LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**

****

*Abdur Rohman Syah Sjadja’ah*

Fakultas Vokasi, Universitas Brawijaya

Email: [abdurrohmansyh02@gmail.com](mailto:abdurrohmansyh02@gmail.com)

**Abstract** (Abstrak)

Praktikum ini bertujuan untuk memberikan pemahaman kepada mahasiswa tentang cara mengakses API melalui simulasi perangkat IoT menggunakan WOKWI. Dalam praktikum ini, mahasiswa akan mempelajari proses pengembangan API RESTful dengan Laravel 11, termasuk penerapan autentikasi berbasis token menggunakan Laravel Sanctum untuk keamanan akses. Simulasi dilakukan menggunakan WOKWI sebagai platform IoT, yang memungkinkan mahasiswa untuk mengintegrasikan API dengan perangkat virtual. Pengujian dilakukan melalui Postman untuk memastikan konektivitas dan fungsionalitas API, sementara XAMPP digunakan sebagai server database. Praktikum ini juga mencakup penggunaan Visual Studio Code (VSCode) sebagai lingkungan pengembangan dan Command Prompt (CMD) untuk eksekusi perintah. Dengan pendekatan yang sistematis, praktikum ini diharapkan dapat meningkatkan kemampuan mahasiswa dalam membangun aplikasi berbasis web yang aman, scalable, dan terintegrasi dengan perangkat IoT melalui simulasi WOKWI.

***Kata kunci*:** API, WOKWI, Laravel 11, VSCode, Postman.

**1.Introduction** (Pendahuluan)

**1.1 Latar Belakang**

Pada saat ini teknologi informasi semakin berkembang, integrasi antara aplikasi berbasis web dan perangkat IoT menjadi semakin penting. Salah satu cara untuk mewujudkan integrasi ini adalah melalui penggunaan Application Programming Interface (API). API memungkinkan komunikasi antara aplikasi dan perangkat IoT secara efisien serta mendukung pengelolaan data secara real-time. Simulasi WOKWI menawarkan solusi praktis untuk memahami konsep ini tanpa memerlukan perangkat keras fisik, sehingga sangat cocok untuk pembelajaran.

Laravel 11 merupakan framework PHP yang populer dan memiliki berbagai fitur unggulan untuk pengembangan API RESTful. Dengan Laravel Sanctum, pengembang dapat menerapkan autentikasi berbasis token untuk memastikan keamanan akses API. Praktikum ini dirancang untuk membantu mahasiswa memahami langkah-langkah teknis dalam membangun API menggunakan Laravel 11 dan mengintegrasikannya dengan simulasi perangkat IoT melalui WOKWI.

Pengujian API dilakukan menggunakan Postman sebagai alat bantu utama untuk mengirimkan permintaan HTTP ke server lokal yang dijalankan melalui XAMPP. Selain itu, Visual Studio Code (VSCode) digunakan sebagai lingkungan pengembangan utama untuk menulis kode program Laravel. Praktikum ini memberikan pengalaman langsung kepada mahasiswa dalam mengembangkan aplikasi modern yang terintegrasi dengan teknologi IoT.

**1.2 Tujuan Eksperimen**

Praktikum ini bertujuan untuk:

* Memahami langkah-langkah pembuatan API RESTful menggunakan Laravel 11.
* Mengintegrasikan API dengan simulasi perangkat IoT melalui platform Wokwi.
* Menerapkan auntetikasi berbasis token menggunakan Laravel sanctum untuk keamanan akses.
* Melakukan pengujian konektivitas dan fungsionalitas API menggunakan Postman.
* Menggunakan XAMPP sebagai server database dan VSCode sebagai lingkungan pengembangan.
* Memberikan pengalaman praktis dalam membangun aplikasi web yang aman dan scalable serta terintegrasi dengan IoT melalui simulasi di wokwi.

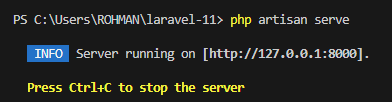
1. **Methodology** (Metodologi)
   1. **Tools & Materials** (Alat & Bahan)

* Laptop
* Internet
* Web Browser (Chrome)
* Postman
* Visual Studio Code
* CMD
* XAMPP
* Wokwi.com
* Ngrok
  1. **Implementation Steps** (Langkah Implementasi)

Buka XAMPP nyalakan Apache dan MySQL

Buka terminal VSCode dan jalankan perintah berikut:

**php artisan serve**

****

**Praktik Akses API Melalui Simulasi WOKWI**

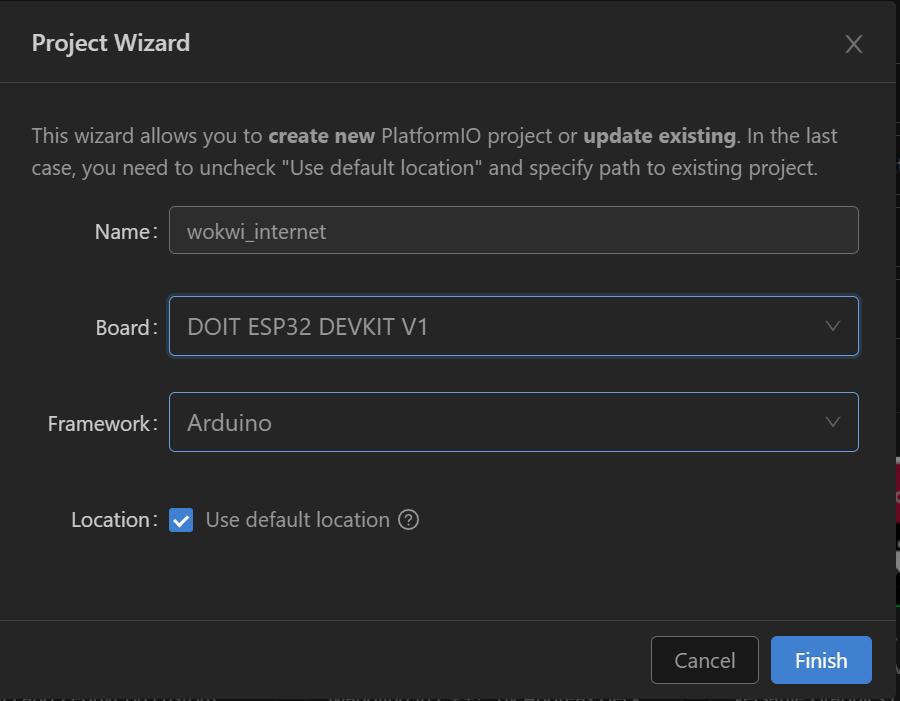
Jalankan API Laravel dengan perintah

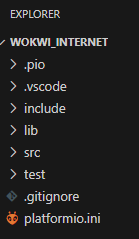
**php artisan serve --host=0.0.0.0 --port=8080**

****

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**

#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.");

    }

  }

}

**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 tunnelling menggunakan Ngrok:

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

**Jalankan PowerShell sebagai Administrator**

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

****

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



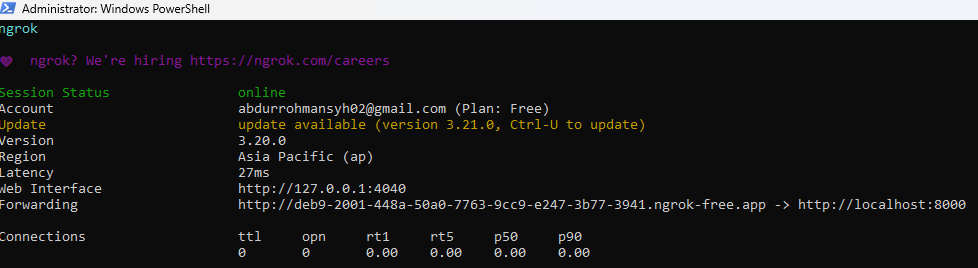
**Tambahkan Authtoken ke konfigurasi Ngrok di PoweShell**

ngrok config add-authtoken 2uFZe3roPhPU6myTHt0C5B8u1LF\_5GWXb4ZXy22y6P3DsmSvS



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

.\ngrok http --scheme=http 8000



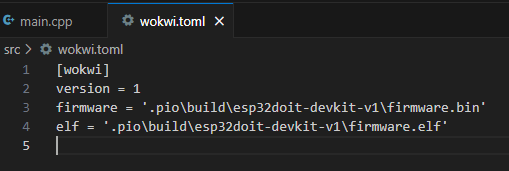
Tambahkan file **wokwi.toml**

[wokwi]

version = 1

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

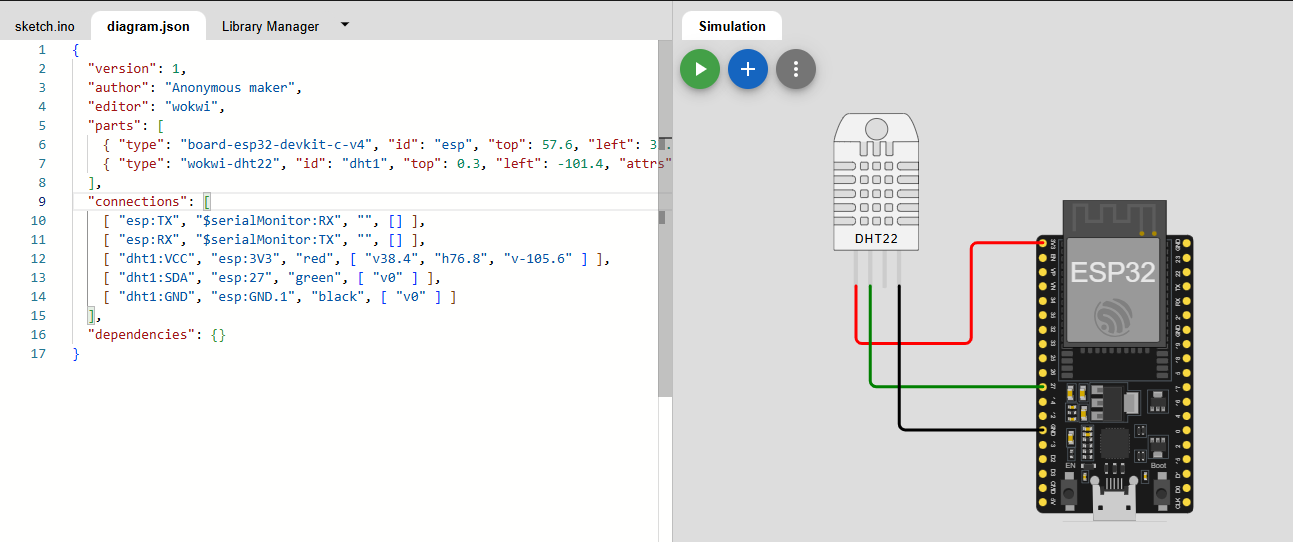
elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'



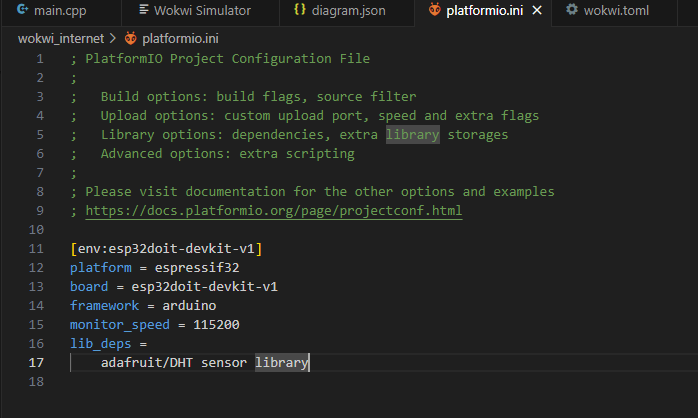
Tambahkan file diagram.json



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



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://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();

  }

}

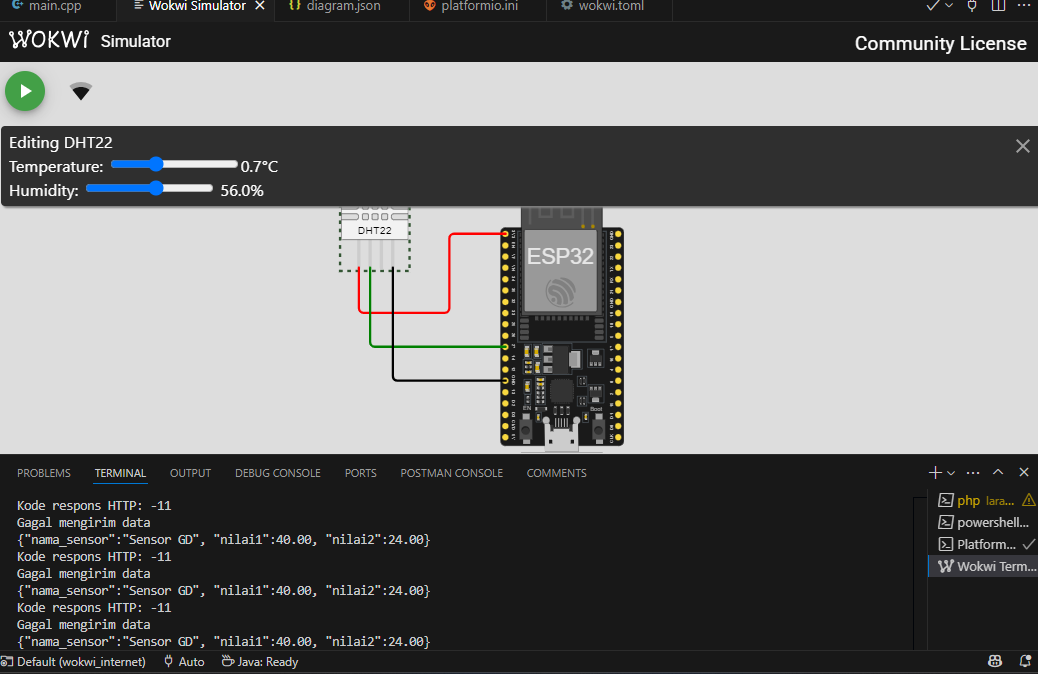
Pada bagian berikut sesuaikan dengan URL NGROK anda

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

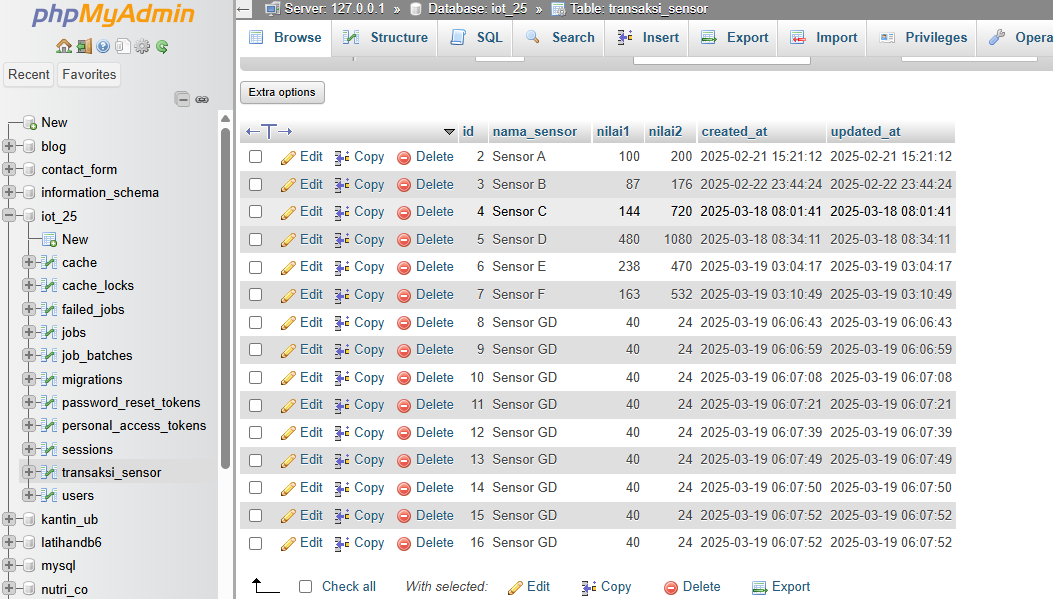
Jalankan simulasi

**> Wokwi Start Simulator**

Berikut adalah tampilan pada simulator dan serial monitor.



Pastikan di database, data telah muncul dan tersimpan



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

**3.1 Experimental Results (Hasil Eksperimen)**  
Praktikum ini berhasil mengimplementasikan akses API melalui simulasi perangkat IoT menggunakan WOKWI dan Laravel 11. Dalam eksperimen ini, kami memfokuskan pada pengembangan API RESTful yang dapat berinteraksi dengan data sensor suhu dan kelembaban yang disimulasikan. Berikut adalah hasil dari setiap tahap praktikum:

* **Pengaturan Lingkungan Pengembangan**: Mahasiswa berhasil mengkonfigurasi lingkungan pengembangan menggunakan Visual Studio Code (VSCode) dan XAMPP sebagai server database. Instalasi Laravel 11 berjalan lancar, dan mahasiswa dapat menjalankan server lokal untuk menguji API.
* **Pembuatan API RESTful**: Setelah pengaturan lingkungan, mahasiswa membuat endpoint API yang mendukung operasi CRUD untuk data suhu dan kelembaban. Endpoint ini dirancang untuk menerima permintaan dari WOKWI dan mengirimkan data yang sesuai. Pengujian awal menunjukkan bahwa endpoint berfungsi dengan baik, mampu menerima dan mengolah data yang dikirimkan.
* **Integrasi dengan WOKWI**: Simulasi perangkat IoT dilakukan di platform WOKWI, menggunakan komponen seperti ESP32 dan sensor DHT11 untuk membaca suhu dan kelembaban. Mahasiswa berhasil menghubungkan mikrokontroler dengan API yang telah dibuat, sehingga data dari sensor dapat dikirim ke server Laravel secara real-time.
* **Pengujian Menggunakan Postman**: Pengujian API dilakukan menggunakan Postman untuk memastikan bahwa endpoint dapat diakses dengan benar dan menghasilkan respons yang diharapkan. Hasil pengujian menunjukkan bahwa semua endpoint berfungsi sesuai dengan spesifikasi, termasuk autentikasi berbasis token menggunakan Laravel Sanctum.
* **Uji Coba Simulasi**: Selama simulasi di WOKWI, mahasiswa dapat melihat data suhu dan kelembaban yang diperoleh dari sensor ditampilkan secara akurat. Simulasi ini juga memungkinkan mahasiswa untuk melakukan debugging jika terjadi kesalahan dalam kode atau konfigurasi.

**3.2 Kesimpulan**

Secara keseluruhan, hasil eksperimen menunjukkan bahwa integrasi antara Laravel 11 dan simulasi WOKWI berjalan dengan baik, memberikan pengalaman praktis dalam membangun sistem pengukuran suhu dan kelembaban berbasis IoT. Dengan pendekatan sistematis ini, mahasiswa tidak hanya memahami aspek teknis pengembangan API tetapi juga pentingnya integrasi antara perangkat lunak dan perangkat keras dalam konteks IoT.

Hasil ini mendukung temuan sebelumnya bahwa penggunaan simulator seperti WOKWI sangat efektif dalam pembelajaran IoT, karena memungkinkan mahasiswa untuk belajar tanpa memerlukan perangkat keras fisik yang mahal atau kompleks

**Appendix** (Lampiran)

