

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

**Praktik Simulasi Sensor Jarak (Ultrasonic) menggunakan Wokwi dan
PlatformIO IDE pada VS Code**

*Lucky Ardiansyah 233140701111015
Fakultas Vokasi, Universitas Brawijaya
Email: luckyardiansyah685@gmail.com*

Abstract (Abstrak)

Praktikum ini bertujuan untuk mensimulasikan dan menguji kinerja sensor jarak ultrasonik menggunakan Wokwi serta mengimplementasikannya melalui PlatformIO IDE pada Visual Studio Code. Sensor ultrasonik bekerja berdasarkan prinsip pengukuran waktu tempuh gelombang suara (echo) untuk menghitung jarak suatu objek. Dalam praktikum ini, rangkaian virtual dibuat di Wokwi untuk menghubungkan sensor HC-SR04 dengan mikrokontroler Arduino Uno. Kode program dikembangkan menggunakan PlatformIO IDE, yang mempermudah manajemen dependensi dan kompilasi kode. Hasil simulasi menunjukkan bahwa sensor mampu mengukur jarak objek dengan tingkat akurasi yang baik, sesuai dengan prinsip kerja dan persamaan fisika terkait. Praktikum ini memberikan wawasan tentang integrasi sensor ultrasonik dalam sistem embedded serta penggunaan alat bantu simulasi untuk pengembangan dan pengujian perangkat keras secara virtual sebelum implementasi fisik.

Kata Kunci: Sensor Ultrasonik, HC-SR04, Wokwi, PlatformIO, Visual Studio Code, Arduino Uno

1. Introduction (Pendahuluan)

1.1 Latar Belakang

Sensor ultrasonik merupakan salah satu komponen penting dalam teknologi pengukuran jarak yang banyak digunakan pada berbagai aplikasi, seperti robotika, sistem keamanan, dan otomatisasi. Sensor ini bekerja dengan memanfaatkan prinsip pantulan gelombang suara ultrasonik untuk menentukan jarak antara sensor dan objek. Salah satu jenis sensor ultrasonik yang umum digunakan adalah HC-SR04. Dalam pengembangan sistem berbasis mikrokontroler, simulasi rangkaian menjadi tahap awal yang penting sebelum implementasi perangkat keras secara langsung. Simulasi memungkinkan pengujian logika program, pengaturan koneksi perangkat, serta identifikasi potensi kesalahan tanpa risiko kerusakan pada komponen fisik. Praktikum ini memanfaatkan Wokwi sebagai platform simulasi untuk merancang dan menguji rangkaian sensor ultrasonik, serta PlatformIO IDE pada Visual Studio Code untuk mengembangkan dan mengelola kode program. Dengan kombinasi ini, praktikan dapat memahami konsep pengukuran jarak menggunakan sensor ultrasonik dan memperdalam keterampilan pemrograman mikrokontroler secara efisien.

1.2 Tujuan Eksperimen

- Memahami prinsip kerja sensor ultrasonik HC-SR04 dalam mengukur jarak.
- Melakukan simulasi rangkaian sensor ultrasonik menggunakan Wokwi.
- Mengembangkan kode program mikrokontroler untuk membaca data jarak dari sensor menggunakan PlatformIO IDE pada Visual Studio Code.
- Menguji hasil simulasi untuk memastikan sensor mampu mengukur jarak secara akurat dan logis.
- Meningkatkan keterampilan dalam menggunakan alat bantu simulasi dan pengembangan perangkat lunak embedded system.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat & Bahan)

- **Wokwi Simulator**
- **Visual Studio Code (VS Code)**
- **Ekstensi Wokwi untuk VS Code**
- **Arduino IDE**
- **ESP32 DevKit V1**
- **Relay Module**

- **Push Button**
- **LED**
- **Kabel Virtual**

2.1 Implementation Steps (Langkah Implementasi)

1. Membuat Simulasi di Wokwi

- Buka [Wokwi](#) dan klik **New Project**.
- Pilih **ESP32 DevKit v1** sebagai mikrokontroler yang digunakan.
- Tambahkan sensor ultrasonik **HC-SR04** ke dalam project.
- Lakukan koneksi perangkat sesuai dengan file **diagram.json**, yaitu:
- **GND** sensor ke **GND** ESP32
- **VCC** sensor ke **3V3** ESP32
- **TRIG** sensor ke **GPIO 5** ESP32
- **ECHO** sensor ke **GPIO 18** ESP32

2. Integrasi dengan PlatformIO IDE pada Visual Studio Code

- Buka **Visual Studio Code**.
- Pastikan **PlatformIO IDE** telah terinstal. Jika belum, instal melalui **Extensions**.
- Buat proyek baru dengan klik **PlatformIO: New Project**.
- Atur konfigurasi proyek:
 1. **Board:** ESP32 Devkit
 2. **Framework:** Arduino
- Buka folder proyek, lalu salin kode program berikut ke file **src/main.cpp**:

```
#include <Arduino.h>

const int trigPin = 5;
const int echoPin = 18;

#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701

long duration;
float distanceCm;
float distanceInch;

void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distanceCm = duration * SOUND_SPEED / 2;
```

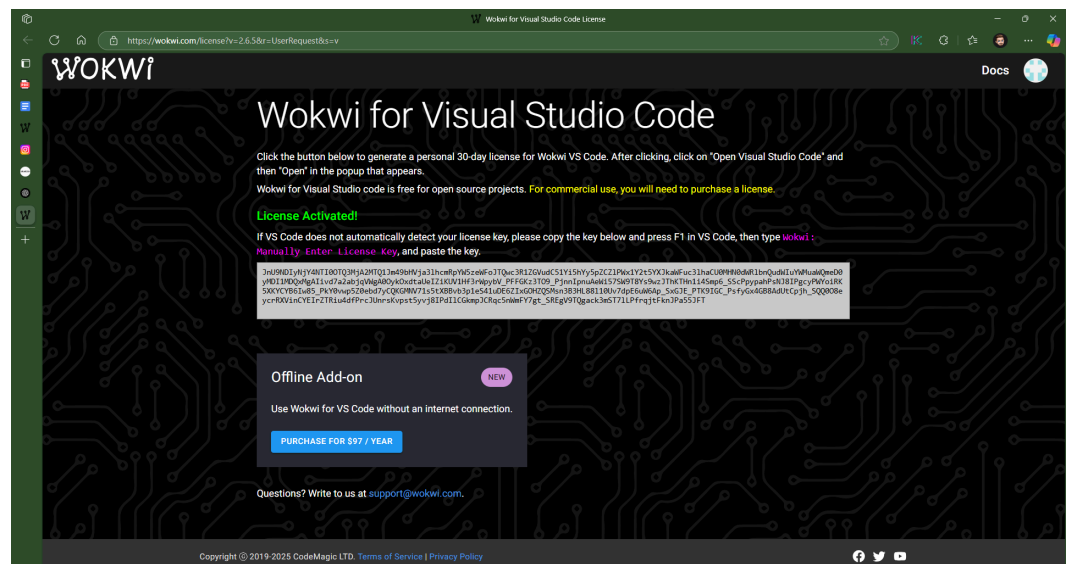
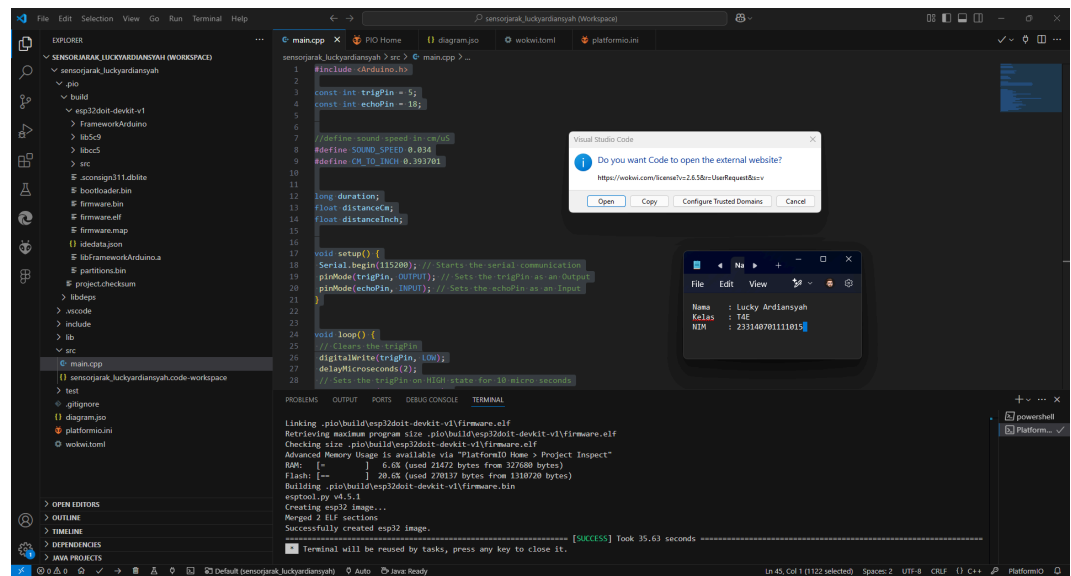
```

distanceInch = distanceCm * CM_TO_INCH;
Serial.print("Distance (cm): ");
Serial.println(distanceCm);
delay(1000);
}

```

3. Integrasi dengan VS Code:

- Buka VS Code dan pastikan ekstensi "Wokwi for VS Code" telah terpasang.
- Salin file diagram.json dan main.cpp dari Wokwi ke dalam proyek di VS Code.
- Buka terminal di VS Code dan jalankan perintah untuk menginisiasi proyek Wokwi:
wokwi init
- Jalankan perintah > Request License untuk mendapatkan lisensi proyek Wokwi:
- Ikuti petunjuk untuk masuk ke akun Wokwi dan mendapatkan kunci lisensi.



4. Menjalankan Simulasi di VS Code:

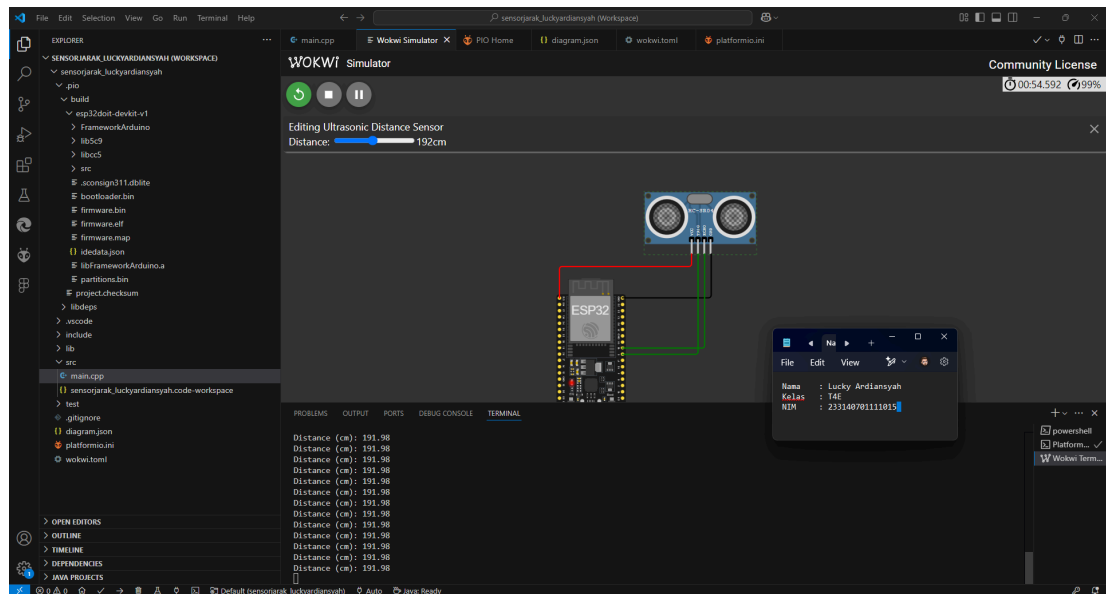
- Untuk menjalankan simulasi Wokwi langsung dari VS Code
- Pastikan tidak ada error dan semua komponen berfungsi seperti yang telah diuji di Wokwi.

5. Verifikasi Kode dan Diagram:

- Pastikan file main.cpp dan diagram.json telah terintegrasi dengan benar.
- Sesuaikan konfigurasi pin sesuai kebutuhan dan jalankan simulasi ulang jika ada perubahan.

3. Result & Discussion (Hasil dan Pembahasan)

3.1 Experimental Result (Hasil Eksperimen)



LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

**Praktik Simulasi Relay, Button & LED menggunakan Wokwi dan
PlatformIO IDE pada VS Code**

*Lucky Ardiansyah 233140701111015
Fakultas Vokasi, Universitas Brawijaya
Email: luckyardiansyah685@gmail.com*

Abstract (Abstrak)

Praktikum ini bertujuan untuk mempelajari proses pembuatan dan pengujian API menggunakan **Laravel 11** serta memanfaatkan **Ngrok** untuk mengakses API secara publik. Laravel, sebagai salah satu framework PHP terpopuler, menyediakan berbagai fitur bawaan yang mempermudah pembuatan RESTful API, termasuk routing, middleware, dan validasi data. Dalam praktikum ini, API dikembangkan untuk menangani permintaan HTTP (GET, POST, PUT, dan DELETE) serta mengelola data sederhana dalam basis data. Ngrok digunakan untuk membuat terowongan aman yang memungkinkan API diakses dari jaringan eksternal tanpa konfigurasi server tambahan. Hasil praktikum menunjukkan bahwa integrasi Laravel dan Ngrok mempermudah proses pengembangan, pengujian, dan kolaborasi dalam membangun layanan backend. Praktikum ini memberikan wawasan mendalam terkait pengembangan API modern, pengujian endpoint, serta pentingnya keamanan dalam implementasi API publik.

Kata Kunci: Laravel 11, Ngrok, RESTful API, HTTP Request, Pengujian API

1. Introduction (Pendahuluan)

1.1 Latar Belakang

Dalam era digital yang semakin berkembang, kebutuhan akan aplikasi yang saling terhubung dan mampu bertukar data secara real-time menjadi sangat krusial. Salah satu teknologi yang mendukung komunikasi antar aplikasi adalah Application Programming Interface (API). API memungkinkan berbagai sistem, baik web, mobile, maupun perangkat lainnya, untuk berinteraksi dan bertukar informasi secara terstruktur.

Laravel, sebagai salah satu framework PHP modern, menawarkan fitur bawaan untuk membangun RESTful API secara efisien dan aman. Versi terbaru, Laravel 11, hadir dengan berbagai peningkatan performa, penyederhanaan sintaks, serta fitur-fitur baru yang mendukung pengembangan API, seperti routing yang fleksibel, middleware untuk keamanan, dan integrasi basis data yang kuat. Namun, pengujian dan demonstrasi API sering kali memerlukan server publik agar endpoint dapat diakses secara eksternal. Di sinilah peran Ngrok menjadi penting. Ngrok memungkinkan pengembang membuat terowongan aman dari localhost ke internet, sehingga API yang berjalan secara lokal dapat diuji dan diakses oleh rekan tim atau aplikasi lain tanpa perlu melakukan konfigurasi server yang kompleks.

Praktikum ini bertujuan untuk memberikan pemahaman mendalam tentang cara membangun API menggunakan Laravel 11, menghubungkannya dengan basis data, serta memanfaatkan Ngrok untuk menguji konektivitas API secara publik. Dengan praktik ini, diharapkan peserta mampu mengembangkan API yang terstruktur, aman, dan siap untuk dikolaborasikan.

1.2 Tujuan Eksperimen

- Memahami konsep dasar API dan prinsip kerja RESTful API.
- Mengimplementasikan pembuatan API menggunakan Laravel 11.
- Mengatur routing, middleware, dan controller untuk menangani berbagai jenis permintaan HTTP (GET, POST, PUT, DELETE).
- Menghubungkan API dengan basis data untuk operasi CRUD (Create, Read, Update, Delete).
- Menggunakan Ngrok untuk membuat API lokal dapat diakses secara eksternal.
- Menguji endpoint API menggunakan aplikasi seperti Postman atau browser untuk memastikan respons berjalan sesuai harapan.
- Memahami pentingnya keamanan dalam mengembangkan API publik, termasuk penerapan token atau autentikasi.

2. Methodology (Metodologi)

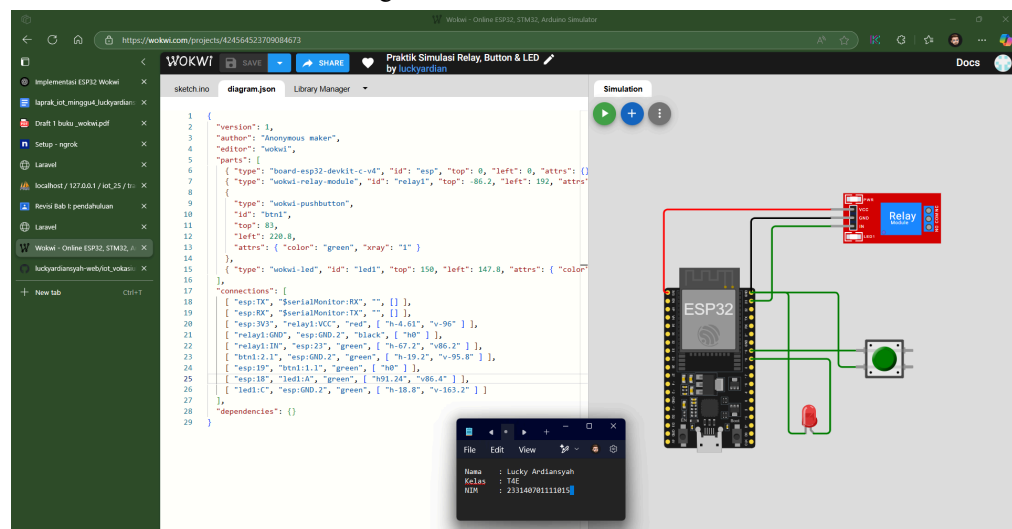
2.1 Tools & Materials (Alat & Bahan)

- Wokwi Simulator
- Visual Studio Code (VS Code)
- Ekstensi Wokwi untuk VS Code
- Arduino IDE
- ESP32 DevKit V1
- Relay Module
- Push Button
- LED
- Kabel Virtual

2.1 Implementation Steps (Langkah Implementasi)

A. Persiapan Proyek di Wokwi:

- Buka situs Wokwi (<https://wokwi.com/>) dan login menggunakan akun yang telah terdaftar.
- Klik "New Project" dan pilih "ESP32 DevKit V1" sebagai board.
- Tambahkan komponen sesuai kebutuhan dengan mengedit file diagram.json
- Komponen yang digunakan meliputi:
 - ESP32 DevKit V1
 - Relay module
 - Push button (warna hijau)
 - LED (warna merah)
- Hubungkan pin sesuai skema berikut:
 - GPIO19 (ESP32) ke push button
 - GPIO18 (ESP32) ke LED
 - GPIO23 (ESP32) ke relay module
 - Pastikan koneksi ground dan VCC sesuai.



B. Menulis Kode di Wokwi:

- Buka file main.cpp dan salin kode berikut:

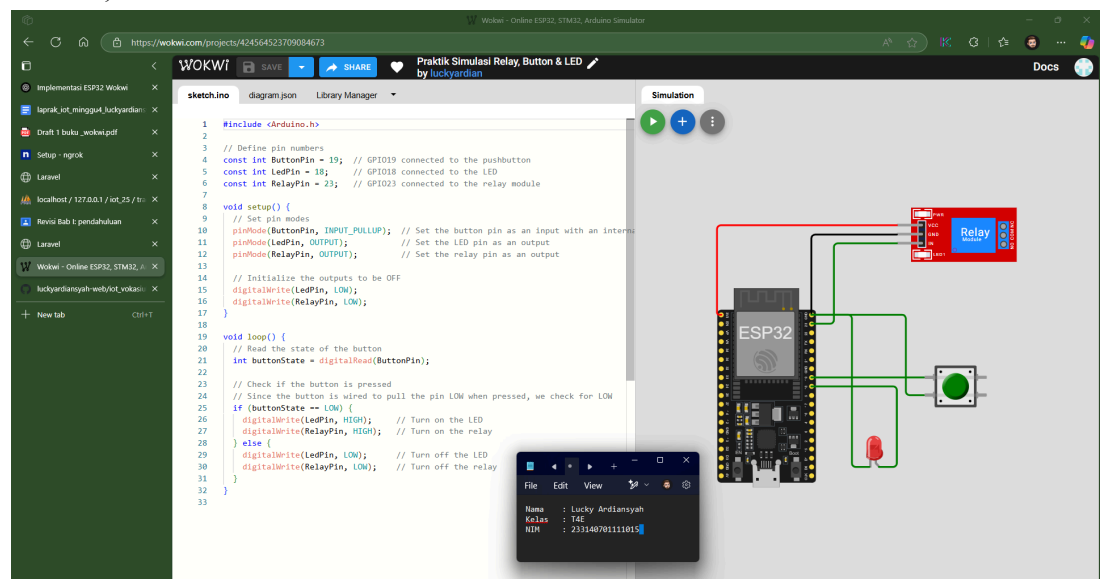
```
#include <Arduino.h>
// Define pin numbers
```



```
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
```

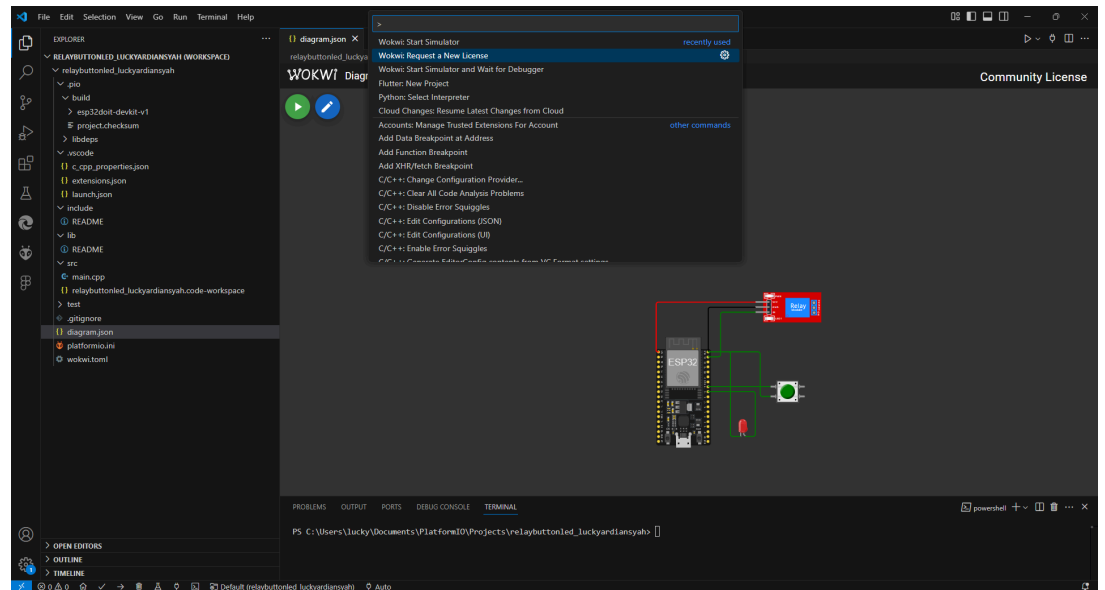
```
void setup() {
  pinMode(ButtonPin, INPUT_PULLUP); // Set the button pin as input with pull-up
  pinMode(LedPin, OUTPUT); // Set LED pin as output
  pinMode(RelayPin, OUTPUT); // Set relay pin as output
  digitalWrite(LedPin, LOW);
  digitalWrite(RelayPin, LOW);
}
```

```
void loop() {
  int buttonState = digitalRead(ButtonPin);
  if (buttonState == LOW) { // Button pressed
    digitalWrite(LedPin, HIGH);
    digitalWrite(RelayPin, HIGH);
  } else { // Button not pressed
    digitalWrite(LedPin, LOW);
    digitalWrite(RelayPin, LOW);
  }
}
```



C. Integrasi dengan VS Code:

- Buka VS Code dan pastikan ekstensi "Wokwi for VS Code" telah terpasang.
- Salin file diagram.json dan main.cpp dari Wokwi ke dalam proyek di VS Code.
- Buka terminal di VS Code dan jalankan perintah untuk menginisiasi proyek Wokwi:
wokwi init
- Jalankan perintah > Request License untuk mendapatkan lisensi proyek Wokwi:
- Ikuti petunjuk untuk masuk ke akun Wokwi dan mendapatkan kunci lisensi.



D. Menjalankan Simulasi di VS Code:

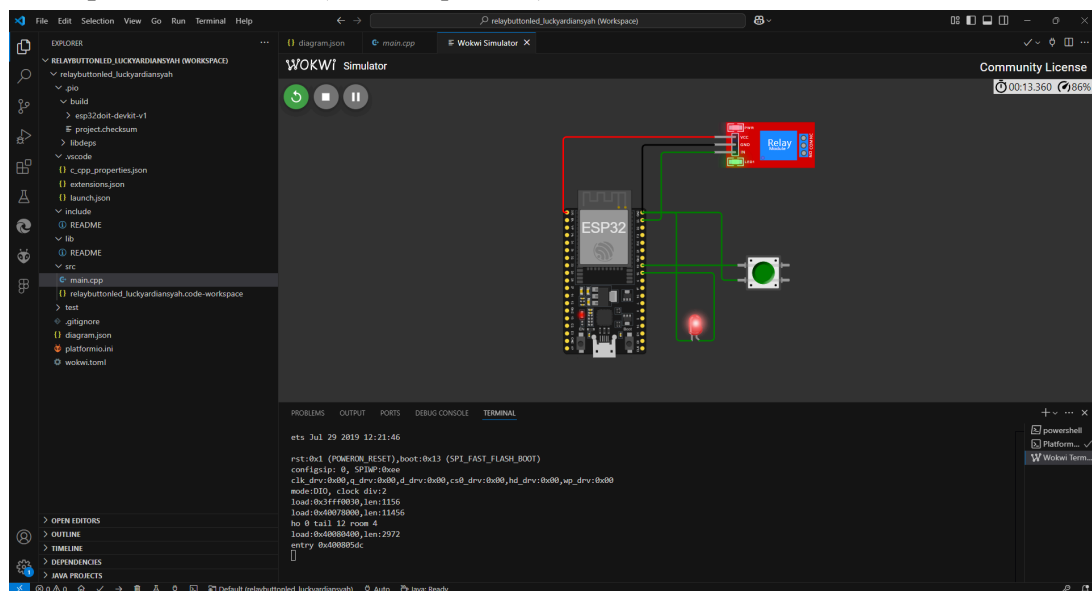
- Untuk menjalankan simulasi Wokwi langsung dari VS Code
- Pastikan tidak ada error dan semua komponen berfungsi seperti yang telah diuji di Wokwi.

E. Verifikasi Kode dan Diagram:

- Pastikan file main.cpp dan diagram.json telah terintegrasi dengan benar.
- Sesuaikan konfigurasi pin sesuai kebutuhan dan jalankan simulasi ulang jika ada perubahan.

4. Result & Discussion (Hasil dan Pembahasan)

3.1 Experimental Result (Hasil Eksperimen)



LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Praktik Pembuatan API Menggunakan Laravel 11 dan Ngrok

*Lucky Ardiansyah 233140701111015
Fakultas Vokasi, Universitas Brawijaya
Email: luckyardiansyah685@gmail.com*

Abstract (Abstrak)

Praktikum ini bertujuan untuk mempelajari proses pembuatan dan pengujian API menggunakan **Laravel 11** serta memanfaatkan **Ngrok** untuk mengakses API secara publik. Laravel, sebagai salah satu framework PHP terpopuler, menyediakan berbagai fitur bawaan yang mempermudah pembuatan RESTful API, termasuk routing, middleware, dan validasi data. Dalam praktikum ini, API dikembangkan untuk menangani permintaan HTTP (GET, POST, PUT, dan DELETE) serta mengelola data sederhana dalam basis data. Ngrok digunakan untuk membuat terowongan aman yang memungkinkan API diakses dari jaringan eksternal tanpa konfigurasi server tambahan. Hasil praktikum menunjukkan bahwa integrasi Laravel dan Ngrok mempermudah proses pengembangan, pengujian, dan kolaborasi dalam membangun layanan backend. Praktikum ini memberikan wawasan mendalam terkait pengembangan API modern, pengujian endpoint, serta pentingnya keamanan dalam implementasi API publik.

Kata Kunci: Laravel 11, Ngrok, RESTful API, HTTP Request, Pengujian API

3. Introduction (Pendahuluan)

1.1 Latar Belakang

Dalam era digital yang semakin berkembang, kebutuhan akan aplikasi yang saling terhubung dan mampu bertukar data secara real-time menjadi sangat krusial. Salah satu teknologi yang mendukung komunikasi antar aplikasi adalah Application Programming Interface (API). API memungkinkan berbagai sistem, baik web, mobile, maupun perangkat lainnya, untuk berinteraksi dan bertukar informasi secara terstruktur.

Laravel, sebagai salah satu framework PHP modern, menawarkan fitur bawaan untuk membangun RESTful API secara efisien dan aman. Versi terbaru, Laravel 11, hadir dengan berbagai peningkatan performa, penyederhanaan sintaks, serta fitur-fitur baru yang mendukung pengembangan API, seperti routing yang fleksibel, middleware untuk keamanan, dan integrasi basis data yang kuat. Namun, pengujian dan demonstrasi API sering kali memerlukan server publik agar endpoint dapat diakses secara eksternal. Di sinilah peran Ngrok menjadi penting. Ngrok memungkinkan pengembang membuat terowongan aman dari localhost ke internet, sehingga API yang berjalan secara lokal dapat diuji dan diakses oleh rekan tim atau aplikasi lain tanpa perlu melakukan konfigurasi server yang kompleks.

Praktikum ini bertujuan untuk memberikan pemahaman mendalam tentang cara membangun API menggunakan Laravel 11, menghubungkannya dengan basis data, serta memanfaatkan Ngrok untuk menguji konektivitas API secara publik. Dengan praktik ini, diharapkan peserta mampu mengembangkan API yang terstruktur, aman, dan siap untuk dikolaborasikan.

1.2 Tujuan Eksperimen

- Memahami konsep dasar API dan prinsip kerja RESTful API.
- Mengimplementasikan pembuatan API menggunakan Laravel 11.
- Mengatur routing, middleware, dan controller untuk menangani berbagai jenis permintaan HTTP (GET, POST, PUT, DELETE).
- Menghubungkan API dengan basis data untuk operasi CRUD (Create, Read, Update, Delete).
- Menggunakan Ngrok untuk membuat API lokal dapat diakses secara eksternal.
- Menguji endpoint API menggunakan aplikasi seperti Postman atau browser untuk memastikan respons berjalan sesuai harapan.
- Memahami pentingnya keamanan dalam mengembangkan API publik, termasuk penerapan token atau autentikasi.

4. **Methodology** (Metodologi)

1. Instalasi dan Konfigurasi Laravel 11

Unduh Laravel 11 menggunakan Composer dengan perintah:

```
composer create-project --prefer-dist laravel/laravel:^11.0 laravel-11
```

- Masuk ke direktori proyek Laravel yang telah dibuat:
cd laravel-11
- **Buat database** di **phpMyAdmin** dengan nama **iot_25**.
- **Konfigurasi koneksi database** dengan mengedit file .env:
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=iot_25
DB_USERNAME=root
DB_PASSWORD=caberg2010

(Pastikan DB_USERNAME dan DB_PASSWORD disesuaikan dengan konfigurasi lokal).

2. Membuat Model dan Migrasi Database

Buat model dan migrasi untuk tabel transaksi_sensor:

```
php artisan make:model TransaksiSensor -m
```

Edit file migrasi yang ada di database/migrations/2025_02_21_074123_create_transaksi_sensors_table.php dengan isi berikut:

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    public function up(): void {
        Schema::create('transaksi_sensor', function (Blueprint $table) {
            $table->id();
            $table->string('nama_sensor', 255);
            $table->integer('nilai1');
            $table->integer('nilai2');
            $table->timestamps();
        });
    }

    public function down(): void {
        Schema::dropIfExists('transaksi_sensor');
    }
};
```

1. Jalankan **migrasi database**:

```
php artisan migrate
```

3. Membuat Model dan Resource API

Edit file model TransaksiSensor.php di app/Models/TransaksiSensor.php:

```
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class TransaksiSensor extends Model {
    use HasFactory;
    protected $table = 'transaksi_sensor';
    protected $fillable = ['nama_sensor', 'nilai1', 'nilai2'];
}
```

1. **Buat Resource API:**

php artisan make:resource TransaksiSensorResource

2. **Edit file TransaksiSensorResource.php** di app/Http/Resources/TransaksiSensorResource.php:

```
namespace App\Http\Resources;
```

```
use Illuminate\Http\Resources\Json\JsonResource;
```

```
class TransaksiSensorResource extends JsonResource {
    public function toArray($request) {
        return [
            'id' => $this->id,
            'nama_sensor' => $this->nama_sensor,
            'nilai1' => $this->nilai1,
            'nilai2' => $this->nilai2,
        ];
    }
}
```

3.

4. Membuat Controller API

Buat controller API dengan perintah:

php artisan make:controller Api/TransaksiSensorController

1. **Edit file TransaksiSensorController.php** di app/Http/Controllers/Api/TransaksiSensorController.php:

```
namespace App\Http\Controllers\Api;
```

```
use App\Models\TransaksiSensor;
use App\Http\Controllers\Controller;
use App\Http\Resources\TransaksiSensorResource;
use Illuminate\Http\Request;
```

```
class TransaksiSensorController extends Controller {
    public function index() {
        return TransaksiSensorResource::collection(TransaksiSensor::all());
    }
}
```

```
public function store(Request $request) {
    $validated = $request->validate([
        'nama_sensor' => 'required|string|max:255',
        'nilai1' => 'required|integer',
        'nilai2' => 'required|integer',
    ]);

    $sensor = TransaksiSensor::create($validated);
    return new TransaksiSensorResource($sensor);
}
```

```

public function show($id) {
    return new TransaksiSensorResource(TransaksiSensor::findOrFail($id));
}

public function update(Request $request, $id) {
    $sensor = TransaksiSensor::findOrFail($id);
    $sensor->update($request->all());
    return new TransaksiSensorResource($sensor);
}

public function destroy($id) {
    TransaksiSensor::destroy($id);
    return response()->json(['message' => 'Deleted successfully'], 204);
}
}

```

5. Menambahkan Routing API

Buka file `routes/api.php` dan ubah isinya menjadi:

```

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Api\TransaksiSensorController;
Route::apiResource('/posts', TransaksiSensorController::class);

```

1. **Pastikan route sudah terdaftar** dengan menjalankan:
php artisan route:list

6. Menjalankan API dengan Laravel Serve

Jalankan server Laravel dengan perintah:

```
php artisan serve
```

Akses API melalui browser atau Postman di `http://127.0.0.1:8000/api/posts`.

7. Testing API Menggunakan Postman

1. **GET data:**
 - Masukkan URL `http://127.0.0.1:8000/api/posts`
 - Pilih method GET
 - Klik SEND dan pastikan data ditampilkan dalam format JSON.
2. **POST data (Insert):**
 - Masukkan URL `http://127.0.0.1:8000/api/posts`
 - Pilih method POST

Pada tab **Body**, pilih raw dan format **JSON**, lalu masukkan:

```

{
  "nama_sensor": "Sensor A",
  "nilai1": 100,
  "nilai2": 200
}

```

○

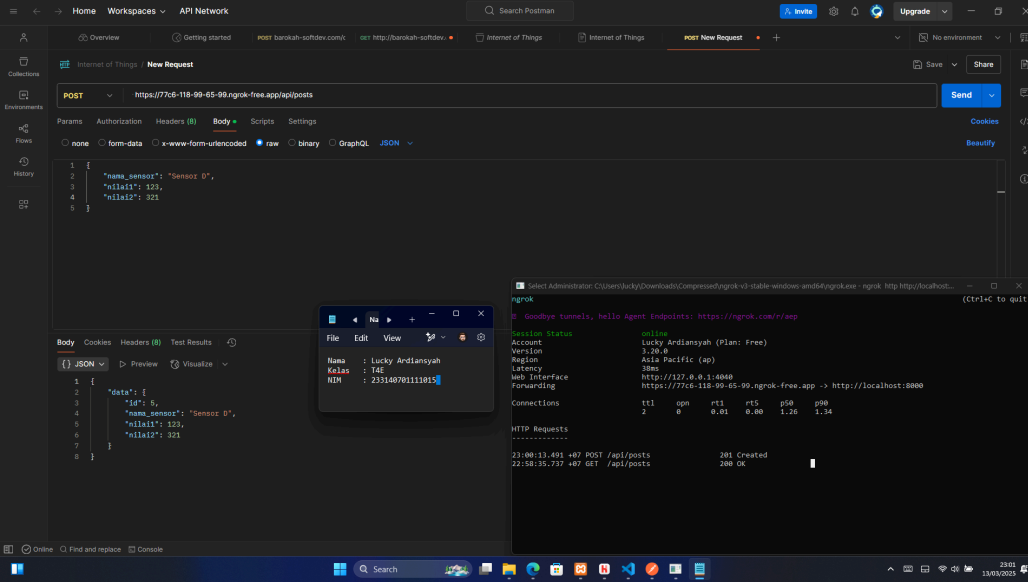
- Klik SEND dan pastikan data berhasil disimpan.

8. Mengonline-kan API Menggunakan Ngrok

1. **Download dan install Ngrok** dari dashboard.ngrok.com/signup.
2. **Buka terminal di direktori ekstraksi Ngrok** dan jalankan perintah:
ngrok http 8000
3. **Dapatkan URL publik** yang diberikan oleh Ngrok, contoh:
<https://e521-2405-8740-6315-xxxx-xxxx-xxxx.ngrok-free.app>
4. **Uji API secara online** dengan Postman menggunakan URL Ngrok:
 - GET: <https://e521-xxxx.ngrok-free.app/api/posts>
 - POST: <https://e521-xxxx.ngrok-free.app/api/posts>

5. Result & Discussion (Hasil dan Pembahasan)

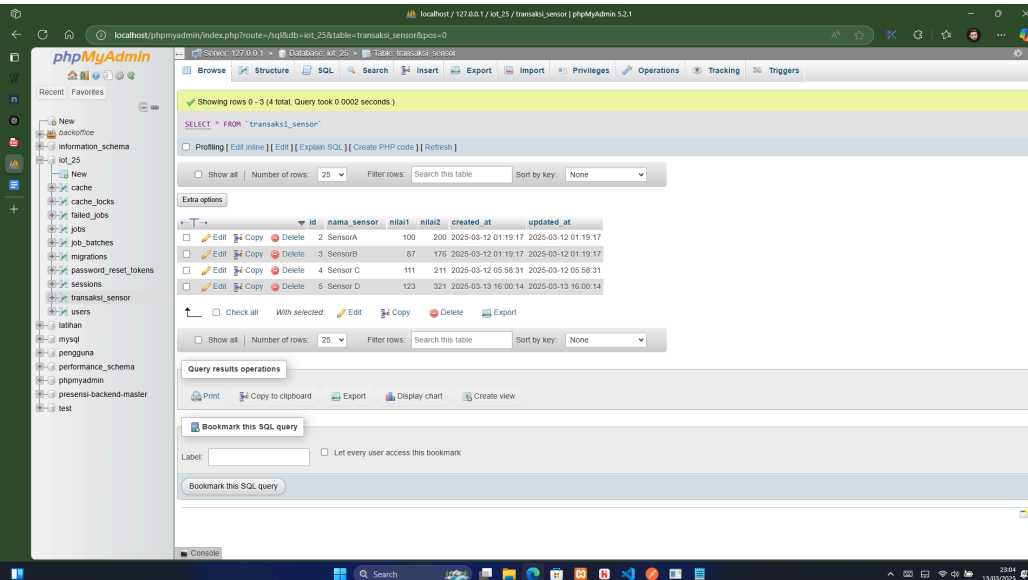
3.1 Experimental Result (Hasil Eksperimen)



The top screenshot shows the Postman interface with a POST request to <https://77c6-118-99-65-99.ngrok-free.app/api/posts>. The request body is a JSON object:

```
{
  "nama_sensor": "Sensor D",
  "nilai1": 123,
  "nilai2": 321
}
```

The response is a 201 Created status. Below the Postman interface, a terminal window shows the Ngrok command: `ngrok http http://localhost:8000`. The terminal output displays the Ngrok status, including the public URL <https://77c6-118-99-65-99.ngrok-free.app> and the local URL <http://localhost:8000>.



The bottom screenshot shows the phpMyAdmin interface. The 'transaksi_sensor' table is selected, and the query results are displayed. The table contains 4 rows of data:

ID	nama_sensor	nilai1	nilai2	created_at	updated_at
2	SensorA	100	200	2025-03-12 01:19:17	2025-03-12 01:19:17
3	SensorB	67	175	2025-03-12 01:19:17	2025-03-12 01:19:17
4	Sensor C	111	211	2025-03-12 05:58:31	2025-03-12 05:58:31
5	Sensor D	123	321	2025-03-13 16:00:14	2025-03-13 16:00:14