

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

This project use Arduino IDE application to write and upload the program to WeMos D1 R2. The written program can be seen below :

5.1.1 Arduino IDE code fix.ino

1. First, declare libraries that needed for wifi, mqtt, measure temperature and D18B20 temperature, there are :

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
```

2. There are the pin we use on WeMos D1 R2 :

```
#define ONE_WIRE_BUS D7 = pin for D18B20 temperature sensor
#define relay D4 = pin for relay
#define trigPin D5 = pin for ultrasonic trigger
#define echoPin D6 = pin for ultrasonic echo
#define ldr A0 = pin for light dependent resistor
```

3. Then, set the ssid, password, and ip address to connect network and create a function to connect network :

```
const char* ssid = "ferdi";
```

```
const char* password = "12341234";
```

```
const char* mqtt_server = "192.168.43.43";
```



4. Then, create a function to receive command from node-red.

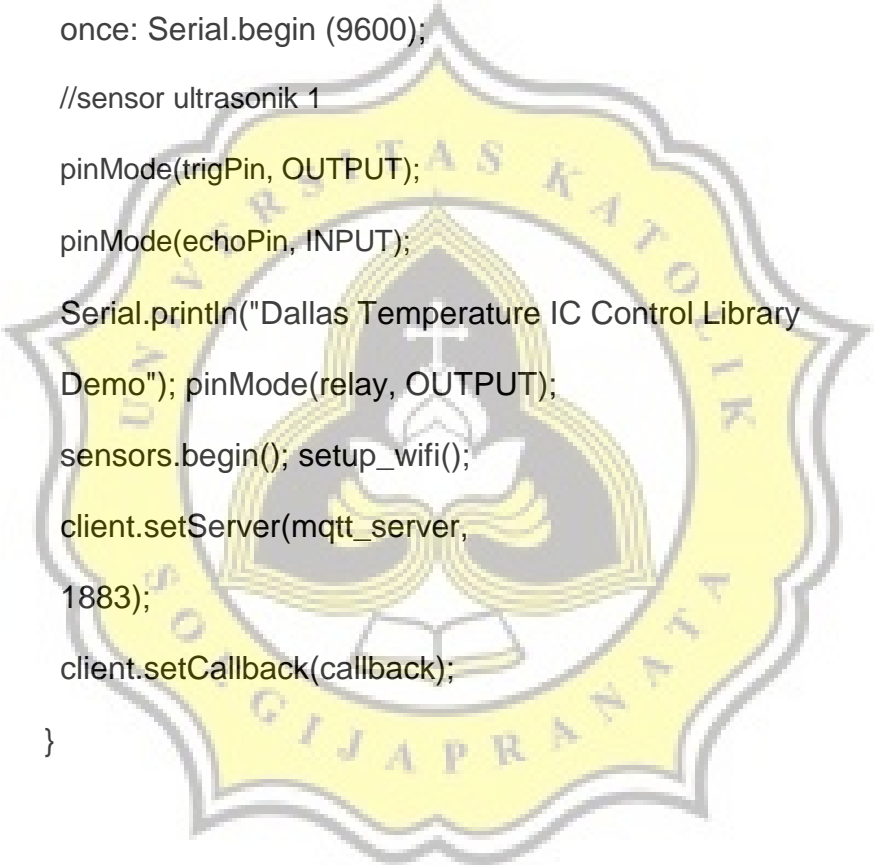
```
void callback(String topic, byte* message, unsigned int length)
```

In this program, this function use to subscribe command from mqtt, by node red program which is decide whether the relay on or off according to optimum temperature. Node red program will publish "1" if the temperature under 32°C, and publish "0" if below 32°C. If this function received value "1", this function will set on the relay, and if "0" the function will set off the relay.

```
if(topic=="penghangat"){  
    Serial.print("Changing Room lamp to ");  
    if((char)message[0] == '1'){  
        digitalWrite(relay, LOW);  
        Serial.print("on");  
        client.publish("CekTerima1", "ON");  
        client.publish("CekTerima", "1");  
    }  
    else if((char)message[0] == '0'){  
        digitalWrite(relay, HIGH);  
        Serial.print("off");  
    }  
}
```

5. create a function to setup the program. In this program we need to set the trigger on ultrasonic as output , and echo as input. Relay set as output, and it will receive command from program.

```
void setup() {  
    // put your setup code here, to run  
    once: Serial.begin (9600);  
    //sensor ultrasonik 1  
    pinMode(trigPin, OUTPUT);  
    pinMode(echoPin, INPUT);  
    Serial.println("Dallas Temperature IC Control Library  
    Demo"); pinMode(relay, OUTPUT);  
    sensors.begin(); setup_wifi();  
    client.setServer(mqtt_server,  
    1883);  
    client.setCallback(callback);  
}
```

The watermark is a yellow shield-shaped logo with a scalloped border. Inside the shield, there is a stylized white and grey emblem featuring a book and a flame. The text "UNIVERSITAS KATOLIK" is written in a semi-circle at the top, and "GIJAPRANATA" is written in a semi-circle at the bottom.

6. Function loop use to keep the program running. Program will keep running and read ultrasonic sensor, DS18B20 sensor and light dependent resistor. The functions to read sensors will keep running by loop function.

```
void loop() {  
    // put your main code here, to run  
    repeatedly: Sensorpertama();  
    Sensorsuhu();  
    cekcahaya();  
    if (!client.connected())  
    { reconnect();  
    }  
    if(!client.loop())  
    client.connect("ESP8266Client");  
    long now = millis();  
    now = millis();  
    // Publishes new value from sensors every 1  
    second if (now - lastMsg > 1000) {  
        lastMsg =  
        now; return;  
    }  
}
```

7. Create a function to read ultrasonic sensor, then publish to mqtt.

Node red program will subscribe, and then measure the feeder capacity

```
void Sensorpertama(){  
    // put your main code here, to run  
    repeatedly: long duration, distance;  
    digitalWrite(trigPin, LOW); // Added this line  
    delayMicroseconds(2); // Added this line  
    digitalWrite(trigPin, HIGH);  
    //delayMicroseconds(1000); - Removed this  
    line delayMicroseconds(10); // Added this  
    line digitalWrite(trigPin, LOW);  
    duration = pulseIn(echoPin, HIGH);  
    distance = (duration/2) / 29.1;  
    int a = distance;  
  
    dataultra1convert = a;  
    //convert dari float ke string  
    dtostrf(dataultra1convert, 6, 2, dataultra1 kirim);  
    client.publish("Sensor1", dataultra1 kirim);  
}
```

8. Create a function to read light dependent resistor, then publish it to mqtt, and then node red program will subscribe and by the data received, node red program will check whether the lamp work properly or not.

```
void cekcahaya() {  
    int a = analogRead(ldr);  
    static char cahaya[100];  
    dtostrf(a, 6, 2, cahaya);  
    client.publish("cekcahaya", cahaya);  
    Serial.println(a);  
}
```



9. Create a function to read DS18B20 temperature sensor, the the data will publish to mqtt and subscribe by node red program.

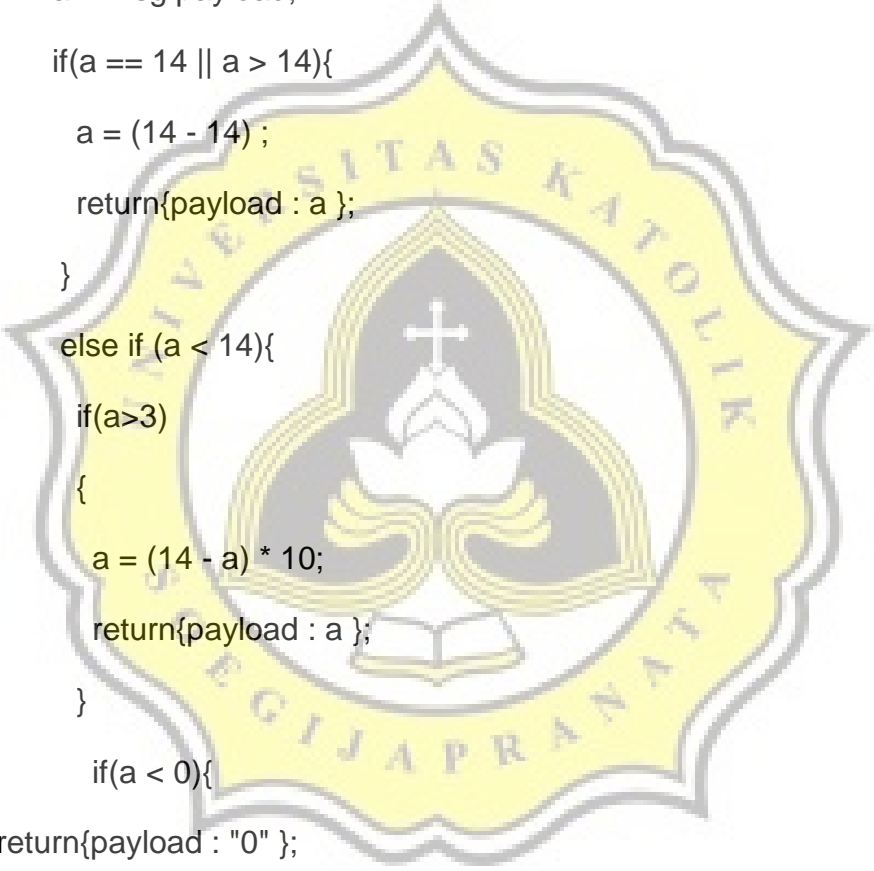
```
void Sensorsuhu(){  
    Serial.print(" Requesting temperatures...");  
    sensors.requestTemperatures(); // Send the command to get temperatures  
    Serial.println("DONE");  
    Serial.print("Temperature is: ");  
    Serial.print(sensors.getTempCByIndex(0)); // Why "byIndex"?  
    // You can have more than one IC on the same bus.  
    // 0 refers to the first IC on the wire  
    //delay(100);  
    datasuhuconvert = sensors.getTempCByIndex(0);  
    dtostrf(datasuhuconvert, 6, 2, datasuhukirim);  
    client.publish("Sensor3", datasuhukirim);  
}
```


5.2.2 Node-Red Code

1. Function food measurement

This function will subscribe distance value that published by mqtt from WeMos program to measure feeder capacity

```
a = msg.payload;
if(a == 14 || a > 14){
  a = (14 - 14) ;
  return{payload : a };
}
else if (a < 14){
  if(a>3)
  {
    a = (14 - a) * 10;
    return{payload : a };
  }
  if(a < 0){
    return{payload : "0" };
  }
}
```

The logo of Universitas Katolik Regi Japranata is a circular emblem with a yellow background and a grey border. It features a central white cross above a white lotus flower, which is set against a dark grey background. The text "UNIVERSITAS KATOLIK" is written in a semi-circle at the top, and "REGI JAPRANATA" is written in a semi-circle at the bottom.

```

}
else if(a > 100){
return{payload : "100" };
}
}

```

2. Function lamp status

This function will check whether the lamp is work properly or not by subscribe the relay status and light intensity that published by mqtt, from WeMos program. Then this function will compare those value, for example : if relay status is on and light intensity is below the threshold, it means the lamp didn't work properly.

```

context.node = context.node || 0;
context.node1 = context.node1 || 0;
if(msg.topic == "CekTerima"){
    context.node = msg.payload;
} else if(msg.topic=="cekcahaya"){
    context.node1 = msg.payload;
}

if(context.node == 1 && context.node1 < 150 ){
    return{topic: "d", payload: "lamp didn't work"}
}

else if(context.node == 0){

```

3. Function Control Relay

This function control relay whether it on or off according to temperature value that subscribed by mqtt, from WeMos program. If temperature value received under 32°C , this function will publish value that received by WeMos program to set the relay on.


```
a = msg.payload;  
if(a < 32){  
  return{payload:"1"};  
}  
else{  
  return{payload:"0"};  
}
```



5.2 Testing

5.2.1 Reading data test

When WeMos turn on, the program will run and request an IP from network and connect to mqtt server. Then the program begin to read data from sensors (as seen in illustration 5.2.1), they are : light dependent resistor, ultrasonic sensor, and DS18B20 temperature sensor. After read data from these sensors, the program publishes the data to mqtt.

The image shows a terminal window titled "/dev/ttyUSB0" with a background watermark of the UIN Ar-Ranata logo. The terminal output displays the following text:

```
load 0x4010f000, len 1384, room 16  
tail 8  
chksum 0x2d  
csum 0DONE  
Temperature is: 30.31128  
Attempting MQTT connection...connected  
Sisa Makanan tersisa 100%  
Requesting temperatures...DONE  
Temperature is: 30.31128  
Sisa Makanan tersisa 100%  
Requesting temperatures...DONE  
Temperature is: 30.31129  
Sisa Makanan tersisa 100%  
Requesting temperatures...DONE  
Temperature is: 30.31128  
Sisa Makanan tersisa 100%  
Requesting temperatures...DONE  
Temperature is: 30.31128  
Sisa Makanan tersisa 100%  
Requesting temperatures...DONE  
Temperature is: 30.31128  
Sisa Makanan tersisa 100%  
Requesting temperatures...DONE  
Temperature is: 30.31129
```

At the bottom of the terminal window, there is a checkbox labeled "Autoscroll" which is currently unchecked.

Illustration 5.2.1

5.2.2 Node-Red work flow and Interface test

The Node-Red subscribe the data that published by program on, then several data will process into the function on Node-Red, they are Feeder measurement, LampStatus and ControlRelay. After that, node-red provide an interface as seen in *illustration 5.2.2*. Node-Red function flow can be seen in *illustration 5.2.3*.

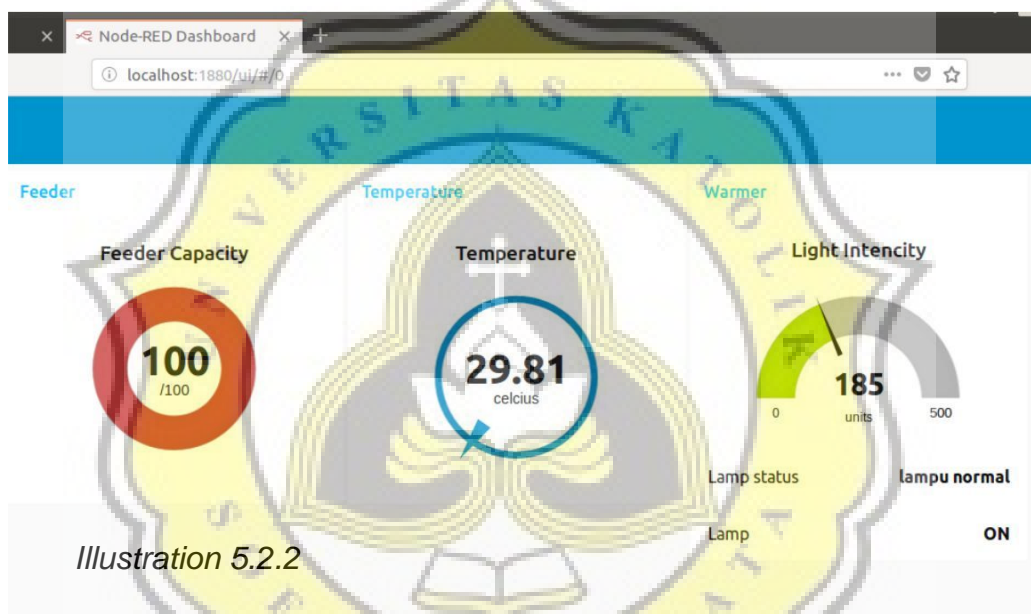


Illustration 5.2.2

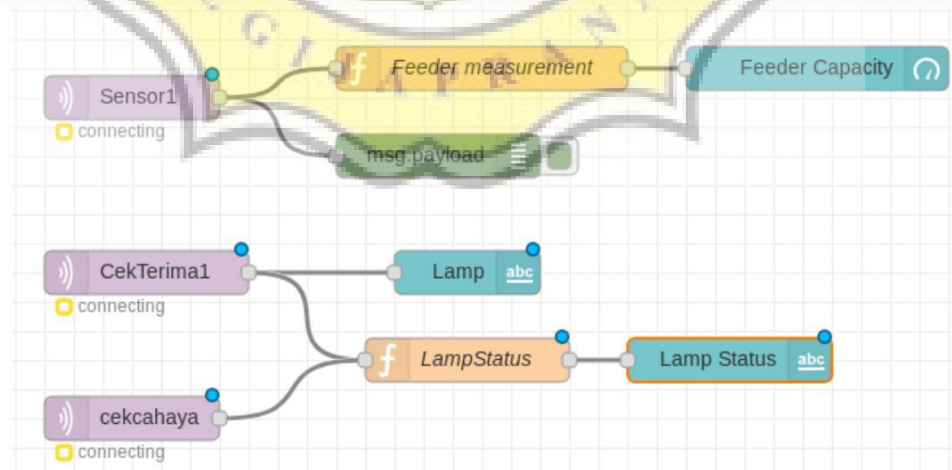


Illustration 5.2.3.

if the lamp is turn on and the light intencity is high, the lamp status showing “lampu normal”. If the lamp is on, but the light intencity is low, the lamp status showing “lampu rusak” as seen in *illustration 5.2.4*.

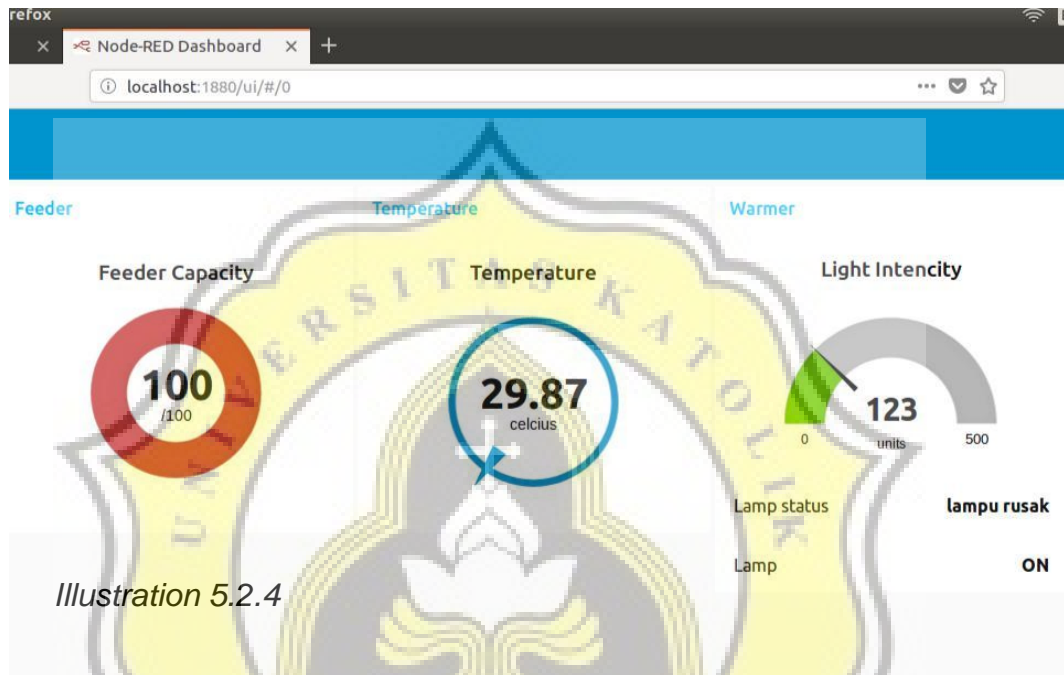


Illustration 5.2.4

Node-Red also subscribe temperature data published by program in WeMos, then provide an interface to monitor, and also the temperature data proceed to Node-Red function named ControlRelay, This function use to send a value that subscribed by WeMos program to determine whether the lamp is on or of depend on current temperature. Node-Red ControlRelay function flow can be seen in *illustration 5.2.5*.

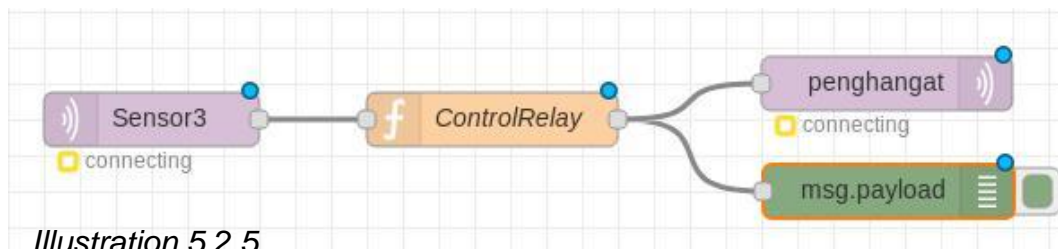


Illustration 5.2.5

5.2.3 Installation

The whole project installation figured out in illustration 5.2.6. Warmer lamp placed on the top of the cage beside LDR sensor. temperature sensor placed on the top of the cage, far from warmer lamp. Feeder placed in corner side of the cage. Ultrasonic sensor placed on the top of the feeder. Temperature sensor WeMos, relay and breadboard installed in the cage roof.



Illustration 5.2.6