**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**

**Fakultas Vokasi, Universitas Brawijaya**

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

*Naila Sahda Aulya*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: naila.sahda18@gmail.com*

**ABSTRAK**

Praktik pembuatan API merupakan salah satu keterampilan penting dalam pengembangan perangkat lunak modern. Laporan ini membahas proses pembuatan API menggunakan Laravel 11 serta integrasinya dengan Ngrok untuk pengujian dan akses jarak jauh. Laravel 11 sebagai framework PHP terbaru menawarkan berbagai fitur yang mempermudah pengembangan API, seperti routing yang lebih sederhana, middleware yang fleksibel, dan dukungan bawaan untuk autentikasi. Ngrok digunakan untuk membuat tunneling yang memungkinkan API diakses secara publik tanpa perlu konfigurasi server yang kompleks. Dalam praktik ini, API dibuat dengan menerapkan metode HTTP seperti GET, POST, PUT, dan DELETE, serta menggunakan database sebagai penyimpanan data. Hasil yang diperoleh menunjukkan bahwa Laravel 11 dan Ngrok dapat digunakan secara efisien dalam pengembangan API, mempercepat proses pengujian dan distribusi layanan.

**Kata Kunci**: API, Laravel 11, Ngrok, Pengembangan Web, Autentikasi

**1. Introduction (Pendahuluan)**

**1.1 Latar belakang praktikum IoT yang dilakukan**

Dalam era digital saat ini, pengembangan API (Application Programming Interface) menjadi aspek krusial dalam pembuatan aplikasi berbasis web dan mobile. API memungkinkan aplikasi untuk berkomunikasi dengan sistem lain secara efisien, mempercepat integrasi data, dan meningkatkan skalabilitas layanan. Laravel, sebagai salah satu framework PHP yang populer, terus berkembang untuk memenuhi kebutuhan pengembang dalam membangun API yang aman dan mudah dikelola. Laravel 11 hadir dengan berbagai peningkatan fitur, termasuk sistem routing yang lebih sederhana, middleware yang lebih fleksibel, dan peningkatan dalam sistem autentikasi.

Sementara itu, Ngrok merupakan alat yang memungkinkan pengembang untuk mengakses aplikasi lokal melalui internet tanpa harus melakukan konfigurasi server secara manual. Dengan Ngrok, API yang dikembangkan dapat diuji dan diakses oleh pihak lain secara lebih mudah, mendukung proses pengembangan yang lebih cepat dan efisien. Oleh karena itu, praktik ini dilakukan untuk memahami bagaimana cara membangun API menggunakan Laravel 11 serta mengintegrasikannya dengan Ngrok guna mempermudah pengujian dan distribusi layanan.

**1.2 Tujuan eksperimen**

Adapun tujuan dari praktik ini adalah sebagai berikut:

1. Mempelajari konsep dasar dan implementasi API menggunakan Laravel 11.
2. Mengimplementasikan metode HTTP seperti GET, POST, PUT, dan DELETE dalam API.
3. Mengintegrasikan API dengan database untuk penyimpanan dan pengelolaan data.
4. Menggunakan Ngrok untuk memungkinkan API dapat diakses secara publik untuk pengujian dan pengembangan lebih lanjut.
5. Mengevaluasi efektivitas Laravel 11 dan Ngrok dalam proses pembuatan dan pengujian API.

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

Dalam eksperimen ini, alat dan bahan yang digunakan meliputi:

Tools (Peralatan):

1. **Komputer/Laptop** – Digunakan sebagai perangkat utama untuk mengembangkan API.
2. **Text Editor/IDE (Integrated Development Environment)** – Contoh: Visual Studio Code atau PHPStorm untuk menulis dan mengelola kode.
3. **Terminal/Command Prompt** – Digunakan untuk menjalankan perintah Laravel dan Ngrok.
4. **Postman atau cURL** – Digunakan untuk menguji endpoint API.
5. **Browser Web** – Digunakan untuk mengecek hasil API jika menggunakan respons berbasis JSON.

Materials (Bahan):

1. **Laravel 11** – Framework PHP yang digunakan untuk membuat API.
2. **PHP 8.2 atau versi yang lebih baru** – Dibutuhkan sebagai runtime untuk Laravel 11.
3. **Composer** – Dependency manager untuk PHP yang digunakan untuk menginstal Laravel.
4. **Database (MySQL atau SQLite)** – Digunakan untuk menyimpan data API.
5. **Ngrok** – Alat untuk membuat tunneling agar API dapat diakses secara publik.
6. **PostgreSQL/MariaDB (opsional)** – Alternatif database jika tidak menggunakan MySQL.

**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

1. Masuk ke direktori proyek:

cd api-laravel11

1. 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. Konfigurasikan 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

1. 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

1. Edit file migration (database/migrations/xxxx\_xx\_xx\_xxxxxx\_create\_xxxx\_table.php) untuk menyesuaikan skema tabel, lalu jalankan migrasi ulang:

php artisan migrate

1. Buat controller untuk API:

php artisan make:controller xxxxController --api

1. 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);

1. Jalankan kembali server Laravel:

php artisan serve

1. 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](https://ngrok.com).
2. Jalankan perintah berikut untuk membuka tunnel ke server Laravel:

ngrok http 8000

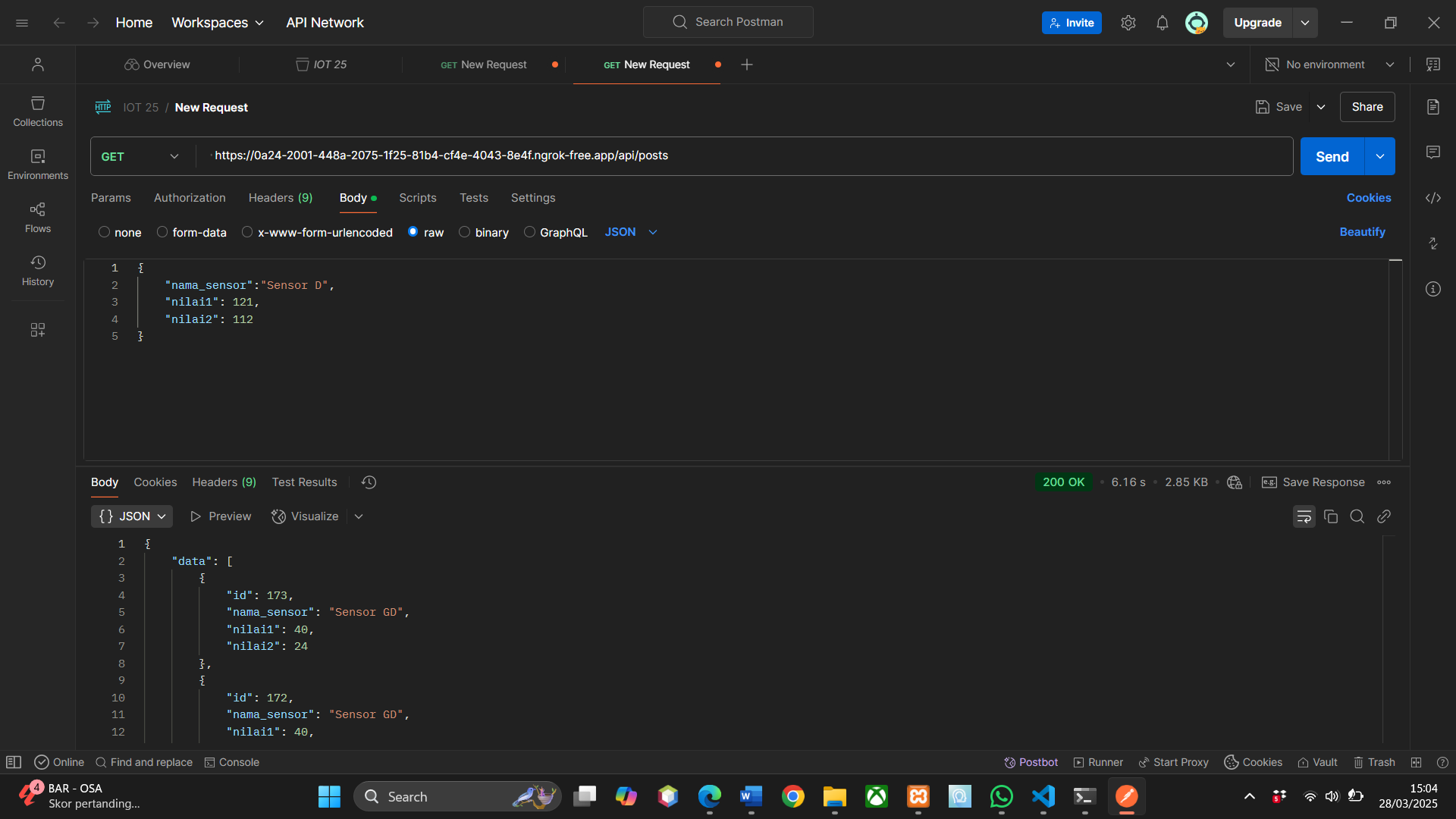
1. Ngrok akan memberikan URL publik seperti https://randomstring.ngrok.io, yang bisa digunakan untuk mengakses API secara eksternal.

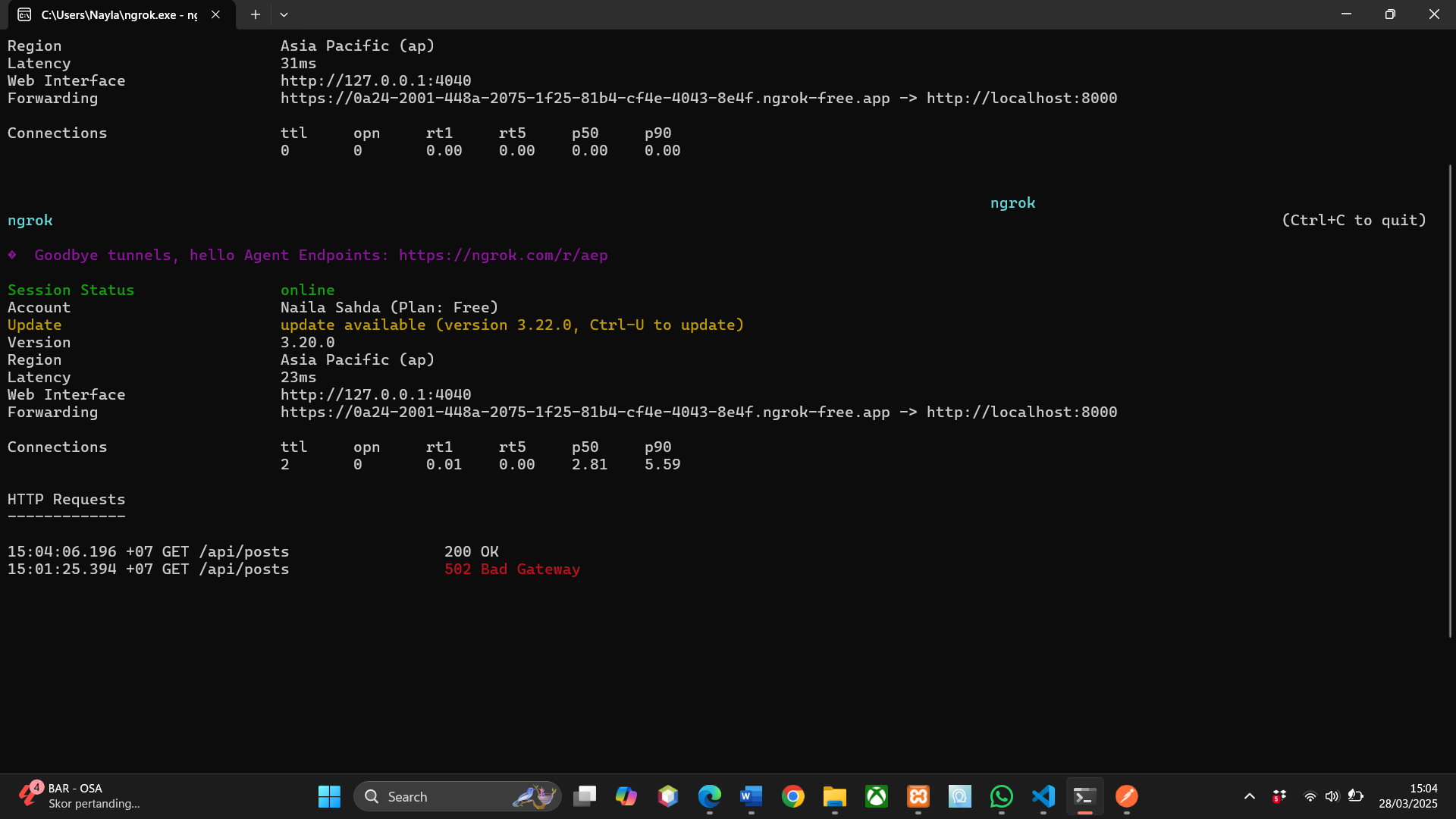
**6. Pengujian API dengan Ngrok**

* Gunakan URL Ngrok yang diberikan untuk mengakses API dari Postman atau perangkat lain:
* https://0a24-2001-448a-2075-1f25-81b4-cf4e-4043-8e4f.ngrok-free.app/api/postsPastikan 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. Data yang diterima dalam format JSON berisi informasi sensor dengan atribut id, nama\_sensor, nilai1, dan nilai2. Penggunaan Ngrok memungkinkan API diakses secara publik, mempermudah pengujian dari berbagai perangkat.

**4. Appendix (Lampiran)**

