**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**

**Akses API Melalui Simulasi WOKWI**

Author : Nelly Fananda Melani

Fakultas Vokasi, Universitas Brawijaya

Email: [nellyfananda15@student.ub.ac.id](mailto:nellyfananda15@student.ub.ac.id)

**Abstract**

This practicum aims to develop and implement an API using Laravel 11 and integrate it with the Ngrok service for public access. The API is designed to communicate with an Internet of Things (IoT) simulation using Wokwi, enabling the transmission and storage of temperature and humidity sensor data into a MySQL database. The development is carried out in the Visual Studio Code (VSCode) environment, utilizing Postman for API testing, XAMPP as the database server, and the Command Prompt (CMD) terminal for command execution. The Laravel API is launched using the command php artisan serve --host=0.0.0.0 --port=8080, ensuring accessibility from any IP address on port 8080. To connect the API with IoT devices, Ngrok is configured using the command ngrok http --scheme=http 8080, ensuring that the generated URL is in HTTP format for ESP32 compatibility. In the simulation, the IoT device, simulated via Wokwi, successfully connects to the virtual WiFi network Wokwi-GUEST and accesses the Laravel API. Temperature and humidity data obtained from the DHT sensor are transmitted to the API and stored in the database. To optimize communication, a monitor speed setting is used to track data transmission status, and lib\_deps is configured to manage sensor dependencies. The results of this practicum demonstrate that the Laravel 11-based API effectively receives and stores data from IoT devices over the network. This implementation provides a deeper understanding of modern web service development and API connectivity within the IoT ecosystem. The study highlights the importance of APIs in facilitating seamless communication between devices and services, making it a valuable skill for future advancements in IoT technology.

**Keywords**— Laravel 11, API, Ngrok, Postman, XAMPP, Wokwi, ESP32, DHT Sensor, Internet of Things (IoT).

**1. Introduction (Pendahuluan)**

Perkembangan pesat teknologi Internet of Things (IoT) dan aplikasi berbasis web telah meningkatkan kebutuhan akan Application Programming Interface (API) yang efisien dan mudah diakses. API memungkinkan pertukaran data antara berbagai sistem dan perangkat, menjadikannya komponen penting dalam pengembangan aplikasi modern, termasuk sistem IoT, otomasi, dan layanan berbasis cloud.

Laravel adalah salah satu framework PHP paling populer untuk pengembangan web, termasuk pembuatan API. Versi terbaru, Laravel 11, menawarkan berbagai fitur yang mempermudah pengembangan API, seperti dukungan bawaan untuk routing, middleware, autentikasi, dan manajemen database yang lebih efisien. Namun, salah satu tantangan utama dalam pengembangan API adalah pengujian dan akses API dari perangkat eksternal tanpa harus menerapkannya ke server langsung.

Untuk mengatasi tantangan ini, Ngrok digunakan sebagai alat tunneling yang memungkinkan API lokal diekspos ke internet tanpa konfigurasi server tambahan. Dengan demikian, pengembang dapat menguji API secara remote, yang sangat praktis untuk proses pengembangan dan debugging. Selain itu, Postman dimanfaatkan sebagai alat untuk mengirim permintaan HTTP dan menganalisis respons API.

Dalam praktikum ini, API dikembangkan menggunakan Laravel 11 dalam lingkungan pengembangan lokal, dengan Visual Studio Code (VSCode) sebagai editor utama, XAMPP sebagai server basis data, dan Command Prompt (CMD) sebagai alat eksekusi perintah. Praktikum mencakup konfigurasi awal Laravel, perancangan model, controller, dan route, serta pengujian endpoint API menggunakan Postman. Ngrok diintegrasikan untuk membuat API dapat diakses secara online tanpa perlu diterapkan ke server eksternal.

Tujuan dari praktikum ini adalah untuk memberikan pemahaman dasar tentang pengembangan API menggunakan Laravel 11, integrasi dengan Ngrok, dan metode pengujian menggunakan Postman. Dengan pemahaman ini, pengembang dapat membangun API yang efisien, aman, dan mudah diakses untuk berbagai perangkat, mendukung pengembangan sistem IoT, aplikasi web, dan teknologi berbasis cloud.

**1.1 Background of the IoT Practicum (Latar Belakang Praktikum IoT)**

Di era transformasi digital, Application Programming Interface (API) memainkan peran penting dalam memungkinkan komunikasi antara berbagai aplikasi dan sistem. API berfungsi sebagai tulang punggung dalam pengembangan aplikasi web dan mobile modern, memungkinkan pertukaran data yang efisien dan integrasi platform yang mulus. Secara khusus, RESTful API banyak digunakan dalam aplikasi Internet of Things (IoT), sistem otomasi, dan layanan berbasis cloud yang memerlukan interaksi data secara real-time.

Salah satu framework yang banyak digunakan untuk pengembangan API adalah Laravel, sebuah framework berbasis PHP yang dikenal karena efisiensi, keamanan, dan kemudahannya. Versi terbaru, Laravel 11, menyediakan fitur yang lebih baik dalam pengembangan API, termasuk autentikasi bawaan, mekanisme routing yang lebih kuat, dan dukungan middleware yang lebih optimal. Namun, dalam tahap pengembangan, sering kali muncul tantangan dalam pengujian API secara lokal, terutama ketika API harus diakses secara remote untuk integrasi dengan sistem eksternal.

Untuk mengatasi tantangan tersebut, Ngrok digunakan sebagai layanan tunneling yang memungkinkan API lokal dapat diakses secara publik melalui internet dengan cara yang aman. Dengan mengintegrasikan Ngrok, pengembang dapat mengekspos API Laravel yang berjalan di localhost tanpa harus menerapkannya ke server eksternal, sehingga mempermudah proses pengujian dan debugging. Selain itu, Postman digunakan sebagai alat utama dalam pengujian API, memungkinkan pengguna untuk mengirim permintaan HTTP, menguji endpoint, dan menganalisis respons API secara interaktif.

Praktikum ini berfokus pada pengembangan dan implementasi API menggunakan Laravel 11 dalam lingkungan pengembangan lokal dengan menggunakan Visual Studio Code (VSCode), XAMPP sebagai server basis data, dan Command Prompt (CMD) sebagai alat eksekusi perintah. Praktikum ini mencakup aspek-aspek dasar dalam pengembangan API, seperti konfigurasi Laravel, pembuatan model, pengaturan controller dan route, serta pengujian endpoint API menggunakan Postman. Dengan memanfaatkan Ngrok, API yang dikembangkan dapat diakses secara remote untuk keperluan pengujian dan integrasi lebih lanjut.

Melalui praktikum ini, peserta diharapkan mendapatkan pengalaman langsung dalam pengembangan API, autentikasi, manajemen basis data, dan akses remote menggunakan Ngrok. Pengetahuan ini menjadi dasar untuk implementasi API yang lebih kompleks, termasuk yang digunakan dalam aplikasi IoT, komputasi awan, dan sistem berbasis web. Dengan memahami cara kerja API dan strategi integrasinya, pengembang akan lebih siap dalam membangun sistem yang skalabel dan aman untuk berbagai kebutuhan teknologi di masa depan.

**1.2 Experimental Objectives (Tujuan Eksperimen)**

Eksperimen ini bertujuan untuk:

1. Membangun dan mengimplementasikan API berbasis Laravel 11 sebagai framework backend yang efisien dan modern untuk mendukung komunikasi data dalam sistem IoT.
2. Memahami konsep dasar API RESTful, termasuk metode HTTP utama (GET, POST, PUT, DELETE) serta mekanisme komunikasi antara klien (IoT device) dan server.
3. Mengkonfigurasi Laravel 11 untuk pengembangan API, mencakup pembuatan model, controller, route, dan middleware guna mengelola data secara efektif dalam aplikasi berbasis IoT.
4. Menguji API menggunakan Postman, memastikan setiap endpoint dapat menangani permintaan (request) dan memberikan respons yang sesuai dengan kebutuhan sistem.
5. Mengintegrasikan API dengan Ngrok, memungkinkan API yang dikembangkan dalam lingkungan lokal dapat diakses secara publik tanpa perlu deployment ke server eksternal, khususnya untuk mendukung pengujian dengan perangkat IoT.
6. Memahami implementasi autentikasi dan keamanan API, termasuk penggunaan token-based authentication yang sesuai dengan kebutuhan aplikasi IoT dan web.
7. Mengimplementasikan koneksi antara API Laravel dan simulasi perangkat IoT pada Wokwi, termasuk pengiriman data suhu dan kelembaban dari sensor DHT ke database MySQL melalui API.
8. Menganalisis peran API dalam pengembangan sistem IoT dan layanan terdistribusi, serta bagaimana API dapat digunakan untuk pertukaran data yang efisien antara perangkat, server, dan aplikasi cloud.

**2. Methodology (Metodologi)**

Eksperimen ini dilakukan dengan membangun, menguji, dan mengintegrasikan API berbasis Laravel 11 dengan Ngrok untuk memungkinkan akses publik dan pengujian menggunakan perangkat IoT pada Wokwi Simulator. Tahapan dalam metodologi ini meliputi:

1. Instalasi dan Konfigurasi: Instal XAMPP, Laravel 11, dan Composer di VSCode, konfigurasi database MySQL di .env, lalu jalankan Laravel.
2. Pembuatan API: Buat Model, Controller, Route, implementasi CRUD, gunakan Migration dan Seeder, serta tambahkan middleware dan validasi.
3. Pengujian API: Uji GET, POST, PUT, DELETE di Postman dan periksa status HTTP serta output JSON.
4. Integrasi Ngrok: Jalankan ngrok http --scheme=http 8080 dan gunakan URL publik dalam Wokwi Simulator.
5. Implementasi Wokwi: Modifikasi main.cpp untuk mengirim suhu & kelembaban ke API dan cek monitor serial.
6. Evaluasi API: Analisis kinerja, keamanan, optimasi query, dan manfaat Laravel 11 serta Ngrok dalam IoT.

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

Eksperimen ini dilakukan secara virtual tanpa memerlukan perangkat keras fisik. Alat dan bahan yang digunakan dalam proyek ini meliputi:

1. Laptop/PC dengan Visual Studio Code sebagai lingkungan pengembangan.
2. Wokwi.com sebagai simulator perangkat IoT.
3. Postman untuk menguji endpoint API.
4. XAMPP sebagai server database MySQL.
5. Browser untuk mengakses API dan simulator.
6. Jaringan internet untuk menjalankan Laravel dan Ngrok.
7. Command Prompt (CMD) atau PowerShell untuk menjalankan perintah Laravel.

**2.2 Implementation Steps (Langkah Implementasi)**

Eksperimen ini dilakukan melalui beberapa langkah utama:

* Nyalakan Xampp (Apache dan MySQL)
* Search Folder LARAVEL-11
* Buka Menggunakan Vscode
* Buka Terminal ketik:

php artisan serve



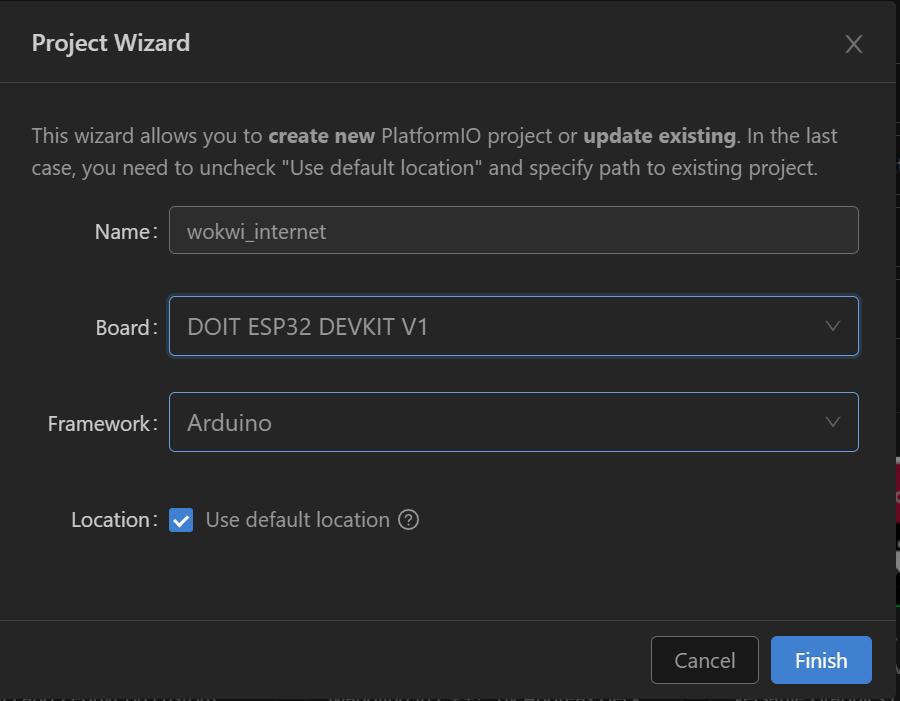
**Praktik Akses API Melalui Simulasi WOKWI**

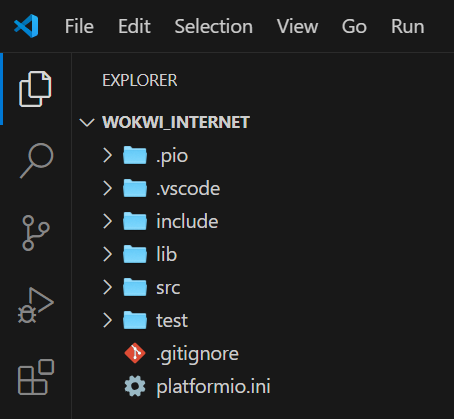
* Jalankan API laravel dengan perintah

**php artisan serve --host=0.0.0.0 --port=8080**

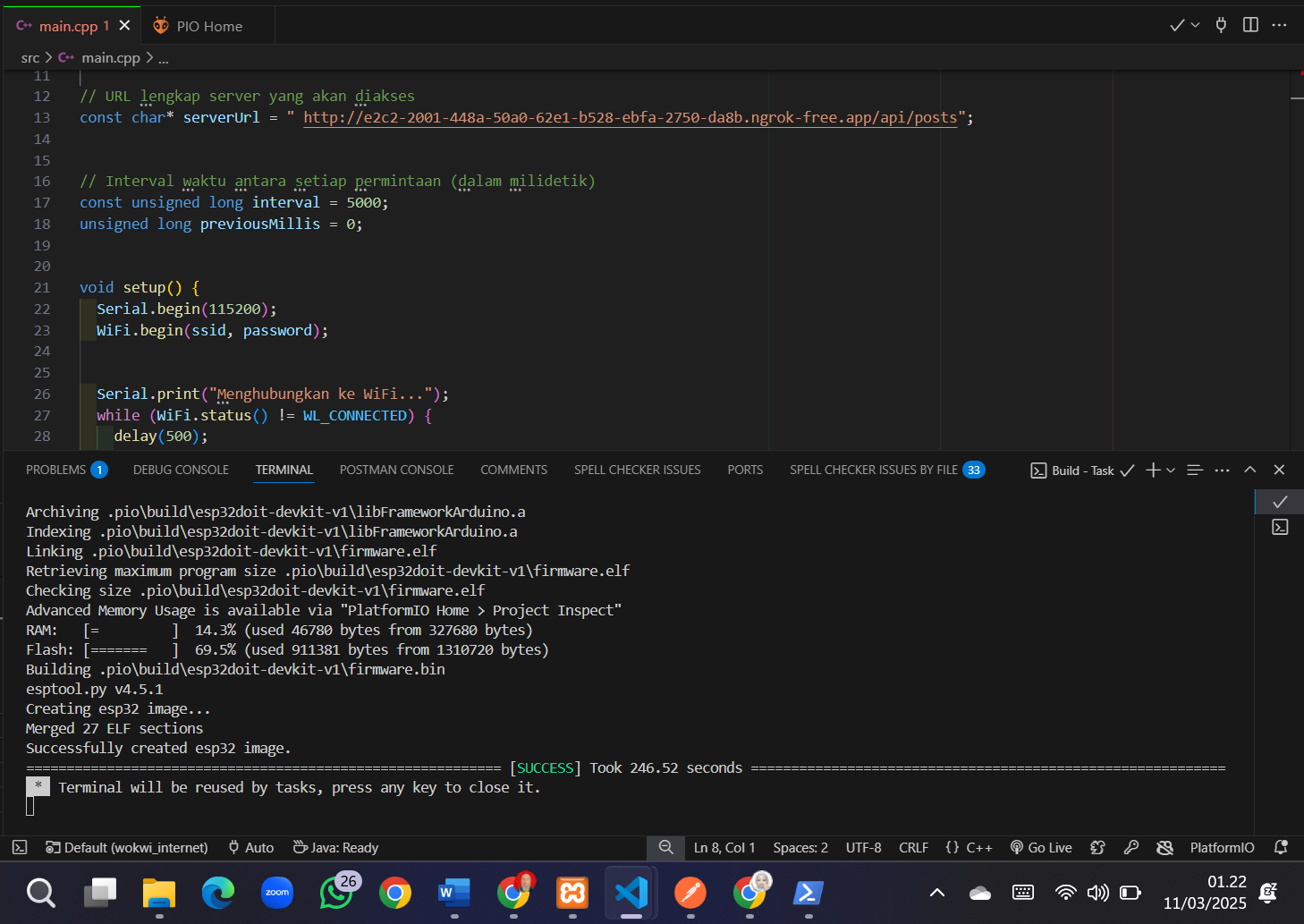
Perintah diatas memastikan API laravel dapat diakses dari IP Address manapun dan memastikan bekerja pada port 8080.

* Buat file baru wokwi simulator di platform.io





* Berikut adalah Script main.cpp



* Berikut adalah script main.cpp

#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://e2c2-2001-448a-50a0-62e1-b528-ebfa-2750-da8b.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.");

    }

  }

}

**Perhatikan pada bagian**

URL lengkap server yang akan diakses

const char\* serverUrl = "http://e2c2-2001-448a-50a0-62e1-b528-ebfa-2750-da8b.ngrok-free.app/api/posts";

URL diatas adalah URL hasil dari generate perintah NGROK di komputer Anda. Sesuaikan dengan alamat URL yang diberikan oleh NGROK. Cara menjalankan perintah NGROK berbeda dengan Bab sebelumnya, perintah berikut memastikan NGROK memberikan alamat URL dalam bentuk http bukan https. Pada saat ini ESP32 yang digunakan hanya support http sehingga pastikan NGROK memberikan URL dalam bentuk http bukan https.

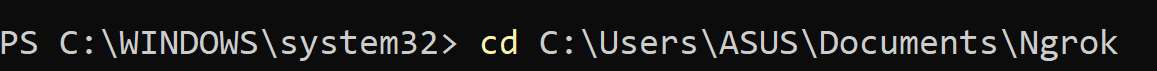
**Steps** tunneling menggunakan Ngrok**:**

Tunneling dengan Ngrok berarti membuat alamat internet sementara untuk mengakses server di komputer

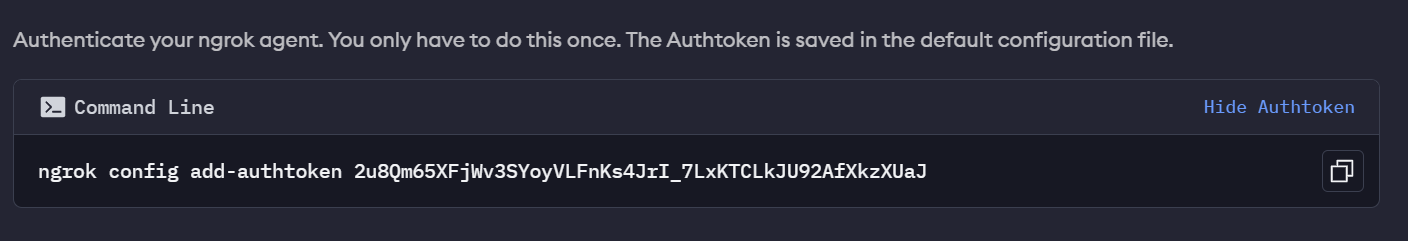
**-> Jalankan PowerShell Administrator**

**-> Masuk folder tempat kita menyimpan atau mengekstrak** **file ngrok.exe**.

cd C:\Users\ASUS\Documents\Ngrok)



**-> Akses dan Copy Authtoken kita melalui** <https://dashboard.ngrok.com/get-started/your-authtoken>



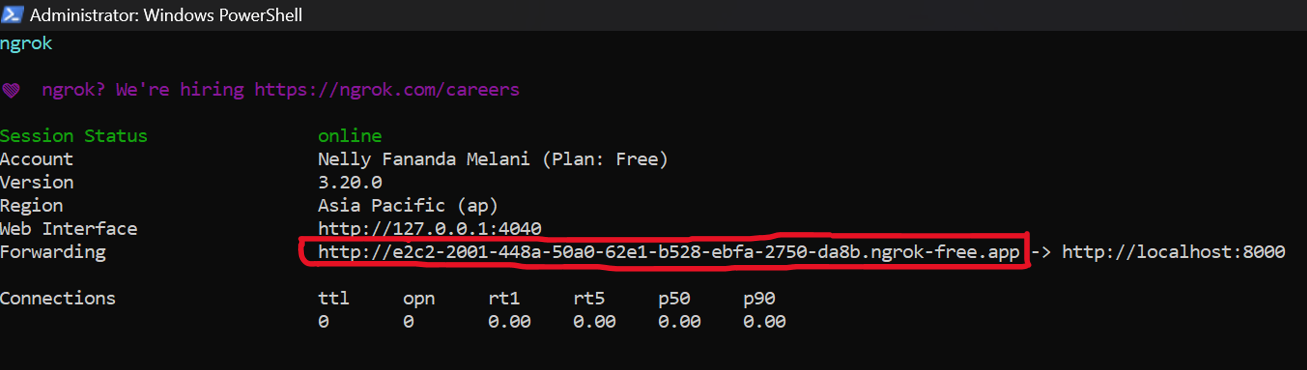
**-> Tambahkan Authtoken ke konfigurasi ngrok di PowerShell:**

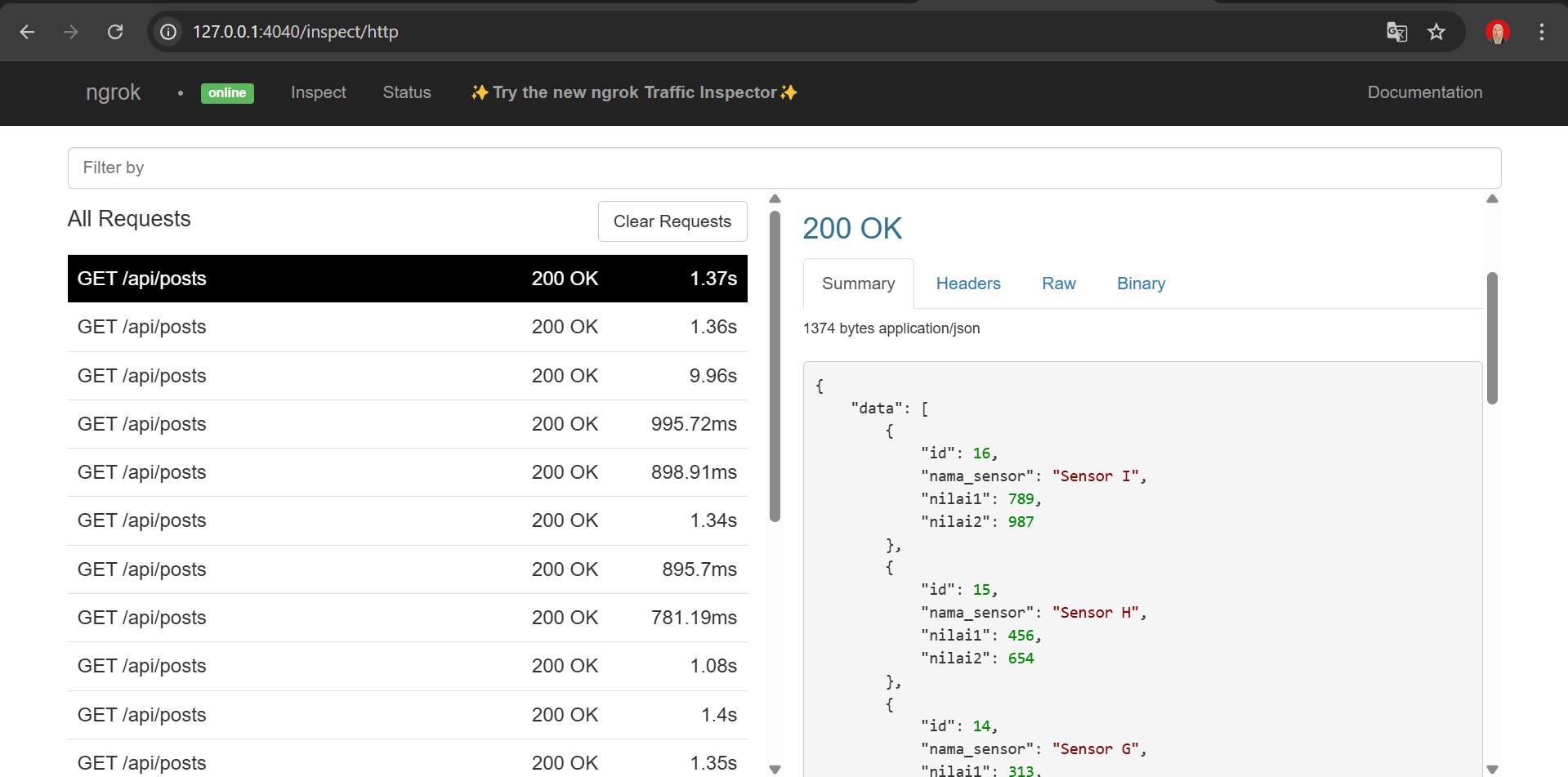
.\ngrok config add-authtoken 2u8Qm65XFjWv3SYoyVLFnKs4JrI\_7LxKTCLkJU92AfXkzXUaJ

 **-> Jalankan perintah untuk membuka akses internet ke server lokal Laravel di port 8000**



.\ngrok http --scheme=http 8000





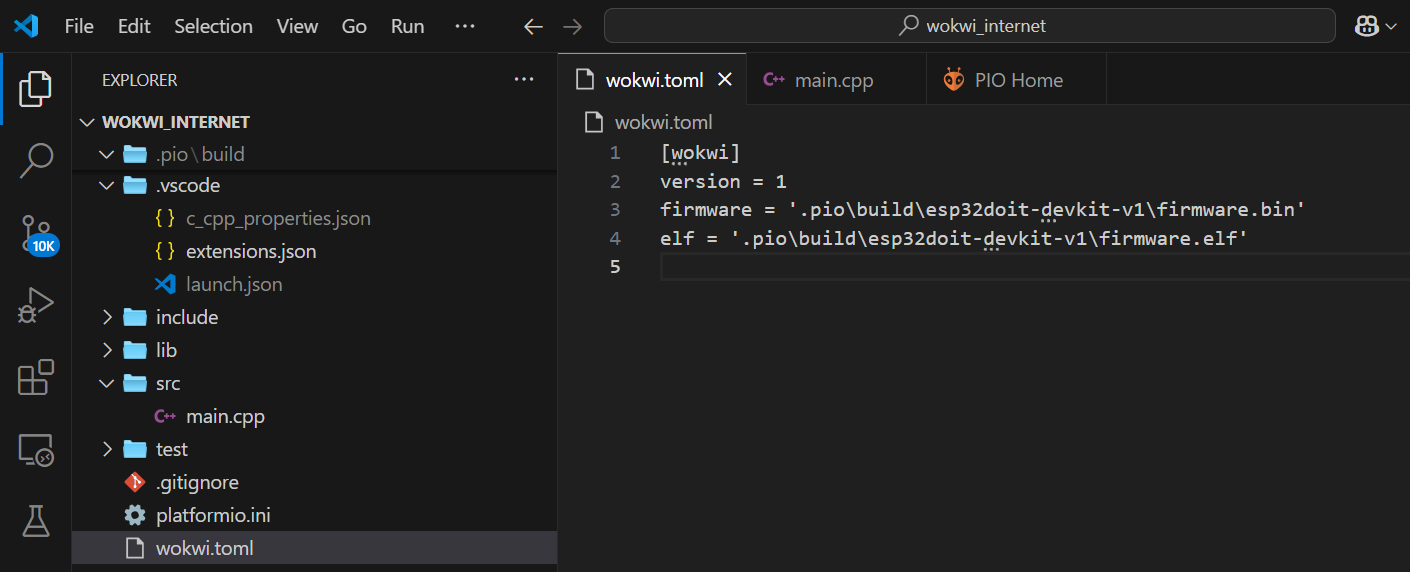
* Tambahkan file **wokwi.toml**

[wokwi]

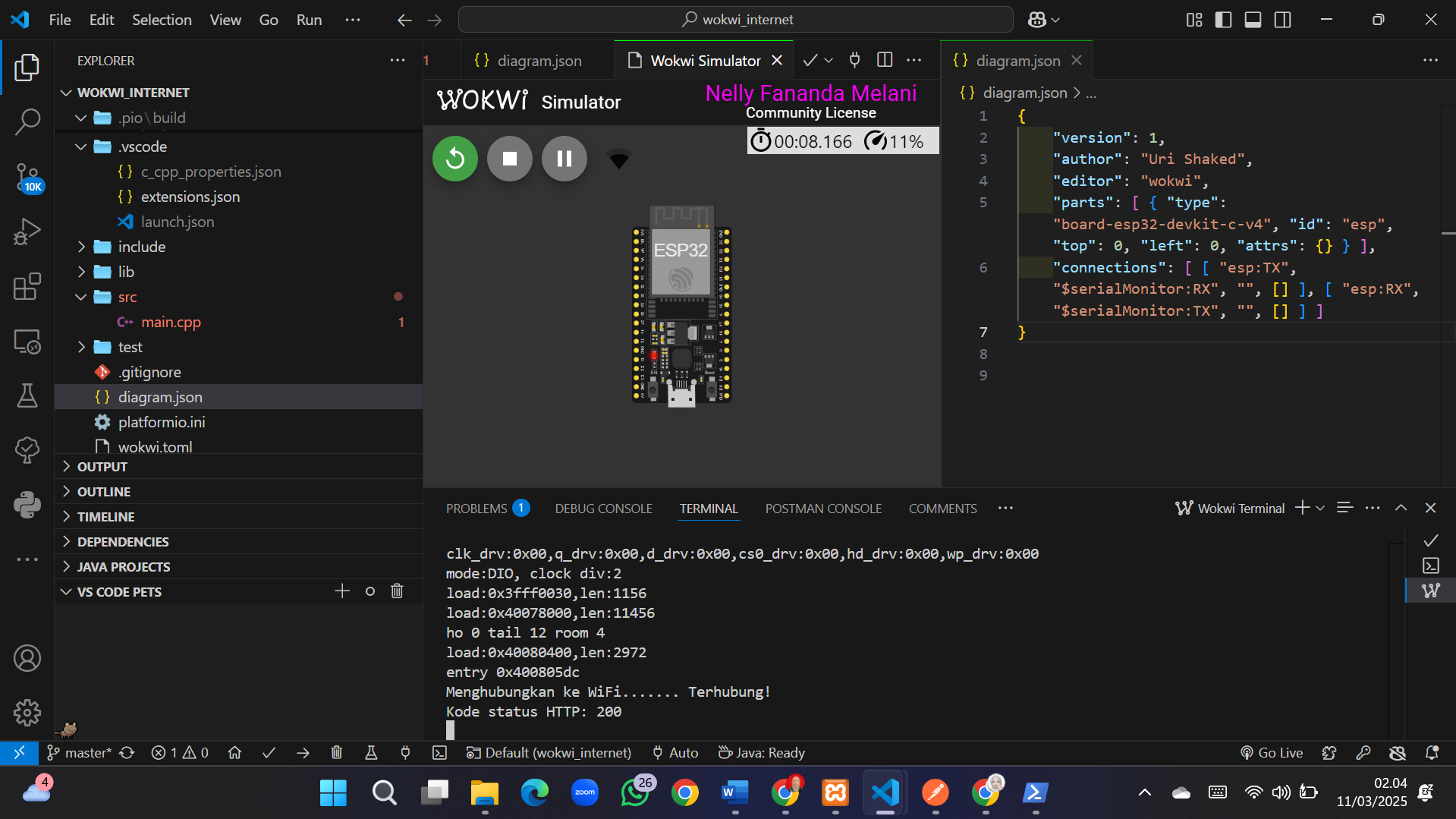
version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'

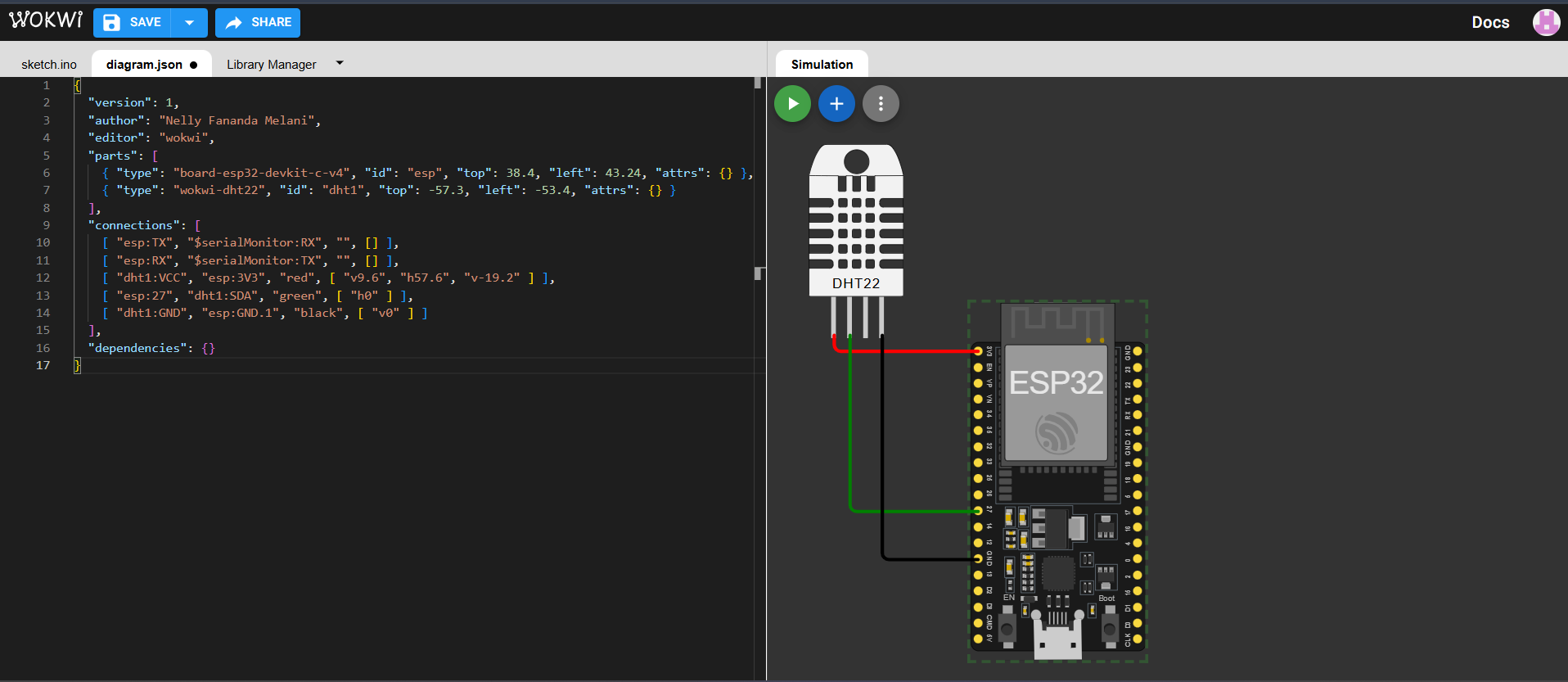


* Tambahkan file **diagram.json**

Simulasi diatas menunjukkan, ESP32 berhasil terhubung ke WIFI Wokwi-GUEST dan berhasil mengakses API laravel yang sudah dibuat pada bab sebelumnya.    
  
**Kode Status HTTP:200**

HTTP status code 200 artinya adalah "OK". Ini berarti bahwa permintaan (request) yang dikirim oleh klien (misalnya browser web atau aplikasi IoT) telah berhasil diproses oleh server. Dengan kata lain, halaman web atau data yang diminta telah berhasil dikirim kembali oleh server dan ditampilkan dengan benar kepada pengguna.

* Berikutnya adalah melakukan modifikasi simulasi dengan menambahkan sensor suhu dan kelembaban. Skenarionya adalah, wokwi simulator akan mengirimkan data suhu dan kelembaban ke API dan menyimpannya ke database mysql seperti yang telah dibuat pada bab sebelumnya.



Rangkai sensor DHT22 dengan ESP32 seperti contoh diatas. Kemudian salin kode **diagram.json** ke file diagram.json yang ada di vscode.

{

  "version": 1,

  "author": "Nelly Fananda Melani",

  "editor": "wokwi",

  "parts": [

    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 38.4, "left": 43.24, "attrs": {} },

    { "type": "wokwi-dht22", "id": "dht1", "top": -57.3, "left": -53.4, "attrs": {} }

  ],

  "connections": [

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

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

    [ "dht1:VCC", "esp:3V3", "red", [ "v9.6", "h57.6", "v-19.2" ] ],

    [ "esp:27", "dht1:SDA", "green", [ "h0" ] ],

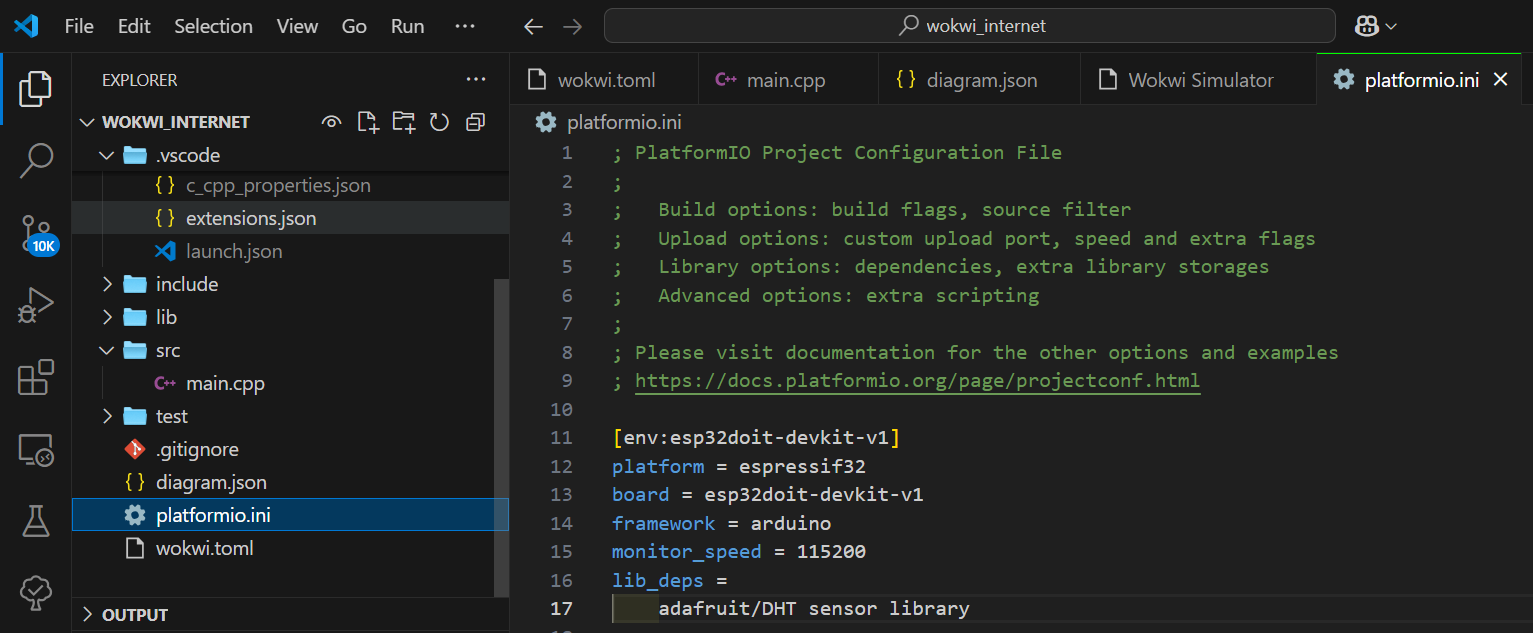
    [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ]

  ],

  "dependencies": {}

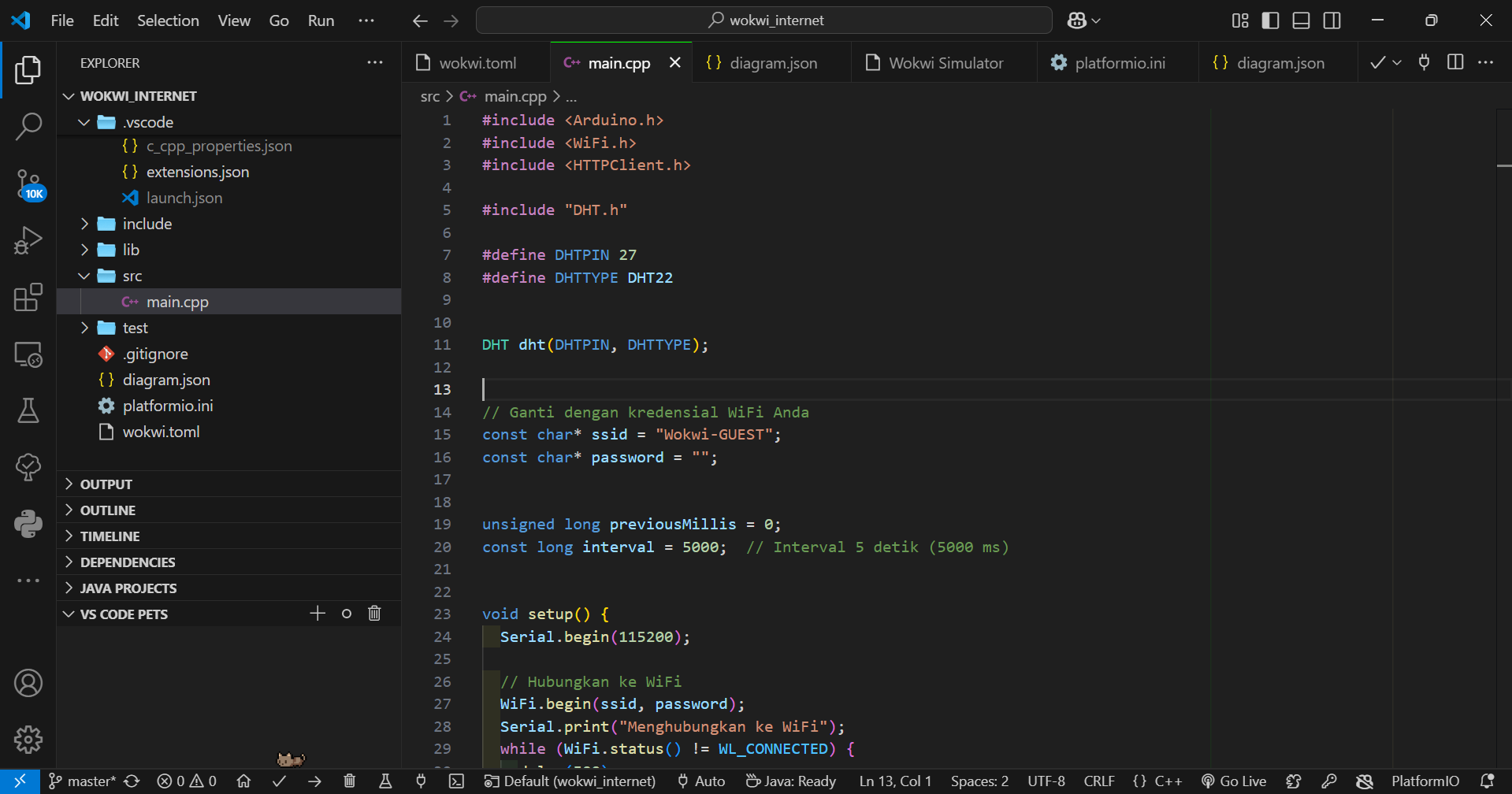
}

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

Pada perubahan diatas, ada tambahan 2 setting yaitu monitor speed dan lib\_deps

Monitor speed digunakan untuk memonitor status pengiriman data dari wokwi simulator ke server api laravel yang telah dibuat. Sedangkan lib\_deps adalah library yang digunakan sensor DHT (sensor suhu dan kelembaban).

* 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://e2c2-2001-448a-50a0-62e1-b528-ebfa-2750-da8b.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();

  }

}

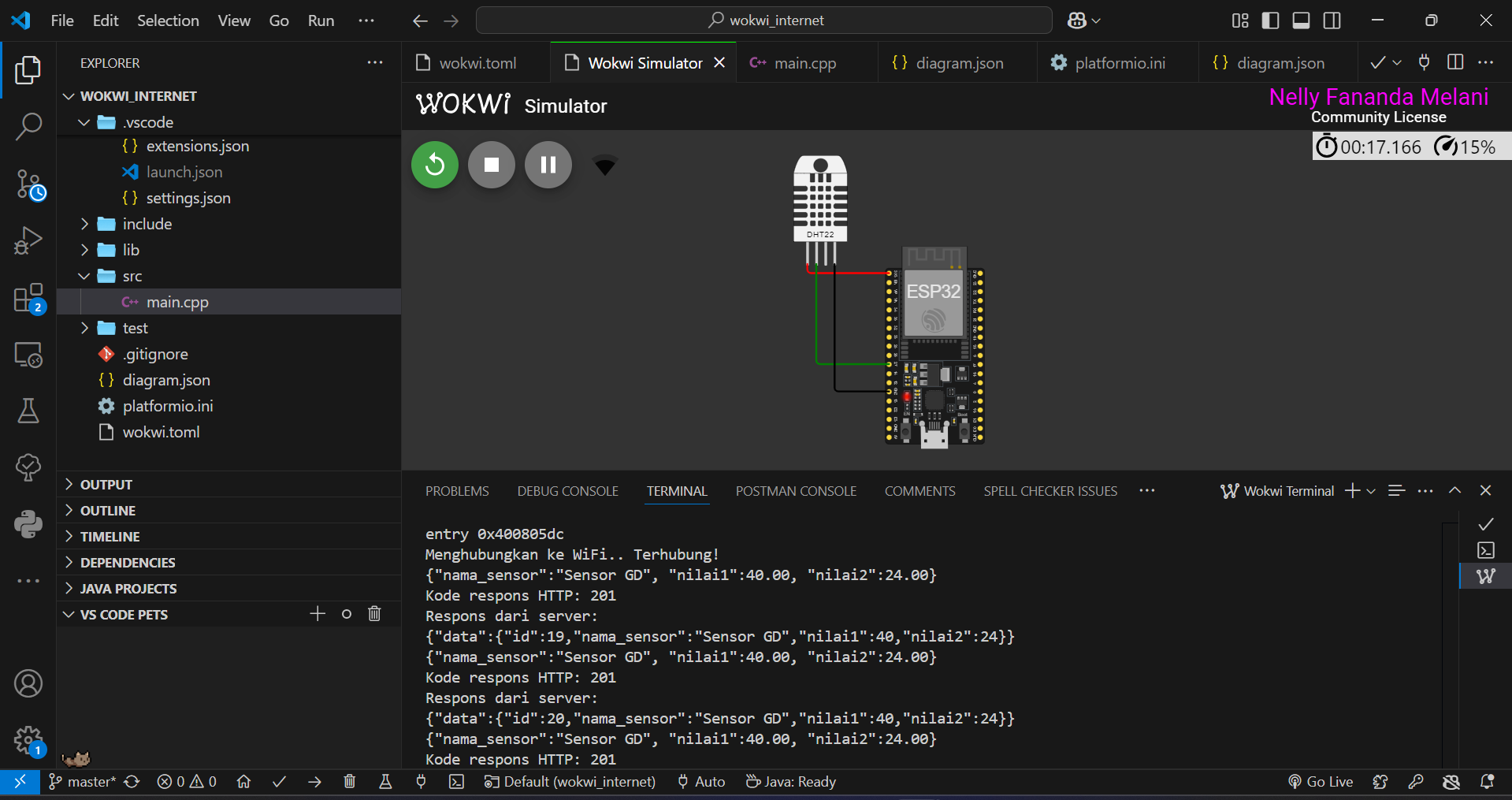
Pada bagian berikut sesuaikan dengan URL NGROK anda

"**http://e2c2-2001-448a-50a0-62e1-b528-ebfa-2750-da8b.ngrok-free.app**/api/posts"; // Ganti dengan URL ngrok yang benar

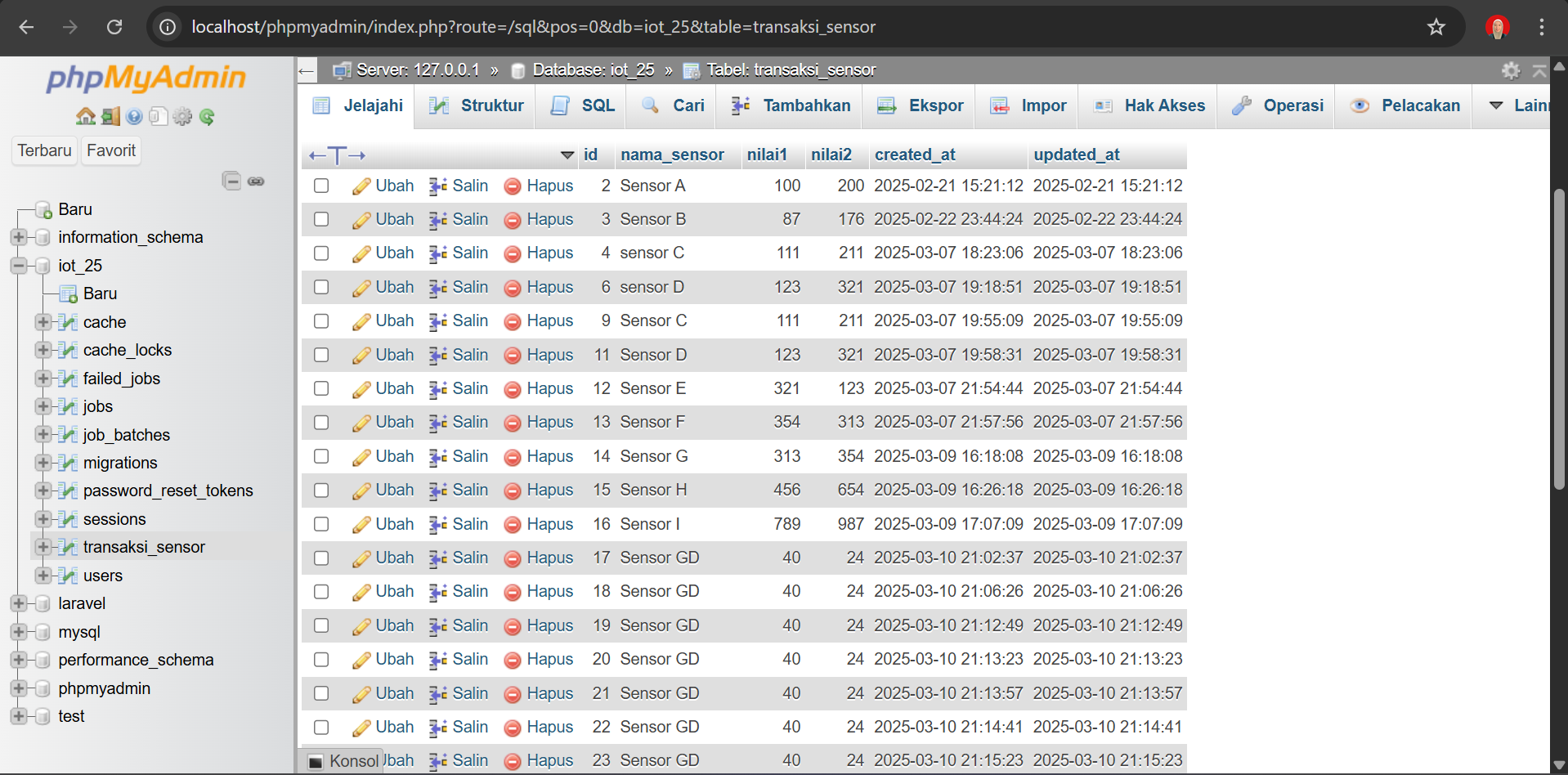
Jalankan simulasi

**> Wokwi Start Simulator**

Berikut adalah tampilan pada simulator dan serial monitor.



* Pastikan di database, data telah muncul dan tersimpan



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

**3.1 Experimental Results (Hasil Eksperimen)**

Pembuatan dan pengujian API menggunakan Laravel 11 serta integrasi dengan Ngrok dan Wokwi Simulator telah berhasil diimplementasikan tanpa kendala signifikan.

Hasil pengujian menunjukkan bahwa:

* Endpoint API berhasil dibuat dan diuji menggunakan Postman, dengan respons sesuai ekspektasi untuk metode GET, POST, PUT, dan DELETE.
* Koneksi database MySQL pada XAMPP berjalan dengan baik, memungkinkan pertukaran data antara API dan database tanpa hambatan.
* Integrasi Ngrok berhasil, memungkinkan API diakses secara publik melalui URL Ngrok yang digunakan dalam Wokwi Simulator.
* Simulasi pada Wokwi menunjukkan bahwa data suhu dan kelembaban berhasil dikirim ke API dan disimpan dalam database.
* Validasi input dan output API berjalan sesuai skenario pengujian, memastikan format data yang dikirim dan diterima sesuai standar JSON.

Berikut adalah rata-rata hasil simulasi:

* Kecepatan respons API: Eksekusi rata-rata cepat dan stabil dalam menangani permintaan.
* Keakuratan data: Data yang dikirim dan diterima sesuai dengan struktur JSON yang diharapkan.
* Stabilitas sistem: API dapat diakses secara berulang tanpa error atau downtime selama pengujian.

**3.2 Discussion (Pembahasan)**

Keberhasilan eksperimen ini menunjukkan bahwa **API berbasis Laravel 11** dapat menangani permintaan HTTP dengan baik serta mengelola data dalam database **MySQL** secara efisien. **Integrasi Ngrok** memungkinkan akses publik tanpa konfigurasi server tambahan, mempermudah proses pengujian, terutama dalam simulasi IoT menggunakan **Wokwi Simulator**.

Faktor yang Mempengaruhi Hasil:

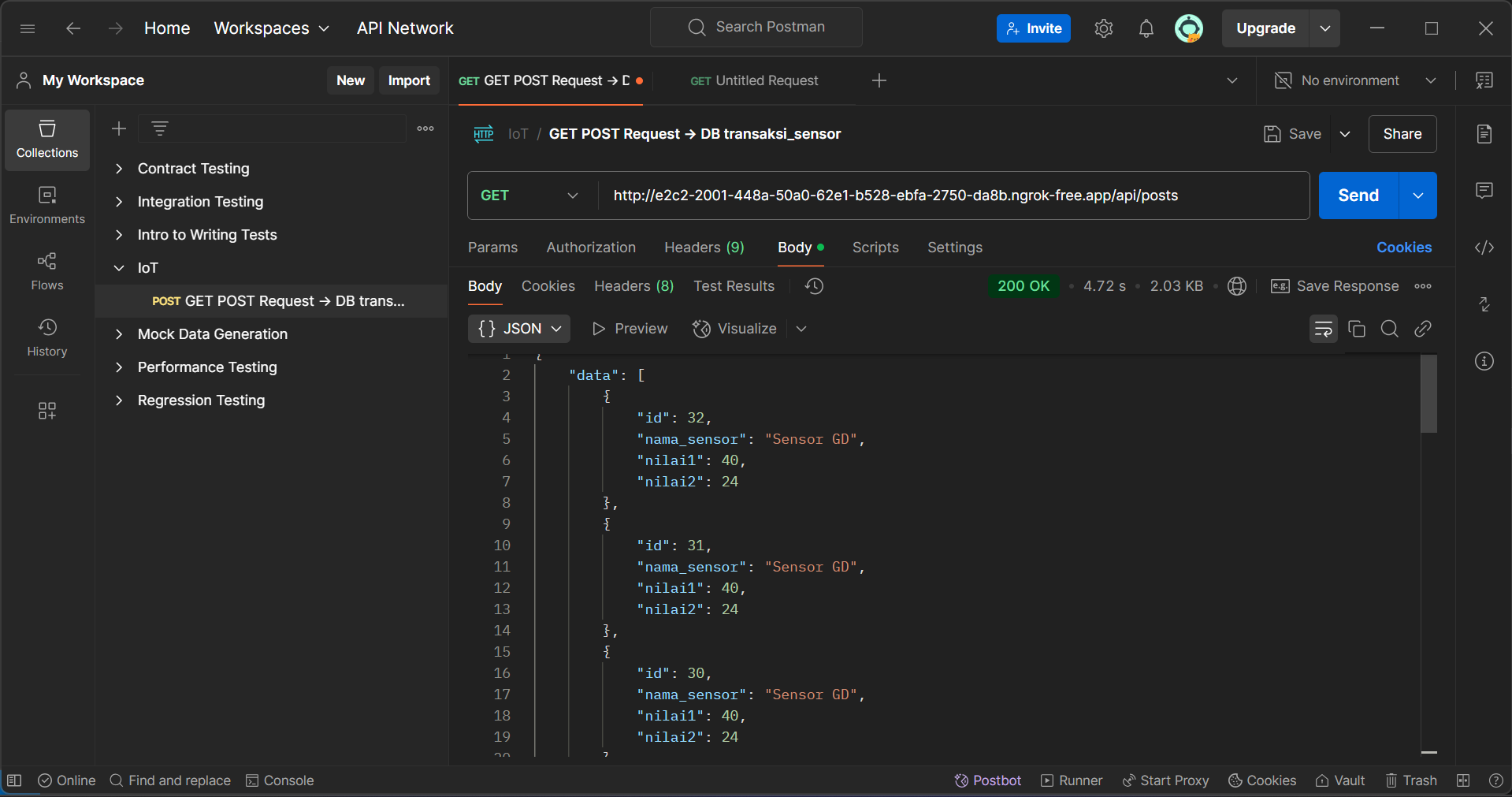
* Kode Program: Kesalahan dalam controller atau route bisa menyebabkan API tidak berfungsi dengan baik.
* Konfigurasi Database: Pengaturan yang salah di file .env bisa membuat API gagal terhubung ke database.
* Keamanan API: Tanpa validasi dan middleware, API rentan terhadap serangan seperti SQL Injection.
* Koneksi Internet: Ngrok butuh koneksi stabil agar API tetap bisa diakses.
* Efisiensi API: Jika tidak dioptimalkan, API bisa lambat saat menangani banyak permintaan.

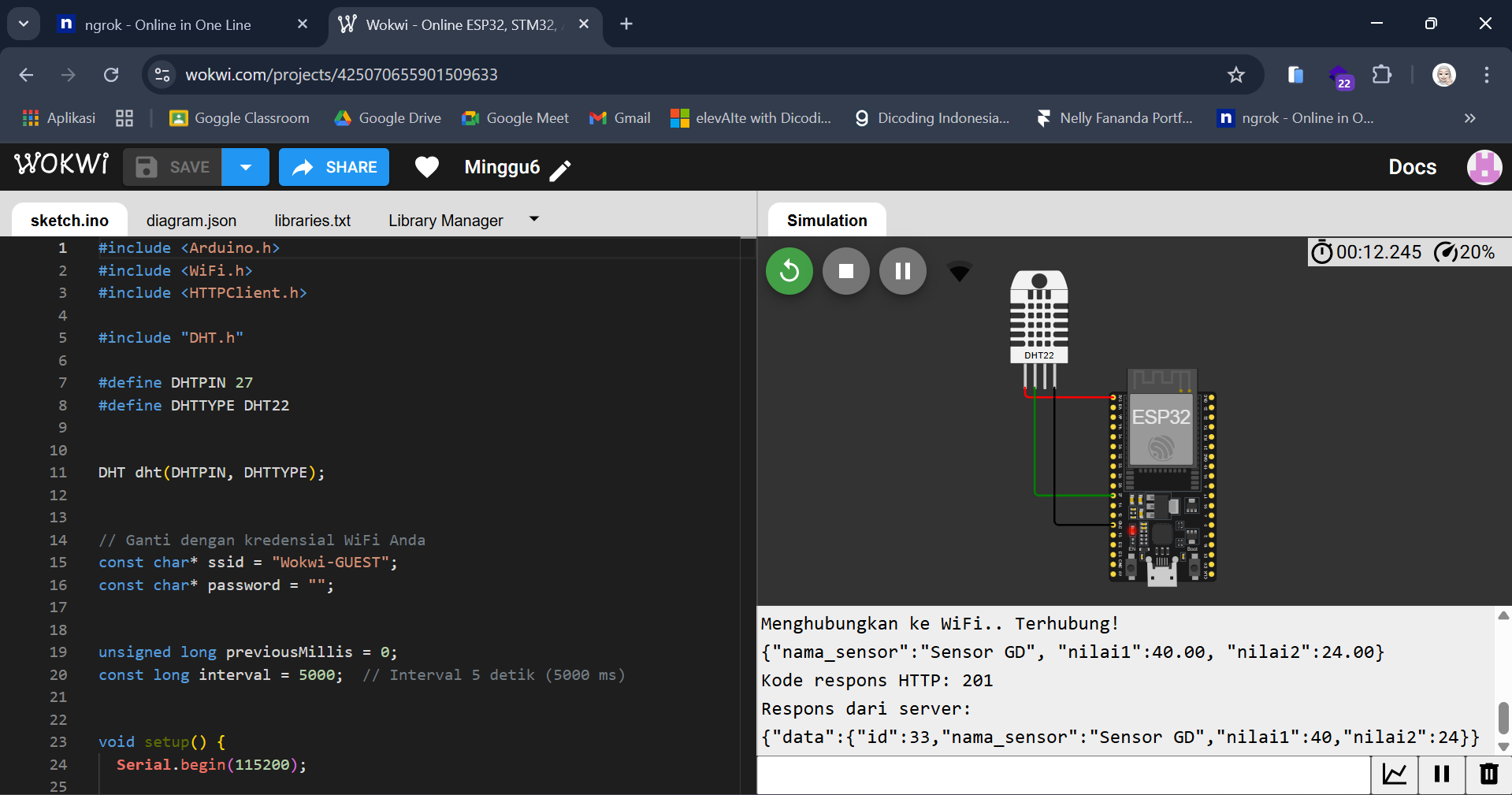
Peningkatan yang Dapat Dilakukan:

* Meningkatkan Keamanan API dengan membatasi akses hanya untuk pengguna yang berwenang.
* Mengoptimalkan Koneksi Database dengan memastikan konfigurasi yang tepat pada file .env.
* Mempercepat Respons API dengan menghindari permintaan data yang berlebihan dalam satu waktu.
* Menerapkan Penyimpanan Sementara (Caching) untuk mengurangi beban pada database.
* Melakukan Pemantauan dan Pencatatan Kesalahan (Logging) untuk memudahkan proses identifikasi dan perbaikan error.

Hasil eksperimen ini dapat menjadi dasar bagi pengembangan sistem berbasis API yang lebih kompleks, seperti integrasi dengan simulator Wokwi, pengolahan data dari sensor suhu dan kelembaban, serta penerapan dalam sistem IoT untuk pemantauan dan pengendalian perangkat secara real-time.

**4. Appendix (Lampiran)**

****

****

