LAMPIRAN

Lampiran 1 Source Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ESP8266HTTPClient.h> // Library untuk HTTP request

// Ganti dengan kredensial WiFi dan token bot Anda
const char* ssid = "Oioi"; // Nama WiFi
const char* password = "00000001"; // Password WiFi
const char* botToken = "7379595880:AAEcQvnXyZzbevsMvtjam-
uNWFrJ7ZCu8tM"; // Token Telegram Bot
const char* chatID = "5060163371"; // ID chat untuk pengiriman
pesan
const char* serverName =
"http://192.168.137.237/cuacaDB/post.php"; // Alamat server
PHP

LiquidCrystal_I2C lcd(0x27, 16, 2);
WiFiClientSecure client;
WiFiClient httpClient; // Deklarasi WiFiClient untuk HTTP
request
UniversalTelegramBot bot(botToken, client);

const int raindropPin = 2;
const int LDRPin = A0;
const int buttonPin = D7;
const int buzzerPin = D3;
#define ENA D8
#define IN1 14
#define IN2 12
#define lamp D0

bool motorRunning = false;
volatile bool buttonPressed = false;
bool buzzerOn = false;
bool manualLampOff = false;
bool manualControl = false;
int lastRaindropValue = 1;
int lastLDRValue = 0;

void ICACHE_RAM_ATTR handleButtonPress() {
```

```
  buttonPressed = true;
}

void sendToServer(const String &status) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;

    // Encode status untuk memastikan tidak ada karakter yang
tidak valid dalam URL
    String encodedStatus = urlencode(status);
    String url = String(serverName) + "?status=" +
encodedStatus;

    Serial.print("Connecting to URL: ");
    Serial.println(url); // Debugging URL

    http.begin(httpClient, url); // Menggunakan WiFiClient
dengan URL

    int httpCode = http.GET();

    if (httpCode > 0) {
      String payload = http.getString();
      Serial.println("HTTP Response code: " +
String(httpCode));
      Serial.println(payload);
    } else {
      Serial.println("Error on HTTP request");
    }
    http.end();
  } else {
    Serial.println("WiFi not connected");
  }
}

String urlencode(String str) {
  String encodedString = "";
  char c;
  char code0;
  char code1;
  char code2;
  for (int i = 0; i < str.length(); i++) {
    c = str.charAt(i);
    if (c == ' ') {
      encodedString += '+';
```

```
    } else if (isalnum(c)) {
      encodedString += c;
    } else {
      code1 = (c & 0xf) + '0';
      if ((c & 0xf) > 9) {
        code1 = (c & 0xf) - 10 + 'A';
      }
      c = (c >> 4) & 0xf;
      code0 = c + '0';
      if (c > 9) {
        code0 = c - 10 + 'A';
      }
      code2 = '\0';
      encodedString += '%';
      encodedString += code0;
      encodedString += code1;
    }
    yield();
  }
  return encodedString;
}

void setup() {
  pinMode(LDRPin, INPUT);
  pinMode(raindropPin, INPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(lamp, OUTPUT);
  digitalWrite(lamp, HIGH);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Alat Pendeteksi");
  lcd.setCursor(0, 1);
  lcd.print("Hujan");
  delay(1000);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Jemuran");
  lcd.setCursor(0, 1);
```

```
  lcd.print("Otomatis");
  delay(1000);

  Serial.begin(115200);
  Serial.println("alat pendeteksi hujan dan jemuran
otomatis");
  delay(1000);

  attachInterrupt(digitalPinToInterrupt(buttonPin),
handleButtonPress, FALLING);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Menghubungkan ke WiFi...");
  }
  Serial.println("Terhubung ke WiFi");

  client.setInsecure();
}

void loop() {
  int LDRValue = analogRead(LDRPin);
  int raindropValue = digitalRead(raindropPin);

  Serial.print("Raindrop Sensor Value: ");
  Serial.println(raindropValue);
  Serial.print("LDR Sensor Value: ");
  Serial.println(LDRValue);

  if (buttonPressed) {
    buttonPressed = false;
    digitalWrite(buzzerPin, LOW);
    buzzerOn = false;
    Serial.println("Button pressed! Buzzer OFF.");
    delay(200);
  }

  if (!manualControl) {
    // Menarik jemuran ke dalam jika terdeteksi hujan
    if (raindropValue == 0 && lastRaindropValue == 1) {
      digitalWrite(buzzerPin, HIGH);
      buzzerOn = true;
      digitalWrite(lamp, LOW);
      manualLampOff = false;
```

```
      Serial.println("Mengirim notifikasi ke Telegram:
Terdeteksi hujan!");
      if (bot.sendMessage(chatID, "Terdeteksi hujan!", "")) {
        Serial.println("Notifikasi terkirim.");
      } else {
        Serial.println("Gagal mengirim notifikasi.");
      }

      if (!motorRunning) {
        motorRunning = true;
        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);
        analogWrite(ENA, 255);
        delay(900);
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        analogWrite(ENA, 0);

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Menarik ke dalam");
        lcd.setCursor(0, 1);
        lcd.print("Hujan");
        sendToServer("Menarik ke dalam (Hujan)"); // Kirim
data ke server
      }
    }
    // Menarik jemuran ke dalam jika terdeteksi gelap
    else if (LDRValue > 800 && raindropValue == 1) {
      if (!manualLampOff) {
        digitalWrite(lamp, LOW);
      }
      if (!motorRunning) {
        motorRunning = true;
        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);
        analogWrite(ENA, 255);
        delay(900);
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        analogWrite(ENA, 0);

        lcd.clear();
        lcd.setCursor(0, 0);
```

```
        lcd.print("Menarik ke dalam");
        lcd.setCursor(0, 1);
        lcd.print("Gelap");

        Serial.println("Mengirim notifikasi ke Telegram:
Terdeteksi Gelap.");
        if (bot.sendMessage(chatID, "Terdeteksi Gelap.", ""))
{
          Serial.println("Notifikasi terkirim.");
        } else {
          Serial.println("Gagal mengirim notifikasi.");
        }
        sendToServer("Menarik ke dalam (Hujan)"); // Kirim
data ke server
      }
    }
    // Menarik jemuran keluar jika cerah dan tidak hujan
    else if (raindropValue == 1 && LDRValue <= 800 &&
motorRunning) {
      if (!manualLampOff) {
        digitalWrite(lamp, HIGH);
      }
      motorRunning = false;
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      analogWrite(ENA, 255);
      delay(900);
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
      analogWrite(ENA, 0);

      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Menarik keluar");
      lcd.setCursor(0, 1);
      lcd.print("Cerah");

      Serial.println("Mengirim notifikasi ke Telegram: Jemuran
ditarik keluar.");
      if (bot.sendMessage(chatID, "Jemuran ditarik keluar
karena cerah.", "")) {
        Serial.println("Notifikasi terkirim.");
      } else {
        Serial.println("Gagal mengirim notifikasi.");
      }
```

```
      sendToServer("Menarik keluar (Cerah)"); // Kirim data ke
server
    }
  }

  if (LDRValue <= 800 && lastLDRValue > 800) {
    manualLampOff = false;
  }

  lastRaindropValue = raindropValue;
  lastLDRValue = LDRValue;

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

  while (numNewMessages) {
    Serial.println("Pesan baru diterima");

    for (int i = 0; i < numNewMessages; i++) {
      String chat_id = String(bot.messages[i].chat_id);
      String text = bot.messages[i].text;

      Serial.println("Pesan: " + text);

      if (text == "/buzzer_off") {
        digitalWrite(buzzerPin, LOW);
        buzzerOn = false;
        bot.sendMessage(chat_id, "Buzzer dimatikan.", "");
        Serial.println("Buzzer dimatikan melalui Telegram.");
      }

      if (text == "/lamp_off") {
        digitalWrite(lamp, HIGH);
        manualLampOff = true;
        bot.sendMessage(chat_id, "Lampu dimatikan.", "");
        Serial.println("Lampu dimatikan melalui Telegram.");
      }

      if (text == "/lamp_on") {
        digitalWrite(lamp, LOW);
        manualLampOff = false;
        bot.sendMessage(chat_id, "Lampu dinyalakan.", "");
        Serial.println("Lampu dinyalakan melalui Telegram.");
      }
```

```
      if (text == "/pull_in") {
        manualControl = true;
        if (!motorRunning) {
          motorRunning = true;
          digitalWrite(IN1, HIGH);
          digitalWrite(IN2, LOW);
          analogWrite(ENA, 255);
          delay(900);
          digitalWrite(IN1, LOW);
          digitalWrite(IN2, LOW);
          analogWrite(ENA, 0);

          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("Menarik ke dalam");
          lcd.setCursor(0, 1);
          lcd.print("Manual");
        }
        bot.sendMessage(chat_id, "Jemuran ditarik ke dalam.",
"");
        Serial.println("Jemuran ditarik ke dalam melalui
Telegram.");
      }

      if (text == "/push_out") {
        manualControl = true;
        if (motorRunning) {
          motorRunning = false;
          digitalWrite(IN1, LOW);
          digitalWrite(IN2, HIGH);
          analogWrite(ENA, 255);
          delay(900);
          digitalWrite(IN1, LOW);
          digitalWrite(IN2, LOW);
          analogWrite(ENA, 0);

          lcd.clear();
          lcd.setCursor(0, 0);
          lcd.print("Menarik keluar");
          lcd.setCursor(0, 1);
          lcd.print("Manual");
        }
        bot.sendMessage(chat_id, "Jemuran ditarik keluar.",
"");
```

```
        Serial.println("Jemuran ditarik keluar melalui
Telegram.");
      }
      // Tambahkan perintah baru untuk mengaktifkan kembali
mode otomatis
      if (text == "/auto_mode") {
        manualControl = false; // Menonaktifkan kontrol manual
        bot.sendMessage(chat_id, "Mode otomatis diaktifkan
kembali.", "");
        Serial.println("Mode otomatis diaktifkan kembali
melalui Telegram.");
      }
      if ( text = "/options"){
        String keyboardJson = "[[\"/buzzer_off\",
\"/lamp_off\"\"/lamp_on\"],[\"/pull_in\",\"/push_out\"],[\"/au
to_mode\"]]";
        bot.sendMessageWithReplyKeyboard(chat_id, "Choose from
one of the following Start", "", keyboardJson, true);
      }
      if (text == "/start")
    {
      String welcome = "Selamat Datang\n";
      welcome += "Silahkan Pilih dibawah.\n\n";
      welcome += "/buzzer_off : Matikan Buzzer\n";
      welcome += "/lamp_off : Matikan Lampu\n";
      welcome += "/lamp_on : Hidupkan Lampu\n";
      welcome += "/pull_in : Tarik Masuk\n";
      welcome += "/push_out : Tarik Keluar\n";
      welcome += "/auto_mode : Mode otomatis\n";
      welcome += "/options : returns the reply keyboard\n";
      bot.sendMessage(chat_id, welcome, "Markdown");
    }
    }

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

  delay(100);
}
```