**LAPORAN PRAKTIKUM**

**INTERNET OF THINGS**

****

**Disusun Oleh:**

Mirza Marwa Rosyidah (233140700111068)

**PRODI D-III TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Relay, Button & LED**

*Mirza Marwa Rosyidah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: mirzamarwa76@gmail.com*

**Abstract**

Pada system IoT (*Internet of Things)* penggunaan relay, tombol, dan lampu LED dalah komponen utama dalam mengontrol perangkat elektronik. Tombol digunakan sebagai alat untuk menginput sehingga dapat memungkinkan interksi secara manual, sedangkan LED sebagai indikator memberikan umpan balik visual tentang status. Relay berfungsi sebagai saklar yang dapat membuat mikontroler untuk dapat mengontrol perangkat berdaya tinggi dengan sinyal berdaya rendah. Tujuan dari praktik ini adlah untuk ini adlah untuk dapat memperoleh pemahaman dan juga dapat menginplementasikan simulasi relay, tombol dan LED dalam system IoT melalui wokwi dan vscode.

Kata Kunci : *Internet of Things, Tombol, Relay, LED, Wokwi, Vscode*

**1. Introduction**

* 1. **Latar belakang**

Penggunaan relay, tombol, dan LED dalam sistem Internet of Things (IoT) sangat penting untuk mengontrol perangkat elektronik. Tombol sebagai input memungkinkan interaksi manual, sementara LED sebagai indikator memberikan umpan balik visual tentang status sistem. Selain itu, relay berfungsi sebagai saklar yang memungkinkan mikrokontroler untuk mengontrol perangkat berdaya tinggi dengan sinyal berdaya rendah. Tujuan dari praktik ini adalah untuk memperoleh pemahaman dan implementasi simulasi relay, tombol, dan LED dalam sistem Internet of Things (IoT).

**1.2 Tujuan eksperimen**

Penelitian ini bertujuan untuk,

1. Memahami prinsip kerja relay, tombol, dan LED dalam sistem IoT.
2. Mengimplementasikan simulasi menggunakan Wokwi dan ESP32.
3. Menganalisis respons sistem terhadap input dari tombol.

**2. Methodology (Metodologi)**

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

Alat dan bahan yang digunakan dalam praktikum ini:

1. Mikrokontroler: ESP32
2. Komponen: Relay 5V, Push Button, LED, Resistor
3. Software: Wokwi, VS Code

**2.2 Implementation Steps (Langkah Implementasi)**

1. Menyusun rangkaian dengan ESP32, relay, tombol, dan LED pada wokwi
2. Menulis program Arduino untuk mengontrol relay berdasarkan input dari tombol.
3. Create new project pada PlatformIO di vscode
4. Mengunggah kode ke VSCode
5. Request a new license untuk menghubungkan simulasi rangkaian yang telah dibuat pada wokwi
6. Mengamati hasilnya simulasi

Kode yang digunakan,

#include <Arduino.h>

// Define pin numbers

const int ButtonPin = 19; // GPIO19 connected to the pushbutton

const int LedPin = 18;    // GPIO18 connected to the LED

const int RelayPin = 23;  // GPIO23 connected to the relay module

void setup()

{

  // Set pin modes

  pinMode(ButtonPin, INPUT\_PULLUP); // Set the button pin as an input with an internal pull-up resistor

  pinMode(LedPin, OUTPUT);          // Set the LED pin as an output

  pinMode(RelayPin, OUTPUT);        // Set the relay pin as an output

  // Initialize the outputs to be OFF

  digitalWrite(LedPin, LOW);

  digitalWrite(RelayPin, LOW);

}

void loop()

{

  // Read the state of the button

  int buttonState = digitalRead(ButtonPin);

  // Check if the button is pressed

  // Since the button is wired to pull the pin LOW when pressed, we check for LOW

  if (buttonState == LOW)

  {

    digitalWrite(LedPin, HIGH);   // Turn on the LED

    digitalWrite(RelayPin, HIGH); // Turn on the relay

  }

  else

  {

    digitalWrite(LedPin, LOW);   // Turn off the LED

    digitalWrite(RelayPin, LOW); // Turn off the relay

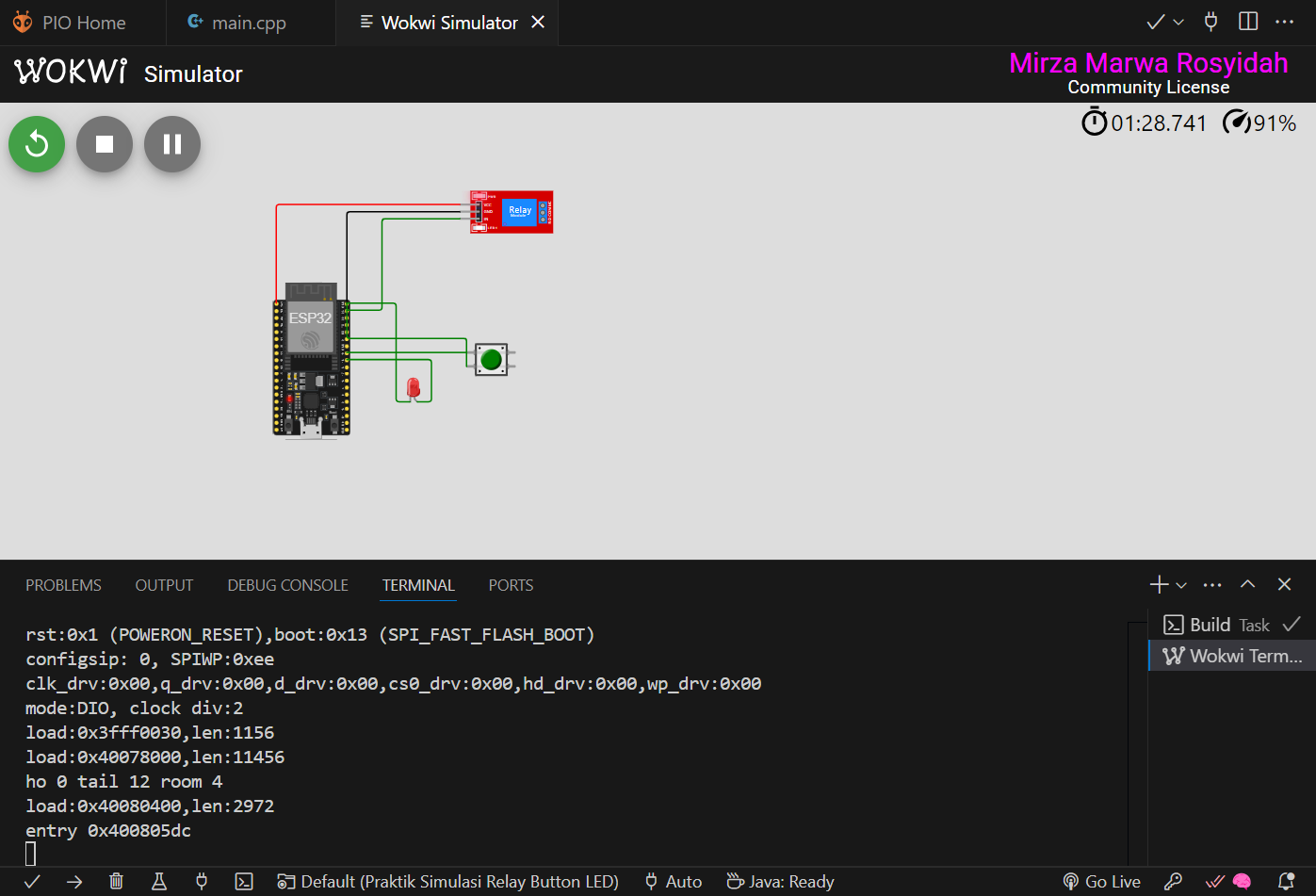
  }

}

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

**3.1 Experimental Results (Hasil Eksperimen)**

Hasil simulasi menunjukkan bahwa sistem dapat merespons input dari tombol dengan mengaktifkan atau menonaktifkan relay yang kemudian menyalakan atau mematikan LED. Jika tombol dikean maka LED akan menyala dan status telay aktif sedangkan saat tombol tidak ditekan maka LED akan mati dan status relay menjadi nonaktif.

**Screenshoot hasil simulasi:**  


**Pembahasan:**

Jika tombol dikean maka LED akan menyala dan status telay aktif sedangkan saat tombol tidak ditekan maka LED akan mati dan status relay menjadi nonaktif.

**4. Appendix**

Kode pemrograman Diagram

{

    "version": 1,

    "author": "Mirza Marwa Rosyidah",

    "editor": "wokwi",

    "parts": [

        {

            "type": "board-esp32-devkit-c-v4",

            "id": "esp",

            "top": 19.2,

            "left": -4.76,

            "attrs": {}

        },

        {

            "type": "wokwi-relay-module",

            "id": "relay1",

            "top": -105.4,

            "left": 249.6,

            "attrs": {}

        },

        {

            "type": "wokwi-pushbutton",

            "id": "btn1",

            "top": 102.2,

            "left": 259.2,

            "attrs": {

                "color": "green",

                "xray": "1"

            }

        },

        {

            "type": "wokwi-led",

            "id": "led1",

            "top": 140.4,

            "left": 167,

            "attrs": {

                "color": "red"

            }

        }

    ],

    "connections": [

        [

            "esp:TX",

            "$serialMonitor:RX",

            "",

            []

        ],

        [

            "esp:RX",

            "$serialMonitor:TX",

            "",

            []

        ],

        [

            "relay1:VCC",

            "esp:3V3",

            "red",

            [

                "h0"

            ]

        ],

        [

            "relay1:GND",

            "esp:GND.2",

            "black",

            [

                "h0"

            ]

        ],

        [

            "relay1:IN",

            "esp:23",

            "green",

            [

                "h-105.6",

                "v124.6"

            ]

        ],

        [

            "btn1:1.l",

            "esp:19",

            "green",

            [

                "h-115.2",

                "v48"

            ]

        ],

        [

            "btn1:2.l",

            "esp:GND.2",

            "green",

            [

                "v-38.2",

                "h-163.2"

            ]

        ],

        [

            "led1:A",

            "esp:18",

            "green",

            [

                "h19.2",

                "v-57.6"

            ]

        ],

        [

            "led1:C",

            "esp:GND.2",

            "green",

            [

                "h-18.8",

                "v-134.4"

            ]

        ]

    ],

    "dependencies": {}

}

1. **Kesimpulan**Sistem berbasis ESP32 dapat digunakan untuk mengontrol relay, tombol, dan LED dengan respons yang baik. Wokwi berguna sebagai media pengujian dan debugging sebelum paraktik secara nyata menugunakan alat nantinya. Simulasi yang telah dilakukan menunjukkan bagaimana relay, tombol, dan LED dapat digunakan dalam sistem IoT untuk mengontrol perangkat elektronik .

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Sensor Jarak (Ultrasonic)**

*Mirza Marwa Rosyidah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: mirzamarwa76@gmail.com*

**Abstract**

Sensor jarak ultrasonic adalah termasuk bagian dari komponen utama dalam sistem IoT yang digunakan untuk mengukur jarak pada objek. Sensor ini mengukur waktu dari pancaran gelombang ultrasonic, cara kerjanya gelombang dapat diukur memalui kapan waktunya gelombang tersenut terpantul Kembali pada objek. Praktikum simulasi ini bertujuan untuk memahami prinsip kerja sensor HC-SR04 yang dihubungkan dengan ESP32 serta menganalisis data jarak yang dihasilkan melalui simulasi di Wokwi, memalui Vscode. Hasil eksperimen nantinya menunjukkan bahwa sensor mampu memberikan pembacaan jarak secara real-time, namun dengan beberapa batasan terkait akurasi dan respons lingkungan.

Kata Kunci : *Internet of Things, ESP32, Ultrasonik, HC-SR04, Wokwi, Vscode*

**1. Introduction**

* 1. **Latar belakang**

Sensor jarak ultrasonik HC-SR04 banyak digunakan dalam berbagai aplikasi IoT, seperti robotika, sistem keamanan, dan otomatisasi industri. Sensor ini bekerja dengan mengirimkan gelombang ultrasonik melalui transduser dan mengukur waktu tempuh pantulannya untuk menentukan jarak. Praktik simulasi ini bertujuan untuk memahami bagaimana sensor ini bekerja dan mengimplementasikan simulasi menggunakan ESP32 dalam Wokwi.

**1.2 Tujuan eksperimen**

Penelitian ini bertujuan untuk,

1. Mengimplementasikan simulasi menggunakan Wokwi dan ESP32.
2. Mengimplementasikan simulasi wokwi pada Vscode
3. Menganalalisis prinsip kerja HC-SR04

**2. Methodology (Metodologi)**

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

Alat dan bahan yang digunakan dalam praktikum ini:

1. Mikrokontroler: ESP32
2. HC-SR04 (Sensor Jarak Ultrasonik)
3. Software: Wokwi, VScode

**2.2 Implementation Steps (Langkah Implementasi)**

1. Menyusun rangkaian dengan SP32 dan HC-SR04 pada wokwi
2. Menulis program Arduino untuk membaca data jarak sensor HC-SR04
3. Create new project pada PlatformIO di vscode
4. Mengunggah kode ke VSCode
5. Request a new license untuk menghubungkan simulasi rangkaian yang telah dibuat pada wokwi
6. Mengamati hasilnya simulasi

Kode yang digunakan,

# #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);

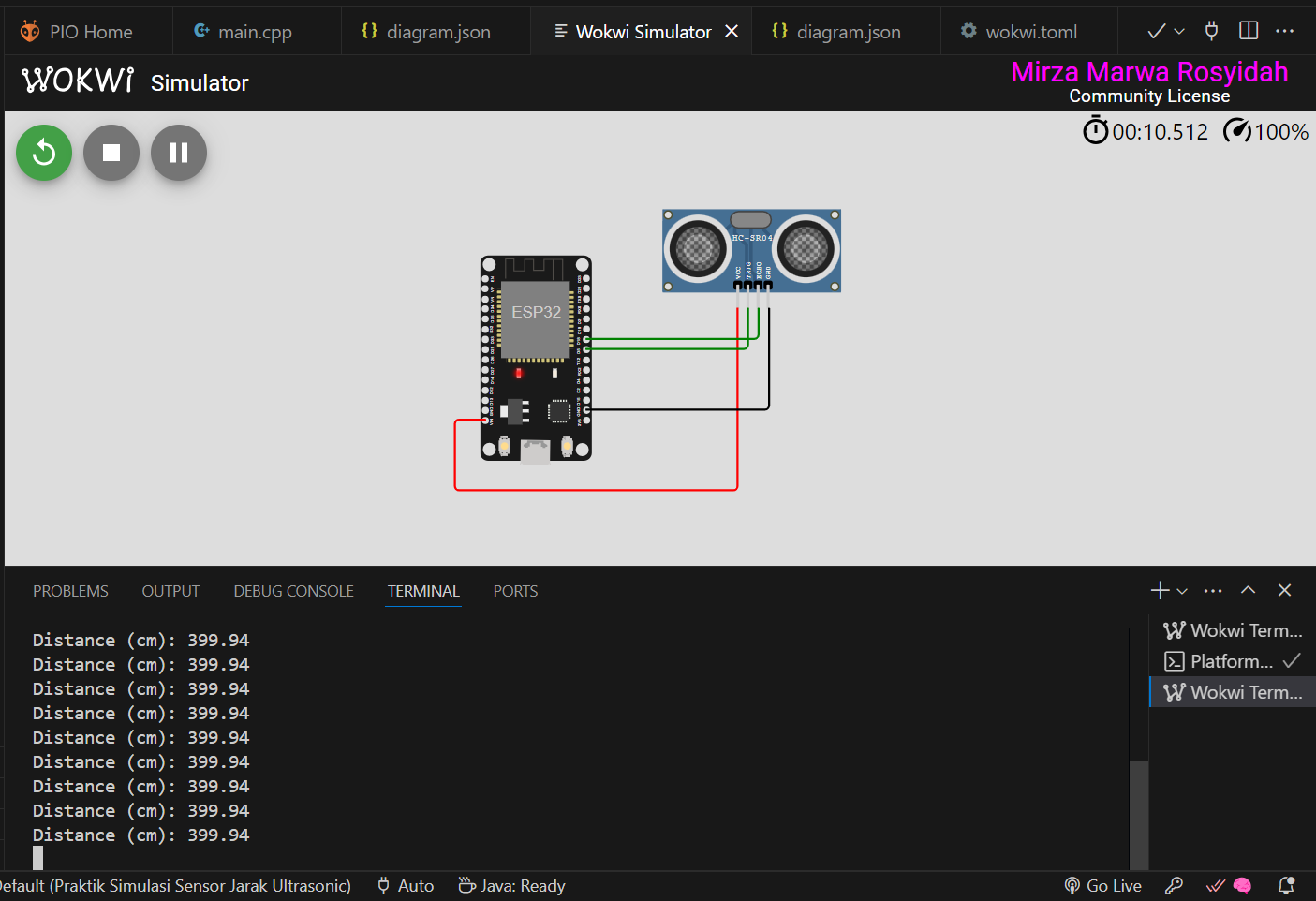
  delay(1000);

}

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

**3.1 Experimental Results (Hasil Eksperimen)**

Hasil simulasi menunjukkan bahwa simulasi sensor HC-SR04 pada vscode berhasil membaca jarak 399.94 cm secara konsisten,

**Screenshoot hasil simulasi:**  


**Pembahasan:**

Jika Ultrasonic Distance Sensor nya diedit maka jarak yang terbaca oleh sensor HC-SR04 juga akan ikut berubah menyusaikan

**4. Appendix**

Kode pemrograman Diagram

{

    "version": 1,

    "author": "programador novato",

    "editor": "wokwi",

    "parts": [

        {

            "type": "wokwi-esp32-devkit-v1",

            "id": "esp",

            "top": -4.9,

            "left": -91.4,

            "attrs": {}

        },

        {

            "type": "wokwi-hc-sr04",

            "id": "ultrasonic1",

            "top": -46.5,

            "left": 82.3,

            "attrs": {}

        }

    ],

    "connections": [

        [

            "esp:TX0",

            "$serialMonitor:RX",

            "",

            []

        ],

        [

            "esp:RX0",

            "$serialMonitor:TX",

            "",

            []

        ],

        [

            "ultrasonic1:VCC",

            "esp:VIN",

            "red",

            [

                "v172.8",

                "h-268.8",

                "v-67.2"

            ]

        ],

        [

            "ultrasonic1:GND",

            "esp:GND.1",

            "black",

            [

                "v0"

            ]

        ],

        [

            "ultrasonic1:TRIG",

            "esp:D5",

            "green",

            [

                "v0"

            ]

        ],

        [

            "ultrasonic1:ECHO",

            "esp:D18",

            "green",

            [

                "v0"

            ]

        ]

    ],

    "dependencies": {}

}

1. **Kesimpulan**Paraktik simulasi ini menunjukkan bagaimana sensor jarak ultrasonic dapat digunaakan pada system IOT. Sensor HC-SR04 dapat digunakan untuk mengukur jarak secara real-time dengan ESP32. Hasil simulasi menunjukkan jarak yang terbaca oleh sendor dipengaruhi oleh parameter menunjukkan sensor memiliki respon terhadap perubahan. Simulasi ini dapat dikembangkan atau diimplementasikan untuk aplikasi system parkir otomatis dan lain sebagainya.

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Pembuatan API Menggunakan Laravel 11 dan Ngrok**

*Mirza Marwa Rosyidah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: mirzamarwa76@gmail.com*

**Abstract**

Praktikum ini bertujuan untuk memahami cara membuat dan mengelola API menggunakan framework Laravelserta menghubungkannya dengan internet menggunakan layanan Ngrok. Implementasi API dilakukan dengan pembuatan model, migrasi database, controller, dan routing dalam Laravel. Hasil eksperimen menunjukkan bahwa API yang dikembangkan dapat diakses secara lokal maupun publik menggunakan Ngrok, serta mampu melakukan operasi CRUD (Create, Read, Update, Delete) melalui Postman. Kesimpulan dari eksperimen ini adalah Laravel menyediakan fitur yang mudah digunakan untuk pembuatan API dan dapat dengan mudah diintegrasikan dengan layanan lain seperti Postman dan Ngrok untuk pengujian.

Kata Kunci : *API, Laravel, Ngrok, IoT, CRUD*

**1. Introduction**

* 1. **Latar belakang**

Internet of Things (IoT) semakin berkembang dengan kebutuhan integrasi data yang cepat dan efisien. Salah satu cara untuk menghubungkan perangkat IoT dengan server adalah melalui API berbasis RESTful yang dikembangkan dengan framework Laravel. Laravel menyediakan berbagai fitur salah satunya mempermudah pembuatan API.

**1.2 Tujuan eksperimen**

Penelitian ini bertujuan untuk,

1. Membuat API menggunakan Laravel
2. Menghubungkan API ke database MySQL.
3. Mengakses API melalui Postman.
4. Mengonlinekan API menggunakan Ngrok

**2. Methodology (Metodologi)**

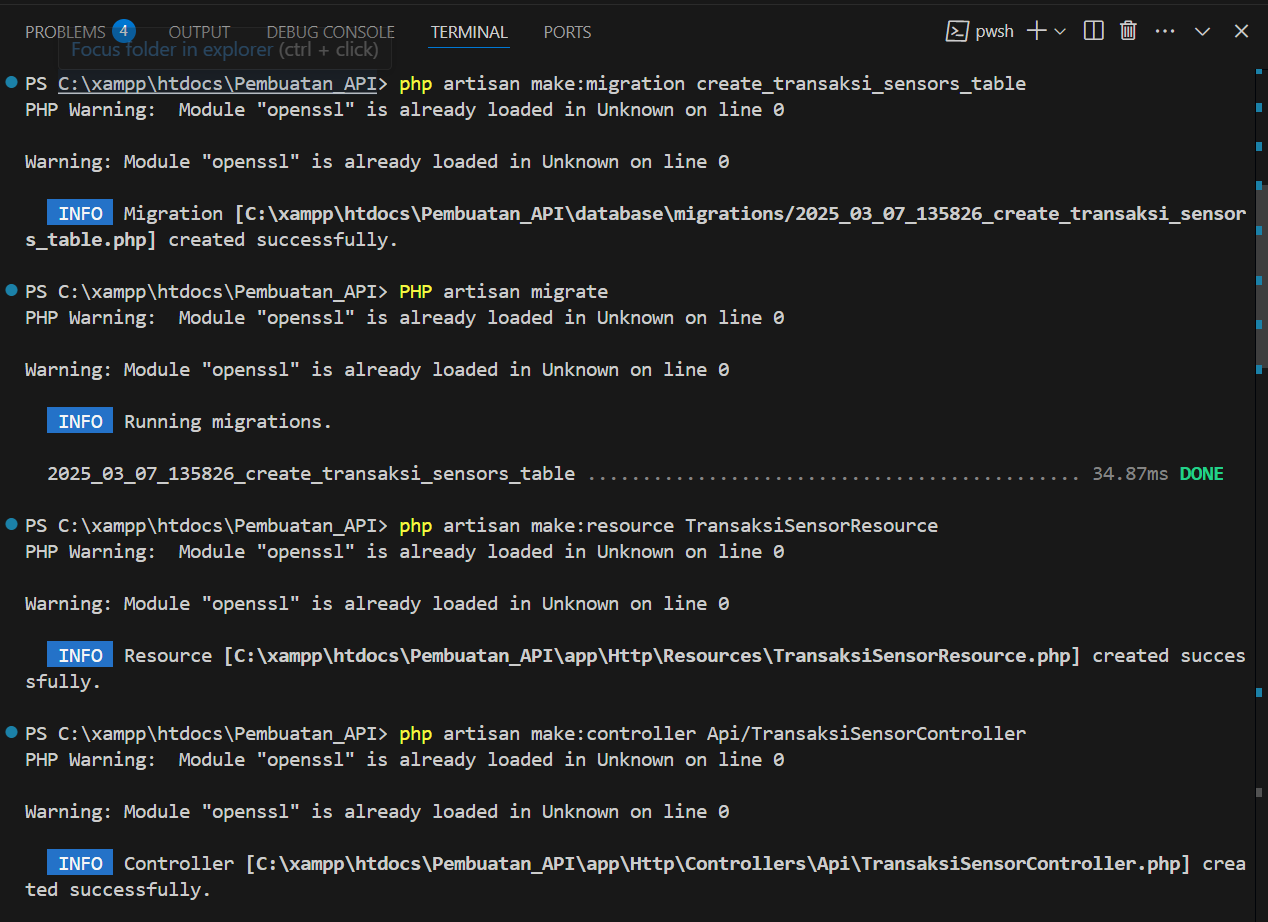
**2.1 Tools & Materials (Alat dan Bahan)**

Alat dan bahan yang digunakan dalam praktikum ini:

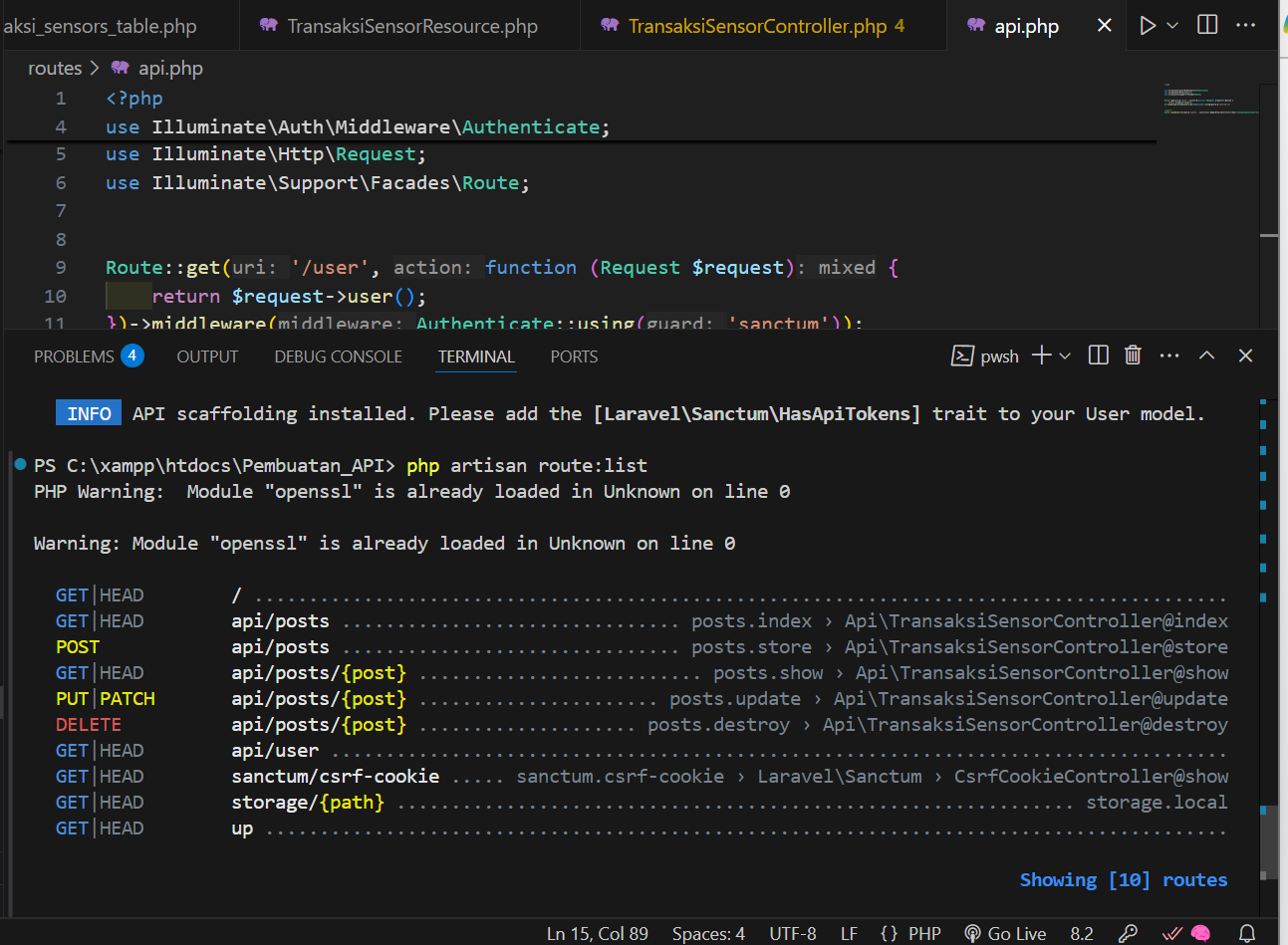
1. Laravel
2. Composer
3. PHP versi 8+
4. Mysql
5. Postman
6. Ngrok

**2.2 Implementation Steps (Langkah Implementasi)**

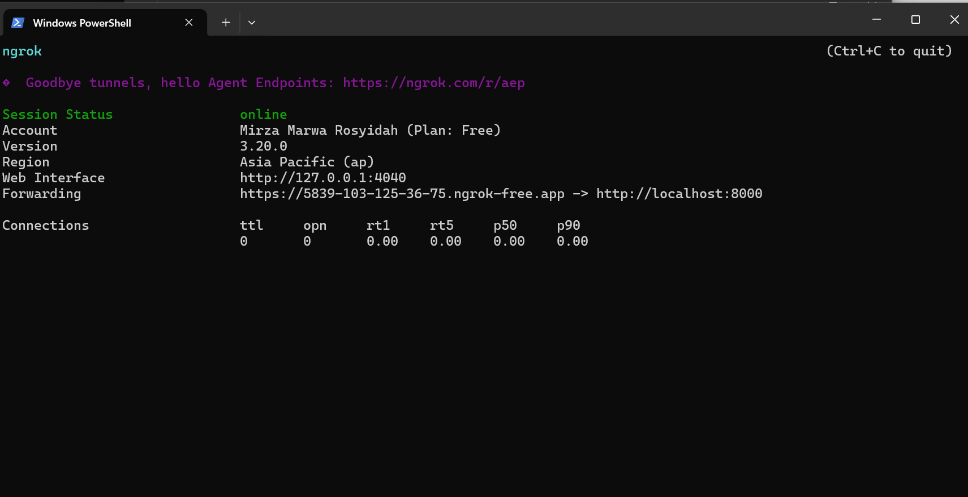
1. Create new projek Laravel memalui cmd/ terminal Vscode
2. Menyalakan control Xampp, membuat data base di phpMyAdmin dengan nama iot\_25
3. Buka projek Laravel yang telah dibuat, buka file .env dan ubah konfigurasi database
4. Make migration dan mengubah codingannya sesuai dengan modul
5. menjalankan semua file migrasi yang ada di dalam folder migration “php artisan migration”
6. make resource dan ubah codingan didalam file suai dengan modul
7. buat API controller dan ubah juga codingannya susai dengan modul



1. Intall API melalui terminal, kemudian file API akan terbentuk pada folder routes
2. Masukkan file codingan pada file API seperti yang tertera pada modul
3. Cek route list nya dan pastikan tampilannya sesuai dengan modul



1. Download postman
2. Jalankan aplikasi Laravel “php artisan serve”
3. Bukak phpMyAdmin bagian table transaksi\_sensor pada database IoT isikan data secara manual
4. Jika sudah buka postman Pada bagian URL masukkan alamat server Laravel <http://127.0.0.1:8000/api/posts> atau bisa diakses melalui url : http://localhost:8000/api/posts
5. Pilih method GET untuk mengambil data dari database, kemudian klik tombol SEND
6. Data akan muncul dalam bentuk json di postman
7. insert data ke tabel di database menggunakan API. Pilih metode POST, pada bagian header kolom key “Content\_Type” pada kolom value “application/json”. Pada bagian body, pilih raw dan ubah ke json
8. masukkan data dengan format sesuai pada modul, kemudian kil send. Jika data berhasil masuk maka tampilan nya akan keluar pada terminal postman dan pada phpmyadmin
9. Login ke web ngrok dan download ngrok. Ektraksi ngrok
10. Buka CMD dari alamat folder ekstraksi
11. Jalankan perintah sesuai pada akun ngrok di CMD. Maka akan muncul URL.



1. Salin URL public yang diberikan grok dan masukkan kedalam postman
2. Ulangi perintah GET dan POST pada postman seperti tadi menggunakan URL ngrok

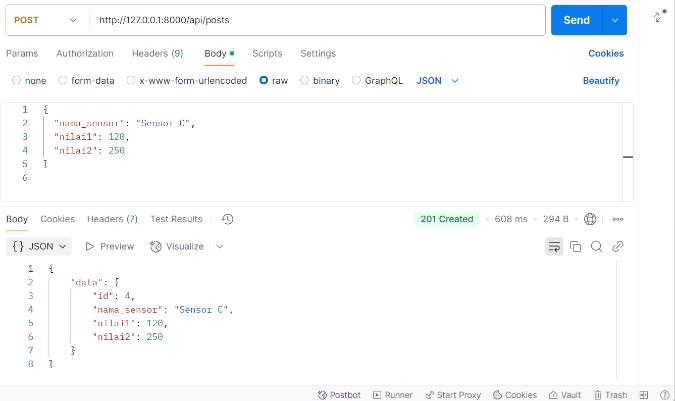
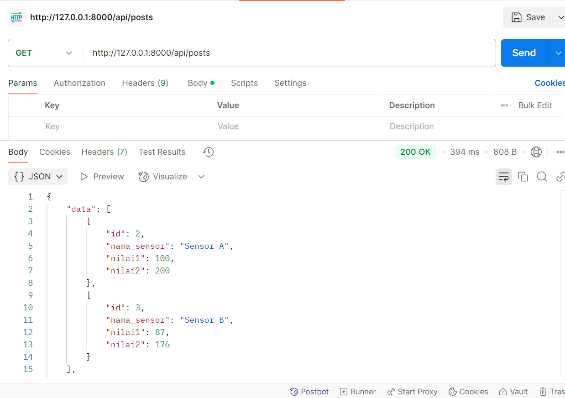
**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

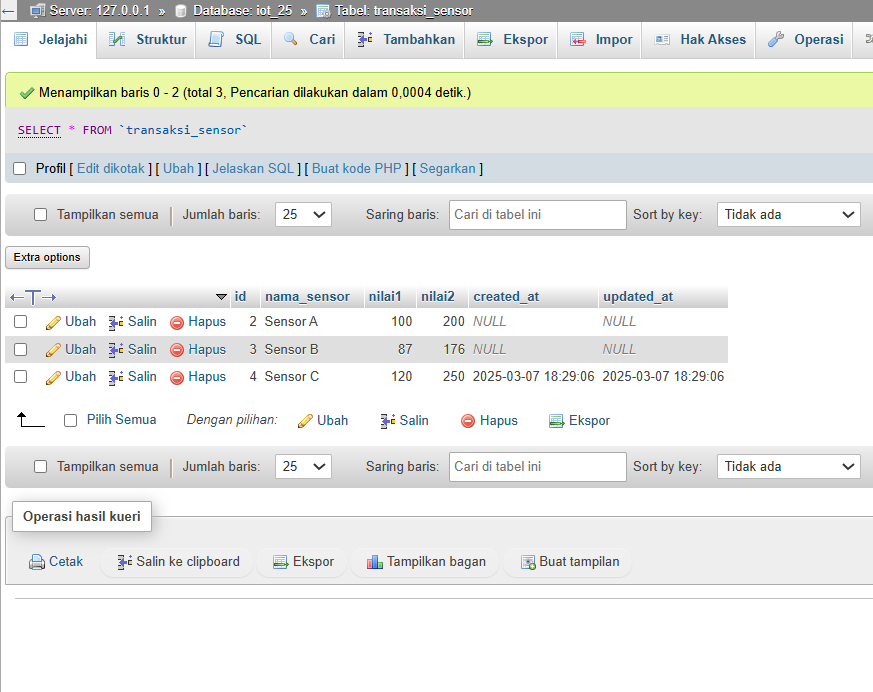
Eksperimen menunjukkan bahwa API yang dikembangkan berhasil diuji menggunakan Postman. Operasi CRUD (POST dan GET) dapat dijalanka. Selain itu, penggunaan Ngrok memungkinkan API untuk diakses melalui URL publik, sehingga memudahkan pengujian dan implementasi di lingkungan yang lebih luas. Seluruh data yang dikirim dan diterima dikonversi dalam format JSON.

**Screenshoot hasil simulasi:**

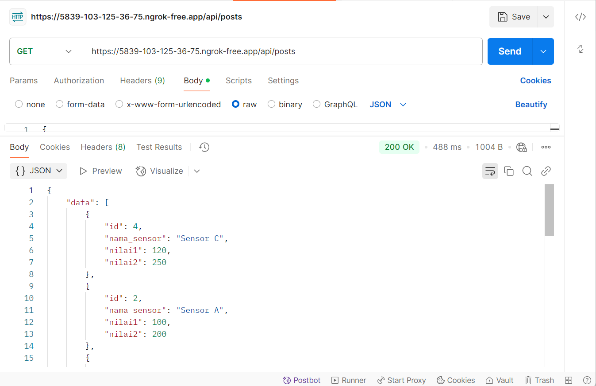
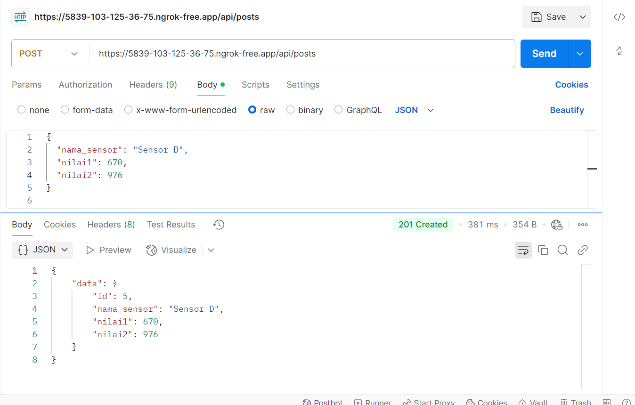
1. Menampilkan data dan menambahkan data pada postman menggunakan URL alamat laravel



Tampilan database pada phpMyAdmin setelah ditambahkan data memalui postman



1. Menampilkan data dan menambahkan data pada postman menggunakan URL alamat ngrok



Tampilan database pada phpMyAdmin setelah ditambahkan data memalui postman

**Pembahasan:**

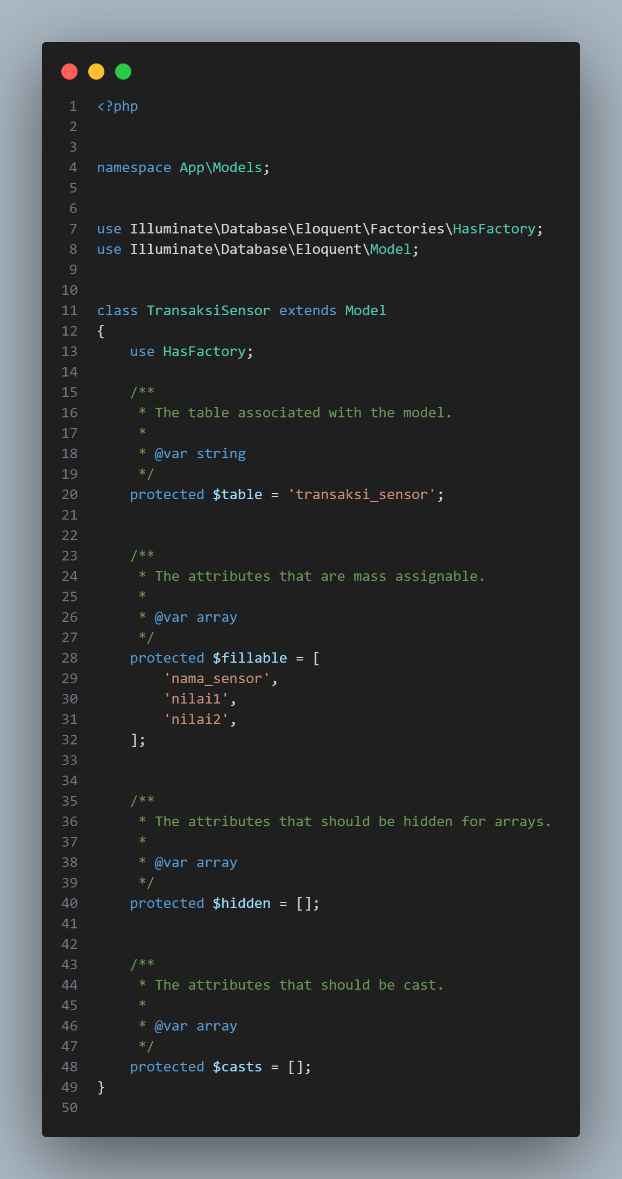
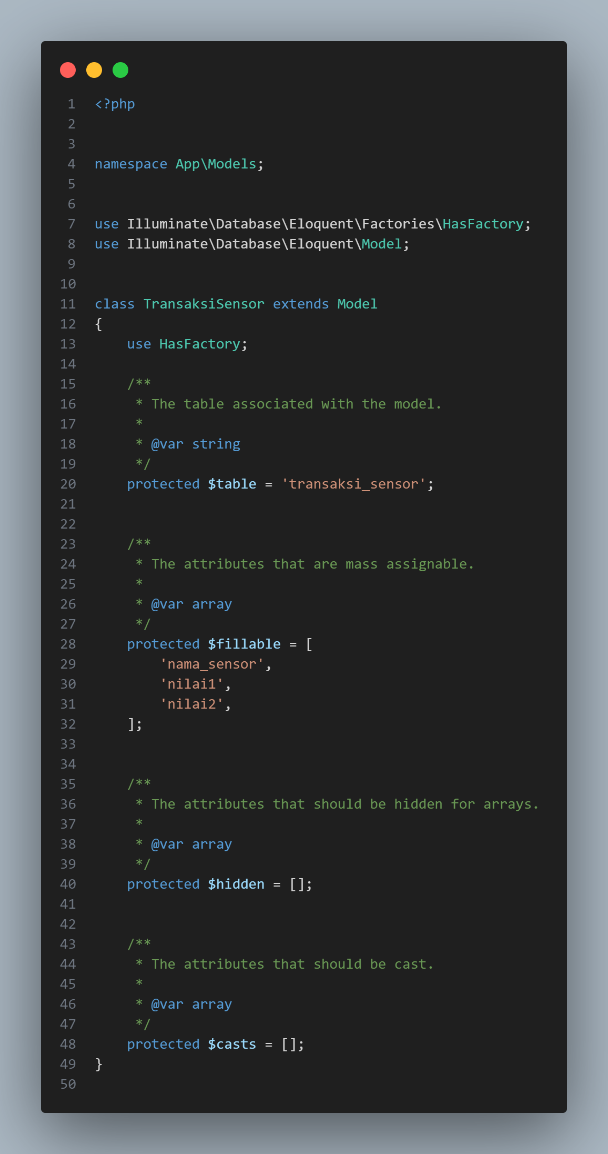
Jika method GET digunakan maka postman akan menambilkan data dan jika jika menggunakan method POST dan mengubah format header dan body sesuai ketentuan makan postman dapat menambahkan data pada database.

**4. Appendix**

Kode pemrograman 2025\_03\_07\_153007\_create\_transaksi\_sensors\_table



Kode pemrograman app/Models/TransaksiSensor.php

Kode pemrograman transaksiSensorResource.php



Kode pemrograman app/Http/Controllers/Api/TransaksiSensorController.php

Kode pemrograman routes/api.php

