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

Fakultas Vokasi , Universitas Brawijaya

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

*Iffah Alayya Azzahro*

*Fakultas Vokasi, Universitas Brawijaya*

*iffaazzahro@gmail.com*

**Abstrak**

Dalam pengembangan aplikasi berbasis web, API (Application Programming Interface) menjadi komponen penting yang memungkinkan komunikasi antara berbagai sistem. Praktikum ini bertujuan untuk mengimplementasikan dan menguji pembuatan API menggunakan Laravel 11 serta mengaksesnya melalui Ngrok. Laravel 11 dipilih karena kemudahan penggunaannya dalam membangun API yang efisien dan terstruktur, sementara Ngrok digunakan untuk menghubungkan API lokal ke internet secara aman. Pemrograman dilakukan menggunakan Visual Studio Code (VS Code) dengan bahasa pemrograman PHP. Hasil praktikum menunjukkan bahwa API yang dikembangkan dapat diakses dengan baik melalui endpoint yang dibuat, serta dapat diuji menggunakan Postman atau alat sejenis. Praktikum ini membantu mahasiswa memahami konsep dasar pembuatan API, penggunaan Laravel dalam pengembangan backend, serta penerapan Ngrok dalam mengakses API secara publik. Kesimpulannya, praktik ini memberikan pengalaman langsung dalam mengembangkan API yang dapat digunakan untuk berbagai keperluan, seperti aplikasi mobile, IoT, atau integrasi sistem berbasis web.

*Keyword : Internet of Things (IoT), API, Laravel, Ngrok*

1. **Introduction (Pendahuluan)**
   1. Latar Belakang

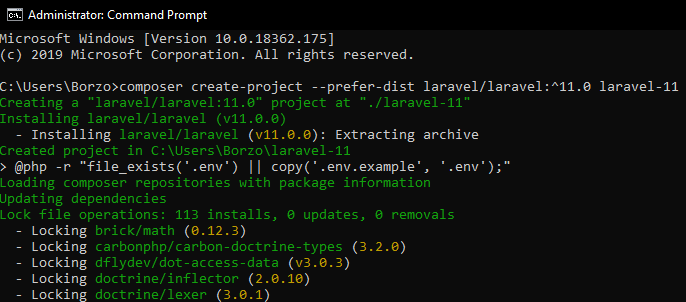
Perkembangan teknologi digital mendorong kebutuhan akan sistem yang dapat saling terhubung dan bertukar data secara efisien. API menjadi salah satu solusi utama dalam pengembangan sistem modern, memungkinkan aplikasi untuk berkomunikasi satu sama lain tanpa harus saling bergantung secara langsung pada struktur internal masing-masing. Laravel, sebagai salah satu framework PHP yang populer, menyediakan berbagai fitur yang memudahkan pembuatan API yang terstruktur dan aman.

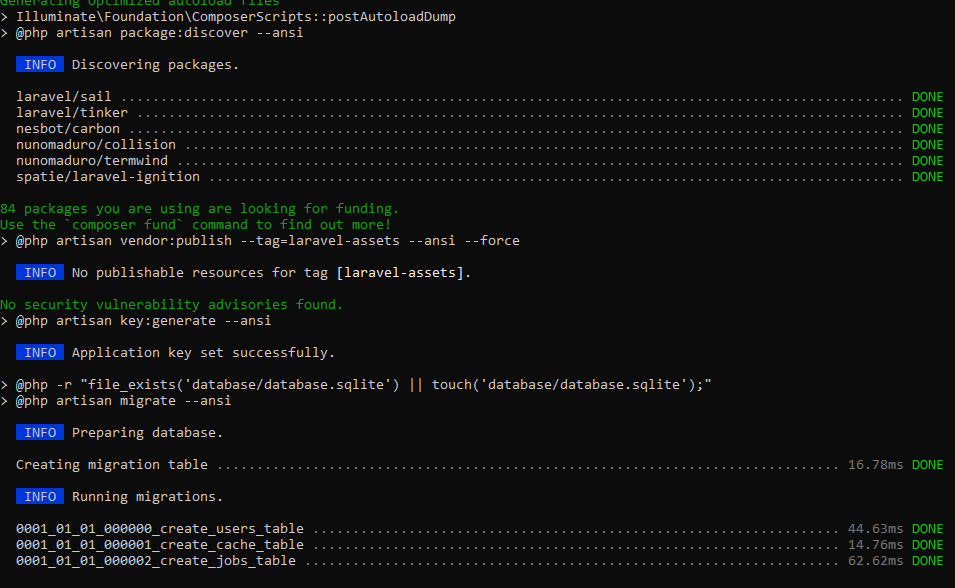
Ngrok digunakan dalam praktikum ini untuk menghubungkan API yang berjalan di lingkungan lokal agar dapat diakses melalui internet tanpa perlu konfigurasi server yang kompleks. Dengan menggunakan Laravel 11 dan Ngrok, mahasiswa mendapatkan pengalaman langsung dalam membangun API, mulai dari perancangan rute, pengolahan data, hingga pengujian menggunakan Postman. Praktikum ini bertujuan untuk membekali mahasiswa dengan pemahaman mengenai pengembangan backend berbasis Laravel serta penerapan teknologi tunneling seperti Ngrok, yang bermanfaat dalam pengembangan dan pengujian API sebelum diterapkan dalam skala yang lebih luas.

Praktikum ini bertujuan untuk memberikan pemahaman mendalam mengenai pembuatan API menggunakan Laravel, serta bagaimana mengakses API secara publik dengan bantuan Ngrok. Mahasiswa akan mendapatkan pengalaman langsung dalam mengembangkan backend berbasis Laravel, memahami mekanisme komunikasi API, serta mengimplementasikan metode HTTP seperti GET, POST, PUT, dan DELETE dalam pengolahan data.

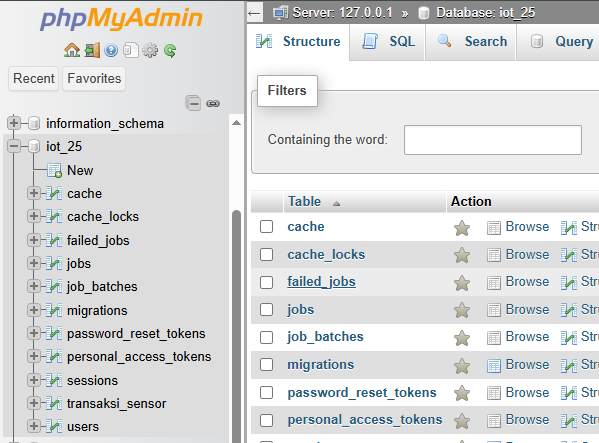
* 1. Tujuan Eksperimen

1. Memahami konsep dasar serta cara membuat dan mengelola API menggunakan framework Laravel 11.
2. Menguji dan menerapkan Ngrok sebagai solusi tunneling untuk mengakses API secara publik dari lingkungan pengembangan lokal.
3. Menguji koneksi API yang dibuat dengan Laravel 11 agar dapat diakses dan digunakan oleh aplikasi klien, seperti frontend atau perangkat IoT.
4. **Methodology (Metodologi)**
   1. Tools & Materials (Alat dan Bahan)
5. Laptop
6. Internet
7. Visual Studio Code
8. Website ( Wokwi.com)
9. Postman
10. Ngrok
11. MySQL
    1. Langkah Implementasi
12. Pembuatan project menggunakan Laravel 11
13. Mendownload paket-paket yang dibutuhkan dengan mengetikkan command berikut pada cmd **composer create-project --prefer-dist laravel/laravel:^11.0 laravel-11**

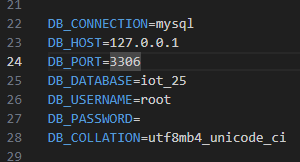




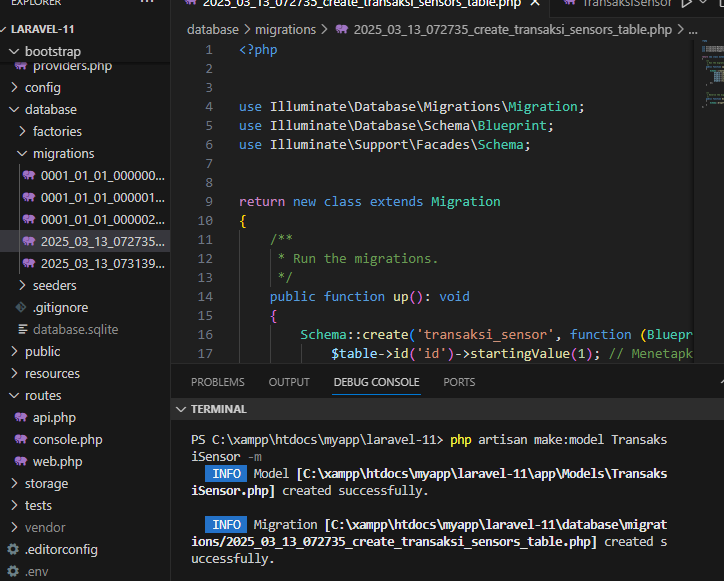
1. Setelah project berhasil dibuat, ketik **cd Laravel-11** untuk memindah direktori dan ketik **code .** untuk memulai membuka project di VSCode
2. Membuat database di phpMyAdmin dengan nama **iot\_25.**



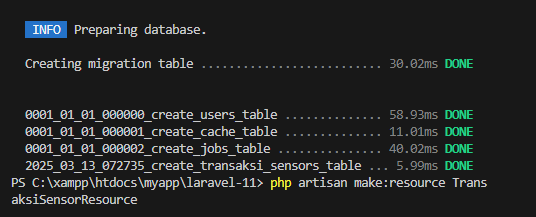
1. Mengubah isi file .env



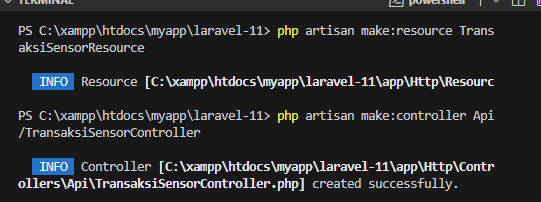
1. Membuat file model  **TransaksiSensor.php** dengan menjalankan perintah berikut di terminal > **php artisan make:model TransaksiSensor -m ,** lalu ubah isinya dengan code yang ada di modul



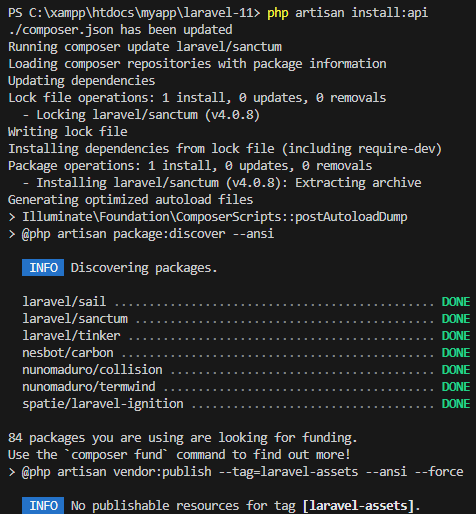
1. Kemudian ubah isi file **app/Models/TransaksiSensor.php** dengan kode yang ada di modul
2. Kemudian jalankan perintah berikut untuk membuat tabel :  **php artisan migrate**



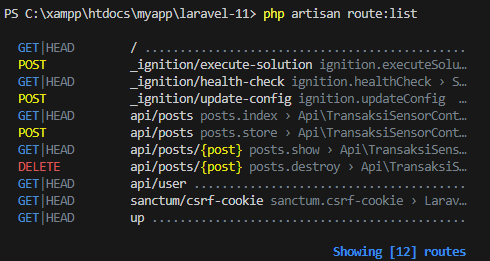
1. Membuat Resource dengan menjalankan perintah : **php artisan make:resource TransaksiSensorResource** ,lalu mengubah isi file tersebut dengan code di modul
2. Buat API controller dengan menjalankan perintah : **php artisan make:controller Api/TransaksiSensorController** ,lalu mengubah isi file tersebut dengan code di modul



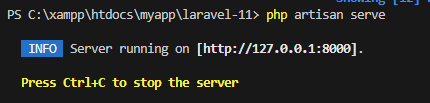
1. Buat route khusus API dengan menjalankan perintah :  **php artisan install:api** ,lalu mengubah isi file tersebut dengan code di modul



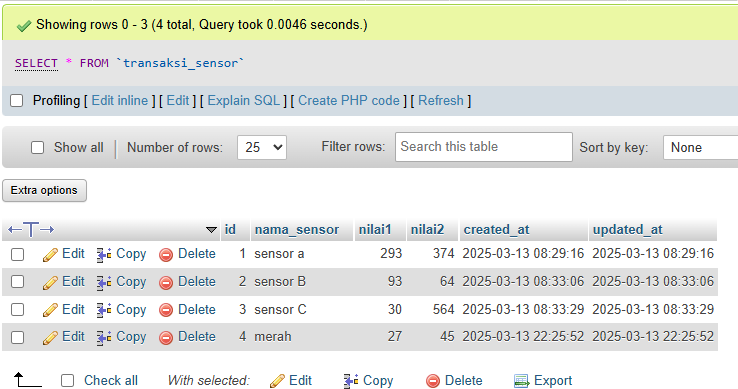
1. Kemudian pastikan routes telah terbentuk dengan menjalankan perintah : **php artisan route:list**



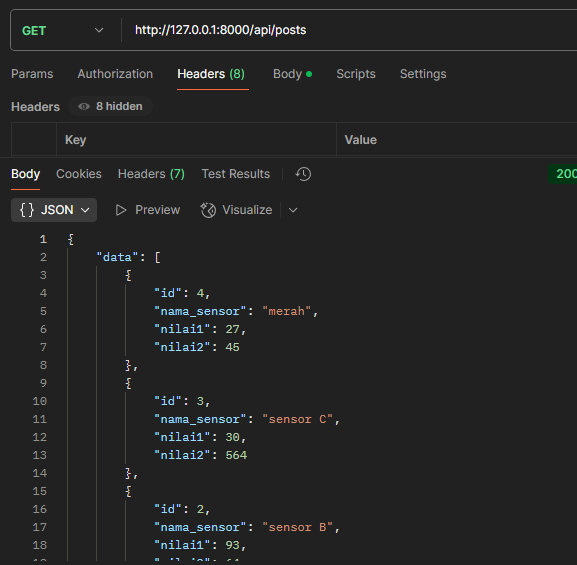
1. Melakukan percobaan akses api, pastikan aplikasi laravel dijalankan dengan perintah **php artisan serve**

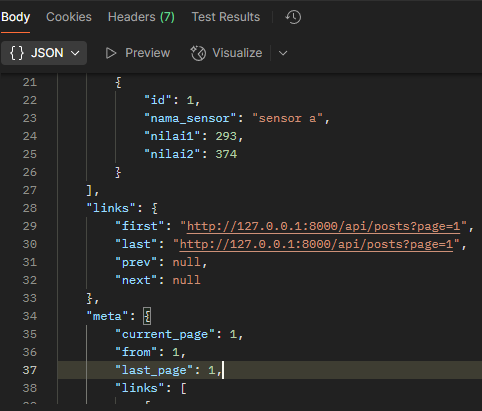


1. Lakukan testing menggunakan postman
2. Pastikan database sudah terisi seperti ini :

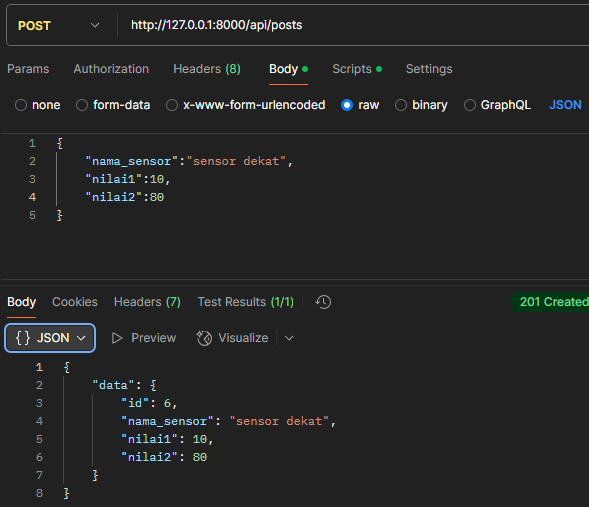


1. Kemudian salin alamat server laravel ke postman dan pilih method **GET**, lalu klik **SEND** seperti dibawah ini :

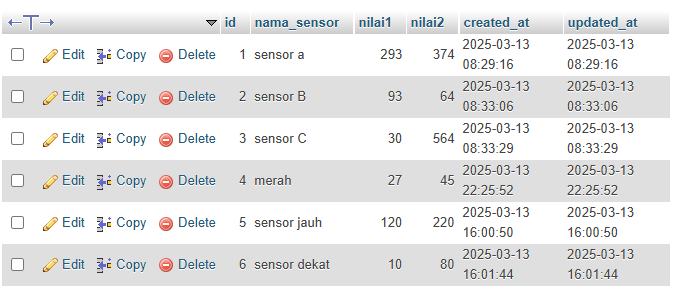




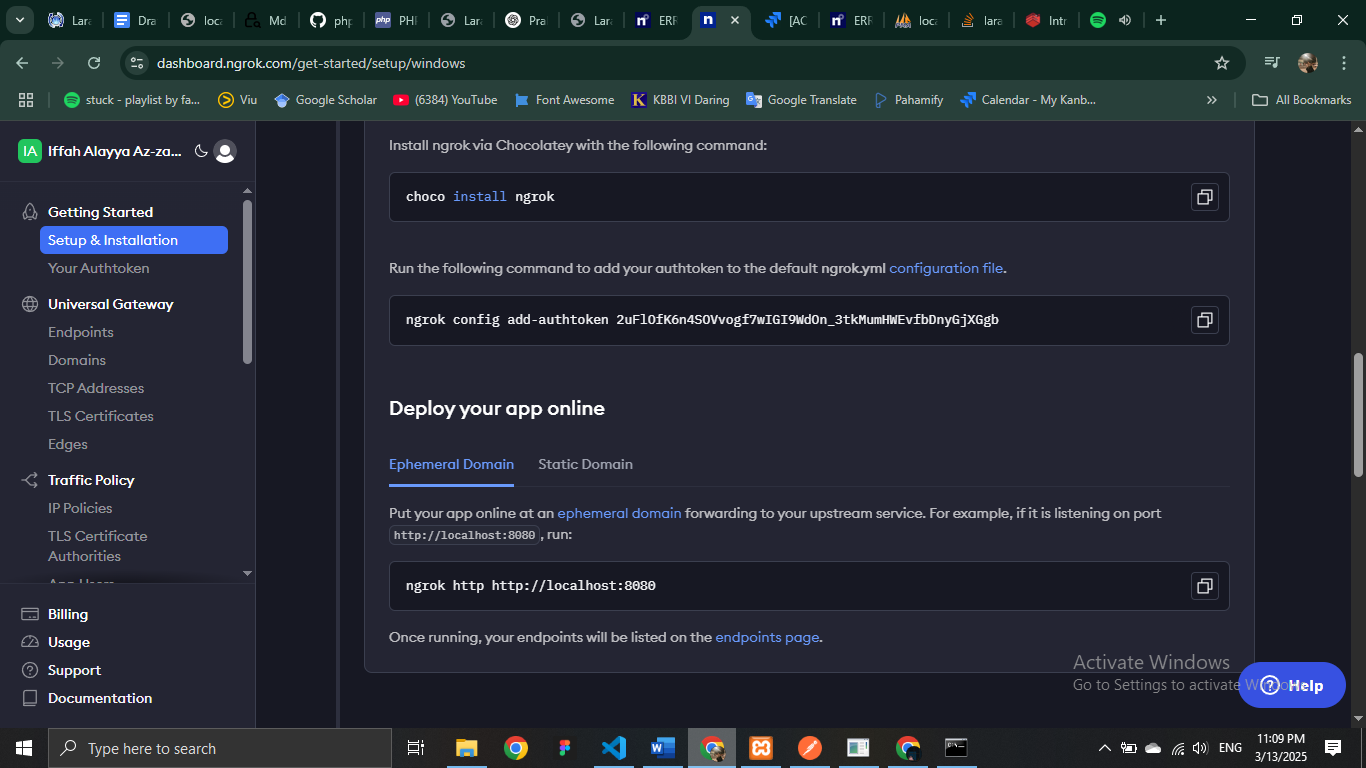
1. Selanjutnya, ubah method menjadi **POST**



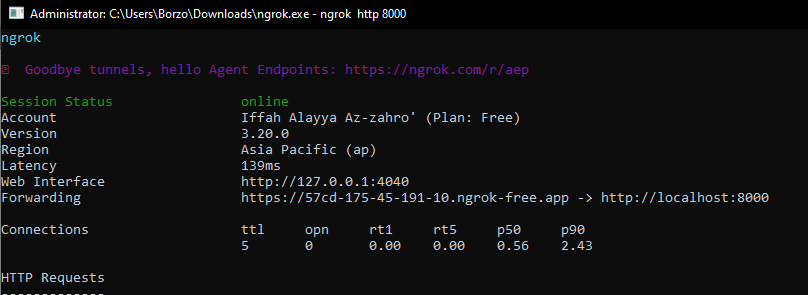
1. Kemudian Cek data yang masuk di phpMyAdmin :



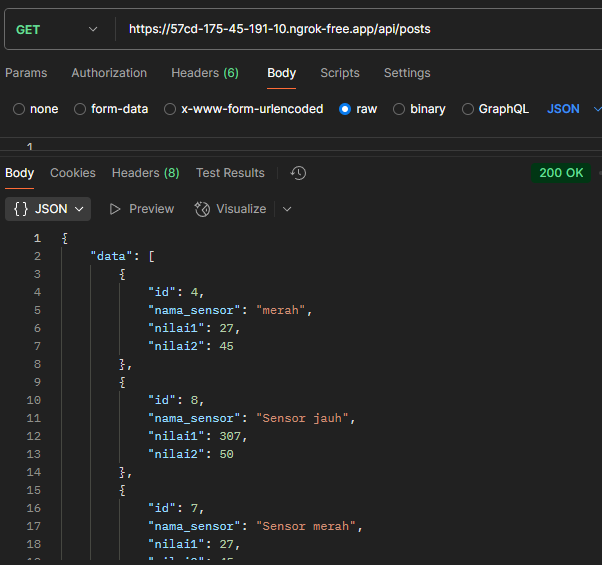
1. Mengonline-kan API menggunakan service ngrok sehingga API dapat diakses melalui device iot atau simulasi wokwi iot
2. Login ke web ngrok, kemudian download aplikasi ngrok



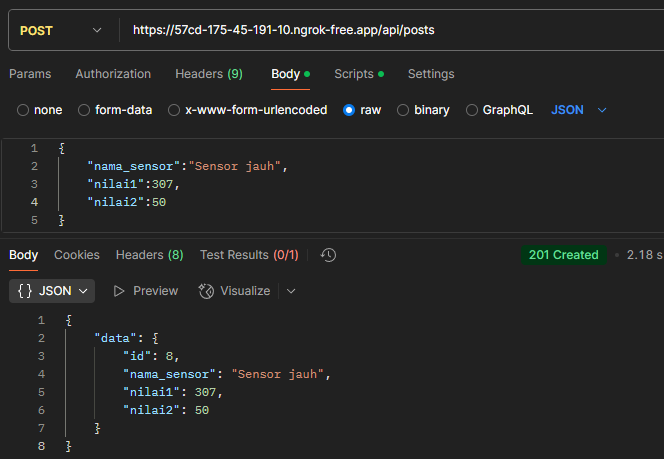
1. Setelah melakukan ekstraksi file, buka prompt lalu jalankan configuration file sesuai yang ada di akun ngrok. Setelah itu jalankan perintah **ngrok http** [**http://localhost:8000**](http://localhost:8080) maka akan muncul tampilan seperti dibawah ini :



1. Kemudian lakukan percobaan **GET API** menggunakan postman menggunakan URL yang diberikan oleh ngrok dengan menambahkan alamat endpoin seperti dibawah ini:



1. Berikutnya lakukan percobaan insert data baru melalui API dengan **method POST**, lalu menambahkan parameter header dan juga body

****

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

Hasil dari praktikum ini adalah berhasilnya pembuatan API yang nantinya akan digunakan untuk project IoT. Dalam praktikum ini, saya menggunakan Laravel 11 sebagai framework utama untuk membangun API, serta Postman sebagai tools untuk menguji endpoint API. Selain itu, Ngrok digunakan untuk membuat API dapat diakses melalui URL publik.

Sampai tahap ini, API yang dibangun menggunakan Laravel 11 sudah dapat berjalan dengan baik, menerima dan merespons permintaan dari simulator WOKWI ESP32 dengan lancar. Dengan adanya URLpublik dari Ngrok, API dapat diakses secara eksternal, memungkinkan pengujian lebih lanjut dan integrasi dengan sistem IoT yang lebih kompleks.

Selama pengujian menggunakan Postman, API dapat memberikan respons sesuai dengan permintaan yang dikirim, baik dalam bentuk GET dan POST. Hal ini menunjukkan bahwa API sudah berfungsi sebagaimana mestinya dalam menangani komunikasi data dari perangkat IoT.

Dengan keberhasilan ini, langkah berikutnya adalah mengoptimalkan API serta mengintegrasikannya dengan perangkat keras ESP32 maupun simulator WOKWI untuk menguji konektivitasnya.

1. **Appendix (Lampiran)**
2. **File Transaksi Sensor**
3. <?php
4. namespace App\Models;
5. use Illuminate\Database\Eloquent\Factories\HasFactory;
6. use Illuminate\Database\Eloquent\Model;
7. class TransaksiSensor extends Model
8. {
9. use HasFactory;
11. /\*\*
12. \* The table associated with the model.
13. \*
14. \* @var string
15. \*/
16. protected $table = 'transaksi\_sensor';
17. /\*\*
18. \* The attributes that are mass assignable.
19. \*
20. \* @var array
21. \*/
22. protected $fillable = [
23. 'nama\_sensor',
24. 'nilai1',
25. 'nilai2',
26. ];
27. /\*\*
28. \* The attributes that should be hidden for arrays.
29. \*
30. \* @var array
31. \*/
32. protected $hidden = [];
33. /\*\*
34. \* The attributes that should be cast.
35. \*
36. \* @var array
37. \*/
38. protected $casts = [];
39. }