LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

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

*Nur Haliza Khairotun Nisa’*

Fakultas Vokasi, Universitas Brawijaya

[leezhaamail@gmail.com](mailto:leezhaamail@gmail.com)

**Abstract (Abstrak)**

Praktik ini bertujuan untuk mempelajari pembuatan API menggunakan framework Laravel 11 dan mengintegrasikannya dengan layanan Ngrok untuk membuat API dapat diakses secara publik melalui internet. Proses dimulai dengan pembuatan RESTful API yang menyediakan endpoint untuk mengelola data sensor IoT, seperti nama sensor dan nilai-nilainya. API ini kemudian diuji menggunakan Postman dan diintegrasikan dengan Ngrok untuk membuatnya dapat diakses dari perangkat lain atau platform IoT seperti ESP32. Hasil praktik menunjukkan bahwa Laravel 11 dapat digunakan untuk membangun API yang responsif dan terstruktur, sementara Ngrok memungkinkan pengujian API tanpa memerlukan hosting berbayar. Praktik ini memberikan wawasan tentang pentingnya API dalam mendukung komunikasi antar perangkat IoT.

*Keywords—Internet of Things, Laravel, API, Ngrok, RESTful*

1. **Introduction (Pendahuluan)**

**1.1 Latar Belakang**

Dalam pengembangan Internet of Things (IoT), API (Application Programming Interface) merupakan elemen kunci yang memungkinkan komunikasi antara perangkat keras (seperti ESP32) dan server backend. Framework Laravel telah menjadi salah satu pilihan populer untuk membangun API karena kemudahan penggunaan, struktur kode yang bersih, serta dukungan fitur yang lengkap. Namun, selama proses pengembangan, sering kali API hanya tersedia di lingkungan lokal dan tidak dapat diakses oleh perangkat lain. Untuk mengatasi masalah ini, layanan Ngrok digunakan untuk membuat tunnel ke server lokal sehingga API dapat diakses secara publik melalui internet.

Praktik ini membahas langkah-langkah pembuatan API sederhana menggunakan Laravel 11 dan integrasinya dengan Ngrok untuk mendukung pengembangan aplikasi IoT.

**1.2 Tujuan Eksperimen**

Praktik ini bertujuan untuk memberikan pemahaman menyeluruh tentang cara membangun API menggunakan framework Laravel 11 dan mengintegrasikannya dengan layanan Ngrok untuk membuat API dapat diakses secara publik melalui internet. Dalam praktik ini, mahasiswa akan mempelajari langkah-langkah pembuatan RESTful API yang menyediakan endpoint untuk mengelola data sensor IoT, seperti nama sensor dan nilai-nilainya. API ini kemudian diuji menggunakan Postman dan diintegrasikan dengan Ngrok untuk memastikan bahwa API dapat diakses dari perangkat lain atau platform IoT seperti ESP32. Selain itu, praktik ini juga bertujuan untuk menjelaskan pentingnya API dalam mendukung komunikasi antar perangkat IoT serta memberikan wawasan tentang penggunaan alat bantu seperti Ngrok dalam pengembangan aplikasi IoT tanpa memerlukan hosting berbayar. Dengan demikian, mahasiswa diharapkan dapat memahami konsep dasar pembuatan API dan penerapannya dalam skenario nyata, seperti pengumpulan data sensor dan integrasi dengan perangkat IoT.

1. **Methodology (Metodologi)**

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

1. Komputer atau laptop dengan koneksi internet.
2. PHP 8.x dan Composer (untuk instalasi Laravel).
3. Laravel 11 (framework backend).
4. Ngrok (layanan tunneling).
5. Postman (untuk pengujian API).
6. Browser web (Google Chrome, Mozilla Firefox, dll.).
7. Database MySQL (menggunakan phpMyAdmin).

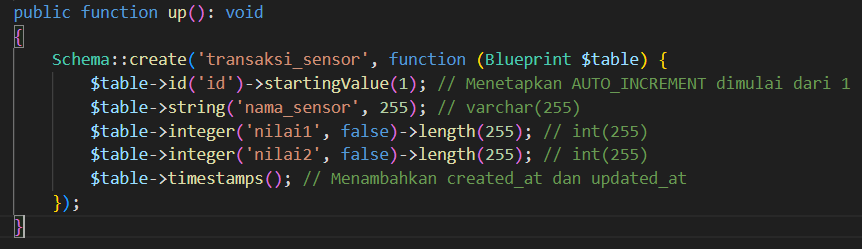
**2.2 Implementation Steps (Langkah Implementasi)**

1. Instalasi Laravel 11
2. Pastikan PHP 8.x dan Composer telah terinstal di sistem Anda.
3. Jalankan perintah berikut di terminal untuk membuat proyek Laravel baru:

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

cd laravel-11

1. Buat database di phpMyAdmin dengan nama iot\_25.
2. Ubah konfigurasi file .env sesuai dengan pengaturan database Anda :
3. DB\_CONNECTION=mysql
4. DB\_HOST=127.0.0.1
5. DB\_PORT=3306
6. DB\_DATABASE=iot\_25
7. DB\_USERNAME=root
8. DB\_PASSWORD=caberg2010
9. DB\_CHARSET=utf8mb4
10. DB\_COLLATION=utf8mb4\_unicode\_ci
11. Membuat Model dan Migration
12. Buat model dan migration untuk entitas "TransaksiSensor":
13. php artisan make:model TransaksiSensor -m
14. Edit file migration (2025\_02\_21\_074123\_create\_transaksi\_sensors\_table.php) untuk menambahkan kolom:



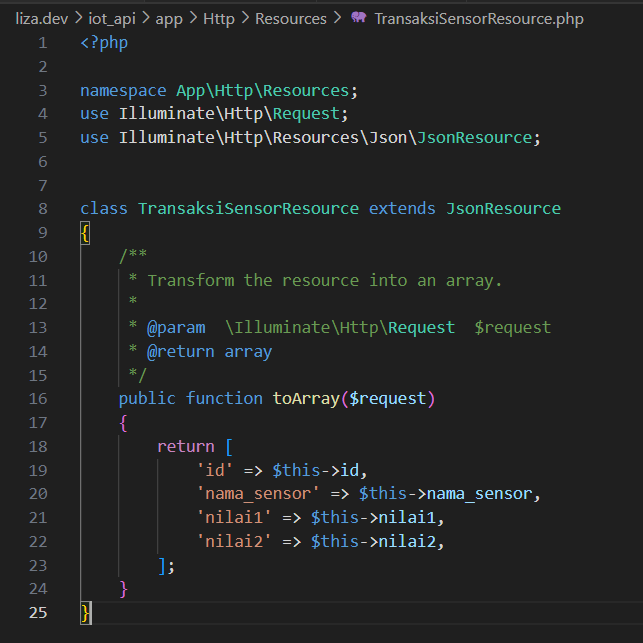
1. Jalankan migration untuk membuat tabel di database

php artisan migrate

1. Membuat Resource dan Controller
2. Buat resource untuk transformasi data:

php artisan make:resource TransaksiSensorResource

1. Edit file TransaksiSensorResource.php:



1. Buat controller untuk menangani request API:

php artisan make:controller Api/TransaksiSensorController

1. Ubah file dan tambahkan logika CRUD di app/Http/Controllers/Api/TransaksiSensorController.php:

<?php

namespace App\Http\Controllers\Api;

use Illuminate\Http\Request;

use App\Models\TransaksiSensor;

use App\Http\Controllers\Controller;

use App\Http\Resources\TransaksiSensorResource;

class TransaksiSensorController extends Controller

{

    /\*\*

     \* index

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

        // Get all transactions from TransaksiSensor model, paginated

        $transaksiSensors = TransaksiSensor::latest()->paginate(5);

        // Return a collection of transactions as a resource

        return TransaksiSensorResource::collection($transaksiSensors);

    }

    /\*\*

 \* Store a newly created resource in storage.

 \*

 \* @param  \Illuminate\Http\Request  $request

 \* @return \Illuminate\Http\Response

 \*/

public function store(Request $request)

{

    $validatedData = $request->validate([

        'nama\_sensor' => 'required|string|max:255',

        'nilai1' => 'required|integer',

        'nilai2' => 'required|integer',

    ]);

    $transaksiSensor = TransaksiSensor::create($validatedData);

    return new TransaksiSensorResource($transaksiSensor);

}

/\*\*

 \* Display the specified resource.

 \*

 \* @param  int  $id

 \* @return \Illuminate\Http\Response

 \*/

public function show($id)

{

    $transaksiSensor = TransaksiSensor::findOrFail($id);

    return new TransaksiSensorResource($transaksiSenso);

}

/\*\*

 \* Update the specified resource in storage.

 \*

 \* @param  \Illuminate\Http\Request  $request

 \* @param  int  $id

 \* @return \Illuminate\Http\Response

 \*/

public function update(Request $request, $id)

{

    $validatedData = $request->validate([

        'nama\_sensor' => 'required|string|max:255',

        'nilai1' => 'required|integer',

        'nilai2' => 'required|integer',

    ]);

    $transaksiSensor = TransaksiSensor::findOrFail($id);

    $transaksiSensor->update($validatedData);

    return new TransaksiSensorResource($transaksiSensor);

}

/\*\*

 \* Remove the specified resource from storage.

 \*

 \* @param  int  $id

 \* @return \Illuminate\Http\Response

 \*/

public function destroy($id)

{

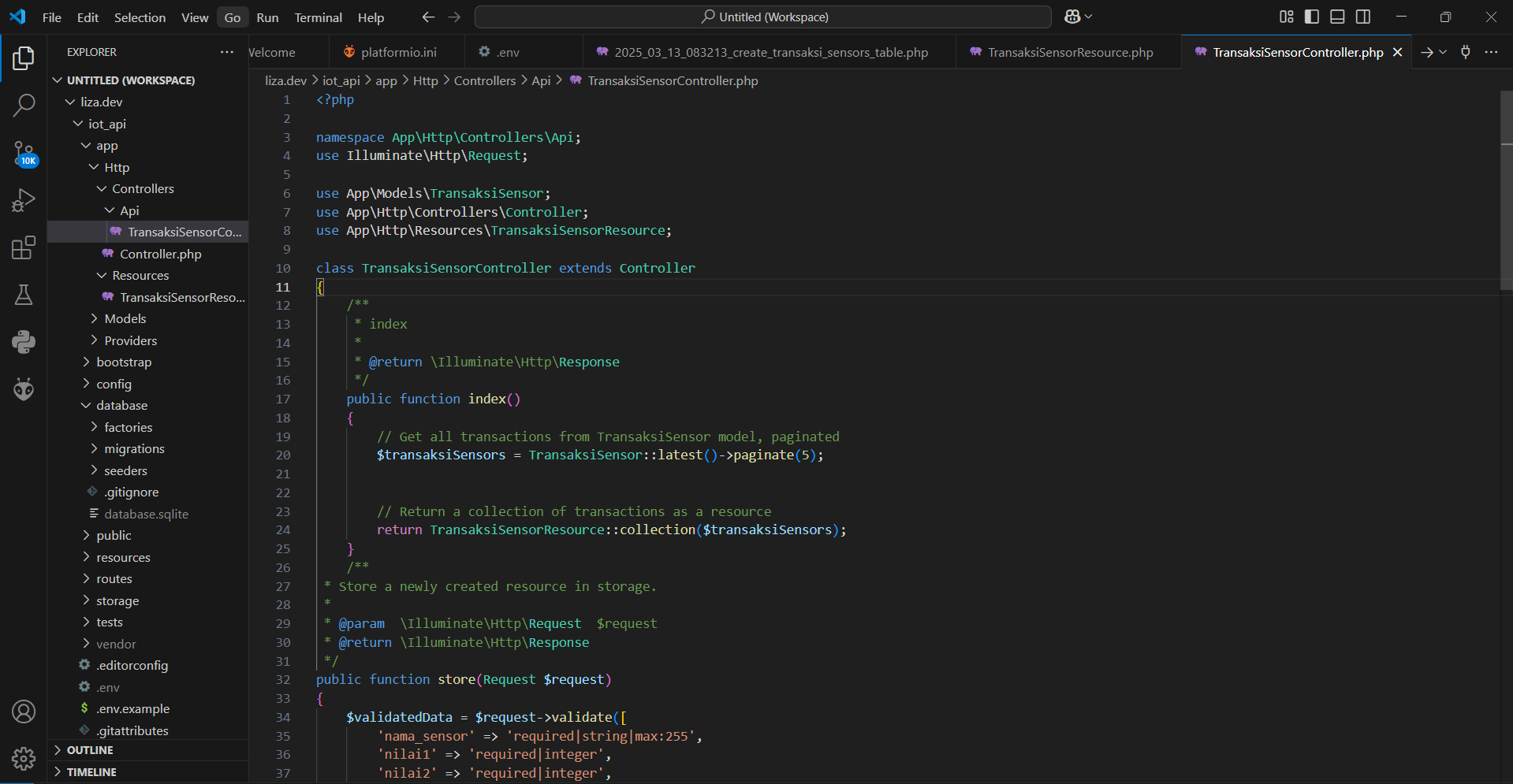
    $transaksiSensor = TransaksiSensor::findOrFail($id);

    $transaksiSensor->delete();

    return response()->json(['message' => 'Deleted successfully'], 204);

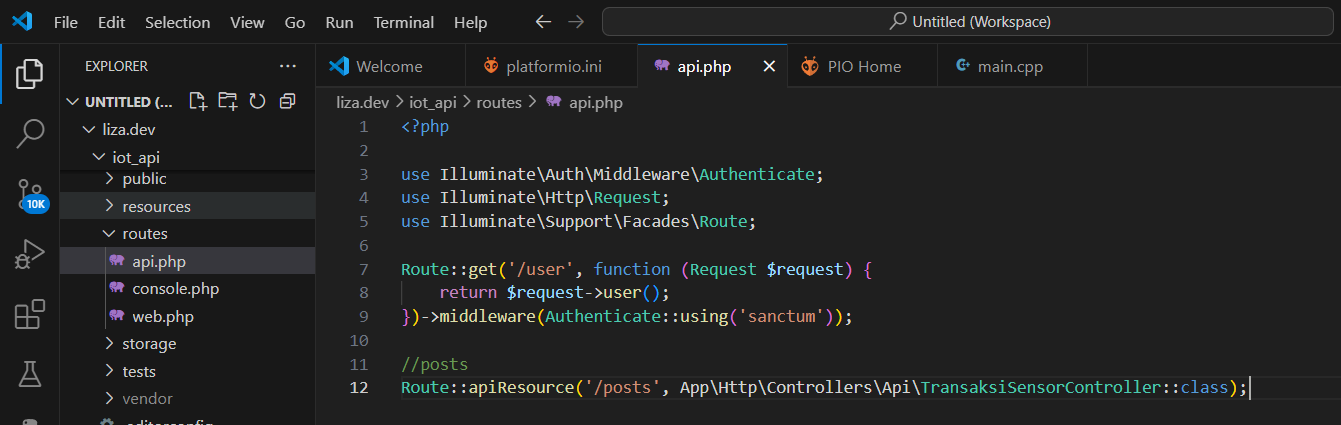
}

}



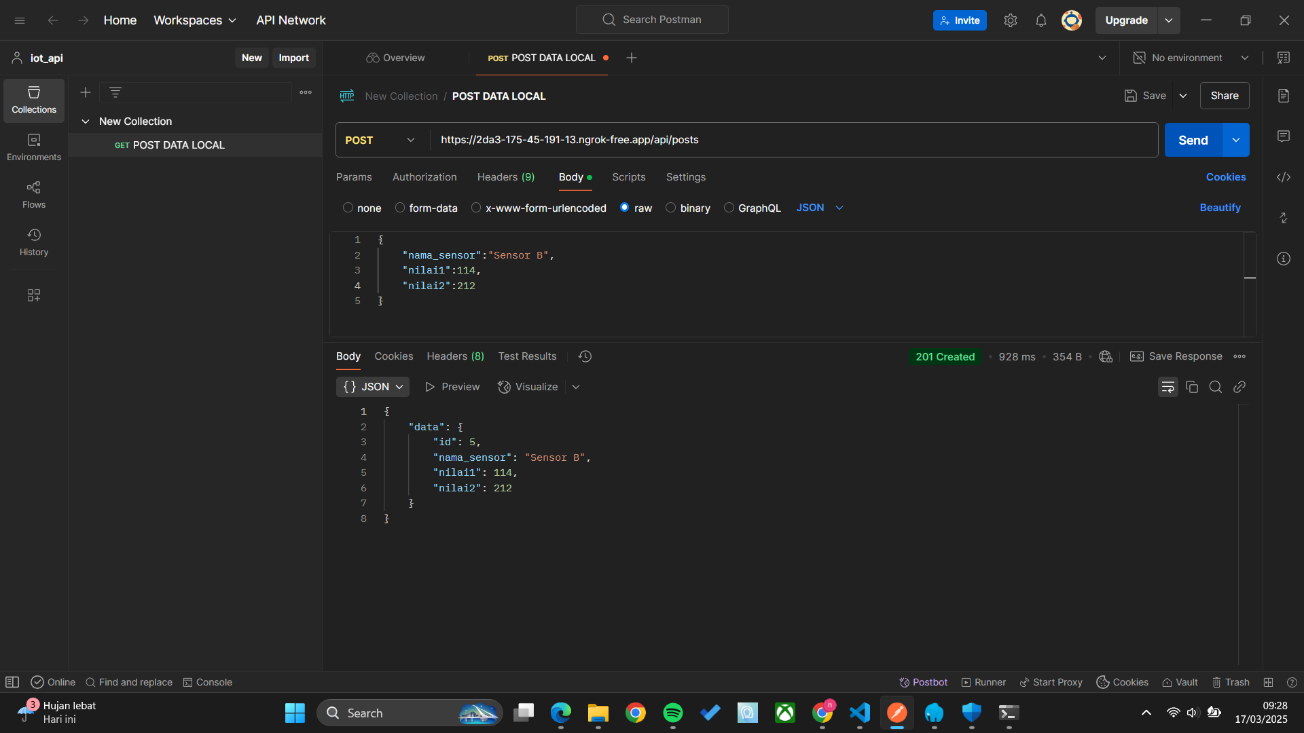
1. Menambahkan Route API
2. Buat route khusus API dengan menjalankan perintah :

php artisan install:api

1. Buka file **routes/api.php**  dan ubah isi file menjadi : 
2. Pengujian API Menggunakan Postman
3. Jalankan server Laravel



1. Uji endpoint API menggunakan Postman:



1. Integrasi dengan Ngrok
2. Unduh Ngrok dari https://ngrok.com/download dan instal sesuai instruksi.
3. Jalankan perintah berikut untuk membuat tunnel ke server lokal Laravel:

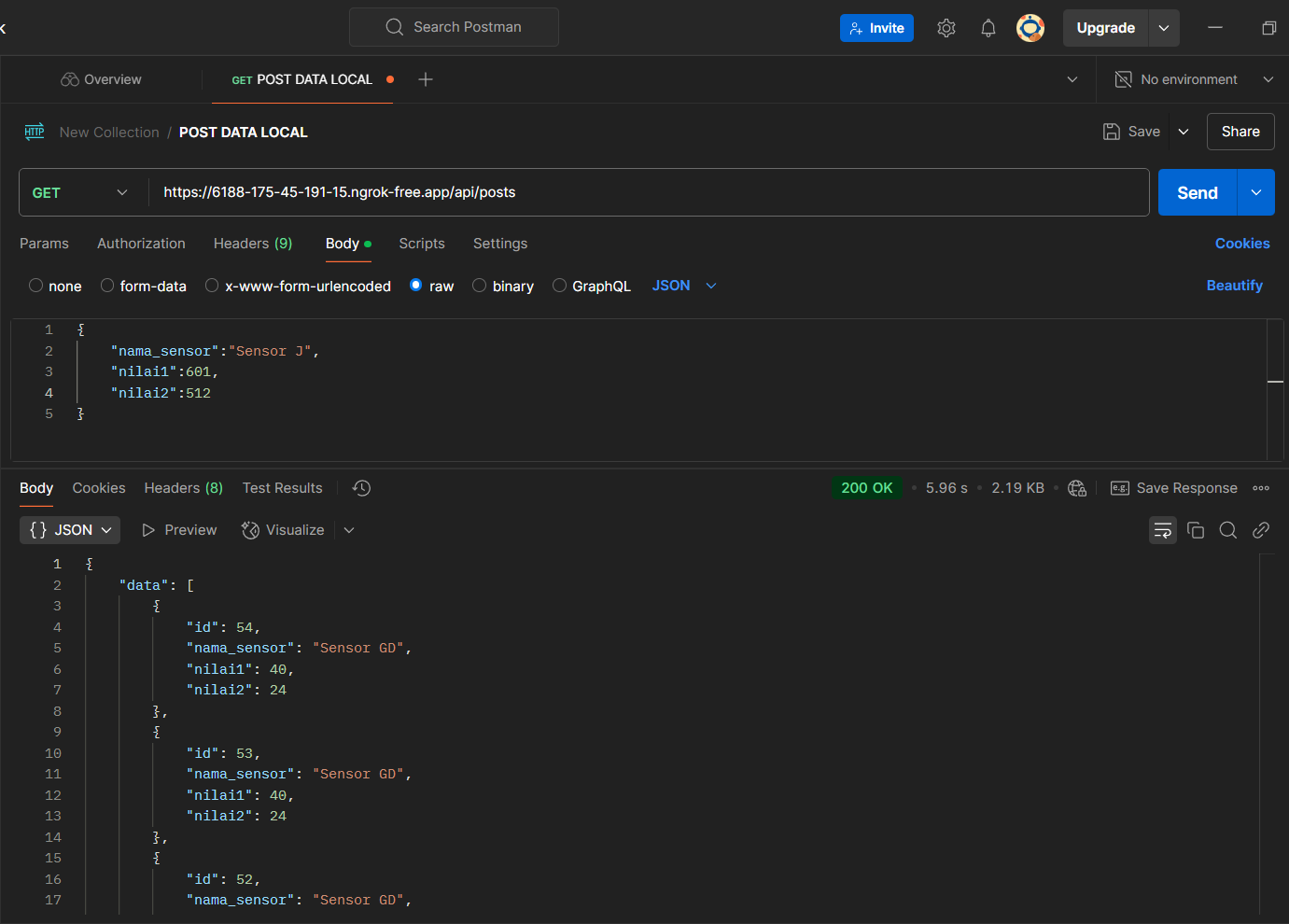


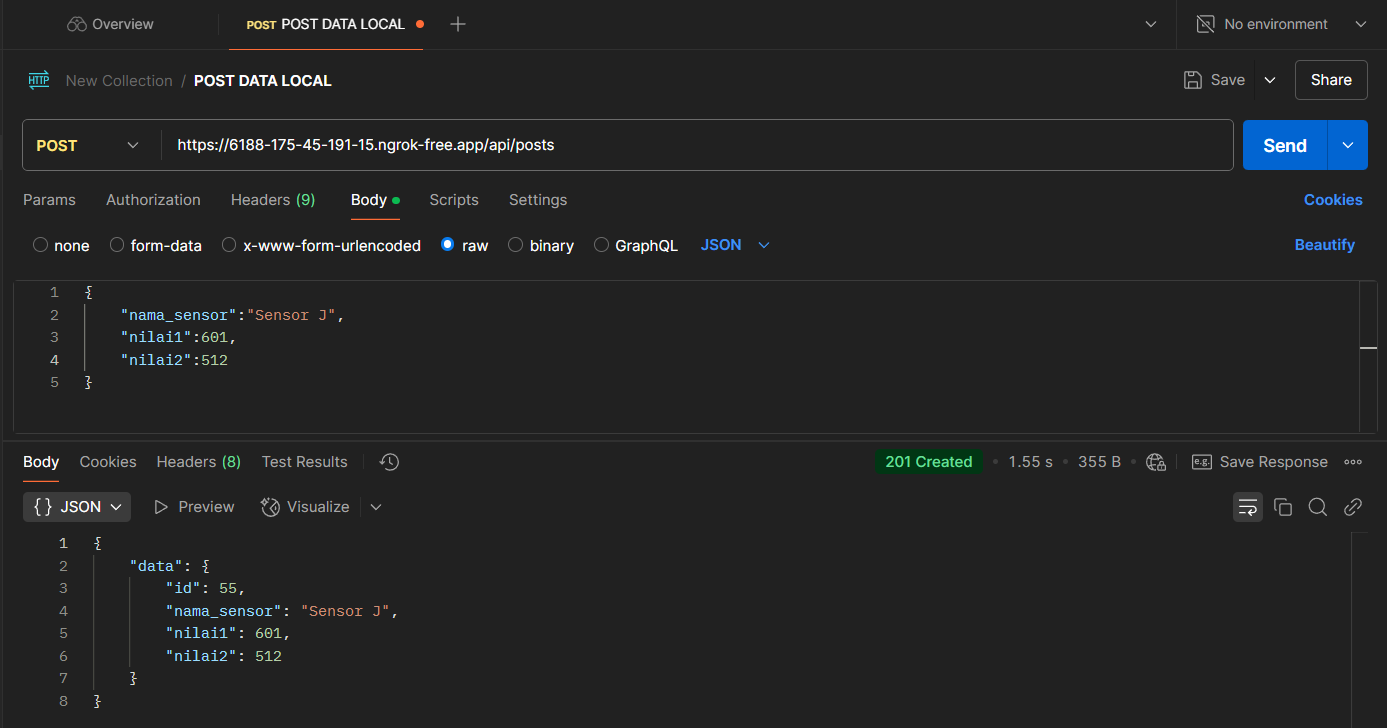
1. Ngrok akan memberikan URL public
2. Akses API menggunakan URL Ngrok untuk memastikan API dapat diakses dari internet.
3. **Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

Berikut adalah hasil dari praktik :

1. Pembuatan API : Berhasil dibuat menggunakan Laravel 11 dengan endpoint CRUD yang berfungsi dengan baik.
2. Pengujian API : Endpoint API berhasil diuji menggunakan Postman dan menghasilkan respons JSON yang sesuai.
3. Integrasi Ngrok : API dapat diakses secara publik melalui URL Ngrok, memungkinkan integrasi dengan perangkat IoT seperti ESP32.
4. **Appendix (Lampiran, jika diperlukan)**
5. Tampilan GET and POST postman ke database





c. Tampilan hasil GET dan POST di database

