LAMPIRAN - LAMPIRAN

1. Codingan Alat

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <Firebase_ESP_Client.h>
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>  // library lcd 16x2
LiquidCrystal_I2C lcd(0x27, 16, 2);

const int sensor = A0;      //inisialisasi pin sensor
int sensorValue = 0;        //variabel menyimpan Nilai Sensor
const int relay = D4;       //inisialisasi pin relay
const int PinBuzzer = D5;   //inisialisasi pin Buzzer
const int PushButton = D0;  //inisialisasi pin PushButton
int sensorValueDry = 950;   // Nilai Sensor Kondisi Kering
int sensorValueWet = 700;   // Nilai Sensor Kondisi Basah
unsigned int localPort = 2390;
bool onPenjadwalan = false;

#define WIFI_SSID "WRA"
#define WIFI_PASSWORD "123456789"
#define API_KEY "AIzaSyD_p18tZz4Y-HW5RGpA4U0BWi6Cv1suqSk"
#define DATABASE_URL "https://tougeku-default-rtdb.asia-southeast1.firebasedatabase.app/"
#define USER_EMAIL "wayanriko57@gmail.com"
#define USER_PASSWORD "123456"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// Define NTP Client to get time (waktu)
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
// Variable to save current epoch time
unsigned long epochTime;
String jadwal[4];
String lastTime;
float moisturePercentage;

void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  timeClient.begin();
  timeClient.setTimeOffset(25200);  // 7*3600 => GMT +7

  config.api_key = API_KEY;
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  config.database_url = DATABASE_URL;
  config.token_status_callback = tokenStatusCallback;  // see addons/TokenHelper.h
  Firebase.reconnectNetwork(true);
  fbdo.setBSSLBufferSize(4096 /* Rx buffer size in bytes from 512 - 16384 */, 1024 /* Tx buffer size in bytes from 512 - 16384 */);
  Firebase.begin(&config, &auth);

  lcd.init();
  lcd.backlight();
  pinMode(sensor, INPUT);             //menetapkan pin sebagai input
  pinMode(relay, OUTPUT);             //menetapkan pin sebagai output
  pinMode(PinBuzzer, OUTPUT);         //menetapkan pin sebagai output
  pinMode(PushButton, INPUT_PULLUP);  //menetapkan pin sebagai output
}

void loop() {
  timeClient.update();
  epochTime = timeClient.getEpochTime();
  Serial.print("Timestamp : ");
  Serial.println(epochTime);
  sensorValue = analogRead(sensor);  //baca nilai sensor
  // int dataAnalog = analogRead(sensor); membaca nilai dari pin sensor
  moisturePercentage = map(sensorValue, sensorValueWet, sensorValueDry, 100, 0);
  moisturePercentage = constrain(moisturePercentage, 0, 100);
  int button_state = digitalRead(PushButton);  //membaca nilai dari pin sensor
  Serial.print("Kelembaban : ");              //inisialisasi lcd
  Serial.print(sensorValue);
  lcd.setCursor(0, 0);
  lcd.print("Kelembaban : ");  //inisialisasi lcd
  lcd.println(moisturePercentage);
  lcd.setCursor(0, 1);
  lcd.println("Status:");  //inisialisasi lcd
  Serial.print("Button : ");
  Serial.println(button_state);

  if (Firebase.ready()) {
    String clockFormat = (String)timeClient.getFormattedTime();
    String statKelembaban = statusKelembaban(moisturePercentage);
    Serial.printf("Update Kelembaban %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/hum"), moisturePercentage) ? "success" : fbdo.errorReason().c_str());
    Serial.printf("Update Status %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/status"), statKelembaban) ? "success" : fbdo.errorReason().c_str());
  }
  cekJadwalPenyiraman();
  delay(1000);
}
```

```
void cekJadwalPenyiraman() {
  // Cek button apakah dikendalikan oleh aplikasi
  String chkBtn = "", kondisiPompa = "";
  Serial.printf("Cek button %s\n", Firebase.RTDB.getString(&fbdo, F("/sensor/button")) ?
chkBtn = fbdo.stringData() : fbdo.errorReason().c_str());
  if (chkBtn == "Pompa diatur melalui aplikasi.") {
    Serial.printf("Set pompa %s\n", Firebase.RTDB.getString(&fbdo, F("/sensor/pump")) ?
kondisiPompa = fbdo.stringData() : fbdo.errorReason().c_str());
    digitalWrite(PinBuzzer, (kondisiPompa == "ON") ? HIGH : LOW);
    digitalWrite(relay, (kondisiPompa == "ON") ? HIGH : LOW);
  }

  // Ambil jadwal di database firebase
  for (int i = 0; i < 4; i++) {
    bool bVal;
    Serial.printf("Cek jadwal yang aktif => %s\n", Firebase.RTDB.getBool(&fbdo, "/jadwal/id" +
String(i + 1) + "/active", &bVal) ? bVal ? "true" : "false" : fbdo.errorReason().c_str());
    if (bVal) {
      Serial.printf("Jadwal aktif jam => %s\n", Firebase.RTDB.getString(&fbdo, "/jadwal/id" +
String(i + 1) + "/time") ? jadwal[i] = fbdo.stringData() : fbdo.errorReason().c_str());
    } else {
      jadwal[i] = "";
    }
  }

  // Cek jadwal berdasarkan waktu, apakah sudah waktunya menyiram
  String currentTime = String(timeClient.getHours()) + ":" + String(timeClient.getMinutes());
  Serial.println("Jam saat ini : " + currentTime);
  for (int i = 0; i < 4; i++) {
    if (jadwal[i].compareTo(currentTime) == 0 && onPenjadwalan == false) {
      digitalWrite(relay, HIGH);  // Menghidupkan pompa
      digitalWrite(PinBuzzer, HIGH);
      Serial.printf("Set Button %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/button"),
"Penyiraman otomatis menggunakan jadwal pada jam " + jadwal[i]) ? "success" :
fbdo.errorReason().c_str());
      Serial.printf("Set Pump %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/pump"), "ON")
? "success" : fbdo.errorReason().c_str());
      lastTime = jadwal[i];
      delay(60000);  // Lama penyiraman. Bebas di atur berapa aja
      digitalWrite(relay, LOW);
      digitalWrite(PinBuzzer, LOW);
      Serial.printf("Set Pump %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/pump"), "OFF")
? "success" : fbdo.errorReason().c_str());
      onPenjadwalan = true;
    }
  }

  Serial.print("Status onPenjadwalan => ");
  Serial.println(onPenjadwalan);

  if (lastTime.compareTo(currentTime) != 0 && onPenjadwalan == true) {
    Serial.print("Ubah onPenjadwalan => ");
    onPenjadwalan = false;
    Serial.println(onPenjadwalan);
  }
  // // Cek penyiraman menggunakan sensor
  if (statusKelembaban(moisturePercentage) == "Media Kering") {
    digitalWrite(relay, HIGH);  // Menghidupkan pompa
    digitalWrite(PinBuzzer, HIGH);
    Serial.printf("Set Button %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/button"),
"Penyiraman otomatis menggunakan sensor") ? "success" : fbdo.errorReason().c_str());
    Serial.printf("Set Pump %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/pump"), "ON") ?
"success" : fbdo.errorReason().c_str());
    delay(10000);  // Lama penyiraman menggunakan sensor
    digitalWrite(relay, LOW);
    digitalWrite(PinBuzzer, LOW);
    Serial.printf("Set Pump %s\n", Firebase.RTDB.setString(&fbdo, F("/sensor/pump"), "OFF") ?
"success" : fbdo.errorReason().c_str());
  }
}

String statusKelembaban(float persen) {
  if (persen > 90) {
    return "Media Sangat Basah";
  } else if (persen <= 90 && persen >= 80) {
    return "Media Masih Basah";
  }
  return "Media Kering";
}
```

2. Datasheet
Esp 8266

Date	Version	Release Notes
2015.12	V4.6	Updated Chapter 3.
2016.02	V4.7	Updated Section 3.6 and Section 4.1.
2016.04	V4.8	Updated Chapter 1.
2016.08	V4.9	Updated Chapter 1.
2016.11	V5.0	Added Appendix Ⅱ “Learning Resources”.
2016.11	V5.1	Changed the power consumption during Deep-sleep from 10 μA to 20 μA in Table 5-2.
2016.11	V5.2	Changed the crystal frequency range from “26 MHz to 52 MHz” to “24 MHz to 52 MHz” in Section 3.3.
2016.12	V5.3	Changed the minimum working voltage from 3.0 V to 2.5 V.
2017.04	V5.4	Changed chip input and output impedance from 50 Ω to 39 + j6 Ω.
2017.10	V5.5	Updated Chapter 3 regarding the range of clock amplitude to 0.8 V ~ 1.5 V.
2017.11	V5.6	Updated VDDPST from 1.8 V ~ 3.3 V to 1.8 V ~ 3.6 V.
2017.11	V5.7	• Corrected a typo in the description of SDIO_DATA_0 in Table 2-1; • Added the testing conditions for the data in Table 5-2.
2018.02	V5.8	• Updated Wi-Fi protocols in Section 1.1; • Updated description of the integrated Tensilica processor in 3.1.
Date	Version	Release Notes

2018.09	V5.9	• Update document cover; • Added a note for Table 1-1; • Updated Wi-Fi key features in Section 1.1; • Updated description of the Wi-Fi function in 3.5; • Updated pin layout diagram; • Fixed a typo in Table 2-1; • Removed Section AHB and AHB module; • Restructured Section Power Management; • Fixed a typo in Section UART; • Removed description of transmission angle in Section IR Remote Control; • Other optimization (wording).
2018.11	V6.0	• Added an SPI pin in Table 4-2; • Updated the diagram of packing information.
2019.08	V6.1	Removed description of the GPIO function in Section 4.1.
2019.08	V6.2	Updated notes on CHIP_EN in Section 5.1
2019.12	V6.3	Add feedback links.
2020.04	V6.4	• Removed the description of “Antenna diversity”; • Updated the feedback links.
2020.07	V6.5	• Updated the description of HSPI in Section 4.3; • Updated links in Appendix.
2020.10	V6.6	• Fixed a typo in Figure 2-1; • Updated the link of *ESP8266 Pin List.*
2022.07	v6.7	• Updated Figure 2-1; • Updated the link of *ESP8266 Hardware Resources.*
2022.10	v6.8	Updated typos in Chapter 6.
2023.02	v6.9	Added link to *Xtensa® Instruction Set Architecture (ISA) Summary* in Section 3.1.1.
2023.06	v7.0	• Added a note on the cover page; • Updated two documents in Appendix.

Soil Moisture

Specification

- Power supply: 3.3v or 5v
- Output voltage signal: 0~4.2v
- Current: 35mA □ Pin definition:

Analog output(Blue wire)
GND(Black wire)
Power(Red wire)
□ Size:
60x20x5mm □
Value
range:
0 ~300 : dry soil
300~700 : humid soil
700~950 : in water

Specification

- Power supply: 3.3v or 5v
- Output voltage signal: 0~4.2v
- Current: 35mA □ Pin definition:

Analog output(Blue wire)
GND(Black wire)

Power(
Red
wire) □

Size:
60x20x5
mm □

Value
range:
0 ~300 : dry soil
300~700 : humid soil
70050 : in water