**Praktik Simulasi Sensor Jarak  
(Ultrasonic)**

*Pelangi Anggel*

Fakultas Vokasi , Universitas Brawijaya

Email: pelangianggel283@gmail.com

**Abstract**

*The integration of IoT devices with cloud-based systems is crucial for real-time data communication, enabling seamless interaction between microcontrollers and servers. This practice focuses on ESP32 microcontroller accessing a REST API through the Wokwi simulation platform, eliminating the need for physical hardware. By simulating API communication, ESP32 sends data in JSON format to a Laravel-based API, which processes and stores the information in a MySQL database. Additionally, Ngrok is used to expose the local API to the internet, while Postman facilitates API testing. By understanding client-server interactions in IoT, this practice provides fundamental knowledge essential for developing cloud-integrated IoT applications*.

*Keywords—* *ESP32, IoT, REST API, Wokwi*

**1. Introduction**

**1.1 Background**

The Internet of Things (IoT) has revolutionized the way devices interact with cloud-based systems, enabling seamless communication between microcontrollers and servers. One of the widely used technologies for this purpose is REST API, which allows microcontrollers like ESP32 to exchange data with web-based applications. However, testing API integration often requires physical hardware, which can be impractical for early-stage development. In this practice, a Wokwi simulation is used to emulate ESP32's interaction with a Laravel API, allowing data transmission via WiFi without actual hardware. By understanding API communication and cloud integration, this practice provides essential knowledge for developing real-time IoT systems.

**1.2 Objectives**

The purpose of this practicum is to understand and implement API access using ESP32 through the Wokwi simulation and integrate it with an API built using Laravel. Participants will learn how to connect the ESP32 to a WiFi network, send data in JSON format to an API endpoint, and ensure that the data is stored in a MySQL database. Additionally, this practicum aims to introduce the use of Ngrok as a bridge to access the API from a local network to the internet and test API responses using Postman. By completing this practicum, participants will gain an understanding of client-server communication in the Internet of Things (IoT) and apply REST API technology for real-time data integration.

**2. Methodology**

**2.1 Tools & Materials**

* Wokwi Simulator (ESP32 simulation platform)
* Laravel Framework (API development)
* Ngrok (Public API tunneling)
* Postman (API testing)
* MySQL Database (Data storage)
* Code Editor (VS Code/Sublime/PHPStorm)
* Materials:
* ESP32 (Virtual in Wokwi)
* WiFi Network
* REST API Endpoint
* JSON Payload
* These tools and materials are essential for API access simulation via Wokwi.
  1. **Implementation Steps**

1. Running the Laravel API
   * Ensure the Laravel API has been created in the previous chapter.
   * Start the Laravel API so it can be accessed over the local network.
2. Creating a Simulation File in Wokwi
   * Open the Wokwi platform.
   * Create a new project and add ESP32 as the microcontroller.
3. Writing the ESP32 Program
   * Create a program file in the Wokwi project to connect ESP32 to WiFi and send data to the API.
4. Running the Simulation
   * Start the Wokwi simulation and monitor the output in the Serial Monitor.
   * Ensure that data is successfully sent to the API by checking the response.
5. Verifying Data in the Database
   * Check if the data sent from Wokwi has been stored in the Laravel API database.
6. Using Ngrok for Public Access
   * Run Ngrok to make the Laravel API accessible from the internet.
   * Update the API URL in the Wokwi program with the URL provided by Ngrok.
7. Testing the API with Postman (Optional)
   * Test the API using Postman with the POST method to verify that the API correctly receives and stores the data.

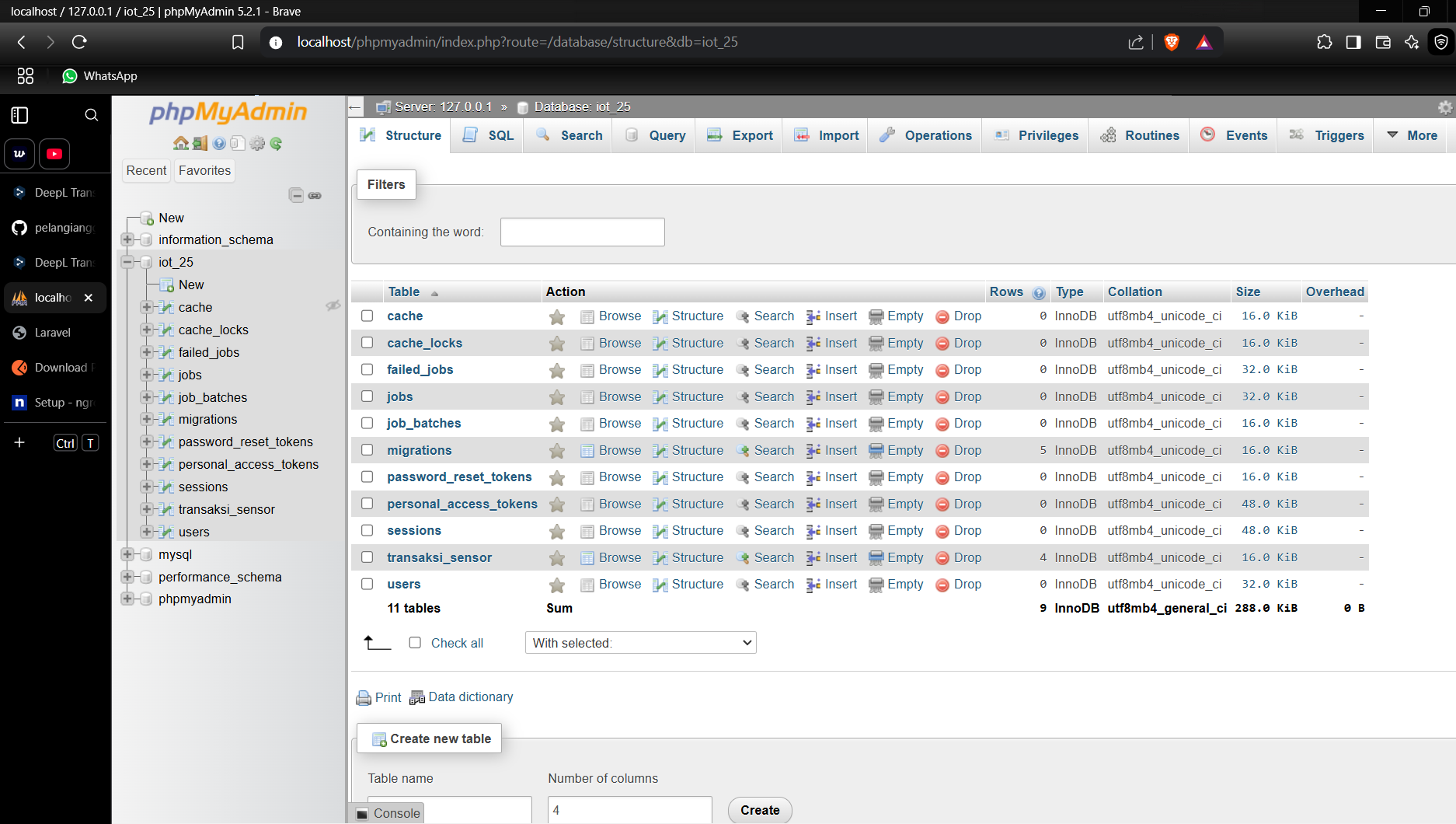
By following these steps, the Wokwi simulation can successfully connect with the Laravel API, allowing data transmission and verification in the database.

**3. Results and Discussion**

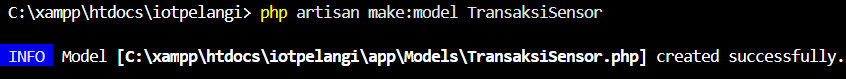
**3.1 Experimental Results**

|  |  |  |
| --- | --- | --- |
| **Platform** | **Completed Task** | **Key Outcome** |
| Wokwi | API access simulation | The ESP32 successfully sent data to the Laravel API and displayed the response in real-time. |
| ESP32 (Simulated) | Sending JSON data to API | The ESP32 correctly transmitted JSON data via WiFi, and the API processed and stored it in the database. |
| Ngrok | API public access setup | The Laravel API was successfully exposed to the internet, allowing external access from the ESP32. |
| Postman | API request testing | API requests were successfully tested, confirming data transmission and server response accuracy. |

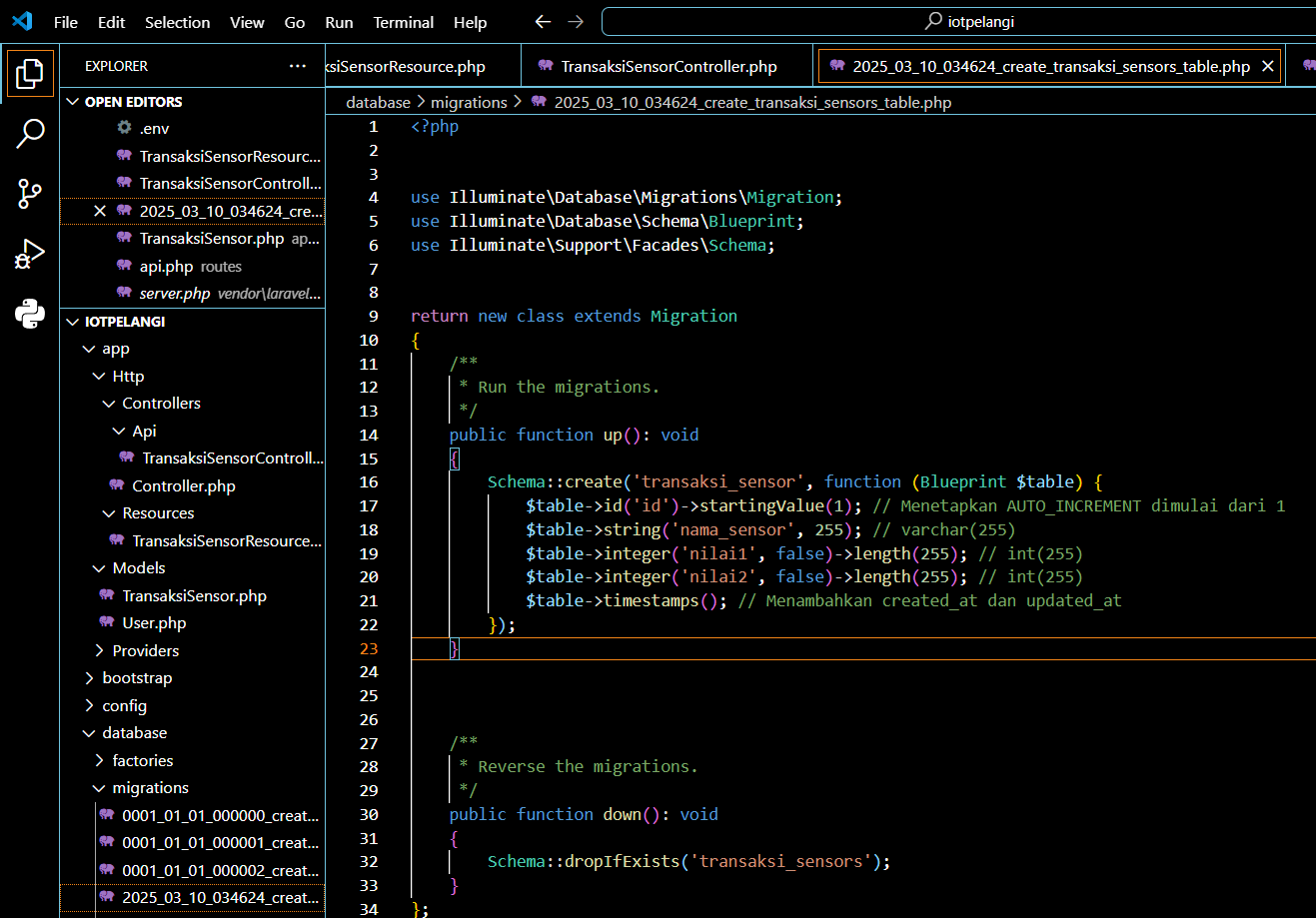
* Database at phpmyadmin (ior\_25) result



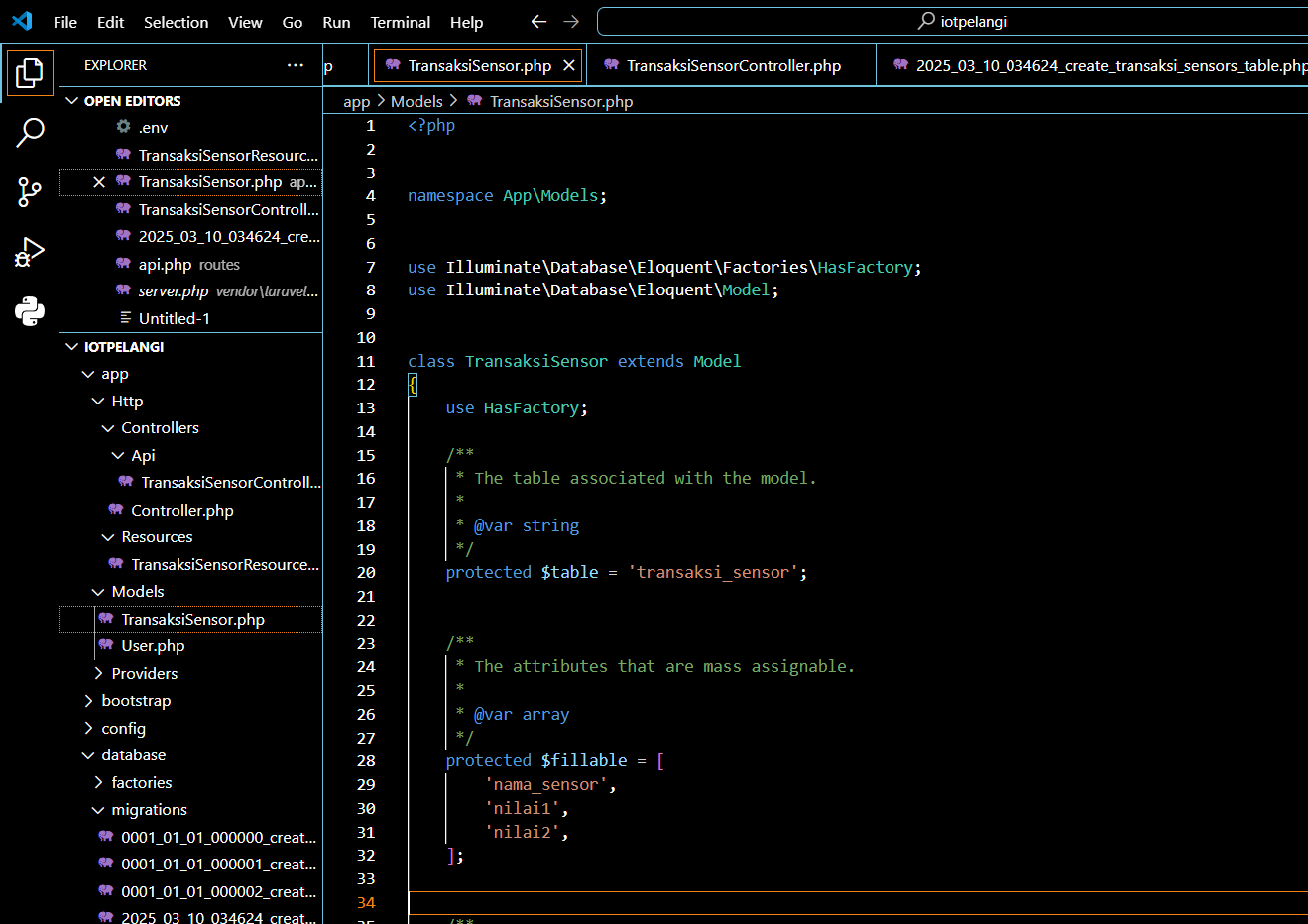
* File model TransaksiSensor.php result



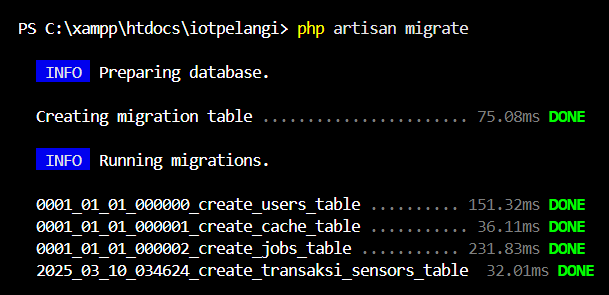
* Fille (**2025\_02\_21\_074123\_create\_transaksi\_sensors\_table.php**) content conversion result



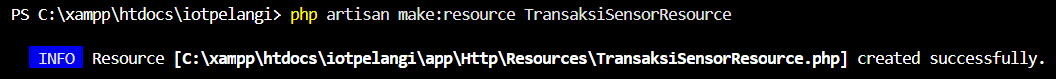
* Fille (**app/Models/TransaksiSensor.php**) content conversion result



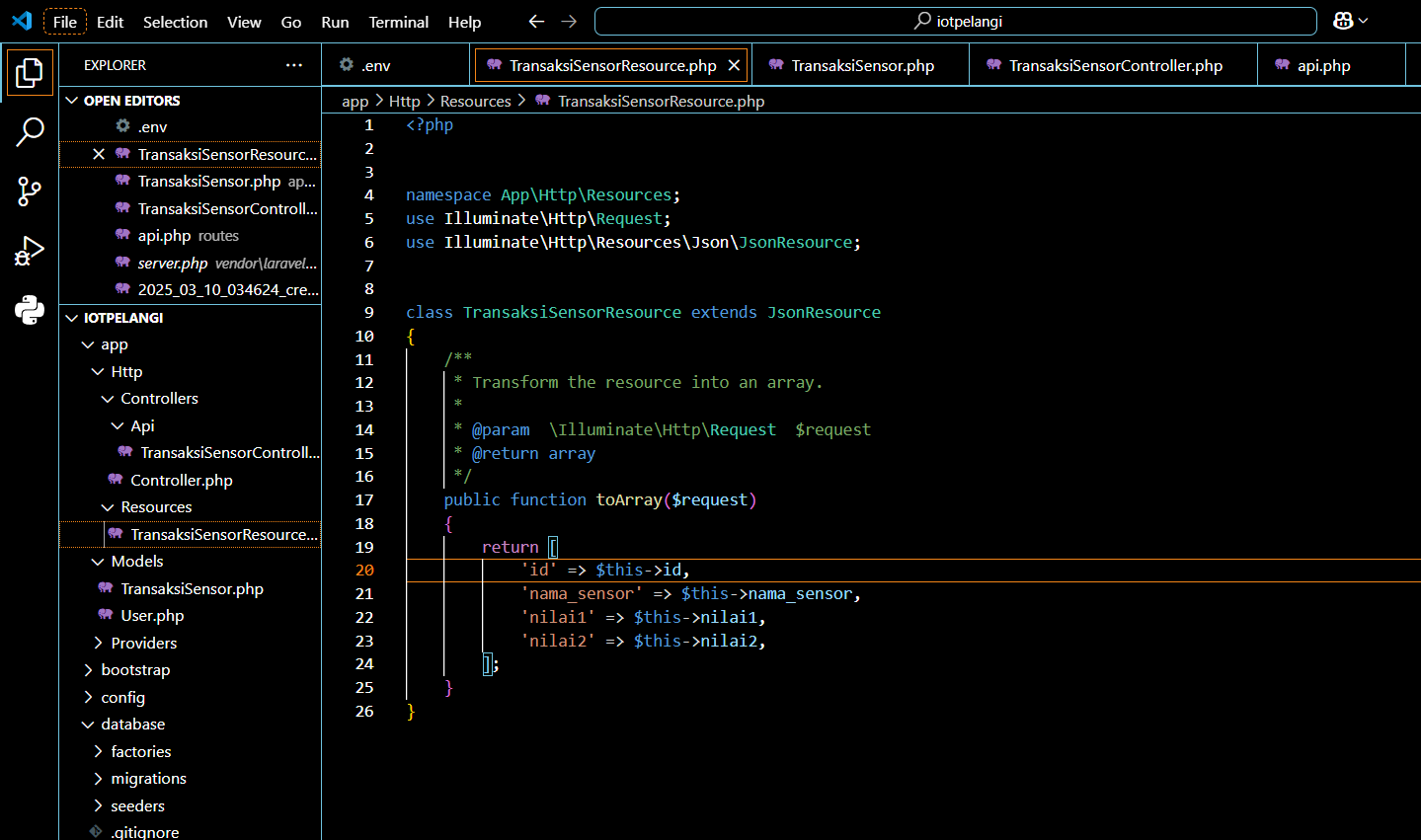
* Tabel (**php artisan migrate**) result



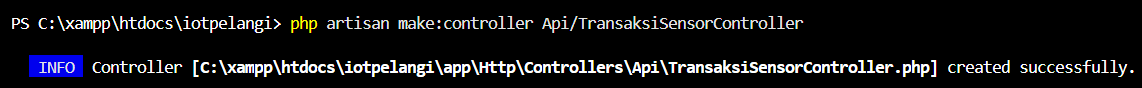
* Resource result



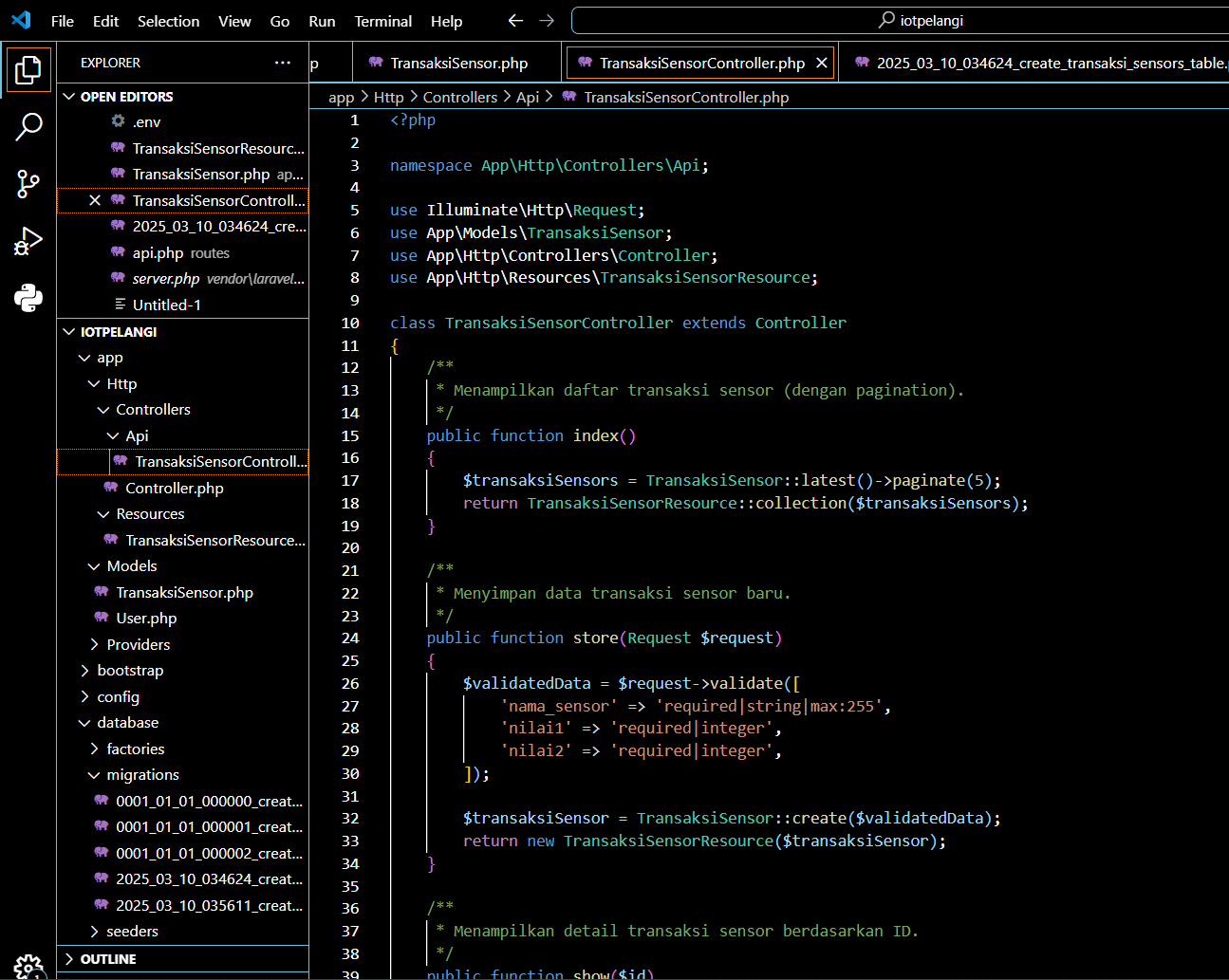
* Fille (**TransaksiSensorResource.php**) content conversion result



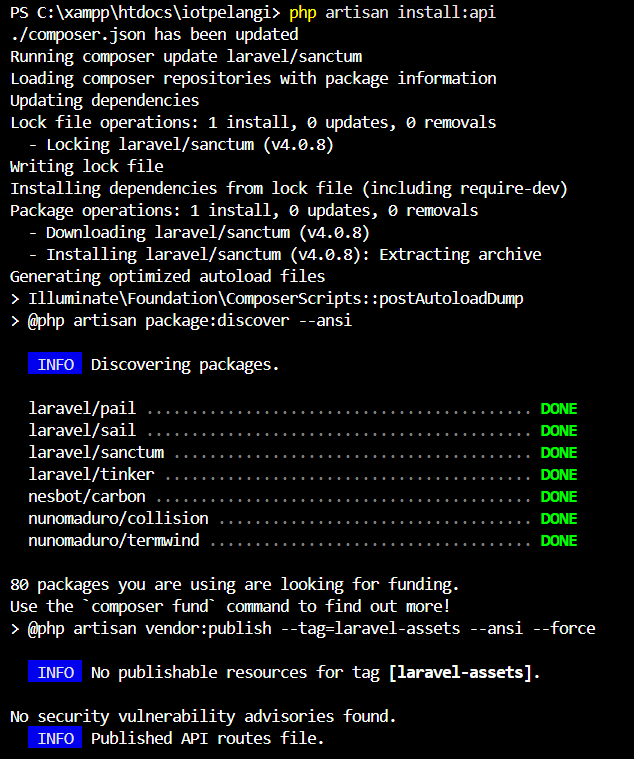
* API controller result



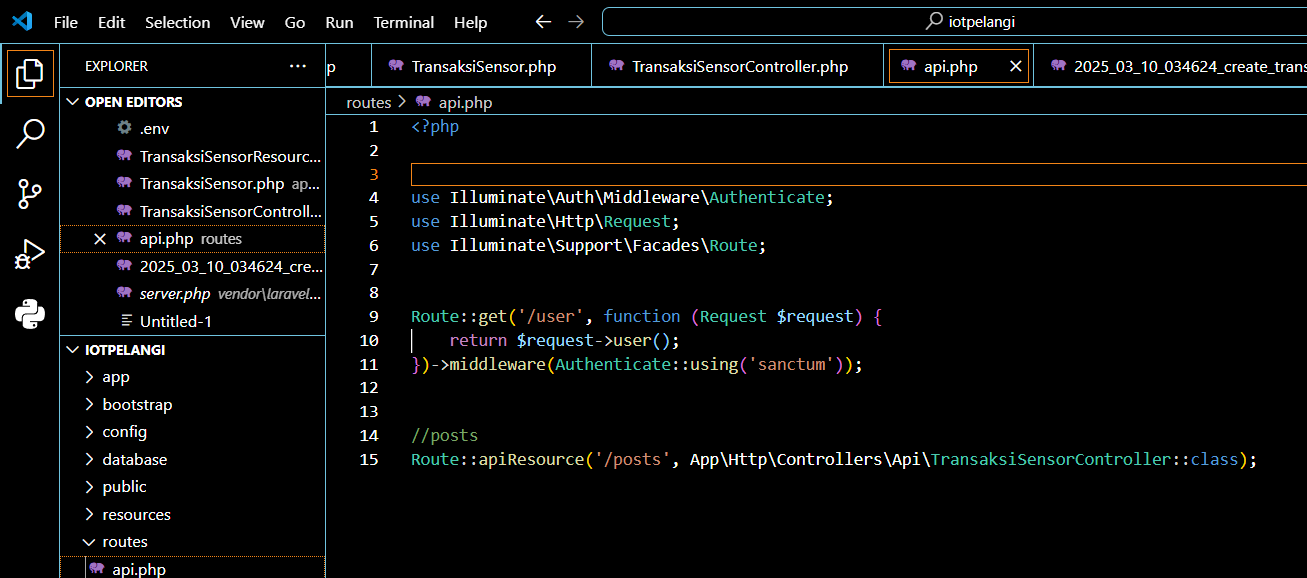
* Fille (**app/Http/Controllers/Api/TransaksiSensorController.php**) content conversion result



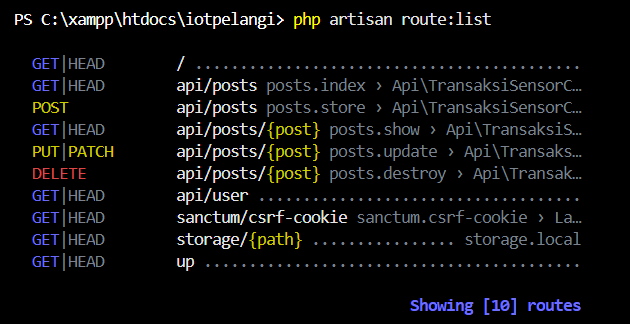
* API-specific route results



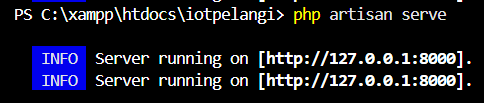
* Fille (**routes/api.php**) content conversion result



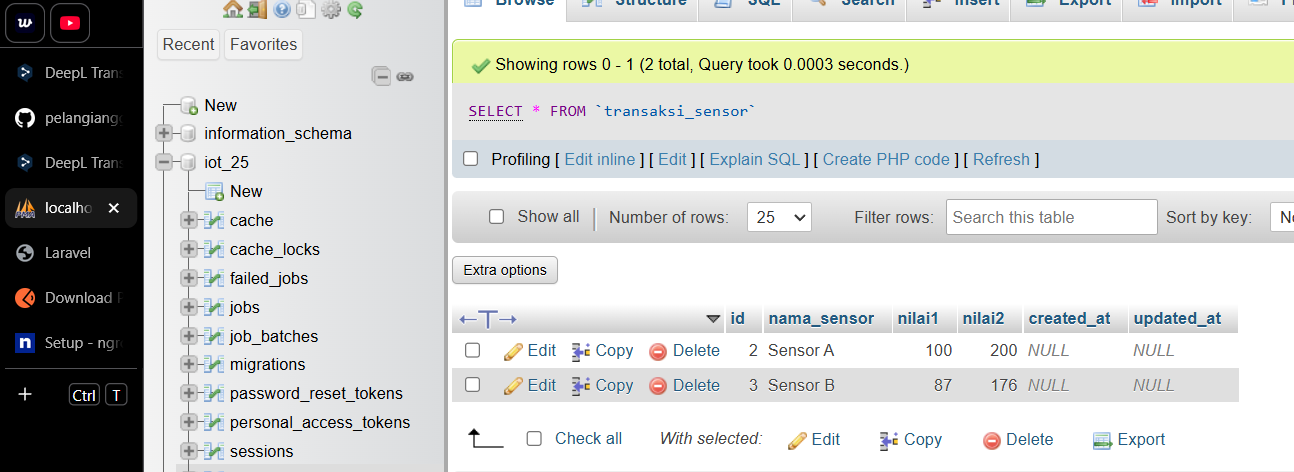
* Routes List result



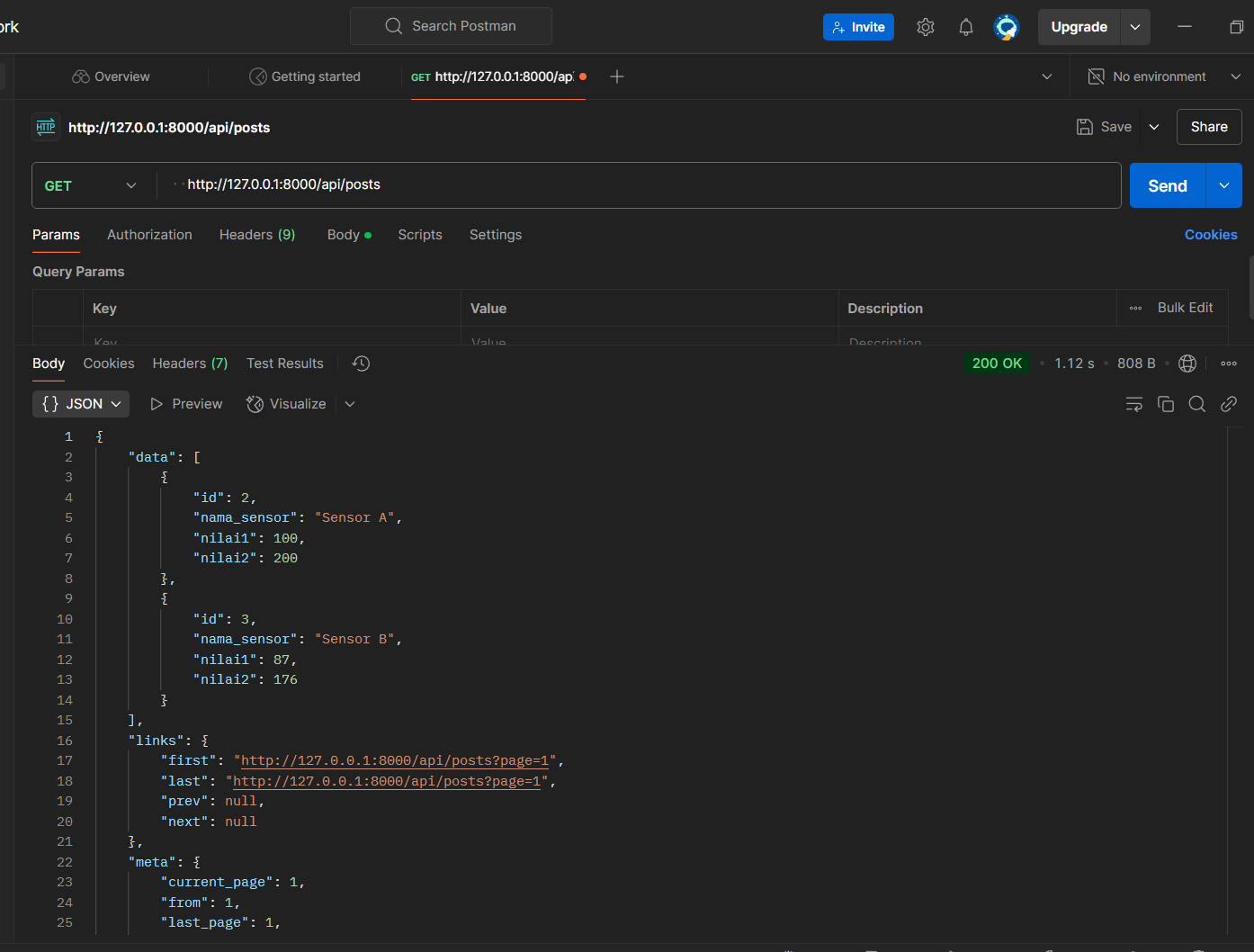
* API access result



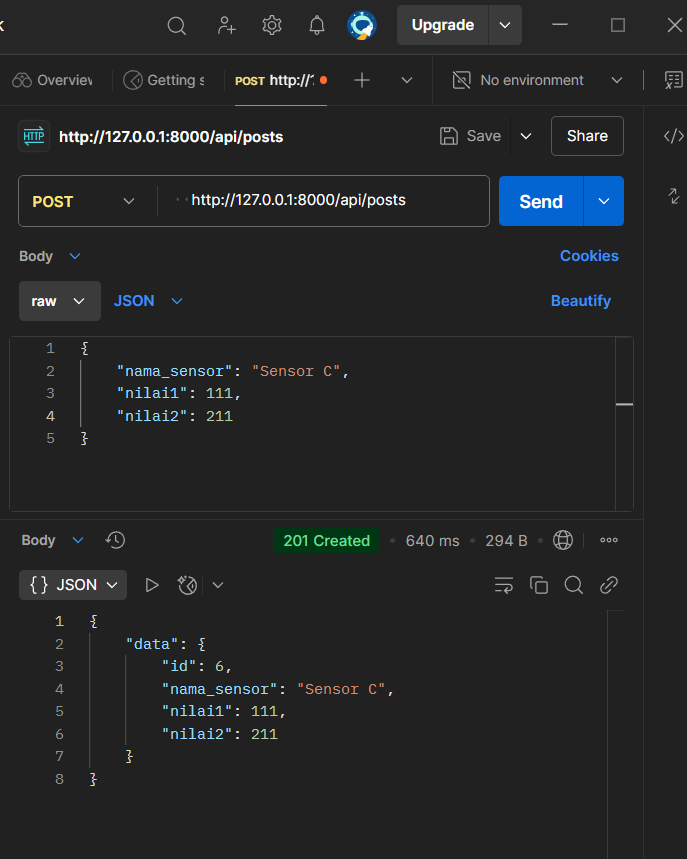
* Two Initial data result



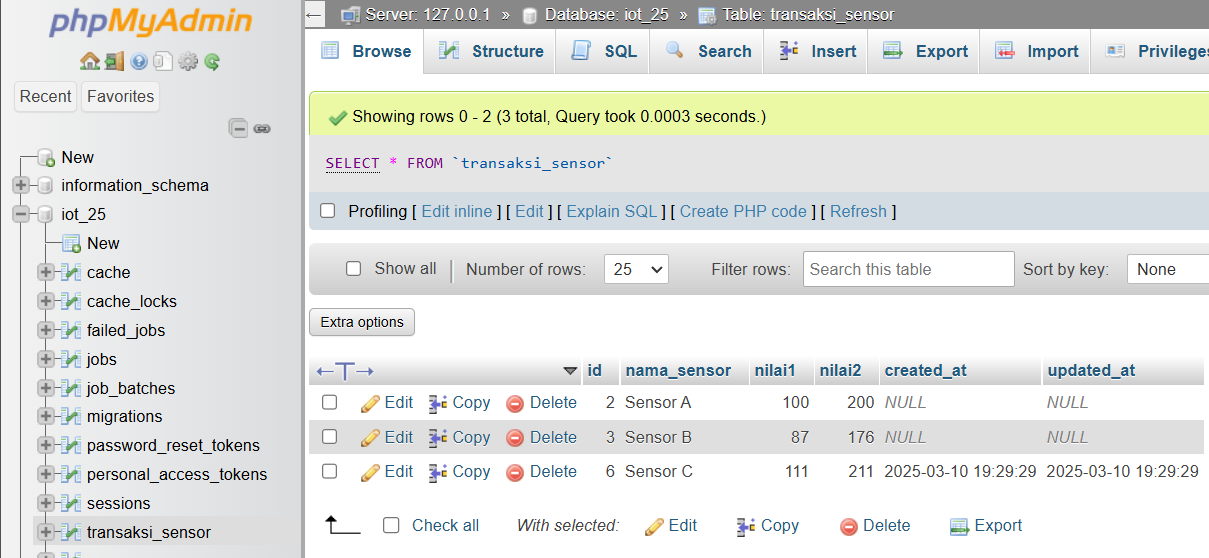
* Results of retrieving two initial data through the postman application



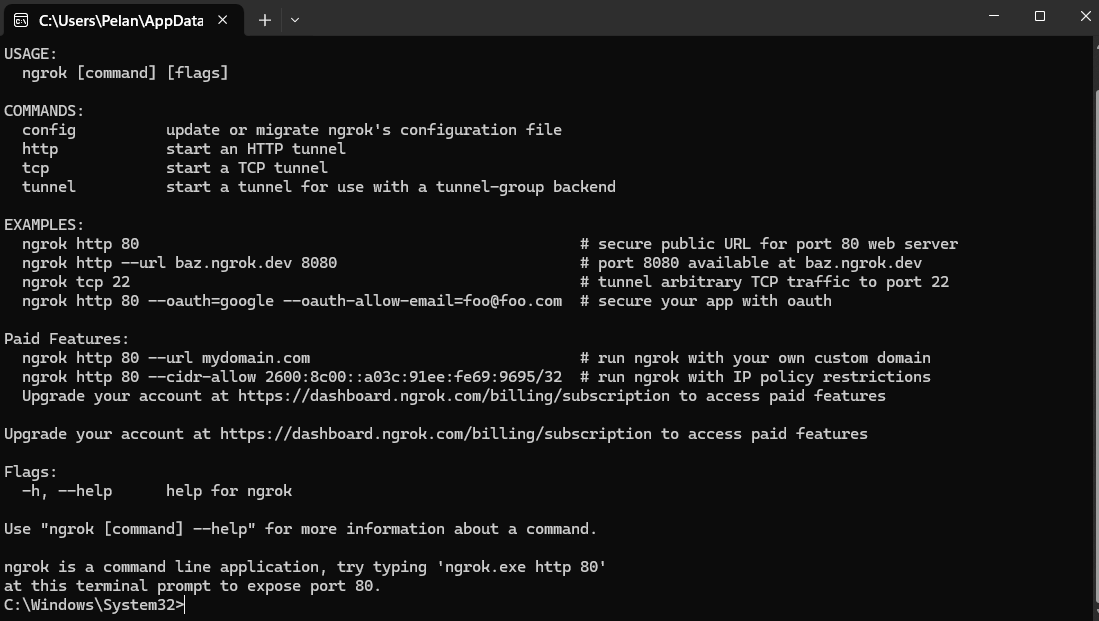
* Result of adding data in postman



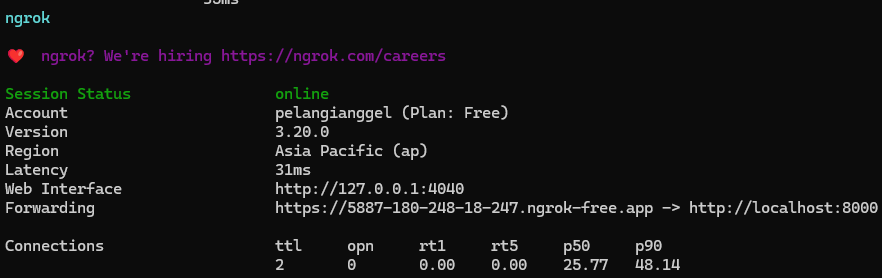
* The result in phpmyadmin



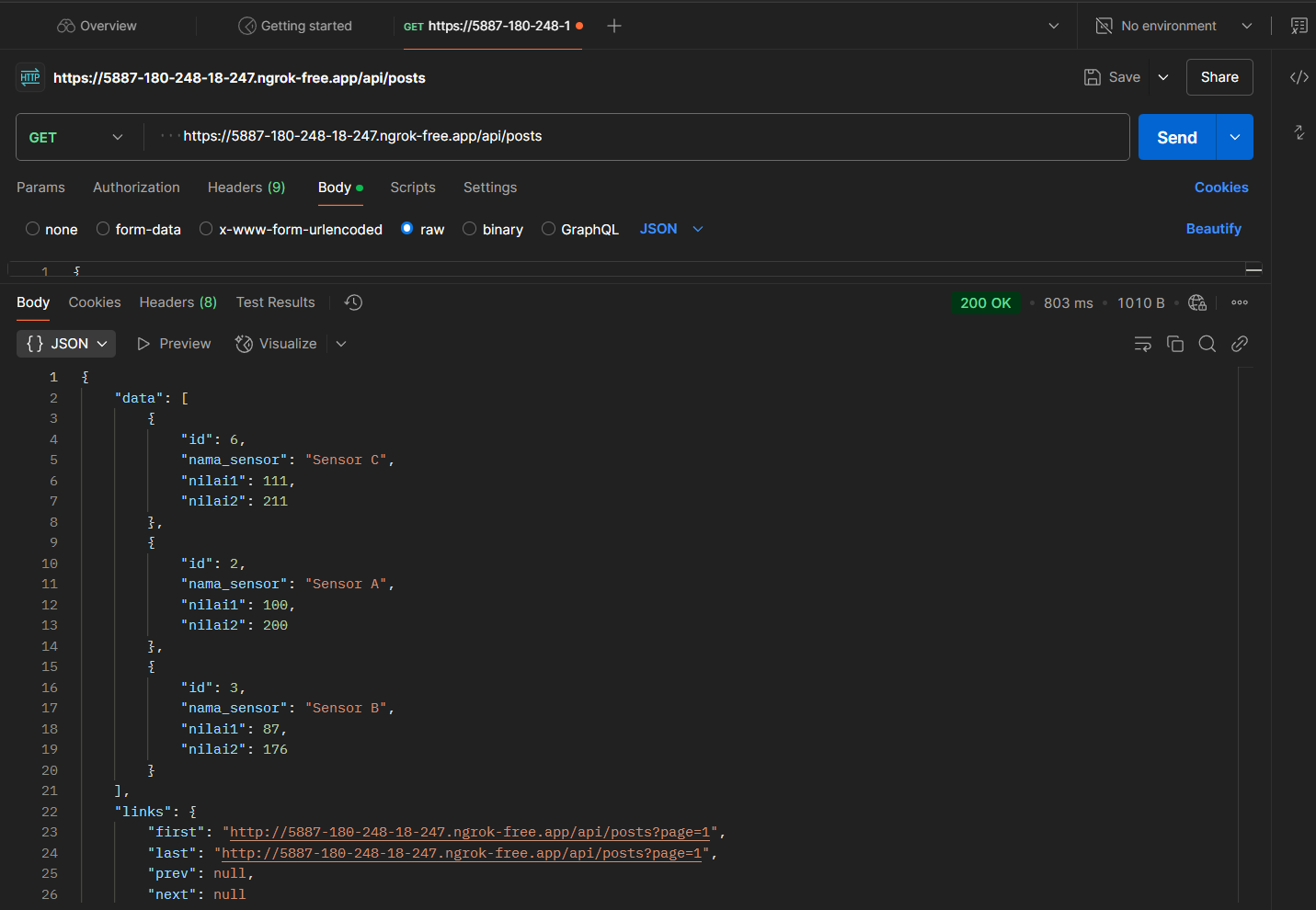
* Ngrok result



* Mengonline kan laravel



* Get API experiment results



* Result of inserting new data via API

