

TUGAS SIB ARKATAMA MULTI SOLUSINDO  
IOT PRAKTIK

---

---

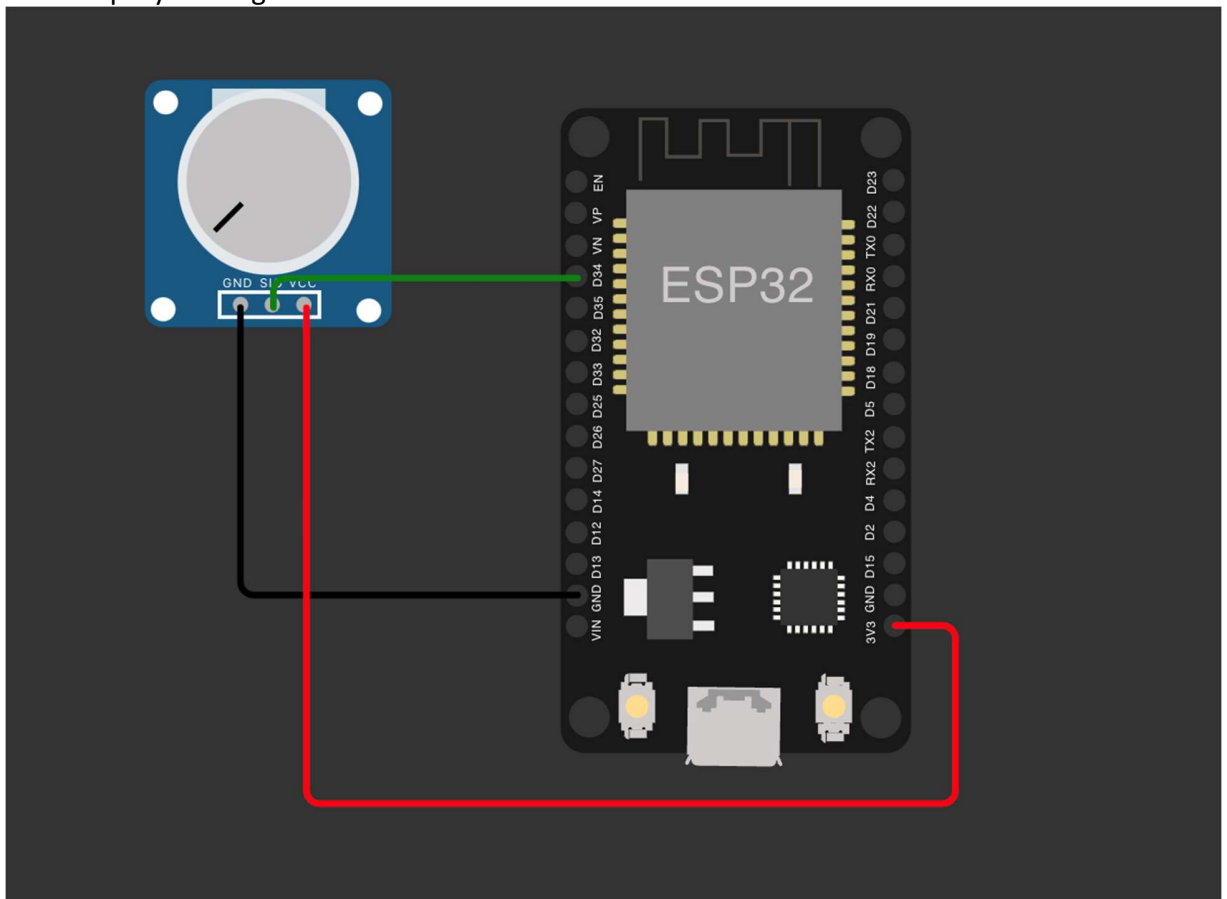
ID Kegiatan : 7582873  
Nama : Evy Nur Imamah  
Kelas : IoT1

**TOOLS!**

1. <https://wokwi.com/>
2. <https://www.postman.com/>
3. <https://laragon.org/>
4. <https://getcomposer.org/>
5. <https://nodejs.org/en>

**Soal!**

1. Buatlah proyek dengan skema berikut :



#### Deskripsi Proyek:

Proyek ini bertujuan untuk mengembangkan sebuah sistem IoT yang mampu mengukur suhu menggunakan sensor potensio. Sensor potensio akan memberikan nilai analog dengan rentang 0 - 4095, yang akan dikonversi menjadi suhu dalam satuan Celcius dengan rentang 0 - 100. Data suhu yang terukur akan dikirimkan ke server IoT yang telah dibuat sebelumnya. Komunikasi antara perangkat ESP32 dan server IoT akan dilakukan menggunakan protokol HTTP dengan memanfaatkan Arduino JSON dan HttpClient.

#### Spesifikasi Proyek:

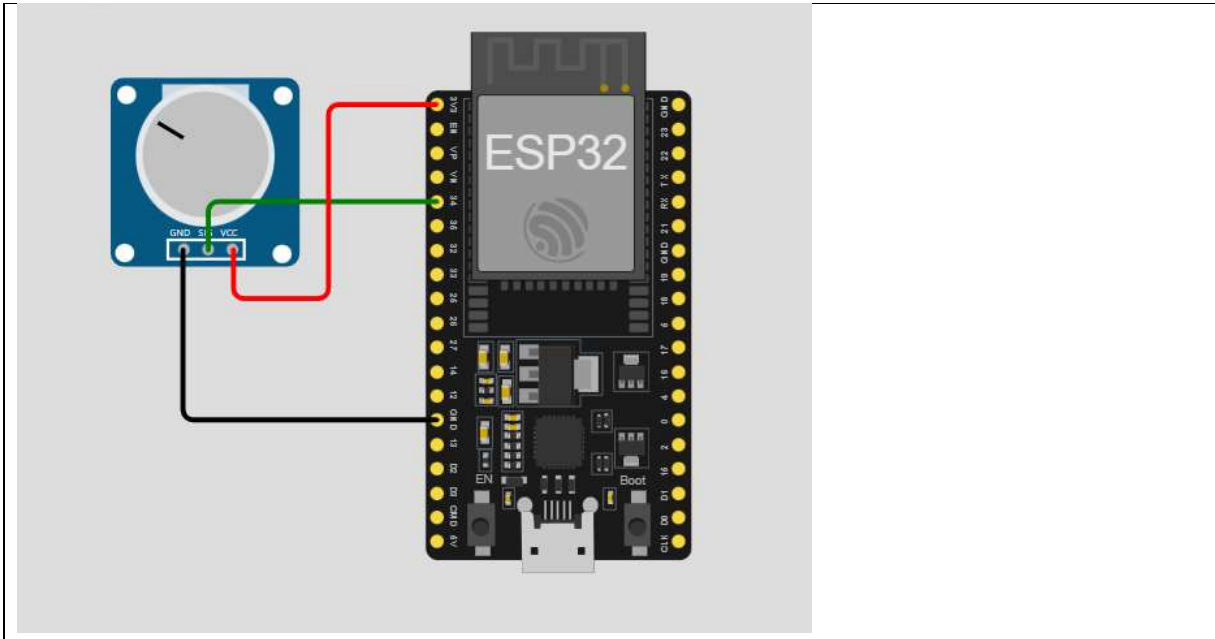
- a. Sensor Potensio untuk Pengukuran Suhu:
  - Menggunakan sensor potensio sebagai sensor suhu. Nilai tegangan yang diberikan oleh sensor potensio akan berubah sesuai dengan suhu lingkungan.
- b. Konversi Tegangan menjadi Suhu Celcius:
  - Menerapkan logika pengonversi untuk mengubah nilai tegangan dari sensor potensio menjadi suhu dalam satuan Celcius.
- c. Komunikasi dengan Server IoT:
  - Menggunakan ESP32 untuk mengirimkan data suhu yang terukur ke server IoT.
  - Menggunakan protokol HTTP untuk mengirimkan data menggunakan HttpClient.
  - Data suhu akan dikirimkan ke endpoint API yang telah dibuat sebelumnya pada server IoT.

#### Langkah-langkah Proyek:

1. Pengaturan Perangkat:
  - Persiapkan perangkat ESP32, sensor potensio, dan pastikan koneksi WiFi tersedia.
2. Pengembangan Kode Program:
  - Tulis kode program untuk ESP32 menggunakan Arduino IDE.
  - Implementasikan logika untuk membaca nilai tegangan dari sensor potensio.
  - Terapkan pengonversi untuk mengubah nilai tegangan menjadi suhu dalam satuan Celcius.
  - Gunakan Arduino JSON untuk mengemas data suhu ke dalam format JSON.
  - Gunakan HttpClient untuk mengirimkan data suhu ke server IoT melalui protokol HTTP.
3. Pengiriman Data Suhu ke Server IoT:
  - Tentukan endpoint API yang akan digunakan untuk menerima data suhu pada server IoT.
  - Pastikan bahwa server IoT dapat menerima data suhu dalam format JSON yang telah ditentukan.
4. Testing:
  - Uji coba kode program pada ESP32 untuk memastikan bahwa sensor potensio dapat membaca suhu dengan benar.
  - Periksa apakah data suhu berhasil dikirimkan ke server IoT dan dapat diakses melalui endpoint API yang telah ditentukan.

**Jawab!**

**1. Rangkaian Skematik (\*gambar/screenshoot)**



**2. Program (.ino,.c)**

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

const char* ssid = "zerone";
const char* password = "1234567890";

String baseUrl = "https://d5d9-2001-448a-5020-a81e-4983-9ebd-9dc0-4d7c.ngrok-free.app";
String temperatureEndpoint = baseUrl + "/temperature";
String potentiometerEndpoint = baseUrl + "/potentiometer";

void setup() {
    Serial.begin(115200);

    Serial.println("Connecting to WiFi...");
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        Serial.println("Connecting...");
        delay(250);
    }
}
```

```

    Serial.println("Connected to WiFi.");
}

void loop() {
    int potValue = analogRead(34); // Read potentiometer value
    int temperature = map(potValue, 0, 4095, 0, 100); // Convert
    potentiometer value to temperature in Celsius
    insertData(temperature, potValue); // Insert temperature and
    potentiometer value
    delay(5000);
    getTemperature(); // Get temperature from server
}

void insertData(int temp, int pot) {
    HTTPClient http;
    // Connect to API
    Serial.println("Connecting to API");
    Serial.println(temperatureEndpoint);
    http.begin(temperatureEndpoint);
    http.addHeader("Content-Type", "application/json");

    // Create JSON object for temperature
    DynamicJsonDocument tempDoc(1024);
    tempDoc["temperature"] = temp;
    String tempBody;
    serializeJson(tempDoc, tempBody);

    // POST request for temperature
    int tempStatusCode = http.POST(tempBody);
    if (tempStatusCode > 0) {
        Serial.println("Temperature sent successfully with status: " +
String(tempStatusCode));
        String tempPayload = http.getString();
        Serial.println(tempPayload);
    } else {
        Serial.println("Error sending temperature: " +
String(tempStatusCode));
    }

    http.end();

    // Connect to API for potentiometer
    Serial.println("Connecting to API for potentiometer");
    Serial.println(potentiometerEndpoint);
}

```

```

http.begin(potentiometerEndpoint);
http.addHeader("Content-Type", "application/json");

// Create JSON object for potentiometer
DynamicJsonDocument potDoc(1024);
potDoc["potentiometer"] = pot;
String potBody;
serializeJson(potDoc, potBody);

// POST request for potentiometer
int potStatusCode = http.POST(potBody);
if (potStatusCode > 0) {
    Serial.println("Potentiometer value sent successfully with status: " +
+ String(potStatusCode));
    String potPayload = http.getString();
    Serial.println(potPayload);
} else {
    Serial.println("Error sending potentiometer value: " +
String(potStatusCode));
}

http.end();
}

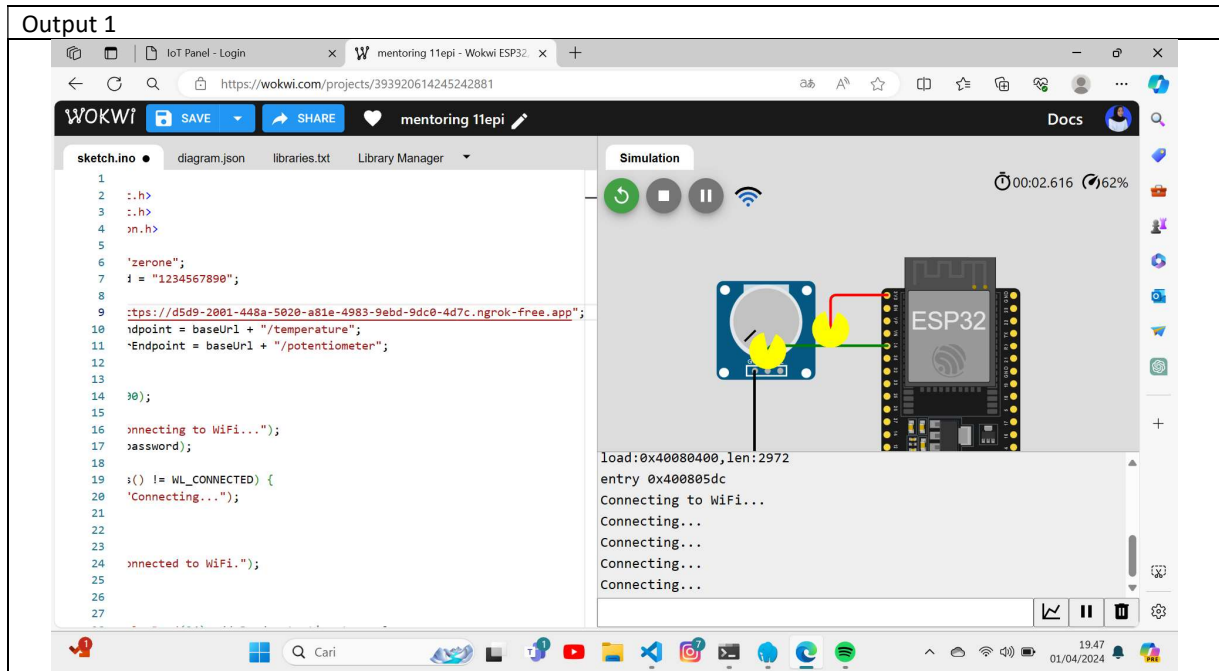
void getTemperature() {
    HTTPClient http;
    // Connect to API
    Serial.println("Connecting to API for temperature");
    Serial.println(temperatureEndpoint);
    http.begin(temperatureEndpoint);
    http.addHeader("Content-Type", "application/json");

    // GET request for temperature
    int statusCode = http.GET();
    if (statusCode > 0) {
        Serial.println("Temperature received successfully with status: " +
String(statusCode));
        String payload = http.getString();
        Serial.println(payload);
    } else {
        Serial.println("Error getting temperature: " + String(statusCode));
    }
    http.end();
}

```

### 3. Hasil Pengerjakan (screenshot dijalankan)

## Output 1



## Output 2

