

Lampiran 3 ESP8266 ke PHP

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <DHT.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <UniversalTelegramBot.h>

Servo myServo;
Servo myServo1;

#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPL6ld0TaGAr"
#define BLYNK_TEMPLATE_NAME "smart garden"
char auth[] = "0EAY1IRWZcONnp3Ge6oXa4vb6e70rY3i";

const char* ssid = "Galaxy A05s 8353";
const char* password = "dika1234";

String BOTtoken =
"7403737105:AAFCy2WWZS370LIQhcakugWfJ6114qGnr40";
String idBot = "HankamGardenia_bot";

String message, message_pompa, message_temperature;

#define DHTTYPE DHT11
#define dht_pin D5
DHT dht(dht_pin, DHTTYPE);

const int moisturePin = A0;

const int ldrPin = D3;

bool Relay = 0;

#define relayPin D0
BlynkTimer timer;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 0, 60000);
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

void checkWiFiConnection() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.println("Connecting to Wi-Fi...");
    delay(1000);
  }
}
```

```

Serial.println("Connected to Wi-Fi");
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());
}

void sendSensorDataToServer() {
    WiFiClient client;
    HTTPClient http;

    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();
    int soilMoisture = analogRead(moisturePin);
    float moisture = (soilMoisture / 1023.0) * 100.0;
    bool LDRvalue = digitalRead(ldrPin);

    if (LDRvalue) {
        myServo.write(15);
        message = "Cahaya Gelap";
    } else {
        myServo.write(80);
        message = "Cahaya Terang";
    }

    if (moisture >= 50) {
        digitalWrite(relayPin, LOW);
        message_pompa = "Pompa air hidup";
    } else {
        digitalWrite(relayPin, HIGH);
        message_pompa = "Pompa air mati";
    }

    if (temperature > 28) {
        message_temperature = String(temperature) + "(Di atas
batas normal)";
        bot.sendMessage("1332023904", "Peringatan! Suhu terlalu
tinggi: " + message_temperature, "");
    } else {
        message_temperature = String(temperature);
    } }

Serial.print("Moisture = ");
Serial.print(moisture);
Serial.println("%");
Serial.print("Kelembaban: ");
Serial.print(humidity);
Serial.println("%");
Serial.print("Suhu: ");

```

```

Serial.print(temperature);
Serial.println("°C");
Serial.print(LDRvalue);
Serial.println(message);
Serial.println(message_pompa);

String postData = "temperature=" + message_temperature +
"&humidity=" + String(humidity) + "&moisture=" +
String(moisture) + "&intensitas_cahaya=" + message +
"&message_pompa=" + message_pompa;
Serial.print("Sending data: ");
Serial.println(postData);

http.begin(client, "http://hankamga.tif-
skripsi.my.id/garden/garden.php");
http.addHeader("Content-Type", "application/x-www-form-
urlencoded");

int httpCode = http.POST(postData);

if (httpCode == HTTP_CODE_MOVED_PERMANENTLY || httpCode ==
HTTP_CODE_FOUND) {
    String newLocation = http.header("Location");
    Serial.print("Redirected to: ");
    Serial.println(newLocation);
    http.begin(client, newLocation);
    httpCode = http.POST(postData);
}

if (httpCode == HTTP_CODE_OK) {
    String response = http.getString();
    Serial.println("Server response: " + response);
} else {
    Serial.print("HTTP POST request failed with error code:
");
    Serial.println(httpCode);
}
http.end();
}

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

        if (text == "/check_status") {
            float temperature = dht.readTemperature();
            float humidity = dht.readHumidity();
            String message = "Suhu: " + String(temperature) + "°C\n";

```

```

if (LDRvalue) {
    message = "Cahaya Gelap";
} else {
    message = "Cahaya Terang";
}

Blynk.virtualWrite(V0, moisture);
Blynk.virtualWrite(V1, temperature);
Blynk.virtualWrite(V2, humidity);
Blynk.virtualWrite(V3, message);
}

void checkWateringTime() {
    timeClient.update();
    int currentHour = timeClient.getHours();
    int currentMinute = timeClient.getMinutes();

    // Waktu penyiraman: 06:00, 12:00, 18:00, dan 00:00
    if ((currentHour == 6 || currentHour == 12 || currentHour ==
18 || currentHour == 0) && currentMinute == 0) {
        Serial.println("Scheduled watering...");
        waterPlants();
    }
}

void waterPlants() {
    digitalWrite(relayPin, LOW);
    delay(10000);
    digitalWrite(relayPin, HIGH);
    Serial.println("Watering completed.");
}

    bot.sendMessage(chat_id, message, "");
}
}

// Berguna untuk mengirim sensor ke blynk
void sendSensorDataToBlynk() {
    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();
    int soilMoisture = analogRead(moisturePin);
    float moisture = (soilMoisture / 1023.0) * 100.0;
    bool LDRvalue = digitalRead(ldrPin);
void setup() {
    Serial.begin(9600);
    Serial.print("Mulai");
    checkWiFiConnection();
    client.setInsecure();
}

```

```

bot.updateToken(BOTtoken);

myServo.attach(15, 500, 2400);
pinMode(ldrPin, INPUT);

pinMode(relayPin, OUTPUT);
digitalWrite(relayPin, HIGH);

Blynk.begin(auth, ssid, password);

dht.begin();

timeClient.begin();

timer.setInterval(3000L, sendSensorDataToServer);
timer.setInterval(3000L, sendSensorDataToBlynk);
timer.setInterval(60000L, checkWateringTime);
}

void loop() {

    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);
    }
    Blynk.run();
    timer.run();
}

```