CHAPTER 5
IMPLEMENTATION AND TESTING

5.1. Implementation

At this stage describes the implementation of the program in the study which consists of sensors reading the water value, calculating the fuzzy algorithm, turning on the led as output, and finally sending the entire result into the database.

```
1. gravityTds.update();
2. tdsValue = gravityTds.getTdsValue(); // Read the TDS sensor
3. ldrValue = analogRead(ldrPin); //Reading the LDR sensor
```

Code above is the reading of the LDR and TDS sensors, in lines 1 – 2 it is used to read the sensor with the gravityTDS library. And in the second line, it functions to store the value read by the LDR sensor into the ldrvalue variable.

```
1. unsigned char ldrClear(){
2. if (ldr <= 50){
3. clear = 1;
4. cloudy = 0;
5. very cloudy = 0;
6. }
7. clear return;
8. }
9.
10. unsigned char ldrCloudy(){
11. if (50 <= ldr && ldr <= 150){
12. clear = ((150-ldr)/100);
13. cloudy = ((ldr-(200))/100);
14. very cloudy = 0;
15. } else if (150 <= ldr && ldr <= 200){
16. clear = 0;
17. cloudy = ((200-ldr)/150);
18. very cloudy = ((ldr-150)/100);
19. }
20. return cloudy;
21. }
```

```
22.
23. unsigned char ldrVery Cloudy(){
24. if (ldr >= 600){
25. clear = 0;
26. cloudy = 0;
27. very cloudy = 1;
28. }
29. return is very cloudy;
30. }
31.
32. unsigned char tdsLittle(){
33. if (tds <= 300){
34. little = 1;
35. medium = 0;
36. Lots = 0;

37. }
38. return a little;
39. }
40.
41. unsigned char tdsMedium(){
42. if (300 <= tds && tds <= 900){
43. little = ((900-tds)/600);
44. medium = ((tds-300)/600);
45. Lots = 0;
46. } else if (900 <= ldr && ldr <= 1000){
47. little = 0;
48. medium = ((1000-tds)/100);
49. Lots = ((tds-900)/100);
50. }
51. medium return;
52. }
53. unsigned char tdsLots(){
54. if (tds >= 1000){
55. few = 0;
56. moderate = 0;
57. Lots = 1;
58. }
```

```
59. return Lots;
}
```

Code above is the calculation stage of the first fuzzy algorithm, namely fuzzification, where this fuzzification is the process of changing the input with a crisp set into a fuzzy set. Each set has its own range of values according to its parameters. This code is obtained based on the graphs and fuzzy formulas previously discussed. Lines 1-10 are fuzzification for water with clear parameters, lines 11-20 are fuzzification for cloudy water, lines 21-28 are fuzzification for very cloudy water, next is fuzzification of TDS sensor in lines 29-38 for low tds parameters, lines 40-49 for tds parameters medium, and last line 50-56 tds a lot.

```
1. void fuzzy_rule(){
2. fuzzyfication();
3. //if ldrClear and tdsLittle then the water is Feasible
4. rol1 = min(ldrClear(), tdsLittle());
5. z1 = 2;
6. //if ldrClear and the tdsMedium then the water is Medium
7. rol2 = min(ldrClear(), tdsMedium());
8. z2 = 4;
9. //if ldrClear and tdsLots then the water is not Feasible
10. rol3 = min(ldrClear(), tdsLots());
11. z3 = 6;
12.
13. //if ldrCloudy and tdsLittle then the water is Medium
14. rol4 = min(ldrCloudy(), tdsSlightly());
15. z4 = 4;
16. //if ldrCloudy and tdsMedium then the water is Medium
17. rol5 = min(ldrCloudy(), tdsMedium());
18. z5 = 4;
19. //if ldrCloudy and tdsLots then water is not Feasible
20. roll6 = min(ldrCloudy(), tdsLots());
21. z6 = 6;
22.
23. //if ldrVeryCloudy and tdsLittle then the water is not Feasible
24. rol7 = min(ldrverycloudy(), tdsLots());
25. z7 = 6;
26. //if ldrVeryCloudy and tdsMedium then the water is not Feasible
27. roll8 = min(ldrverycloudy(), tdsMedium());
28. z8 = 6;
29. //if ldrVeryCloudy and tdsMedium then the water is not Feasible
30. roll9 = min(ldrvery cloudy(), tdslots());
31. z9 = 6;
```

The second fuzzy stage is making fuzzy rules, where in this study there are 9 possible fuzzy rules. This rule is useful for knowing the relationship between input and output. At this stage, it will look for the MIN value or the smallest value of the membership of the fuzzy variable that has previously been calculated through the fuzzification process. Line 2 is used to call the fuzzification process data. Line 4 to find the smallest value of the fuzzification results between parameters. The 5th line is used to find the output value according to the rule, if it is included in decent water, it will use a predetermined constant value, and so on.

```
1. float defuzzification(){
2. fuzzy_rule();
3.
4.Float                                A                                    =
   ((roll1*z1)+(roll2*z2)+(roll3*z3)+(roll4*z4)+(roll5*z5)+(roll6*z6)+(roll
   7*z7) + (roll8*z8)+(roll9*z9));
5.                  float                  B                  =
(roll1+roll2+roll3+roll4+roll5+roll6+roll7+roll8+roll9);
6. float C = A/B;
7. return C;
}
```

The last is defuzzification, this stage determines the final result of all fuzzy stages to determine the water output, the method used in this Sugeno fuzzy is the defuzzification method of the weighted average method or finding the average.

```
1. if(defuzzification() == 2){
2. digitalWrite(7, HIGH);
3. } else {
4. digitalWrite(7, LOW);
5. }
6.
7. if(defuzzyfication() == 4){
8. digitalWrite(6, HIGH);
9. } else {
10. digitalWrite(6, LOW);
11. }
12.
13. if(defuzzyfication() == 6){
14. digitalWrite(5, HIGH);
15. } else {
16. digitalWrite(5, LOW);
```

```
17. }
```

The code above is a step to turn on the LED lights according to the results obtained from the fuzzy algorithm. This LED light will indicate which level the water quality is at. Rows 1-5 for green LEDs which means the water is of decent quality, rows 6-10 for yellow LEDs where the water is of moderate quality, and finally rows 11-15 for red LEDs which indicate unfit water.

```
1. #include <SPI.h>
2. #include <Ethernet.h>
3.
4. byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
5.
6. char server[] = "192.168.1.7"; // Laptop IP
7. IPAddress ip(192,168,137,177); // IP Ethernet
8. EthernetClient client;
```

In this code, it is a connection process between the server and the client, to send the sensor results on the Arduino into the database and display it on the website. The first line is a special Arduino internal library that handles synchronous serial with the SPI, the second line is the ethernet shield library, the fourth line is the mac address of the ethernet shield, the sixth line is for declaring the server IP that will be used as the address for the sensor result receiver, the eighth line is used for specify the IP client assigned to transfer data.

```
1. if (client.connect(server, 80)) {
2. Serial.println("");
3. Serial.println("connected");
4. // Make a HTTP request:
5.  Serial.print("GET  /arduino_mysql/sensor.php?dataLDR=");
6. client.print("GET /arduino_mysql/sensor.php?dataLDR=");
7. Serial.print(ldrValue);
8. client.print(ldrValue);
```

The first line in the code above is used to determine whether the client is connected to the server on port 80, the second line gives a space, the third line determines if the client is connected it will display a connected notification, the fifth and sixth lines are the client's job to send data to the database.

```
1. function connect(){
2.  $this->link  =  mysqli_connect('localhost','root','')  or  die('Cannot
connect to the DB');
3.  mysqli_select_db($this->link,'sensor_air')  or  die('Cannot  select  the
DB');
4. }
```

The code above is the step to enter the database, where in line 1 is a function to connect

to the database, line 2 is to enter with MySql server on localhost, the fourth line is to enter the database in question.

```
1. function storeInDB($ldr, $tds, $defuzzification){
2. $query = "insert into data_sensor set ldr='".$ldr."', tds='".$tds."',
defuzzification=' ".$defuzzify."'";
3. $result = mysqli_query($this->link,$query) or die('Errant query:
'.$query);
4. }
```

In lines 1 – 3 is the code to create a function that will connect to the database and enter the sensor results into a table in the database.

```
5. if($_GET['dataLDR'] != '' and $_GET['dataTDS'] != '' and
$_GET['defuzzification'] != ''){
6. $air=new air($
_GET['dataLDR'],$_GET['dataTDS'],$_GET['defuzzification']); 7. }
```

In lines 1 – 3 is the process to retrieve sensor data from Arduino into the database.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta http-equiv="refresh" content="5">
5. </head>
6. <body>
7. <style>
8. #wntable {
9. border-collapse: collapse;
10.width: 50%;
11. }
12.
13. #wntable td, #wntable th {
14. border: 1px solid #ddd;
15. padding: 8px;
16. }
17.
18. #wntable tr:nth-child(even){background-color: #f2f2f2;} 19.
20. #wntable tr:hover {background-color: #ddd;}
21.
22. #wntable th {
23. padding-top: 12px;
24. padding-bottom: 12px;
```

```
25. text-align: left;
26. background-color: #00A8A9;
27. color: white;
28. }
29. </style>
30.
31. <div id="cards" class="cards" align="center">
32. <h1> Data Sensor Air Bersih dan Air Keruh </h1> 33.
<table id="wntable">
34. <tr>
35. <th>NO</th>
36. <th>LDR</th>
37. <th>TDS</th>
```



```
38. <th>DEFUZZIFIKASI</th>
39. <th>Waktu</th>
40. </tr>
41. <?php
42.
43. $sql = mysqli_query($koneksi, "SELECT * FROM dht_data ORDER BY id DESC");
44.
45. if(mysqli_num_rows($sql) == 0){
46. echo '<tr><td colspan="14">Data Tidak Ada.</td></tr>';
47. }else{
49. while($row = mysqli_fetch_assoc($sql)){
50. echo '
51. <tr>
52. <td>'.$no.'</td>
53. <td>'.$row['ldr'].'</td>
54. <td>'.$row['tds'].'</td>
55. <td>'.$row['defuzzifikasi'].'</td>
56. <td>'.$row['waktu'].'</td>
57. </tr>
58. ';
59. $no++;  }
61. }
62. ?>
63. </table>
64. </div>
65. </body>
```

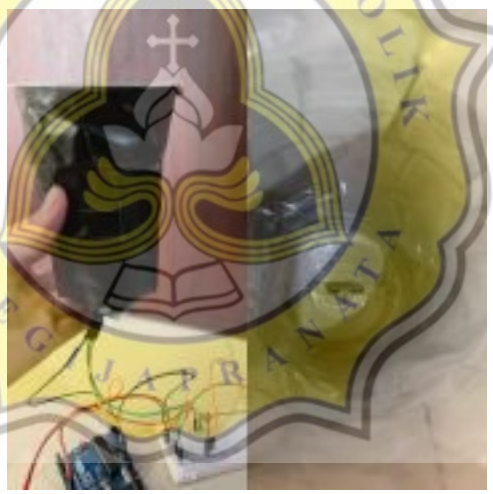
```
66. </html>
```

Tahap terakhir adalah membuat page untuk website, dimana website ini akan menampilkan hsil dari database. Pada baris 4 bertugas untuk memuat ulang web setiap 5 detik, baris 7 – 9 merupakan code untuk membuat table serta memberi warna, baris 31 – 41 membuat nama atau lable pada table, baris 43 mengambil data dari table yang berada di database, baris 45 – 47 memberi pemeberitahuan jika data tidak ada pada database maka di website juga menampilkan data tidak tersedia dan sebaliknya, baris 48 setiap data yang baru masuk akan dimulai dari nomor 1 atau paling atas pada table, baris 49 – 58 untuk memasukkan function dari database kedalam table.

5.2 Testing

At this stage the results of the test start from the Arduino connected to the internet, the sensor reads the water, performs the fuzzy logic process, the sensor results enter the database, and displays it on the web.

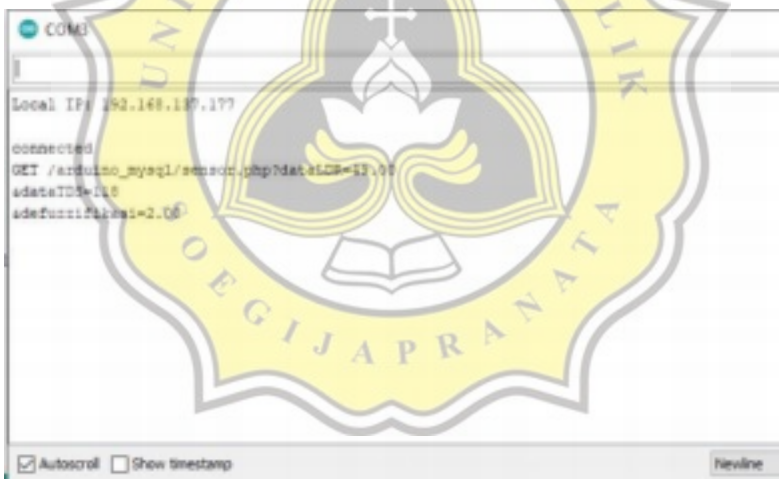
1. The LDR sensor is attached to the bottom of the container, then the water sample in the cup is inserted into the container, and the TDS sensor is inserted directly into the water, from the top it will be given a flash.



Ilustration 5.2.1 LDR Sensor

Ilustration 5.2.2 TDS and LDR sensor

2. After the upload process on the Arduino is complete, the serial monitor will display a notification that the Arduino has successfully connected to the internet, and the sensor results have been obtained.



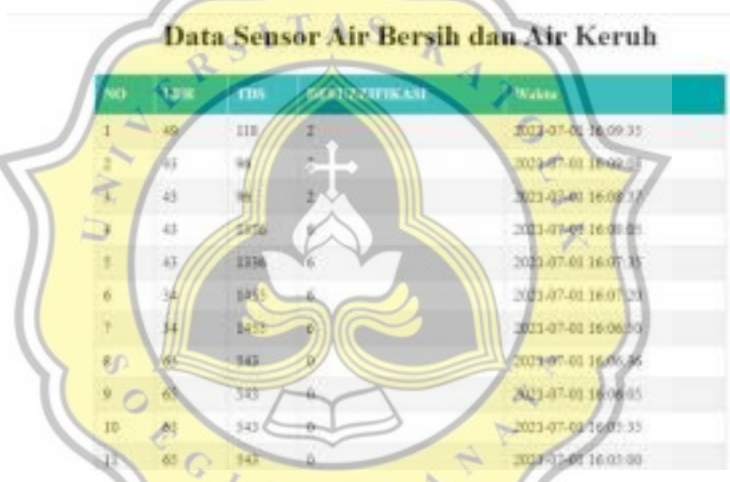
Ilustration 5.2.3 : Serial Monitor Arduino IDE

3.After Arduino gets the sensor results and successfully connects to the internet, the data will automatically be sent directly to the specified database.

	id	waktu	ldr	tds	defuzzifikasi
Edit Copy Delete	99	2021-07-01 16:06:36	65	543	0
Edit Copy Delete	100	2021-07-01 16:06:50	34	1455	6
Edit Copy Delete	101	2021-07-01 16:07:20	34	1455	6
Edit Copy Delete	102	2021-07-01 16:07:35	43	1336	6
Edit Copy Delete	103	2021-07-01 16:08:05	43	1336	6
Edit Copy Delete	104	2021-07-01 16:08:37	43	96	2
Edit Copy Delete	105	2021-07-01 16:09:08	43	96	2
Edit Copy Delete	106	2021-07-01 16:09:35	49	118	2

Ilustration 5.2.4 : database

4. From the data that has been entered into the database, it will then be displayed on the web which will display the results of the water from the LDR and TDS sensors.



Ilustration 5.2.5 : Water Sensor Data Web View

5.2.1 Fuzzy Logic Testing Sugeno

In this process, the fuzzy logic is tested manually, where the author uses the results from the sensor as follows: LDR sensor = 49, TDS sensor = 118.

The first step is to determine the membership value using the formula:
Graphic_Up = (value – min) / (max – min);
Graph_Down = (max – value) / (max – min);

Judging from the degree of LDR membership, the ldr value of 1015 is included in the clear ldr membership degree, and the tds value of 118 is included in the low membership degree.

Fuzzifikasi

LDR degree membership value:

- MIN = 50
- MAX = 150
- Value = 49
- Graphic_Down = Clear
- Clear = (150 – 49) / (150 – 50)
  = 101 / 100
  = 1.01

The value of 1.01 is in the clear membership degree, then the moderate and very cloudy membership degree is 0.

Then for the 118 tds value to be in the little tds membership degree, then the calculation of the tds membership value is as follows:

TDS degree membership value :

- MIN = 500
- MAX = 600
- Value = 118
- Graphic_Down = little
- little = (600 – 118) / (600 – 500)
  = 482 / 100
  = 4.82

Because the value of 118 is in the little membership degree, the moderate and multiple membership degrees are 0.

Inference

After completing the fuzzification, then carry out the implication process using the MIN of the predetermined rules.

1. IF ldrClear and tdsLittle Then water Feasible
   α = MIN(1.01 ; 4.82) = 1.01
   z = 2
2. IF ldrClear and tdsMedium Then water Medium
   α = MIN(0;0) = 0
   z = 0
3. IF ldrClear and tdsLots Then water Not Feasible
   α = MIN(0;0) = 0

z = 0

4. IF ldrCloudy and tdsLittle Then water Medium

α = MIN(0;0) = 0

z = 0

5. IF ldrCloudy and tdsMedium Then water Medium

α = MIN(0;0) = 0

z = 0

6. IF ldrCloudy and tdsLots Then water Not Feasible

α = MIN(0;0) = 0

z = 0

7. IF ldrVeryCloudy and tdsLittle Then water Not Feasible

α = MIN(0;0) = 0

z = 0

8. IF ldrVeryCloudy and tdsMedium Then water Not Feasible

α = MIN(0;0) = 0

z = 0

9. IF ldrVeryCloudy and tdsLots Then water Not Feasible

α = MIN(0 ; 0) = 0

z = 0

Defuzzification

The next step is to calculate the defuzzification using the formula to find the average (average) with the formula:

$$Z = \frac{\sum \alpha i z i}{\sum \alpha i}$$

It can be seen in the calculation of each fuzzy rule, then and z which have a value are in rule 1.

$$Z = \frac{1.01 \times 2}{1.01}$$

$$= 2$$

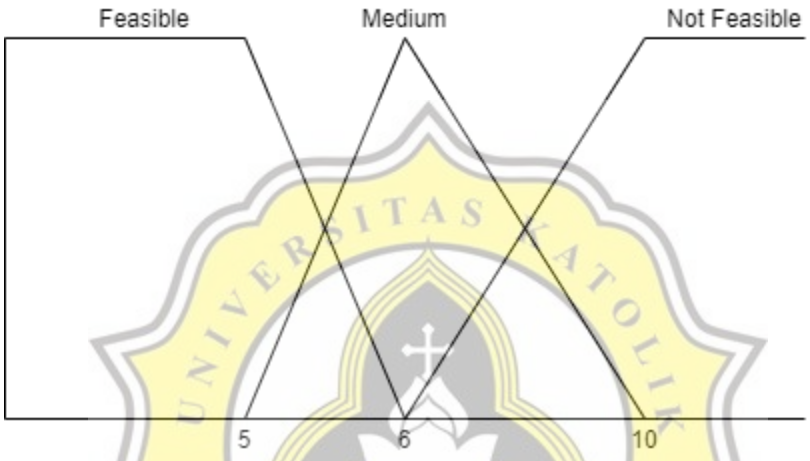
5.2.2 Fuzzy Logic Testing Mamdani

In this method, it is different from the Sugeno fuzzy method, where in the Mamdani method the output used is in the form of a fuzzy set calculation value. The water quality output has a fuzzy set in the form of a specified range of values.

Feasible = 0 – 5

Medium = 5 – 10

Not Feasible = 10



Ilustration 5.2.6 : Output membership degree graph

layak 1 , x <= 5

c - x / c - b , b <= x <= c

6 - x / 6 - 5 , 5 <= x <= 6

6 - x / 1 , 5 <= x <= 6

Sedang x - a / b - a , a <= x <= b

x - 5 / 6 - 5 , 5 <= x <= 6

x - 5 / 1 , 5 <= x <= 6

c - x / c - b , b <= x <= c

10 - x / 10 - 6 , b <= x <= 10

m

10 - x / 4 , 6 <= x <= 10

banyak x - a / b - a , a <= x <= b

x - 6 / 10 - 6 , 6 <= x <= 10

x - 6 / 4 , 6 <= x <= 10

1 , x >= 10

With an example calculation:

Rule 1: If ldr is clear and tds is little then water is feasible = 1.01

Defuzzification =

Z = 6 – x / 1 = 1.01

= 6 – (1*1.01)

= 5 (feasible)

The following are the experimental results of 10 water samples, using the Sugeno and Mamdani methods. Then use the TDS Meter as a tool that measures water quality in addition to sensors. The data obtained in this study include the following table:

Table 5.1 Testing

Water Sample	**TDS Meter**	**Reference**	**Sugeno**	**Mamdani**
Aquades	92	Feasible	Feasible	Feasible
Faucet Water	259	Feasible	Medium	Medium
Salt Water	1843	Not Feasible	Not Feasible	Not Feasible
Rice Water	420	Medium	Medium	Medium
Tea	685	Medium	Medium	Medium
Orange Water	383	Medium	Medium	Medium
Coca-Cola	493	Medium	Medium	Medium
Milk	2161	Not Feasible	Not Feasible	Not Feasible

Coffee	1717	Not Feasible	Not Feasible	Not Feasible
Flour Water	522	Medium	Not Feasible	Not Feasible

Reference to this testing uses the World Health Organization guidelines on water quality guidelines as follows:

Organoleptic properties

The presence of dissolved solids in water may affect its taste (*1*). The palatability of drinking-water has been rated by panels of tasters in relation to its TDS level as follows: excellent, less than 300 mg/litre; good, between 300 and 600 mg/litre; fair, between 600 and 900 mg/litre; poor, between 900 and 1200 mg/litre; and unacceptable, greater than 1200 mg/litre (*1*). Water with extremely low concentrations of TDS may also be unacceptable because of its flat, insipid taste.

Ilustration 5.2.7 water quality guidelines according to WHO

Ilustration 5.2.8 : TDS&EC Meter