

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

Praktik Pembuatan API Menggunakan Laravel 11 Dan Ngrok

M Bimo Amarulloh

Fakultas Vokasi, Universitas Brawijaya

Email: bimoamar@gmail.com

Abstrak

Laravel 11 merupakan salah satu framework PHP yang populer digunakan dalam pengembangan aplikasi berbasis web, termasuk dalam pembuatan API (Application Programming Interface). API memungkinkan komunikasi antara aplikasi yang berbeda secara efisien. Salah satu tantangan dalam pengembangan API adalah pengujian dan akses dari lingkungan eksternal, terutama saat aplikasi masih dalam tahap pengembangan di lingkungan lokal. Ngrok hadir sebagai solusi untuk membuat tunneling yang memungkinkan API lokal dapat diakses secara publik melalui internet. Penelitian ini bertujuan untuk mengimplementasikan pembuatan API menggunakan Laravel 11 serta mengonfigurasikannya dengan Ngrok agar dapat diakses dari luar jaringan lokal. Metode yang digunakan dalam penelitian ini meliputi instalasi dan konfigurasi Laravel 11, pembuatan endpoint API menggunakan Laravel, serta integrasi dengan Ngrok untuk menguji aksesibilitas API. Hasil dari penelitian ini menunjukkan bahwa penggunaan Ngrok dapat mempermudah pengujian API yang dikembangkan secara lokal tanpa harus melakukan deploy ke server. Dengan demikian, integrasi Laravel 11 dan Ngrok dapat meningkatkan efisiensi dalam proses pengembangan API, khususnya bagi para pengembang yang ingin melakukan pengujian langsung dari berbagai perangkat atau layanan pihak ketiga secara cepat dan praktis.

Kata Kunci: API, Laravel 11, Ngrok, Tunneling, Pengembangan Web

Abstract

Laravel 11 is one of the most popular PHP frameworks used for developing web-based applications, including API (Application Programming Interface) development. APIs enable efficient communication between different applications. One of the challenges in API development is testing and accessing it from external environments, especially when the application is still in the development stage on a local server. Ngrok provides a solution by creating a tunnel that allows local APIs to be accessed publicly over the internet. This study aims to implement API development using Laravel 11 and configure it with Ngrok to enable external access. The methodology includes installing and configuring Laravel 11, creating API endpoints using Laravel, and integrating Ngrok to test API accessibility. The results of this study show that Ngrok facilitates the testing of locally developed APIs without requiring deployment to a server. Thus, integrating Laravel 11 and Ngrok enhances efficiency in API development, particularly for developers who need to test their APIs directly from various devices or third-party services quickly and practically.

Keywords: API, Laravel 11, Ngrok, Tunneling, Web Development

1. Introduction (Pendahuluan)

1.1 Latar belakang

Dalam era digital saat ini, pengembangan aplikasi berbasis web semakin berkembang, terutama dalam pembuatan dan pengelolaan API (Application Programming Interface). API memiliki peran penting dalam memungkinkan komunikasi antara berbagai aplikasi dan sistem, mendukung pertukaran data yang efisien serta integrasi dengan layanan pihak ketiga. Laravel, sebagai salah satu framework PHP yang populer, terus mengalami pembaruan, dengan Laravel 11 menghadirkan berbagai peningkatan dalam performa dan fitur yang mendukung pengembangan API secara lebih optimal. Salah satu tantangan dalam pengembangan API adalah pengujian dan aksesibilitas dari lingkungan eksternal, terutama ketika API masih dalam tahap pengembangan di server lokal. Ngrok hadir sebagai solusi dengan menyediakan tunneling yang memungkinkan API lokal diakses secara publik melalui internet tanpa perlu melakukan deploy ke server. Hal ini sangat bermanfaat bagi pengembang yang ingin melakukan pengujian langsung dari berbagai perangkat atau layanan eksternal. Dalam praktik ini, API dikembangkan menggunakan Laravel 11 dan diuji aksesibilitasnya menggunakan Ngrok. Proses ini mencakup instalasi dan konfigurasi Laravel 11, pembuatan endpoint API, serta integrasi dengan Ngrok untuk memastikan API dapat diakses dari luar jaringan lokal. Praktik ini diharapkan dapat memberikan wawasan kepada pengembang mengenai pentingnya penggunaan Ngrok dalam pengujian API serta meningkatkan efisiensi dalam proses pengembangan aplikasi berbasis web.

1.2 Tujuan eksperimen

1. Memahami konsep dasar API serta cara kerja Laravel 11 dalam mengelola dan menyajikan data melalui endpoint.
2. Mempelajari proses pembuatan dan pengujian API menggunakan Laravel 11, termasuk konfigurasi routing, controller, dan model.
3. Mengintegrasikan Ngrok untuk memungkinkan API yang dikembangkan secara lokal dapat diakses dari jaringan eksternal.
4. Menguji efektivitas Ngrok sebagai alat bantu dalam pengujian API tanpa perlu melakukan deploy ke server.
5. Menganalisis performa dan keamanan API yang dikembangkan serta mengevaluasi potensi implementasinya dalam sistem berbasis web dan aplikasi mobile.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat dan Bahan)

Alat

1. **Komputer/Laptop** – Digunakan untuk menginstal dan menjalankan Laravel 11 serta Ngrok.
2. **Koneksi Internet** – Diperlukan untuk mengunduh paket Laravel, menginstal dependensi, serta mengakses API melalui Ngrok.
3. **Laravel 11** – Framework PHP yang digunakan untuk mengembangkan API.
4. **Ngrok** – Alat tunneling yang digunakan untuk mengakses API lokal secara publik.
5. **Postman atau Insomnia** – Digunakan untuk menguji endpoint API yang telah dibuat.
6. **Visual Studio Code (VS Code)** – Sebagai code editor untuk menulis dan mengedit kode Laravel.
7. **Composer** – Manajer paket PHP yang digunakan untuk menginstal Laravel dan dependensinya.

Bahan

1. **Kode Program API Laravel** – Skrip yang berisi endpoint untuk menangani permintaan data.

2. **Database MySQL atau SQLite** – Digunakan sebagai penyimpanan data yang akan diakses melalui API.
3. **Environment File (.env)** – Digunakan untuk menyimpan konfigurasi aplikasi Laravel seperti database dan pengaturan Ngrok.
4. **JSON Response** – Format data yang dikembalikan oleh API untuk dikonsumsi oleh aplikasi klien.

2.2 Implementation Steps (Langkah Implementasi)

Adapun Langkah-langkah implementasi dalam pembuatan API Laravel menggunakan ngrok

1. Instalasi dan Konfigurasi Laravel 11

1. Pastikan PHP 8.2+ dan Composer telah terinstal.
2. Buat proyek Laravel baru dengan perintah berikut:
`composer create-project laravel/laravel api-laravel11`
3. Masuk ke direktori proyek:
`cd api-laravel11`
4. Jalankan Laravel menggunakan server bawaan:
`php artisan serve`

2. Pembuatan Database dan Migrasi

1. Buat database baru (misalnya, `api_database`) pada MySQL atau SQLite.
2. Konfigurasi file `.env` untuk koneksi database:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=iot_25
DB_USERNAME=root
DB_PASSWORD=
```

```
DB_CHARSET=utf8mb4
DB_COLLATION=utf8mb4_unicode_ci
```

3. Jalankan migrasi database:
`php artisan migrate`

3. Pembuatan Model, Controller, dan Routing API

1. Buat model beserta migration untuk tabel `users` atau `produk` (contoh untuk `produk`):
`php artisan make:model Produk -m`

2. Edit file migration (database/migrations/xxxx_xx_xx_XXXXXX_create_xxxx_table.php) untuk menyesuaikan skema tabel, lalu jalankan migrasi ulang:

php artisan migrate

3. Buat controller untuk API:

php artisan make:controller xxxxController --api

4. Tambahkan metode untuk CRUD di xxxxController.php, contoh:

```
<?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($transaksiSensor);
}

```

```

/**
 * 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);
}

}

```

4. Menjalankan dan Menguji API

1. Tambahkan routing API di routes/api.php:
use App\Http\Controllers\ProdukController;
Route::apiResource('/posts', App\Http\Controllers\Api\TransaksiSensorController::class);
2. Jalankan kembali server Laravel:
php artisan serve
3. Uji endpoint API menggunakan Postman atau browser dengan mengakses
<http://127.0.0.1:8000/api/posts>.

5. Integrasi dengan Ngrok

1. Unduh dan instal Ngrok dari ngrok.com.
2. Jalankan perintah berikut untuk membuka tunnel ke server Laravel:
ngrok http 8080

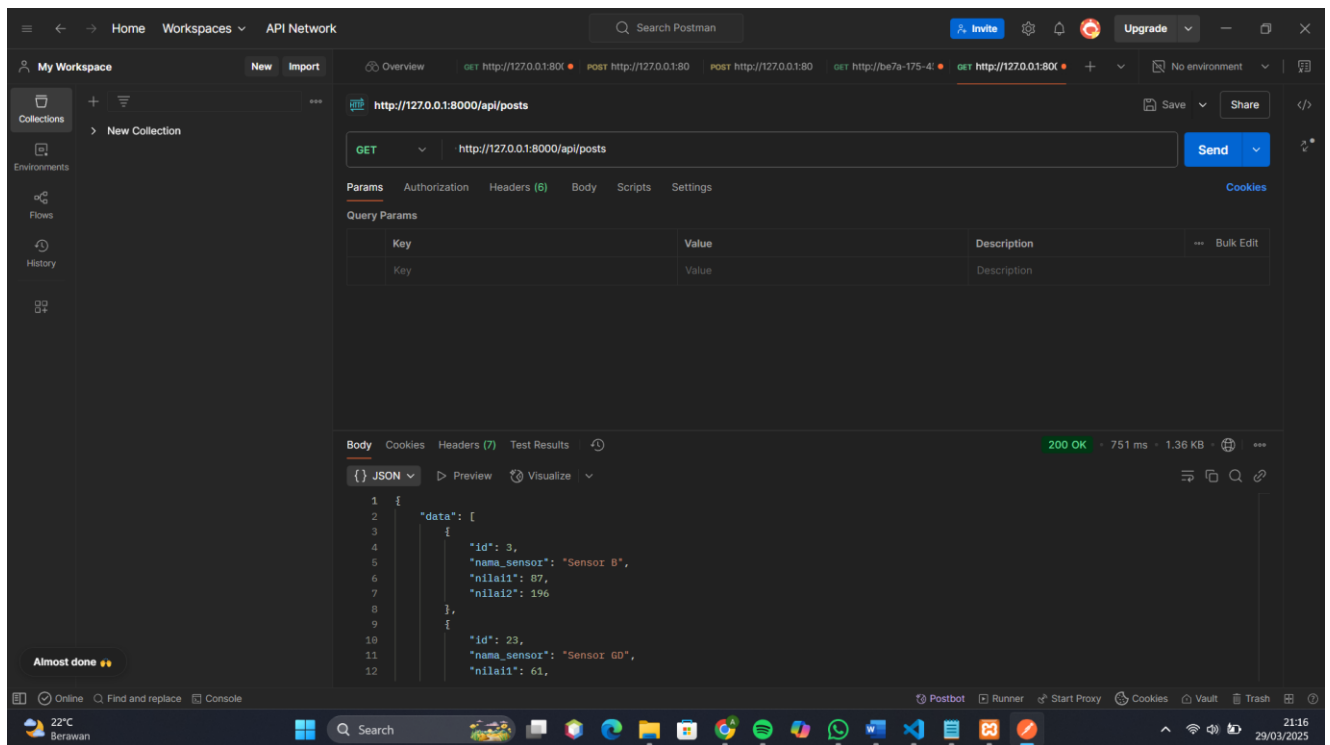
3. Ngrok akan memberikan URL publik seperti <https://randomstring.ngrok.io>, yang bisa digunakan untuk mengakses API secara eksternal.

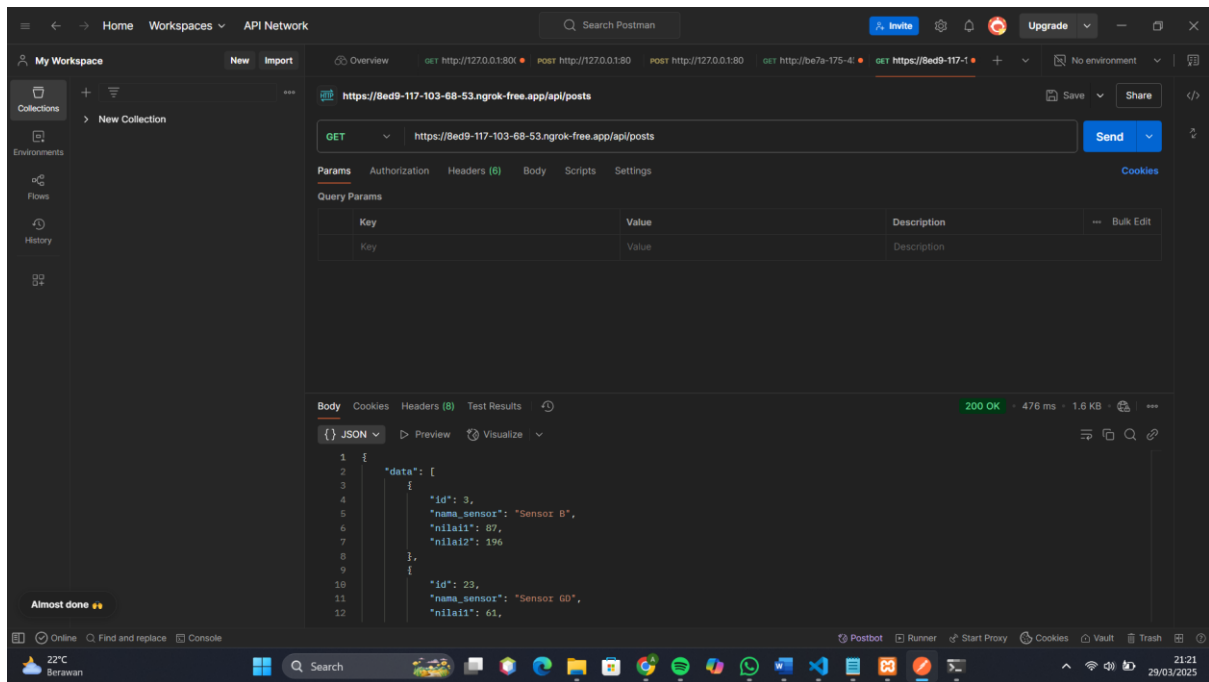
6. Pengujian API dengan Ngrok

- Gunakan URL Ngrok yang diberikan untuk mengakses API dari Postman atau perangkat lain:
- Pastikan API berjalan dengan baik dengan menguji berbagai metode seperti GET, POST, PUT, dan DELETE.

3. Results and Discussion (Hasil dan Pembahasan)

3.1 Experimental Results (Hasil Eksperimen)





Gambar menunjukkan pengujian API menggunakan Postman dengan metode **GET** pada endpoint yang diakses melalui **Ngrok**. API mengembalikan respons **200 OK**, menandakan permintaan berhasil. Penggunaan Ngrok memungkinkan API diakses secara publik, mempermudah pengujian dari berbagai perangkat.

4. Appendix (Lampiran)

```
app > Http > Controllers > Api > TransaksiSensorController.php > TransaksiSensorController
1 <?php
2
3
4 namespace App\Http\Controllers\Api;
5 use Illuminate\Http\Request;
6
7
8 use App\Models\TransaksiSensor;
9 use App\Http\Controllers\Controller;
10 use App\Http\Resources\TransaksiSensorResource;
11
12
13 class TransaksiSensorController extends Controller
14 {
15     /**
16      * index
17      *
18      * @return \Illuminate\Http\Response
19      */
20     public function index()
21     {
22         // Get all transactions from TransaksiSensor model, paginated
23         $transaksiSensors = TransaksiSensor::latest()->paginate(5);
24
25         // Return a collection of transactions as a resource
26         return TransaksiSensorResource::collection($transaksiSensors);
27     }
28
29     /**
30      * Store a newly created resource in storage.
31      *
32      * @param \Illuminate\Http\Request $request
33      * @return \Illuminate\Http\Response
34      */
35     public function store(Request $request)
36     {
37
```