Tugas 4

Laporan Praktik Akses API Melalui Simulasi WOKWI

Nama : Bintang Putra Nala Saki
Kelas : T4C
NIM 233140700111077

Fakultas Vokasi
Universitas Brawijaya
Email :bintangsakti867@gmail.com

ABSTRAK

Praktik akses API melalui simulasi Wokwi bertujuan untuk memahami bagaimana perangkat IoT dapat berkomunikasi dengan server melalui protokol HTTP. Simulasi ini memungkinkan pengembang untuk menguji konektivitas dan respons API tanpa perangkat keras fisik. Dalam laporan ini, dijelaskan bagaimana menggunakan Wokwi untuk mengakses API, mengirim permintaan HTTP, serta mengelola respons data yang diterima. Hasil dari praktik ini menunjukkan bahwa simulasi Wokwi sangat berguna dalam mengembangkan dan menguji fitur IoT sebelum implementasi nyata dilakukan.

PENDAHULUAN

Dalam pengembangan sistem IoT, akses ke Application Programming Interface (API) menjadi aspek penting untuk memungkinkan komunikasi antara perangkat dan server. API memungkinkan perangkat mengirim dan menerima data secara real-time, yang sangat dibutuhkan dalam berbagai aplikasi IoT seperti smart home, monitoring lingkungan, dan otomasi industri. Untuk menguji akses API, pengembang sering kali menggunakan simulator seperti Wokwi sebelum menerapkannya pada perangkat fisik. Dalam bab ini, akan dibahas praktik akses API menggunakan simulasi Wokwi sebagai alat uji coba dalam pengembangan sistem IoT berbasis cloud.

LATAR BELAKANG

Seiring dengan berkembangnya teknologi IoT, kebutuhan akan komunikasi yang efisien antara perangkat dan server menjadi semakin penting. Salah satu metode yang digunakan adalah dengan memanfaatkan API berbasis HTTP yang memungkinkan pertukaran data antara perangkat dan cloud. Namun, pengujian akses API pada perangkat fisik sering kali menghadapi kendala seperti keterbatasan perangkat, biaya, dan waktu. Oleh karena itu, simulasi perangkat IoT menggunakan Wokwi menjadi solusi efektif untuk mengembangkan, menguji, dan menyempurnakan akses API sebelum implementasi pada perangkat nyata.

TUJUAN

Adapun tujuan dari praktik ini adalah:

1. Memahami konsep dasar akses API dalam konteks IoT.
2. Menggunakan simulasi Wokwi untuk menguji komunikasi antara perangkat dan API.
3. Mengidentifikasi dan menangani respons API dalam simulasi Wokwi.
4. Mengevaluasi efektivitas simulasi Wokwi dalam pengembangan sistem IoT.

HASIL DAN PEMBAHASAN

1. Persiapan Simulasi di Wokwi

Sebelum memulai praktik, diperlukan beberapa langkah persiapan agar simulasi dapat berjalan dengan baik. Pengguna harus memiliki akun Wokwi dan mengakses editor Wokwi untuk menyiapkan simulasi. Selain itu, endpoint API yang akan digunakan dalam praktik harus dipastikan tersedia agar dapat menerima dan merespons permintaan dari simulator.

2. Implementasi dan Pengujian Akses API

Dalam praktik ini, perangkat ESP32 digunakan sebagai perangkat virtual untuk menghubungkan ke internet dan berkomunikasi dengan API melalui protokol HTTP. Setelah konfigurasi jaringan dilakukan, perangkat virtual akan melakukan permintaan HTTP ke API yang telah ditentukan. Permintaan ini bisa berupa pengambilan data (GET) atau pengiriman data (POST). Jika koneksi berhasil, server akan memberikan respons berupa data atau status yang menandakan keberhasilan atau kegagalan permintaan.

3. Analisis Respons API

Setelah mengirim permintaan ke API, simulator akan menampilkan respons yang diterima dalam bentuk kode status dan data yang dikembalikan oleh server. Respons ini menunjukkan apakah permintaan telah diproses dengan benar atau terdapat kesalahan dalam komunikasi. Jika terjadi kesalahan, maka analisis lebih lanjut dilakukan untuk mengidentifikasi penyebabnya, seperti masalah jaringan atau kesalahan dalam endpoint API.

4. Evaluasi Keberhasilan Simulasi

Dari hasil pengujian, dapat dilihat bahwa simulasi Wokwi memungkinkan perangkat virtual untuk menghubungkan ke jaringan dan berkomunikasi dengan server API tanpa perlu perangkat keras fisik. Keberhasilan pengiriman dan penerimaan data menunjukkan bahwa metode ini dapat digunakan sebagai alat uji sebelum implementasi sistem IoT yang sebenarnya. Selain itu, pengembang dapat mengidentifikasi dan memperbaiki kesalahan lebih awal dalam proses pengembangan.

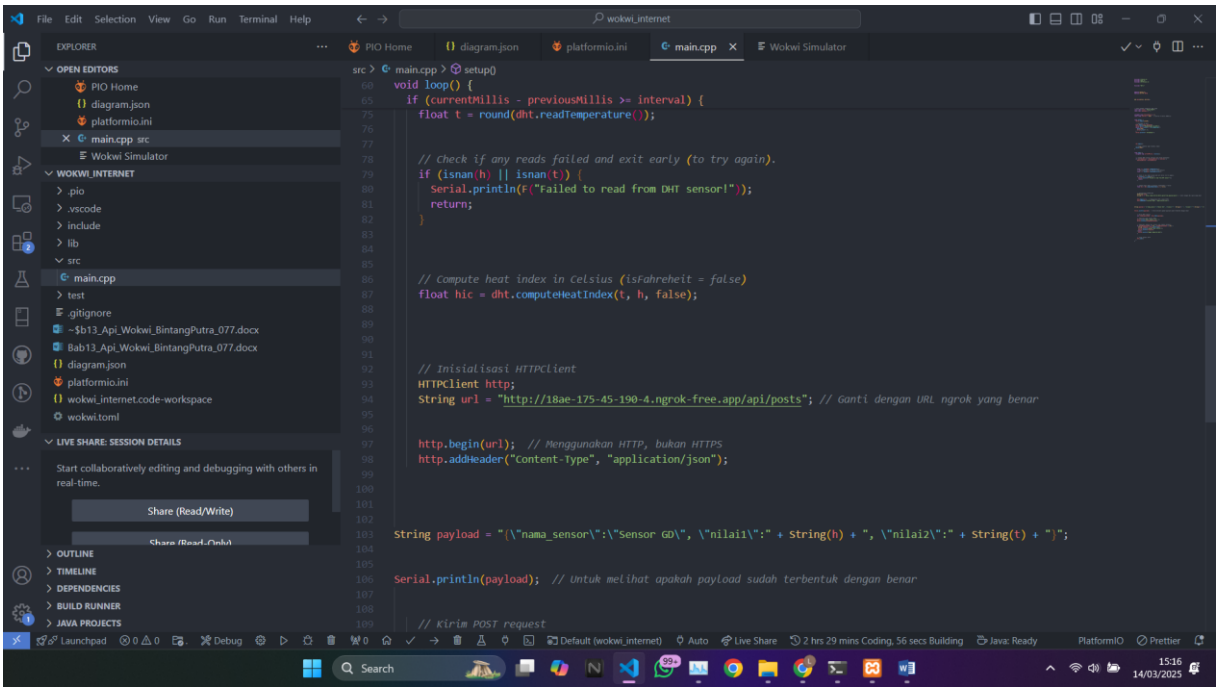
KESIMPULAN

Praktik akses API melalui simulasi Wokwi memberikan gambaran bagaimana perangkat IoT berinteraksi dengan server melalui protokol HTTP. Penggunaan Wokwi sebagai simulator memungkinkan pengembang untuk menguji konektivitas dan respons API tanpa memerlukan perangkat keras fisik. Dari hasil pengujian, dapat disimpulkan bahwa simulasi ini sangat bermanfaat untuk mengembangkan dan menguji fitur IoT sebelum implementasi nyata dilakukan. Dengan demikian, Wokwi dapat menjadi alat yang efektif dalam proses pengembangan sistem berbasis IoT.

BAB 4 LAMPIRAN & DOKUMENTASI

1. Proyek Simulasi Relay, Relay & Button

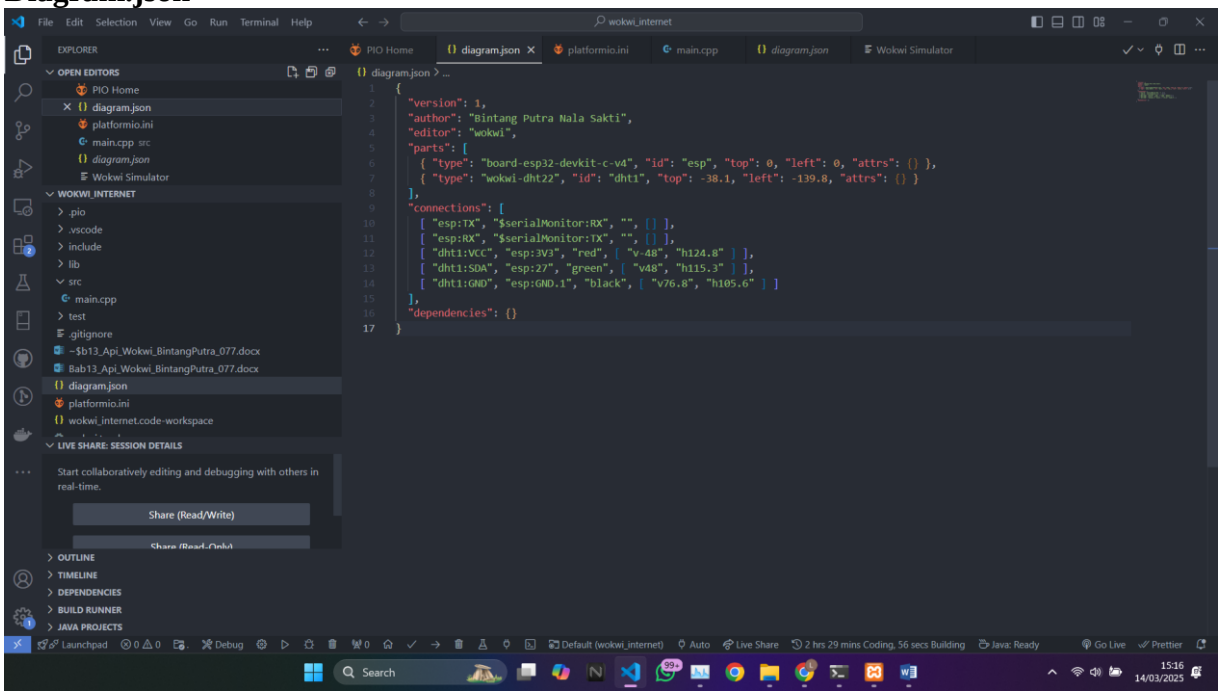
a. Main.cpp




```
#include <Arduino.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include "DHT.h"

// Definisi pin dan tipe sensor DHT22
#define DHTPIN 27
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// Kredensial WiFi
const char* ssid = "Wokwi-GUEST";
const char* password = "";

// URL server ngrok (pastikan tidak ada spasi berlebih)
const char* server = "http://86f5-36-73-241-171.ngrok-free.app/api/posts";

// Timer untuk interval pengiriman data
unsigned long previousMillis = 0;
const long interval = 5000;  // 5 detik

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

  // Hubungkan ke WiFi
  WiFi.begin(ssid, password);
  Serial.print("Menghubungkan ke WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nTerhubung ke WiFi!");

  // Inisialisasi DHT sensor
```

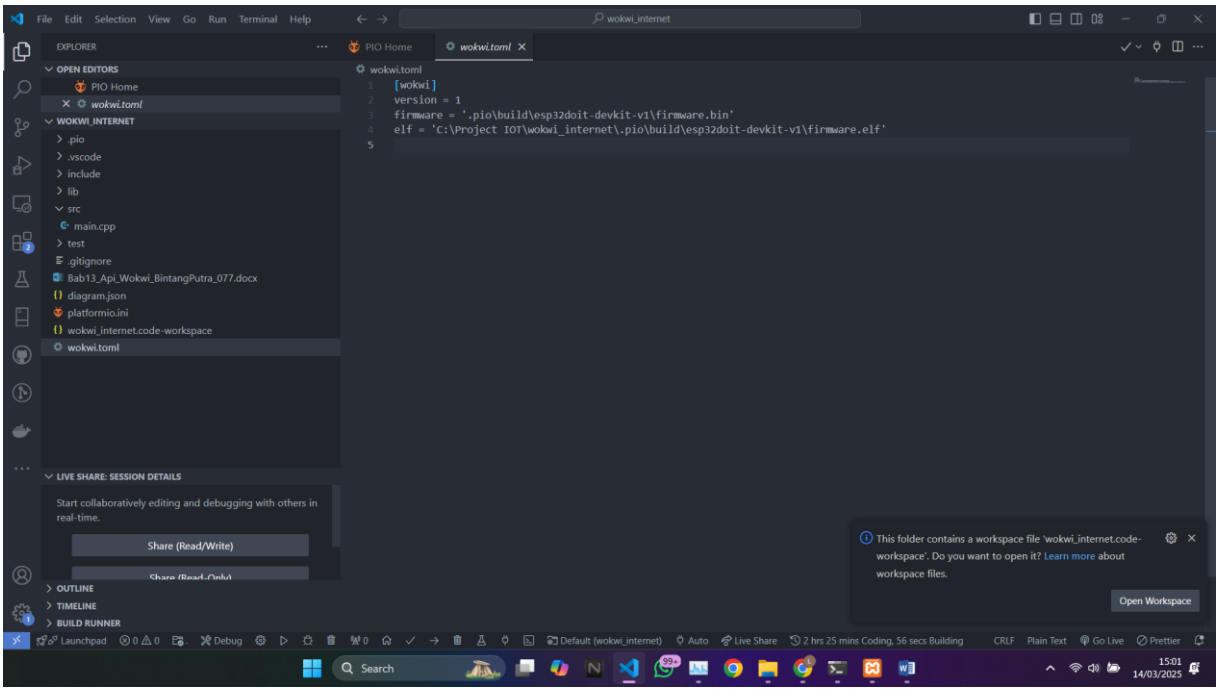
```
  dht.begin();
  delay(1000);
}
void loop() {
  unsigned long currentMillis = millis();

  // Kirim data setiap interval 5 detik
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    // Membaca data dari sensor
    float h = dht.readHumidity();
    float t = dht.readTemperature(); // Celsius

    // Pastikan pembacaan berhasil
    if (isnan(h) || isnan(t)) {
      Serial.println("Gagal membaca data dari sensor DHT!");
      return;
    }
    // Konversi ke integer sebelum mengirimkan data
    int humidity = (int)h;
    int temperature = (int)t;

    // Membuat JSON payload
    String payload = "{\"nama_sensor\":\"Sensor GD\", \"nilai1\":" + String(humidity) + ",
\"nilai2\":" + String(temperature) + "}";

    // Debugging: Cek payload sebelum dikirim
    Serial.println("Payload JSON:");
    Serial.println(payload);

    // Inisialisasi HTTP Client
    HTTPClient http;
    http.begin(server);  // Menggunakan HTTP, bukan HTTPS
    http.addHeader("Content-Type", "application/json");

    // Kirim POST request
    int httpResponseCode = http.POST(payload);

    // Menampilkan hasil respons HTTP
    Serial.print("Kode respons HTTP: ");
    Serial.println(httpResponseCode);

    if (httpResponseCode == 200 || httpResponseCode == 201) {
      String response = http.getString();
      Serial.println("Respons dari server:");
      Serial.println(response);
    } else {
      Serial.println("Gagal mengirim data ke server!");
    }
    // Tutup koneksi HTTP
    http.end();
  }
}
```

b. Diagram.json



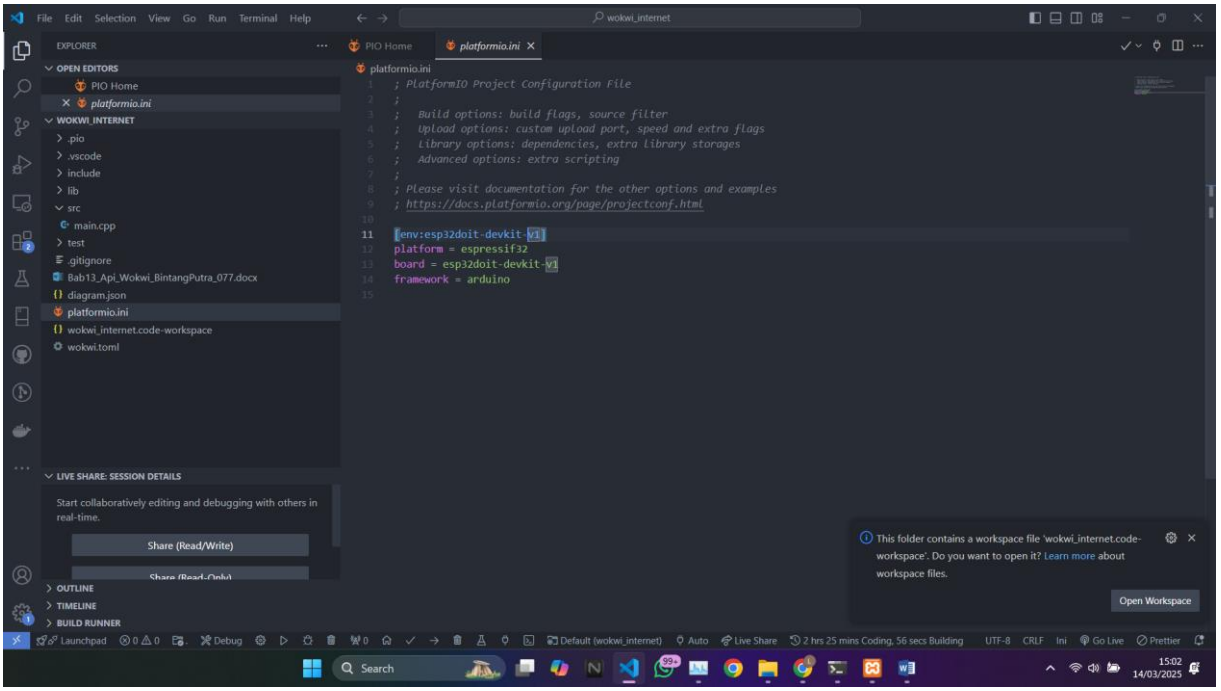

```
{
 "version": 1,
 "author": "Bintang Putra Nala Sakti",
 "editor": "wokwi",
 "parts": [
  { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },
  { "type": "wokwi-dht22", "id": "dht1", "top": -38.1, "left": -139.8, "attrs": {} }
 ],
 "connections": [
  [ "esp:TX", "$serialMonitor:RX", "", [] ],
  [ "esp:RX", "$serialMonitor:TX", "", [] ],
  [ "dht1:VCC", "esp:3V3", "red", [ "v-48", "h124.8" ] ],
  [ "dht1:SDA", "esp:27", "green", [ "v48", "h115.3" ] ],
  [ "dht1:GND", "esp:GND.1", "black", [ "v76.8", "h105.6" ] ]
 ],
 "dependencies": {}
}
```


C. Wokwi.toml



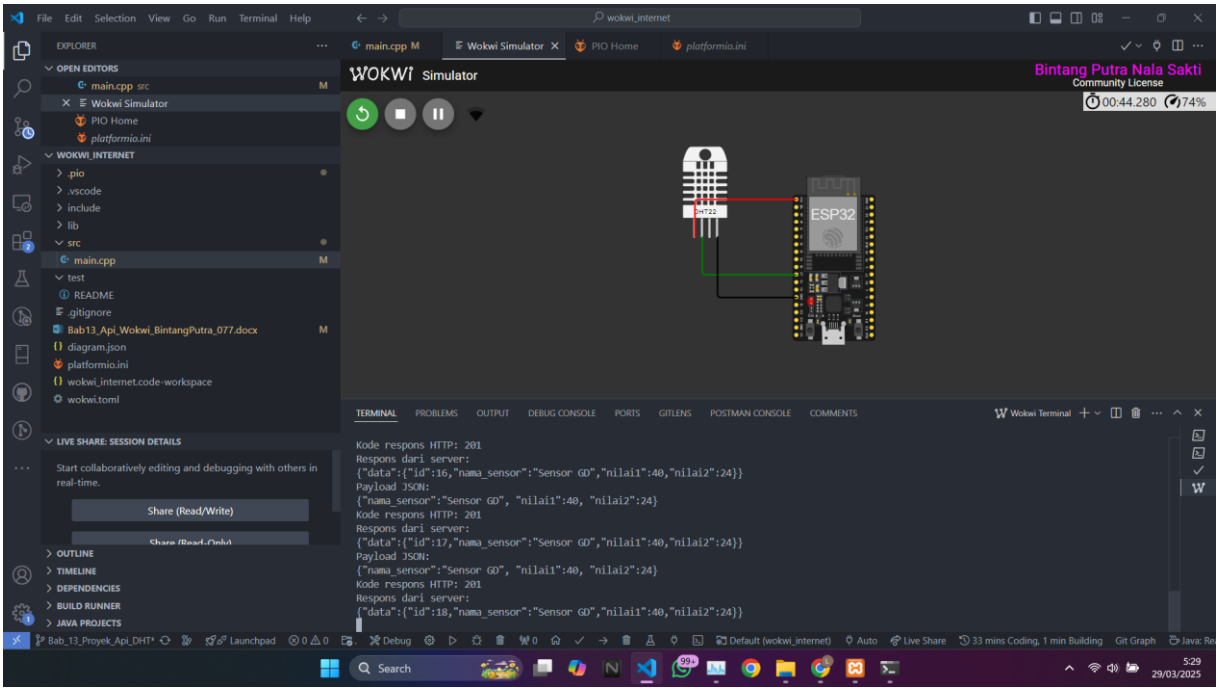
[wokwi]
version = 1
firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'
elf = 'C:\Project IOT\wokwi_internet\.pio\build\esp32doit-devkit-v1\firmware.elf'

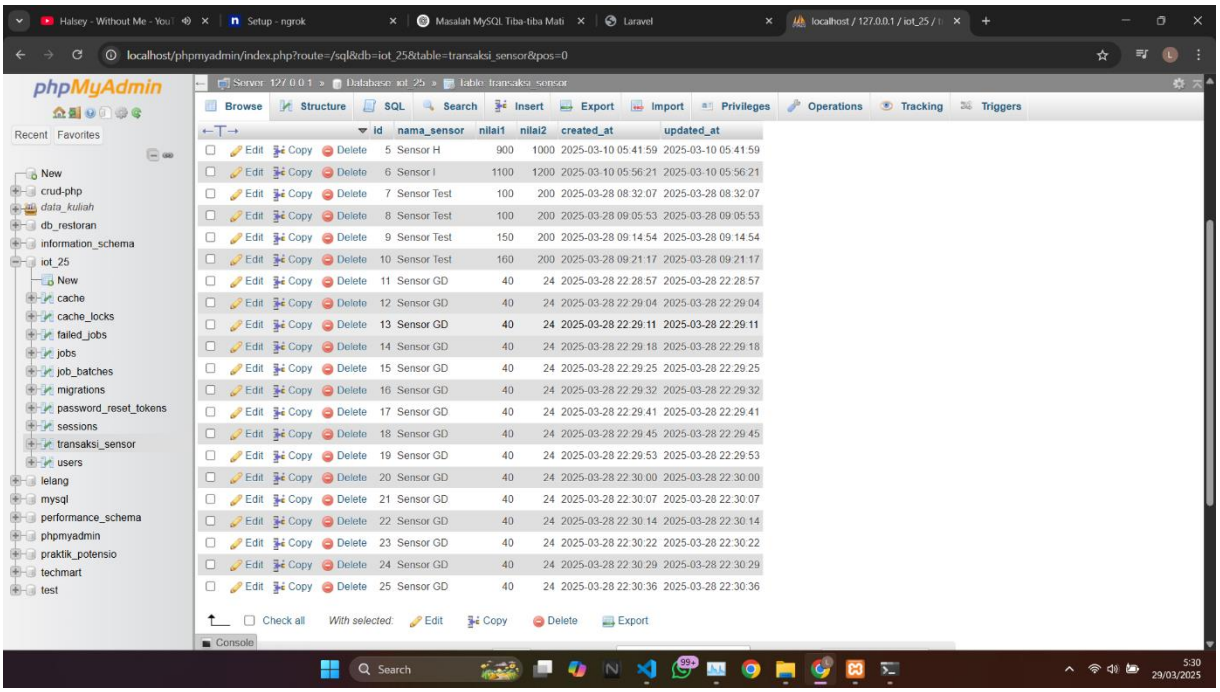
D.Platformio

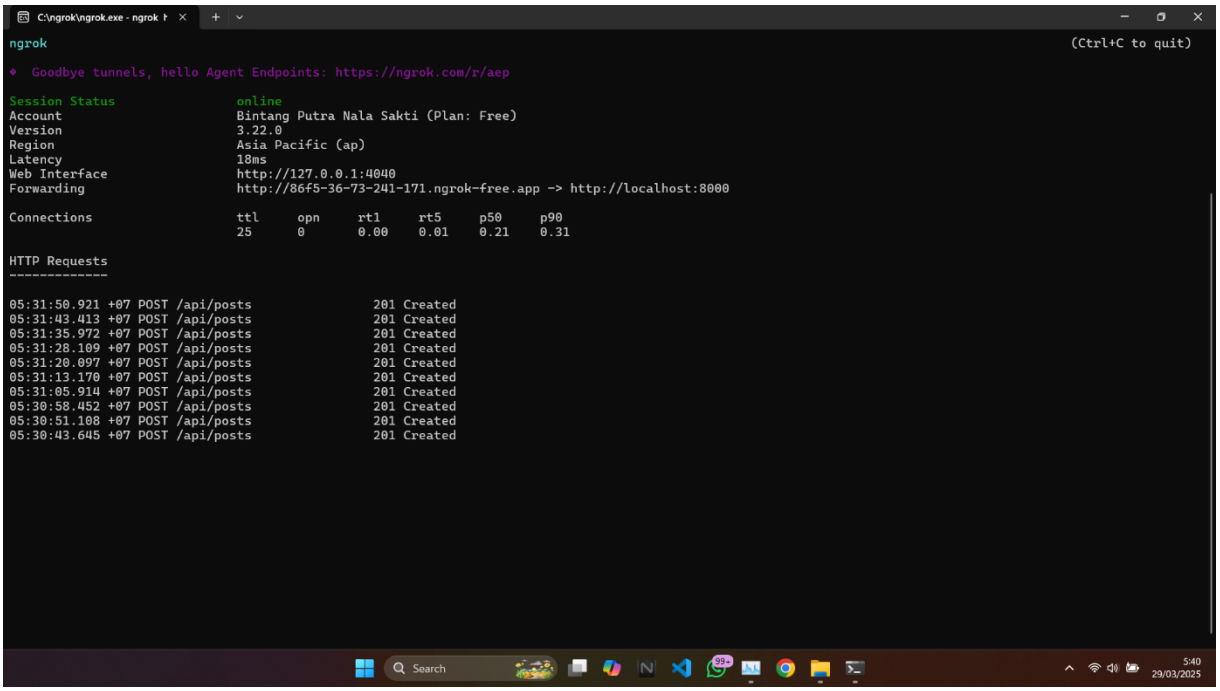


[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
monitor_speed = 115200
lib_deps = adafruit/DHT sensor library

E. Dokumentasi Hasil Kodingan






C:\ngrok\ngrok.exe - ngrok
ngrok
(Ctrl+C to quit)
Goodbye tunnels, hello Agent Endpoints: https://ngrok.com/r/aep
Session Status online
Account Bintang Putra Nala Sakti (Plan: Free)
Version 3.22.0
Region Asia Pacific (ap)
Latency 18ms
Web Interface http://127.0.0.1:4040
Forwarding http://86f5-36-73-241-171.ngrok-free.app -> http://localhost:8000
Connections ttl opn rt1 rt5 p50 p90
25 0 0.00 0.01 0.21 0.31
HTTP Requests
05:31:50.921 +07 POST /api/posts 201 Created
05:31:43.413 +07 POST /api/posts 201 Created
05:31:35.972 +07 POST /api/posts 201 Created
05:31:28.109 +07 POST /api/posts 201 Created
05:31:20.097 +07 POST /api/posts 201 Created
05:31:13.170 +07 POST /api/posts 201 Created
05:31:05.914 +07 POST /api/posts 201 Created
05:30:58.452 +07 POST /api/posts 201 Created
05:30:51.108 +07 POST /api/posts 201 Created
05:30:43.645 +07 POST /api/posts 201 Created
Search
5:40
29/03/2025