

Laporan Minggu 6 Praktikum Wokwi

**Pembuatan API
Menggunakan Laravel 11 dan Ngrok**

Gunawan Danyarto

University of Brawijaya Vocation Faculty

Email : gunawandany11@gmail.com

Abstract

This experiment aims to develop a RESTful API using Laravel 11 and expose it to the internet via Ngrok. Laravel, a powerful PHP framework, is used for building robust APIs, while Ngrok helps in local API tunneling. This report covers the setup process, API development, and integration with Ngrok for external accessibility. The results confirm that the API is successfully created, tested, and accessed over the internet using Ngrok.

1. Introduction

1.1 Background of the IoT Experiment

Developing APIs is a crucial aspect of modern IoT systems, allowing seamless communication between devices and applications. Laravel 11 simplifies API creation, while Ngrok provides an easy way to expose local servers for testing. This experiment demonstrates the process of building an API and making it accessible over the internet.

1.2 Objective of the Experiment

1. Set up Laravel 11 for API development.
2. Create and test a simple REST API in Laravel.
3. Use Ngrok to expose the API for external access.
4. Verify API functionality using tools like Postman or a web browser

2. Methodology

2.1 Tools & Materials

Framework: Laravel 11

Server: Ngrok

Database: MySQL

Tools: Postman, XAMPP

IDE: VS Code

2.2 Implementation Steps

2.2.1 Laravel 11 Setup

1. Install Laravel 11 using Composer:

```
composer create-project laravel/laravel my-api
```

2. Navigate to the project directory:

```
cd my-api
```

3. Serve the application locally:

```
php artisan serve
```

2.2.2 API Development

1. Create a new controller for the API:

```
php artisan make:controller Api/TransaksiSensorController --api
```

2. Define routes in routes/api.php:

```
use App\Http\Controllers\Api\TransaksiSensorController;  
Route::apiResource('transaksi-sensor',  
    TransaksiSensorController::class);
```

3. Implement CRUD methods in TransaksiSensorController.php.

2.2.3 Exposing API with Ngrok

1. Start Laravel server:

```
php artisan serve --host=127.0.0.1 --port=8000
```

2. Run Ngrok to create a public URL:

```
ngrok http 8000
```

3. Copy the generated Ngrok URL and test it using Postman.

```
Cmder
Server version: 8.0.30 MySQL Community Server - GPL
Copyright (c) 2000, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

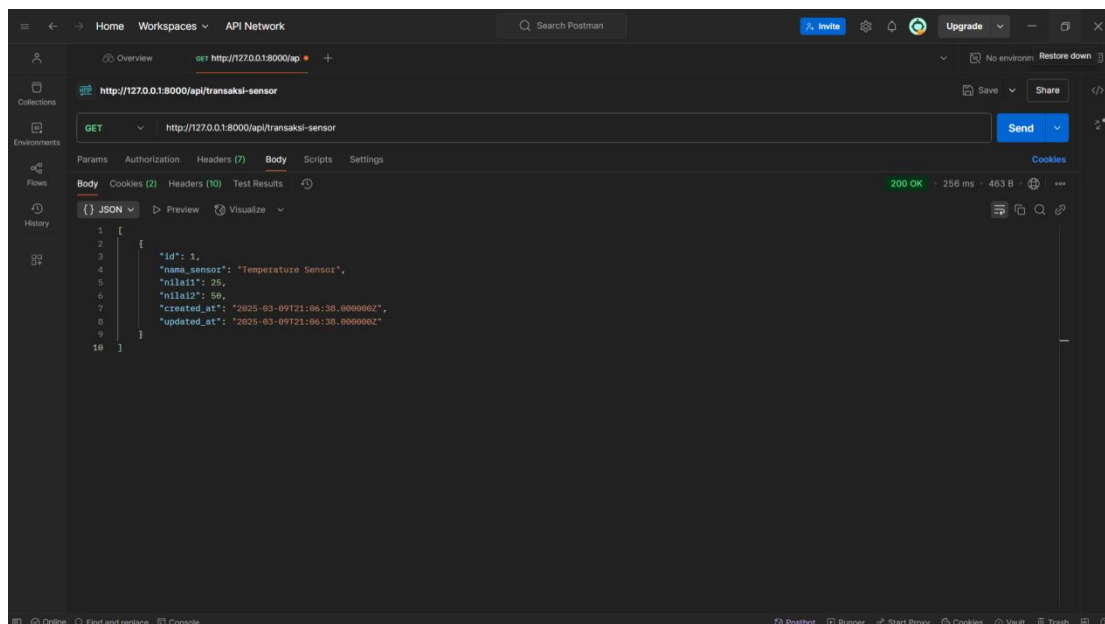
mysql> USE iot_25;
Database changed
mysql> SELECT * FROM transaksi_sensor;
Empty set (0.00 sec)

mysql> INSERT INTO transaksi_sensor (sensor_name, value, created_at, updated_at) VALUES ('Temperature Sensor', 25.5, NOW(), NOW());
ERROR 1054 (42S22): Unknown column 'sensor_name' in 'field list'
mysql> DESC transaksi_sensor;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id | bigint unsigned | NO | PRI | NULL | auto_increment |
| nama_sensor | varchar(255) | NO | | NULL | |
| nilai1 | int | NO | | NULL | |
| nilai2 | int | NO | | NULL | |
| created_at | timestamp | YES | | NULL | |
| updated_at | timestamp | YES | | NULL | |
+-----+
6 rows in set (0.01 sec)

mysql> SHOW COLUMNS FROM transaksi_sensor;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id | bigint unsigned | NO | PRI | NULL | auto_increment |
| nama_sensor | varchar(255) | NO | | NULL | |
| nilai1 | int | NO | | NULL | |
| nilai2 | int | NO | | NULL | |
| created_at | timestamp | YES | | NULL | |
| updated_at | timestamp | YES | | NULL | |
+-----+
6 rows in set (0.00 sec)

mysql> INSERT INTO transaksi_sensor (sensor_data, created_at, updated_at) VALUES (25.5, NOW(), NOW());
ERROR 1054 (42S22): Unknown column 'sensor_data' in 'field list'
mysql> \c
mysql> INSERT INTO transaksi_sensor (nama_sensor, nilai1, nilai2, created_at, updated_at) VALUES ('Temperature Sensor', 25, 50, NOW(), NOW());
Query OK, 1 row affected (0.00 sec)

mysql> |
```



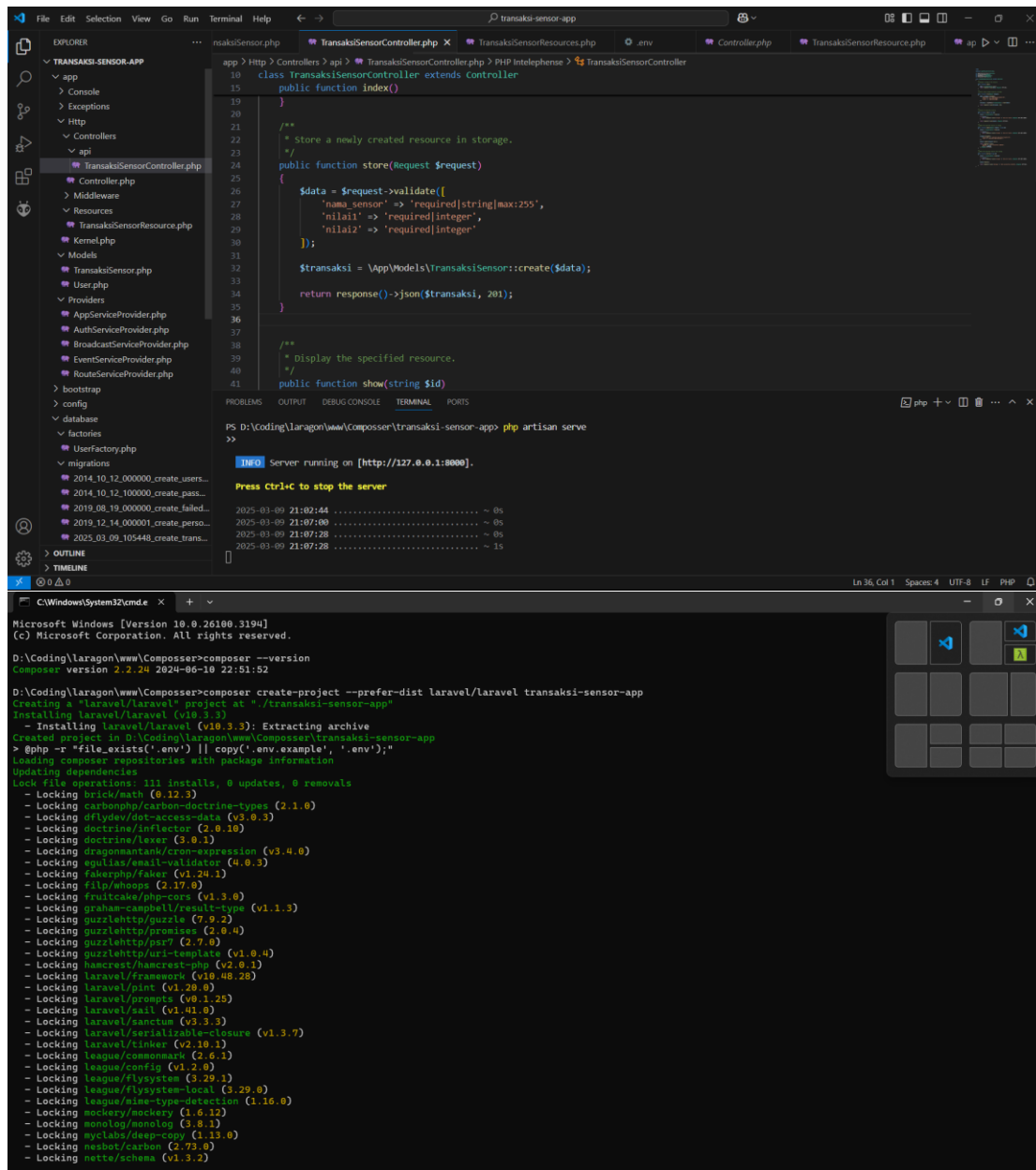
3. Results and Discussion

3.1 Experimental Results

The API successfully responded to GET, POST, PUT, and DELETE requests.

The Ngrok URL provided secure external access to the API.

API calls via Postman and the browser returned expected JSON responses.



3.2 Discussion

- 1. Benefits of Laravel 11:** Simplifies API development with built-in routing and middleware.
- 2. Advantages of Ngrok:** Enables quick testing by exposing local servers to the internet.
- 3. Challenges:** Authentication and security measures should be implemented before deploying publicly.

The screenshot displays two windows side-by-side. The left window is a web browser at the URL `https://16db-111-94-127-62.ngrok-free.app/api/transaksi`, showing a JSON response with two sensor records. The right window is a terminal running the `ngrok` command, displaying the service's status, account information, and a list of incoming HTTP requests.

```
{
  "id": 1,
  "nama_sensor": "Sensor C",
  "nilai1": 111,
  "nilai2": 221,
  "created_at": null,
  "updated_at": null
},
{
  "id": 2,
  "nama_sensor": "Sensor C",
  "nilai1": 111,
  "nilai2": 221,
  "created_at": "2025-03-09T14:41:51.000000Z",
  "updated_at": "2025-03-09T14:41:51.000000Z"
}
}
```

```
ngrok
ngrok
* Found a bug? Let us know: https://github.com/ngrok/ngrok
Session Status      online
Account             GunawanDanyarto (Plan: Free)
Version             3.28.0
Region              Asia Pacific (ap)
Latency             73ms
Web Interface       [REDACTED]
Forwarding           [REDACTED]
Connections          ttl    opn    rt1    rt5    p50    p90
                   20     0     0.16   0.05   0.08   0.23

HTTP Requests
-----
22:01:53.440 +07 GET /favicon.ico           200 OK
22:01:53.501 +07 GET /api/transaksi-sensor  200 OK
22:01:53.060 +07 GET /api/transaksi-sensor  200 OK
22:01:53.039 +07 GET /favicon.ico           200 OK
22:01:53.859 +07 GET /favicon.ico           200 OK
22:01:52.490 +07 GET /favicon.ico           200 OK
22:01:52.690 +07 GET /api/transaksi-sensor  200 OK
22:01:52.118 +07 GET /api/transaksi-sensor  200 OK
22:01:49.756 +07 GET /favicon.ico           200 OK
22:01:49.357 +07 GET /api/transaksi-sensor  200 OK
```

5. Conclusion

This experiment demonstrated the successful creation of a RESTful API using Laravel 11 and its exposure using Ngrok. The integration allowed seamless external access, confirming the API's functionality. This method is useful for testing and rapid development in IoT applications.