# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

# Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**

*Faiz Henri Kurniawan*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: faizhenrik@student.ub.ac.id*

### **1. Menjalankan API Laravel**

Jalankan API Laravel dengan perintah berikut:

php artisan serve --host=0.0.0.0 --port=8080 (in my case 8000)

Perintah di atas memastikan API Laravel dapat diakses dari IP Address manapun dan bekerja pada port 8080.

### **2. Membuat File Baru di Wokwi Simulator**

Buat file main.cpp di PlatformIO dengan kode berikut:

#include <WiFi.h>

#include <HTTPClient.h>

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

const char\* serverUrl = "http://your-ngrok-url/api/posts";

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();

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

if (WiFi.status() == WL\_CONNECTED) {

HTTPClient http;

http.begin(serverUrl);

int httpResponseCode = http.GET();

Serial.print("Kode status HTTP: ");

Serial.println(httpResponseCode);

http.end();

} else {

Serial.println("WiFi tidak terhubung.");

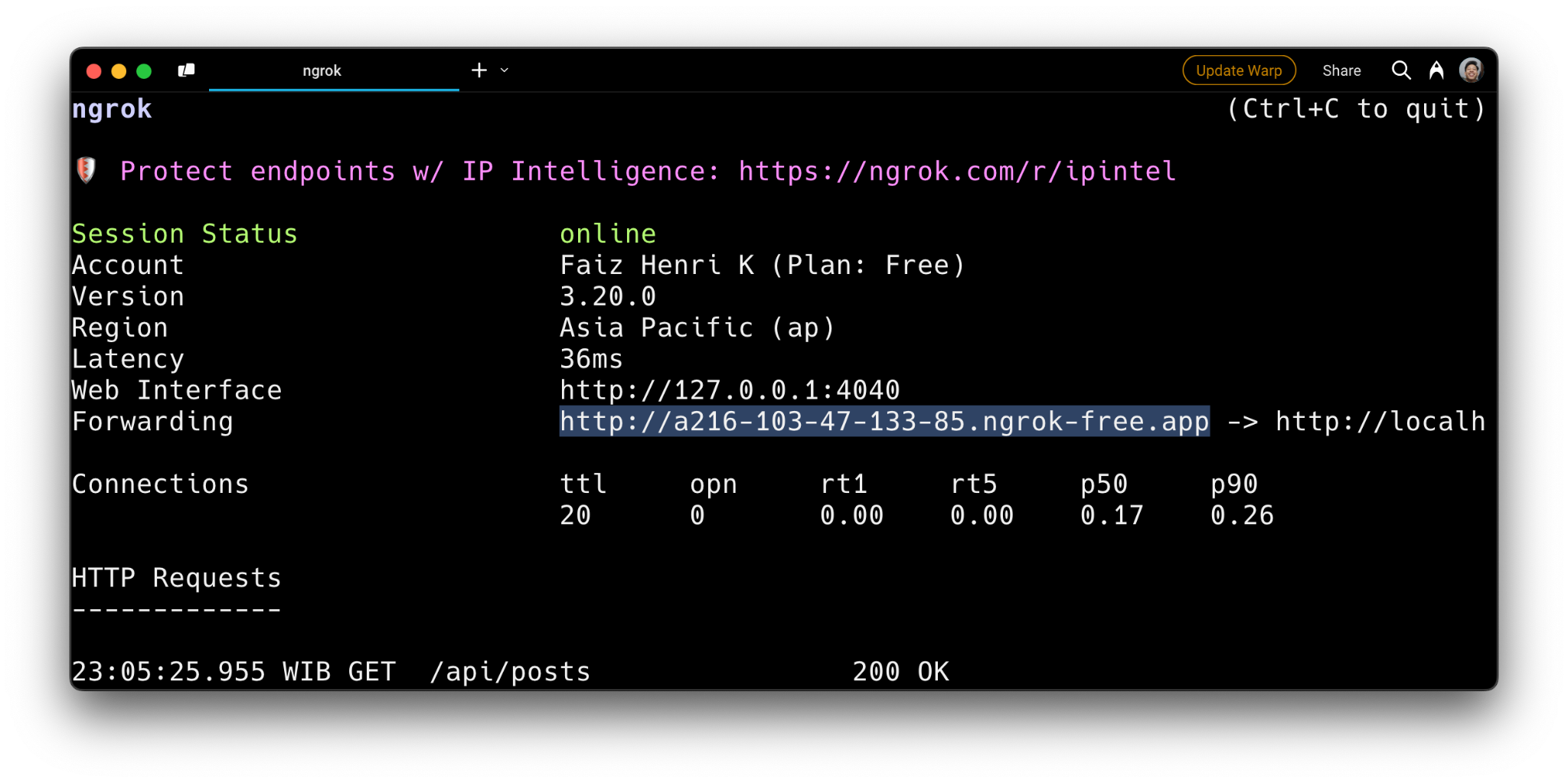
}

}

}

Gantilah serverUrl dengan URL yang diberikan oleh ngrok. Jalankan ngrok dengan perintah berikut:

ngrok http --scheme=http 8080



### **3. Menambahkan File Konfigurasi**

#### **wokwi.toml**

[wokwi]

version = 1

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

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

#### **diagram.json**

{

"version": 1,

"author": "Faiz Henri K",

"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", "", [] ]]

}

### **4. Menjalankan Simulasi**

Gunakan perintah berikut untuk memulai simulator:

Wokwi Start Simulator

Jika berhasil, akan muncul **Kode Status HTTP: 200** pada serial monitor.

### **5. Modifikasi untuk Menggunakan Sensor DHT22**

Tambahkan sensor DHT22 untuk mengirim data suhu dan kelembaban ke API Laravel.

#### **diagram.json (dengan DHT22)**

{

"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", [] ],

[ "esp:GND.1", "dht1:GND", "black", [] ],

[ "dht1:SDA", "esp:27", "green", [] ]

]

}

#### **main.cpp (dengan DHT22)**

#include <Arduino.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 27

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

unsigned long previousMillis = 0;

const long interval = 5000;

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!");

dht.begin();

}

void loop() {

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

float h = round(dht.readHumidity());

float t = round(dht.readTemperature());

if (isnan(h) || isnan(t)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

HTTPClient http;

String url = "http://your-ngrok-url/api/posts";

http.begin(url);

http.addHeader("Content-Type", "application/json");

String payload = "{\"nama\_sensor\":\"Sensor GD\", \"nilai1\":" + String(h) + ", \"nilai2\":" + String(t) + "}";

Serial.println(payload);

int httpResponseCode = http.POST(payload);

Serial.print("Kode respons HTTP: ");

Serial.println(httpResponseCode);

if (httpResponseCode == 200 || httpResponseCode == 201) {

String response = http.getString();

Serial.println("Respons dari server:");

Serial.println(response);

} else {

Serial.println("Gagal mengirim data");

}

http.end();

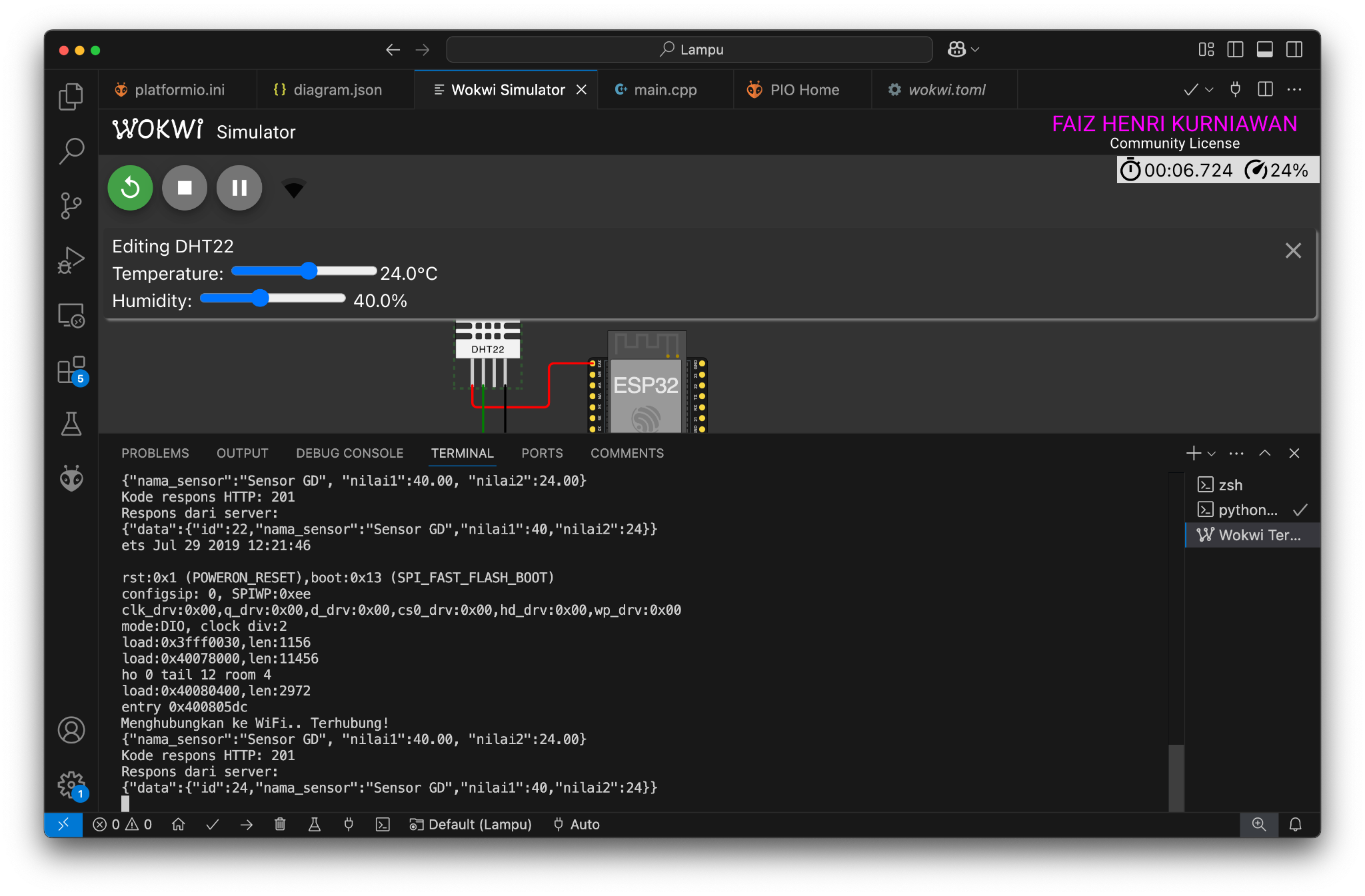
}

}

### **6. Menjalankan Simulasi dengan Sensor**

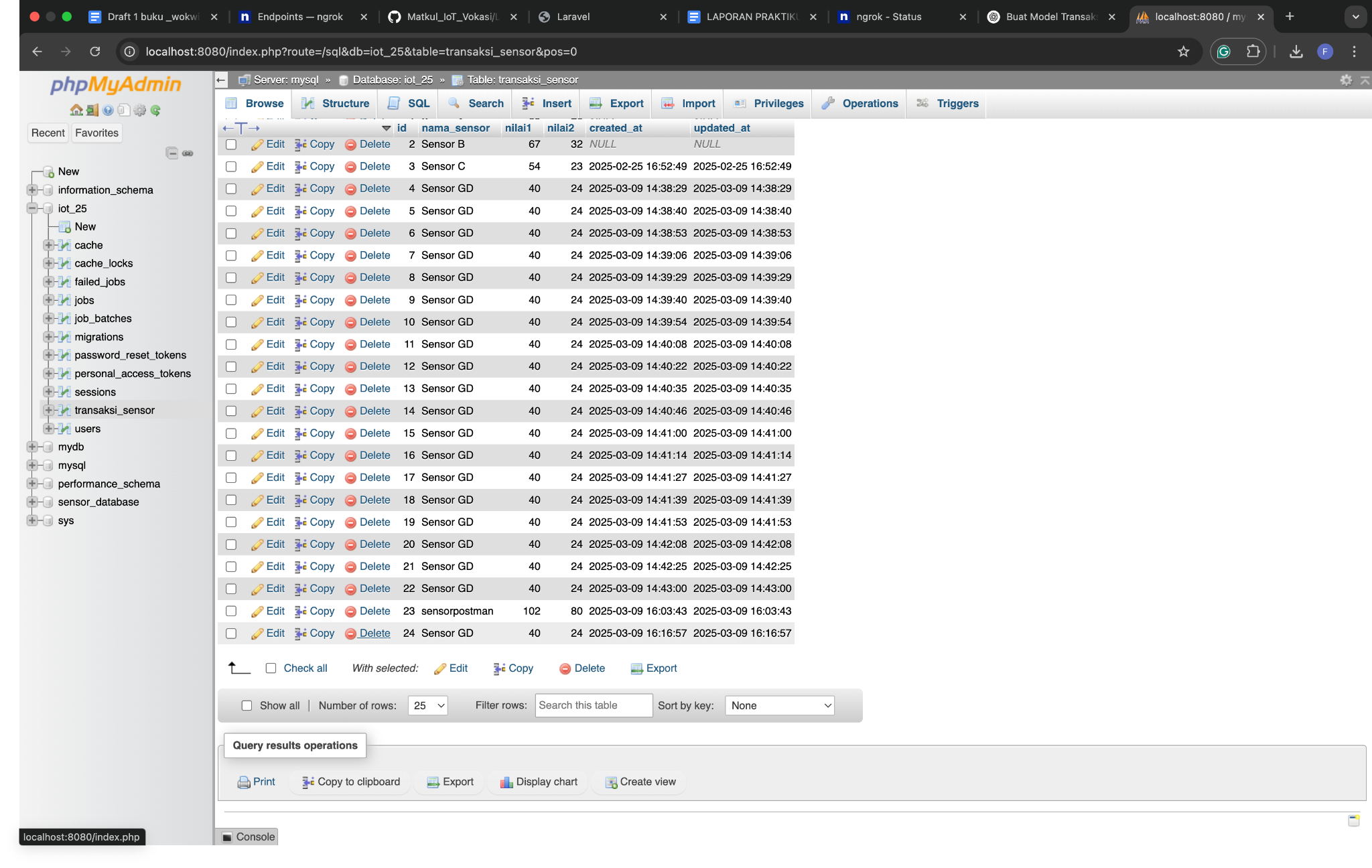
Gunakan perintah berikut:

Wokwi Start Simulator



### **7. Verifikasi Data di Database**

Pastikan bahwa data suhu dan kelembaban berhasil disimpan dalam database MySQL yang digunakan oleh Laravel.



**Kesimpulan:** Dengan langkah-langkah ini, kita berhasil melakukan simulasi pengiriman data suhu dan kelembaban menggunakan Wokwi, ESP32, dan API Laravel. Status HTTP 200 menandakan bahwa permintaan berhasil diproses oleh server.

**Catatan:** Pastikan untuk mengganti your-ngrok-url dengan URL yang diberikan oleh ngrok setiap kali dijalankan.