

# CHAPTER 5

## IMPLEMENTATION AND TESTING

### 5.1 Implementation

#### 5.1.1. Source code from Arduino IDE

```
1. const char *apiKey = "F688STUHKV9FG0YP";
```

Code to declaration variable of apiKey. It was obtained from Website IoT Thingspeak.

```
1. const int maks_gas = 5000;
```

Code to Declaration of maks\_gas. It used to limit the value of Gas Sensor.

```
1. persen_gas = map(analogRead(gas), 0, 1023, 0, 100); //range 0-100
```

```
2. gas_ppm = map(analogRead(gas), 0, 1023, 0, 1000); //0-10000
```

Row 1 code program to change the voltage reading value to data percent. Row 2 code program to change the voltage reading value to data bit.

```
1. client.print(String("GET ") + resource + apiKey + "&field1=" +  
persen_gas + "&field2=" + gas_ppm +
```

```
2. " HTTP/1.1\r\n" +
```

```
3. "Host: " + server + "\r\n" +
```

```
4. "Connection: close\r\n\r\n");
```

row 1 – 4 this code program is the key in sending data to thingspeak.

```
1. const int maks_gas = 299;
```

```
2. if(count >= 300)
```

```
3. {
```

```
4.     //fungsi untuk komunikasi ke thingspeak
```

```
5.     makeHttpRequest();
```

```
6.     count = 0;
```

```
7. }
```

```
8. else{
```

```
9.     if(gas_ppm > maks_gas)
```

```
10.    {
```

```
11.        digitalWrite(led1, HIGH);
```

```
12.        digitalWrite(led2, HIGH);
```

```
13.        tone(buzzPin, 2000, 1000);
```

```
14.        Serial.println(gas_ppm);
```

```
15.        Serial.println("Gas Bocor");
```

```
16.    }
```

```

17.   else
18.   {
19.       digitalWrite(led1, LOW);
20.       digitalWrite(led2, LOW);
21.       noTone(buzzPin);
22.   }
23.   delay(1000);
24.   count = count+1;
25. }
26. Serial.print(count);

```

Row 2, 6, 24 and 26 code program to set timer for upload data to Website Thingspeak. Row 6 to restore the count from the beginning after reaching count 300. Row 24 code to add count +1. And row 26 code program to print the count in Serial monitor arduino. Row 4 code program to communicate the thingspeak. Row 9 – 22 code program to run the program when the ppm gas exceeds the specified limit or not. Row 11 and 12, code to turn on the led lights. Row 13 to Turn on the Buzzer. Row 14 - 15 code program to print the value of gas ppm and “Gas Bocor” in Serial monitor arduino. Row 19 – 21 code program to Turn off the led lights and buzzer. Row 23 code program to waiting for the time 1 second. Row 1 code program to declare and determine the limit of sensor value.

```

1. const char BotToken[] = "1170607343:AAGk4PY9QkAkHHG1Hd81S8148Yg-
W8tU1rs";
2. #define admin_id "1097071057"

```

Row 1 code program to Declare the Bot Token that is collected from Telegram (@BotFather). Row 2 code program to Declare the id that is collected from telegram (@IDBot : /getid).

```

1.   String gas = "Nilai Gas : ";
2.   gas += int(gas_ppm);
3.   gas += " PPM\n";
4.   gas += "gas Bocor maksimal gaes!\n";
5.   bot.sendMessage(admin_id, gas, "");

```

Row 1-5 code program to Send SMS Notification from Arduino to Bot telegram.

### 5.1.2. Source code from PHP

```

1.<td>' . $row['id'] . '</td>
2.<td>' . $row['Date_time'] . '</td>
3.<td>' . $row['gas_persen'] . '</td>
4.<td>' . $row['gas_ppm'] . '</td>

```

Row 1-4 code program to Display data from Database in tabular form. Row 1 code program to print id. Row 2 code program to display date and time. Row 3 code program to display the gas\_persen. Row 4 code program to display the gas\_ppm.

### 5.1.3. Implementation of Fuzzy Logic Tsukamoto algorithm

1. LPG gas detected by a sensor called MQ-5. Gas set results are obtained as follows:

GAS PPM	Status
$\leq 299$	Safe
$\geq 300$	Leaks
$\geq 800$	Danger

Table 1. Results Rule (Hakim, L. and Yonatan, V., 2017, 'Deteksi Kebocoran gas LPG menggunakan Detektor Arduino dengan Algoritma Fuzzy Logic Mamdani', Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi), Vol 1, No.2.)

2. Based on the results of the above obtained a rule formed consisting of :

- [R1] IF  $160 \geq 294$  THEN Safe
- [R2] IF  $485 \geq 0 \geq 586$  THEN Leaks
- [R3] IF  $1818 \geq 0 \geq 1818$  THEN Danger

3. Display table of sensor value that was analyze with Fuzzy Logic Tsukamoto

254	safe
294	safe
182	safe
177	safe
173	safe
488	Leaks
231	safe
165	safe
261	safe
521	Leaks
485	Leaks
200	safe
166	safe
165	safe
172	safe
170	safe
173	safe
166	safe

Image 13. Display Data Fuzzy

```

1.$gas = mysqli_query($koneksi, "SELECT gas_ppm, case
2.when gas_ppm >= 50 and gas_ppm <= 299 then 'safe'
3.when gas_ppm > 300 and gas_ppm <= 799 then 'Leaks'
4.when gas_ppm > 800 then 'Danger' end as Status FROM feeds");

```

Row 1 – 4 code program php to Display Data Sensor Value and applied the fuzzy logic algorithm. And \$gas in Row 1 code program to declare variable of gas.

## 5.2 Testing

```

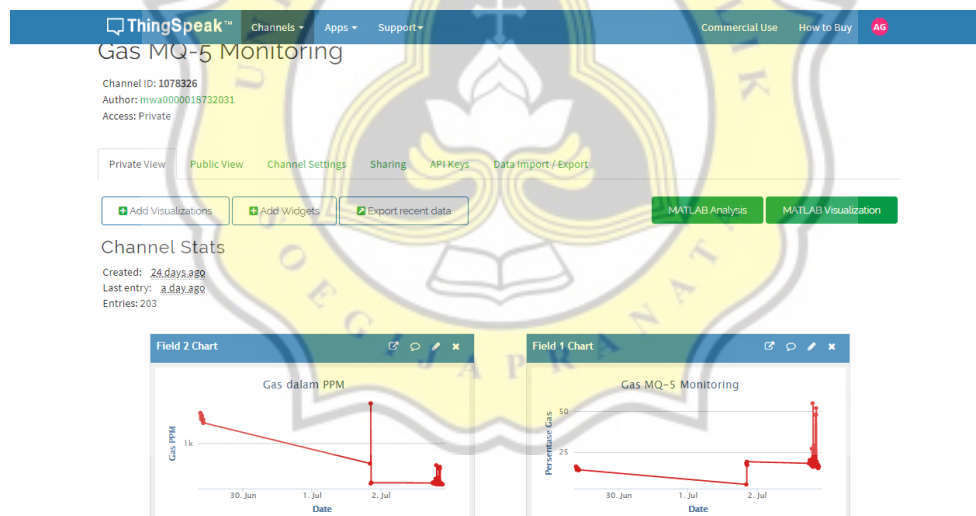
Connecting to: realm 51.Wifi sukses terkoneksi: 7510, IP addr : 192.168.43.86
Gas PPM : 5171.00Gas Persentase : 51.00Connecting to api.thingspeak.com
Permintaan sumber: /update?api_key=
HTTP/1.1 200 OK
Date: Sat, 20 Jun 2020 08:16:19 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 3
Connection: close
Status: 200 OK
X-Frame-Options: SAMEORIGIN
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, PUT, OPTIONS, DELETE, PATCH
Access-Control-Allow-Headers: origin, content-type, X-Requested-With
Access-Control-Max-Age: 1800
ETag: W/"922c7954216ccfe7a61def609305cald"
Cache-Control: max-age=0, private, must-revalidate
X-Request-Id: 62a0ce5a-ad4b-48c8-a8df-88afd7f538fe
X-Runtime: 0.037104
X-Powered-By: Phusion Passenger 4.0.57
Server: nginx/1.9.3 + Phusion Passenger 4.0.57

127
closing connection
6Gas Value: 5024.00
10Gas Value: 5043.00
20Gas Value: 5229.00
30Gas Value: 5102.00
40Gas Value: 5161.00
50Gas Value: 5102.00
60Gas Value: 5180.00
70Gas Value: 5063.00
80Gas Value: 5053.00
90Gas Value: 4965.00
100Gas Value: 5122.00
110Gas Value: 5080.00

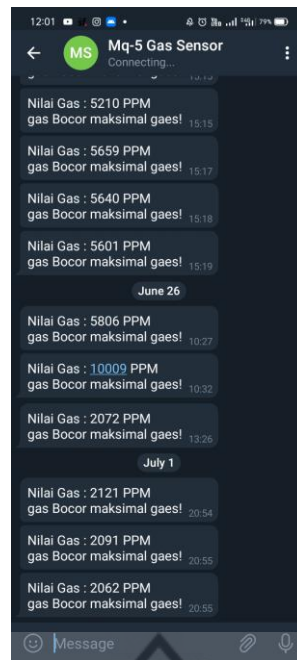
```

Image 14. Serial Monitor Arduino

Testing for code program arduino IDE to running the tools. This tool is successfully connected to the WiFi hotspot.



Testing for Thingspeak to read data from sensor Arduino and displays data in graphical form. The tools of Arduino successfully send data to thingspeak and displayed.



Testing for Telegram messenger to notify the user if there is a gas leakage. The message sent successfully to Telegram from Arduino IDE code. The telegram also reads the amount of PPM read from the sensor.

