TUGAS BAB 13

**Integrasi API Laravel 11 ke dalam WOKWI**

****

Nama : Muhammmad Farhan Baihaqy

Kelas : T4C

NIM : 233140700111065

***Muhammad Farhan Baihaqi***

***Fakultas Vokasi, Universitas Brawijaya Email:***[***mfarhanbaihaqy@gmail.com***](mailto:kadavi2945@student.ub.ac.id)

# Abstrak

# Pembahasan pada laporan ini difokuskan pada uji kemampuan ESP32 virtual digunakan untuk mengakses antarmuka API melalui platform simulasi IOT WOKWI. Yang mencakup tahap konektivitas Wi-Fi, pengambilan data dari server Laravel yang di-expose via Ngrok, serta pengembangan sistem dengan integrasi sensor DHT22 virtual untuk monitoring suhu dan kelembaban. Berdasarkan uji yang telah dilakukan, dapat disimpulkan bahwa ESP32 virtualnya mampu berinteraksi dengan API secara real-time termasuk data hasil pembacaan sensor dapat ditransmisikan ke basis data MySQL melalui protokol HTTP POST. Teknologi IoT dan API dikombinasikan secara efektif dalam pengelolaan data terdistribusi.

# Pendahuluan

laporan ini membahas bagaimana pembuatan API menggunakan Laravel 11 dan pengintegrasian dengan Ngrok untuk akses eksternal. Berdasarkan rancangan pada bagian sebelumnya, API bertugas melakukan manajemen permintaan data melalui aplikasi dari pengguna fenomenologis yang disajikan dengan respons terstruktur. Perincian fitur lanjutan dari Laravel dan implementasinya, yaitu routing, middleware, dan Eloquent ORM. Ngrok fungsi utamanya adalah untuk memberikan akses global pada server lokal; membuka kemungkinan akses api untuk klien luar langganan sehingga ia dapat melakukan pengujian secara real-time tanpa biaya deploy.

# Latar Belakang

pada konteks pengembangan IoT modern, sinergi antara perangkat embedded seperti ESP32 dan API berbasis Laravel telah menjadi kunci proses pembentukan sistem otomatisasi dan manajemen data berdasarkan cloud. Melalui WOKWI, simulasi memungkinkan pengujian terhadap interaksi ESP32 yang bertugas sebagai perangkat IoT, dengan API berbasis laravel untuk selalu berjalan secara real-time sehingga meminimalisir kemungkinan error sekaligus memastikan proses validasi alur data, autentikasi, serta kapabilitas tersebut sebelum direalisasikan fisik. Lebih lanjut, proses dengan WOKWI memungkinkan hardware berintegrasi dengan layanan cloud teroptimalisasi selain menggunakan protokol REST/HTTP bekerja dengan ideal.

# Tujuan

# Konfigurasi koneksi Wi-Fi pada ESP32 tahap awal IoT.

# Ngrok sebagai jembatan akses API Laravel lokal.

# Kirim data sensor via HTTP POST ke database.

# Simulasi WOKWI verifikasi respons API dan integrasi sistem.

# Alat dan Bahan

* Laravel 11
* PHP dan Composer
* Database MySQL
* Ngrok
* Postman untuk pengujian API
* VS Code sebagai editor kode
* wokwi
* platform IO

**2.2 Implementation Steps (Langkah Implementasi)**

1. Menyalakan Layanan Api dengan membuka projek laravel sebelumnya

Aktifkan API dengan perintah terminal:  
php artisan serve --host=0.0.0.0 –port=8000

2.Konfigurasi Simulator WOKWI didalam vscode\

* **wokwi.toml**:

[wokwi]

version = 1

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

elf = '[C:\](../../../../../C:/)...\firmware.elf'

* **diagram.json**:

{"version":1,"parts":[{"type":"board-esp32-devkit-c-v4","id":"esp"}],...}

* **platformio.ini**:

monitor\_speed = 115200

lib\_deps = adafruit/DHT sensor library

3.Isi file main.cpp dengan syntax berikut

#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://3144-2001-448a-90b0-cbc-9f53-fcf6-b462-1cd3.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();

}

}

4.Jalankan Simulatornya di vscode

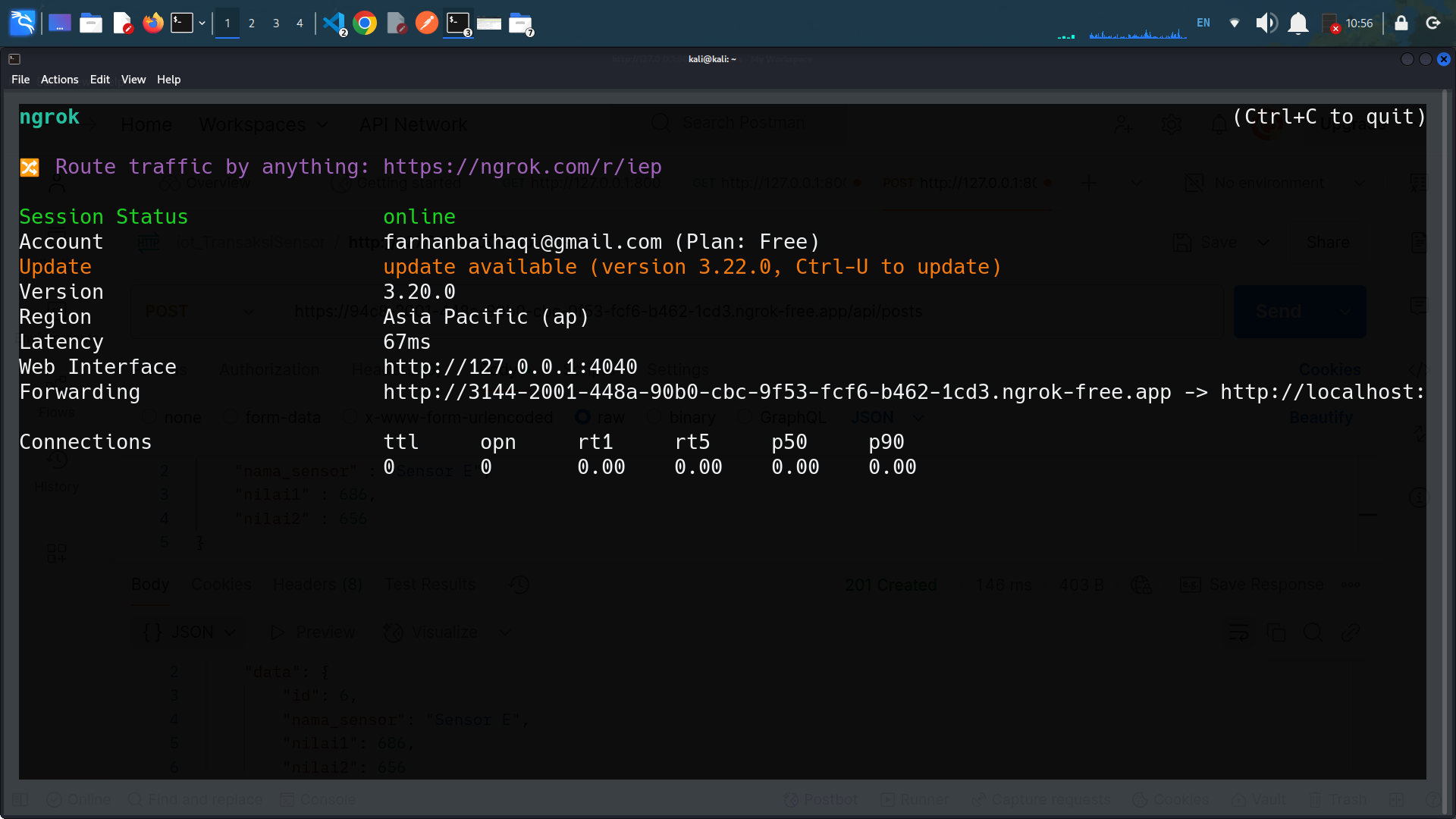
di dalam file diagram.json klik run

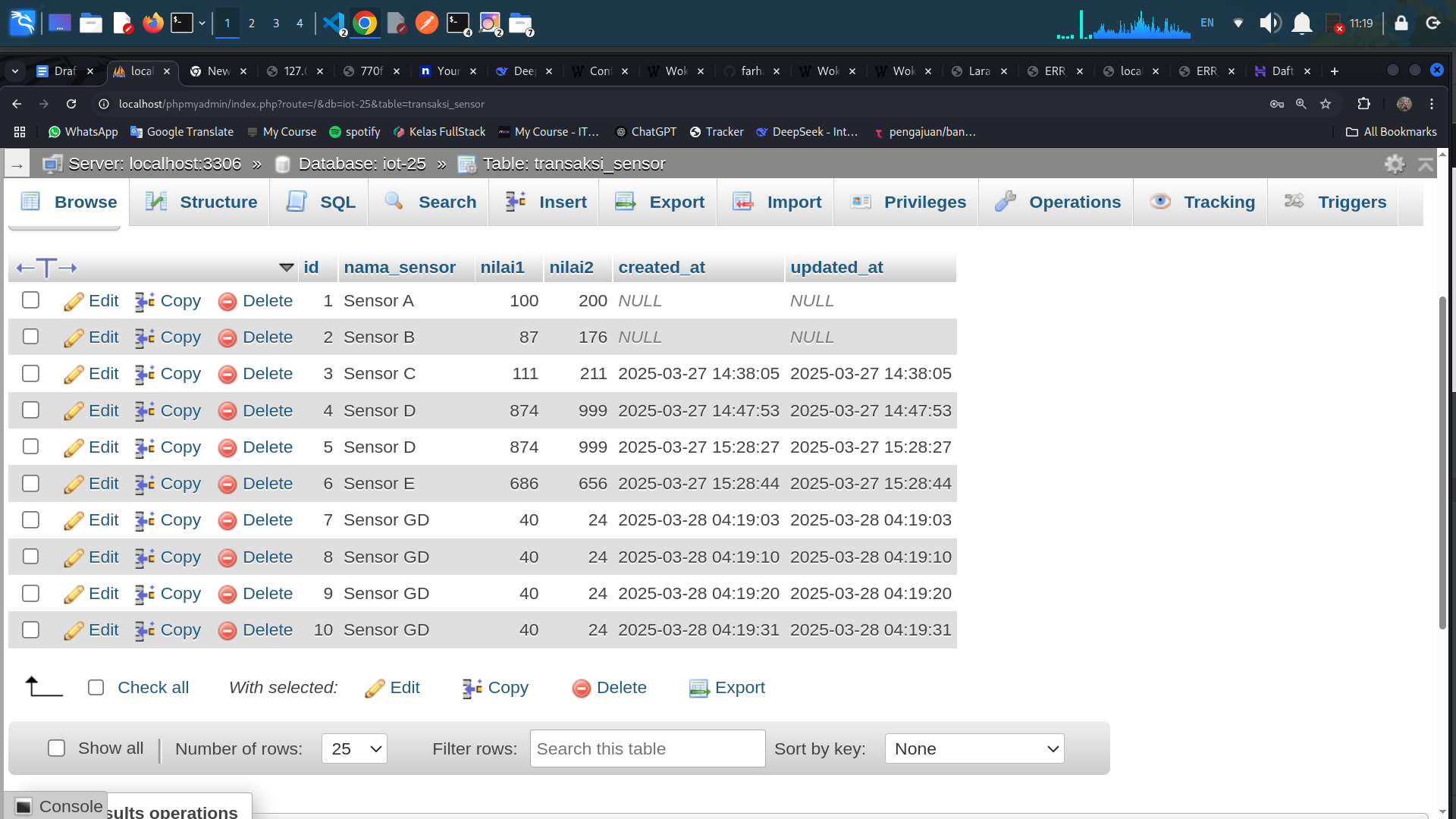
# Results and Discussion (Hasil dan Pembahasan)

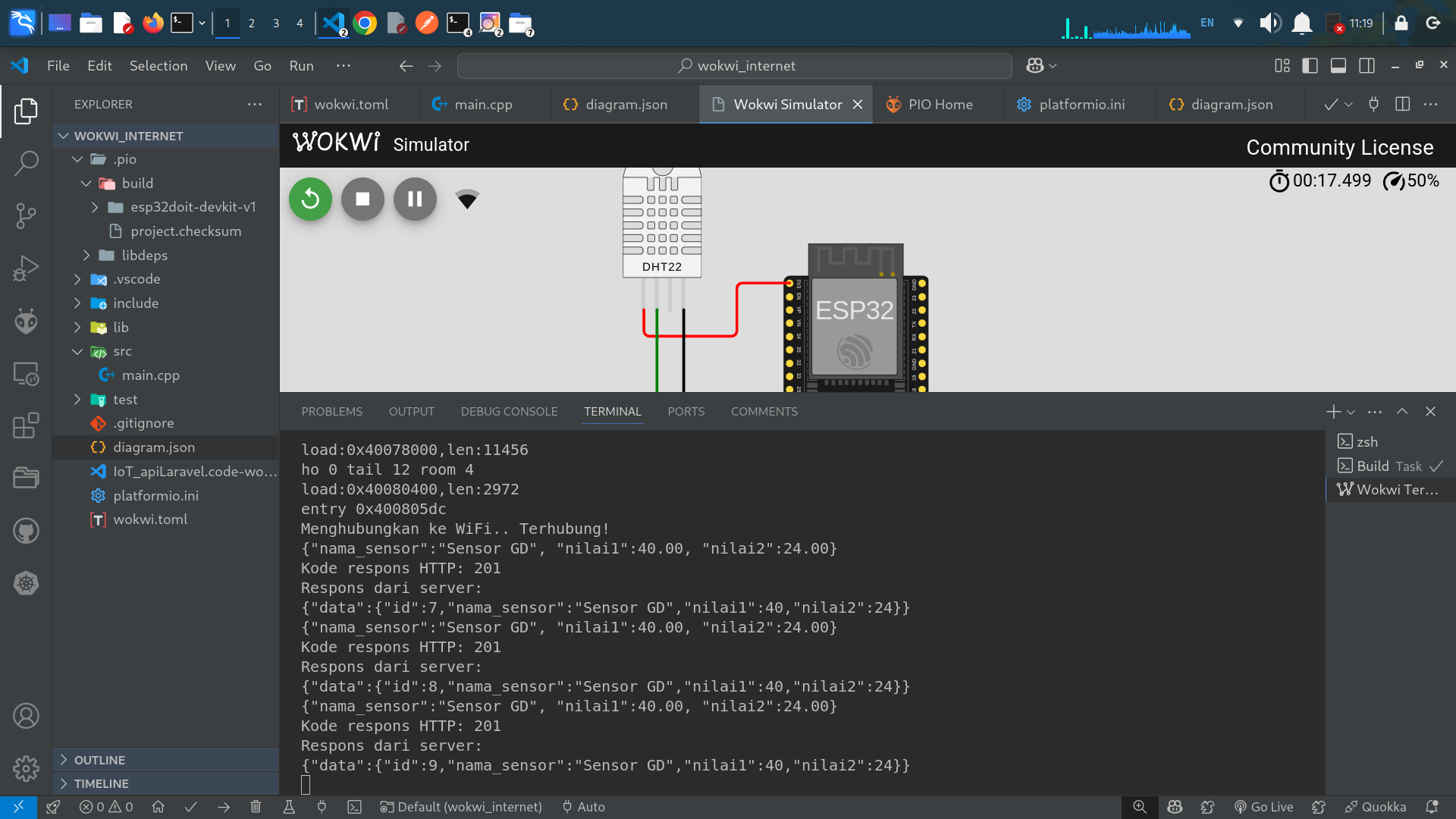
**Koneksi WiFi**: ESP32 terkoneksi ke Wokwi-GUEST dengan output:  
Menghubungkan ke WiFi... Terhubung!

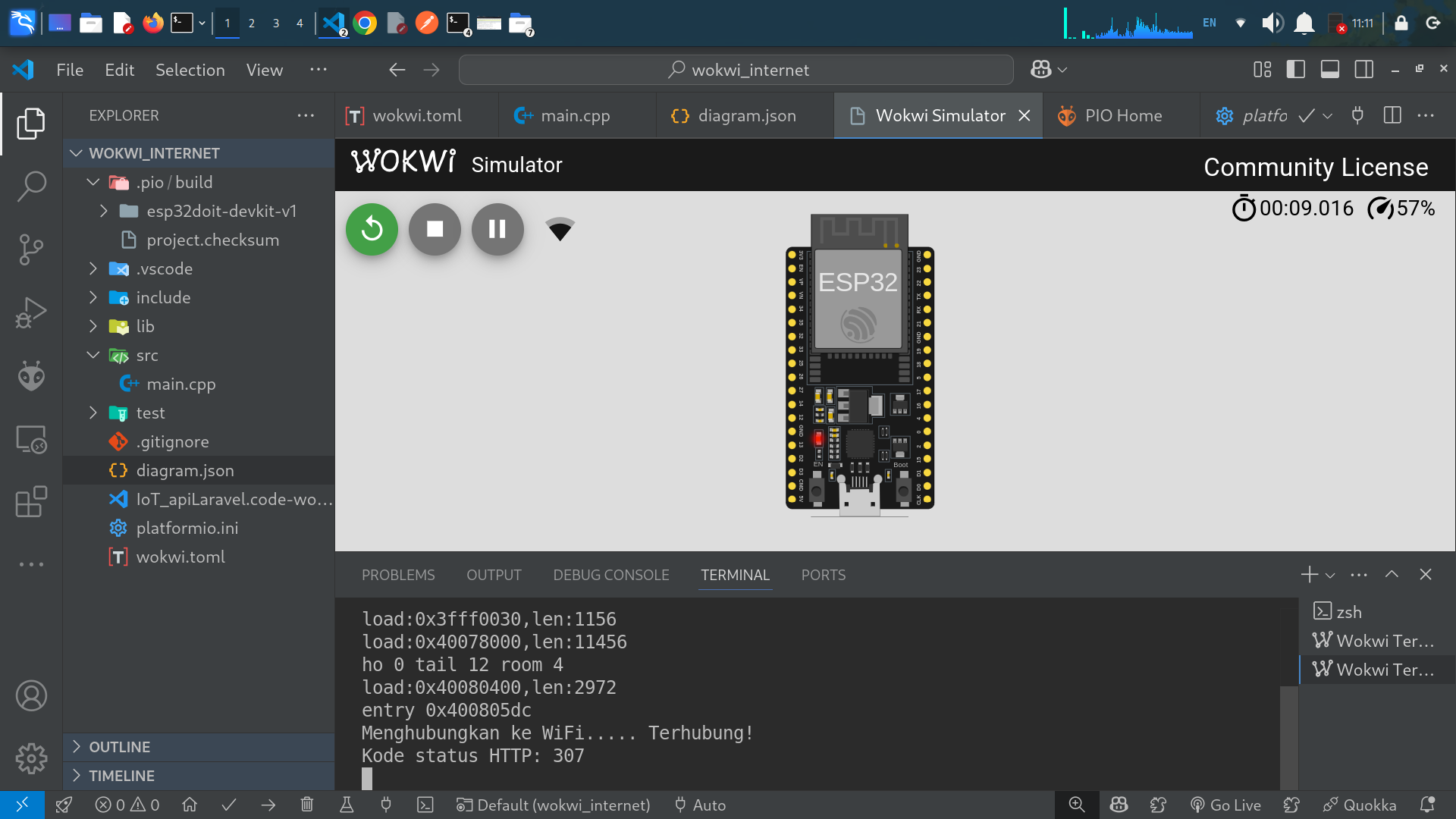
**Akses API**: Respons server sukses (HTTP 200) dengan pesan:  
{"success":true,"message":"Data tersimpan"}

* 1. **Experimental Results (Hasil Eksperimen)**

****

****

****

****