

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **1. 1 Kesimpulan**

Setelah menyelesaikan pembahasan yang sudah di uraikan, maka dalam kesimpulan prototype suhu dan kelembaban ini dapat dibuat dengan menggunakan modul ESP32 dengan sensor DHT11 dan menggunakan arduino IDE sebagai pembuatan programnya sehingga dalam prototype monitoring ini dapat digunakan alatnya, sehingga alat ini dapat memberikan juga notifikasi lewat aplikasi telegram sehingga memudahkan untuk mengetahui suhu dan kelembaban tersebut.

Dalam suhu dan kelembaban ini juga terdapat beberapa kendala seperti dalam penggunaan alat DHT11 kadang error dan Ketika jaringan kurang stabil maka suhu dan kelembaban ruangan itu akan kurang efisien hasilnya.

#### **1. 2 Saran**

pada hasil penelitian ini masih jauh dari kata sempurna, oleh karena itu masih dibutuhkan kritik dan saran untuk membangun/proses berjalannya prototype.



## DAFTAR PUSTAKA

- Arifianto, D., Sulistyono, A., & Nilogiri, A. (2022). Sistem Monitoring Suhu Dan Kelembaban Ruangan Server Berbasis Arduino Menggunakan Metode Fuzzy Logic Dengan Buzzer Dan Telegram Bot Sebagai Notifikasi. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 7(1), 67–75. <https://doi.org/10.32528/justindo.v7i1.5135>
- Cartika, I., Suwarni Tri Rahayu, Rofik Sinung Basuki, & Thomas Agoes Soetiarso. (2022). Pertumbuhan dan Hasil Tanaman Bawang Putih pada Berbagai Penambahan Lama Penyinaran Lampu LED Putih. *Jurnal Agronomi Indonesia (Indonesian Journal of Agronomy)*, 50(1), 57–64. <https://doi.org/10.24831/jai.v50i1.39300>
- D, I. L. H., Sofwan, A., Rosnelly, R., Wardoyo, R., Rogério dos Santos Alves; Alex Soares de Souza, et al, SUCIANTINI, S., Degei, F. M., Tanaamah, A. R., Wowor, A. D., Informasi, F. T., Kristen, U., Wacana, S., & Diponegoro, J. (2015). Penerapan Fuzzy Logic Pada Sistem Pengaturan Jumlah Air Berdasarkan Suhu Dan Kelembaban. *Seminar Nasional Informatika (SemnasIF 2011)*, 3(April), 358–365. <http://biodiversitas.mipa.uns.ac.id/M/M0102/M010232.pdf>
- Dzulkiflih, D., & Khansa, F. K. (2022). Rancang Bangun Perangkat Nyamuk Otomatis Menggunakan Sensor Suhu Dan Kelembaban Dht11 Berbasis Arduino Uno. *Inovasi Fisika Indonesia*, 11(2), 28–37. <https://doi.org/10.26740/ifi.v11n02.p28-37>
- Hj. A. Irmayani p , Asrul, M. N. kaliky. (2020). Desain Bangun Ayakan Alat Mesin Tanaman Perkebunan. *Jurnal Telekomunikasi, Kendali Dan Listrik*, 2(1), 12–22.
- Ibrahim, M., & Sugiarto, B. (2023). Rancang Bangun Rumah Pintar (Smart Home) Berbasis Internet Of Things (IoT). *Infotek : Jurnal Informatika Dan Teknologi*, 6(1), 1–10. <https://doi.org/10.29408/jit.v6i1.5365>
- Ichsan, M. (2020). *VoteTEKNIKA*. 8(4).
- Khalifa, A. A. M., & Prawiroredjo, K. (2022). Model Sistem Pengendalian Suhu dan Kelembaban Ruangan Produksi Obat Berbasis NodeMCU ESP32. *Jurnal ELTIKOM*, 6(1), 13–25. <https://doi.org/10.31961/eltikom.v6i1.415>
- Mulyanto, A. D. (2020). Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian. *Matics*, 12(1), 49. <https://doi.org/10.18860/mat.v12i1.8847>
- Prayitno, E., Juliasari, N., & Ariyani, P. F. (2019). Monitoring Dan Pengontrolan Suhu Serta Kelembaban Penyimpanan Bahan Makanan Berbasis Web Dengan Metode. *Proceeding Sintak 2019*, 236–241.
- Sunanto, S., Firdaus, R., & Makmur Setiawan Siregar. (2021). Implementasi Logika Fuzzy Mamdani Pada Kendali Suhu dan Kelembaban Ruang Server. *Jurnal CoSciTech (Computer Science and Information Technology)*, 2(2), 128–136. <https://doi.org/10.37859/coscitech.v2i2.3362>

## INISIASI KODE PROGRAM

```
#include <LiquidCrystal_I2C.h>

#include <DHT.h>

#include <DHT_U.h>

#include <Fuzzy.h>

#include <WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#define WIFI_SSID "Gunungsitoli"
#define WIFI_PASSWORD "april2017"
#define BOT_TOKEN "6020026630:AAGUue5IMDTvK6Jb2Swt5fLTc7g2XB7ND0"

//krisDevice_bot

#define DHTTYPE DHT11
#define pinDHT1 5

const unsigned long BOT_MTBS = 1000; // mean time between scan messages

WiFiClientSecure secured_client;

UniversalTelegramBot bot(BOT_TOKEN, secured_client);

unsigned long bot_lasttime; // last time messages' scan has been done

int fanStatus = 2;

int buzzStatus = 0;
```

```
//-----INISIALISASI KEANGGOTAAN FUZZY
```

```
Fuzzy *fuzzy = new Fuzzy();
```

```
// FuzzyInput
```

```
FuzzySet *tempNormal1 = new FuzzySet(0, 0, 32, 35);
```

```
FuzzySet *tempHangat1 = new FuzzySet(32, 35, 35, 38);
```

```
FuzzySet *tempPanas1 = new FuzzySet(35, 38, 100, 100);
```

```
// FuzzyInput
```

```
FuzzySet *humiRendah1 = new FuzzySet(0, 0, 50, 70);
```

```
FuzzySet *humiNormal1 = new FuzzySet(50, 70, 70, 90);
```

```
FuzzySet *humiTinggi1 = new FuzzySet(70, 90, 100, 100);
```

```
// FuzzyOutput
```

```
FuzzySet *normal = new FuzzySet(0, 0, 25, 50);
```

```
FuzzySet *warning = new FuzzySet(25, 50, 50, 75);
```

```
FuzzySet *danger = new FuzzySet(50, 75, 100, 100);
```

```
//-----
```

```
const int pinYellow=12, pinRed=13, pinGreen=14, pinBlue=27;
```

```
const int pinFan=25,pinBuzzer=26;
```

```
//1854456707
```

```
//0=normal, 1=warning, 2=danger, 3=error
```

```
unsigned int valueStatusResult=0,lastValueStatusResult=0;
```

```
unsigned int valueStatus1=0;
```

```
float valueTemp1=0,valueHumi1=0,valueDefuzzy=50;
```

```

String status1,statusResult;

unsigned long lastTime=0;

DHT dhtRoom1(pinDHT1,DHTTYPE);

LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  Serial.begin(115200);
  setupFuzzy();
  dhtRoom1.begin();
  lcd.init();
  // lcd.begin();
  lcd.backlight();

  pinMode(pinYellow,OUTPUT);pinMode(pinRed,OUTPUT);
  pinMode(pinGreen,OUTPUT); pinMode(pinBlue,OUTPUT);
  pinMode(pinFan,OUTPUT); pinMode(pinBuzzer,OUTPUT);
  digitalWrite(pinYellow,HIGH);digitalWrite(pinRed,HIGH);
  digitalWrite(pinGreen,HIGH);digitalWrite(pinBlue,HIGH);
  digitalWrite(pinFan,HIGH);digitalWrite(pinBuzzer,HIGH);

  Serial.print("Connecting to Wifi SSID ");

  Serial.print(WIFI_SSID);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for
api.telegram.org

  while (WiFi.status() != WL_CONNECTED)
  {

```



```

Serial.print(".");
delay(500);
}
Serial.println(WiFi.localIP());
}
void loop() {
if(WiFi.status() != WL_CONNECTED)
{
Serial.print("Disconnected. Reconnecting to Wifi SSID ");
Serial.print(WIFI_SSID);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED){
Serial.print(".");
delay(500);
}
}
cekTelegram();
readSensor();
actionFuzzy();
displayValue();
 kirimNotif();
delay(100);
}

void kirimNotif(){
if(lastValueStatusResult!=valueStatusResult && valueStatusResult==3){
String pesan = "ALAT ERROR";

```



```

    bot.sendMessage("1854456707", pesan, "");

}

if(lastValueStatusResult!=valueStatusResult && valueStatusResult==2){

    String pesan = "STATUS DANGER \n";

    pesan+= "SUHU \t: " + String(valueTemp1) + " Celcius\n";

    pesan+= "Humidity\t: " + String(valueHumi1) + " %\n";

    bot.sendMessage("1854456707", pesan, "");

}

}

void cekTelegram(){

    if (millis() - bot_lasttime > BOT_MTBS){

        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        while (numNewMessages)

        {

            Serial.println("got response");

            handleNewMessages(numNewMessages);

            numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        }

        bot_lasttime = millis();

    }

}

void readSensor(){

    valueTemp1 = dhtRoom1.readTemperature();

    valueHumi1 = dhtRoom1.readHumidity();

}

```



```

void displayValue(){
    lcd.setCursor(0,0);lcd.print("TEMP :  `C");
    lcd.setCursor(0,1);lcd.print("HUMI :  %");
    lcd.setCursor(9,0);lcd.print(valueTemp1);
    lcd.setCursor(9,1);lcd.print(valueHumi1);

    Serial.println("-----");
    Serial.print("Temperature 1 : ");Serial.print(valueTemp1);
    Serial.print(" C\tHumi 1 : ");  Serial.println(valueHumi1);
    Serial.print("Status : ");    Serial.print(statusResult);
    Serial.print("\tResult : ");  Serial.println(valueDefuzzy);
    Serial.println("-----");
}
void actionFuzzy(){
//-----FUZZIFIKASI TEMPERATURE 1 & TEMPERATURE 2 TO 0-100-----
    if (isnan(valueTemp1)){
        fuzzy->setInput(1, 0);
        fuzzy->setInput(2, 0);
    }
    else {
        fuzzy->setInput(1, valueTemp1);
        fuzzy->setInput(2, valueHumi1);
    }

    fuzzy->fuzzify();
    valueDefuzzy = fuzzy->defuzzify(1);
    lastValueStatusResult=valueStatusResult;

```





```

if (isnan(valueTemp1)) valueStatusResult = 3;
else if (valueDefuzzy >= 62.5)      valueStatusResult = 2;
else if (valueDefuzzy >= 37.5)     valueStatusResult = 1;
else                                valueStatusResult = 0;

```

```

if(valueStatusResult==3){
    digitalWrite(pinRed,LOW); digitalWrite(pinYellow,LOW);
    digitalWrite(pinGreen,LOW);
    if(fanStatus==2)digitalWrite(pinFan,LOW);
    for(int i=0;i<1;i++){
        digitalWrite(pinBlue,HIGH);
        digitalWrite(pinBuzzer,LOW);
        delay(500);
        digitalWrite(pinBlue,LOW);
        digitalWrite(pinBuzzer,HIGH);
        delay(500);
    }
}
else if(valueStatusResult==2){
    digitalWrite(pinRed,HIGH); digitalWrite(pinYellow,LOW);
    digitalWrite(pinGreen,LOW); digitalWrite(pinBlue,LOW);
    if(buzzStatus==0)buzzStatus = 1;
    if(fanStatus==2)digitalWrite(pinFan,LOW);
}
else if(valueStatusResult==1){
    digitalWrite(pinRed,LOW); digitalWrite(pinYellow,HIGH);
    digitalWrite(pinGreen,LOW); digitalWrite(pinBlue,LOW);

```



```

if(buzzStatus==0)buzzStatus = 1;
if(fanStatus==2)digitalWrite(pinFan,LOW);
}
else{
digitalWrite(pinRed,LOW); digitalWrite(pinYellow,LOW);
digitalWrite(pinGreen,HIGH); digitalWrite(pinBlue,LOW);
buzzStatus = 0;
if(fanStatus==2)digitalWrite(pinFan,HIGH);
}
if (valueStatusResult==3)statusResult="ERROR";
else if (valueStatusResult==2)statusResult="DANGER";
else if (valueStatusResult==1)statusResult="WARNING";
else statusResult="NORMAL";

if(fanStatus==0)digitalWrite(pinFan,HIGH);
if(fanStatus==1)digitalWrite(pinFan,LOW);
if(buzzStatus==1){
buzzStatus=2;
digitalWrite(pinBuzzer,LOW);
lastTime=millis();
}
if(buzzStatus==2&&(millis()-lastTime>=2000)){
digitalWrite(pinBuzzer,HIGH);
buzzStatus=3;
}
if(buzzStatus==0){
digitalWrite(pinBuzzer,HIGH);

```



```
}  
}
```

```
void handleNewMessages(int numNewMessages){
```

```
  Serial.print("NEW Messages ");
```

```
  Serial.println(numNewMessages);
```

```
  for (int i = 0; i < numNewMessages; i++)
```

```
  {
```

```
    String chat_id = bot.messages[i].chat_id;
```

```
    String text = bot.messages[i].text;
```

```
    String from_name = bot.messages[i].from_name;
```

```
    if (from_name == "")
```

```
      from_name = "Guest";
```

```
    if (text == "/KIPASON")
```

```
    {
```

```
      digitalWrite(pinFan,LOW);
```

```
      fanStatus = 1;
```

```
      bot.sendMessage(chat_id, "KIPAS is ON", "");
```

```
    }
```

```
    if (text == "/KIPASOFF")
```

```
    {
```

```
      fanStatus = 0;
```

```
      digitalWrite(pinFan,HIGH);
```



```

    bot.sendMessage(chat_id, "KIPAS is OFF", "");
}
if (text == "/KIPASAUTO")
{
    fanStatus = 2;
    bot.sendMessage(chat_id, "KIPAS is AUTO", "");
}

if (text == "/STATUS")
{
    String statusFan;
    if (fanStatus==0)statusFan="KIPAS\t: MANUAL OFF";
    else if (fanStatus==1)statusFan="KIPAS\t: MANUAL ON";
    else
        statusFan="KIPAS\t: AUTOMATIC";
    String pesan = "STATUS ALAT :\n\n";
    pesan+= "SUHU \t: " + String(valueTemp1) + " Celcius\n";
    pesan+= "Humidity\t: " + String(valueHumi1) + " %\n";
    pesan+= "Status\t: " + String(statusResult) + "\n";
    pesan+= statusFan+"\n";
    bot.sendMessage(chat_id, pesan, "");
}

if (text == "/START")
{
    String welcome = "PENGATUR SUHU RUANGAN, " + from_name + ".\n\n";
    welcome += "CHAT ID : "+String(chat_id)+"\n";
    welcome += "/KIPASON : UNTUK MENYALAKAN KIPAS\n";
}

```



```

welcome += "/KIPASOFF : UNTUK MEMATIKAN KIPAS\n";
welcome += "/KIPASAUTO : UNTUK MEMATIKAN KIPAS\n";
welcome += "/STATUS : MENDAPATKAN STATUS ALAT\n";
bot.sendMessage(chat_id, welcome, "Markdown");
}

bot.sendMessage(chat_id, "PESAN : \"\" + text + "\" TELAH DITERIMA", "");
}
}
void setupFuzzy(){
//----- INISIALISASI INPUT DAN OUTPUT FUZZY SERTA REGISTER RULE-----
-----[
FuzzyInput *temperature1 = new FuzzyInput(1);

temperature1->addFuzzySet(tempNormal1);
temperature1->addFuzzySet(tempHangat1);
temperature1->addFuzzySet(tempPanas1);
fuzzy->addFuzzyInput(temperature1);

// FuzzyInput
FuzzyInput *temperature2 = new FuzzyInput(2);

temperature2->addFuzzySet(humiRendah1);
temperature2->addFuzzySet(humiNormal1);
temperature2->addFuzzySet(humiTinggi1);
fuzzy->addFuzzyInput(temperature2);

// FuzzyOutput

```



```
FuzzyOutput *speedFan = new FuzzyOutput(1);
```

```
speedFan->addFuzzySet(normal);
```

```
speedFan->addFuzzySet(warning);
```

```
speedFan->addFuzzySet(danger);
```

```
fuzzy->addFuzzyOutput(speedFan);
```

```
FuzzyRuleAntecedent *tempNormal1humiRendah1 = new FuzzyRuleAntecedent();
```

```
tempNormal1humiRendah1->joinWithAND(tempNormal1, humiRendah1);
```

```
FuzzyRuleAntecedent *tempNormal1humiNormal1 = new FuzzyRuleAntecedent();
```

```
tempNormal1humiNormal1->joinWithAND(tempNormal1, humiNormal1);
```

```
FuzzyRuleAntecedent *tempNormal1humiTinggi1 = new FuzzyRuleAntecedent();
```

```
tempNormal1humiTinggi1->joinWithAND(tempNormal1, humiTinggi1);
```

```
FuzzyRuleAntecedent *tempHangat1humiRendah1 = new FuzzyRuleAntecedent();
```

```
tempHangat1humiRendah1->joinWithAND(tempHangat1, humiRendah1);
```

```
FuzzyRuleAntecedent *tempHangat1humiNormal1 = new FuzzyRuleAntecedent();
```

```
tempHangat1humiNormal1->joinWithAND(tempHangat1, humiNormal1);
```

```
FuzzyRuleAntecedent *tempHangat1humiTinggi1 = new FuzzyRuleAntecedent();
```

```
tempHangat1humiTinggi1->joinWithAND(tempHangat1, humiTinggi1);
```

```
FuzzyRuleAntecedent *tempPanas1humiRendah1 = new FuzzyRuleAntecedent();
```

```
tempPanas1humiRendah1->joinWithAND(tempPanas1, humiRendah1);
```

```
FuzzyRuleAntecedent *tempPanas1humiNormal1 = new FuzzyRuleAntecedent();
```

```
tempPanas1humiNormal1->joinWithAND(tempPanas1, humiNormal1);
```

```
FuzzyRuleAntecedent *tempPanas1humiTinggi1 = new FuzzyRuleAntecedent();
```

```
tempPanas1humiTinggi1->joinWithAND(tempPanas1, humiTinggi1);
```

```
FuzzyRuleConsequent *thentempNormal1humiRendah1 = new FuzzyRuleConsequent();  
FuzzyRuleConsequent *thentempNormal1humiNormal1 = new FuzzyRuleConsequent();  
FuzzyRuleConsequent *thentempNormal1humiTinggi1 = new FuzzyRuleConsequent();
```

```
FuzzyRuleConsequent *thentempHangat1humiRendah1 = new FuzzyRuleConsequent();  
FuzzyRuleConsequent *thentempHangat1humiNormal1 = new FuzzyRuleConsequent();  
FuzzyRuleConsequent *thentempHangat1humiTinggi1 = new FuzzyRuleConsequent();
```

```
FuzzyRuleConsequent *thentempPanas1humiRendah1 = new FuzzyRuleConsequent();  
FuzzyRuleConsequent *thentempPanas1humiNormal1 = new FuzzyRuleConsequent();  
FuzzyRuleConsequent *thentempPanas1humiTinggi1 = new FuzzyRuleConsequent();
```

```
thentempNormal1humiRendah1->addOutput(normal);  
thentempNormal1humiNormal1->addOutput(warning);  
thentempNormal1humiTinggi1->addOutput(danger);
```

```
thentempHangat1humiRendah1->addOutput(warning);  
thentempHangat1humiNormal1->addOutput(warning);  
thentempHangat1humiTinggi1->addOutput(danger);
```

```
thentempPanas1humiRendah1->addOutput(danger);  
thentempPanas1humiNormal1->addOutput(danger);  
thentempPanas1humiTinggi1->addOutput(danger);
```

```
FuzzyRule *fuzzyRule1 = new FuzzyRule(1, tempNormal1humiRendah1,  
thentempNormal1humiRendah1);
```

```
FuzzyRule *fuzzyRule2 = new FuzzyRule(2, tempNormal1humiNormal1,  
thentempNormal1humiNormal1);
```

```
FuzzyRule *fuzzyRule3 = new FuzzyRule(3, tempNormal1humiTinggi1,  
thentempNormal1humiTinggi1);
```

```
FuzzyRule *fuzzyRule4 = new FuzzyRule(4, tempHangat1humiRendah1,  
thentempHangat1humiRendah1);
```

```
FuzzyRule *fuzzyRule5 = new FuzzyRule(5, tempHangat1humiNormal1,  
thentempHangat1humiNormal1);
```

```
FuzzyRule *fuzzyRule6 = new FuzzyRule(6, tempHangat1humiTinggi1,  
thentempHangat1humiTinggi1);
```

```
FuzzyRule *fuzzyRule7 = new FuzzyRule(7, tempPanas1humiRendah1,  
thentempPanas1humiRendah1);
```

```
FuzzyRule *fuzzyRule8 = new FuzzyRule(8, tempPanas1humiNormal1,  
thentempPanas1humiNormal1);
```

```
FuzzyRule *fuzzyRule9 = new FuzzyRule(9, tempPanas1humiTinggi1,  
thentempPanas1humiTinggi1);
```

```
fuzzy->addFuzzyRule(fuzzyRule1);
```

```
fuzzy->addFuzzyRule(fuzzyRule2);
```

```
fuzzy->addFuzzyRule(fuzzyRule3);
```

```
fuzzy->addFuzzyRule(fuzzyRule4);
```

```
fuzzy->addFuzzyRule(fuzzyRule5);
```

```
fuzzy->addFuzzyRule(fuzzyRule6);
```

```
fuzzy->addFuzzyRule(fuzzyRule7);
```

```
fuzzy->addFuzzyRule(fuzzyRule8);
```

```
fuzzy->addFuzzyRule(fuzzyRule9);
```

```
}
```

