

Praktik Pembuatan API Menggunakan Laravel 11 & NGROK

Ardan Pramudya

Fakultas Vokasi, Universitas Brawijaya

Email : ardanpramudya91@gmail.com

Abstract (Abstrak)

Praktikum ini bertujuan untuk merancang dan mengimplementasikan pembuatan API (Application Programming Interface) menggunakan framework Laravel 11 dan Ngrok. API berfungsi sebagai antarmuka yang memungkinkan komunikasi antara aplikasi dengan server. Laravel 11 dipilih karena kemampuannya dalam memudahkan pengembangan aplikasi web dan API, sementara Ngrok digunakan untuk membuat tunnel yang memungkinkan akses lokal ke internet. Hasil eksperimen menunjukkan bahwa API yang dibuat dapat berfungsi dengan baik dan dapat diakses melalui internet menggunakan Ngrok.

1.1 Latar Belakang

Pengembangan API menjadi semakin penting dalam era digital saat ini, terutama untuk menghubungkan berbagai aplikasi dan layanan. Laravel 11 sebagai framework PHP yang populer, menyediakan berbagai fitur yang memudahkan pengembangan API. Ngrok, di sisi lain, memungkinkan pengembang untuk mengekspos server lokal ke internet, sehingga memudahkan pengujian dan debugging. Dalam praktikum ini, API sederhana akan dibuat menggunakan Laravel 11 dan diakses melalui internet menggunakan Ngrok.

1.2 Tujuan Eksperimen

Merancang dan mengimplementasikan pembuatan API sederhana menggunakan Laravel 11.

Menggunakan Ngrok untuk mengekspos server lokal ke internet.

Menguji fungsionalitas API yang dibuat dan memastikan dapat diakses melalui internet.

2. METHODOLOGY

2.1 Tools & Materials (Alat dan Bahan)

1. **Laravel 11** sebagai framework pengembangan API.
2. **Ngrok** untuk membuat tunnel dan mengekspos server lokal ke internet.
3. **Composer** untuk manajemen dependensi Laravel.
4. **Postman** untuk pengujian API.
5. **Text Editor** (Visual Studio Code atau sejenisnya) untuk penulisan kode.
6. **XAMPP** atau Laravel Sail sebagai server lokal.

2.2 Implementation Steps (Langkah Implementasi)

1. **Persiapan Lingkungan Pengembangan**
Menginstal Composer dan Laravel 11.
Membuat proyek Laravel baru.
Menginstal Ngrok dan memastikan dapat dijalankan di terminal.
2. **Pembuatan API Sederhana**
Membuat controller untuk API.
Menambahkan route API di file routes/api.php.
Mengimplementasikan method untuk mengembalikan data sederhana.
3. **Menjalankan Server Lokal**
Menjalankan server Laravel.
Server akan berjalan di `http://127.0.0.1:8000`.
4. **Mengekspos Server Lokal Menggunakan Ngrok**
Menjalankan Ngrok untuk mengekspos server lokal ke internet.
Ngrok akan memberikan URL publik yang dapat diakses dari internet.
5. **Pengujian API Menggunakan Postman**
Membuka Postman dan mengirim request GET ke URL yang diberikan oleh Ngrok.
Memastikan response dari API sesuai dengan yang diharapkan.

3.1 Hasil Eksperimen

API berhasil dibuat menggunakan Laravel 11 dan dapat diakses melalui server lokal. Ngrok berhasil mengekspos server lokal ke internet, sehingga API dapat diakses dari luar jaringan lokal.

Pengujian menggunakan Postman menunjukkan bahwa API merespons dengan data yang sesuai.

3.2 Pembahasan

Faktor yang Mempengaruhi Kinerja:

Kualitas Kode: Kode yang terstruktur dan efisien akan mempengaruhi kinerja API.

Koneksi Internet: Koneksi internet yang stabil diperlukan untuk mengakses API melalui Ngrok.

Konfigurasi Ngrok: Pengaturan yang tepat pada Ngrok akan memastikan server lokal dapat diakses dengan baik.

The screenshot displays a development environment with two main windows. The top window is Visual Studio Code, showing a Laravel 11 project. The Explorer sidebar on the left lists the project structure, including directories like 'database', 'migrations', 'seeder', and 'resources', and files like 'api.php', 'console.php', 'web.php', 'storage', 'tests', 'vendor', '.editorconfig', '.env', '.env.example', '.gitattributes', '.gitignore', 'artisan', 'composer.json', 'composer.lock', 'package.json', 'phpunit.xml', 'postcss.config.js', 'README.md', 'tailwind.config.js', and 'vite.config.js'. The main editor area shows the '.env' file with configuration for a Laravel application, including database connection details (MySQL, host 127.0.0.1, port 3306, database 'iot_25', user 'root', password 'root'). The bottom window is a web browser displaying the phpMyAdmin interface. The browser's address bar shows the URL '127.0.0.1:8000/phpmyadmin/index.php?route=/database/structure&db=iot_25'. The phpMyAdmin interface shows the 'Structure' tab for the 'iot_25' database. It lists 11 tables: 'cache', 'cache_locks', 'failed_jobs', 'jobs', 'job_batches', 'migrations', 'password_reset_tokens', 'personal_access_tokens', 'sessions', 'transaksi_sensor', and 'users'. Each table entry includes icons for actions like 'Browse', 'Structure', 'Search', 'Insert', 'Empty', and 'Drop'. The 'Sum' row at the bottom indicates 11 tables, 6 InnoDB engines, and a total size of 288.0 KiB.





