**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)** Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui   
Simulasi WOKWI**



*William jan randall ginting*  
 Fakultas Vokasi, Universitas Brawijaya  
 Email: janrandall.ginting@gmail.com

## **Abstract (Abstrak)**

Bab ini membahas praktik akses Application Programming Interface (API) melalui simulasi mikrokontroler menggunakan platform WOKWI. Dalam simulasi ini, pengguna dapat mempelajari bagaimana mikrokontroler seperti ESP32 terhubung ke internet dan melakukan komunikasi data dengan layanan berbasis web melalui protokol HTTP. Proses pengambilan dan pengiriman data menggunakan API dieksplorasi secara praktis, termasuk pengaturan koneksi Wi-Fi, pemanggilan API dengan metode GET atau POST, serta penanganan respons dari server. Simulasi ini memberikan pemahaman langsung tentang integrasi perangkat IoT dengan layanan cloud, yang menjadi elemen penting dalam pengembangan sistem IoT modern.

**Kata Kunci:**  
API, WOKWI, ESP32, IoT, HTTP, Simulasi Mikrokontroler, Koneksi Wi-Fi, Cloud, GET, POST.

## **1. Introduction (Pendahuluan)**

### **1.1 Latar Belakang**

### **Perkembangan teknologi Internet of Things (IoT) telah mendorong terciptanya berbagai inovasi yang memungkinkan perangkat fisik terhubung dan berkomunikasi melalui jaringan internet. Salah satu komponen penting dalam pengembangan sistem IoT adalah penggunaan Application Programming Interface (API) sebagai jembatan antara perangkat keras dengan layanan digital berbasis cloud. Melalui API, perangkat dapat mengirim atau menerima data secara real-time, sehingga memungkinkan integrasi yang efisien dengan berbagai platform digital.**

### **Namun, dalam tahap pembelajaran dan pengembangan awal, penggunaan perangkat fisik secara langsung seringkali menghadirkan tantangan, seperti keterbatasan alat dan potensi kerusakan perangkat. Oleh karena itu, simulasi menjadi solusi yang efektif untuk memahami konsep dan praktik pengembangan IoT, khususnya dalam hal akses API.**

### **WOKWI hadir sebagai platform simulasi mikrokontroler berbasis web yang memungkinkan pengguna untuk membuat, menjalankan, dan menguji kode program tanpa perlu perangkat keras fisik. Dengan dukungan terhadap mikrokontroler seperti ESP32 dan fitur konektivitas internet, WOKWI menjadi sarana yang ideal untuk melakukan praktik akses API secara virtual.**

### **Melalui bab ini, pembaca akan diajak untuk memahami bagaimana proses akses API dilakukan dalam konteks IoT menggunakan simulasi WOKWI, mulai dari konfigurasi koneksi internet, pengiriman permintaan API, hingga pengolahan data yang diterima. Pembelajaran ini diharapkan dapat memberikan fondasi yang kuat dalam membangun sistem IoT yang terintegrasi dan berbasis data.**

## **1.2 Tujuan**

## **Tujuan dari praktik ini adalah sebagai berikut:**

## **Menjelaskan konsep dasar penggunaan API dalam sistem Internet of Things (IoT).**

## **Memberikan pemahaman praktis tentang cara mengakses API menggunakan mikrokontroler melalui simulasi di platform WOKWI.**

## **Melatih kemampuan dalam menghubungkan mikrokontroler (seperti ESP32) ke jaringan internet menggunakan kode program.**

## **Mendemonstrasikan proses pengambilan dan pengiriman data ke layanan web melalui metode HTTP (GET dan POST).**

## **Menunjukkan bagaimana data yang diperoleh dari API dapat diolah dan ditampilkan melalui simulasi perangkat.**

## **2. Methodology (Metodologi)**

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

**Tools:**

1. **WOKWI Simulator** – Platform simulasi mikrokontroler berbasis web yang mendukung ESP32 dan koneksi internet.
2. **Visual Studio Code (VS Code)** – Digunakan untuk menulis dan mengelola kode program secara efisien sebelum disalin ke WOKWI.
3. **XAMPP** – Aplikasi lokal yang digunakan untuk membuat server web dan API endpoint berbasis PHP/MySQL secara lokal.
4. **Web Browser** – Untuk mengakses WOKWI, testing API, dan melihat output dari server lokal.
5. **Koneksi Wi-Fi** – Diperlukan untuk menjalankan simulasi online dan menghubungkan ESP32 ke internet secara virtual melalui WOKWI.
6. **Postman (opsional)** – Untuk menguji API lokal (dari XAMPP) sebelum dihubungkan dengan ESP32.

**Material (Simulasi Virtual):**

1. **ESP32 Board** – Mikrokontroler yang digunakan dalam simulasi sebagai perangkat utama untuk mengakses API.
2. **Sensor Virtual (opsional)** – Misalnya sensor suhu atau kelembapan, tergantung kebutuhan data yang ingin dikirim.
3. **API Endpoint** – Bisa berupa API publik (seperti OpenWeatherMap) atau API lokal yang dibuat melalui XAMPP.
4. **Library Arduino** – WiFi.h dan HTTPClient.h, digunakan dalam kode ESP32 untuk koneksi internet dan komunikasi HTTP.

### **2.2 Implementation Steps (Langkah Implementasi)**

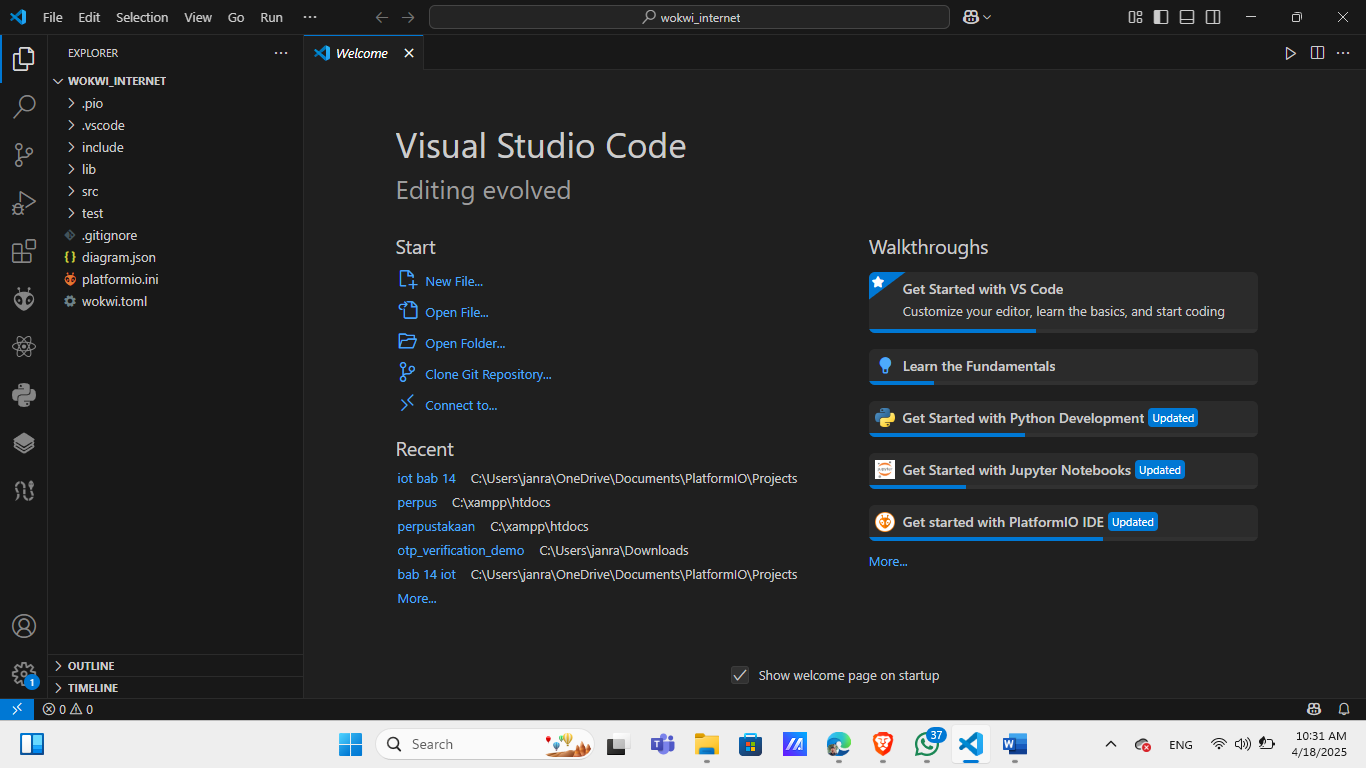
1. Pastikan Hardware ESP32 dikenali oleh komputer

**Jalankan API laravel dengan perintah**

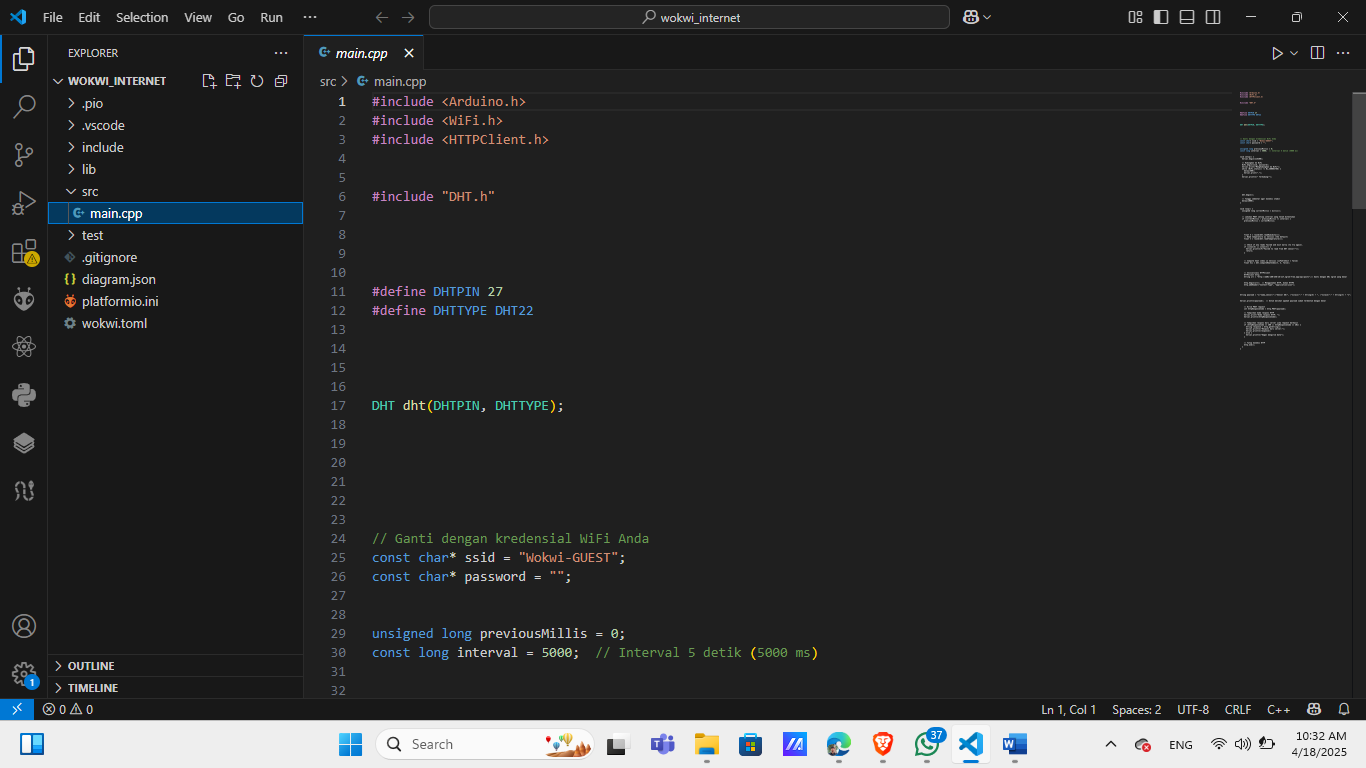
**php artisan serve --host=0.0.0.0 --port=8080**

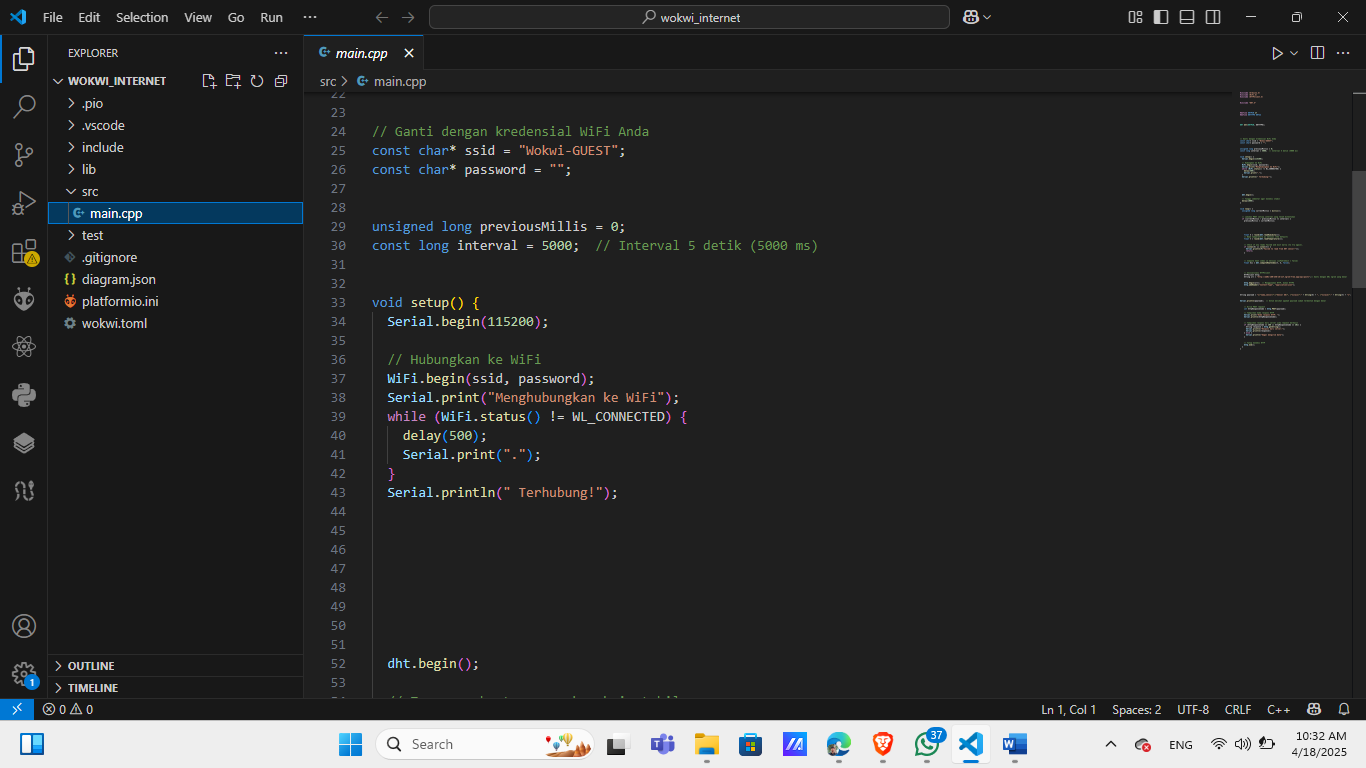
**Perintah diatas memastikan API laravel dapat diakses dari IP Address manapun dan memastikan bekerja pada port 8080.**

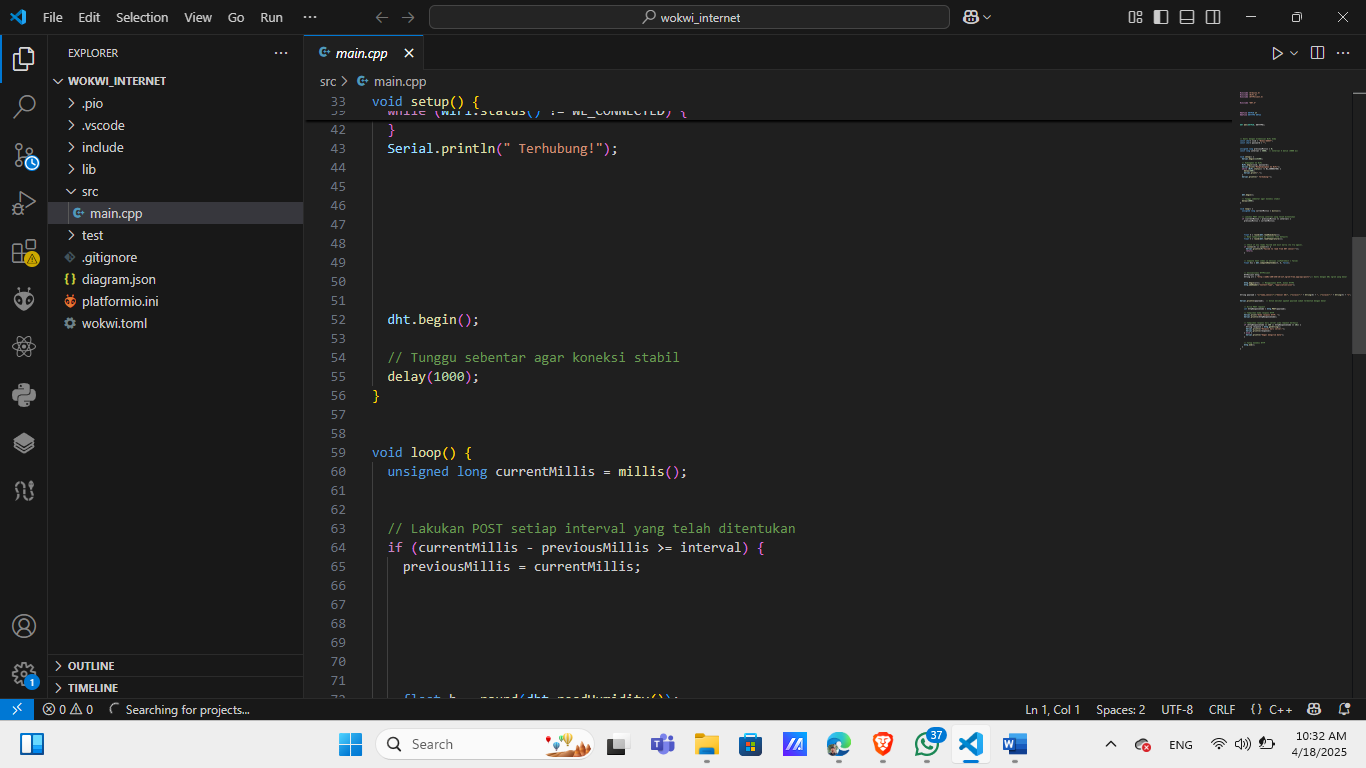
**Buat file baru wokwi simulator di platform.io**

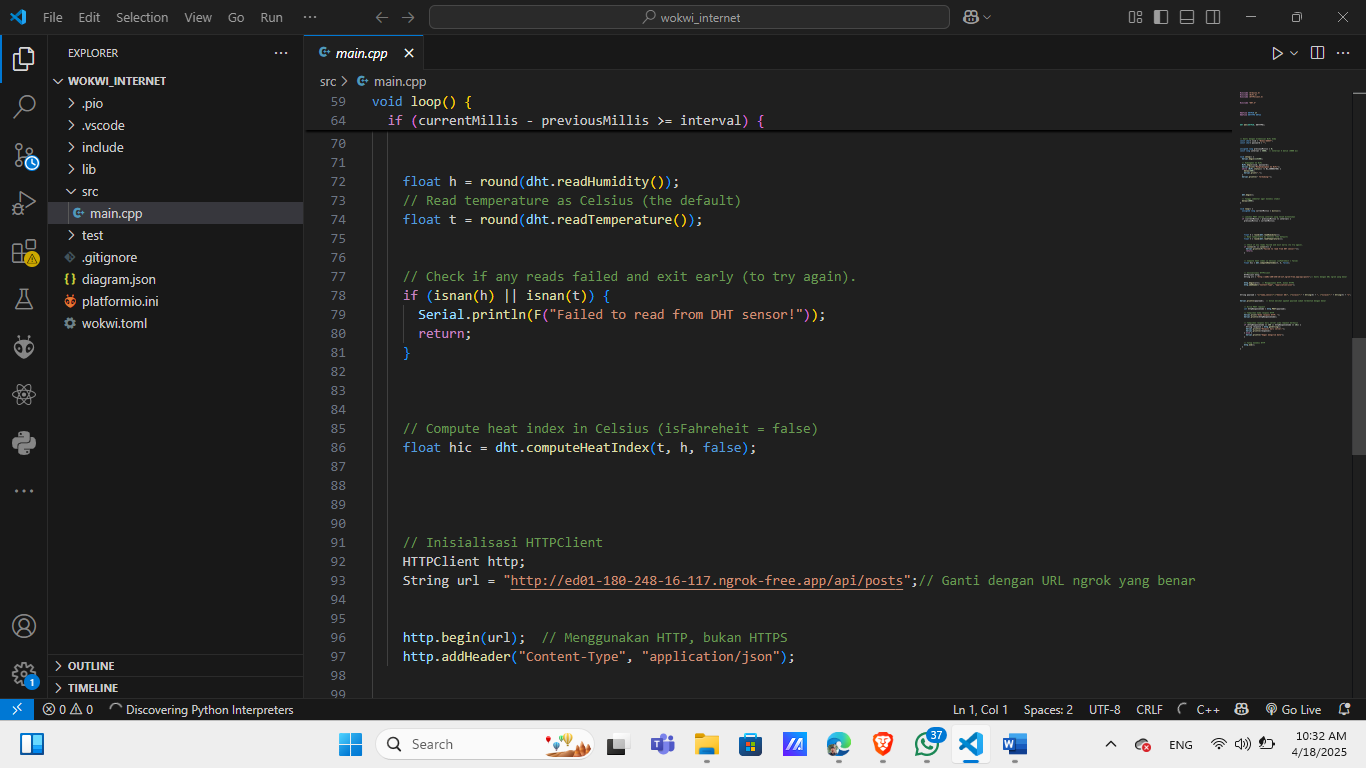


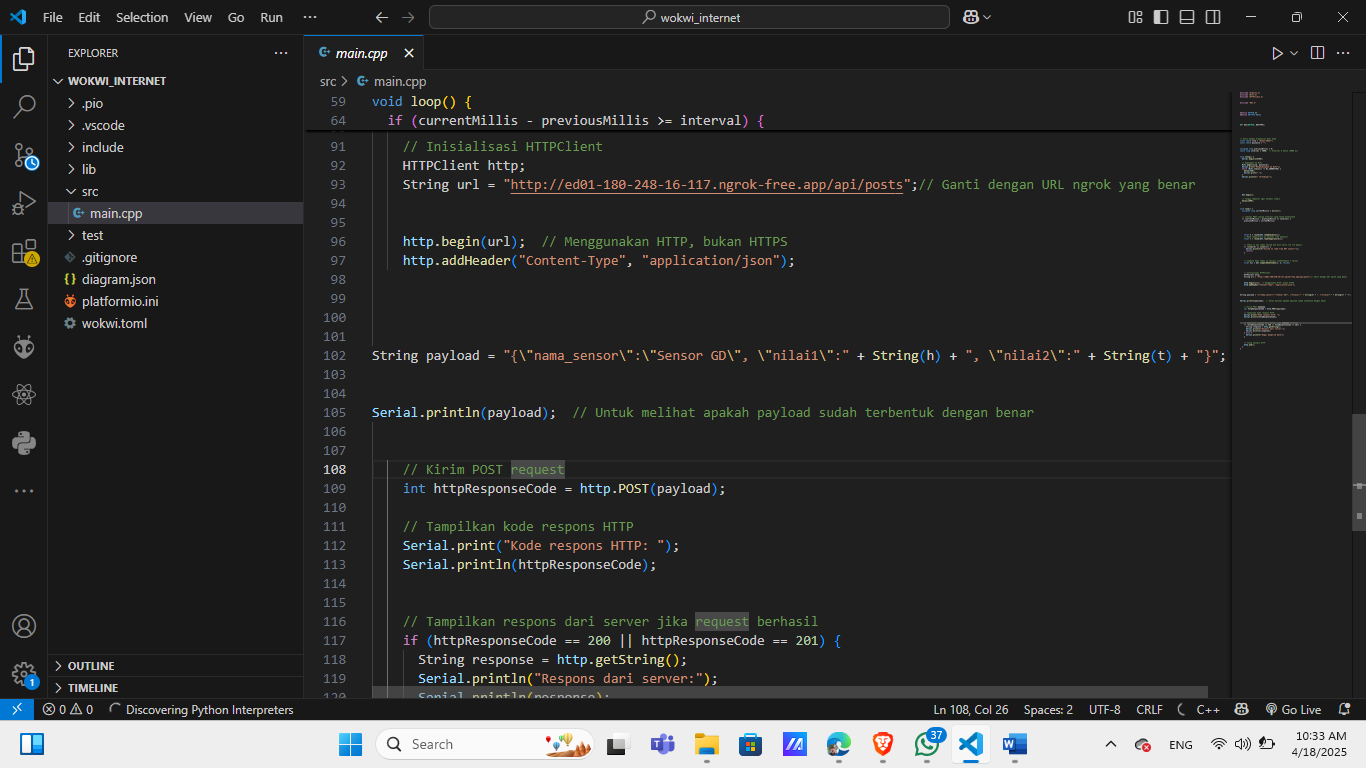
**Ubah main.cpp**

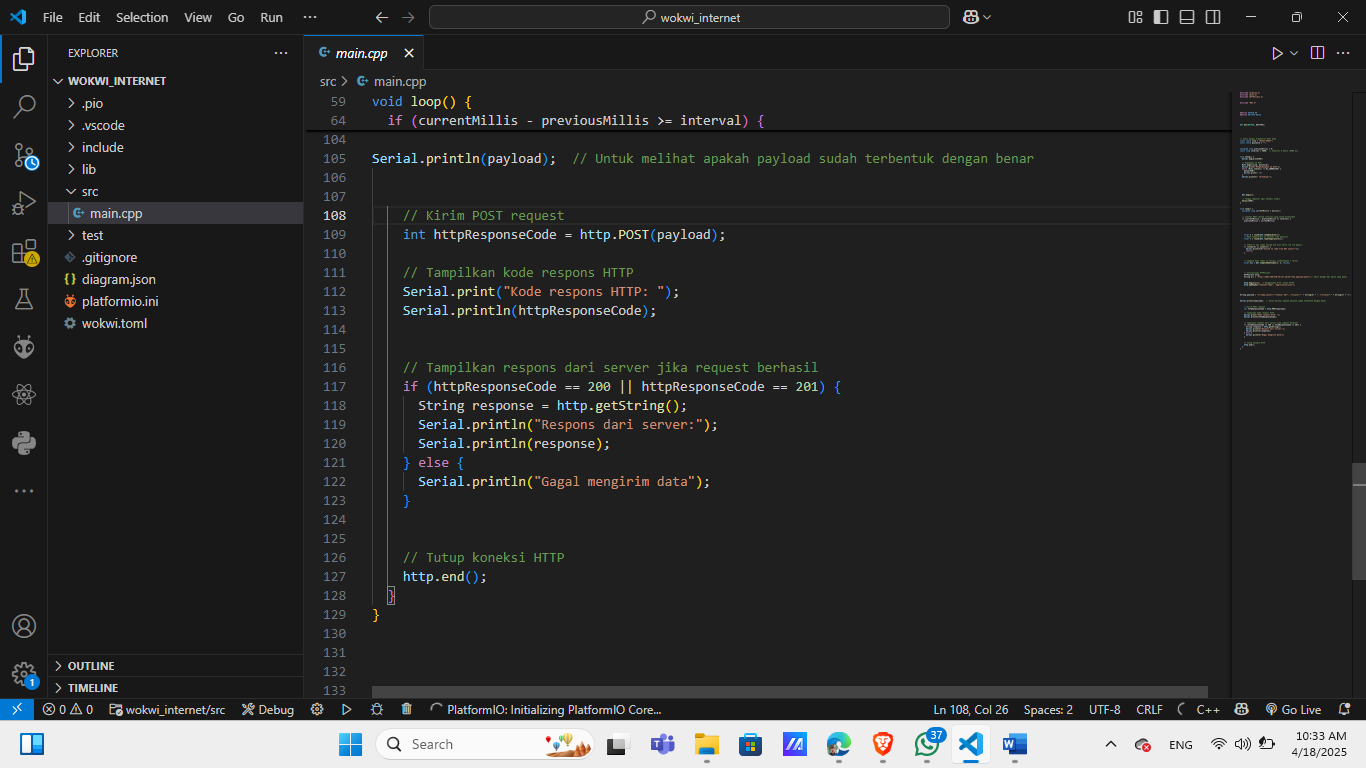












**Perhatikan pada bagian**

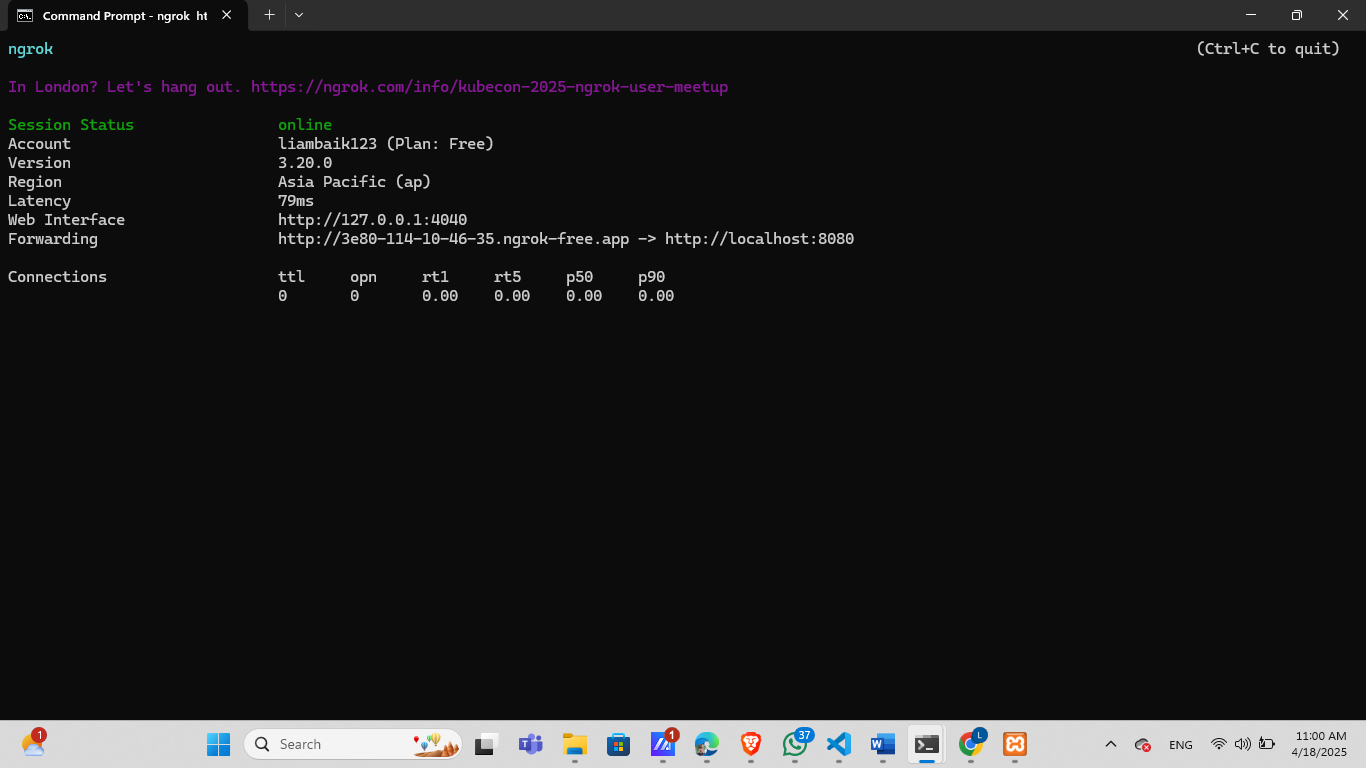
**// URL lengkap server yang akan diakses**

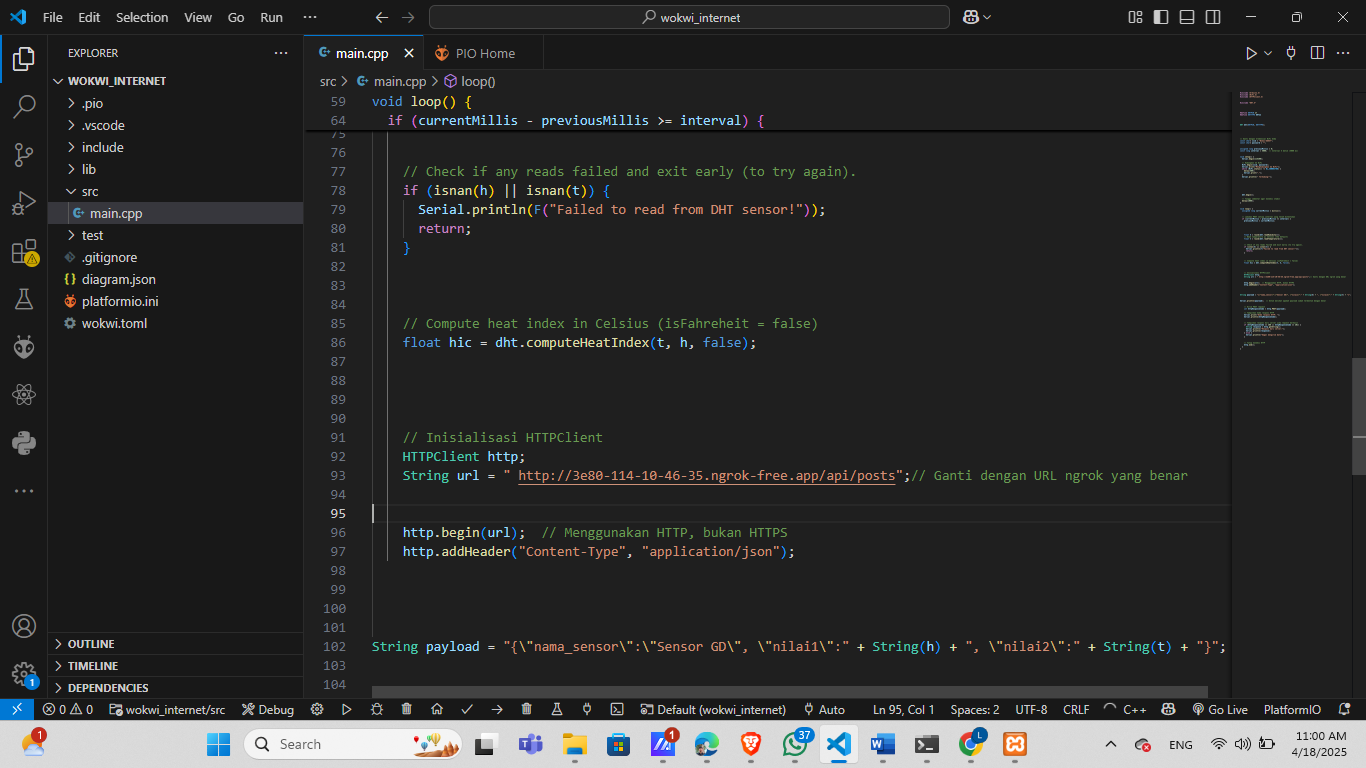
**const char\* serverUrl = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts";**

**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.**

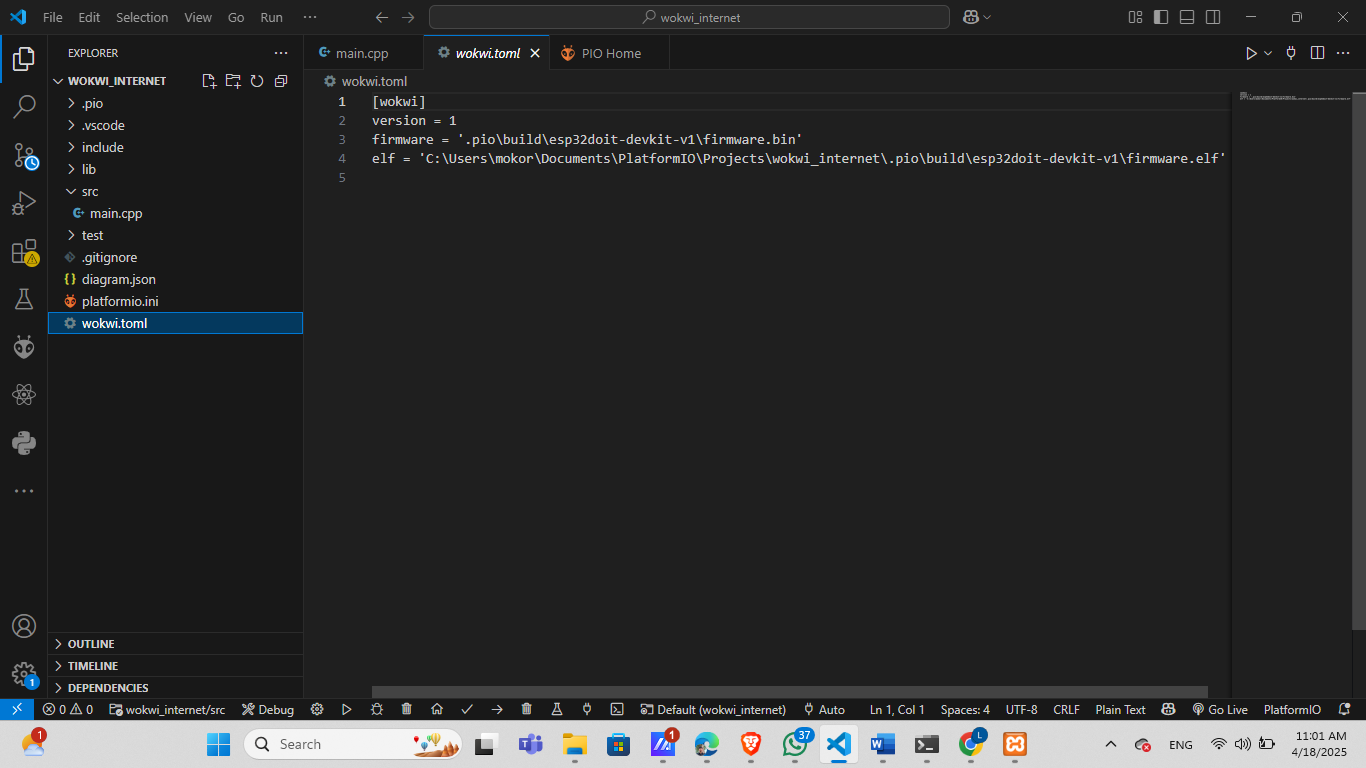
**ngrok http --scheme=http 8080**

**Sesuaikan alamat port 8080 sesuai dengan port berjalannya aplikasi Laravel Anda.**

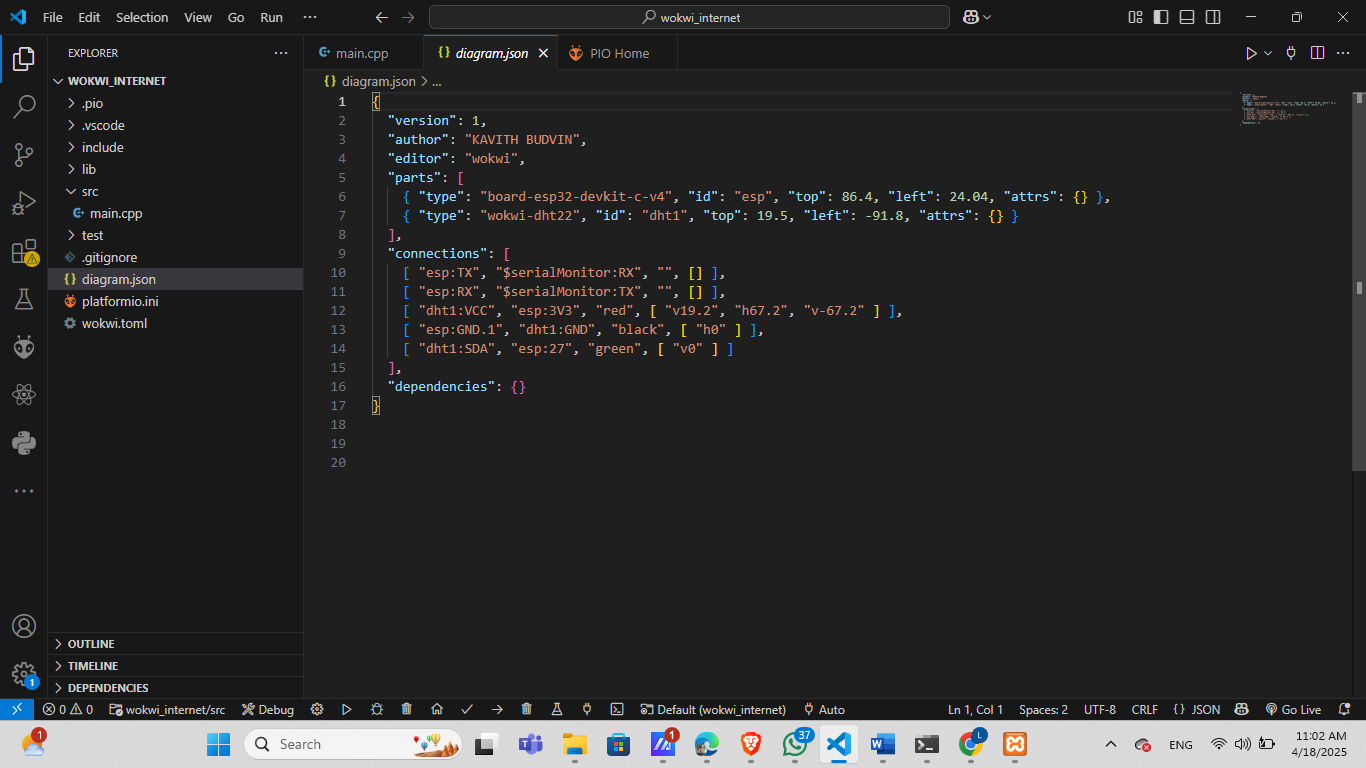




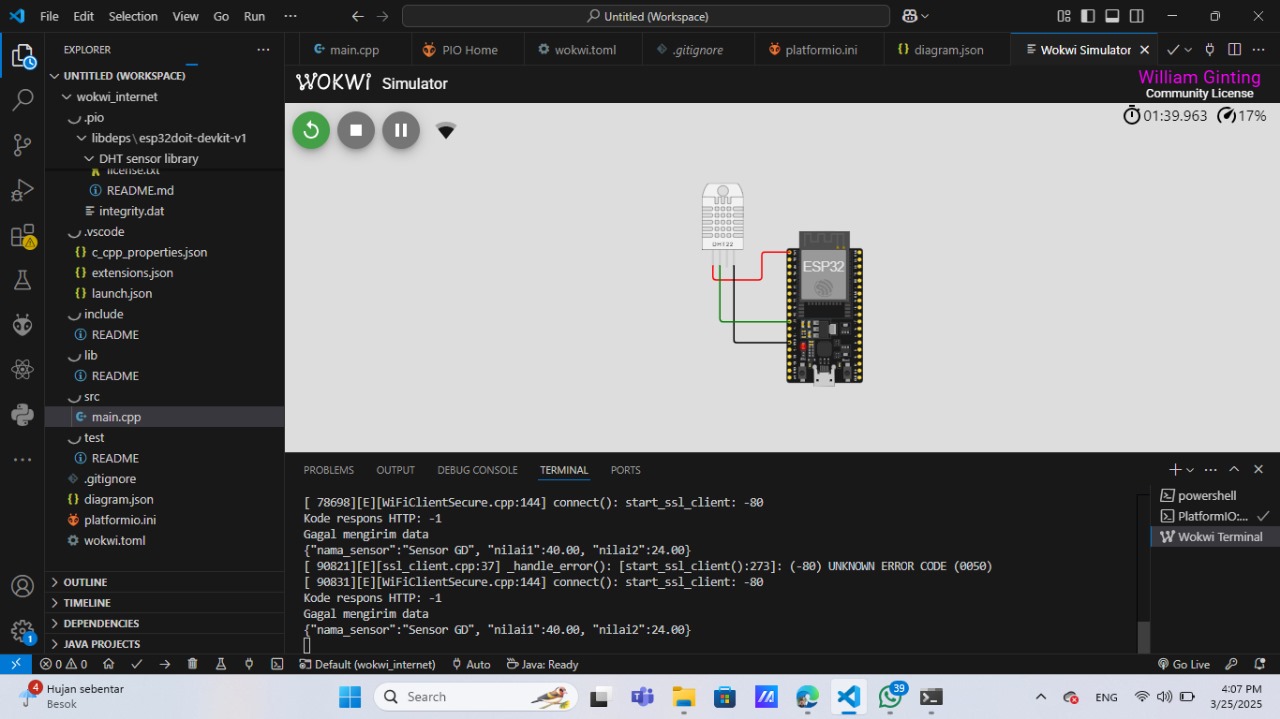
Tambahkan file **wokwi.toml**



Tambahkan file **diagram.json**



**Wokwi Start Simulator**



Jka berhasil maka bsa mengirim data seperti ini

