LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

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

*Assyfaul Khoiriyah*

Fakultas Vokasi, Universitas Brawijaya

[assyfaassyfa0@gmail.com](mailto:assyfaassyfa0@gmail.com)

**Abstract (Abstrak)**

Internet of Things (IoT) memungkinkan perangkat lunak untuk berkomunikasi dan bertukar data melalui jaringan, memberikan efisiensi dalam berbagai bidang, termasuk pemantauan lingkungan. Dalam eksperimen ini, ESP32 digunakan sebagai mikrokontroler yang berfungsi untuk membaca data dari sensor DHT22 yang mengukur suhu dan kelembapan. Data yang diperoleh dikirim ke server berbasis laravel menggunakan metode HTTP POST. Untuk mengatasi keterbatasan akses server lokal, digunakan Ngrok sebagai perantara agar API dapat diakses secara publik. Hasil ekperimen ini menunjukkan bahwa sistem yang dibangun dapat berjalan dengan baik, dengan data sensor berhasil dikirim dan disimpan di database sevara real-time.

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

Perkembangan teknologi IoT semakin pesat, memungkinakan perangkat untuk saling terhubung dan berbai informasi dalam berbagai aplikasi, seperti pemantauan lingkungan, otomasi rumah, dan sistem industri. Salah satu komponen utama dalam sistem IoT adalah sensor yang mampu mengumpulkan data dari lungkungan sekitar. ESP32 merupakan salah satau mikrokontroler yang banyak digunakan dalam proyek IoT karena memiliki dukungan konektivitas WiFi yang memungkinkan pengiriman data secara real-time.

Dalam ekperimen ini, sensor DHT22 digunakan untuk mengukur suhu dan kelembapan udara. Data yang diperoleh kemudian dikirmkan ke server berbasis laravel menggunakan protokol HTTP POST. Namun, salah sati tantangan dalam pengembangan API berbasis laravel adalah keterbatasan akses server lokal yang tidal dapat diakses dari jaringan ekternal. Untuk mengatasi hal ini, digunakan Ngrok sebagai layanan tunneling yang memungkinkan API laravel daat diakses melalui URL publik. Penelitian ini bertujuan memahami proses pengiriman data dari perangkat IoT ke server menggunakan laravel, serta menguji fungsionalitas ngrok dalam mempublikasikan API secara online.

**1.2 Tujuan Eksperimen**

Tujuan dari eksperimen ini adalah:

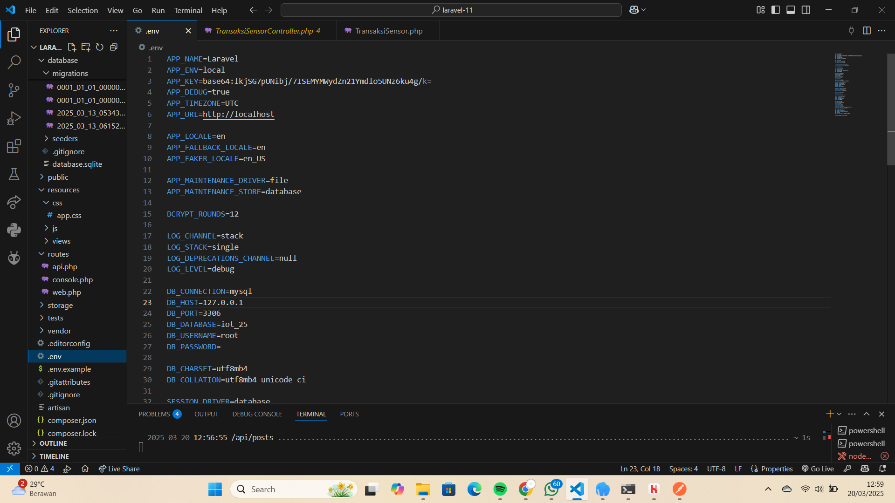
1. Mengintegrasikan ESP32 dengan sensor DHT22 untuk membaca suhu dan kelembaban.
2. Mengirim data sensor ke server laravel menggunakan HTTP POST.
3. Menggunakan Ngrok untuk menghubungkan API lokal ke internet.
4. Menganalisis eror yang terjadi, seperti HTTP 400, dan mencari solusinya.
5. **Methodology (Metodologi)**

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

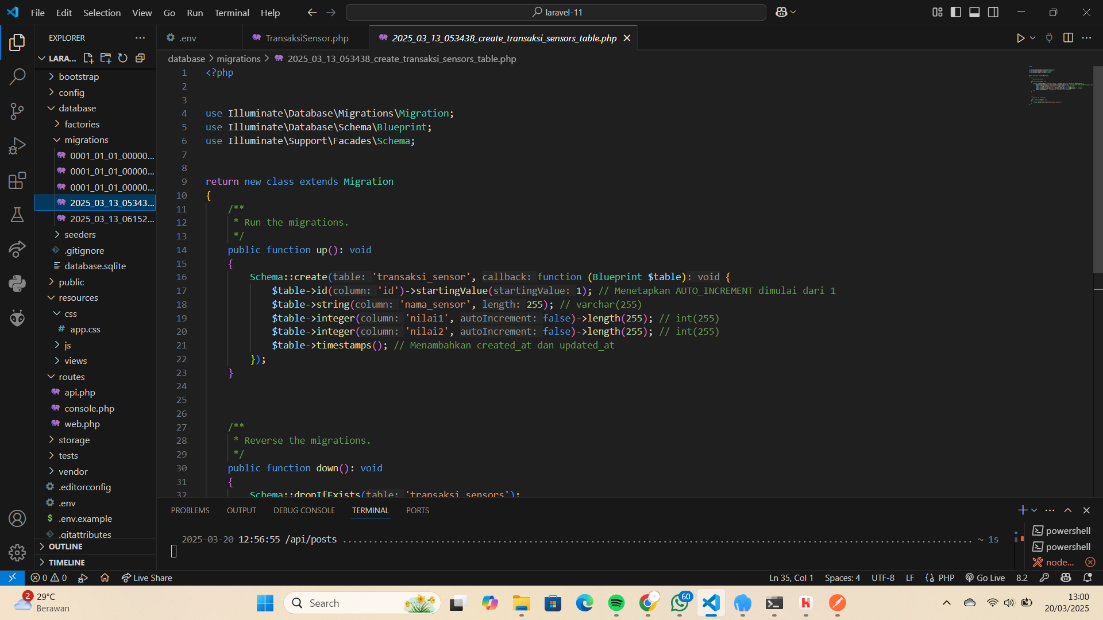
* ESP32
* Sensor DHT22
* **Arduino IDE** (untuk pemrograman ESP32)
* **Laravel** (sebagai backend API)
* **Ngrok** (untuk membuat API dapat diakses dari internet)
* **Postman** (untuk menguji API secara manual)
* **MySQL** (untuk menyimpan data sensor)

**2.2 Implementation Steps (Langkah Implementasi)**

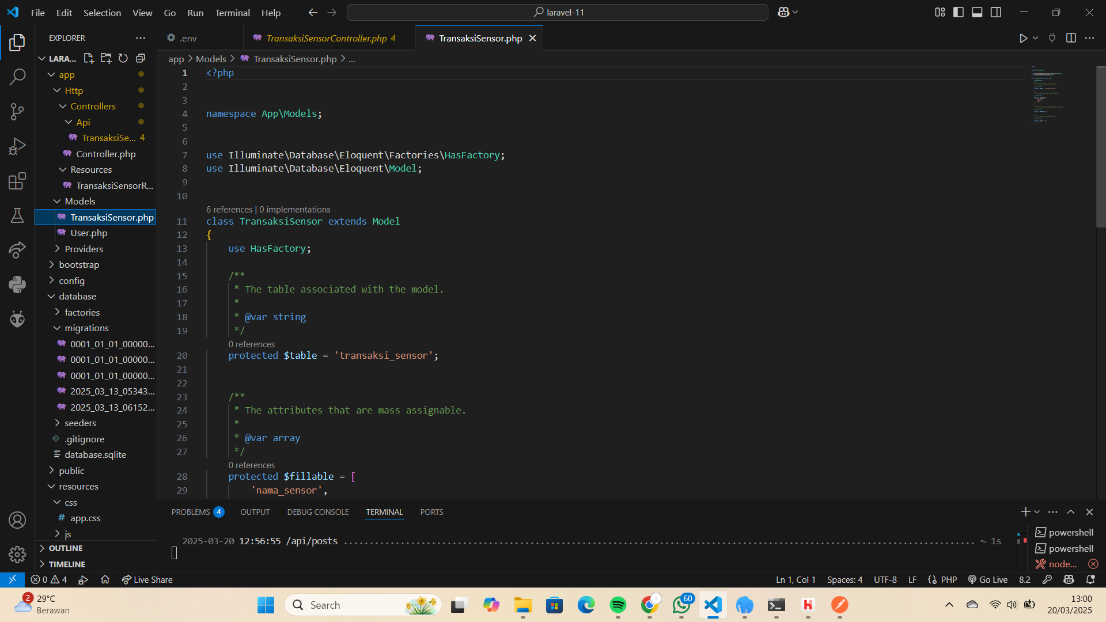
1. Buat database di phpmyadmin dengan nama **iot\_25**
2. Ubah isi konfigurasi file .env

****

1. Buat file TransaksiSensor.php
2. Kemudian ubah file 2025\_02\_21\_074123\_create\_transaksi\_sensors\_table.php



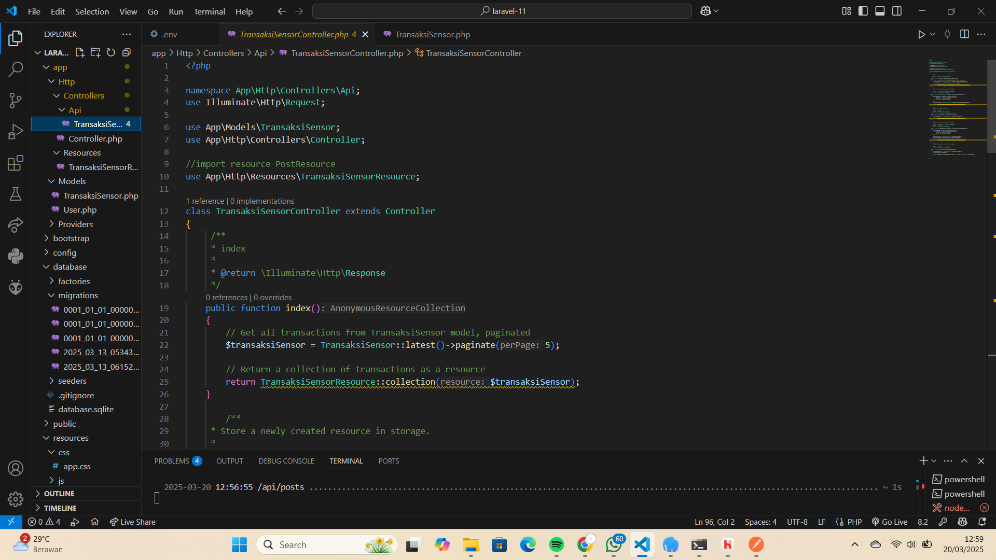
1. Kemudian ubah isi file app/Models/TransaksiSensor.php



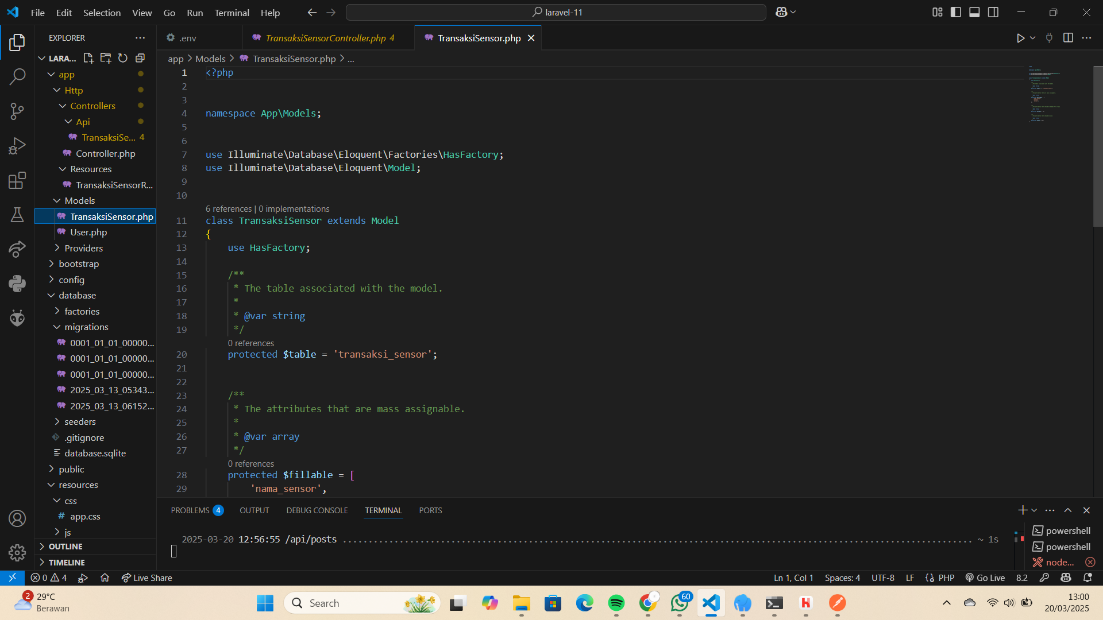
1. Kemudian jalankan perintah dengan:

* Php artisan migrate
* Php artisan make:resource TransaksiSensorResource

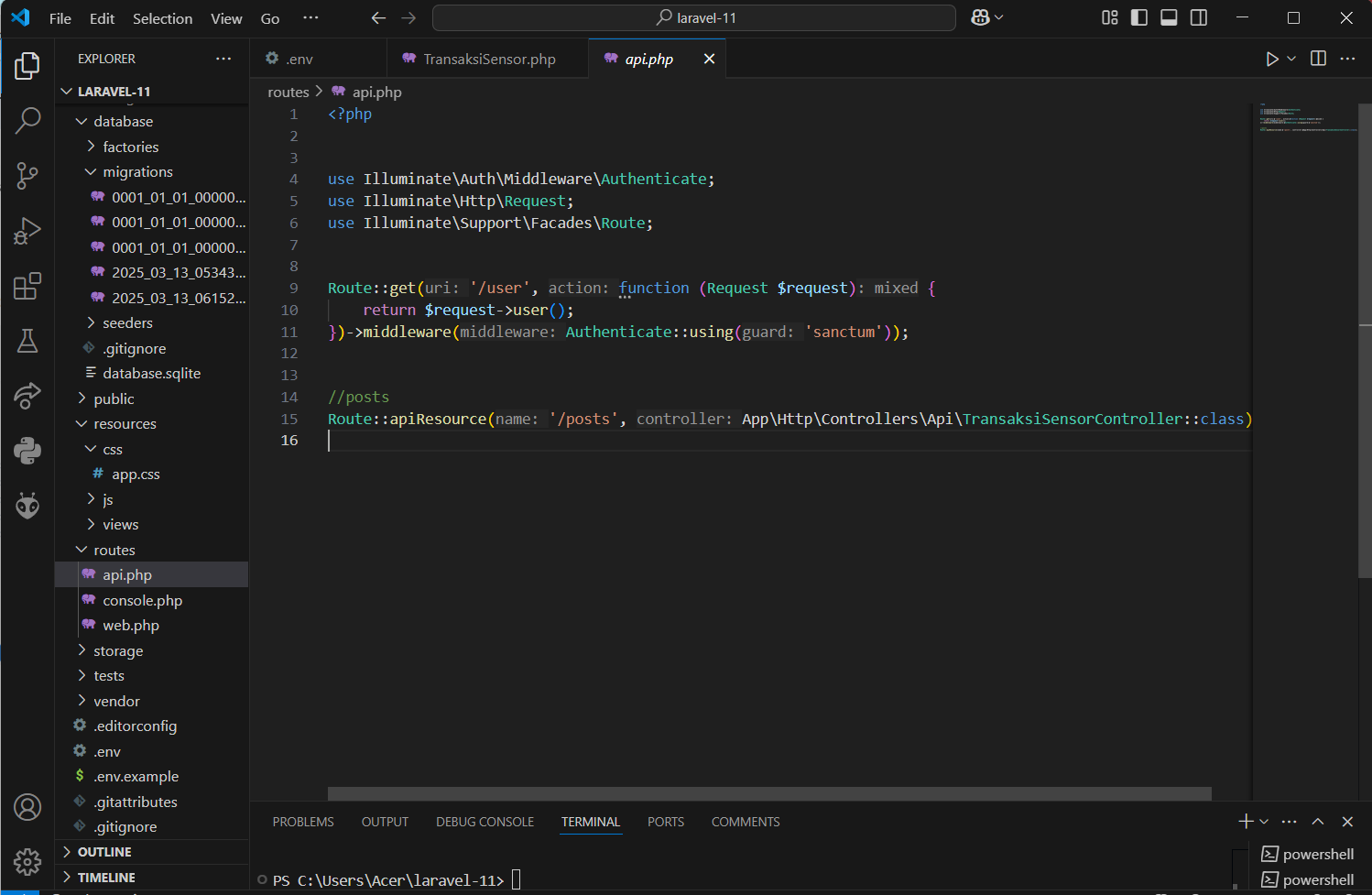
1. Ubah isi file TransaksiSensorResource.php  yang ada di folder app-Http-Resources



1. Ubah isi file app/Http/Controllers/Api/TransaksiSensorController.php



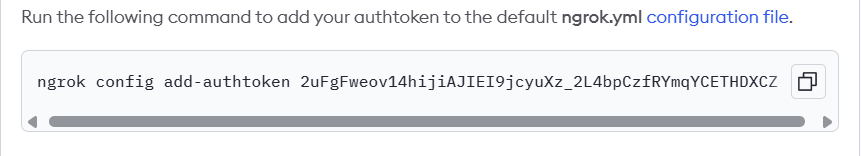
1. Buka file routes/api.php dan ubah isi file menjadi:



1. Kemudian download postman
2. Untuk melakukan percobaan akses API, pastikan aplikasi laravel dijalankan dengan perintah:

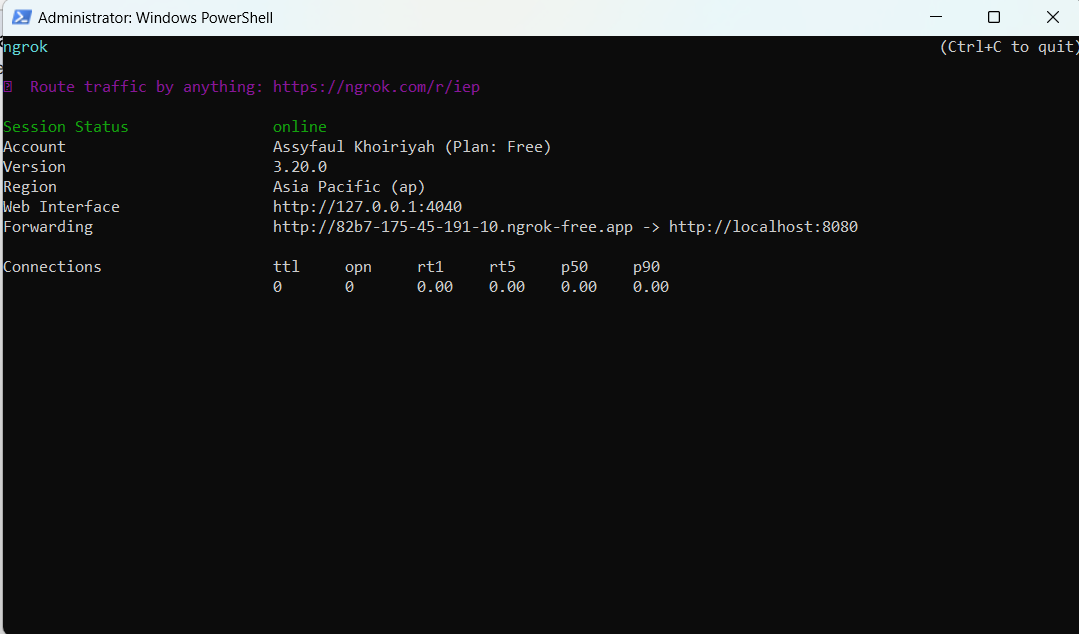
Php artisan serve

1. Pilih method GET untuk mengambil data dari database , kemudian klik tombol SEND
2. Lakukan percobaan insert data ke table di database menggunakan API
3. Buka command prompt dari alamat folder ekstraksi
4. Kemuadian jalankan perintah sesuai yang ada di akun ngrok:



1. Kemudian jalankan perintah berikut untuk mengonlinekan laravel melalui port 8000

Ngrok http <http://localhost:8080>

****

1. Kemudian lakukan percobaan menggunakan postman menggunakan URL yang diberikan oleh ngrok : <http://82b7-175-45-191-10.ngrok-free.app>
2. Sampai disini API yang dibangun menggunakan laravel sudah dapat berjalan dengan baik dan dapat diakses melalui URL publik.
3. Pada bab berikutnya,  akses API akan dilakukan melalui simulator WOKWI ESP32 yang telah dibuat pada bab sebelumnya.
4. **Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

Eksperimen ini membuktikan bahwa ESP32 dan sensor DHT22 dapat diintegrasikan dengan baik menggunakan Laravel sebagai backend API. Dengan bantuan Ngrok, API dapat diakses secara publik, memungkinkan data sensor dikirim dan diakses dari jaringan eksternal. Pengujian menggunakan Postman menunjukkan bahwa API dapat menangani permintaan GET dan POST dengan baik. Error yang muncul dapat diatasi dengan memperbaiki format request dan konfigurasi API. Keberhasilan eksperimen ini membuka peluang untuk pengembangan lebih lanjut dalam sistem pemantauan berbasis IoT.

1. **Appendix (Lampiran, jika diperlukan)**

<?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');

    }

};