LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

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

****

*Ramdan Hidayat*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*hramdan247@gmail.com*](mailto:hramdan247@gmail.com)

**Abstrak**

Perkembangan teknologi web saat ini menuntut integrasi data yang cepat, efisien, dan aman antar aplikasi. Representational State Transfer (REST) API menjadi salah satu solusi utama dalam komunikasi antar sistem. Laravel, sebagai salah satu framework PHP yang populer, menyediakan kemudahan dalam membangun REST API secara cepat dan terstruktur. Dalam praktik ini, digunakan Laravel versi 11 yang dilengkapi dengan fitur modern untuk pengembangan API. Selain itu, Ngrok dimanfaatkan sebagai alat tunneling untuk mengakses server lokal melalui jaringan internet. Praktik ini bertujuan untuk memberikan pemahaman praktis dalam membangun dan menguji REST API menggunakan Laravel 11 serta mengeksposnya secara publik menggunakan Ngrok.

*Keyword : framework, laravel, ngrok*

1. **Pendahuluan** 
   1. **Latar Belakang**

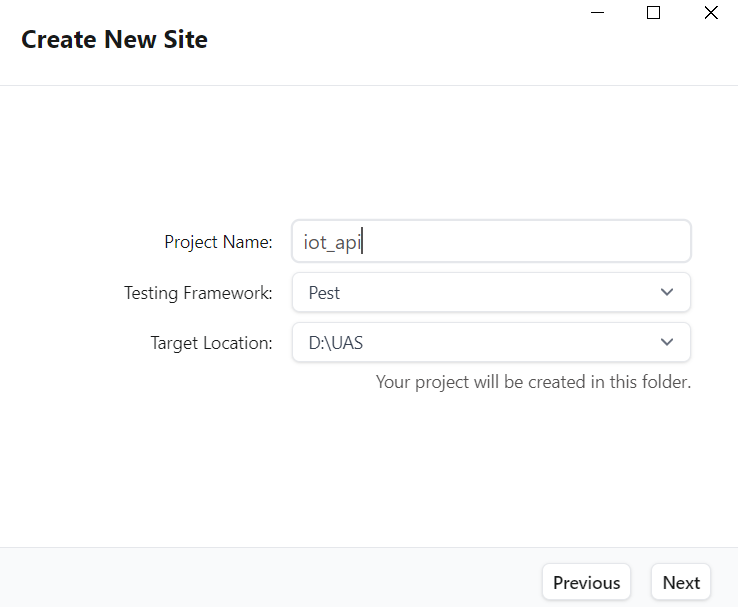
Dalam dunia pengembangan aplikasi modern, kebutuhan akan komunikasi data antar sistem semakin tinggi. Salah satu metode yang paling umum digunakan untuk memenuhi kebutuhan tersebut adalah Application Programming Interface (API). API memungkinkan sistem yang berbeda untuk saling bertukar data dengan standar yang telah ditentukan. Laravel sebagai framework PHP modern menyediakan fitur lengkap untuk membangun API, seperti routing, controller, middleware, hingga fitur autentikasi.

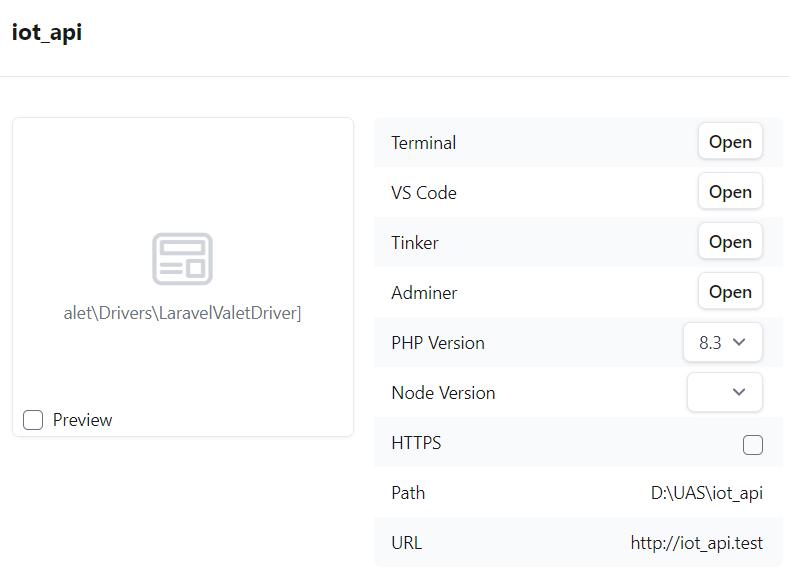
Namun, dalam pengujian API secara langsung melalui internet, sering kali dibutuhkan server online. Untuk mengatasi keterbatasan tersebut, Ngrok hadir sebagai solusi untuk mengekspos server lokal ke publik secara sementara dan aman. Dengan Ngrok, developer dapat menguji API lokal secara langsung dari perangkat lain atau tim penguji tanpa perlu deploy ke server online.

Melalui praktik ini, mahasiswa diharapkan memahami bagaimana membangun API menggunakan Laravel 11 serta bagaimana memanfaatkan Ngrok untuk pengujian API secara efisien.

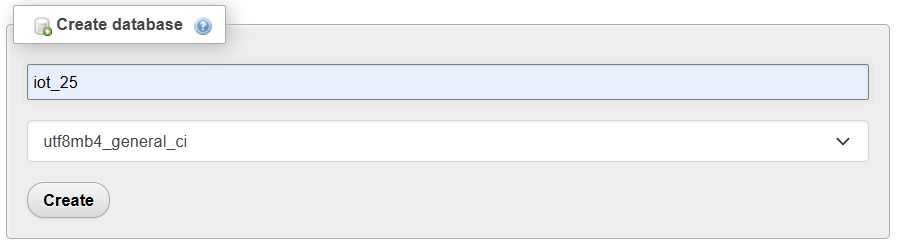
* 1. **Tujuan Eksperimen**

1. Mempelajari dasar-dasar pembuatan REST API menggunakan Laravel 11..
2. Mengimplementasikan routing, controller, dan model untuk keperluan API.
3. Menguji API secara lokal menggunakan Postman atau aplikasi sejenis..
4. Mengekspos API lokal ke internet menggunakan Ngrok.
5. Mengetahui alur kerja pengujian API dalam lingkungan pengembangan.
6. **Metodologi**
   1. **Alat dan Bahan**
7. Laptop
8. Ngrok
9. Laravel
10. Internet
11. Postman
    1. **Langkah Implementasi**
12. Buat project (iot\_api) menggunakan laravel 11 dengan HERD dan tunggu proses penginstalan hingga selesai

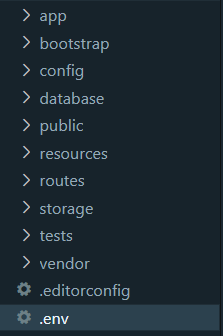
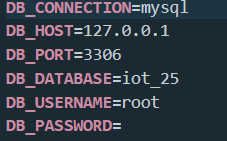




1. Buat database di phpmyadmin dengan nama iot\_25



1. Lalu buka teks editor dan ubah dari isi konfigurasi file.env

DDB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

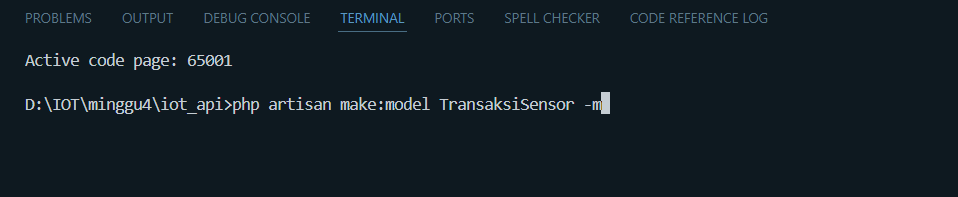
DB\_DATABASE=iot\_25

DB\_USERNAME=root

DB\_PASSWORD=

Note(sesuaikan db\_database dengan di phpmyadmin yang telah dibuat )

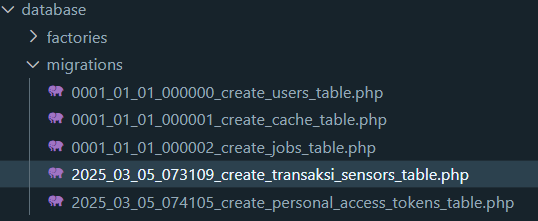
1. Buat file Transaksi sensor.php dengan cara menjalankan perintah di terminal



Maka perintah yang dijalankan akan berada di folder sbg berikut



1. Lalu ubah file dari 2025\_03\_05\_073109\_create\_transaksi\_sensors\_table.php didalam folder database/migrations



dengan kodenya

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

/\*\*

\* Run the migrations.

\*/

public function up(): void

{

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

$table->id('id')->startingValue(1); // Menetapkan AUTO\_INCREMENT dimulai dari 1

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

$table->integer('nilai1', false)->length(255); // int(255)

$table->integer('nilai2', false)->length(255); // int(255)

$table->timestamps(); // Menambahkan created\_at dan updated\_at

});

}

/\*\*

\* Reverse the migrations.

\*/

public function down(): void

{

Schema::dropIfExists('transaksi\_sensors');

}

};

1. Kemudian ubah isi file **app/Models/TransaksiSensor.php**



Dengan kode sebagai berikut

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class TransaksiSensor extends Model

{

use HasFactory;

/\*\*

\* The table associated with the model.

\*

\* @var string

\*/

protected $table = 'transaksi\_sensor';

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array

\*/

protected $fillable = [

'nama\_sensor',

'nilai1',

'nilai2',

];

/\*\*

\* The attributes that should be hidden for arrays.

\*

\* @var array

\*/

protected $hidden = [];

/\*\*

\* The attributes that should be cast.

\*

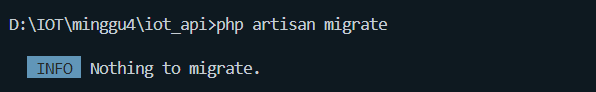
\* @var array

\*/

protected $casts = [];

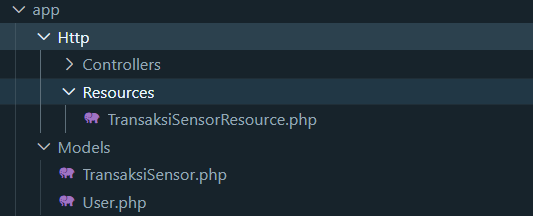
}

1. Kemudian jalankan perintah di terminal **php artisan migrate :**



Karena disini sudah di migrasi maka output yang muncul adalah seperti diatas

1. Lalu buat resource dengan menjalankan perintah di terminal **php artisan make:resource TransaksiSensorResource**



1. Lalu ubah isi file **TransaksiSensorResource.php** berada di dalam folder app/Http/Resources dengan kode berikut

<?php

namespace App\Http\Resources;

use Illuminate\Http\Request;

use Illuminate\Http\Resources\Json\JsonResource;

class TransaksiSensorResource extends JsonResource

{

/\*\*

\* Transform the resource into an array.

\*

\* @param \Illuminate\Http\Request $request

\* @return array

\*/

public function toArray($request)

{

return [

'id' => $this->id,

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

'nilai1' => $this->nilai1,

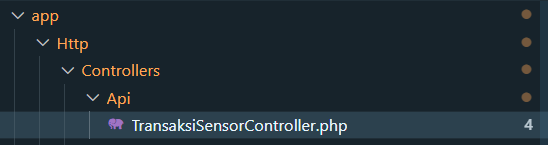
'nilai2' => $this->nilai2,

];

}

}

1. Buat API controler dengan perintah di terminal **php artisan make:controller Api/TransaksiSensorController**



Lalu ubah isi foldernya dengan kode

<?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);

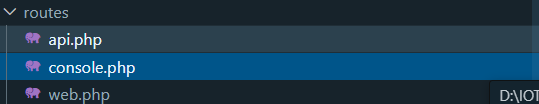
$transaksiSensor->delete();

return response()->json(['message' => 'Deleted successfully'], 204);

}

}

Lalu buat Route khusus API dengan perintah di terminal **php artisan install:api**



Lalu ubah isi filenya dengan kode

<?php

use Illuminate\Auth\Middleware\Authenticate;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Route;

Route::get('/user', function (Request $request) {

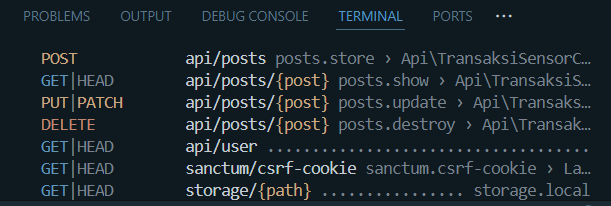
return $request->user();

})->middleware(Authenticate::using('sanctum'));

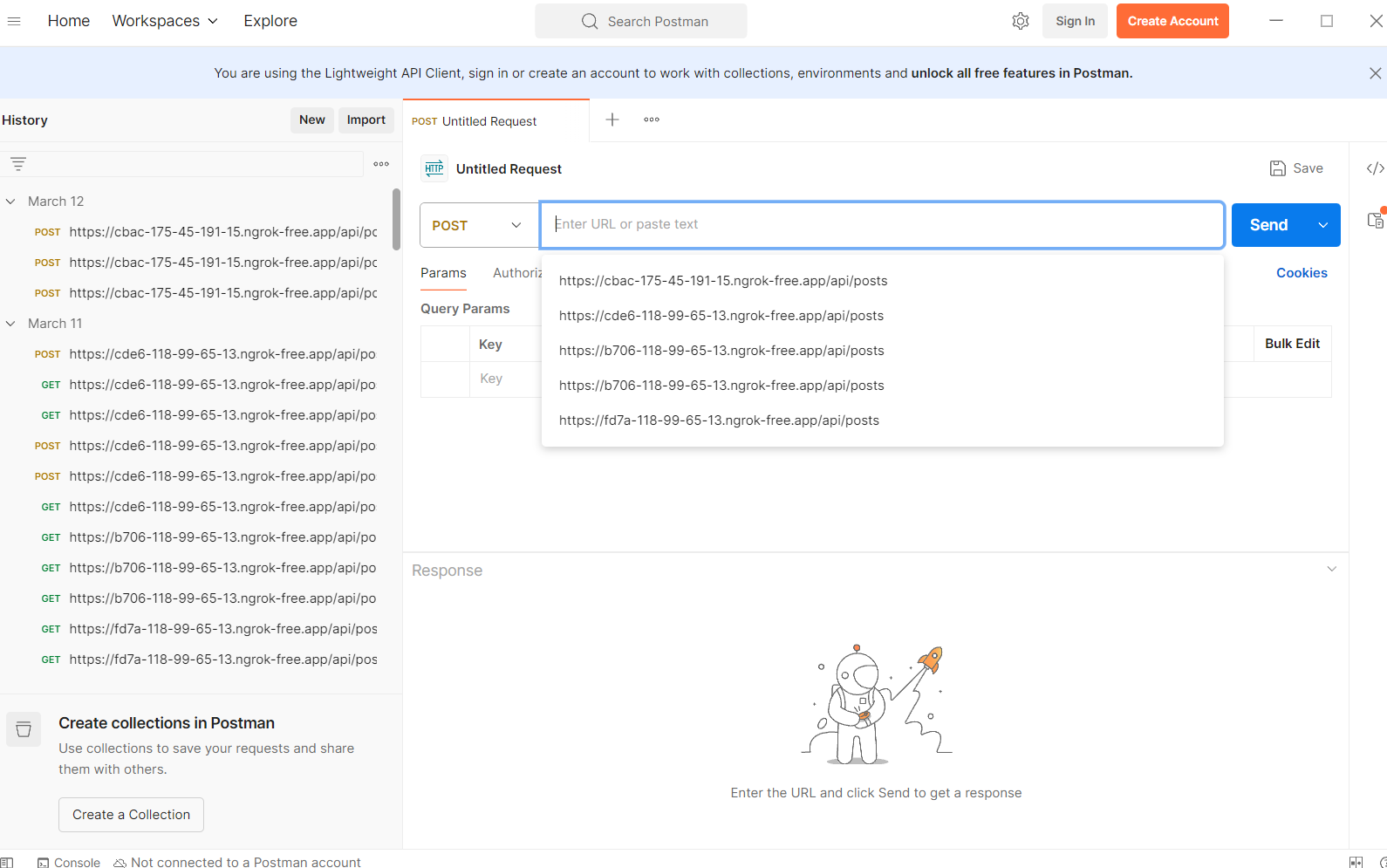
//posts

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

Pastikan routes telah terbentuk dengan memerintahkan terminal dengan **phpartisanroute:list**

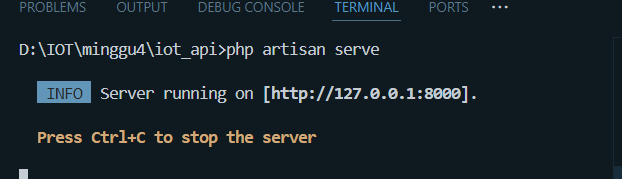


1. Dengan itu maka kita bisa menjalankan testing dengan tools POSTMAN yang bisa di download pada link <https://www.postman.com/downloads/>



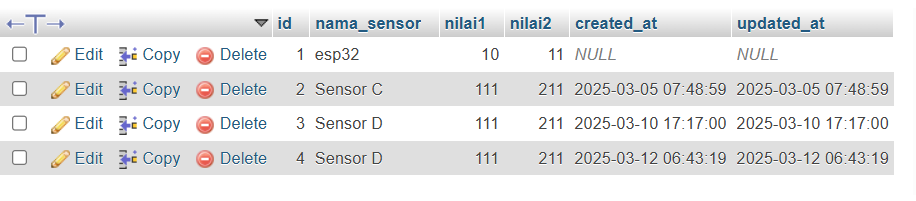
Contoh tampilan POSTMAN yang sudah di download

Sebelum melakukan percobaan jalankan laravel dengan perintah di terminal **php artisan serve**

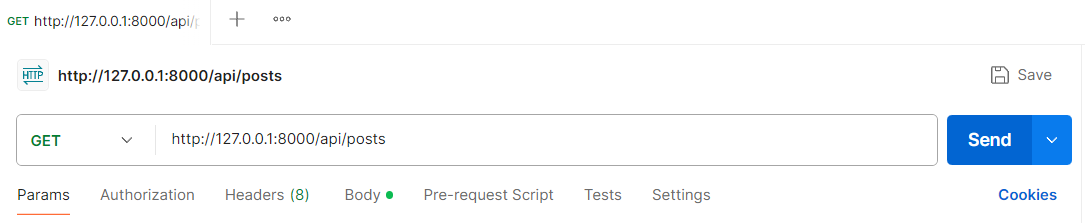


Pastikan servernya berjalan

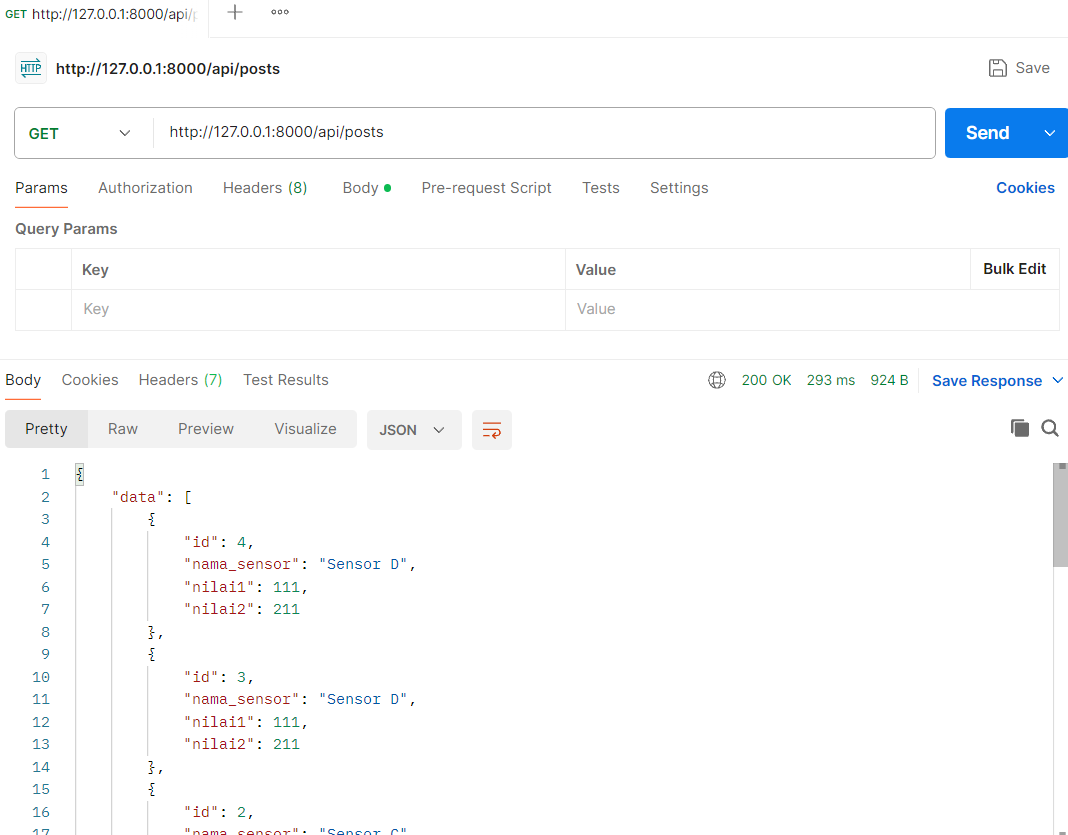
Untuk mengetahui apakah datanya bisa dilihat maka isi database di phpmyadmin



Lalu buka POSTMAN dan masukan alamat dari laravel [**http://127.0.0.1:8000/api/posts**](http://127.0.0.1:8000/api/posts)lalu pilih method GET**,** lalu klik tombol SEND

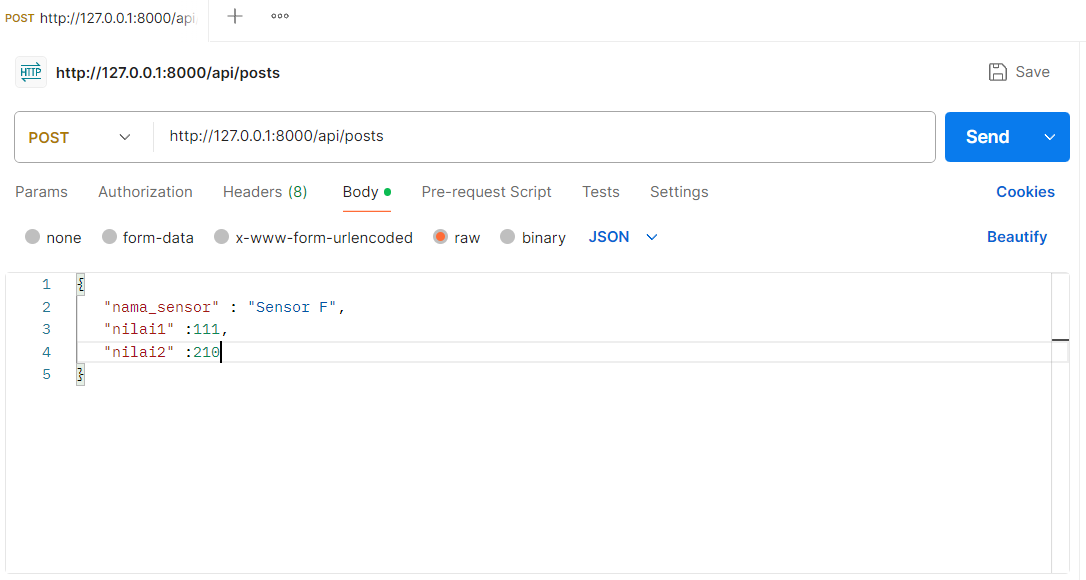


Pastikan data telah berhasil di tampilkan

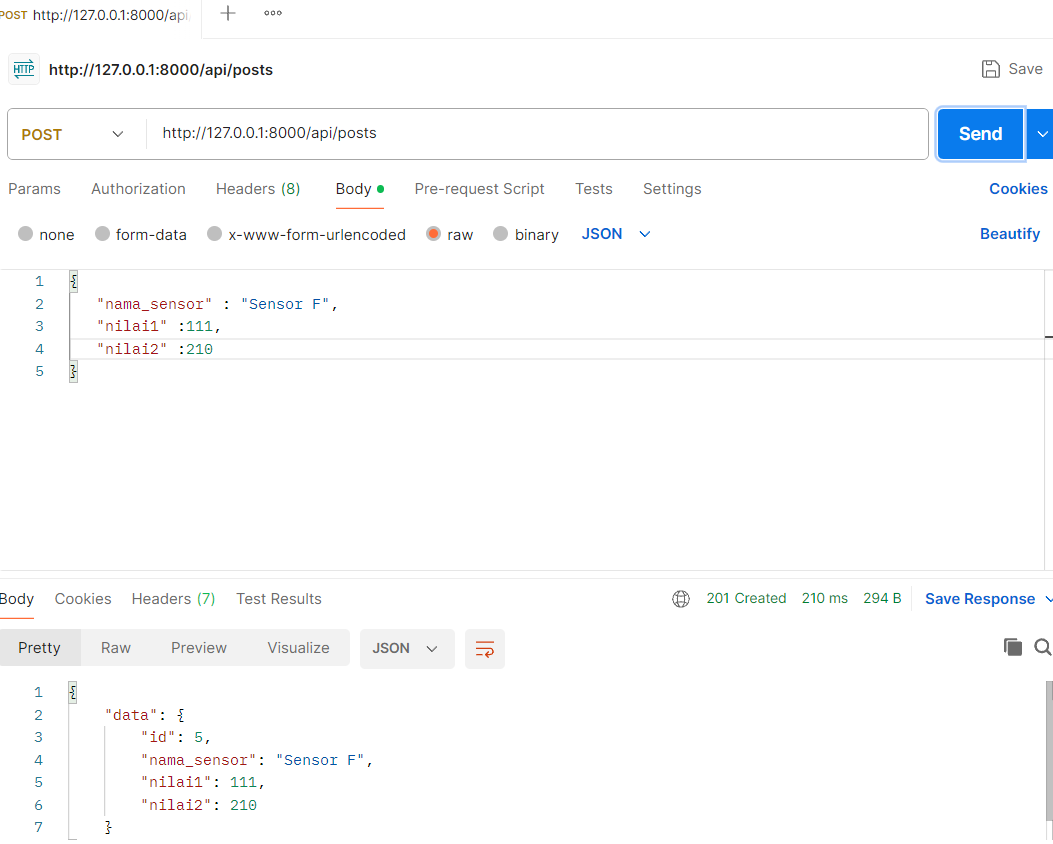


API tleah berfungsi untuk mengambil data dari database,

1. Lalu ubah method GET menjadi POST dan pada body isikan data yang sesuai dengan database



Pastikan data berhasil dikirim ke database



Lalu check apakah data berhasil masuk di database

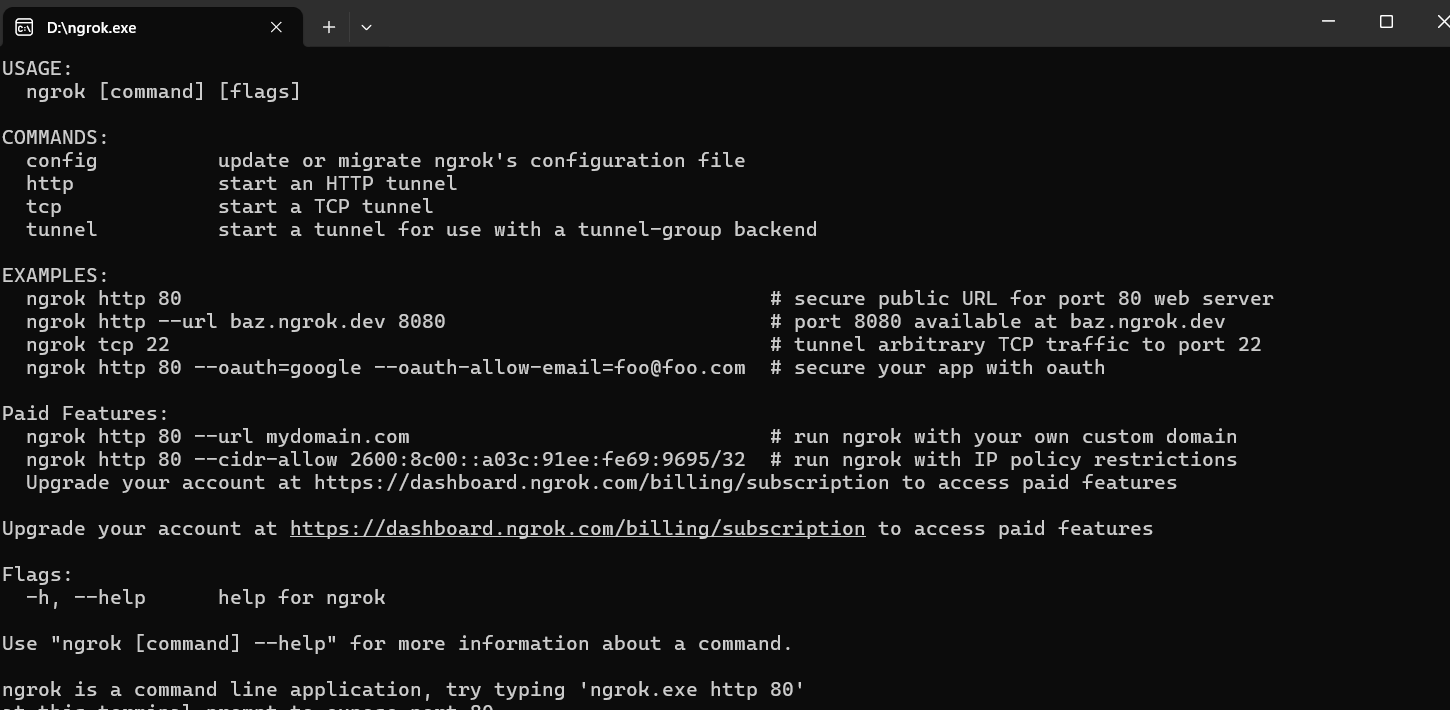


Data berhasil masuk

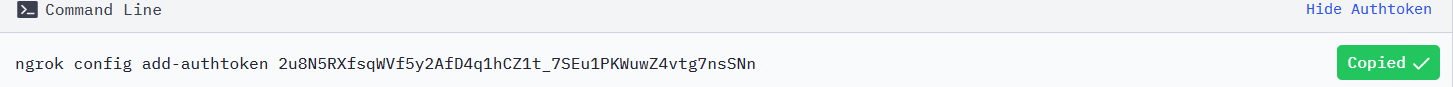
1. Untuk API bida diakses online maka menggunakan service NGROK yang dimana device dari IOT bisa mnegakses Internet

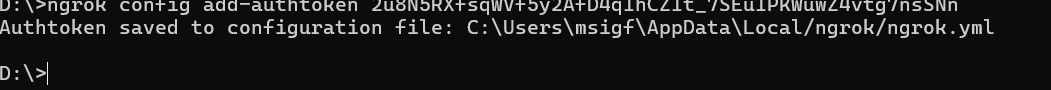
Download dan instal NGROK pada link [**https://dashboard.ngrok.com/signup**](https://dashboard.ngrok.com/signup)ketika sudah di dwonload ekstak file ngrok

Lalu buka CommandPrompt dari alamat ngrok

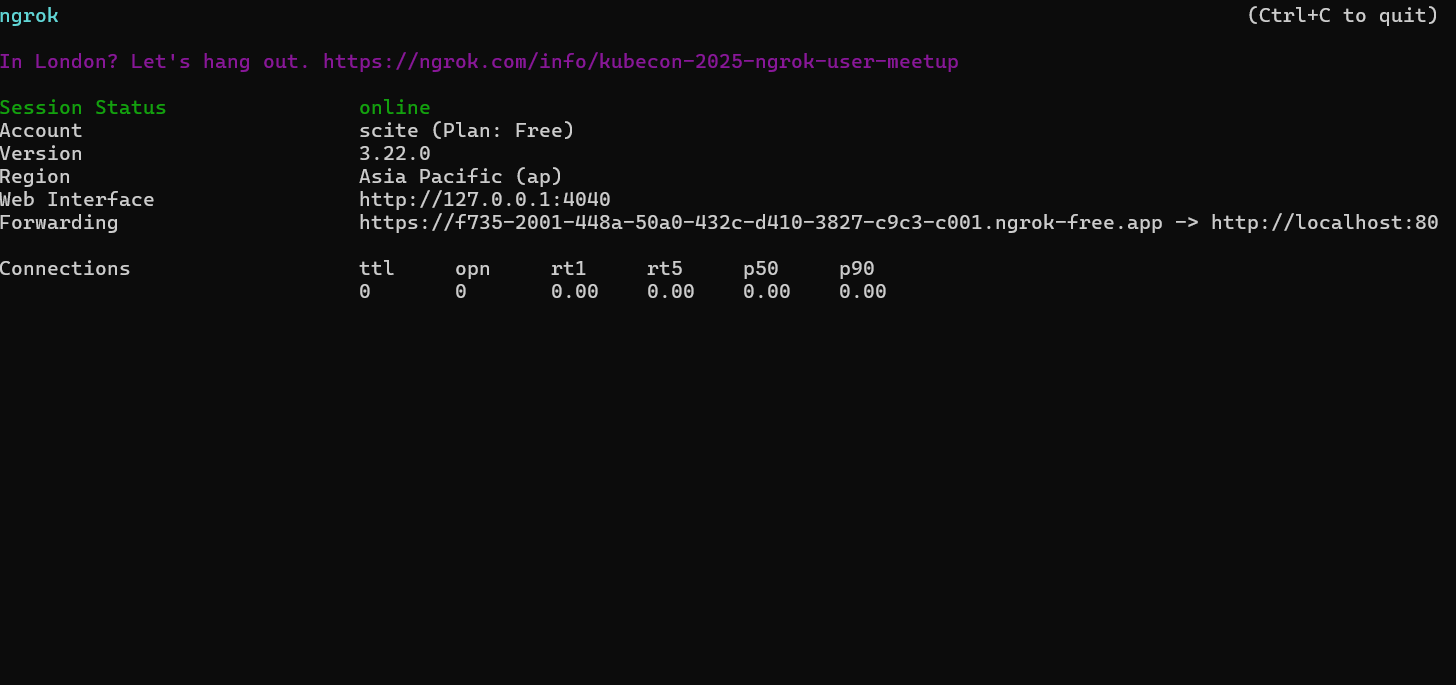


Kemudian jalankan perintah sesuai dengan ngrok





Kemudian jalankan perintah **ngrok http** [**http://localhost:8000**](http://localhost:8080)

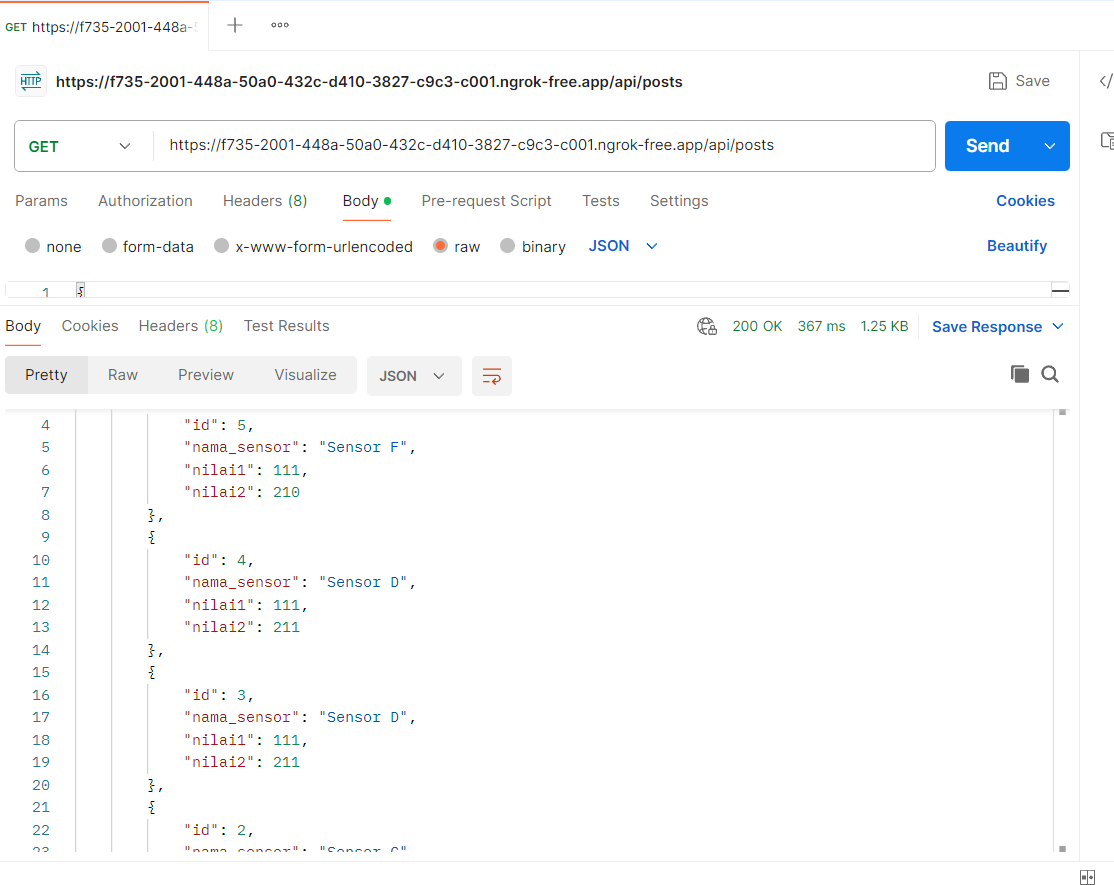


Kemudian lakukan percobaan menggunakan postman menggunakan URL yang diberikan oleh ngrok. Pada contoh ini, ngrok memberikan URL publik yang dapat diakses melalui internet pada alamat , sesuaikan dengan milik Anda .

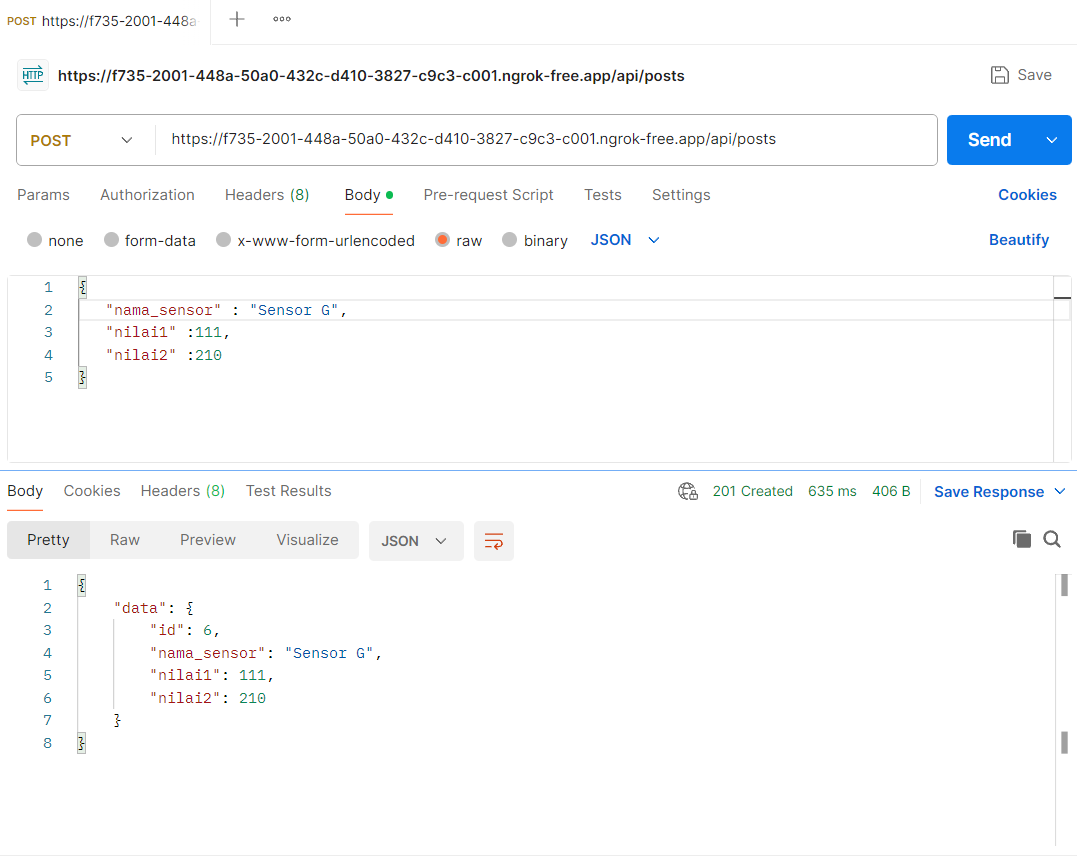
<https://f735-2001-448a-50a0-432c-d410-3827-c9c3-c001.ngrok-free.app>

untuk melakukan GET API, url tambahkan degan/api/posts

<https://f735-2001-448a-50a0-432c-d410-3827-c9c3-c001.ngrok-free.app/api/posts>



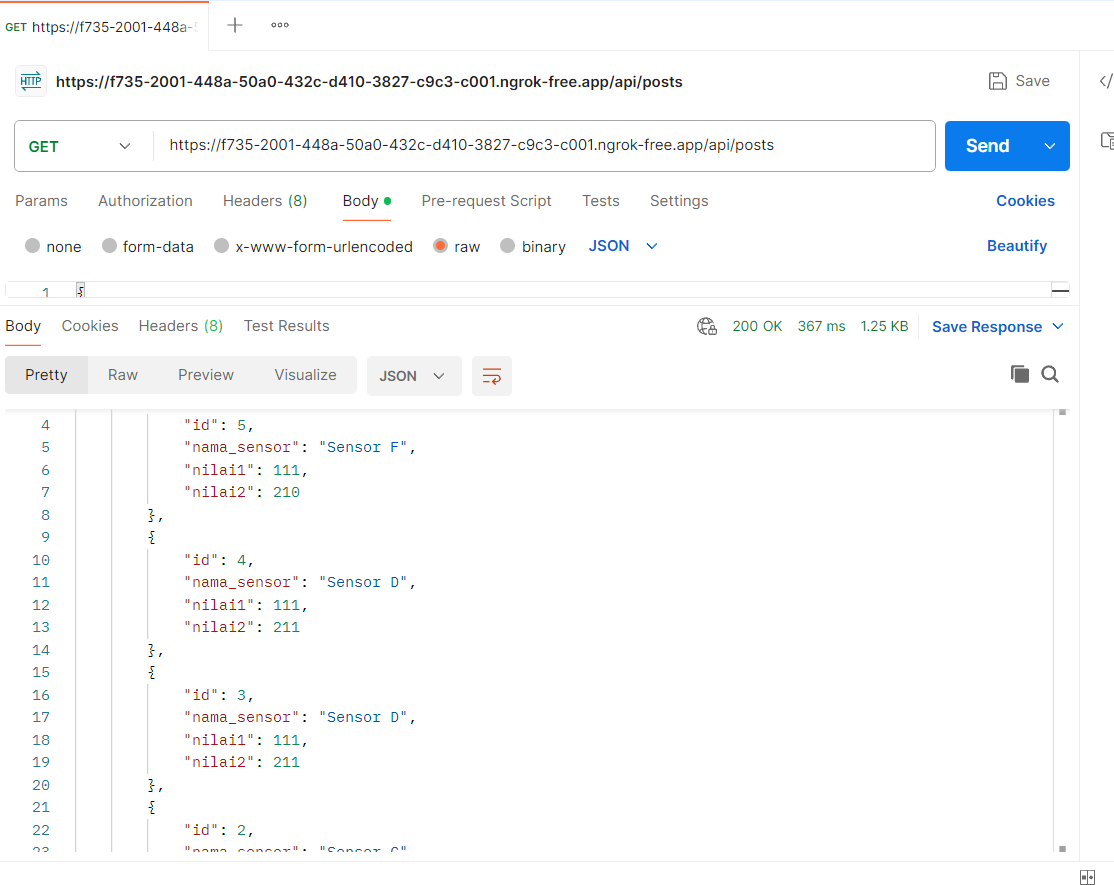
Ubah method menjadi POST dan sesuaikan dengan database

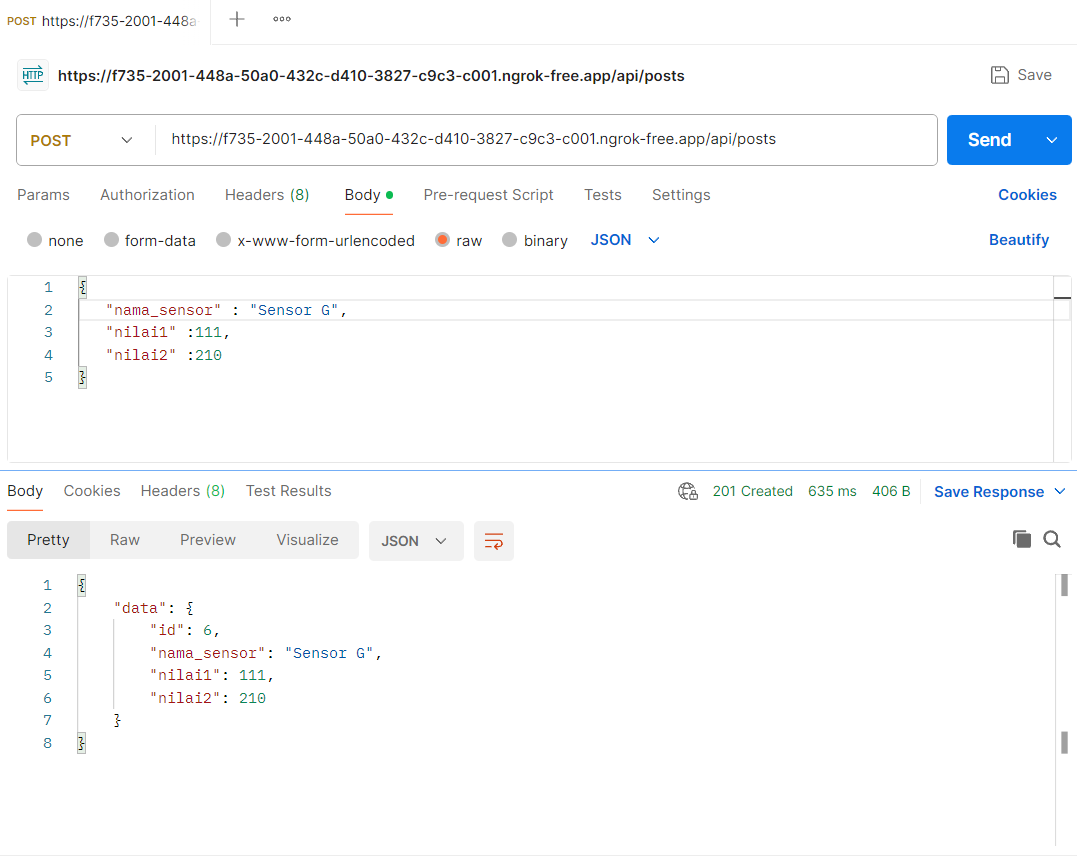


Dengan ini API sudah dibangun menggunakan laravel dan dapat digunakan dan diakses melalui URL publik

1. **Hasil dan Pembahasan** 
   1. **Hasil Eksperimen**

Hasil dari praktikum ini adalah dengan menjalankan laravel dengan mengonlinekan dengan ngrok maka IOT dapat mengakses intenernet dan dapat diakses melalui URL publik dengan contohnya adalah





.