LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**

*Nur Haliza Khairotun Nisa’*

Fakultas Vokasi, Universitas Brawijaya

[leezhaamail@gmail.com](mailto:leezhaamail@gmail.com)

**Abstract (Abstrak)**

Praktik ini bertujuan untuk mempelajari cara mengakses API menggunakan simulasi WOKWI dengan mikrokontroler ESP32. Proses dimulai dengan menjalankan server Laravel yang diakses melalui Ngrok untuk menyediakan URL publik dalam bentuk HTTP. Selanjutnya, simulasi dilakukan di platform WOKWI dengan menulis kode program untuk menghubungkan ESP32 ke WiFi dan mengirimkan data suhu serta kelembaban dari sensor DHT22 ke API Laravel. Data yang dikirimkan kemudian disimpan ke database MySQL melalui endpoint API. Hasil praktik menunjukkan bahwa ESP32 berhasil terhubung ke WiFi, mengirimkan data ke API, dan menerima respons HTTP 200, yang menandakan keberhasilan proses pengiriman data. Praktik ini memberikan wawasan tentang pentingnya integrasi antara perangkat IoT dan API dalam pengembangan aplikasi berbasis Internet of Things.

*Keywords—Internet of Things, ESP32, API, Laravel, Ngrok*

1. **Introduction (Pendahuluan)**

**1.1 Latar Belakang**

Dalam pengembangan Internet of Things (IoT), komunikasi antara perangkat keras seperti mikrokontroler ESP32 dan server backend menjadi elemen kunci. API (Application Programming Interface) digunakan sebagai jembatan untuk mengirim dan menerima data antara perangkat IoT dan server. Framework Laravel telah menjadi salah satu pilihan populer untuk membangun API karena kemudahan penggunaan dan struktur kode yang bersih. Namun, untuk memastikan API dapat diakses oleh perangkat IoT secara online, layanan Ngrok digunakan untuk membuat tunnel ke server lokal sehingga API dapat diakses melalui internet.

Simulasi WOKWI digunakan untuk menguji integrasi antara ESP32, sensor DHT22, dan API Laravel. Dalam praktik ini, data suhu dan kelembaban dari sensor DHT22 dikirimkan ke API Laravel melalui HTTP POST request dan disimpan ke database MySQL.

**1.2 Tujuan Eksperimen**

Praktik ini bertujuan untuk memberikan pemahaman menyeluruh tentang cara mengakses API menggunakan simulasi WOKWI dengan mikrokontroler ESP32 dan sensor DHT22 dalam aplikasi Internet of Things (IoT). Melalui praktik ini, mahasiswa akan mempelajari langkah-langkah integrasi antara perangkat keras seperti ESP32 dan server backend Laravel melalui API. Praktik dimulai dengan menjalankan server Laravel yang diakses melalui Ngrok untuk menyediakan URL publik dalam bentuk HTTP, kemudian dilanjutkan dengan pengembangan kode program di platform WOKWI untuk membaca data suhu dan kelembaban dari sensor DHT22 dan mengirimkannya ke API Laravel menggunakan HTTP POST request. Data yang dikirimkan kemudian disimpan ke database MySQL melalui endpoint API. Selain itu, praktik ini juga bertujuan untuk menjelaskan pentingnya API dalam mendukung komunikasi antar perangkat IoT serta memberikan wawasan tentang penggunaan alat bantu seperti Ngrok dan WOKWI dalam pengembangan IoT secara efisien. Dengan demikian, mahasiswa diharapkan dapat memahami konsep dasar integrasi perangkat IoT dengan API dan penerapannya dalam skenario nyata.

1. **Methodology (Metodologi)**

**2.1 Tools & Materials (Alat dan Bahan)**

1. Komputer atau laptop dengan koneksi internet.
2. Platform WOKWI (https://wokwi.com ).
3. Visual Studio Code (VSCode) dengan ekstensi PlatformIO.
4. Mikrokontroler ESP32 (simulasi di WOKWI).
5. Sensor DHT22 (simulasi di WOKWI).
6. Laravel 11 (framework backend).
7. Ngrok (layanan tunneling).
8. Database MySQL (menggunakan phpMyAdmin).

**2.2 Implementation Steps (Langkah Implementasi)**

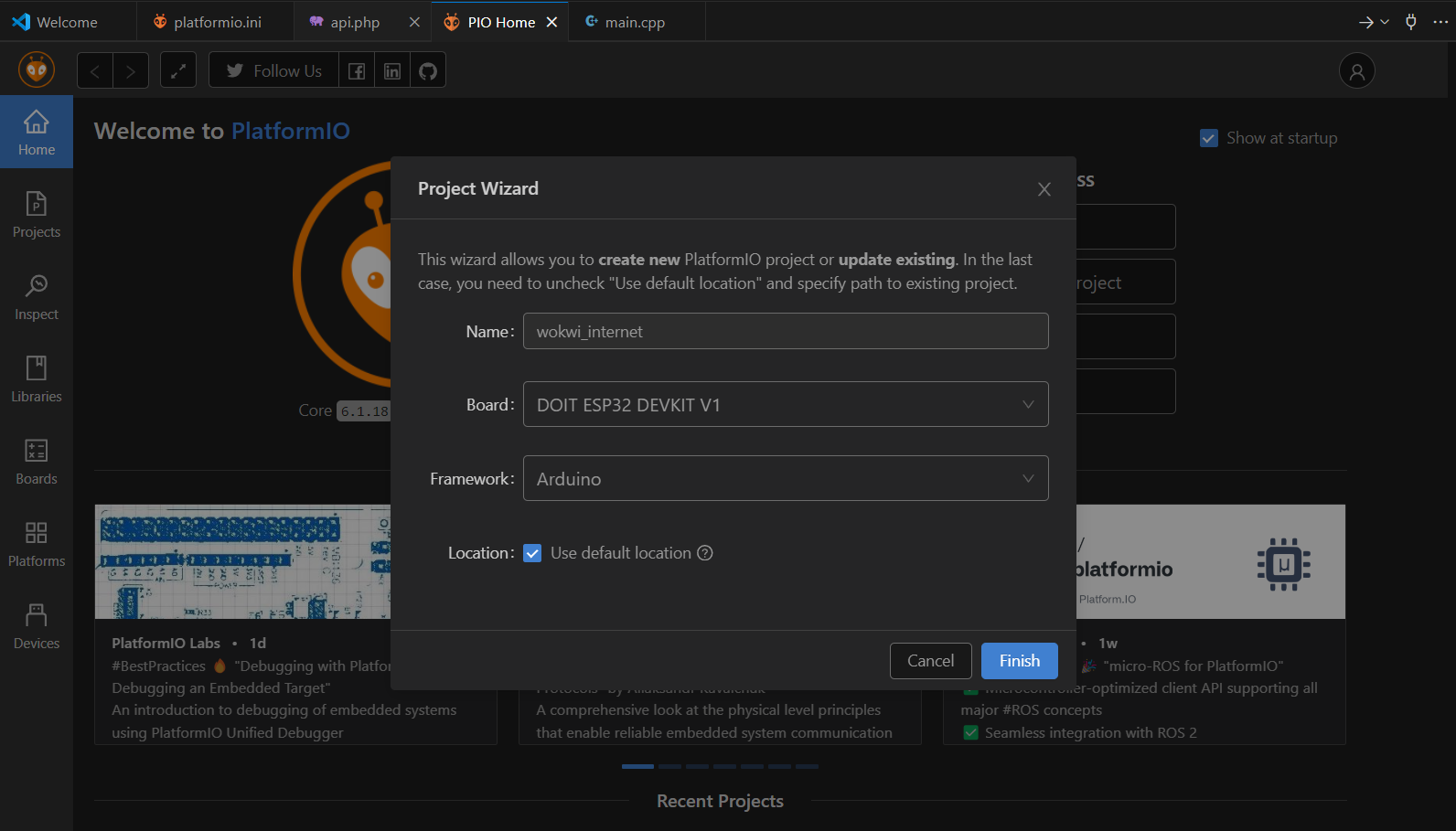
1. Menjalankan Server Laravel
2. Jalankan server Laravel dengan perintah berikut:

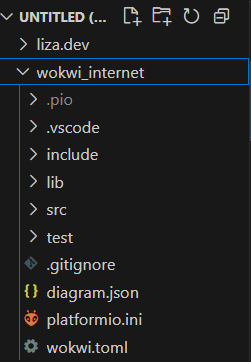
php artisan serve --host=0.0.0.0 --port=8000

1. Pastikan server dapat diakses dari IP address manapun pada port 8000.
2. Mengonfigurasi Ngrok
3. Jalankan perintah berikut untuk membuat tunnel ke server lokal Laravel:



1. Menyiapkan file baru wokwi simulator di platform.io
2. Buat resource untuk transformasi data:





1. Tambahkan script main.cpp sebagai berikut :

#include <WiFi.h>

#include <HTTPClient.h>

// Ganti dengan kredensial jaringan Wi-Fi Anda

// const char\* ssid = "Lab IT";

// const char\* password = "labit2024";

 const char\* ssid = "Wokwi-GUEST";

 const char\* password = "";

// URL lengkap server yang akan diakses

const char\* serverUrl = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts";

// Interval waktu antara setiap permintaan (dalam milidetik)

const unsigned long interval = 5000;

unsigned long previousMillis = 0;

void setup() {

  Serial.begin(115200);

  WiFi.begin(ssid, password);

  Serial.print("Menghubungkan ke WiFi...");

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  Serial.println(" Terhubung!");

}

void loop() {

  unsigned long currentMillis = millis();

  // Periksa apakah interval waktu telah berlalu

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    if (WiFi.status() == WL\_CONNECTED) {

      HTTPClient http;

      // Inisialisasi HTTPClient dengan URL server

      http.begin(serverUrl);

      // Mengirim permintaan HTTP GET

      int httpResponseCode = http.GET();

      // Menampilkan kode status HTTP

      Serial.print("Kode status HTTP: ");

      Serial.println(httpResponseCode);

      // Menutup koneksi

      http.end();

    } else {

      Serial.println("WiFi tidak terhubung.");

    }

  }

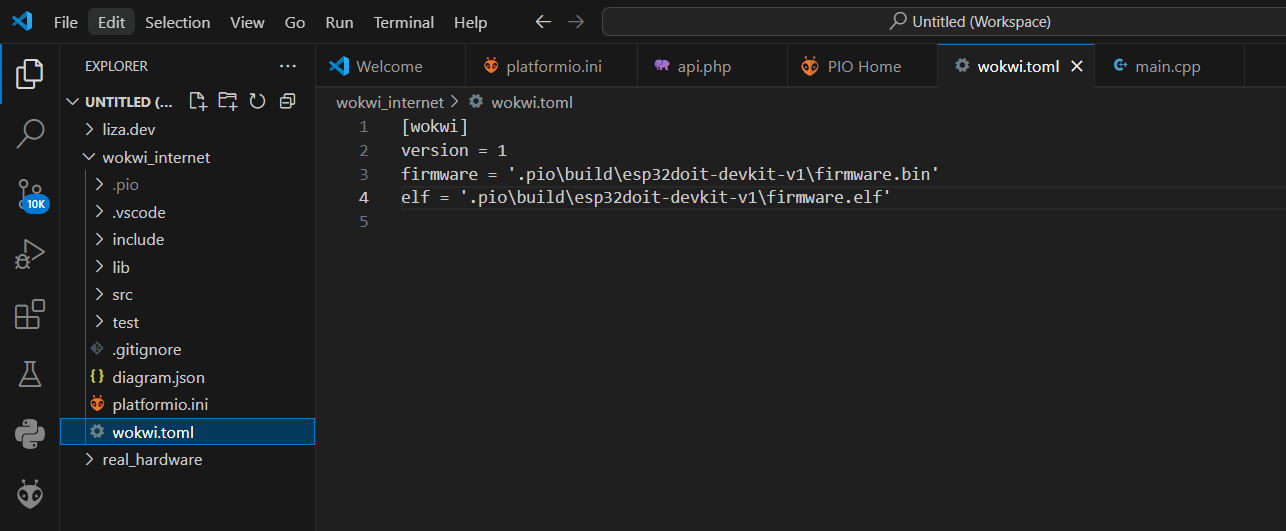
}

1. Perhatikan pada bagian berikut, ubah lah dan sesuaikan lah dengan Alamat url yang diberikan oleh Ngrok.

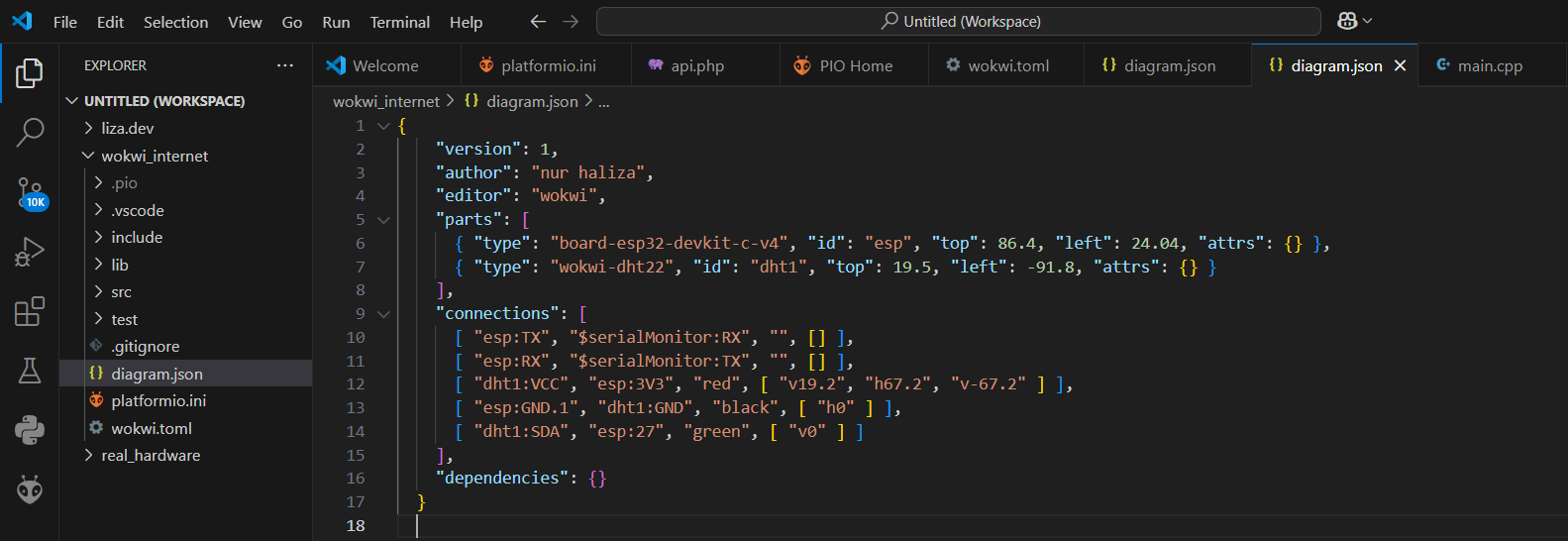
// URL lengkap server yang akan diakses

const char\* serverUrl = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts";

1. Tambahkan file wokwi.toml

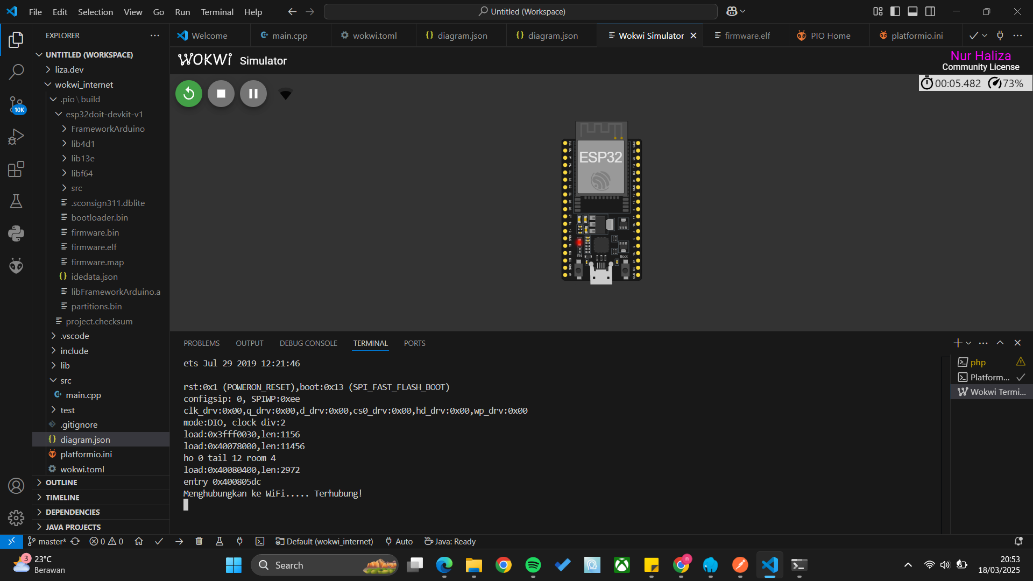


1. Tambahkan Diagram.json

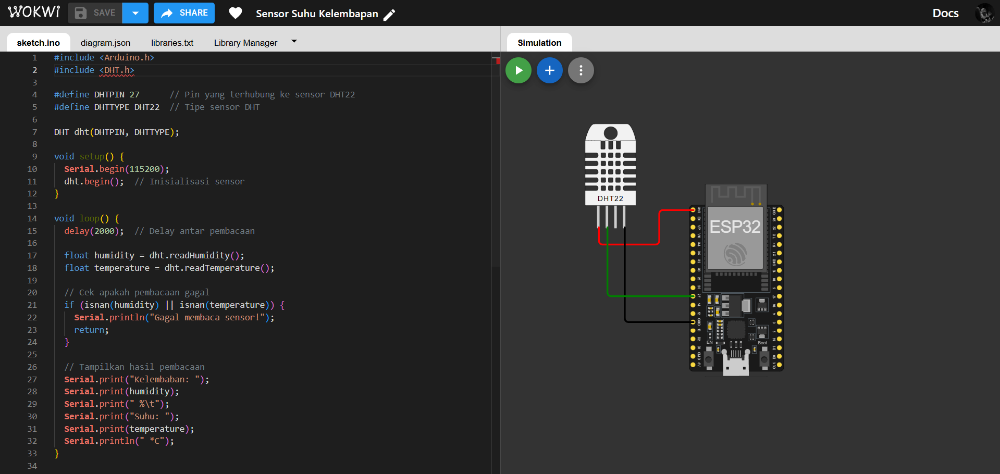


1. Kemudian build file main.cpp dan jalankan simulasi dengan perintah

**> Wokwi Start Simulator**



1. Modifikasi simulasi dengan menambahkan sensor suhu dan kelembaban
2. Rangkai sensor DHT22 dengan ESP32 di Wokwi



1. Salin kode **diagram.json** ke file diagram.json yang ada di vscode.

{

    "version": 1,

    "author": "nur haliza",

    "editor": "wokwi",

    "parts": [

      { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 86.4, "left": 24.04, "attrs": {} },

      { "type": "wokwi-dht22", "id": "dht1", "top": 19.5, "left": -91.8, "attrs": {} }

    ],

    "connections": [

      [ "esp:TX", "$serialMonitor:RX", "", [] ],

      [ "esp:RX", "$serialMonitor:TX", "", [] ],

      [ "dht1:VCC", "esp:3V3", "red", [ "v19.2", "h67.2", "v-67.2" ] ],

      [ "esp:GND.1", "dht1:GND", "black", [ "h0" ] ],

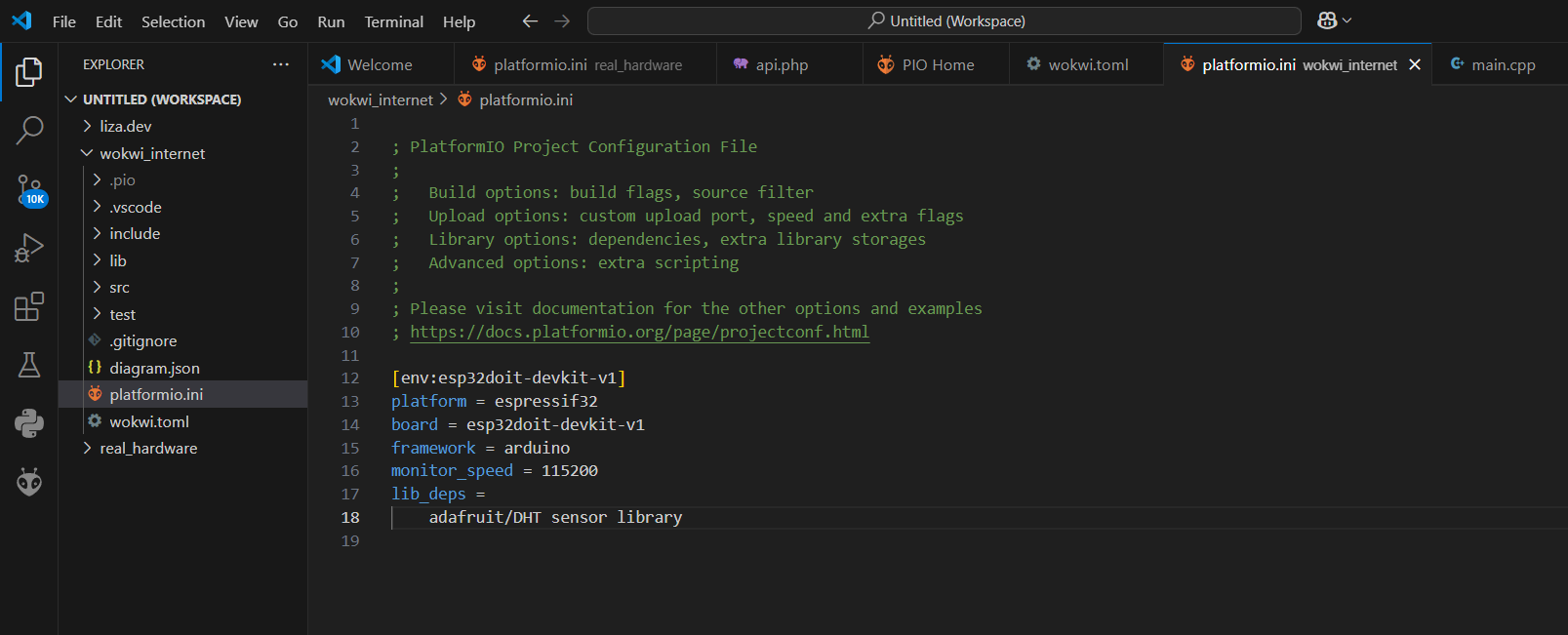
      [ "dht1:SDA", "esp:27", "green", [ "v0" ] ]

    ],

    "dependencies": {}

  }

1. Kemudian ubah setting file **platformio.ini** sebagai berikut :



1. Modifikasi file **main.cpp**

#include <Arduino.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 27

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// Ganti dengan kredensial WiFi Anda

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

unsigned long previousMillis = 0;

const long interval = 5000;  // Interval 5 detik (5000 ms)

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(" Terhubung!");

  dht.begin();

  // Tunggu sebentar agar koneksi stabil

  delay(1000);

}

void loop() {

  unsigned long currentMillis = millis();

  // Lakukan POST setiap interval yang telah ditentukan

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    float h = round(dht.readHumidity());

    // Read temperature as Celsius (the default)

    float t = round(dht.readTemperature());

    // Check if any reads failed and exit early (to try again).

    if (isnan(h) || isnan(t)) {

      Serial.println(F("Failed to read from DHT sensor!"));

      return;

    }

    // Compute heat index in Celsius (isFahreheit = false)

    float hic = dht.computeHeatIndex(t, h, false);

    // Inisialisasi HTTPClient

    HTTPClient http;

    String url = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar

    http.begin(url);  // Menggunakan HTTP, bukan HTTPS

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

String payload = "{\"nama\_sensor\":\"Sensor GD\", \"nilai1\":" + String(h) + ", \"nilai2\":" + String(t) + "}";

Serial.println(payload);  // Untuk melihat apakah payload sudah terbentuk dengan benar

    // Kirim POST request

    int httpResponseCode = http.POST(payload);

    // Tampilkan kode respons HTTP

    Serial.print("Kode respons HTTP: ");

    Serial.println(httpResponseCode);

    // Tampilkan respons dari server jika request berhasil

    if (httpResponseCode == 200 || httpResponseCode == 201) {

      String response = http.getString();

      Serial.println("Respons dari server:");

      Serial.println(response);

    } else {

      Serial.println("Gagal mengirim data");

    }

    // Tutup koneksi HTTP

    http.end();

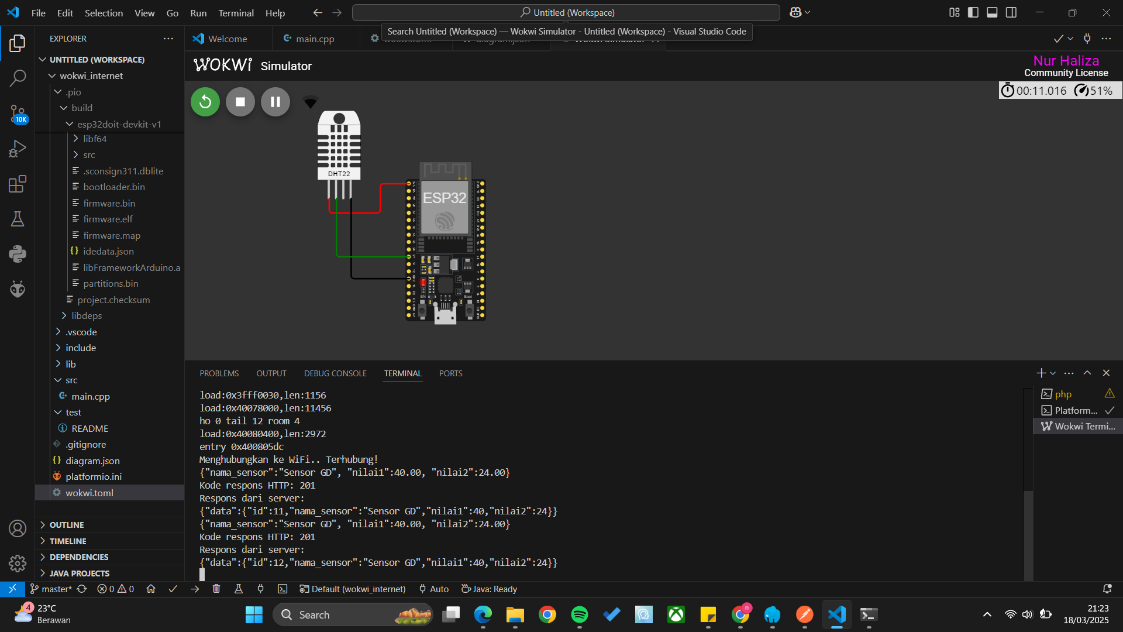
  }

}

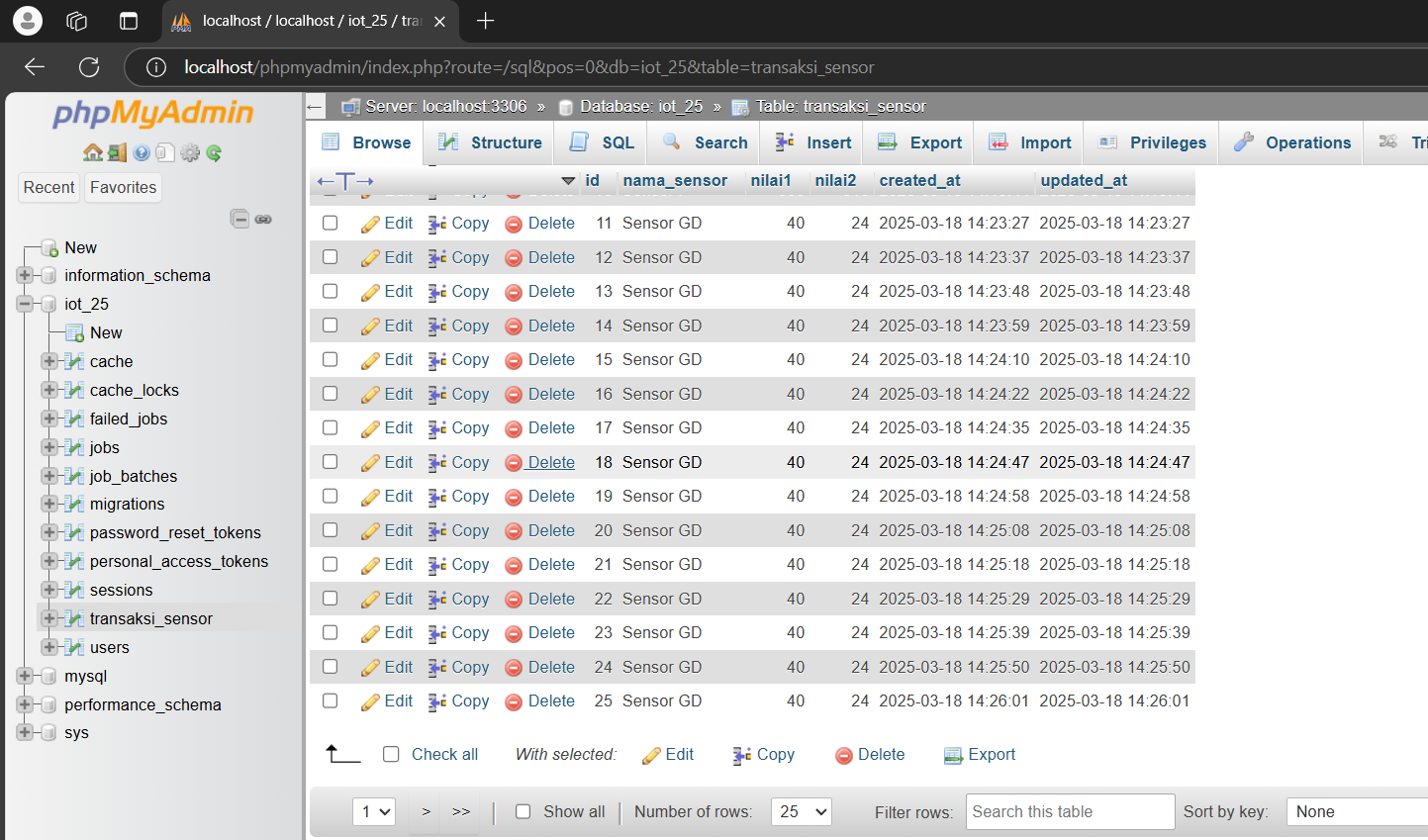
1. Pada bagian berikut sesuaikan dengan URL Ngrok anda.

"http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar

1. Jalankan simulasi Wokwi
2. Tampilan pada simulator dan serial monitor :



1. Tanpilan pada database

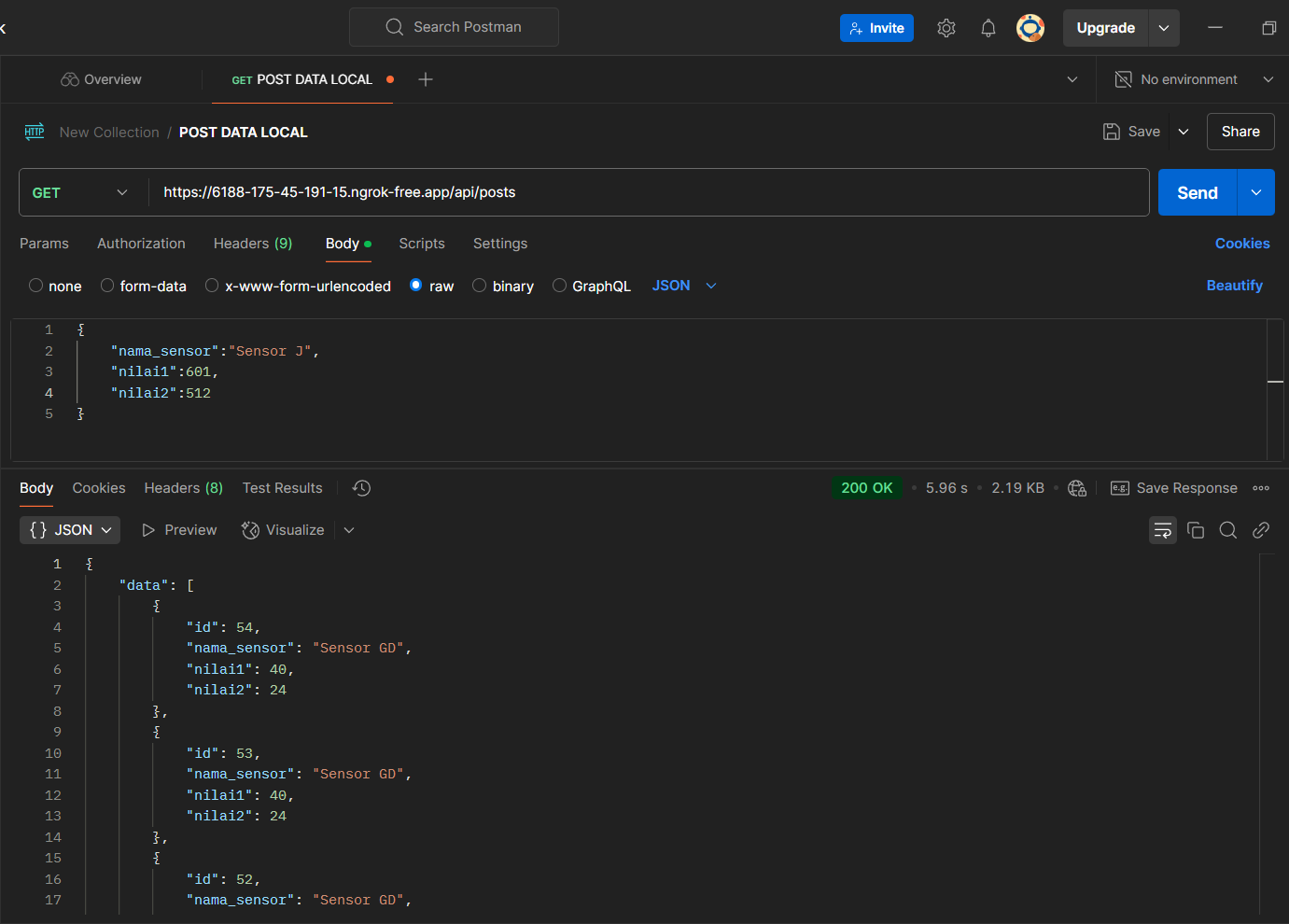


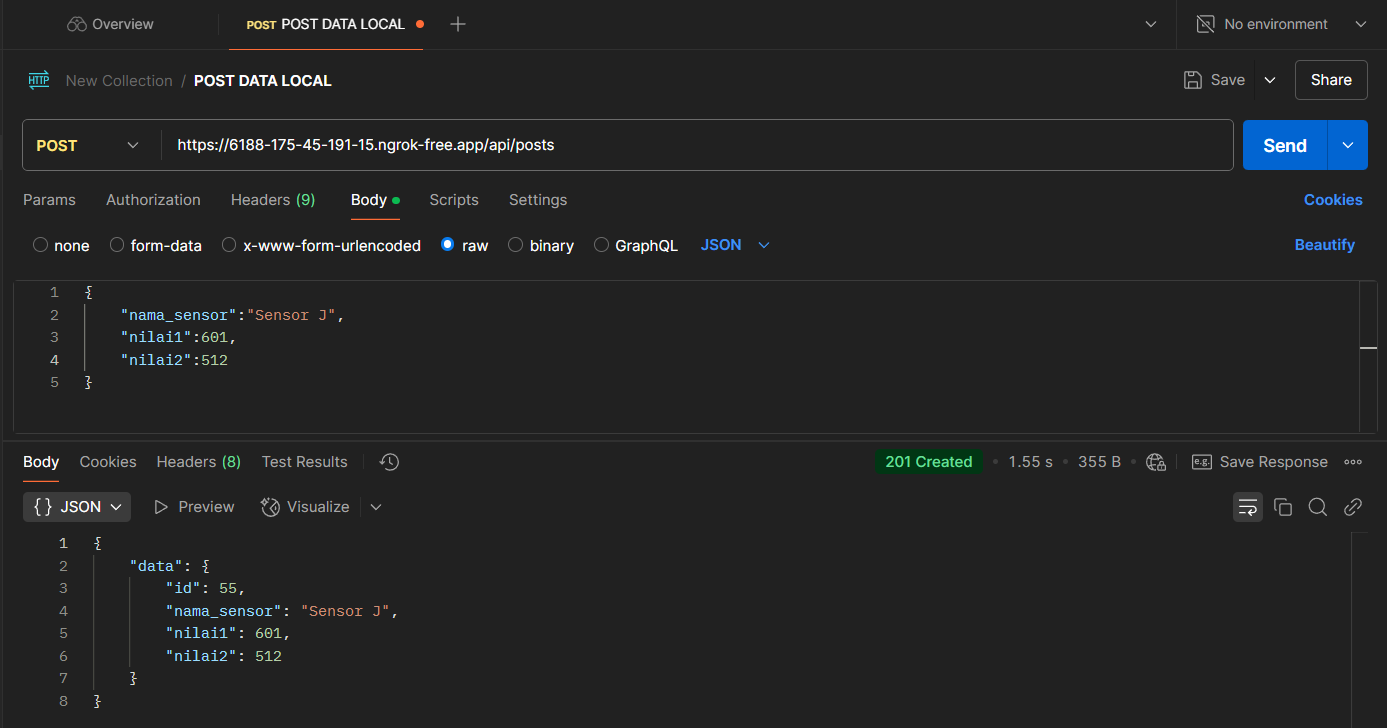
1. **Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

Berikut adalah hasil dari praktik :

1. Skematik Proyek : Berhasil dirancang dengan sensor DHT22 terhubung ke pin GPIO ESP32.
2. Logika Pemrograman : Kode program berhasil diimplementasikan di VSCode dan diintegrasikan dengan WOKWI Simulator.
3. Simulasi : Sistem berhasil membaca data suhu dan kelembaban dari sensor DHT22, mengirimkannya ke API Laravel melalui HTTP POST request, dan menerima respons HTTP 200. Data berhasil disimpan ke database MySQL. Contoh respons JSON dari API:
4. **Appendix (Lampiran, jika diperlukan)**
5. Tampilan GET and POST postman ke database





c. Tampilan hasil GET dan POST di database

