LAPORAN PRAKTIKUM INTERNET OF THINGS

(IoT)

FAKULTAS VOKASI, UNIVERSITAS BRAWIJAYA



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

*Andrian Alfiani*

Fakultas Vokasi, Universitas Brawijaya

*Email: wildantaw@gmail.com*

**Abstrak**

Laravel 11 hadir dengan berbagai peningkatan performa dan fitur yang lebih efisien dalam pengelolaan API. Salah satu tantangan dalam pengembangan API adalah bagaimana cara mengaksesnya dari perangkat lain atau mengujinya di lingkungan lokal. Untuk mengatasi hal ini, Ngrok digunakan sebagai solusi untuk membuat tunnel yang memungkinkan API yang berjalan di localhost dapat diakses secara publik melalui internet.

*Keywords : Postman, Laravel 11 , API ,PHP my admin, Ngrok Laravel 11.*

**1. Pendahuluan**

* 1. **Latar Belakang**

Pembuatan praktikum ini dilakukan untuk memahami bagaimana membuat API menggunakan Laravel 11 dimana code editor yang di gunakan adalah Visual Studio Code.

**1.2 Tujuan eksperimen**

1. Memahami cara membuat API di Laravel 11.

1. Melakukan simulasi test API menggunakan Postman/Thunder.

**2. Methodology (Metodologi)**

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

* Laravel 11
* Visual Studio Code
* PHP my admin
* Nrok

**2.2 Implementation Steps (Langkah Implementasi)**

1. Buat proyek Larave

2. Masuk ke direktori proyek cd my-laravel-api

3. Buat database di phpMyAdmin atau MySQL CLI

4. Konfigurasi database di .env echo "

5. Buat migrasi

6. Edit file migrasi di database/migrations/

7. Buat model

8. Edit model Transaksi di app/Models/Transaksi.php

9. Jalankan migrasi php artisan migrate

10. Buat resource untuk API

11. Edit file resource di app/Http/Resources/TransaksiResource.php

12. Buat controller API

13. Edit file controller di app/Http/Controllers/Api/TransaksiController.php

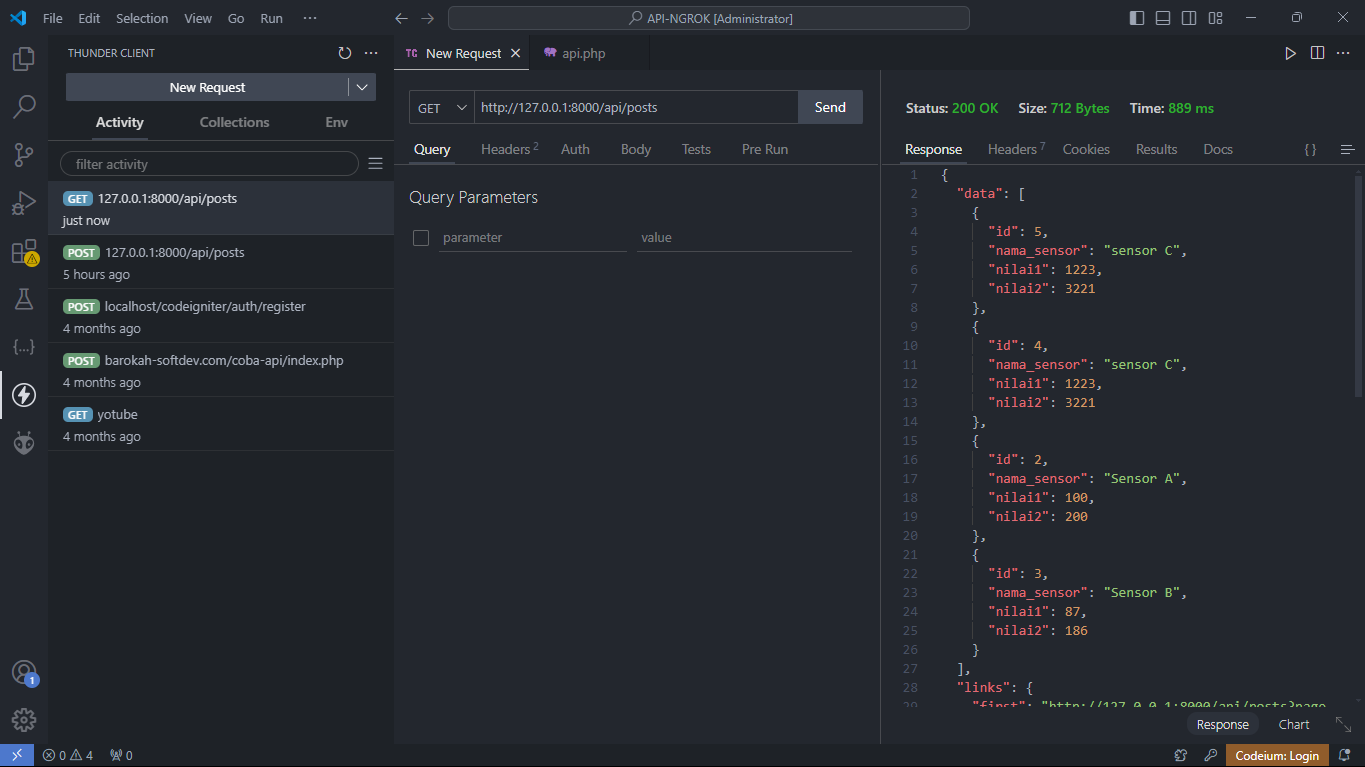
14. Tambahkan route API di routes/api.php

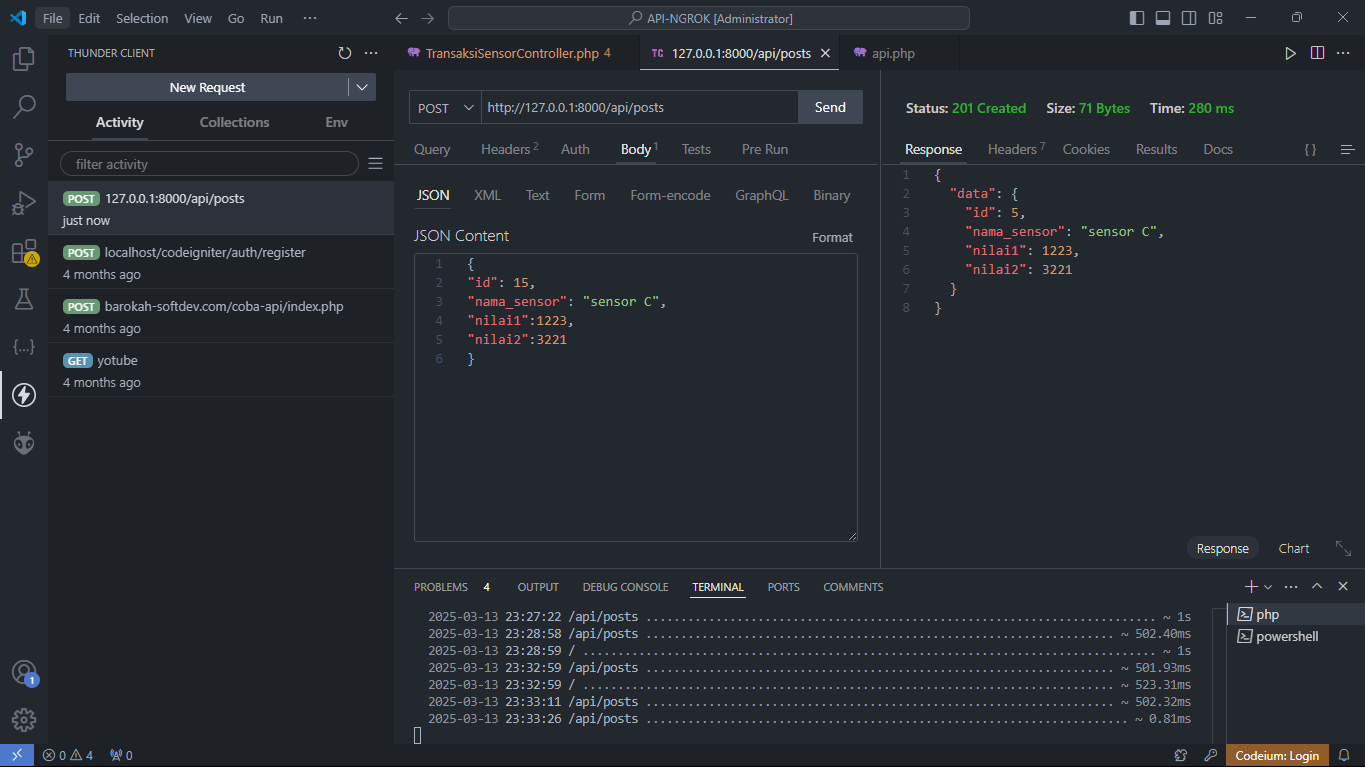
15. Jalankan route list php artisan route:list, Start server Laravel php artisan serve & Online-kan API dengan Ngrok

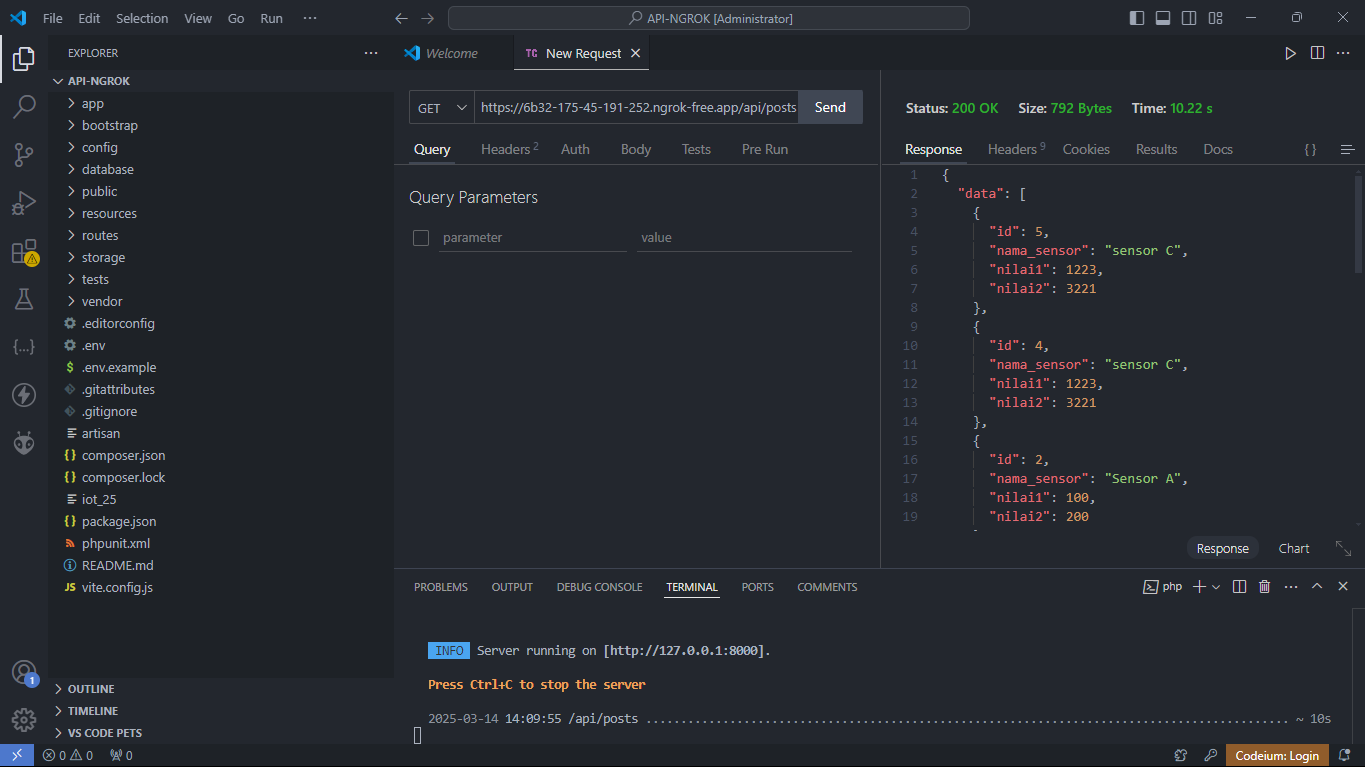
16. Integrasikan dengan ESP32 di Wokwi (gunakan URL dari Ngrok)

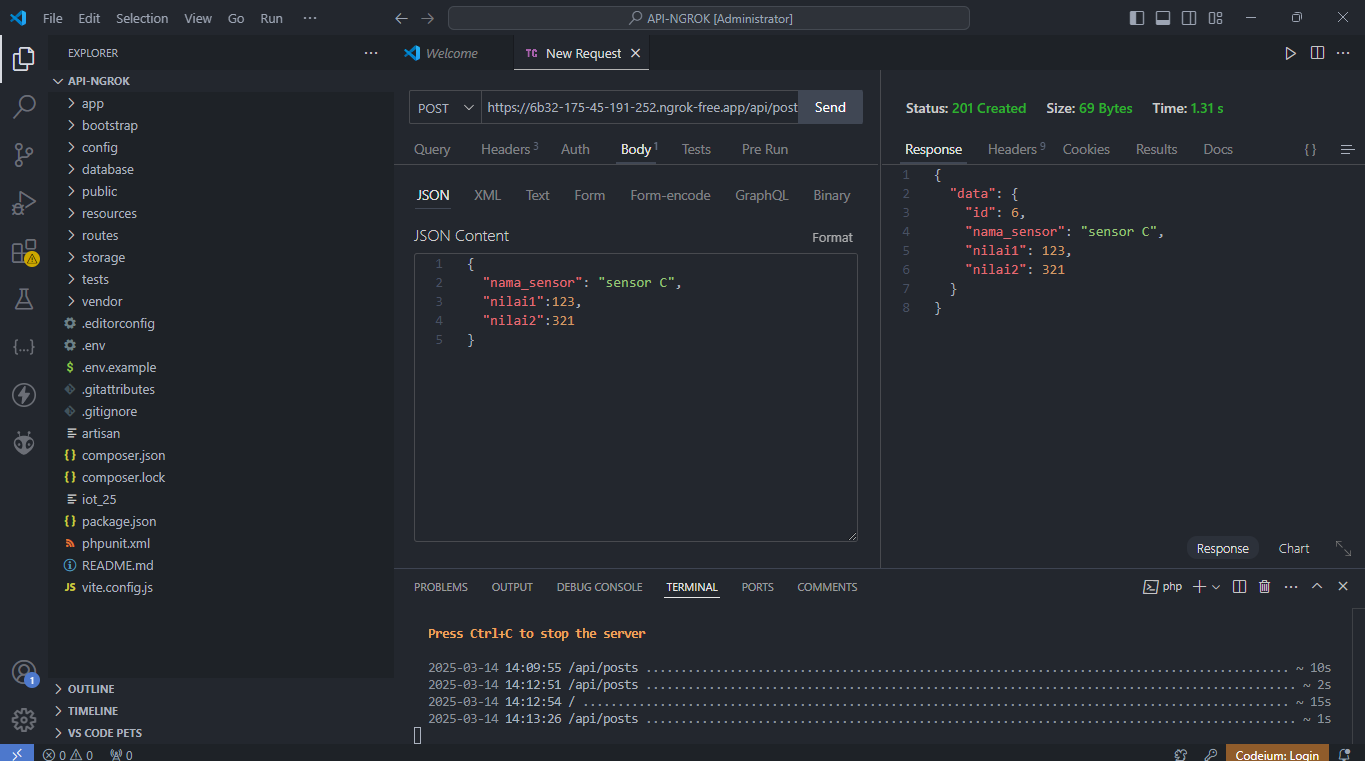
**3. Results and Discussion (Hasil dan Pembahasan)**

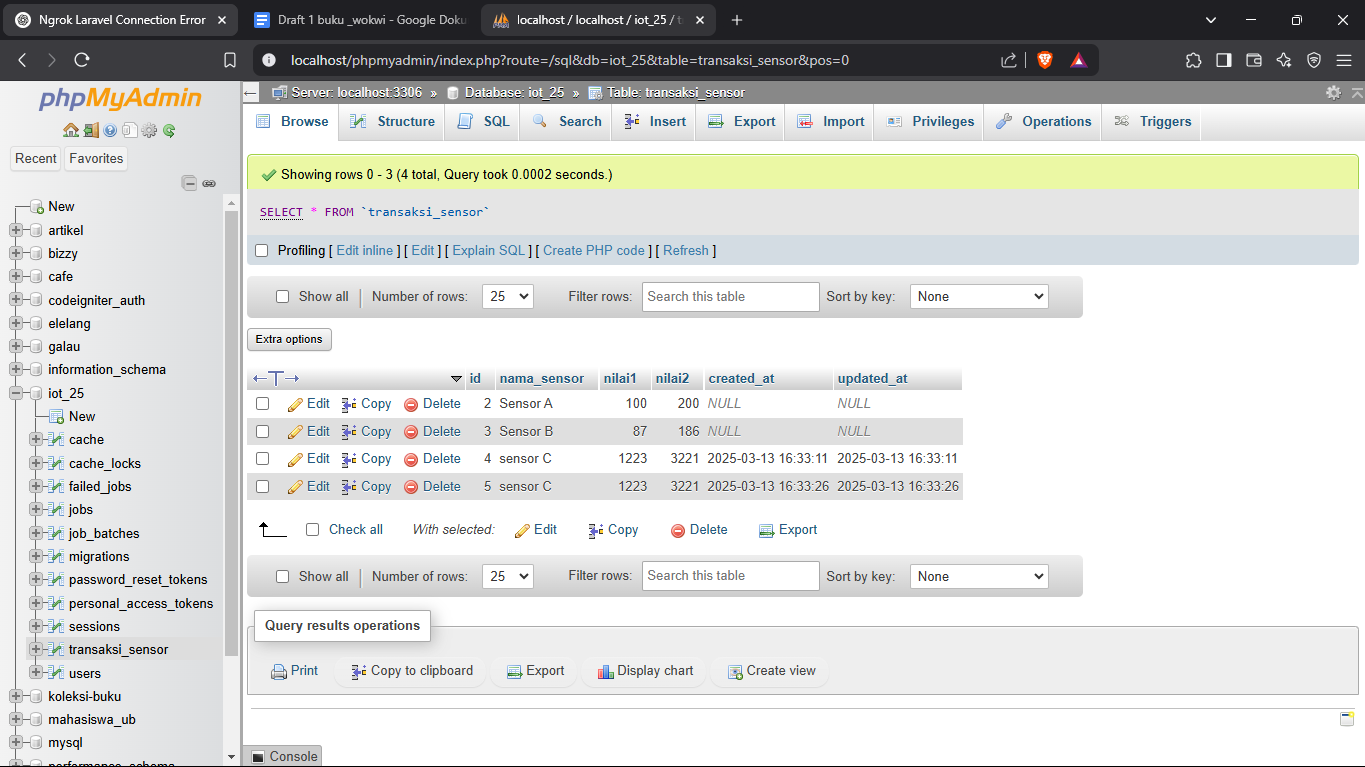
**3.1 Experimental Results**

****

****

****

****

****

**4. Appendix**

Migration:  
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('id')->startingValue(1); // Auto Increment dimulai dari 1

$table->string('nama\_sensor', 255);

$table->integer('nilai1');

$table->integer('nilai2');

$table->timestamps();

});

}

public function down(): void

{

Schema::dropIfExists('transaksi\_sensor');

}

};

Model:  
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',

];

}

**Resource API**

**namespace App\Http\Resources;**

**use Illuminate\Http\Request;**

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

**class TransaksiSensorResource extends JsonResource**

**{**

**public function toArray($request): array**

**{**

**return [**

**'id' => $this->id,**

**'nama\_sensor' => $this->nama\_sensor,**

**'nilai1' => $this->nilai1,**

**'nilai2' => $this->nilai2,**

**];**

**}**

**}**

**CONTROLLER API**

**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::latest()->paginate(5));**

**}**

**public function store(Request $request)**

**{**

**$validatedData = $request->validate([**

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

**'nilai1' => 'required|integer',**

**'nilai2' => 'required|integer',**

**]);**

**$sensor = TransaksiSensor::create($validatedData);**

**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::findOrFail($id)->delete();**

**return response()->json(['message' => 'Data berhasil dihapus'], 204);**

**}**

**}**

**FILE ROUTE API**

**use App\Http\Controllers\Api\TransaksiSensorController;**

**Route::apiResource('/posts', TransaksiSensorController::class);**