LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui**

**Simulasi WOKWI**

***Imran Darajati***

***Fakultas Vokasi, Universitas Brawijaya***

***Email :*** [***imran.darajati@gmail.com***](mailto:imran.darajati@gmail.com)

**Abstrac**

Praktik ini bertujuan untuk mengakses API yang telah dibuat menggunakan Laravel 11 melalui simulasi ESP32 di Wokwi. ESP32 dalam simulasi akan mengirimkan request HTTP ke API Laravel menggunakan WiFiClientSecure dan menerima respons dari API. Akses dilakukan melalui Ngrok, yang memberikan URL publik ke API Laravel. Hasil praktik menunjukkan bahwa ESP32 di Wokwi dapat mengirim dan menerima data dari API secara real-time.

*Kata Kunci: ESP32, Wokwi, API, Laravel 11, Ngrok, HTTP Request, Postman.*

1. **Pendahuluan**

**1.1 Latar Belakang**

Dalam pengembangan Internet of Things (IoT), perangkat seperti ESP32 sering digunakan untuk berkomunikasi dengan server melalui API berbasis HTTP. Namun, saat mengembangkan perangkat IoT, sering kali diperlukan simulasi sebelum implementasi ke perangkat fisik.

Wokwi adalah simulator ESP32 berbasis web yang memungkinkan pengujian HTTP request ke API Laravel sebelum menghubungkannya ke perangkat ESP32 sebenarnya. Dengan bantuan Ngrok, API Laravel yang berjalan di localhost dapat diakses oleh ESP32 di Wokwi.

**1.2 Tujuan Eksperimen**

Tujuan dari eksperimen ini adalah :

1. Menghubungkan ESP32 di Wokwi dengan API Laravel menggunakan HTTP request.
2. Menggunakan Ngrok untuk membuka akses API ke internet.
3. Mengirim dan menerima data API melalui ESP32 di Wokwi.
4. **Metodologi**

**2.1 Alat dan Bahan**

* Software yang Digunakan:

1. Visual Studio Code (VS Code) – Untuk mengembangkan API Laravel.
2. XAMPP – Untuk menjalankan database MySQL.
3. Ngrok – Untuk memberikan URL publik ke API Laravel.
4. Postman – Untuk menguji API sebelum digunakan oleh ESP32.
5. Wokwi Simulator – Untuk menjalankan simulasi ESP32 dan mengakses API.

**2.2 Langkah Implementasi**

1. Menjalankan API Laravel :

* Jalankan Laravel di VS Code:

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

* Jalankan Ngrok untuk memberikan URL publik:

ngrok http --scheme=http 8080

* Salin URL Ngrok yang diberikan, misalnya:

<https://abcd1234.ngrok.io>

1. Membuat Project Wokwi di PlatformIO
2. Membuat Project Baru :

* Buka VS Code dan pilih PlatformIO Home.
* Buat Project Baru dengan Board ESP32 Dev Module.
* Buka folder project dan tambahkan file berikut:
* wokwi.toml
* diagram.json
* Modifikasi platformio.ini

1. Percobaan 1: ESP32 Terhubung ke WiFi dan Mengakses API :

Pada percobaan pertama, ESP32 di Wokwi diuji untuk terhubung ke WiFi dan melakukan request HTTP ke API Laravel.

* Buka Wokwi dan buat proyek baru di PlatformIO.
* Tambahkan file berikut:
* wokwi.toml
* diagram.json
* Konfigurasi wokwi.toml

Buat file wokwi.toml di dalam folder project dan tambahkan kode berikut :

[wokwi]

version = 1

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

elf = 'C:\Users\mokor\Documents\PlatformIO\Projects\wokwi\_internet\.pio\build\esp32doit-devkit-v1\firmware.elf'

* Konfigurasi diagram.json
* Tambahkan file diagram.json untuk mendefinisikan rangkaian ESP32.

{

"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", "", [] ] ]

}

* Kode ESP32 untuk Percobaan 1

Tambahkan kode berikut di src/main.cpp untuk menghubungkan ESP32 ke WiFi dan mengakses API Laravel:

#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"; // Sesuaikan dengan URL API Laravel dari Ngrok Anda

// 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.");

}

}

}

1. Percobaan 2: Mengirim Data Sensor Suhu & Kelembaban ke API :

Setelah ESP32 berhasil mengakses API, percobaan berikutnya adalah menambahkan sensor DHT22 untuk membaca suhu dan kelembaban lalu mengirimkannya ke API Laravel.

* Tambahkan sensor DHT22 ke 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": {}

}

* Modifikasi platformio.ini untuk menambahkan library DHT :

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

monitor\_speed = 115200

lib\_deps =

adafruit/DHT sensor library

* Perbarui main.cpp untuk membaca suhu & kelembaban dan mengirimkannya ke API :

#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"; // Sesuaikan dengan URL API Laravel dari Ngrok Anda

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();

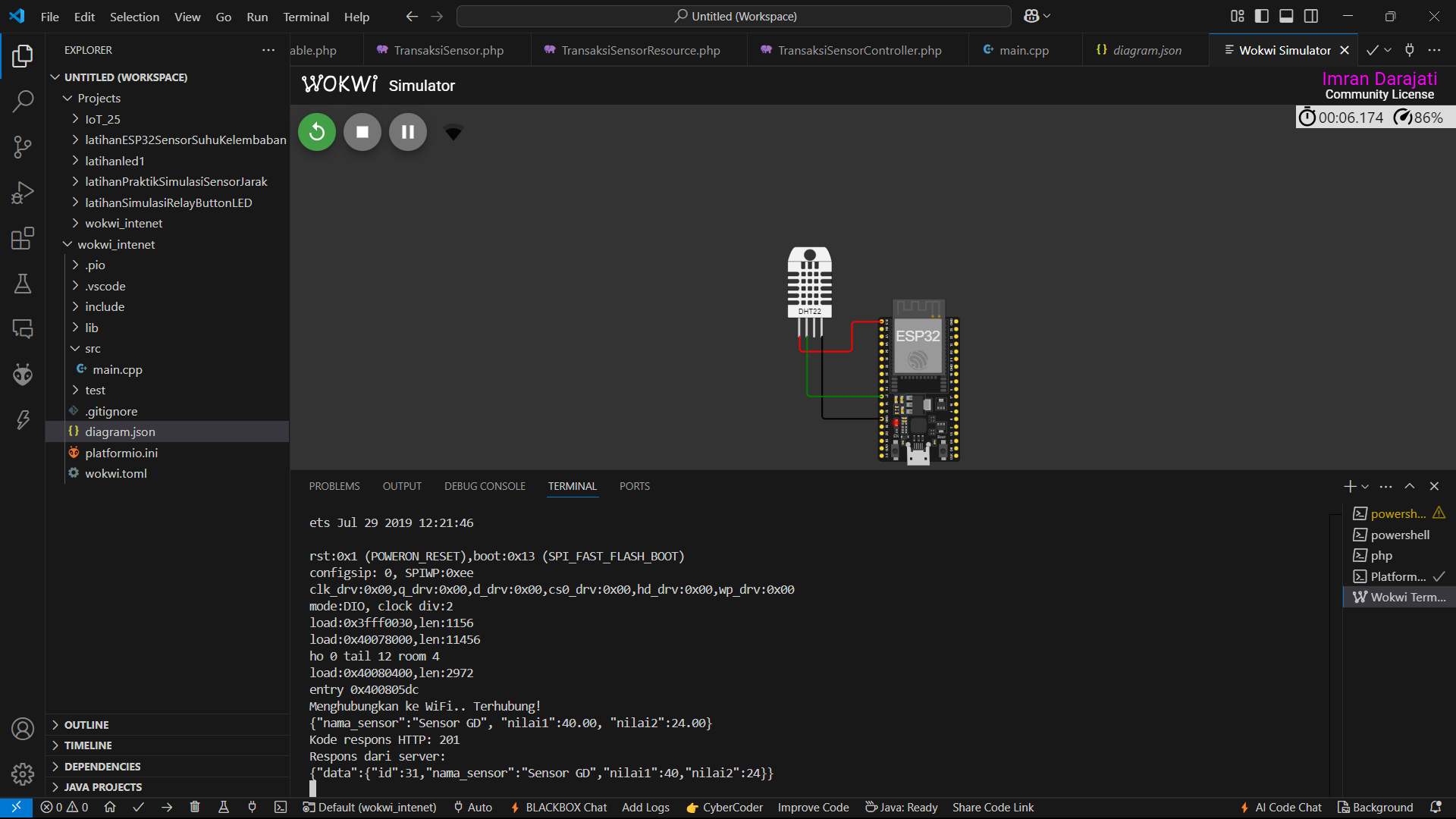
}

}

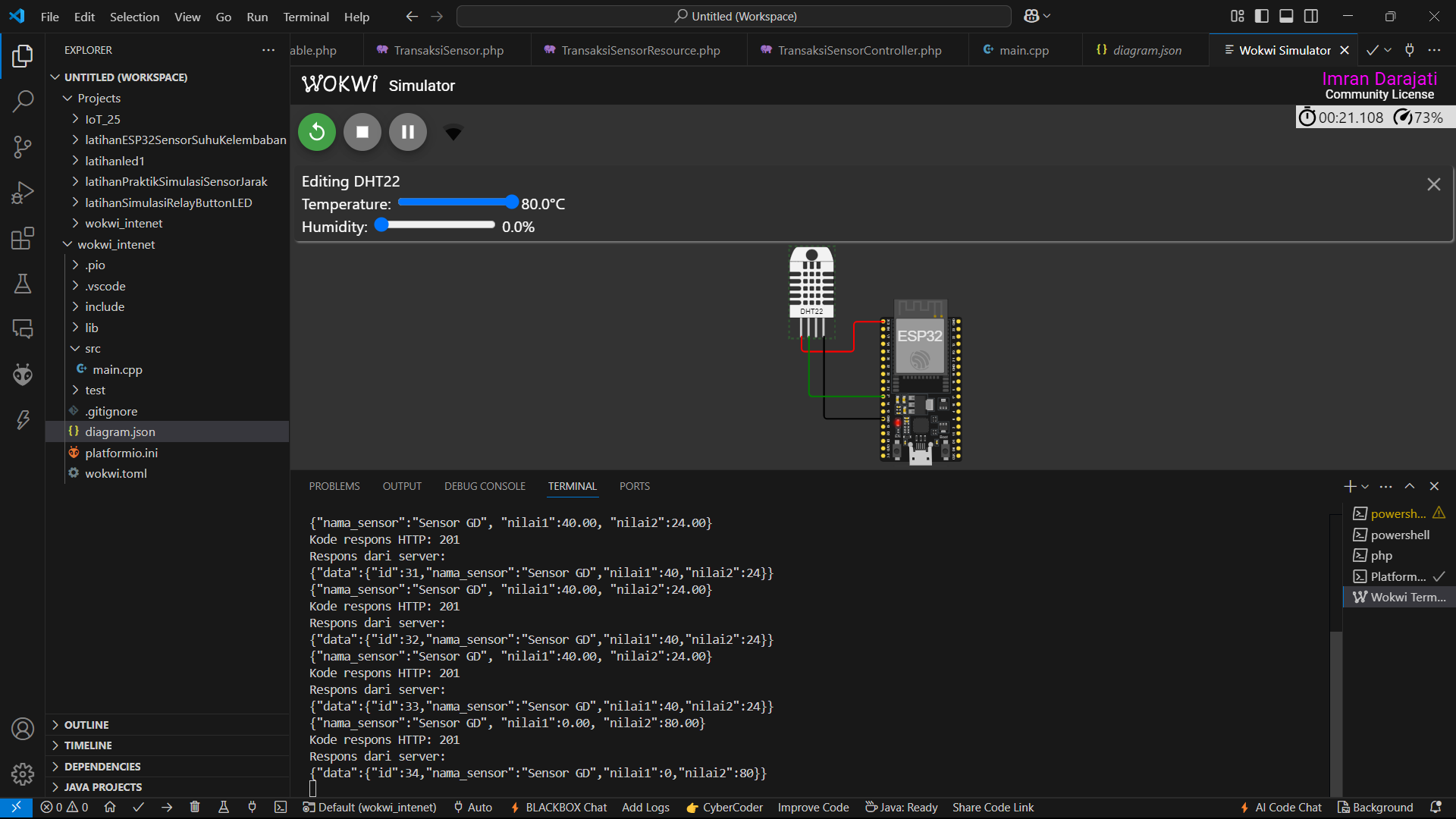
1. **Hasil dan Pembahasan**

Berikut adalah tampilan hasil pengujian akses API melalui simulasi Wokwi :

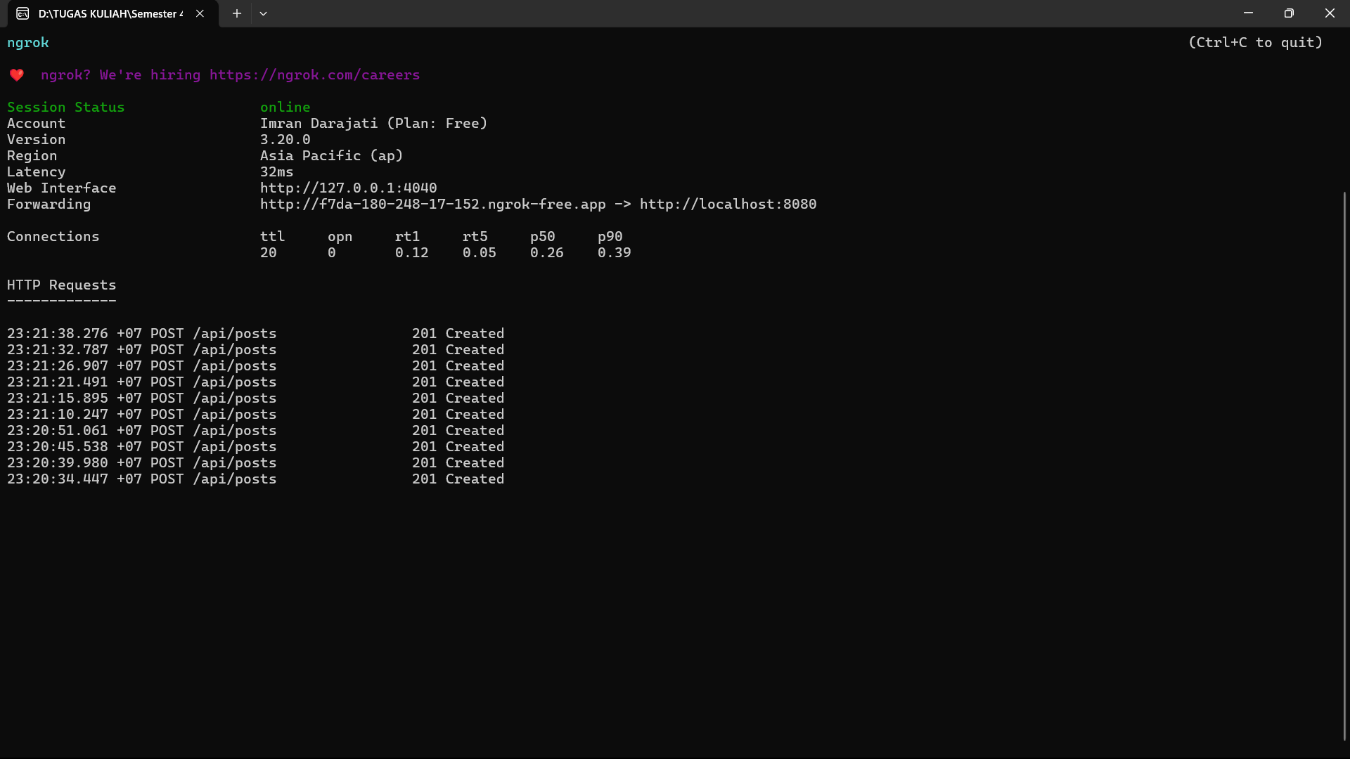
1. ESP32 berhasil terhubung ke WiFi Wokwi-GUEST dan melakukan request ke API Laravel menggunakan HTTPClient :



1. Hasil pengiriman data dari ESP32 ke API Laravel, yang dapat dilihat dari Serial Monitor di Wokwi :



1. Tampilan URL Ngrok yang aktif dan digunakan oleh ESP32 di Wokwi untuk mengakses API Laravel secara online :



1. Tampilan data sensor suhu dan kelembaban yang dikirim ke API, ditampilkan di database MySQL :

