**Praktik Pembuatan API Menggunakan Laravel 11 dan Ngrok Disusun Untuk Memenuhi Tugas Mata Kuliah :**

**Internet Of Things**

**Dosen Pengampu : Rachmad Andri Atmoko, S.ST, M.T, MCF**



**Disusun Oleh :**

Daffa Freda Muhammad Nehru

233140707111012

T4F

PROGRAM STUDI TEKNOLOGI INFORMASI(

FAKULTAS VOKASI

UNIVERSITAS BRAWIJAYA

2025

**1. PENDAHULUAN**

**1.1 Latar Belakang**

Internet of Things (IoT) merupakan konsep teknologi yang menghubungkan perangkat-perangkat fisik ke dalam jaringan internet untuk saling bertukar data. Dalam pengembangan sistem IoT, kebutuhan akan komunikasi data yang cepat dan efisien menjadi sangat penting. Salah satu metode komunikasi yang umum digunakan adalah melalui RESTful API.

Laravel merupakan salah satu framework PHP yang banyak digunakan dalam pengembangan aplikasi web, termasuk untuk pembuatan API. Versi terbaru Laravel 11 menawarkan berbagai fitur yang mempermudah proses pengembangan, salah satunya dalam hal manajemen API. Selain itu, untuk keperluan pengujian dan integrasi dengan perangkat IoT seperti ESP32, diperlukan layanan tunneling seperti Ngrok agar API yang berjalan secara lokal dapat diakses secara global melalui internet.

Praktikum ini bertujuan untuk memperkenalkan proses pembuatan API sederhana menggunakan Laravel 11 dan menghubungkannya ke jaringan internet menggunakan Ngrok. Dengan cara ini, API yang dibuat dapat diakses oleh perangkat-perangkat IoT secara real time, baik melalui jaringan lokal maupun jaringan publik.

**1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, rumusan masalah dalam praktikum ini adalah:

1. Bagaimana cara membuat API menggunakan Laravel 11?
2. Bagaimana cara menghubungkan API lokal agar dapat diakses melalui internet menggunakan Ngrok?
3. Bagaimana API tersebut dapat digunakan untuk membaca dan menyimpan data dari atau ke perangkat IoT?

**1.3 Tujuan Praktikum**

Tujuan dari praktikum ini adalah sebagai berikut:

1. Mempelajari proses pembuatan RESTful API menggunakan Laravel 11.
2. Mampu mengonfigurasi dan menjalankan API Laravel agar terhubung dengan database MySQL.
3. Mengimplementasikan Ngrok untuk meng-online-kan API lokal ke jaringan publik.
4. Menguji konektivitas dan fungsionalitas API melalui aplikasi pihak ketiga seperti Postman.

**1.4 Manfaat Praktikum**

**Adapun manfaat dari praktikum ini adalah:**

* **Memberikan pemahaman praktis tentang pengembangan API dengan Laravel.**
* **Mengetahui cara menghubungkan sistem lokal dengan dunia luar melalui layanan tunneling (Ngrok).**
* **Menjadi dasar pengembangan sistem IoT berbasis web yang terhubung secara real-time.**
* **Meningkatkan keterampilan dalam integrasi backend dan frontend, serta komunikasi antara sistem dan perangkat IoT.**

**2. Alat dan Bahan**

* Laravel 11 (PHP Framework)
* Composer (Dependency Manager PHP)
* XAMPP / phpMyAdmin (MySQL Database)
* Ngrok (Public Tunneling Service)
* Postman (API Client)
* Visual Studio Code (Editor)
* Command Prompt / Terminal
* Browser

**3. Langkah-Langkah Praktikum**

**3.1 Setup Laravel Project**

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

cd laravel-11

**3.2 Konfigurasi Database**

Buat database di phpMyAdmin dengan nama: iot\_25

Ubah file .env:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=iot\_25

DB\_USERNAME=root

DB\_PASSWORD=

**3.3 Membuat Model dan Migration**

php artisan make:model TransaksiSensor -m

Ubah file migrasi create\_transaksi\_sensors\_table.php:

Schema::create('transaksi\_sensor', function (Blueprint $table) {

$table->id('id')->startingValue(1);

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

$table->integer('nilai1');

$table->integer('nilai2');

$table->timestamps();

});

Jalankan migrasi:

php artisan migrate

**3.4 Model Laravel**

Ubah file app/Models/TransaksiSensor.php:

class TransaksiSensor extends Model

{

use HasFactory;

protected $table = 'transaksi\_sensor';

protected $fillable = ['nama\_sensor', 'nilai1', 'nilai2'];

}

**3.5 Membuat Resource**

php artisan make:resource TransaksiSensorResource

Ubah isi file TransaksiSensorResource.php:

public function toArray($request)

{

return [

'id' => $this->id,

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

'nilai1' => $this->nilai1,

'nilai2' => $this->nilai2,

];

}

**3.6 Membuat Controller API**

php artisan make:controller Api/TransaksiSensorController

Isi controller:

* index() untuk get data
* store() untuk insert data
* show() untuk melihat data by ID
* update() untuk update data
* destroy() untuk delete data

**3.7 Menambahkan API Routes**

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

Cek routes:

php artisan route:list

**3.8 Pengujian API dengan Postman**

1. Jalankan Laravel:

php artisan serve

1. Buka Postman, masukkan URL:
   * GET: <http://127.0.0.1:8000/api/posts>
   * POST: Tambahkan header Content-Type: application/json, lalu body:

{

"nama\_sensor": "Sensor A",

"nilai1": 100,

"nilai2": 200

}

Hasil:

{

"data": [

{

"id": 1,

"nama\_sensor": "Sensor A",

"nilai1": 100,

"nilai2": 200

}

]

}

**4. Meng-online-kan API dengan Ngrok**

**4.1 Jalankan Laravel pada Port 8080**

php artisan serve --host=0.0.0.0 --port=8080

**4.2 Jalankan Ngrok**

ngrok http --scheme=http 8080

Ngrok akan memberikan URL seperti:

http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app

Akses:

* GET: http://<NGROK\_URL>/api/posts
* POST: http://<NGROK\_URL>/api/posts

**s5. Hasil dan Pembahasan**

**5.1 Hasil Output Postman**

* Kode status: 200 OK menandakan API berhasil diakses.
* Insert data berhasil menambahkan record baru ke tabel transaksi\_sensor.
* Data dapat diakses dari URL publik Ngrok.

**5.2 Pembahasan**

* API berhasil dibuat menggunakan Laravel 11.
* API dapat diakses dari mana saja melalui link Ngrok.
* API sudah siap digunakan oleh device seperti ESP32 atau simulasi Wokwi.

**6. Kesimpulan**

* Laravel 11 dapat digunakan untuk membuat REST API sederhana.
* Dengan Ngrok, API lokal dapat diakses secara publik.
* API telah diuji dengan Postman dan berhasil mengelola data CRUD dari database MySQL.

**7. Dokumentasi**

**Tampilan API di Postman (GET)**

**Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.**

**Tampilan API Insert (POST)**

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

**Tampilan phpMyAdmin**

Sebuah gambar berisi teks, software, Laman internet, Ikon komputer

Konten yang dihasilkan AI mungkin salah.

**Tampilan Ngrok**

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.