# LAPORAN PROJEK

# KELOMPOK 8

****

Oleh :

Moch Dava Dwi Putra 20230040131

Fauzan Alifian Fadhlurrahman 20230040202

# TEKNIK INFORMATIKA

# UNIVERSITAS NUSA PUTRA

# 2025

[LAPORAN PROJEK 1](#_Toc188609857)

[KELOMPOK 8 1](#_Toc188609858)

[TEKNIK INFORMATIKA 1](#_Toc188609859)

[UNIVERSITAS NUSA PUTRA 1](#_Toc188609860)

[2025 1](#_Toc188609861)

[BAB I 3](#_Toc188609862)

[PENDAHULUAN 3](#_Toc188609863)

[1. Latar Belakang 3](#_Toc188609864)

[2. Rumusan Masalah 3](#_Toc188609865)

[3. Tujuan 3](#_Toc188609866)

[BAB II 4](#_Toc188609867)

[METODOLOGI 4](#_Toc188609868)

[1. Desain Sistem 4](#_Toc188609869)

[2. Perancangan Website 5](#_Toc188609870)

[a.Use Case Diagram 5](#_Toc188609871)

[b.Use Case Utama 5](#_Toc188609872)

[3. Teknologi yang Digunakan 11](#_Toc188609873)

[a.Express.js 11](#_Toc188609874)

[4. MySQL 11](#_Toc188609875)

[BAB III 12](#_Toc188609876)

[PENGUJIAN 12](#_Toc188609877)

[Berisi pengujian API testing dari aplikasi 12](#_Toc188609878)

[BAB IV 20](#_Toc188609879)

[PENUTUP 20](#_Toc188609880)

[1. Kesimpulan 20](#_Toc188609881)

[2. Saran 20](#_Toc188609882)

# BAB I

# PENDAHULUAN

## 1. Latar Belakang

Dalam era digitalisasi yang terus berkembang, teknologi informasi memegang peran penting dalam berbagai aspek kehidupan, termasuk dalam dunia bisnis. Salah satu bentuk implementasi teknologi informasi yang signifikan adalah penggunaan sistem berbasis database dan bahasa pemrograman untuk mendukung operasional bisnis. Sistem semacam ini mampu meningkatkan efisiensi, akurasi, dan kemudahan dalam pengelolaan data, termasuk dalam kegiatan penjualan produk.

Velg kendaraan merupakan salah satu komponen yang memiliki permintaan pasar yang tinggi, baik untuk kebutuhan penggantian maupun modifikasi. Oleh karena itu, penting bagi para pelaku bisnis di sektor ini untuk memiliki sistem manajemen penjualan yang efektif. Dengan adanya sistem berbasis database yang terintegrasi dengan aplikasi berbasis JavaScript, pelaku bisnis dapat mengelola data produk, transaksi, dan pelanggan secara lebih terstruktur dan efisien.

## 2. Rumusan Masalah

1. Bagaimana merancang sistem penjualan velg yang terintegrasi dengan database untuk mempermudah pengelolaan data produk, transaksi, dan pelanggan?
2. Bagaimana sistem ini dapat meningkatkan efisiensi operasional, mengurangi kesalahan data, dan memberikan pengalaman pengguna yang lebih baik?

## 3. Tujuan

1. Merancang dan mengembangkan sistem penjualan velg berbasis database yang mampu mengelola data produk, transaksi, dan pelanggan secara terstruktur dan efisien.
2. Meningkatkan efisiensi operasional bisnis penjualan velg dengan mengurangi potensi kesalahan data dan mempercepat proses transaksi.

# BAB II

# METODOLOGI

## 1. Desain Sistem

Sistem ini dirancang dan dikembangkan untuk memudahakan penjualan dengan menggunakan sistem penjualan velg berbasis database yang mampu mengelola data produk, transaksi, dan pelanggan secara terstruktur dan efisien.

Meningkatkan efisiensi operasional bisnis penjualan velg dengan mengurangi potensi kesalahan data dan mempercepat proses transaksi, dan Memberikan solusi yang mendukung pelaku usaha dalam mengoptimalkan manajemen penjualan dan meningkatkan pengalaman pelanggan.

1. Platform Berbasis Web

System ini berbasis web agar mudah di akses oleh konsumen melalui berbagai perangkat, seperti laptop, tablet, atau smartphone. Platform ini menggunakan pendekatan desain yang variative agar konsumen mudah dalam memutuskan untuk membeli velk yang di inginkan.

1. Halaman Beranda
2. Berisi pengantar tentang platform, termasuk tujuan dan manfaatnya
3. Menampilkan fitur seperti pencarian produk, katalog produk, dan keranjang.
4. Akun konsumen
5. Konsumen daoat membuat akun untuk mengakses fitur utama, seperti mencari produk
6. Setiap akun dapat mengetahui tentang informasi ketersediaan barang yang ada
7. Fitur utama system
8. Halaman interaktif yang menampilkan seluruh produk dengan kategori, harga, dan deskripsi.
9. Fitur filter dan sort untuk mempermudah pencarian produk.
10. System menyediakan fitur keranjang agar pengguna dapat menambahkan produk ke keranjang belanja untuk pembelian.
11. Sistem pencarian dan rekomendasi
12. Pencarian berbasis filter dan sort unutk memudahkan pencarian produk
13. Algoritma rekomendasi dengan menampilkan produk dengan mengikuti produk yang konsumen cari
14. Manajemen Produk
15. Penambahan, pengeditan, dan penghapusan data produk (nama, tipe, harga, stok, dan gambar).
16. Sistem notifikasi untuk stok produk yang hampir habis.
17. Panel Admin
18. Administrator memiliki akses ke panelnuntuk mengelola pengguna, produk, ketersediaan barang.
19. Alur Sistem Secara Keseluruhan
20. Pendaftaran dan LogIn pengguna
21. Pencarian dan Pemilihan Produk
22. Pengelolaan Keranjang Belanja
23. Proses Chekout dan Pembayaran
24. Pengelolaan Pesanan Oleh Admin
25. Pengiriman dan pelacakan

## 2. Perancangan Website

### Use Case Diagram

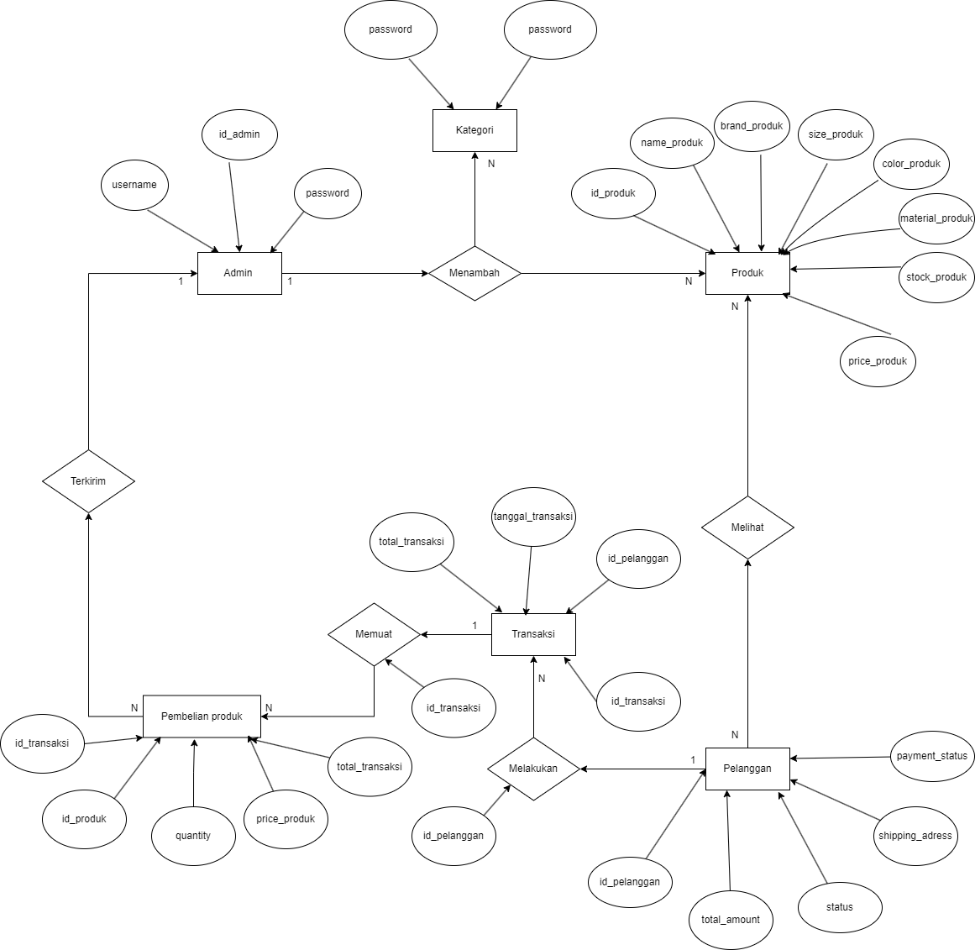


### b. Use Case Utama

Ada dua aktor utama untuk use case ini, pertama ada pengguna atau konsumen yang menggunakan platform ini utntuk mencari dan membeli produk. Kedua administrator, pengelola yang bertanggung jawab atas update setiap produk. Berikut alurnya:

1. Pendaftaran dan Login (konsumen)
2. Pencarian dan pemilihan produk (konsumen)
3. Manajemen keranjang belanja (konsumen)
4. Checkout dan pembayaran (konsumen)
5. Pengelolaan produk (admin)
6. Pengelolaan pesanan (admin)
7. Laporan penjualan (admin)
8. Pelacakan pengiriman (konsumen)
   1. ERD (Enttity-Relationship Diagram) database
9. Penjelasan

ERD menunjukan struktur basis data untuk system ini, termasuk entitas, atribut, dana hubunkdcmfmgan antar entitas. Berikut adalah entitas utama dan hubungannya:



* 1. Hubungan Antar Entitas

1. **Admin dan Produk (1:N)**  
   Admin memiliki hubungan 1:N dengan Produk, karena satu admin dapat menambahkan beberapa produk ke dalam sistem. Namun, setiap produk hanya bisa ditambahkan oleh satu admin.
2. **Kategori dan Produk (1:N)**  
   Kategori memiliki hubungan 1:N dengan Produk, karena satu kategori dapat memiliki banyak produk (misalnya, kategori velg mobil dapat memiliki beberapa produk velg berbeda). Sebaliknya, setiap produk hanya memiliki satu kategori.
3. **Produk dan Transaksi (N:M)**  
   Produk memiliki hubungan N:M dengan Transaksi melalui entitas "Pembelian Produk". Ini berarti satu produk dapat muncul dalam banyak transaksi (dibeli oleh banyak pelanggan), dan satu transaksi dapat mencakup banyak produk (pelanggan membeli lebih dari satu produk).
4. **Transaksi dan Pelanggan (1:N)**  
   Transaksi memiliki hubungan 1:N dengan Pelanggan, karena satu pelanggan dapat melakukan beberapa transaksi (misalnya, membeli di waktu yang berbeda). Namun, setiap transaksi hanya dilakukan oleh satu pelanggan.
5. **Pelanggan dan Transaksi (1:N)**  
   Hubungan ini menggambarkan bahwa satu pelanggan dapat melakukan banyak transaksi (misalnya, belanja berulang kali), tetapi satu transaksi hanya dapat dimiliki oleh satu pelanggan.
6. **Pelanggan dan Produk (N:M)**  
   Pelanggan memiliki hubungan N:M dengan Produk melalui entitas "Transaksi" dan "Pembelian Produk". Ini berarti satu pelanggan dapat membeli banyak produk, dan satu produk dapat dibeli oleh banyak pelanggan.
7. **Transaksi dan Pembelian Produk (1:N)**  
   Transaksi memiliki hubungan 1:N dengan Pembelian Produk, karena satu transaksi dapat mencakup beberapa item produk (misalnya, pelanggan membeli beberapa jenis velg dalam satu transaksi).
8. **Pembelian Produk dan Produk (N:1)**  
   Pembelian Produk memiliki hubungan N:1 dengan Produk, karena banyak pembelian produk dapat mengacu pada produk yang sama (misalnya, pelanggan A dan pelanggan B membeli velg dengan model yang sama).
9. **Transaksi dan Pengiriman (1:1 atau 1:N)**  
   Setiap transaksi memiliki hubungan 1:1 atau 1:N dengan status pengiriman ("Terkirim"), tergantung pada desain sistem. Setiap transaksi dapat memiliki status pengiriman seperti "Terkirim", "Dalam Pengiriman", atau "Belum Dikirim"
   1. Endpoint

a. Authentication Endpoints

* + - 1. OST/register
    - Parameter : username, email, password
    - Response : user data & token

2. POST/login

* + - Parameter : email, password
    - Response : Token JWT

3. GET/ Profile

* + - Header : Bearer token
    - Response : User Profile data

4. PUT/profile

* + - Header : Breaker Token
    - Response : User data
    - Response : Update profile

b. Product Endpoints

5. GET/products

* + - Query : page, limit, search, sort
    - Response : List products

6. GET /api/products/:id

- Parameter: id

- Response: Detail product

7. POST /api/products (Admin)

- Header: Bearer Token

- Body: Product data

- Response: Created product

8. PUT /api/products/:id (Admin)

- Header: Bearer Token

- Parameter: id

- Body: Updated data

- Response: Updated product

9. DELETE /api/products/:id (Admin)

- Header: Bearer Token

- Parameter: id

- Response: Success message

c. Order Endpoints

10. POST /api/orders

- Header: Bearer Token

- Body: Order details

- Response: Created order

11. GET /api/orders

- Header: Bearer Token

- Query: page, limit

- Response: List orders

12. GET /api/orders/:id

- Header: Bearer Token

- Parameter: id

- Response: Order detail

13. PUT /api/orders/:id/status (Admin)

- Header: Bearer Token

- Parameter: id

- Body: status

- Response: Updated order

14. GET /api/orders/user

- Header: Bearer Token

- Response: User's orders

d. Cart Endpoints

15. POST /api/cart

- Header: Bearer Token

- Body: Product & quantity

- Response: Updated cart

16. GET /api/cart

- Header: Bearer Token

- Response: Cart items

17. PUT /api/cart/:id

- Header: Bearer Token

- Parameter: id

- Body: Updated quantity

- Response: Updated cart

18. DELETE /api/cart/:id

- Header: Bearer Token

- Parameter: id

- Response: Updated cart

e. Admin Dashboard Endpoints

19. GET /api/admin/dashboard

- Header: Bearer Token

- Response: Dashboard stats

20. GET /api/admin/reports/sale

- Header: Bearer Token

- Query: start\_date, end\_date

- Response: Sales report

## 3. Teknologi yang Digunakan

Berikut adalah penjelasan dan kegunaan teknologi yang disebutkan dalam membangun program ini:

### a. Express.js

1. Penjelasan

Express.js adalah framework backend untuk Node.js yang ringan dan fleksibel, dan cepat. Frame work ini mempermudah pembuatan aplikasi web dan API dengan menyediakan berbagi fitur bawaan untuk menangani routing, middleware, dan interaksi database.

1. Kegunaan dalam program

1. Routing: mengatur jalur endpoint API seperti /register, /order, ordrer-items.

2. Middleware: menyediakan mekanisme untuk validasi data, autentikasi, dan login.

3. Manajemen API: mengelola interaksi antara fronted dan database melalui RESTful API.

4. Skalabilitas: mendukung pengembangan aplikasi yang beasr dan modular dengan struktur kode yang terorganiasi.

## 4. MySQL

a. Penjelasan

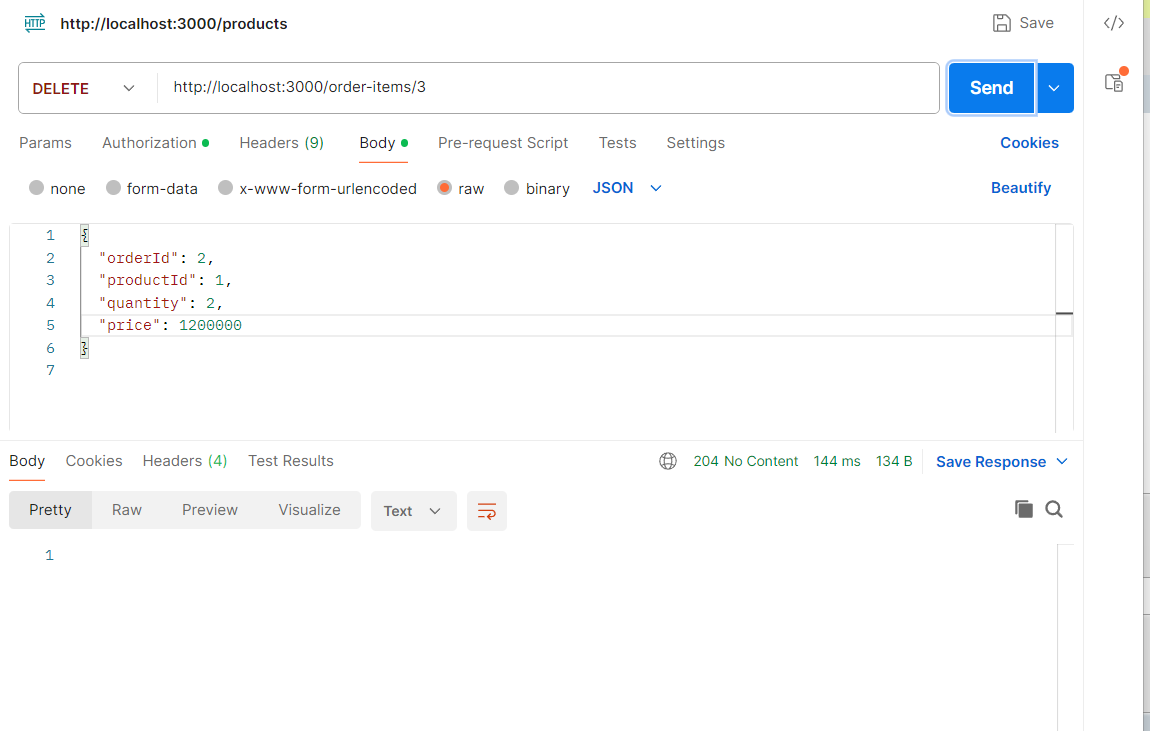
MySQL adalah sistem manajemen basis data relasional (RDBMS) yang populer, open-source, dan cocok untuk aplikasi web karena kecepatan, skalabilitas, dan kemudahan integrasinya dengan backend.

1. Kegunaan dalam Program
   1. Penyimpanan Data: Menyimpan data penting, seperti informasi pengguna, order, payment, dan laporan.
   2. Relasi Antar Data: Mengelola pembelian dengan satu entitas dapat membeli beberapa produk
2. Keamanan Data: Mendukung fitur keamanan seperti enkripsi password dan pembatasan akses data.

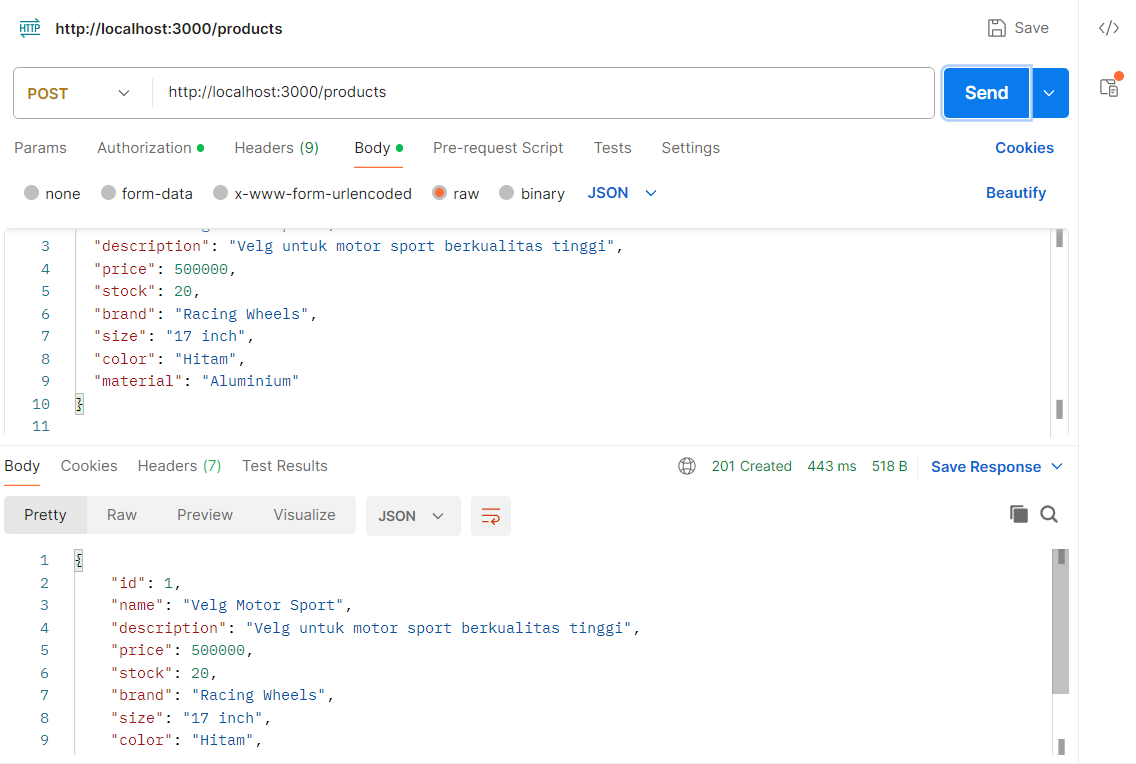
# BAB III

# PENGUJIAN

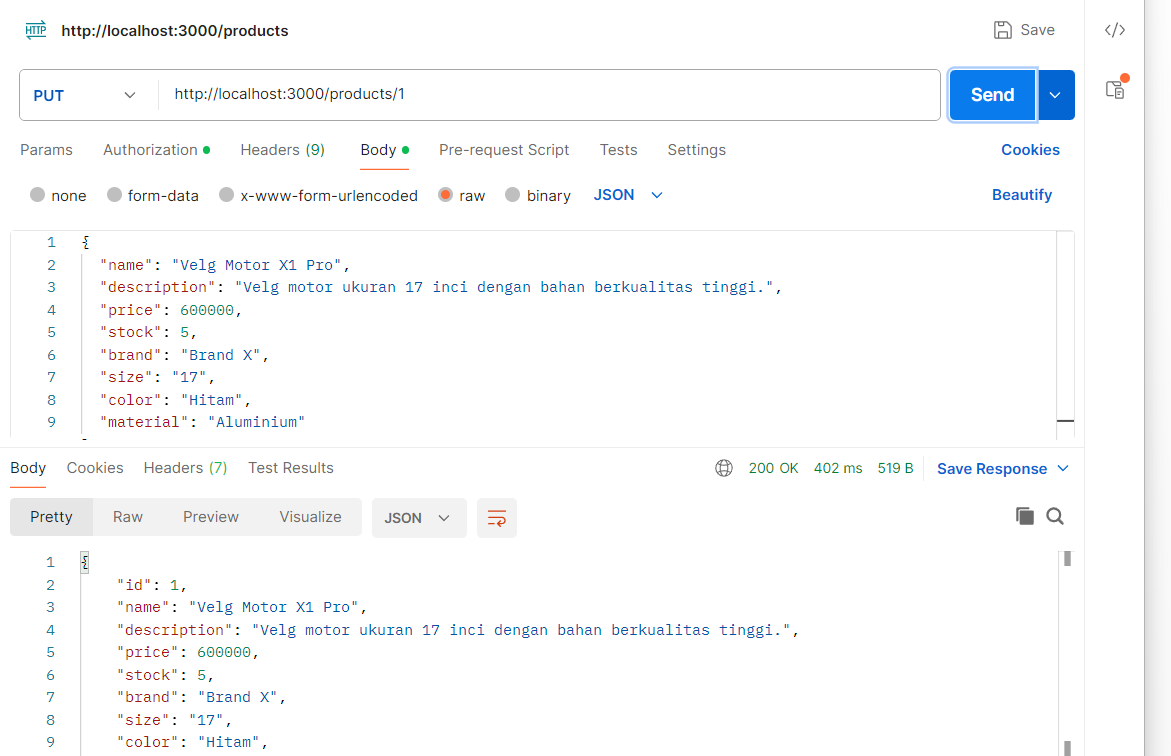
## Berisi pengujian API testing dari aplikasi

1.DELETE/order-items/{orderItemID}2.2.

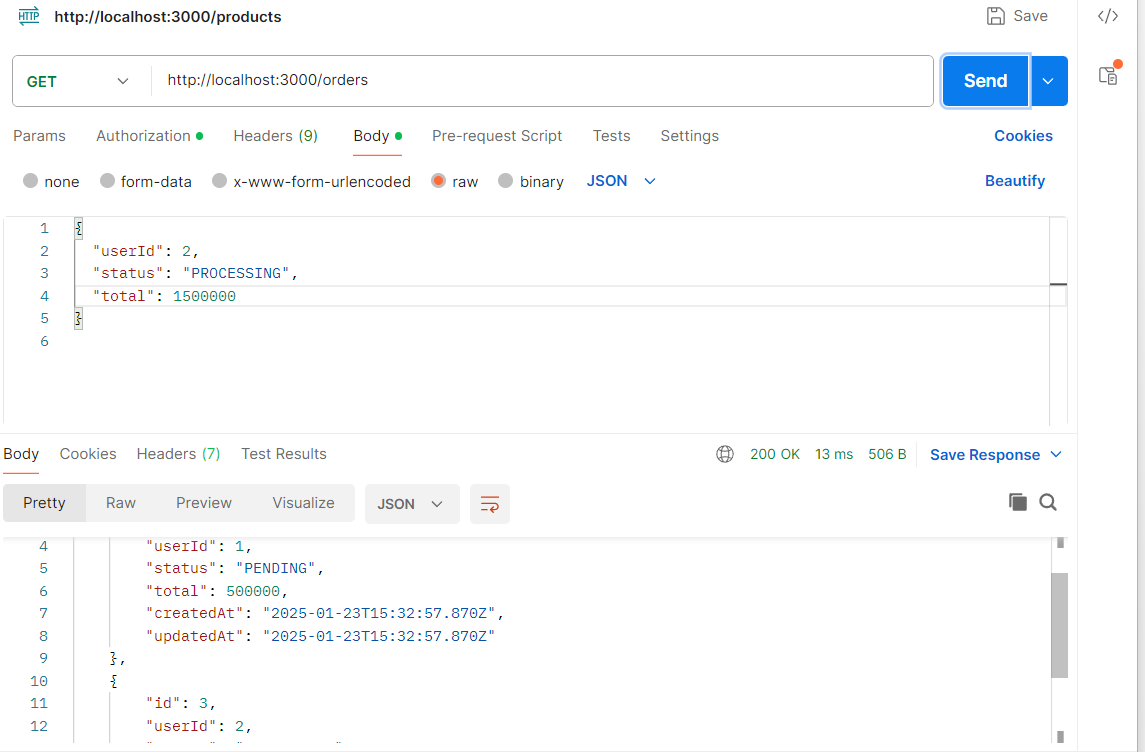
2.POST/PRODUCTS



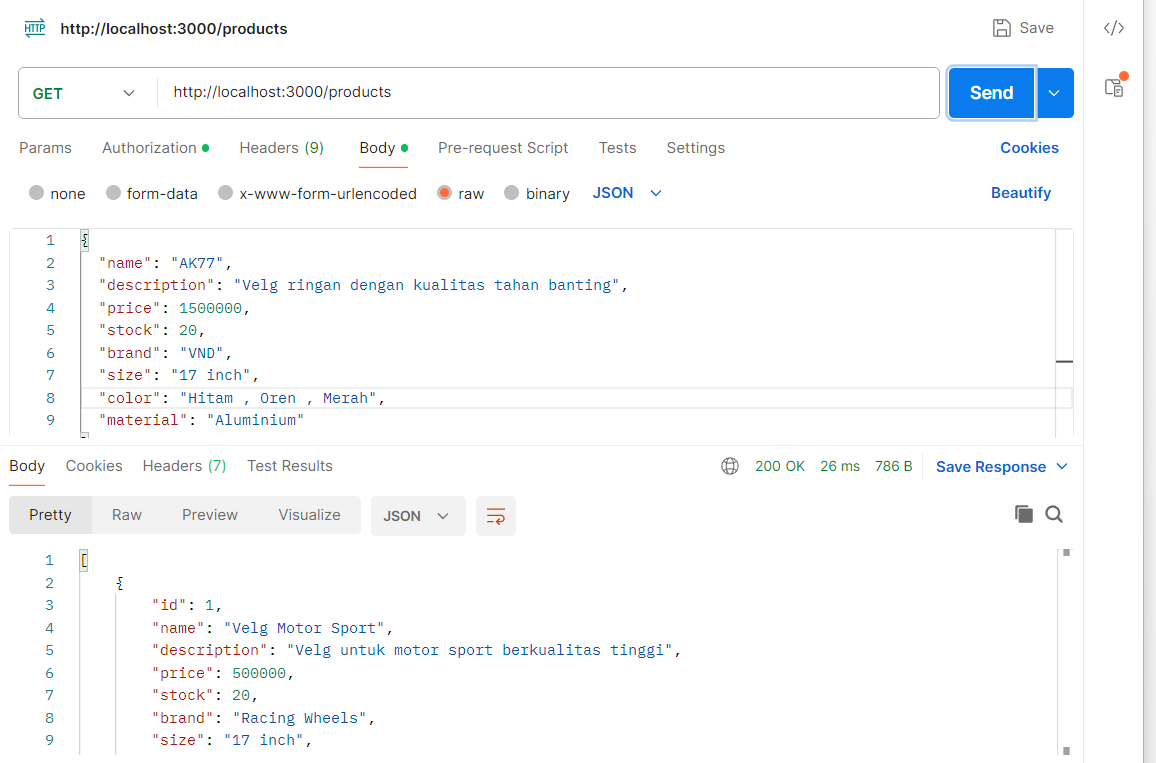
3. PUT/PRODUCTS

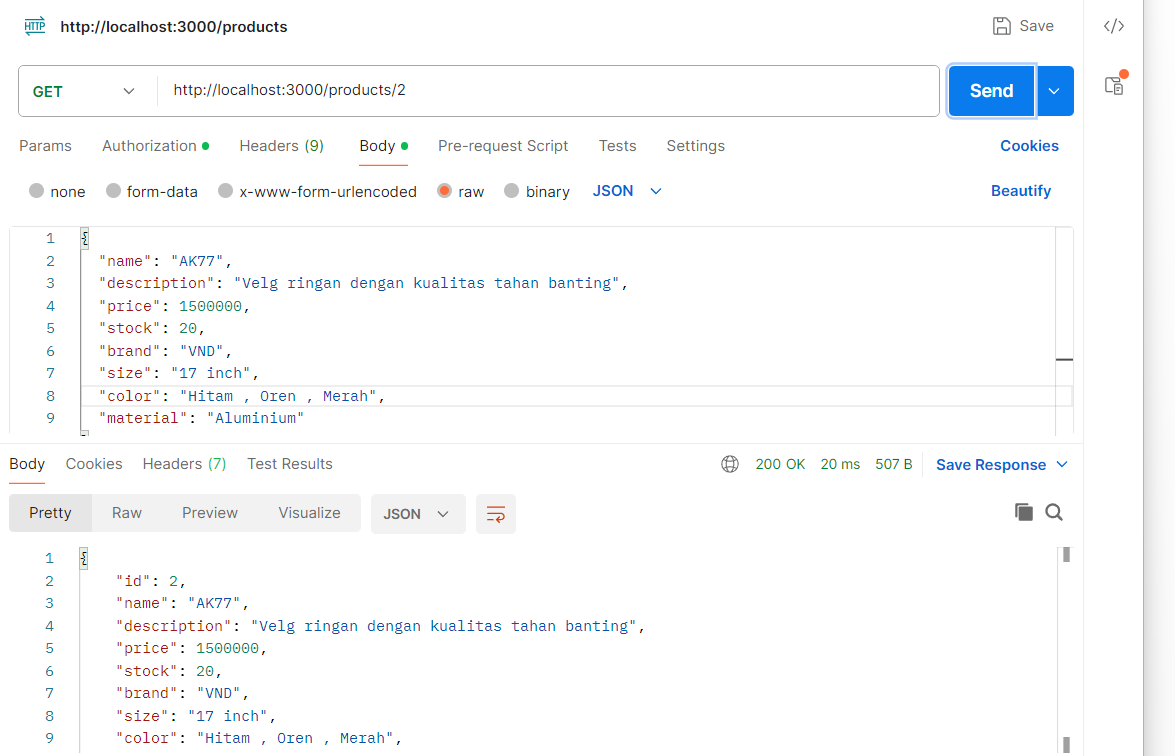


4. GET/ORDERS

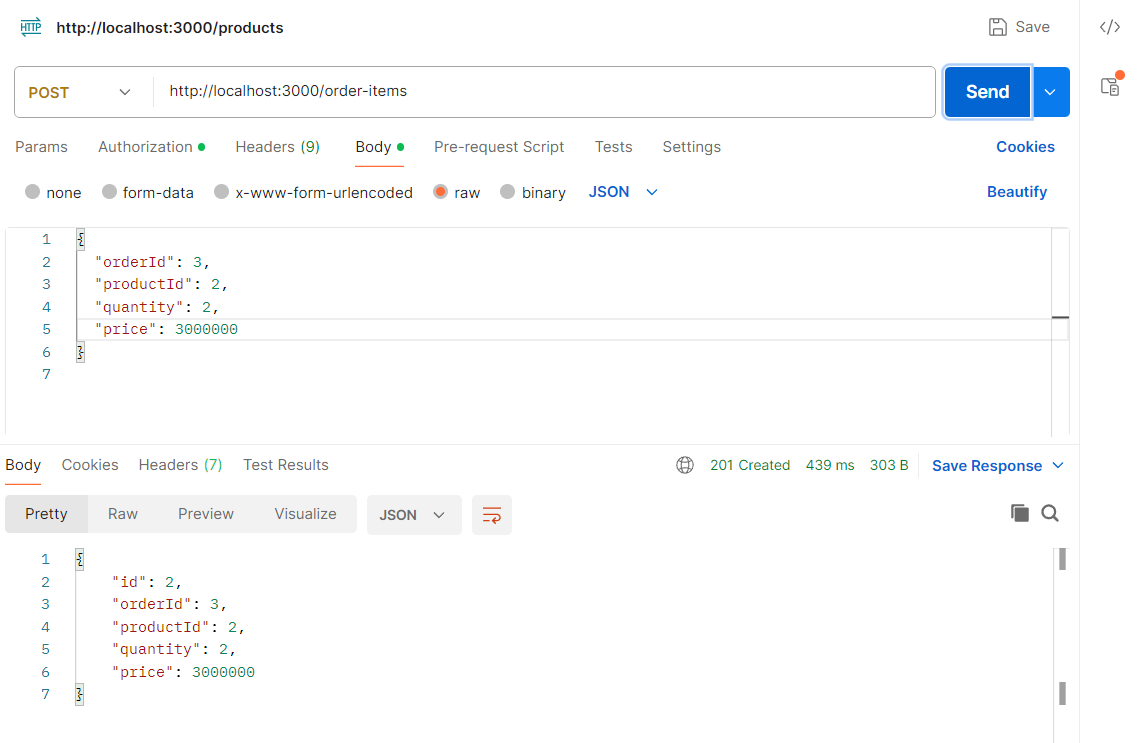


5. GET/PRODUCTS

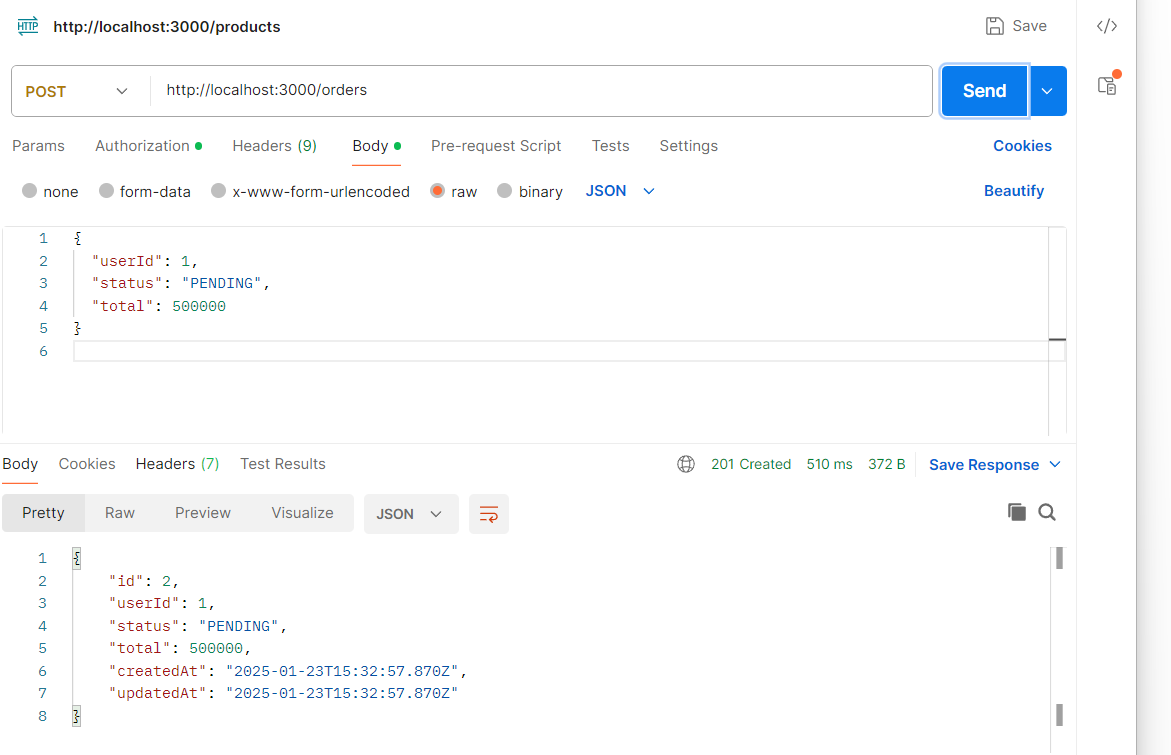
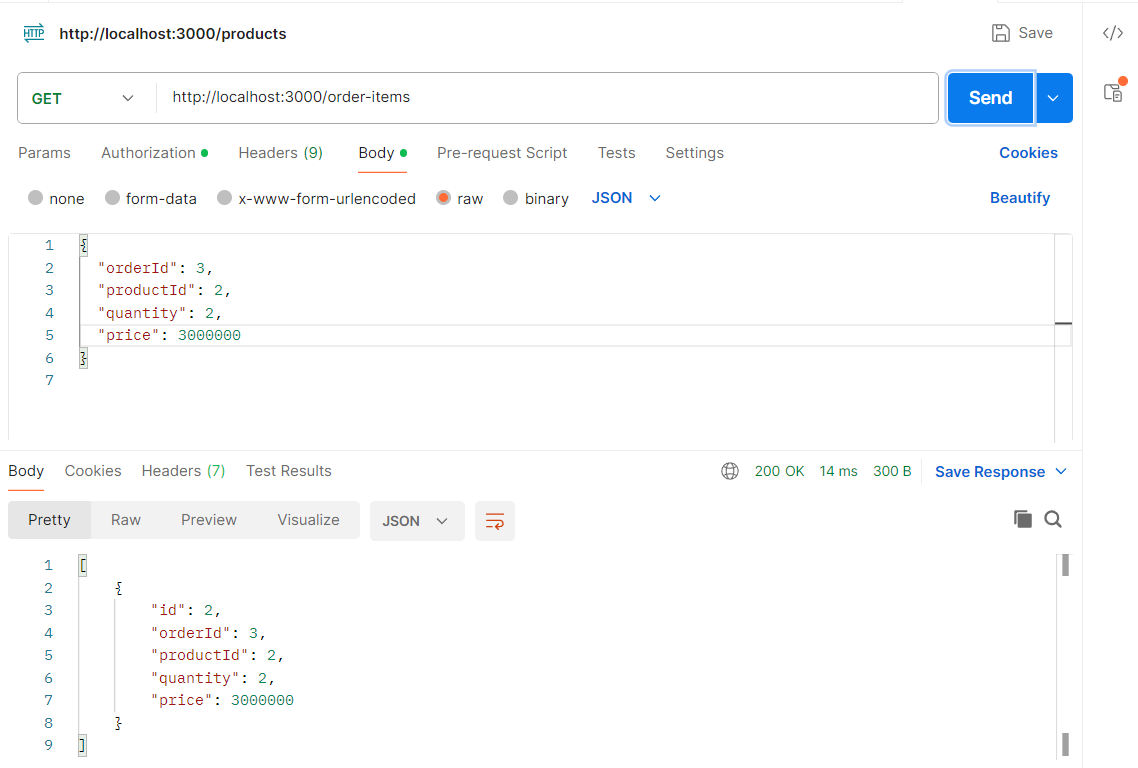


6.GET/PRODUCTS/{PRODUCTSId}

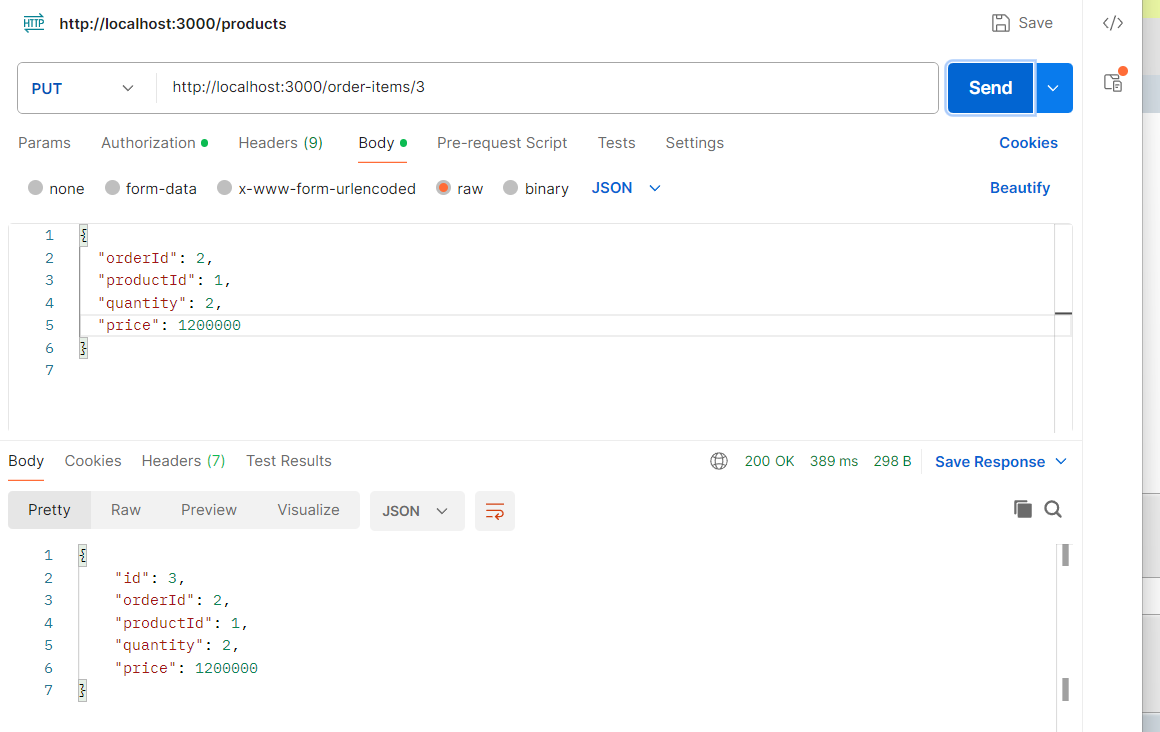
7. POST/ORDER-ITEMS



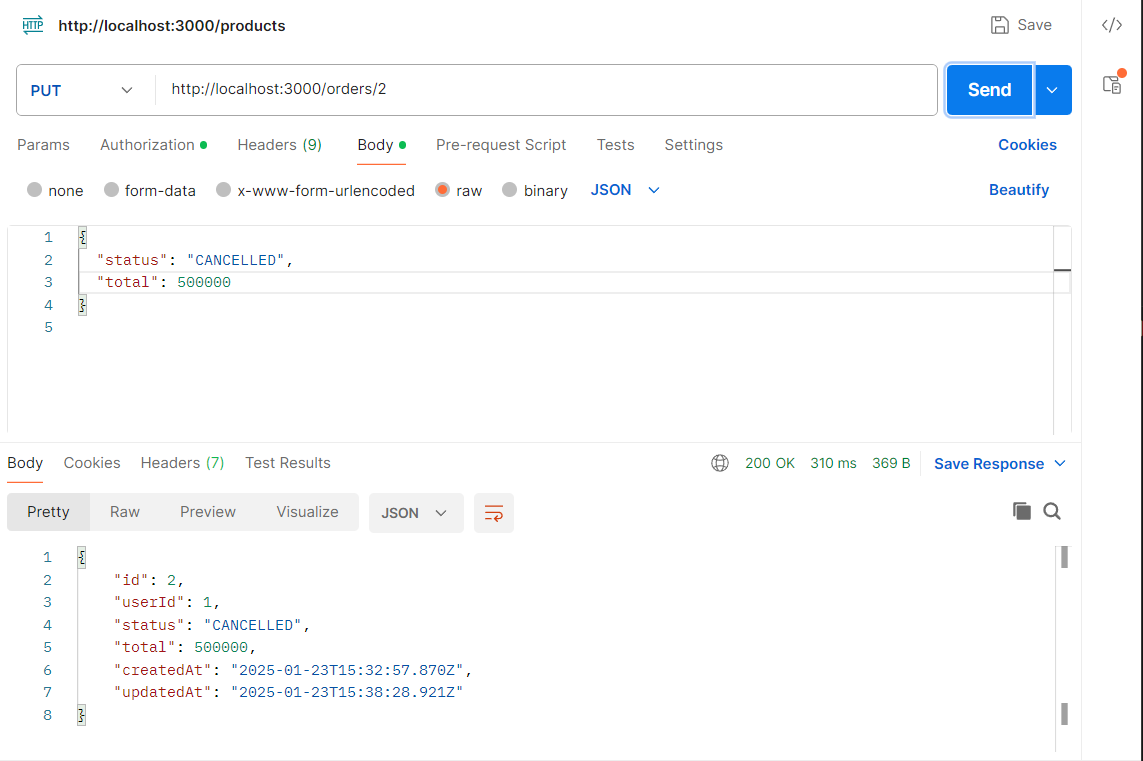
8. POST/ORDERS

9.GET/ORDERS-ITEMS

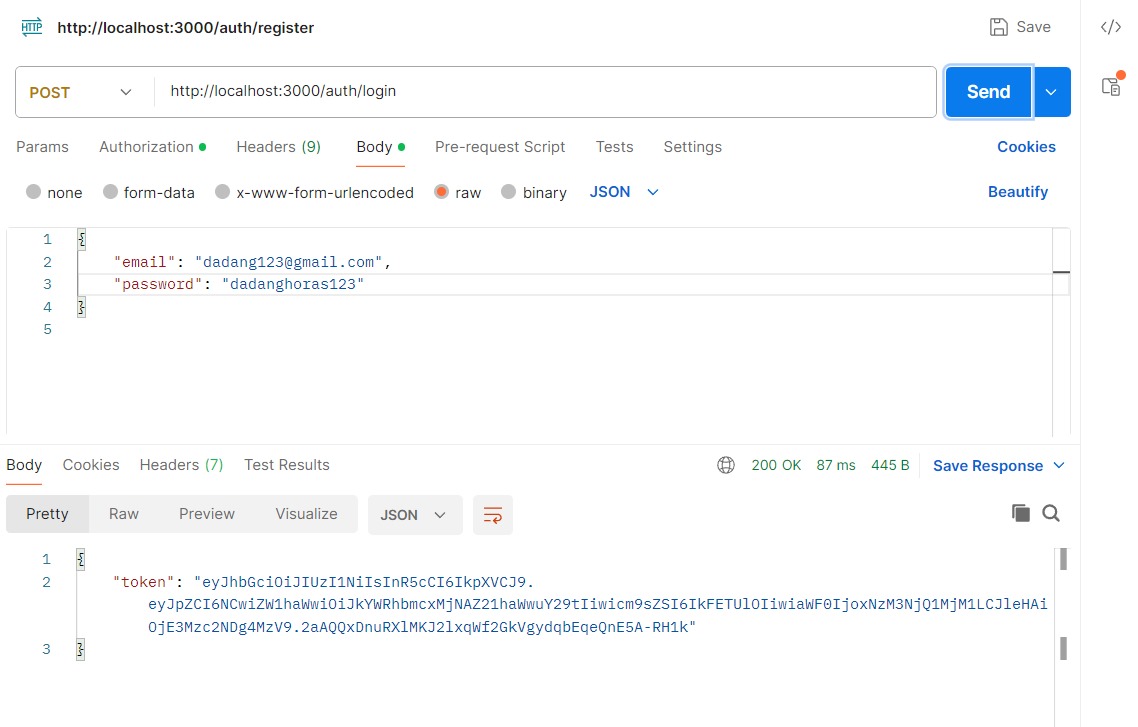
10.PUT/ORDER-ITEMS/{order-itemsId}



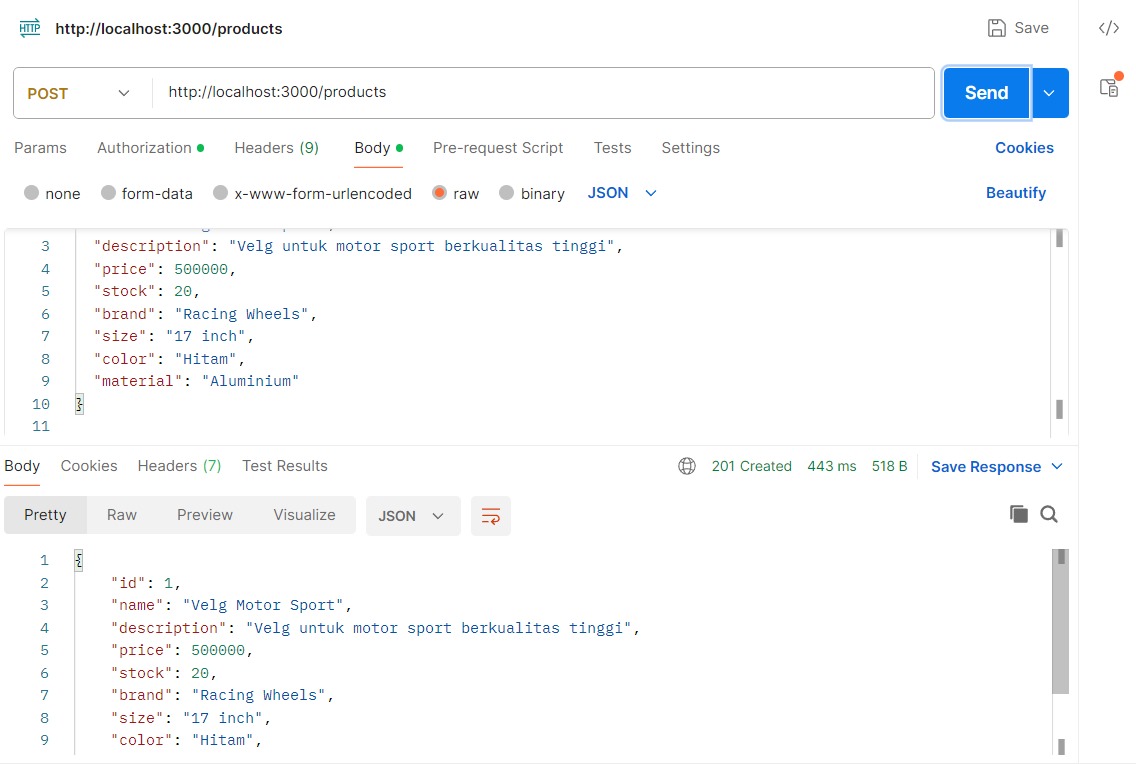
11.PUT/ORDERS/{ordersId}



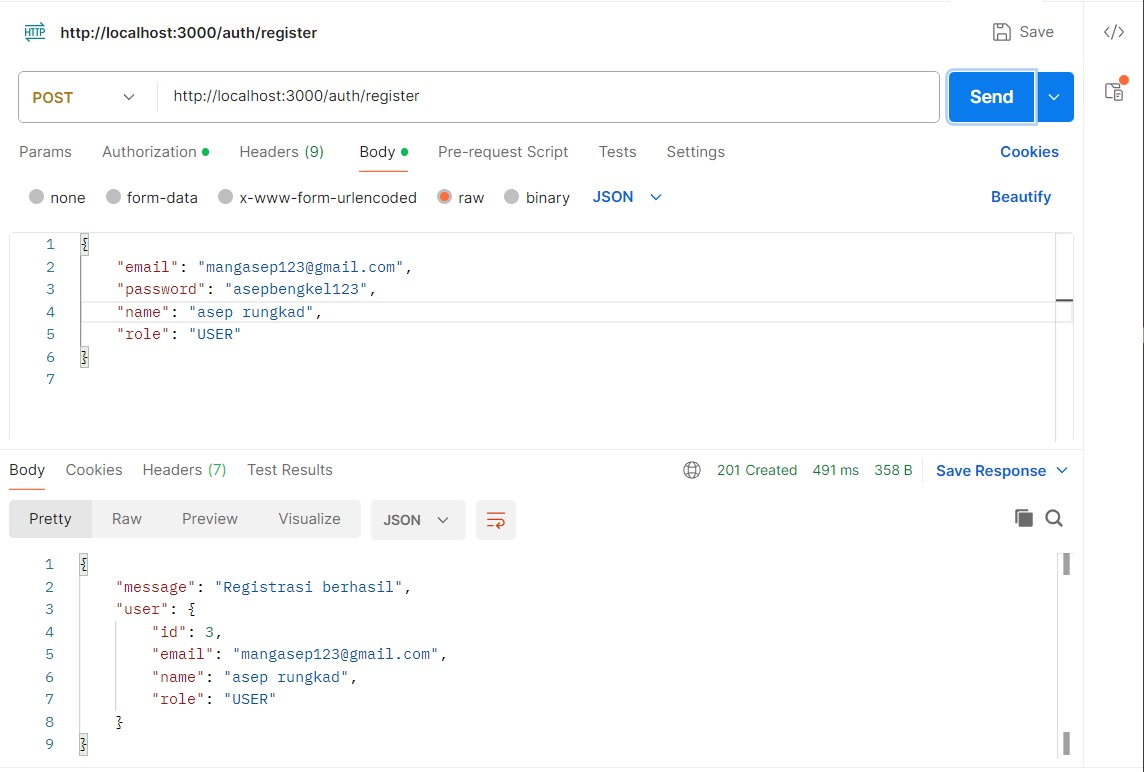
12. POST/auth/login



13. POST/PRODUCTS



14.POST/auth/REGISTER



# BAB IV

# PENUTUP

## 1. Kesimpulan

Dalam penelitian ini, pengembangan web penjualan velg telah berhasil memberikan solusi yang efektif untuk meningkatkan aksesibilitas, kemudahan, dan efisiensi dalam proses penjualan velg. Sistem yang dirancang memungkinkan pelanggan untuk mencari, membandingkan, dan membeli produk dengan mudah melalui fitur-fitur seperti pencarian produk, kategori, dan integrasi pembayaran online. Selain itu, sistem admin mempermudah pengelolaan inventaris dan pesanan, yang berdampak positif pada efisiensi.

Namun demikian, pengembangan lebih lanjut masih diperlukan, seperti integrasi fitur pelacakan pengiriman, personalisasi rekomendasi produk, dan optimisasi performa situs untuk perangkat seluler. Dengan demikian, web penjualan velg ini diharapkan dapat menjadi platform yang berkelanjutan dan berdaya saing tinggi di pasar otomotif.

## 2. Saran

Kedepannya system diharapkan mempunyai rancangan UI/UX dan memiliki fitur pelengkap lainnya.