# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

# Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**

**Author(s)** Muhammad Hafizh Al Furqon

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [*muhammad.hafizhalfurqon@gmail.com*](mailto:muhammad.hafizhalfurqon@gmail.com)

Perkembangan teknologi web semakin mendorong kebutuhan akan API (Application Programming Interface) yang efisien dan mudah diakses. Laravel, sebagai salah satu framework PHP modern, menyediakan fitur untuk membangun API secara cepat dan aman. Namun, keterbatasan akses ke server lokal menjadi kendala saat pengujian dan pengembangan. Ngrok hadir sebagai solusi dengan memungkinkan server lokal diekspos ke internet melalui tunneling. Laporan ini memaparkan langkah-langkah pembuatan API menggunakan Laravel 12, integrasi dengan Ngrok untuk akses publik, serta pengujian akses API melalui simulasi ESP32 di Wokwi. Hasil praktikum menunjukkan bahwa kombinasi Laravel dan Ngrok tidak hanya mempercepat proses pengembangan API, tetapi juga memudahkan integrasi perangkat IoT melalui simulasi.

**Keywords**—*API, Laravel 11, Ngrok, IoT, Tunneling*

**1. Introduction**

**1.1 Latar belakang**

API menjadi tulang punggung integrasi sistem dalam pengembangan perangkat lunak modern, khususnya pada aplikasi berbasis IoT. Laravel memudahkan pembuatan API dengan menyediakan berbagai fitur bawaan yang aman dan scalable. Akan tetapi, untuk menguji API pada lingkungan lokal dan mengintegrasikannya dengan perangkat IoT, diperlukan metode agar API dapat diakses dari jaringan eksternal. Ngrok menyediakan layanan tunneling yang memungkinkan server lokal Laravel dapat diakses melalui URL publik. Dengan demikian, praktikum ini bertujuan mengimplementasikan API dengan Laravel dan mengaksesnya melalui simulator ESP32 di Wokwi.

**1.2 Tujuan eksperimen**

1. Membangun API menggunakan Laravel 12 dengan implementasi operasi CRUD pada data sensor.
2. Mengkonfigurasi database (iot\_25) dan melakukan migrasi model menggunakan Eloquent ORM.
3. Mengintegrasikan API dengan Ngrok untuk akses dari luar jaringan lokal.
4. Mengakses API melalui simulator ESP32 di Wokwi menggunakan modul WiFi dan HTTPClient.

**2. Methodology (Metodologi)**

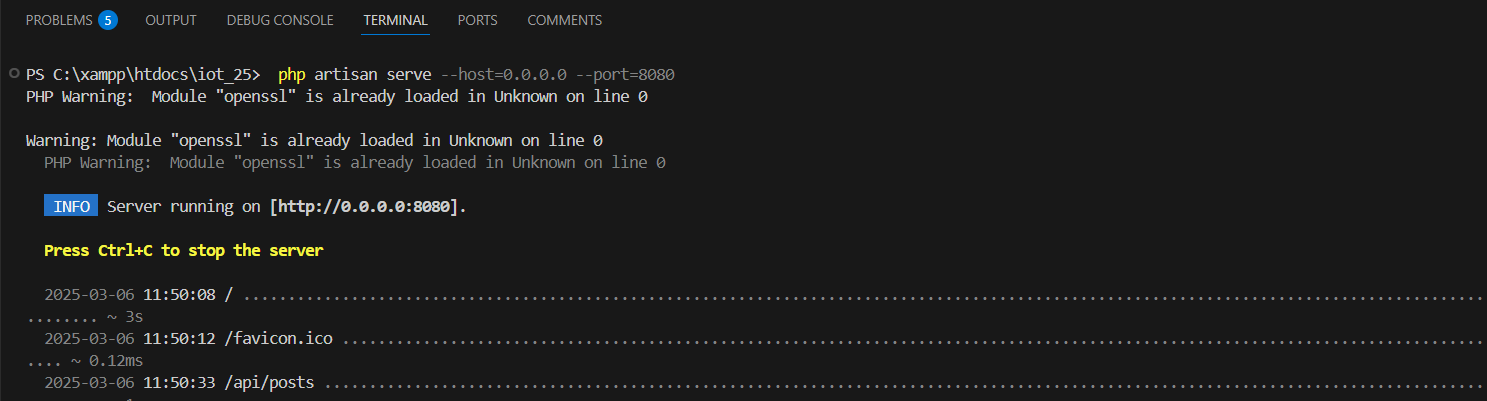
**2.1 Tools & Materials (Alat dan Bahan)**

* **Perangkat Lunak:**
  + Laravel 12
  + PHP 8+
  + Composer
  + MySQL & phpMyAdmin
  + Ngrok
  + Postman
* **Perangkat Keras:**
  + Laptop/PC dengan koneksi internet

**2.2 Implementation Steps (Langkah Implementasi)**

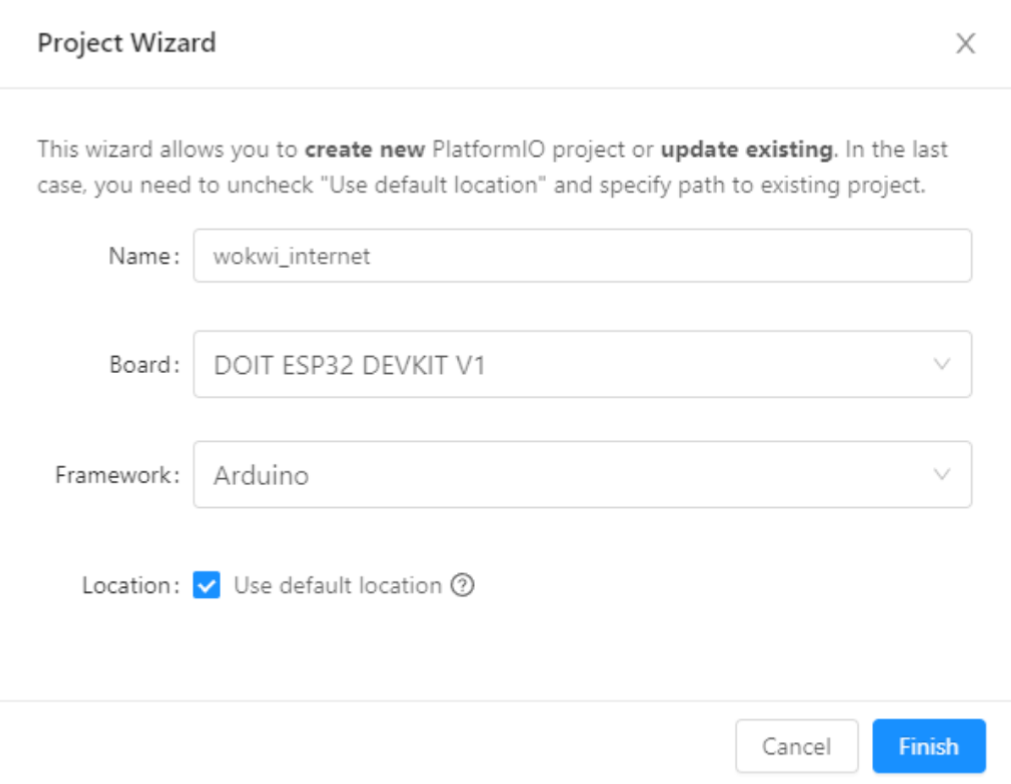
1. Jalankan aplikasi Laravel 11 API yang sebelumnya sudah dibuat

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



Menggunakan host 0.0.0.0 agar API laravel dapat diakses dari IP Address manapun

1. Buat project Wokwi baru menggunakan PlatformIO untuk mengirim hasil data dari ESP32 melalui API Laravel dengan board **DOIT ESP32 DEVKIT V1** dan framework **Arduino**

****

1. Pada src/main.cpp gunakan script berikut.

#include <WiFi.h>

#include <HTTPClient.h>

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.");

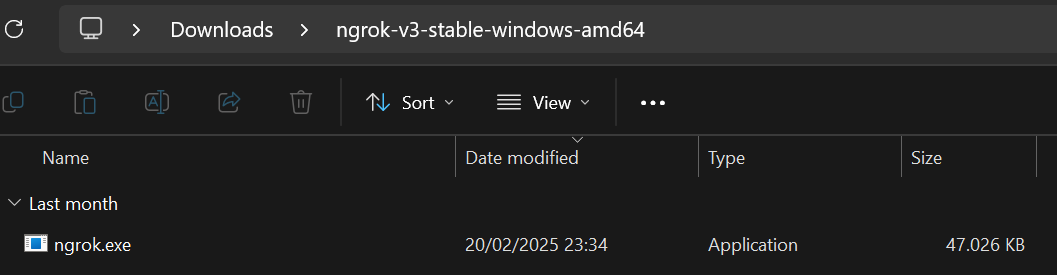
}

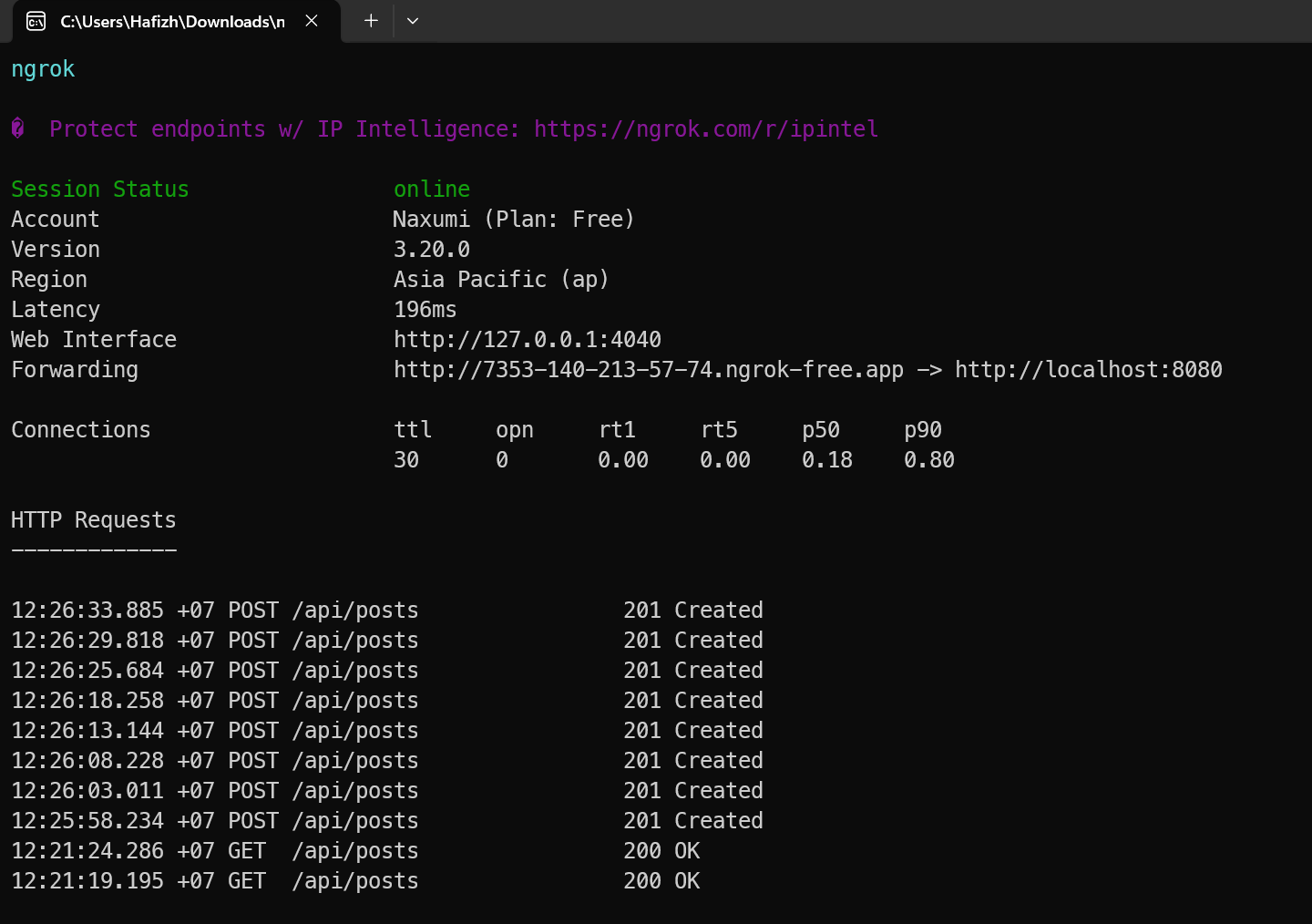
}

}

Pada bagian kredensial WiFi karena kita menggunakan versi gratis Wokwi, kita akan menggunakan Public Gateway dari Wokwi dengan SSID **Wokwi-GUEST** dan tanpa password.

Perhatikan bahwa pada bagian const char\* serverUrl kita menggunakan alamat yang sudah dibuat menggunakan Ngrok. Jalankan aplikasi Ngrok kemudian ketik perintah **ngrok http --scheme=http 8080**





1. Tambahkan wokwi.toml dan jangan lupa build src/main.cpp

[wokwi]

version = 1

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

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

1. Tambahkan file diagram.json dengan isi berikut

**{**

**"version": 1,**

**"author": "Uri Shaked",**

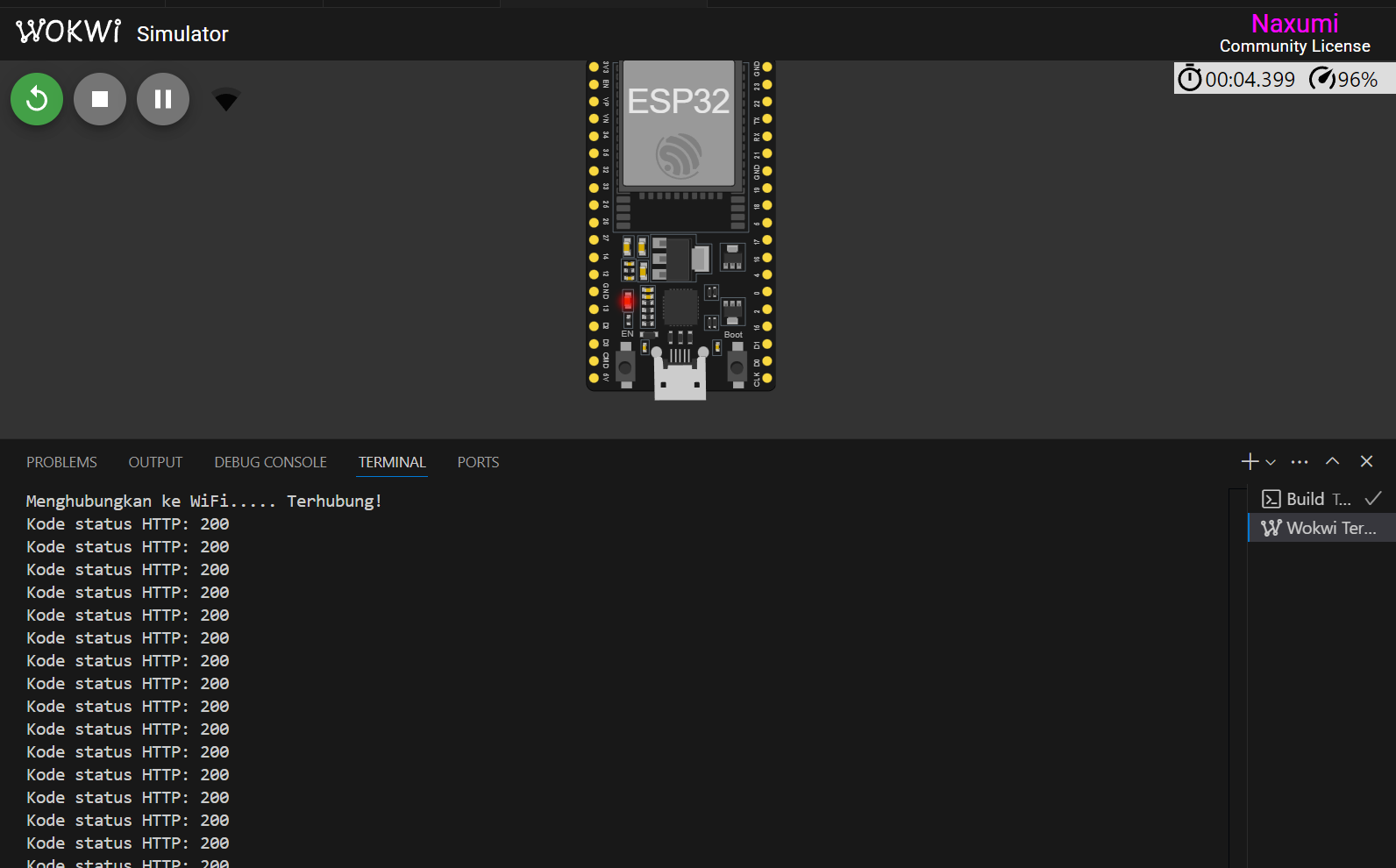
**"editor": "wokwi",**

**"parts": [ { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} } ],**

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

**}**

1. Start simulator Wokwi kemudian jika status code HTTP adalah 200 maka berhasil terkoneksi dengan WiFi Wokwi-GUEST dan berhasil mengakses API Laravel



1. Sekarang ubah kodingan project untuk mengirim sensor data DHT22.

Diagram.json

{

"version": 1,

"author": "KAVITH BUDVIN",

"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": {}

}

Tambahkan **monitor\_speed** untuk memonitor status pengiriman data dari ESP32 sensor DHT22 dan **lib\_deps** untuk sensor DHT (suhu dan kelembapan)

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://7353-140-213-57-74.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

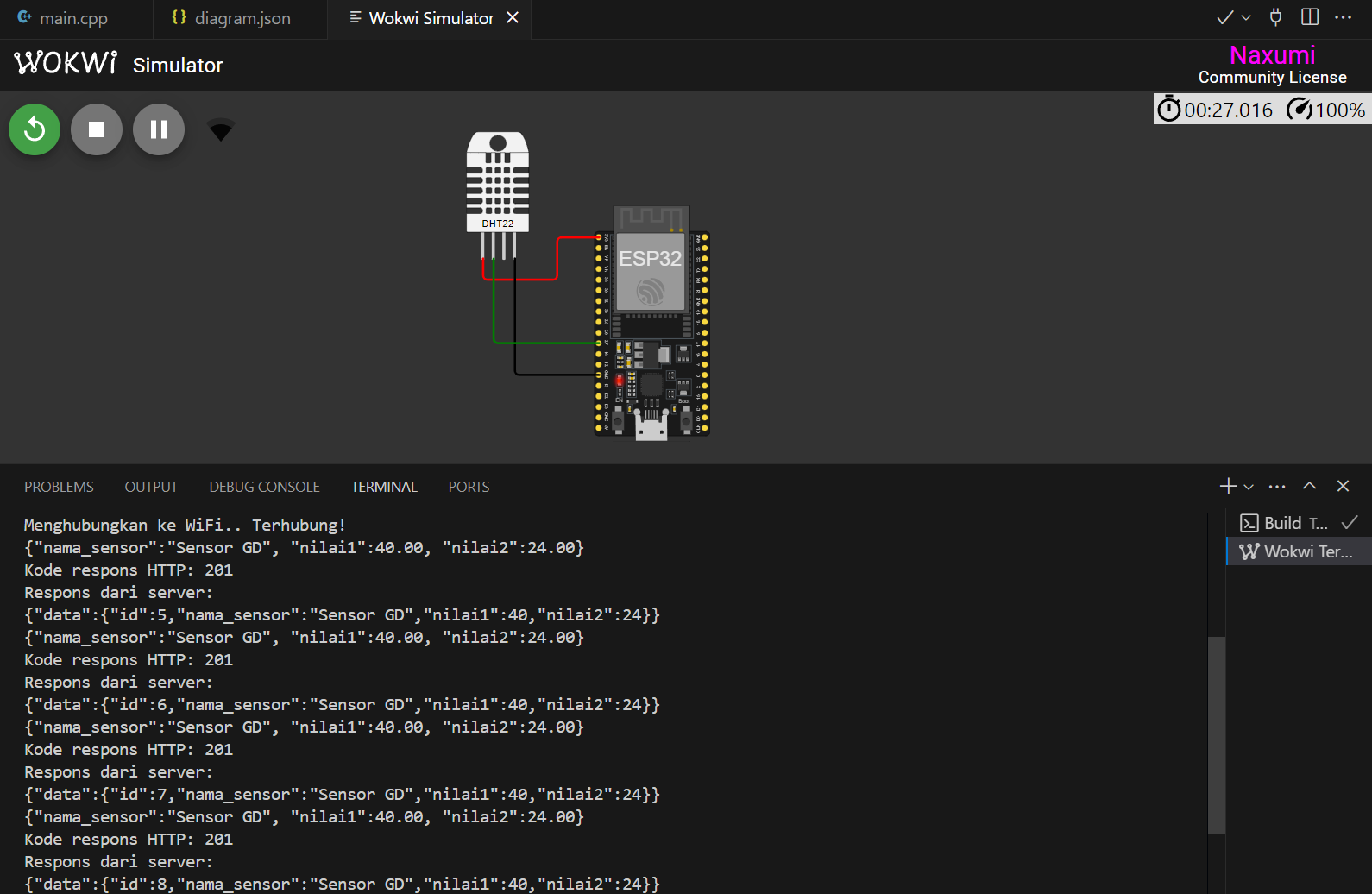
http.end();

}

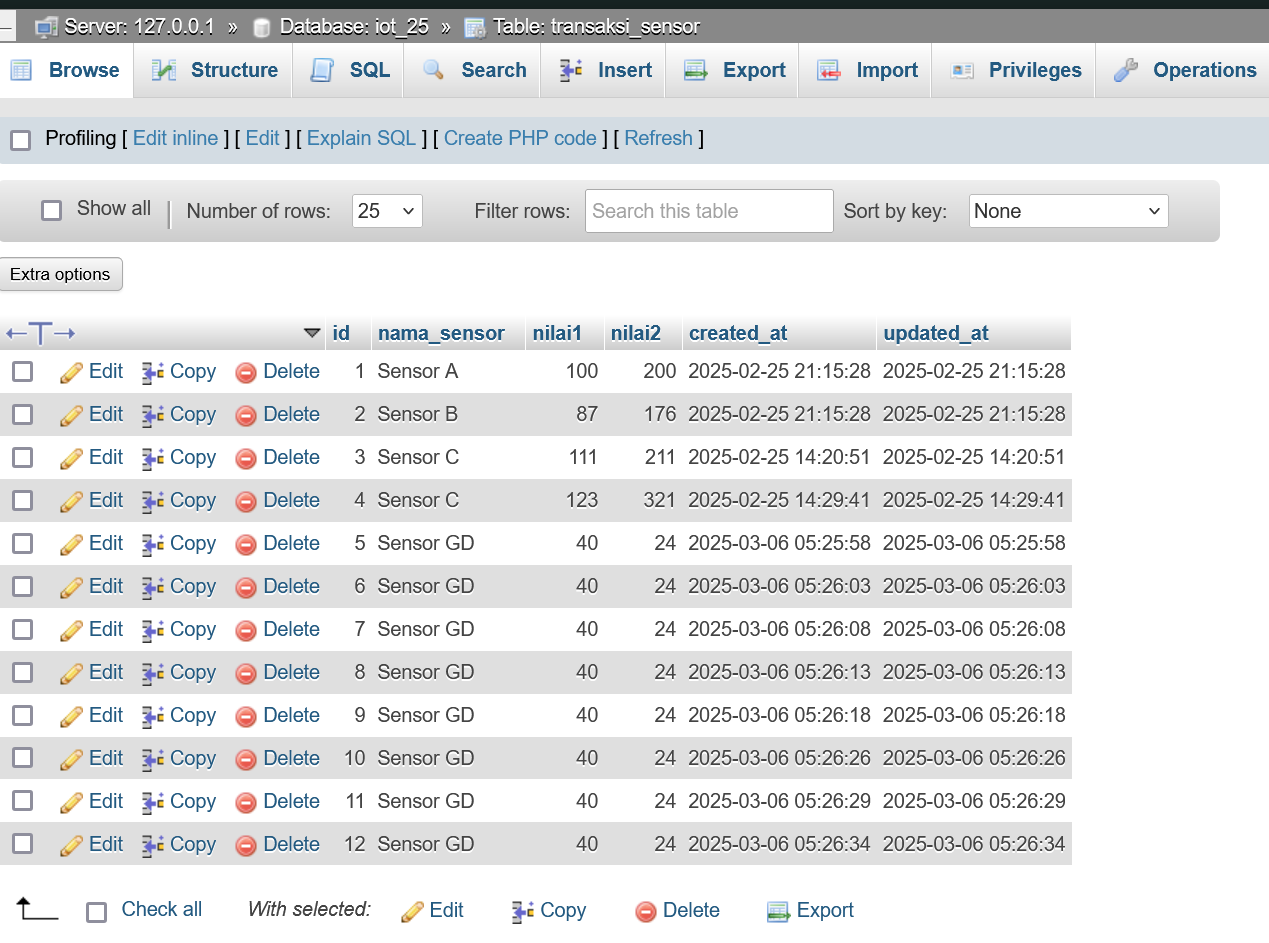
}

Ubah halaman pada const char\* serverUrl sesuai dengan halaman yang di-generate oleh ngrok.

1. Start simulasi Wokwi



Seperti terlihat bahwa status code http adalah 201 yang berarti berhasil dan respons dari server juga terlihat. Pada database yang sudah dibuat data juga muncul



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

* **Backend API:**API Laravel berhasil dibuat dengan endpoint CRUD untuk model *TransaksiSensor*. Pengujian dengan Postman menunjukkan data dapat diambil, dimasukkan, dan diubah melalui API.
* **Ngrok:**Dengan Ngrok, API dapat diakses secara publik. URL Ngrok yang dihasilkan memungkinkan perangkat eksternal (atau simulator) untuk mengakses API.
* **Simulasi WOKWI:**Kode ESP32 pada Wokwi berhasil mengirim data (misalnya, data suhu dan kelembaban dari sensor DHT22) ke API Laravel menggunakan HTTP POST. Tampilan serial monitor menampilkan kode respons HTTP (200/201) sebagai indikasi pengiriman data berhasil.
* **Integrasi Sistem:**Integrasi antara backend API (Laravel), tunneling (Ngrok), dan frontend IoT (simulator Wokwi ESP32) menunjukkan sistem IoT yang komprehensif, dimana data sensor dapat dikirim dan disimpan ke database secara real-time..