LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Relay, Button & LED (Minggu 3)**

*Fayola Carani Malya  
Fakultas Vokasi, Universitas Brawijaya  
Email: fayolacarani@gmail.com*

**Abstract**   
Relay, button, dan LED adalah komponen penting dalam dunia elektronika yang sering digunakan untuk membangun sistem kontrol sederhana hingga kompleks. Relay berfungsi sebagai saklar elektrik yang dapat diaktifkan dengan sinyal kecil untuk mengendalikan perangkat dengan daya lebih tinggi. Button digunakan sebagai input mekanis untuk mengaktifkan atau memicu suatu aksi, sementara LED (Light Emitting Diode) merupakan komponen output yang memberikan indikasi visual.

Dalam praktik ini, simulasi dilakukan untuk mempelajari integrasi antara relay, button, dan LED menggunakan mikrokontroler sebagai pengendali utama. Proses simulasi mencakup pengaturan relay untuk mengontrol LED, pemrograman button sebagai input trigger, serta implementasi logika sederhana untuk mengatur hubungan antara ketiga komponen tersebut. Hasil dari simulasi menunjukkan bahwa sistem dapat bekerja sesuai dengan skenario yang dirancang, seperti mengaktifkan atau mematikan LED melalui relay dengan trigger dari button. Praktik ini memberikan pemahaman dasar tentang penggunaan relay, button, dan LED dalam berbagai aplikasi, seperti sistem otomasi rumah, kontrol perangkat elektronik, dan pengendalian beban listrik.

Keywords —Relay, Button, LED, Simulasi Elektronika, Sistem Kontrol

**1. Introduction** (Pendahuluan)

**1.1 Latar belakang**

Komponen elektronika seperti relay, button, dan LED memiliki peran yang signifikan dalam sistem kontrol dan otomasi. Relay adalah saklar elektronik yang memungkinkan pengendalian perangkat dengan daya lebih tinggi menggunakan sinyal listrik kecil, sehingga sering digunakan dalam sistem otomasi industri dan rumah pintar. Button, sebagai input mekanis, berfungsi untuk memberikan sinyal atau memicu suatu aksi dalam sistem elektronik. Sementara itu, LED (Light Emitting Diode) adalah komponen output yang sering digunakan untuk memberikan indikasi visual, baik sebagai penanda status sistem maupun sebagai alat komunikasi sederhana antara perangkat dan pengguna.

Integrasi ketiga komponen ini memungkinkan terciptanya sistem kontrol sederhana yang fleksibel dan dapat disesuaikan dengan berbagai kebutuhan. Dengan memanfaatkan mikrokontroler sebagai pengendali utama, hubungan antara relay, button, dan LED dapat diatur untuk menjalankan logika kontrol yang kompleks, seperti menyalakan dan mematikan perangkat secara otomatis.

Praktik simulasi ini bertujuan untuk mempelajari prinsip kerja dan implementasi relay, button, dan LED dalam membangun sistem kontrol dasar. Simulasi ini juga memberikan wawasan tentang bagaimana ketiga komponen tersebut dapat diintegrasikan dan diatur menggunakan mikrokontroler untuk menciptakan solusi otomasi yang efektif dan efisien.

**1.2 Tujuan eksperimen**

Penelitian ini bertujuan :   
1. Memahami prinsip kerja relay sebagai saklar elektrik untuk mengendalikan perangkat dengan daya lebih tinggi

2. Mempelajari fungsi button sebagai input mekanis untuk memicu aksi tertentu

3. Mengaplikasikan LED sebagai indicator visual dalam sistem kontrol

4. Mengintegrasikan relay, button, dan LED menggunakan mikrokontroler sebagai pengendali utama

5. Mengeksplorasi potensi aplikasi sistem control ini dalam otomasi rumah dan pengendalian perangkat elektronik  
6. Mengimplementasikan logika control sederhana untuk mengatur hubungan antara relay, button, dan LED

**2. Methodology (Metodologi)**

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

Alat dan Bahan yang digunakan :   
1. Mikrokontroler : ESP 32 Devkit VI  
2. Alat : Button dan Relay Module  
3. LED : Berwarna merah dengan pin GPIO 18  
3. Software : Wokwi, Visual Studio Code

**2.2 Implementation Steps (Langkah Implementasi)**

1. Melakukan simulasi di Wokwi untuk simulasi Relay, Button, & LED sebelum ke Visual Studio Code (code editor).

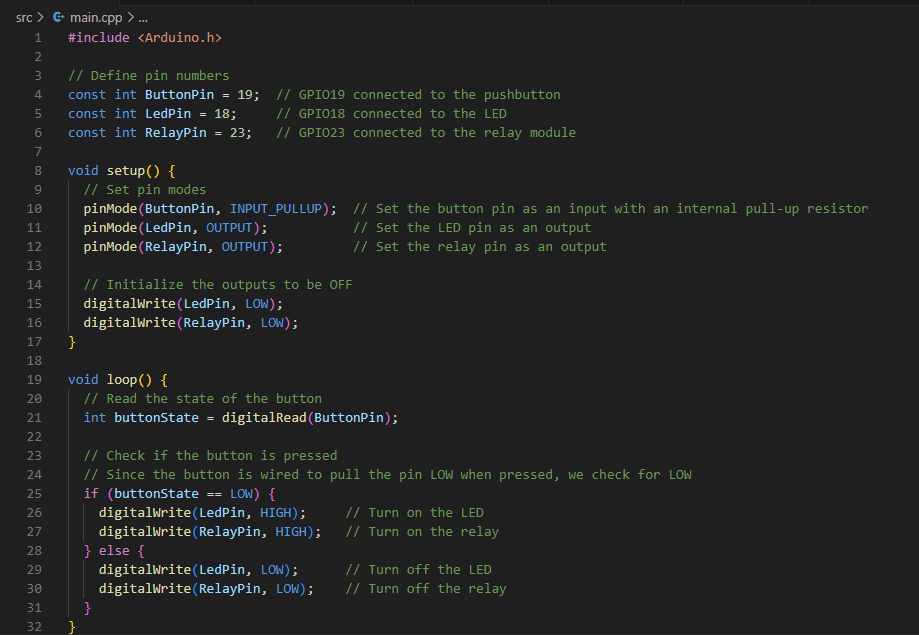
2. Menambahkan ESP32 DevKit V1, Relay Module, Button, dan LED lalu dihubungkan sesuai skema yaitu

* Relay Module : - pin VCC pada relay pin 3.3V pada ESP32
* Pin GND pada relay GND pada ESP32
* Pin IN pada relay pin GPIO 23
* Button : dihubungkan pada GND dan pin GPIO 19
* LED : dihubungkan pada GND dan pin GPIO 18

3. Menuliskan kode program di Wokwi

4. Setelah berhasil disimulasikan di Wokwi, kode program di unggah ke Visual Studio Code dengan membuat proyek baru di PlatformIO dengan board ESP32 DevKit V1.

5. Menuliskan kode pada main.cpp



6. Membuat file diagram.json dan menuliskan program sama persis dengan yang sudah dituliskan pada simulasi diwokwi.

7. Membuat file wokwi.toml untuk menghubungkan firmware hasil kompilasi ke Wokwi.

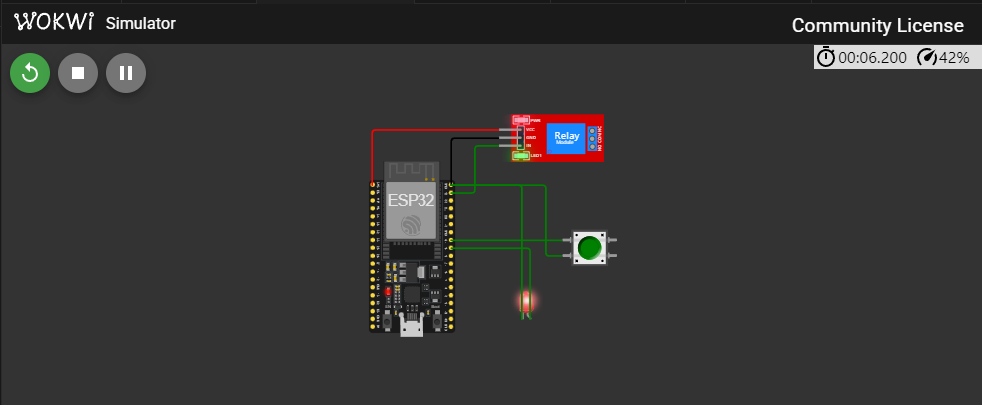
8. Selanjutnya jalankan program dengan memilih build.

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

**3.1 Experimental Results (Hasil Eksperimen)**

Setelah kode program di unggah di Visual Studio Code, akan memunculkan hasil :   
- Lampu pada relay module dan LED akan hidup ketika kita klik button  
- Apabila button dilepas / tidak diklik akan ke posisi semua yaitu LED mati.

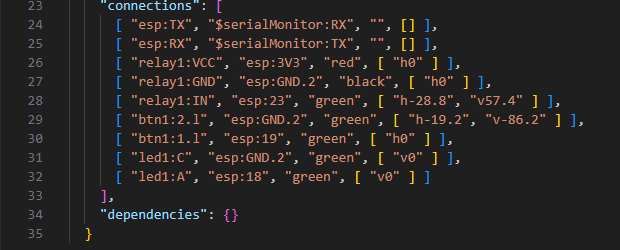
**Screenshoot hasil simulasi pada Visual Studio Code :**



**Pembahasan :   
-** Simulasi berjalan sesuai dengan yang diharapkan  
- Ketika button ditekan, lampu pada relay dan LED akan hidup.

**4. Appendix (Lampiran, jika diperlukan)**





LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Sensor Jarak (Ultrasonic) (Minggu 4)**

*Fayola Carani Malya  
Fakultas Vokasi, Universitas Brawijaya  
Email: fayolacarani@gmail.com*

**Abstract**   
Sensor ultrasonic adalah perangkat elektronik yang digunakan untuk mengukur jarak suatu objek menggunakan gelombang suara ultrasonik. Sensor ini bekerja dengan mengirimkan gelombang suara dan menghitung waktu pantulan gelombang tersebut kembali ke sensor. Dalam eksperimen ini, dilakukan simulasi menggunakan sensor ultrasonic untuk mengukur jarak pada platform WOKWI, yang diintegrasikan dengan board ESP32. Tujuan dari simulasi ini adalah untuk memahami prinsip kerja sensor ultrasonic, memprogram ESP32 untuk membaca data jarak, dan menampilkan hasil pengukuran jarak secara real-time. Eksperimen ini juga menunjukkan bagaimana teknologi IoT dapat diterapkan untuk memantau data dari sensor secara digital.

Keywords —Sensor Ultrasonic, ESP32, IOT, Pengukuran Jarak, Wokwi

**1. Introduction** (Pendahuluan)

**1.1 Latar belakang**

Dalam dunia teknologi modern, pengukuran jarak memiliki peran penting dalam berbagai aplikasi, seperti robotika, kendaraan otonom, sistem keamanan, dan kontrol industri. Salah satu perangkat yang sering digunakan untuk mengukur jarak adalah sensor ultrasonic. Sensor ini bekerja dengan memanfaatkan gelombang suara ultrasonik untuk mendeteksi keberadaan dan jarak suatu objek dengan akurasi yang tinggi.

ESP32, sebagai salah satu mikrokontroler yang mendukung konektivitas Wi-Fi dan Bluetooth, memberikan peluang untuk mengintegrasikan sensor ultrasonic ke dalam sistem berbasis IoT (Internet of Things). Dengan kemampuan ini, pengukuran jarak tidak hanya dapat dilakukan secara lokal, tetapi juga dapat diakses dan dimonitor secara real-time melalui jaringan.

Praktik simulasi sensor ultrasonic menggunakan platform WOKWI bertujuan untuk mempelajari prinsip kerja sensor, cara memprogram ESP32 untuk membaca data dari sensor, dan bagaimana data tersebut dapat dimanfaatkan dalam aplikasi teknologi berbasis IoT. Melalui simulasi ini, diharapkan peserta dapat memahami implementasi perangkat keras dan perangkat lunak dalam sistem pengukuran jarak secara efektif.

**1.2 Tujuan eksperimen**

Penelitian ini bertujuan :   
1. Memahami prinsip kerja sensor ultrasonic dalam pengukuran jarak menggunakan e=gelombang suara

2. Memprogram ESP32 untuk membaca data jarak yang diperoleh dari sensor ultrasonic

3. Mensimulasikan pengukuran jarak menggunakan platform Wokwi

4. Menampilkan hasil pengukuran jarak secara real-time pada serial monitor atau antarmuka visual

5. Mengaplikasikan konsep IOT untuk memantau data sensor secara digital

**2. Methodology (Metodologi)**

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

Alat dan Bahan yang digunakan :   
1. Mikrokontroler : ESP 32 Devkit VI  
2. Sensor Ultrasonic HC-SR04  
3. Software : Wokwi, Visual Studio Code

**2.2 Implementation Steps (Langkah Implementasi)**

1. Melakukan simulasi di Wokwi untuk simulasi Relay, Button, & LED sebelum ke Visual Studio Code (code editor).

2. Menambahkan ESP32 DevKit V1, HC-SR04 Ultrasonic Distance Sensor lalu dihubungkan sesuai skema yaitu

- pin pada HC GND dipasangkan dengan GND pada ESP 32

- pin pada HC ECHO dipasangkan dengan pin D18 pada ESP 32

- pin pada HC TRIG dipasangkan dengan pin D5 pada ESP 32

- pin pada HC VCC dipasangkan dengan VIN pada ESP 32

3. Menuliskan kode program di Wokwi

4. Setelah berhasil disimulasikan di Wokwi, kode program di unggah ke Visual Studio Code dengan membuat proyek baru di PlatformIO dengan board ESP32 DevKit V1.

5. Menuliskan kode pada main.cpp

#include <Arduino.h>

const int trigPin = 5;

const int echoPin = 18;

//define sound speed in cm/uS

#define SOUND\_SPEED 0.034

#define CM\_TO\_INCH 0.393701

long duration;

float distanceCm;

float distanceInch;

void setup() {

 Serial.begin(115200); // Starts the serial communication

 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

 pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}

void loop() {

 // Clears the trigPin

 digitalWrite(trigPin, LOW);

 delayMicroseconds(2);

 // Sets the trigPin on HIGH state for 10 micro seconds

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds

 duration = pulseIn(echoPin, HIGH);

  // Calculate the distance

 distanceCm = duration \* SOUND\_SPEED/2;

  // Convert to inches

 distanceInch = distanceCm \* CM\_TO\_INCH;

  // Prints the distance in the Serial Monitor

 Serial.print("Distance (cm): ");

 Serial.println(distanceCm);

 // Serial.print("Distance (inch): ");

 // Serial.println(distanceInch);

  delay(1000);

}

6. Membuat file diagram.json dan menuliskan program sama persis dengan yang sudah dituliskan pada simulasi diwokwi.

7. Membuat file wokwi.toml untuk menghubungkan firmware hasil kompilasi ke Wokwi.

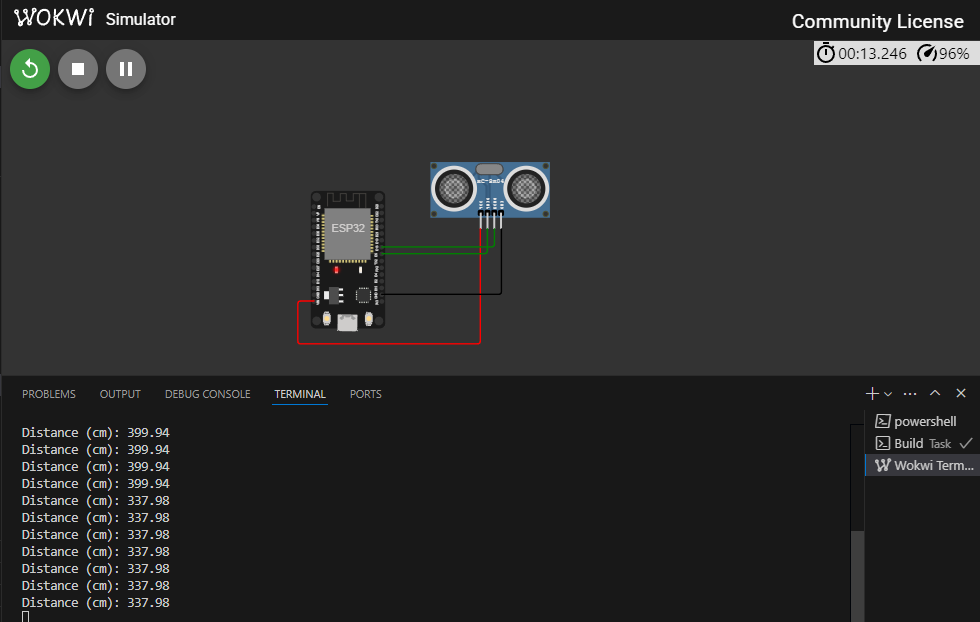
8. Selanjutnya jalankan program dengan memilih build.

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

**3.1 Experimental Results (Hasil Eksperimen)**

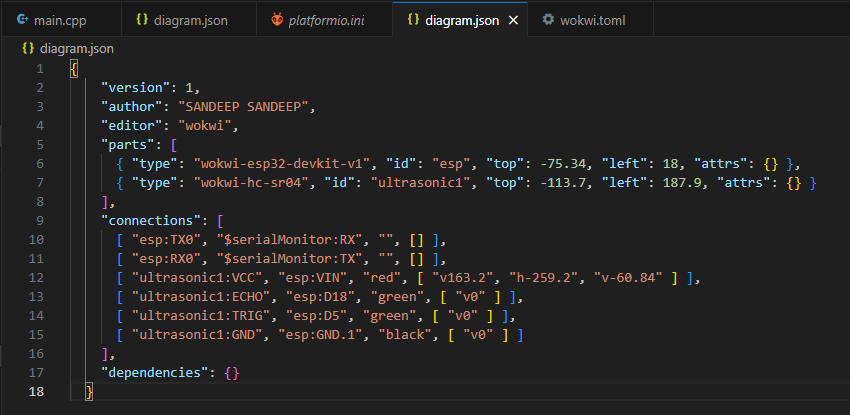
Setelah kode program di unggah di Visual Studio Code, akan memunculkan hasil :   
- Data ukur jarak pada terminal   
- Bisa merubah jarak ketika kita klik HC-SR04

**Screenshoot hasil simulasi pada Visual Studio Code :**



**Pembahasan :   
-** Simulasi berjalan sesuai dengan yang diharapkan.

**4. Appendix (Lampiran, jika diperlukan)**



LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Pembuatan API menggunakan Laravel 11 dan Ngrok (Minggu 5)**

*Fayola Carani Malya  
Fakultas Vokasi, Universitas Brawijaya  
Email: fayolacarani@gmail.com*

**Abstract**   
Pembuatan API (Application Programming Interface) merupakan langkah penting dalam pengembangan aplikasi modern untuk memungkinkan komunikasi antara berbagai sistem atau platform. Laravel, sebagai salah satu framework PHP populer, menawarkan berbagai fitur untuk membangun API dengan cepat, efisien, dan aman. Dalam praktik ini, digunakan Laravel versi 11 untuk membuat API sederhana yang dapat diakses melalui internet menggunakan NGrok, sebuah alat untuk menghubungkan localhost ke jaringan publik.

Tujuan dari eksperimen ini adalah mempelajari proses pengembangan API menggunakan Laravel, memahami penggunaan NGrok untuk mengakses API secara online, dan menguji komunikasi API dengan metode HTTP seperti GET dan POST. Praktik ini menunjukkan cara membangun sistem backend yang dapat diintegrasikan dengan aplikasi frontend atau perangkat IoT.

Keywords —Laravel 11, API, Ngrok, Framework PHP, HTTP Request

**1. Introduction** (Pendahuluan)

**1.1 Latar belakang**

Dalam era digital, komunikasi antara aplikasi menjadi kebutuhan yang esensial untuk membangun sistem yang terintegrasi. API (Application Programming Interface) memainkan peran penting dalam menghubungkan aplikasi backend dengan berbagai platform seperti aplikasi mobile, web, maupun perangkat IoT. Dengan API, pengembang dapat menyediakan layanan yang dapat diakses oleh berbagai sistem dengan cara yang standar dan aman.

Laravel, sebagai salah satu framework PHP yang populer, menawarkan kemudahan dalam membangun API. Versi terbaru Laravel 11 memberikan peningkatan fitur yang mendukung pengembangan API secara cepat dan efisien, termasuk pengelolaan routing, validasi data, dan keamanan endpoint.

Selain itu, pengujian API sering kali memerlukan akses dari jaringan eksternal. NGrok menjadi solusi untuk menjembatani koneksi localhost ke jaringan publik. Dengan NGrok, pengembang dapat menguji API mereka secara online tanpa perlu mengatur server fisik.

Praktik ini bertujuan untuk memberikan pemahaman mendalam mengenai pembuatan API menggunakan Laravel 11, mengintegrasikan NGrok untuk pengujian online, dan mengaplikasikan metode HTTP seperti GET dan POST dalam proses komunikasi data.

**1.2 Tujuan eksperimen**

Penelitian ini bertujuan :   
1. Memahami prinsip kerja relay sebagai saklar elektrik untuk mengendalikan perangkat dengan daya lebih tinggi

2. Mempelajari fungsi button sebagai input mekanis untuk memicu aksi tertentu

3. Mengaplikasikan LED sebagai indicator visual dalam sistem kontrol

4. Mengintegrasikan relay, button, dan LED menggunakan mikrokontroler sebagai pengendali utama

5. Mengeksplorasi potensi aplikasi sistem control ini dalam otomasi rumah dan pengendalian perangkat elektronik  
6. Mengimplementasikan logika control sederhana untuk mengatur hubungan antara relay, button, dan LED

**2. Methodology (Metodologi)**

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

Alat dan Bahan yang digunakan :   
1. Mikrokontroler : ESP 32 Devkit VI  
2. Alat : Button dan Relay Module  
3. LED : Berwarna merah dengan pin GPIO 18  
3. Software : Wokwi, Visual Studio Code

**2.2 Implementation Steps (Langkah Implementasi)**

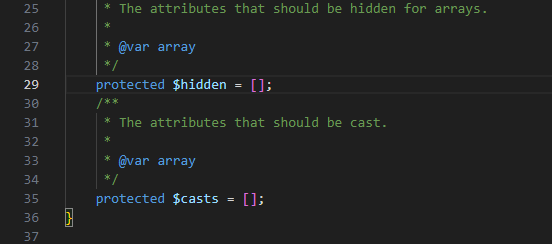
1. Memulai dengan download paket paket pada Laravel 11. Lalu buat file baru dengan mengetikkan kode “composer create-project --prefer-dist laravel/laravel:^11.0 laravel-11”

2. Buat Database dengan nama iot\_25.

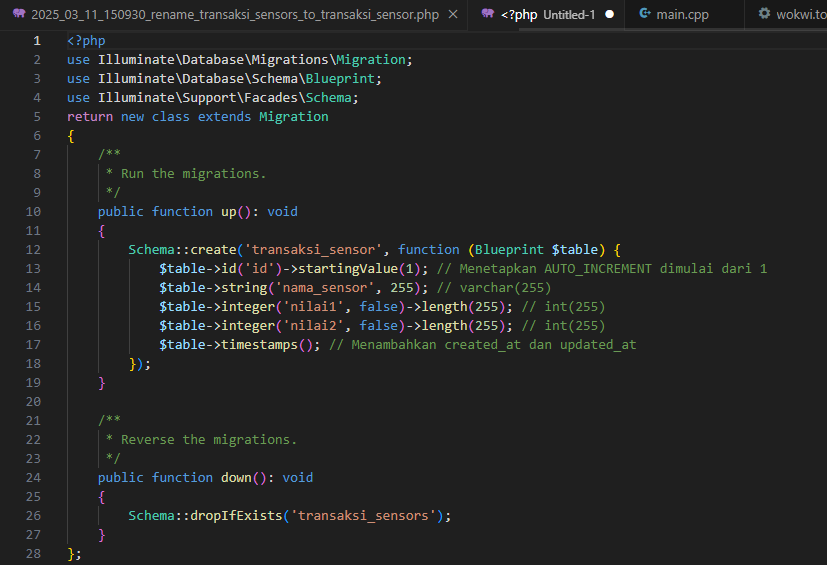
3. Ubah konfigurasi file .env

4. Buat file model bernama “TransaksiSensor.php”. Ubah pula code filenya sesuai denga nada pada modul yaitu





5. Buatlah migrasinya dan ubah kodenya



6. Jalankan perinta “php artisan migrate” untuk membuat table

7. Buat Resource dengan menjalankan perintan “php artisan make:resource TransaksiSensorResource”

8. Ubah juga isi dari file TransaksiSensorResource.php

9. Buat API Controller pada file baru API/TransaksiSensorController. Ubah juga isi dari file tersebut

10. Buat route khusus API dengan perintah “php artisan install:api”

11. Ubah file pada routes/api.php

12. Kemudian jalankan “php artisan route:list” untuk memastikan bahwa routes telah terbentuk

13. Lakukan testing pada postman dan jalankan laravelnya dengan perintah “php artisan serve”

14. Masukkan url alamat server yaitu <http://127.0.0.1:8000/api/posts> dengan method GET.

15. Untuk mencoba kirim data, pada bagian body bisa diubah ke method POST

16. Onlinekan API menggunakan service ngrok sehingga API dapat diakses melalui device iot atau simulasi wokwi IOT

17. jalankan perintah pada terminal NGROK yaitu “ngrok http <http://localhost”8000>”

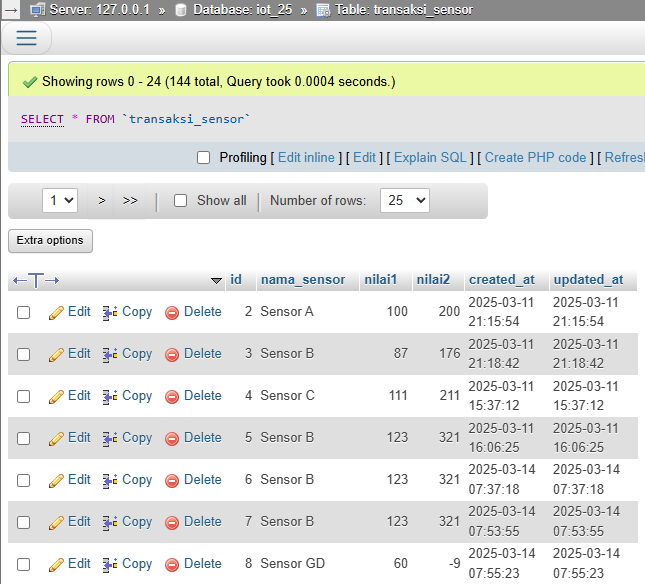
18. Lakukan percobaan menggunakan postman dengan menggunakan URL yang telah diberikan oleh NGROK

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

**3.1 Experimental Results (Hasil Eksperimen)**

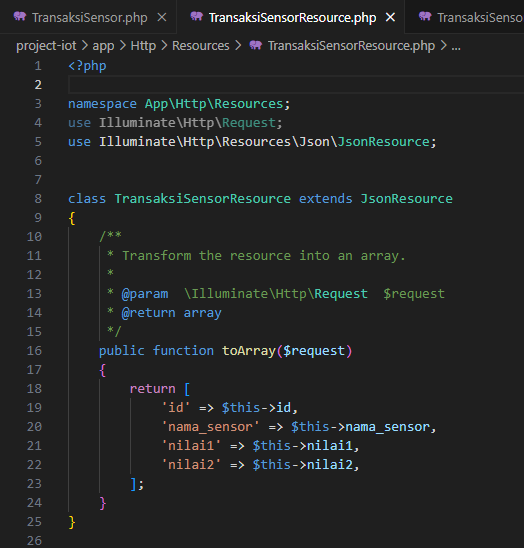
- API yang telah dibuat berhasil dijalankan dengan menggunakan postman.   
- Dapat menjalankan proses GET & POST untuk menambahkan data dan membaca Data  
- Dengan mengguanakan NGROK, memungkinkan API untuk diakses melalui URL Publik.  
- Data juga tersimpan pada database

**Screenshoot hasil simulasi pada Visual Studio Code :**

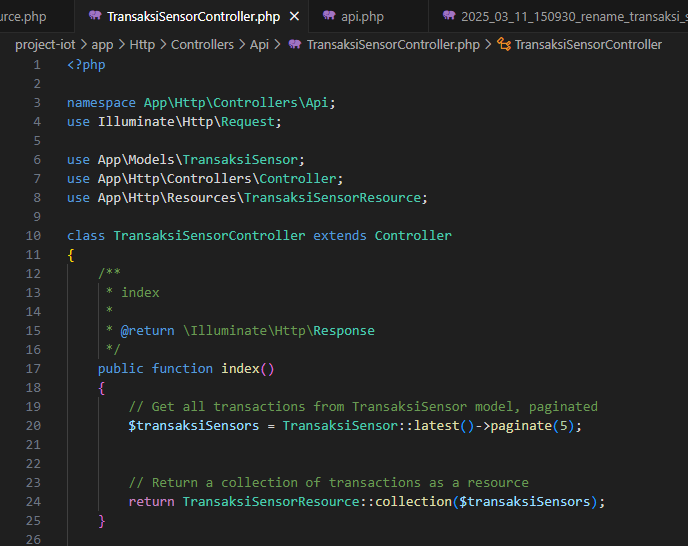
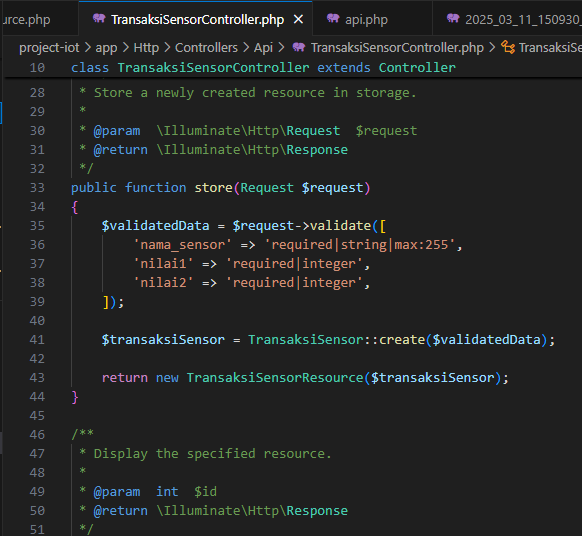


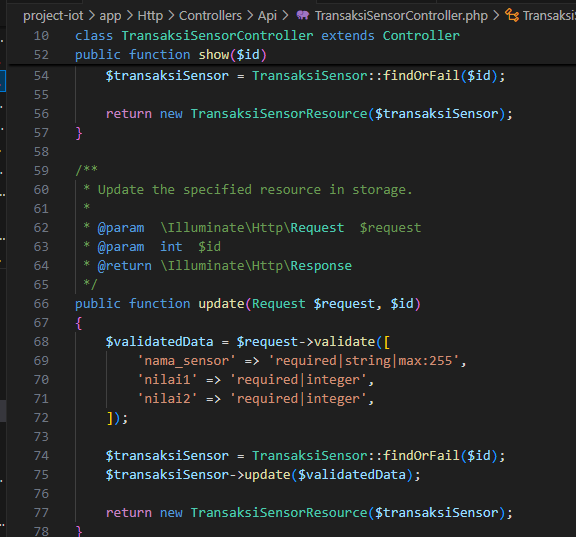
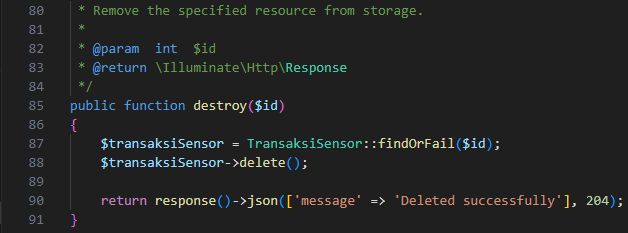
**Pembahasan :   
-** Simulasi berjalan sesuai dengan yang diharapkan  
- Ketika button ditekan, lampu pada relay dan LED akan hidup.

**4. Appendix (Lampiran, jika diperlukan)**

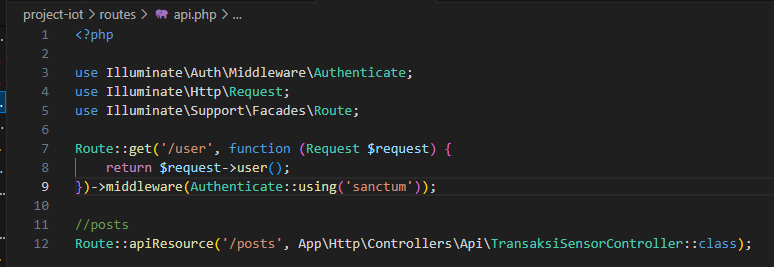
TransaksiSensorResource.php  


TransaksiSensorResource.php

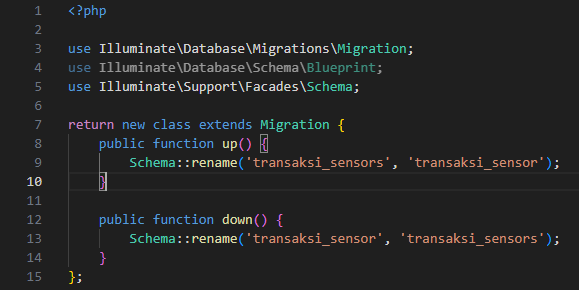
 

Api.php



2025\_03\_11\_150930\_rename\_transaksi\_sensors\_to\_transaksi\_sensor.php

****