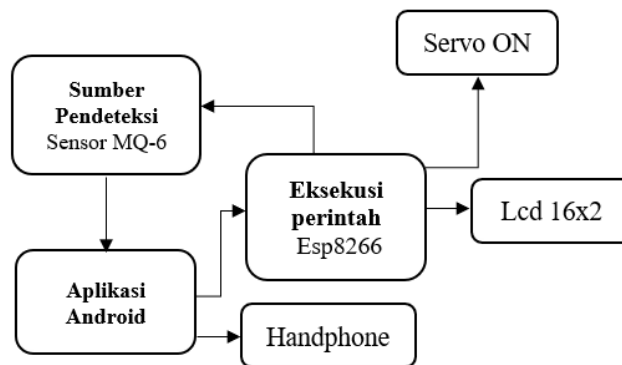


Fig. 1: System Block Diagram on Nodemcu ESP8266

The description of the block diagram above is as follows:

1. Nodemcu Esp8266 functions as the brain and controller function in the electronic system system.
2. The MQ-6 sensor sends messages via nodemcu esp8266 to the android application, the MQ-6 sensor functions to measure the level of LPG gas levels which will be the object of this research.
3. Then the display of gas levels will appear in the android application and 16x2 LCD that has been connected via nodemcu esp8266.
4. And on the mobile phone display, a graphical display of the detected gas levels will appear.
5. Then the ON servo motor functions as a GAS regulator opener.

### 3.3. Block Diagram of Application System and Firebass

A software system block diagram is a visual representation of the various software components in a system. It helps in understanding the structure and relationships between these components. You can see the picture below:

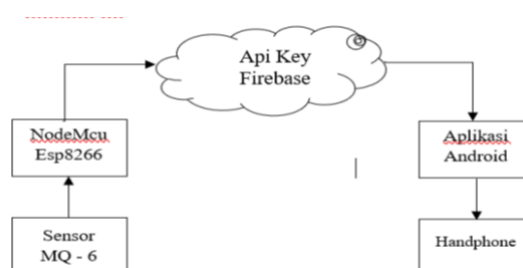


Fig. 2: Software Block Diagram

### 3.3. Overall Tool Set

This circuit functions to run the tool system so that the tool can work according to the inputted programme instructions.

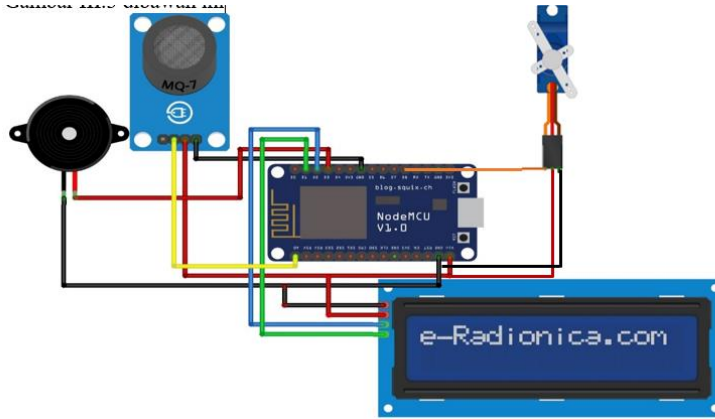


Fig. 3: Overall Tool Set

### 3.4. Interface Design

Interface specification is a display of the programme that will be built for interface needs with the user. The interface design of the gas sensor monitoring system to be built is as follows [10]:

1. Splash Screen Layout

Layout Splash Screen is the initial screen that is opened before the application is run. Can be seen in Figure III.4 (a)

2. Login Layout

Layout Login in this study does not use the database only as a key to open the application. Can be seen in Figure III.4 (b)

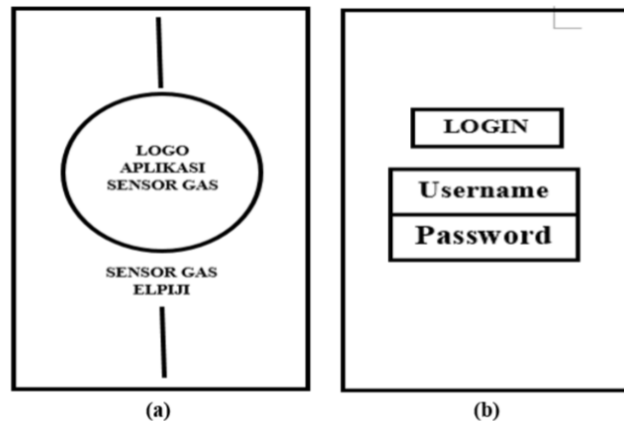


Fig. 4: Layout Shape Splash Screen & Login Layout

3. Home Layout

Layout Home as the initial display when the application is opened after login contains about the application. Can be seen in Figure III.5 (a)

4. Layout Check Gas Status

This layout functions as monitoring the gas sensor while displaying the graph. Can be seen in Figure III.5 (b)

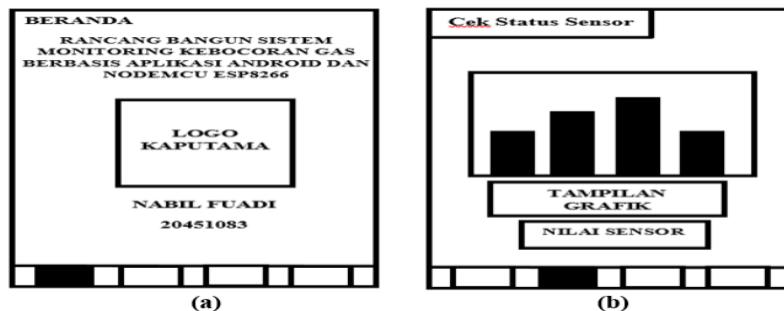


Fig. 5: Form Home Layout & Sensor Status Check Layout

5. Sensor Data History Layout

Layout History serves as a place for old sensor data. Can be seen in Figure III.6 (a)

6. Layout Logout

This layout serves as a place to exit the application. Can be seen in Figure III.6 (b)

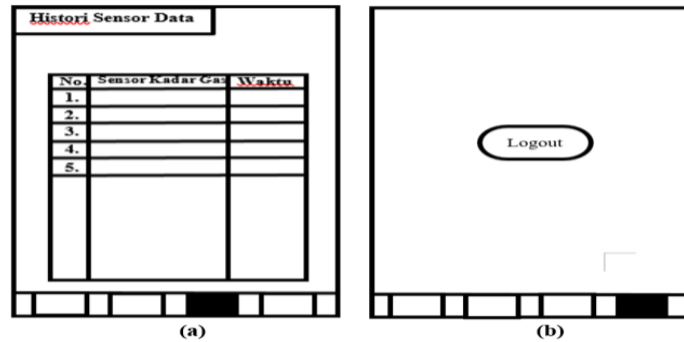


Fig. 6: Shape Sensor History Layout & Logout Layout

3.4. System Flowchart

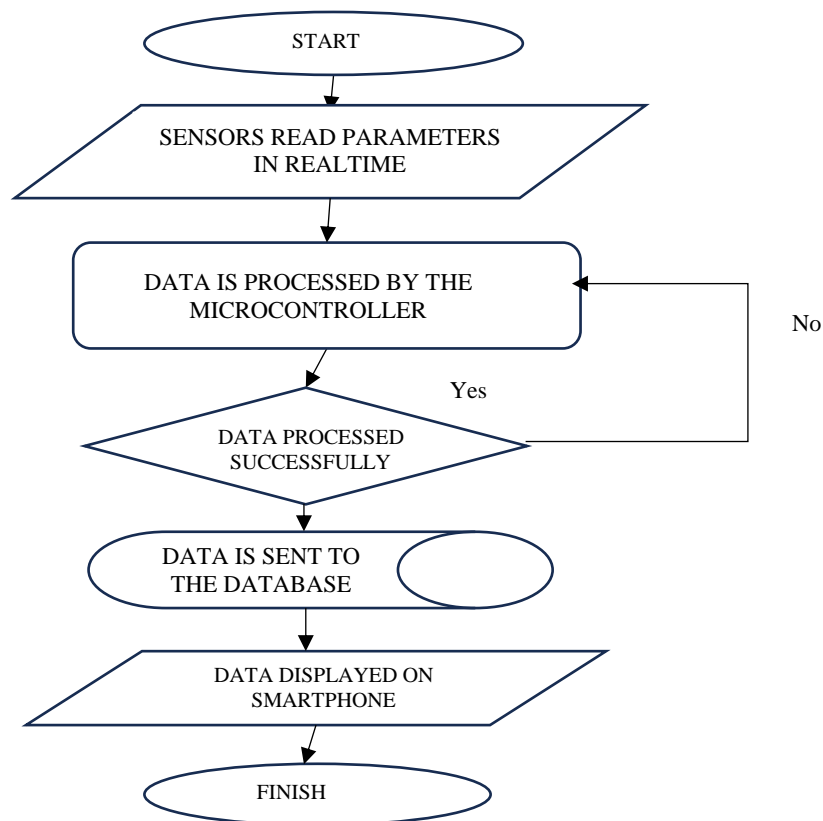
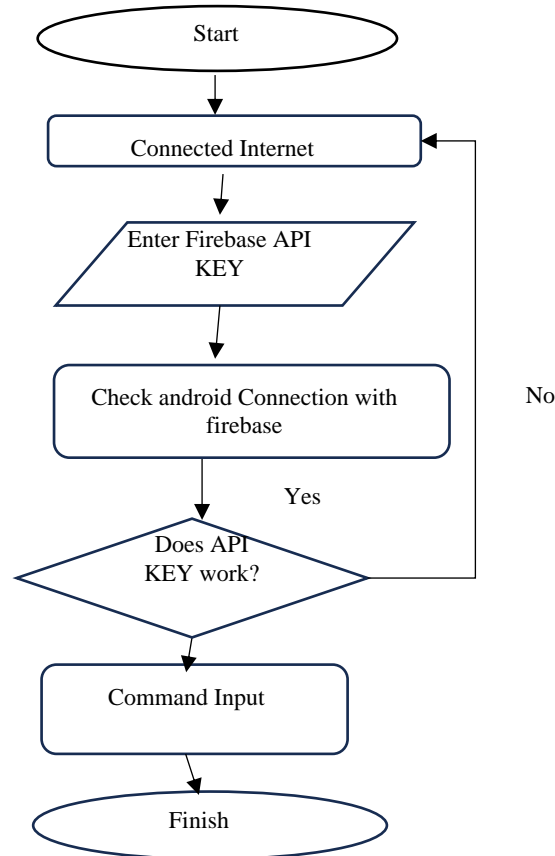


Fig. 7: Flowchart of Device Monitoring System

Description:

From the flowchart above, it can be explained that the design of this tool consists of several flows.

1. Start the programme
2. MQ-6 sensor reads gas levels in realtime
3. Then the microcontroller processes the data, if successful it will be sent to the database and if not it will continue to process these parameters.
4. The value sent to the database will be sent to the smartpone and will display the value of the MQ-6 sensor.
5. And finished.



**Fig. 8:** Flowchart of IoT System on Firebase

Description:

1. Start programme
2. Connect to the internet
3. Input firebase Api Key to android studio
4. If successful enter the command, if not successful check the api key from firebase whether it is running or not
5. And finished.

## 4. Discussion and Implementation

### 4.1. Discussion

In this chapter, it will be explained and shown how the results of testing the design of the tool made along with the discussion. The results of the tests carried out are a tool made or designed and programmed using the Arduino application. The tools made will be used to develop a system that is able to detect gas leaks and monitor environmental conditions based on android applications and NodeMCU ESP8266.

### 4.2. Component Testing

To find out whether the NodeMcu ESP8266 Microcontroller circuit has worked properly on the tool, testing is carried out by giving the command programme to the Microcontroller by inputting data from the computer into the Microcontroller. In doing the installation, first connect the computer with the downloader via a USB cable to the microcontroller circuit. To test the tool with commands can be done with several steps, among others:

1. The first step is to run the Arduino software, after the application loads it will look like a display as shown in Figure 9.

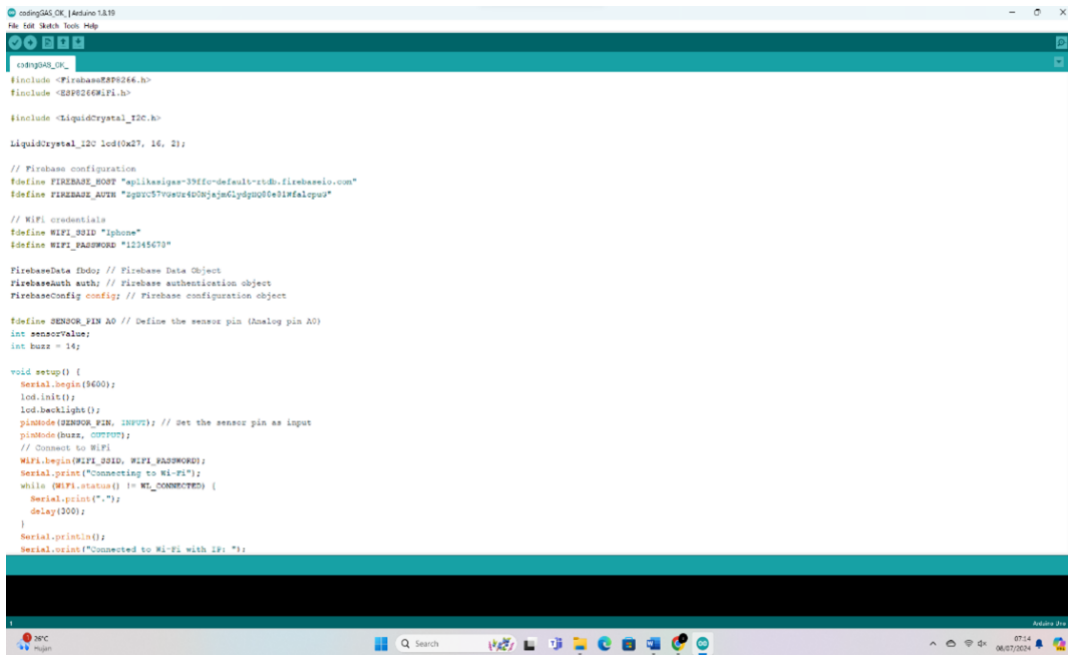


Fig. 9: Arduino Software Display

2. Furthermore, to program the Esp32 Board Microcontroller, namely by typing the programme according to what is needed in the tool. As seen in Figure 10.

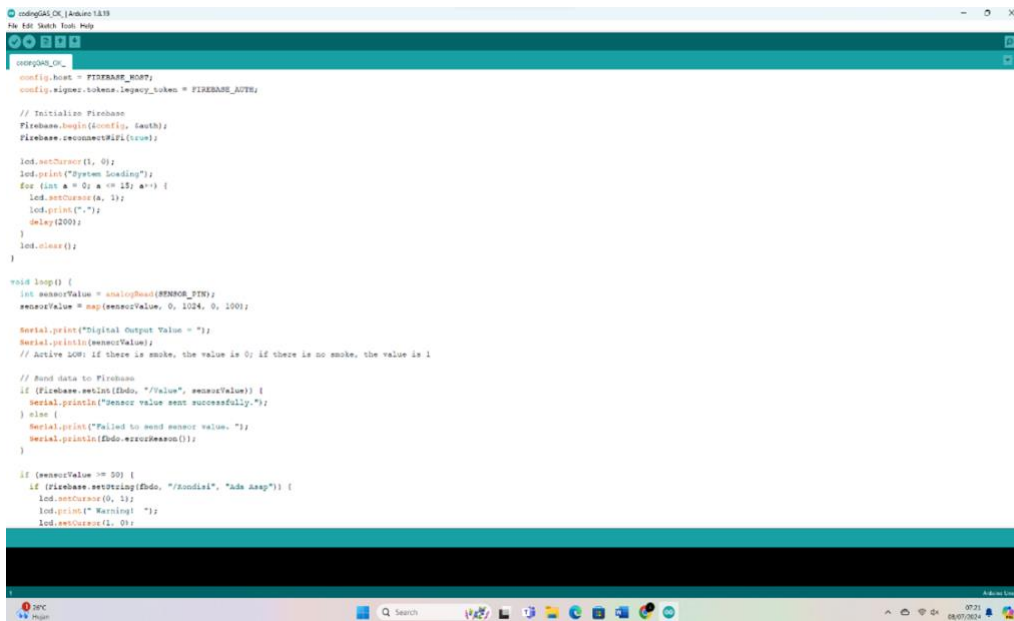


Fig. 10: Programme View

3. Before continuing the microcontroller installation stage on the completed programme, the programme is first saved before being compiled. To save the programme can be seen in Figure 11.

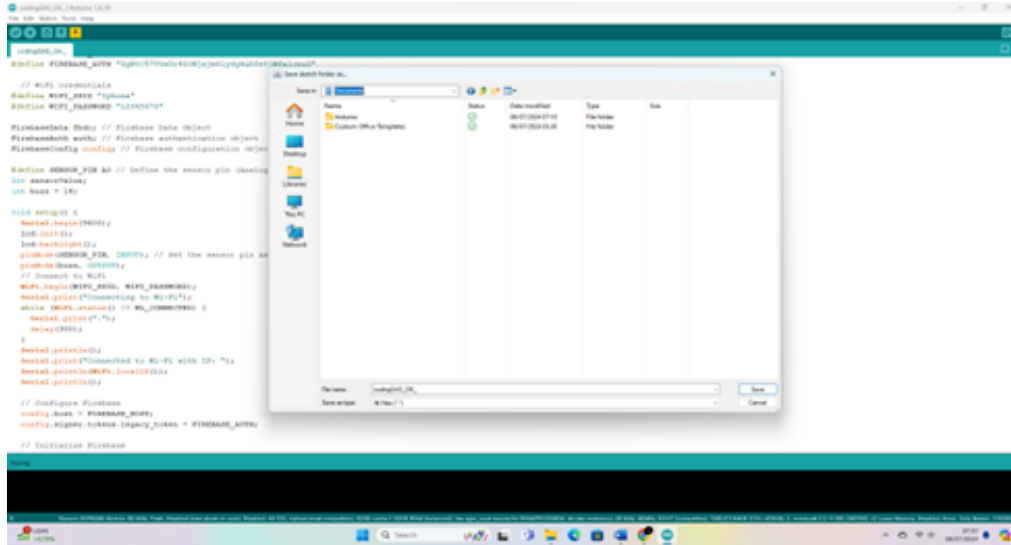


Fig. 11: File Storage Process

4. To continue the microcontroller installation stage, the programme is first checked by clicking the "Compile" button to set the programme into the ChipMicrocontroller. It can be seen whether the programme made has errors or not, if successful it will say "No errors". Compile process can be seen in Figure 11.

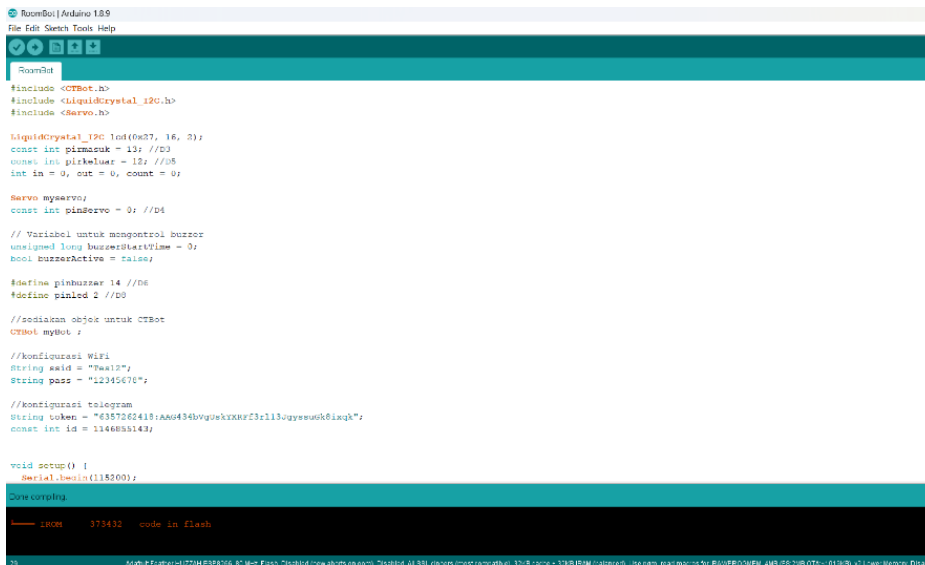


Fig. 12: Compile Result

### 4.3 Android Studio Software Testing

An important process to verify that the app you are developing is functioning properly, meets user expectations, and fulfils the desired quality standards. This testing is done in various stages to ensure that every aspect of the app has been thoroughly tested before it is released to the end users. The following are some of the common types of testing done in Android app development:

1. Unit testing is a type of testing that focuses on testing smaller units of application code, such as classes or methods. Unit testing is done to ensure that each block of code functions correctly and is isolation-free. Can be seen in Figure 13 below.

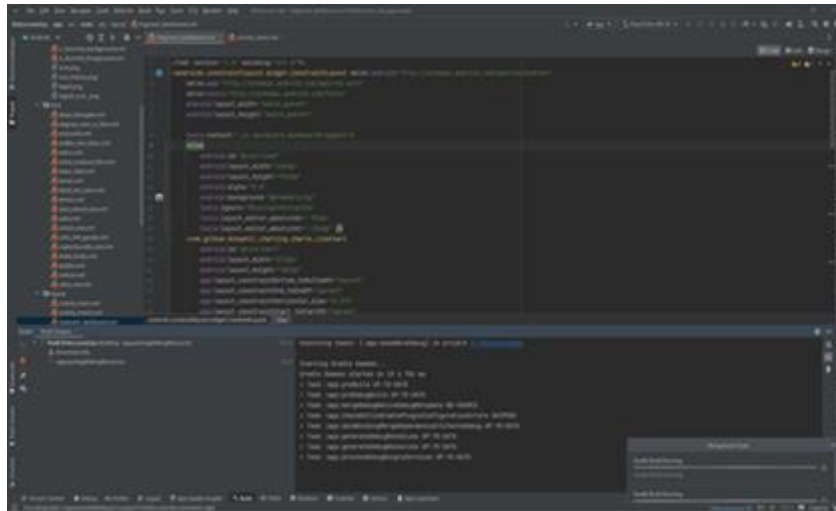


Fig. 13: Programme View

2. Integration Testing: This technique tests how the different components in the application interact with each other. The purpose of integration testing is to ensure that each component of the application functions properly when integrated into the firebase console. Can be seen in Figure 14 below.

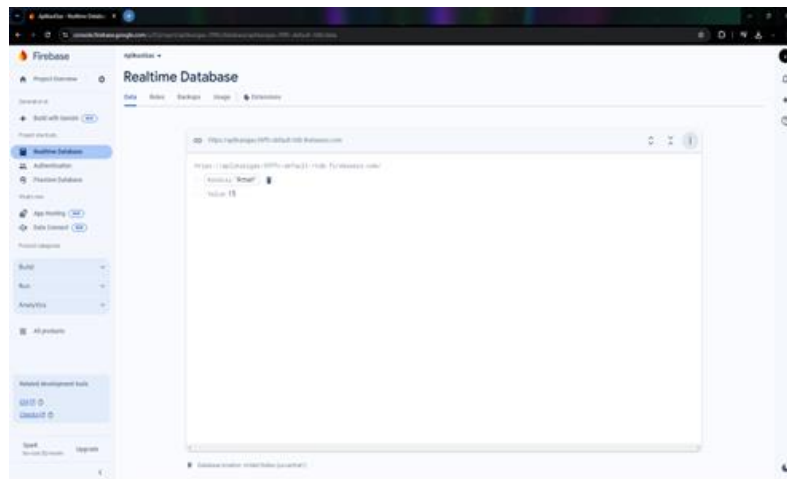


Fig. 14: Programme View to Firebase

3. UI Testing: User interface verification aims to confirm that user interface elements, such as buttons, input fields, and navigation, work properly. It can be seen in Figure 15.

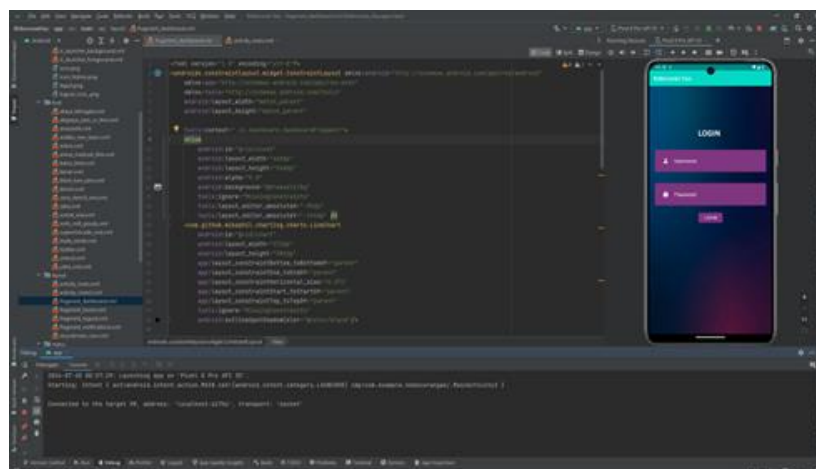


Fig. 15: Testing the Programme Interface

#### 4.4 Hardware Testing

After all the circuits that have been designed in "Design of a gas leak monitoring system based on android applications and nodemcu esp8266", then all the circuits that have been completed are united.

#### 4.5 Overall Results of Hardware

Devices After the hardware device is programmed to the microcontroller and has been executed using the downloader, the program has automatically entered the microcontroller. Can be seen in Figure 16.

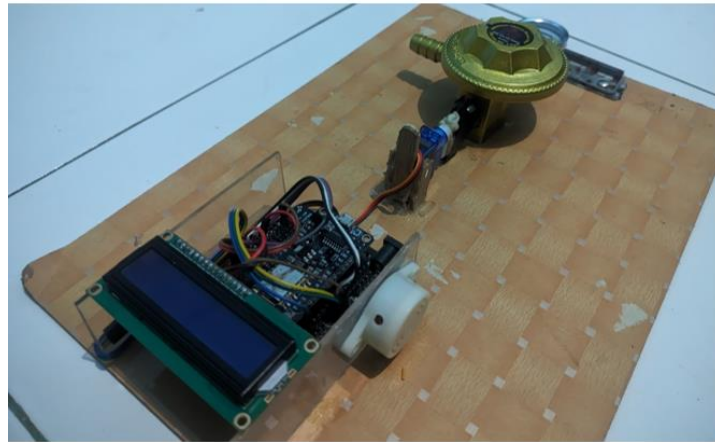


Fig. 16: Device Overall View

##### 4.5.1 Test Result of GAS Content Value

For the results of testing the value of GAS levels in a normal or standby position. Can be seen in Figure 17 below.

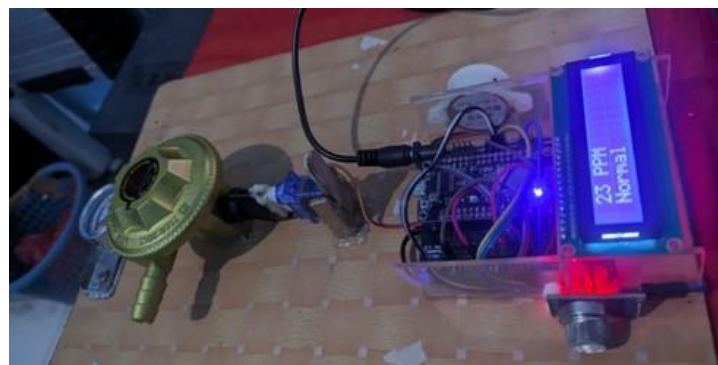


Fig. 17: Test Result og GAS Content Value

##### 4.5.2 Regulator Test Result

For the results of regulator testing When the servo moves to open the lever on the regulator, it aims to stop the GAS leak. Can be seen in

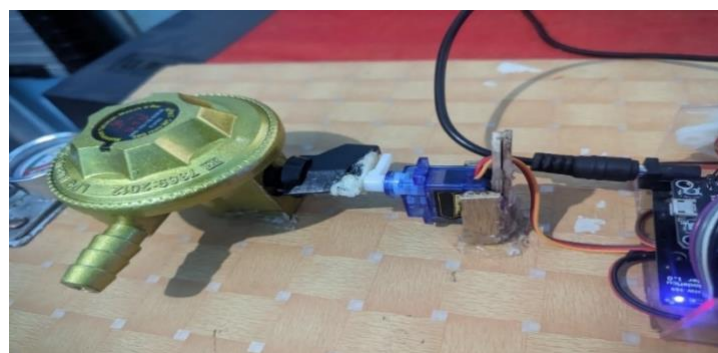


Fig. 18: Servo Regulator Testing Result

#### 4.6 Android Device Testing Result

After the android device device is programmed, it will automatically connect to the firebase console and will display the value of the microcontroller device. Can be seen in Figure 19.

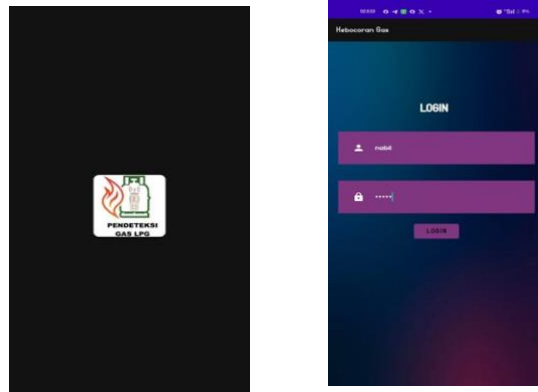


Fig. 19: Android Device Display Result

#### 4.6.1 Home Interface Testing Result

It includes results from testing the interface of the application or system home page. This test may include functionality such as navigation, layout, and integration of other interface elements. Can be seen in Figure 20.



Fig. 20: Home Interface Testing Result

#### 4.6.2 Testing Result for the GAS Level Check Interface

This section will include test results for interfaces used to check gas levels or similar features in the application or system you are testing. It can be seen in Figure 21.

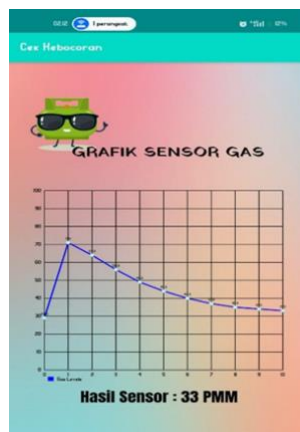


Fig. 21: Testing Result for the GAS Level Check Interface

### 4.6.3 GAS Data History Interface Testing Result

This refers to the test results for interfaces that display historical data related to gas levels, such as graphs, data filters, or usage history. Can be seen in Figure 22.



Fig. 22: GAS Data History Interface Testing Result

## 5. Conclusion

After carrying out the design and manufacture stages of the system which is then continued with the testing and analysis stages, the following conclusions can be drawn:

1. From the design that has been made, how to design a gas leak monitoring system is first to prepare a very important tool needed first, Nodemcu Esp8266 as a controller, data receiver, and data processing and WIFI signal receiver that has been connected to android. Then the Mq-6 gas sensor works to detect gas leaks and the Android studio application that makes the gas leak application.
2. By designing a monitoring system that integrates hardware such as (gas sensor, NodeMCU ESP8266) and software (android application) so that it can function efficiently in detecting gas leaks, sending data in real time, and providing warnings to users by connecting the mq-6 sensor device to the NodeMCU ESP8266 then programmed and can be connected to wifi on the device as a local network in sending and receiving data.
3. The existence of this application in the development that has been made is suitable for application in various contexts, including household, industrial, and commercial. In addition, the use of IoT technology in this application provides convenience in remote monitoring and effective management of gas leaks.

## Reference

- [1] Daru, A. F., & Whisnumurti Adhiwibowo. (2021). Application of Mq2 Sensor for Gas Leak Detection and Bb02 Sensor for Fire Detection with Blynk Application Controller. *Journal of Information and Communication Technology*, 12 (1), 37-43. <https://doi.org/10.51903/Jtikp.V12i1.229>.
- [2] Pandega, D. M., & Marcos, H. (2023). Design of Gas Leak Detection Prototype Using Mq-6 Sensor for Household. *Journal of Computer Engineering and Systems*, 4(1), 1-9. <https://doi.org/10.33365/Jtikom.V4i1.2333>.
- [3] Putra, Y. D., & Suhendi, H. (2021). Making a Gas Leak Detection Tool for the Use of Liquefied Petroleum Gas (Lpg) Tubes Using an Mq 6 Sensor. *Journal Of Innovation Research And Knowledge*, 1(6), 903. <https://www.bajangjournal.com/index.php/Jirk/article/view/621>.
- [4] Roihan, A., Permana, A., & Mila, D. (2020). Monitoring Gas Leaks Using Arduino Uno and Esp8266 Microcontrollers Based on the Internet Of Things. *Icit Journal*, 2(2), 170-183. <https://doi.org/10.33050/Icit.V2i2.30>.
- [5] Silalahi, A., Hartama, D., Kirana, I. O., Gunawan, I., & Sumarno, S. (2022). Design of a Leak Detection Tool on Gas Tubes Using Sms-Based Arduino. *Krisnadana Journal*, 1(3), 48-58. <https://doi.org/10.58982/Krisnadana.V1i3.178>.
- [6] Suhaybatul Aslamia. (2019). Human Detection Robot as a Room Security System Using Pear Sensor with Xbee Communication Media Leonardo's Arduino-based. *Advantages of Arduino from Microcontroller Hardware Platform*, 4-27. <https://www.google.com/arduino/pdf>.
- [7] F. Fadillah, R. Buatun, and S. Ramadani, "IoT-based Hydroponic Plant Monitoring System", *j. of artif. intell. and eng. appl.*, vol. 3, no. 1, pp. 38–43, Oct. 2023.
- [8] R. Sinaga, N. Novriyenni, and S. Syahputra, "Design of an Automatic Water Faucet System Using the IOT Based HC-SR04 Sensor", *j. of artif. intell. and eng. appl.*, vol. 3, no. 1, pp. 274–278, Oct. 2023.
- [9] Rizky Hidayat, N. Novriyenni, and Siswan Syahputra, "Design of a Temperature Control and Object Motion Detection System in the Server Room Using IOT-Based Wemos D1", *j. of artif. intell. and eng. appl.*, vol. 3, no. 1, pp. 323–327, Oct. 2023.
- [10] Muhammad Alfanny, Achmad Fauzi, and Hermansyah Sembiring, "Design and Build a System for Turning on an IoT-Based Motorcycle Starter Using Voice", *j. of artif. intell. and eng. appl.*, vol. 3, no. 1, pp. 357–362, Oct. 2023.