

**LAPORAN DESAIN DAN PERANCANGAN REST API
MARKETPLACE ONLINE**

Dosen Pengampu : Alun Sujjada, S.Kom, M.T



Disusun Oleh :

Asep Rohmat (20240040122)

Eneng Salva Haniva Syah (20240040296)

Shifa Puteri Nurohman (20240040282)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NUSA PUTRA
2025**

1. Pendahuluan

Perkembangan teknologi digital mendorong perubahan besar dalam proses jual beli yang sebelumnya dilakukan secara konvensional menjadi berbasis online. Salah satu bentuk implementasi dari perkembangan tersebut adalah marketplace online, yaitu sebuah platform yang mempertemukan penjual dan pembeli dalam satu sistem terintegrasi.

Dalam pengembangan sistem marketplace, diperlukan sebuah backend yang mampu mengelola data secara terstruktur, aman, dan dapat diakses oleh berbagai platform seperti website maupun aplikasi mobile. Oleh karena itu, pada laporan ini dilakukan perancangan dan desain REST API Marketplace menggunakan konsep RESTful API.

REST API berfungsi sebagai penghubung antara frontend dengan database, sehingga seluruh proses seperti pengelolaan user, produk, pesanan, pembayaran, hingga integrasi layanan pihak ketiga dapat berjalan dengan baik. Selain itu, penggunaan arsitektur RESTful dipilih karena sifatnya yang *stateless* dan memiliki skalabilitas tinggi, yang memungkinkan sistem untuk menangani lonjakan trafik transaksi dengan lebih stabil. Dengan pemisahan yang jelas antara sisi klien dan server, pengembang dapat melakukan pembaruan fitur pada aplikasi mobile tanpa harus mengganggu kinerja sistem secara keseluruhan.

2. Tema Sistem

Tema yang diangkat pada perancangan REST API ini adalah Marketplace Online. Sistem marketplace ini dirancang sebagai platform multifungsi yang memfasilitasi interaksi ekonomi digital secara *end-to-end*. Secara spesifik, sistem ini memungkinkan:

- Manajemen Akun: Pengguna dapat melakukan registrasi dan login secara aman untuk mengakses fitur-fitur platform.
- Manajemen Produk: Penjual memiliki kendali penuh untuk menambahkan, menyunting, dan mengelola inventaris produk mereka secara *real-time*.
- Proses Pemesanan: Pembeli dapat melakukan pencarian, pemilihan, hingga pemesanan produk secara praktis melalui antarmuka yang responsif.
- Proses Pemesanan: Pembeli dapat melakukan pencarian, pemilihan, hingga pemesanan produk secara praktis melalui antarmuka yang responsif.
- Transparansi Status: Pengguna dapat melihat dan melacak status pesanan secara akurat, mulai dari tahap pembayaran hingga pengiriman barang.

Tema marketplace dipilih karena relevan dengan kebutuhan industri saat ini dan sangat ideal untuk menerapkan konsep REST API yang kompleks. Selain itu, tema ini memungkinkan implementasi berbagai teknologi modern, seperti penggunaan database relasional untuk menjaga integritas data, penerapan autentikasi JWT (JSON Web Token) untuk keamanan akses, serta integrasi API publik (seperti kurir pengiriman atau *payment gateway*) untuk memperkaya fungsionalitas aplikasi.

3. Perancangan Database

Perancangan database dalam sistem marketplace ini merupakan fondasi utama untuk memastikan bahwa seluruh data transaksi, inventaris, dan informasi pengguna dapat dikelola secara terstruktur, konsisten, dan memiliki integritas yang tinggi. Database yang dirancang menggunakan model relasional yang bertujuan untuk meminimalkan redundansi data melalui proses normalisasi, sehingga sistem tetap responsif meski menangani volume data yang besar.

The screenshot shows the MySQL Workbench interface. At the top, it displays 'Server 127.0.0.1 > Database: marketplace_db'. Below this is a 'Filters' section with a search input field labeled 'Containing the word:'. The main area shows a table of tables with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The tables listed are categories, orders, order_items, products, and users. A summary at the bottom indicates 5 tables, 10 rows, InnoDB storage engine, utf8mb4_general_ci collation, and a total size of 192.0 Kib. Below the table is a 'Create new table' dialog box with fields for 'Table name' (empty) and 'Number of columns' (set to 4), and a 'Create' button.

3.1 Arsitektur Data dan Entity Relationship Diagram (ERD)

ERD mendefinisikan hubungan logis antar entitas yang terlibat dalam ekosistem marketplace. Perancangan ini menjamin bahwa setiap aksi yang dilakukan pengguna seperti pendaftaran, penambahan produk, hingga pembayaran terekam dengan akurasi yang tepat dalam tabel-tabel yang saling terhubung.

Entitas utama dalam sistem ini meliputi:

- Users: Berfungsi menyimpan informasi identitas seperti kredensial login, profil pengguna, dan peran (*role*) sebagai penjual atau pembeli.
- Products: Menyimpan katalog barang secara mendetail, yang memungkinkan penjual mengelola deskripsi, harga, dan jumlah stok secara mandiri.
- Orders & Order Items: Mencatat setiap aktivitas pembelian, detail item yang dibeli, serta status pemrosesan pesanan secara *real-time*.

- Payments: Mengintegrasikan data transaksi dengan status pembayaran untuk mendukung transparansi finansial antara penjual dan pembeli.

3.2 Kamus Data dan Skema Tabel

Untuk mendukung integrasi REST API yang efisien, setiap tabel dirancang dengan tipe data yang optimal guna mempercepat proses kueri (*querying*) dan pertukaran format data JSON.

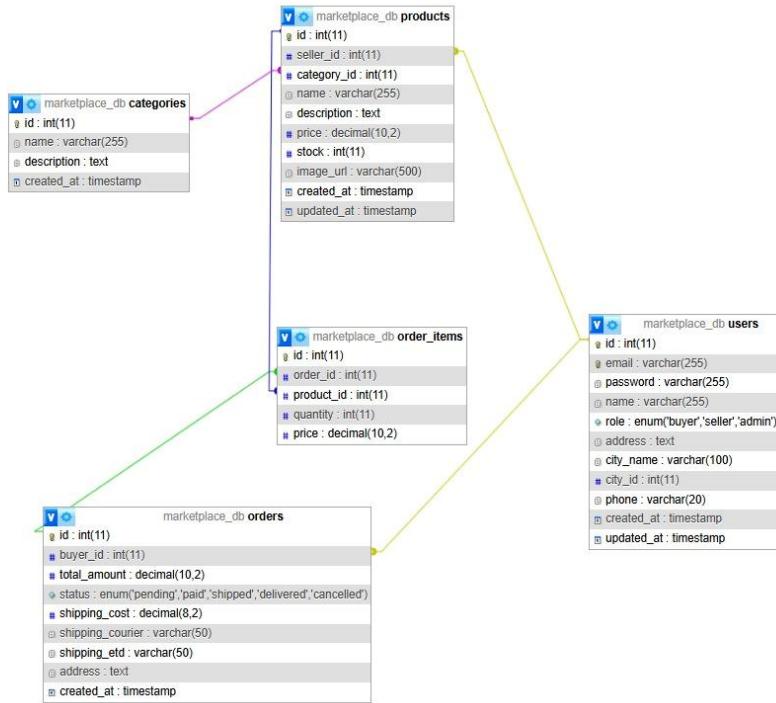
| Nama Tabel | Fungsi Utama | Atribut kunci |
|-------------|------------------------------------|--|
| User | Autentikasi dan identitas | user_id (PK), username, email, password_hash, role_type, created_at |
| Categories | Klasifikasi produk untuk pencarian | category_id (PK), category_name, slug |
| Products | Manajemen inventaris penjual | product_id (PK), user_id (FK), category_id (FK), name, price, stock_quantity |
| Orders | Header transaksi pemesanan | order_id (PK), user_id (FK), total_amount, order_status, order_date |
| Order_Items | Detail produk dalam satu transaksi | item_id (PK), order_id (FK), product_id (FK), quantity, subtotal |

3.3 Relasi dan Integritas Data

Sistem ini mengandalkan hubungan antar-tabel (relasi) untuk menjaga konsistensi informasi:

- One-to-Many (Users to Products): Menggambarkan bahwa satu pengguna dengan peran penjual dapat memiliki banyak daftar produk dalam katalognya.
- One-to-Many (Users to Orders): Menggambarkan satu pembeli yang dapat melakukan riwayat pesanan berkali-kali di dalam platform.
- Many-to-Many (Orders to Products): Diimplementasikan melalui tabel perantara Order_Items, yang memungkinkan satu pesanan berisi berbagai

jenis produk, dan satu produk dapat dipesan oleh banyak pengguna yang berbeda.



3.4 Keamanan Pada Level Database

Sesuai dengan kebutuhan akan sistem yang aman, database ini tidak menyimpan informasi sensitif dalam bentuk teks asli (*plain text*). Kata sandi pengguna diproses menggunakan teknik *hashing* sebelum disimpan. Selain itu, penggunaan *Foreign Key* (FK) memastikan integritas referensial; misalnya, sebuah pesanan tidak dapat dibuat jika data produk atau data pengguna tidak ditemukan dalam database, sehingga meminimalisir kesalahan data sistem.

4. Perancangan Endpoint API

Perancangan *endpoint API* merupakan tahap penentuan jalur (*path*) dan metode komunikasi yang digunakan oleh *frontend* untuk berinteraksi dengan server. Setiap *endpoint* dirancang mengikuti prinsip RESTful agar mudah dipahami, konsisten, dan efisien dalam pertukaran data menggunakan format JSON.

4.1 Mekanisme Komunikasi API

Seluruh permintaan (*request*) yang masuk akan diproses melalui beberapa tahapan utama:

- **Endpoint URL:** Alamat unik yang dituju untuk mengakses sumber daya tertentu.

- HTTP Methods: Penggunaan metode seperti GET (mengambil data), POST (mengirim data baru), PUT (memperbarui data), dan DELETE (menghapus data).
- Header & Auth: Menyertakan *Authorization Token* berbasis JWT untuk memastikan keamanan akses pada fitur-fitur sensitif.
- Response Code: Server akan memberikan status seperti 200 OK untuk keberhasilan atau 401 Unauthorized untuk kegagalan akses.

4.2 Daftar Endpoint Utama

Pada proyek ini, backend aplikasi dikembangkan menggunakan arsitektur **RESTful API** yang berfungsi sebagai penghubung antara frontend dan database. Secara keseluruhan, sistem ini memiliki **23 endpoint API** yang digunakan untuk menangani berbagai kebutuhan fungsional aplikasi.

Endpoint-endpoint tersebut mencakup proses autentikasi pengguna (registrasi dan login), pengelolaan data pengguna, pengelolaan produk dan kategori, pengelolaan pesanan, serta perhitungan biaya pengiriman. Setiap endpoint dirancang untuk mendukung operasi Create, Read, Update, dan Delete (CRUD) sesuai dengan kebutuhan sistem.

Keamanan akses endpoint diterapkan menggunakan JSON Web Token (JWT), sehingga hanya pengguna yang telah terautentikasi yang dapat mengakses layanan tertentu. Selain itu, sistem juga menerapkan pembatasan hak akses berdasarkan peran pengguna untuk memastikan setiap pengguna hanya dapat menggunakan fitur yang sesuai.

Dengan adanya total **30 endpoint API**, sistem diharapkan mampu mendukung seluruh proses bisnis aplikasi marketplace secara terstruktur, aman, dan efisien.

| No | Module | Method | Endpoint |
|----|----------|--------|-----------------------|
| 1 | AUTH | POST | /api/auth/register |
| 2 | AUTH | POST | /api/auth/login |
| 3 | USERS | GET | /api/users |
| 4 | USERS | GET | /api/users/:id |
| 5 | USERS | PUT | /api/users/:id |
| 6 | USERS | GET | /api/users/me/profile |
| 7 | USERS | PUT | /api/users/me/profile |
| 8 | PRODUCTS | GET | /api/products |
| 9 | PRODUCTS | GET | /api/products/:id |
| 10 | PRODUCTS | POST | /api/products |
| 11 | PRODUCTS | PUT | /api/products/:id |
| 12 | PRODUCTS | DELETE | /api/products/:id |

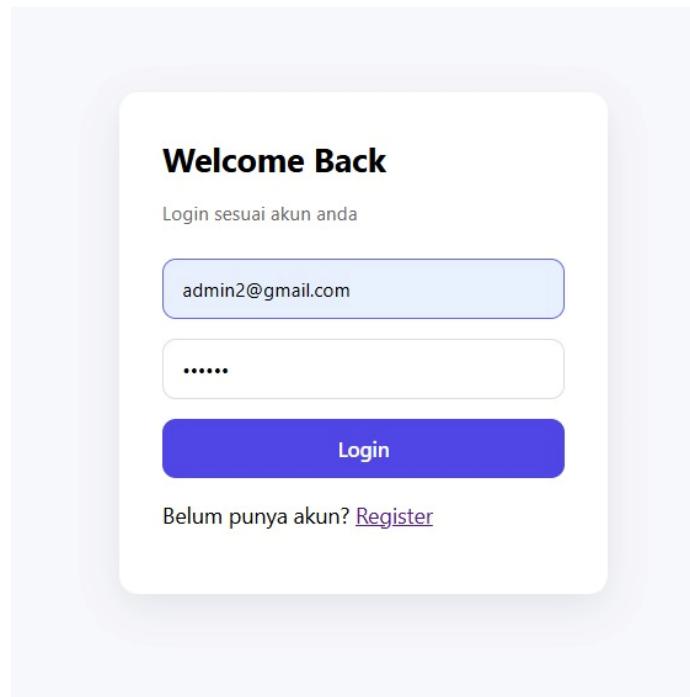
| | | | |
|----|------------|--------|------------------------|
| 13 | CATEGORIES | GET | /api/categories |
| 14 | CATEGORIES | GET | /api/categories/:id |
| 15 | CATEGORIES | POST | /api/categories |
| 16 | CATEGORIES | PUT | /api/categories/:id |
| 17 | CATEGORIES | DELETE | /api/categories/:id |
| 18 | ORDERS | POST | /api/orders |
| 19 | ORDERS | GET | /api/orders |
| 20 | ORDERS | GET | /api/orders/my |
| 21 | ORDERS | GET | /api/orders/:id |
| 22 | ORDERS | PUT | /api/orders/:id/status |
| 23 | SHIPPING | GET | /api/shipping/cost |

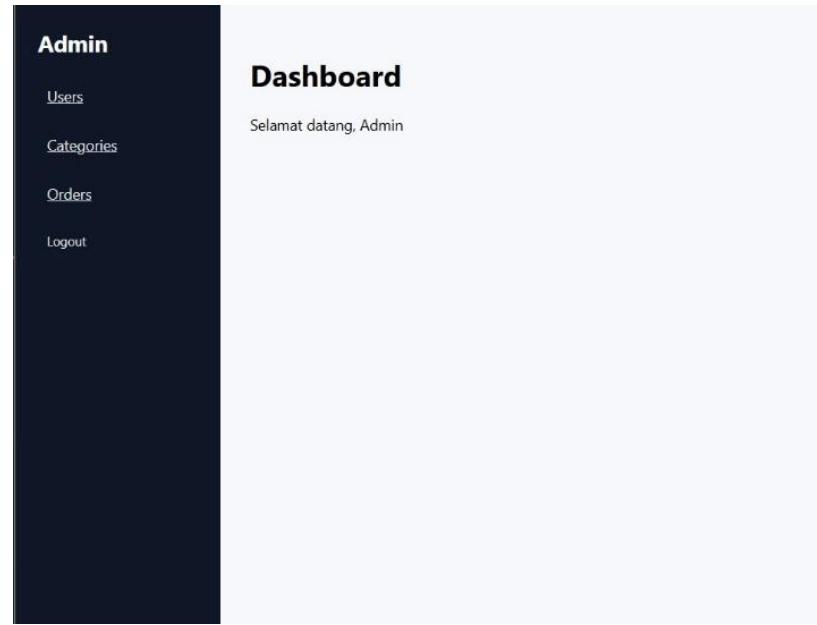
Berdasarkan kebutuhan sistem marketplace yang telah dirancang, berikut adalah daftar *endpoint* yang dikelompokkan berdasarkan fungsinya:

- Otentikasi dan Manajemen Admin

Kelompok ini bertujuan untuk menangani akses masuk dan pendaftaran pengguna ke dalam sistem secara aman

- POST /api/auth/login : Untuk masuk (login) ke sistem dan mendapatkan token (JWT) sebagai bukti autentikasi.





- GET /api/users Untuk melihat daftar semua user yang terdaftar di aplikasi (khusus admin).

The screenshot shows the 'Users Management' section of the admin interface. It features a table listing users with columns for ID, Name, Email, Role, and Action (Aksi). Below the table is a form titled 'Hapus User' with fields for 'ID user' and a red 'Delete User' button.

| ID | Nama | Email | Role | Aksi |
|----|------------|---------------------|--------|-------------------------|
| 9 | Seller Pro | sell8e@market.com | seller | <button>Detail</button> |
| 11 | Admin Baru | admin2@gmail.com | admin | <button>Detail</button> |
| 12 | asep | buyerasep@gmail.com | buyer | <button>Detail</button> |
| 14 | sifa hadju | asifa@gmail.com | buyer | <button>Detail</button> |

Hapus User

- GET /api/users/:id untuk melihat detail data satu user tertentu berdasarkan ID

Admin

Back

Logout

User Detail

Name

Address

Phone

Save

- PUT /api/users/:id untuk masuk ke user detail jadi menampil detail user

Admin

Back

Logout

User Detail

Name

Address

Phone

Save

- GET /api/categories untuk menampilkan halaman kategori management

The screenshot shows a dark-themed administrative interface. On the left, a sidebar has 'Admin' at the top, followed by 'Dashboard' and 'Logout'. The main area has a title 'Categories Management' and a blue button '+ Tambah Category'. Below is a table with columns 'ID', 'Name', 'Description', and 'Aksi'. The data rows are:

| ID | Name | Description | Aksi |
|----|-------------|-----------------------|--|
| 3 | Books | Buku dan literatur | <button>Edit</button> <button>Hapus</button> |
| 1 | Electronics | Gadget dan elektronik | <button>Edit</button> <button>Hapus</button> |
| 4 | Electronics | Gadget & Smartphone | <button>Edit</button> <button>Hapus</button> |
| 2 | Fashion | Pakaian dan aksesoris | <button>Edit</button> <button>Hapus</button> |
| 8 | Olahraga | sepatu | <button>Edit</button> <button>Hapus</button> |

Below the table is a section titled 'Tambah Category'.

- POST /api/categories untuk menambah kategori

The screenshot shows the same dark-themed administrative interface. The 'Categories Management' page includes a blue '+ Tambah Category' button. Below it is a table with the same structure as the first screenshot. At the bottom is a 'Tambah Category' form with fields for 'Name' and 'Description', and a 'Save' button.

- **DELETE /api/users/:id** menghapus user berdasarkan id

Admin

[Dashboard](#)

[Logout](#)

Users Management

| ID | Nama | Email | Role | Aksi |
|----|------------|---------------------|--------|------------------------|
| 9 | Seller Pro | sell8e@market.com | seller | Detail |
| 11 | Admin Baru | admin2@gmail.com | admin | Detail |
| 12 | asep | buyerasep@gmail.com | buyer | Detail |
| 14 | sifa hadju | asifa@gmail.com | buyer | Detail |

Hapus User

9 [Delete User](#)

- **PUT /api/categories/:id** untuk mengedit berdsarkan id

Admin

[Dashboard](#)

[Logout](#)

Categories Management

[+ Tambah Category](#)

| ID | Name | Description | Aksi |
|----|-------------|-----------------------|--|
| 3 | Books | Buku dan literatur | Edit Hapus |
| 1 | Electronics | Gadget dan elektronik | Edit Hapus |
| 4 | Electronics | Gadget & Smartphone | Edit Hapus |
| 2 | Fashion | Pakaian dan aksesoris | Edit Hapus |
| 8 | Olahraga | sepatu | Edit Hapus |

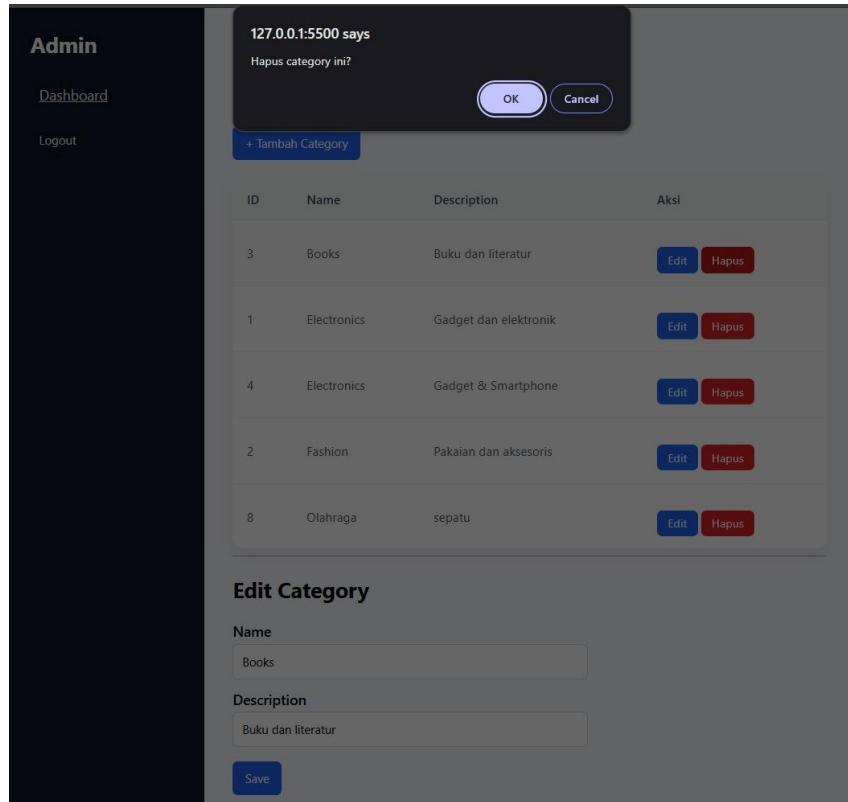
Edit Category

Name
Books

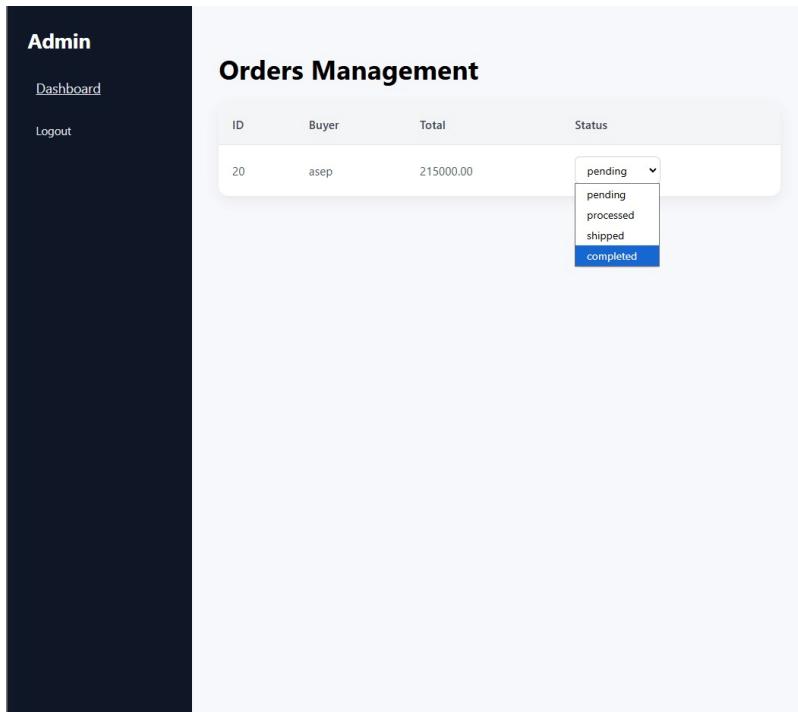
Description
Buku dan literatur

[Save](#)

- **DELETE /api/categories/:id** menghapus kategori



- **PUT /api/orders/:id/status** halaman ordesr semua buyer untuk metubah status pengiriman



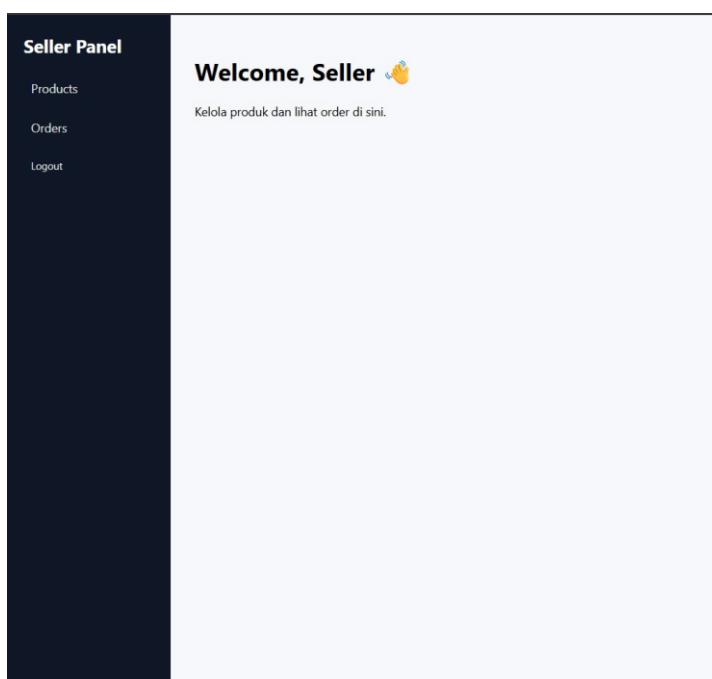
B. Otentikasi dan Manajemen Seller

Kelompok ini bertujuan untuk menangani proses pendaftaran dan autentikasi **seller** agar dapat mengakses sistem secara aman. Melalui fitur ini, seller dapat melakukan login ke dalam aplikasi dan menggunakan layanan yang disediakan untuk mengelola produk serta memantau pesanan yang masuk.

- POST /api/auth/register untuk membuat akun seller mengisi nama email pasaword dan memilih role karna ini seller rolenya pilih seller

The screenshot shows a registration form titled "Register Account". It contains four input fields: "seller" (with placeholder "asepseller"), "asepseller@gmail.com" (with placeholder "asepseller@gmail.com"), "....." (with placeholder "....."), and a dropdown menu set to "Seller". Below the form is a blue "Register" button. At the bottom left, there is a link "Sudah punya akun? Login".

- POST /api/auth/login untuk login page dan akan masuk ke dashboard seller



- POST /api/products menampilkan product management

The screenshot shows a dark-themed user interface for a seller panel. On the left, a sidebar titled "Seller Panel" contains links for "Dashboard" and "Logout". The main content area is titled "Products Management" and features a table with one row of data. The table columns are labeled "ID", "Name", "Price", "Stock", "Category", and "Aksi". The single row displays ID 2, Name "Mouse", Price 200000.00, Stock 50, Category "Electronics", and buttons for "Edit" and "Delete". A blue button labeled "+ Tambah Produk" is located at the top left of the table.

| ID | Name | Price | Stock | Category | Aksi |
|----|-------|-----------|-------|-------------|---|
| 2 | Mouse | 200000.00 | 50 | Electronics | <button>Edit</button> <button>Delete</button> |

- POST /api/products untuk menambah produk harus mengisi nama deskripsi harga stock lategori dan image url

The screenshot shows the same dark-themed user interface for the seller panel. The "Products Management" page now includes a form for adding a new product. The form fields are: "Name" (input field), "Description" (input field), "Price" (input field), "Stock" (input field), "Category" (dropdown menu with "Books" selected), and "Image URL" (input field). Below the form is a blue "Save" button.

| ID | Name | Price | Stock | Category | Aksi |
|----|-------|-----------|-------|-------------|---|
| 2 | Mouse | 200000.00 | 50 | Electronics | <button>Edit</button> <button>Delete</button> |

Name

Description

Price

Stock

Category
 Books

Image URL

Save

- PUT /api/products/:id untuk mengedit produk berdasarkan id

Seller Panel

Products Management

| ID | Name | Price | Stock | Category | Aksi |
|----|-------|-----------|-------|-------------|---|
| 2 | Mouse | 200000,00 | 50 | Electronics | <button>Edit</button> <button>Delete</button> |

Name

Description

Price

Stock

Category

Image URL

Save

- DELETE /api/products/:id untuk menghapus produk berdasarkan id

Seller Panel

127.0.0.1:5500 says
Hapus produk?

OK **Cancel**

| ID | Name | Price | Stock | Category | Aksi |
|----|-------|-----------|-------|-------------|---|
| 2 | Mouse | 200000,00 | 50 | Electronics | <button>Edit</button> <button>Delete</button> |

Name

Description

Price

Stock

Category

Image URL

Save

- GET /api/categories untuk melihat pilihan kategori untuk produk

| ID | Name | Price | Stock | Category | Aksi |
|----|-------|-----------|-------|-------------|---|
| 2 | Mouse | 200000.00 | 50 | Electronics | <button>Edit</button> <button>Delete</button> |

- GET /api/categories/:id untuk melihat pilihan kategori untuk produk berdasarkan id

| ID | Name | Price | Stock | Category | Aksi |
|----|-------|-----------|-------|-------------|---|
| 2 | Mouse | 200000.00 | 50 | Electronics | <button>Edit</button> <button>Delete</button> |

- GET /api/orders untuk menampilkan buyer yang memesan

The screenshot shows a dark-themed user interface for a 'Seller Panel'. On the left, a vertical sidebar has a dark blue header with the text 'Seller Panel' in white. Below this, there are two menu items: 'Dashboard' and 'Logout', both in a smaller white font. The main content area has a light gray background. At the top of this area, the title 'Orders Masuk' is displayed in bold black text. Below the title is a table with a thin gray border. The table has columns labeled 'ID', 'Buyer', 'Total', 'Status', and 'Tanggal'. There is one data row visible, which includes the values: ID 20, Buyer asep, Total 215000.00, Status pending, and Tanggal 1/31/2026, 9:50:34 PM.

| ID | Buyer | Total | Status | Tanggal |
|----|-------|-----------|---------|-----------------------|
| 20 | asep | 215000.00 | pending | 1/31/2026, 9:50:34 PM |

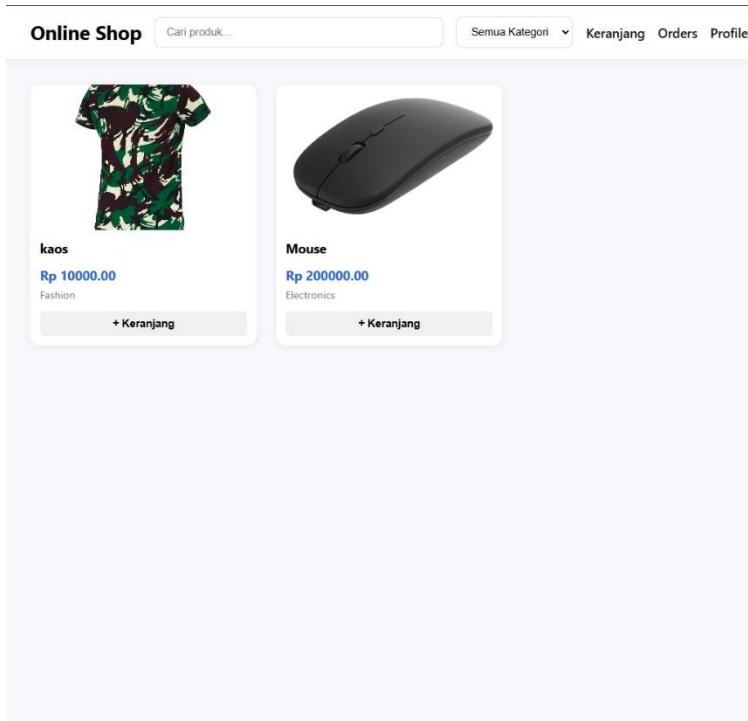
C. Otentikasi dan Manajemen Buyer

Kelompok ini bertujuan untuk menangani proses pendaftaran dan autentikasi **buyer** agar dapat mengakses sistem secara aman. Melalui fitur ini, buyer dapat melakukan login ke dalam aplikasi dan menggunakan layanan yang disediakan untuk melihat produk, melakukan pemesanan, serta mengelola data profil dan riwayat pesanan.

- POST /api/auth/register untuk membuat akun buyer/pembeli

The screenshot shows a registration form titled "Register Account". It contains four input fields: a first name field with "salva hanipa", an email field with "salvaaaa@gmail.com", a password field with ".....", and a dropdown for user type with "Buyer" selected. Below the form is a blue "Register" button and a link "Sudah punya akun? [Login](#)".

- POST /api/auth/login masuk ke dasboard buyer



- GET /api/users/me/profile halaman profile

The screenshot shows the Seller Panel dashboard. On the left, there is a dark sidebar with the title "Seller Panel" at the top, followed by "Dashboard" and "Logout" options. The main content area is titled "Orders Masuk". It contains a table with the following columns: ID, Buyer, Total, Status, and Tanggal. There is one row of data: ID 20, Buyer asep, Total 215000.00, Status pending, and Tanggal 1/31/2026, 9:50:34 PM.

| ID | Buyer | Total | Status | Tanggal |
|----|-------|-----------|---------|-----------------------|
| 20 | asep | 215000.00 | pending | 1/31/2026, 9:50:34 PM |

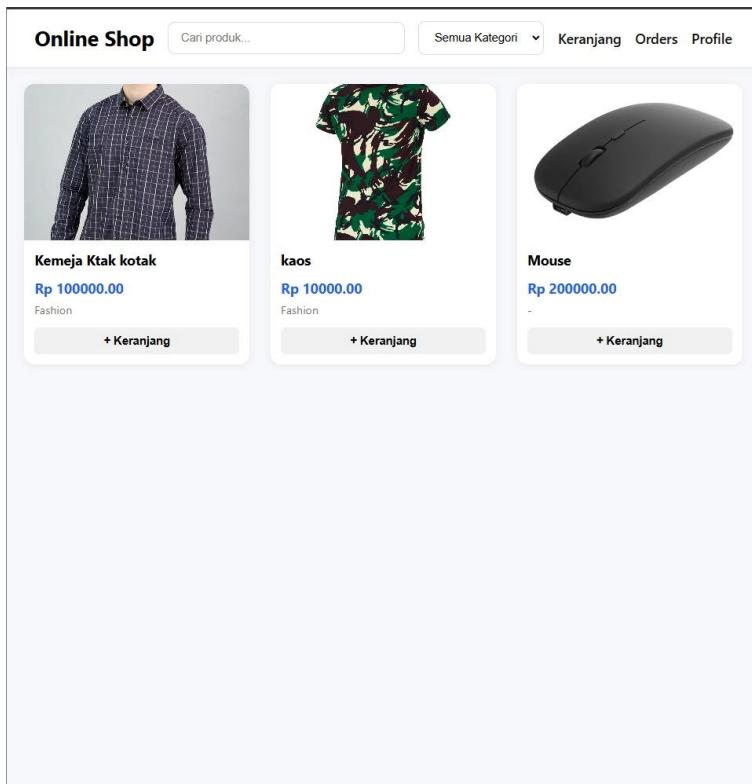
- PUT /api/users/me/profile untuk mengedit profile

The screenshot shows the "Edit Profile" form. At the top left is a "Kembali" button. In the center is a large circular placeholder for a profile photo, labeled "Foto Profil" below it. The form itself has the following fields:

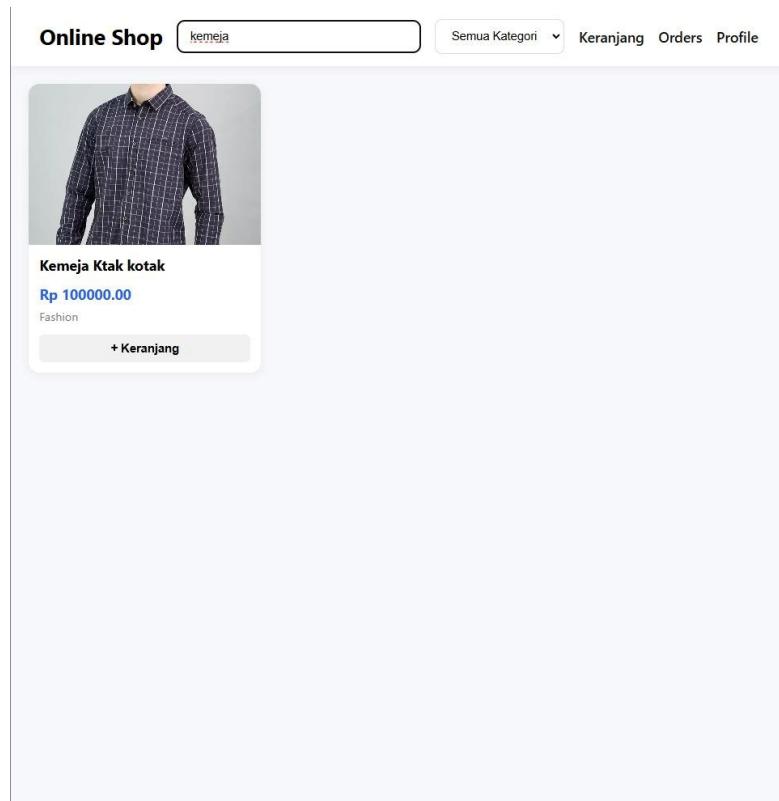
- Nama:** salva hanipa
- Alamat:** palabuan
- No HP:** 02154

A large blue "Simpan Perubahan" button is located at the bottom right of the form.

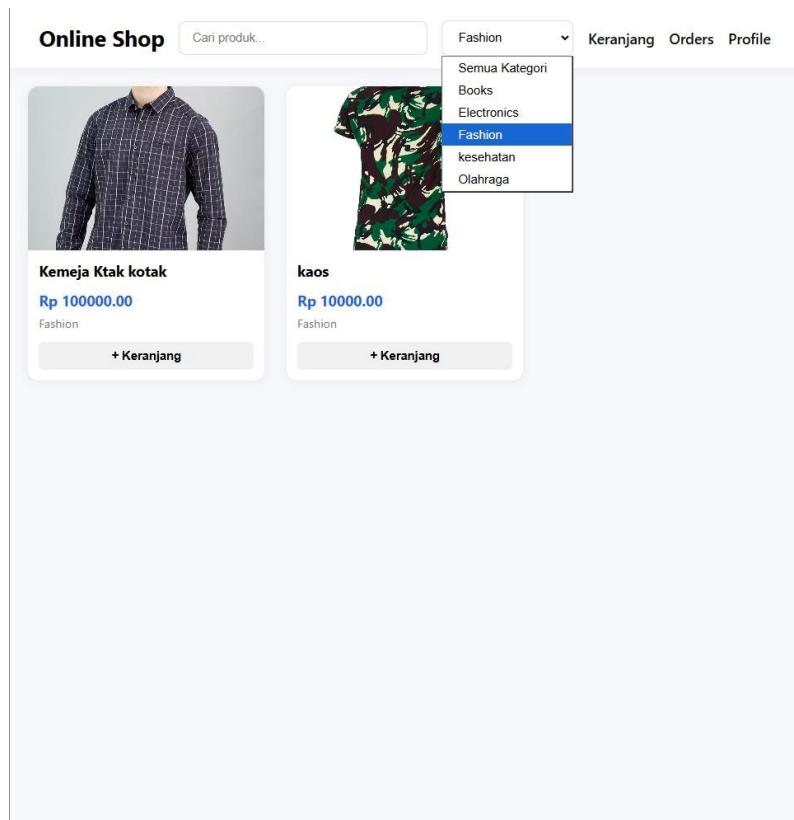
- GET /api/products untuk melihat semua produk



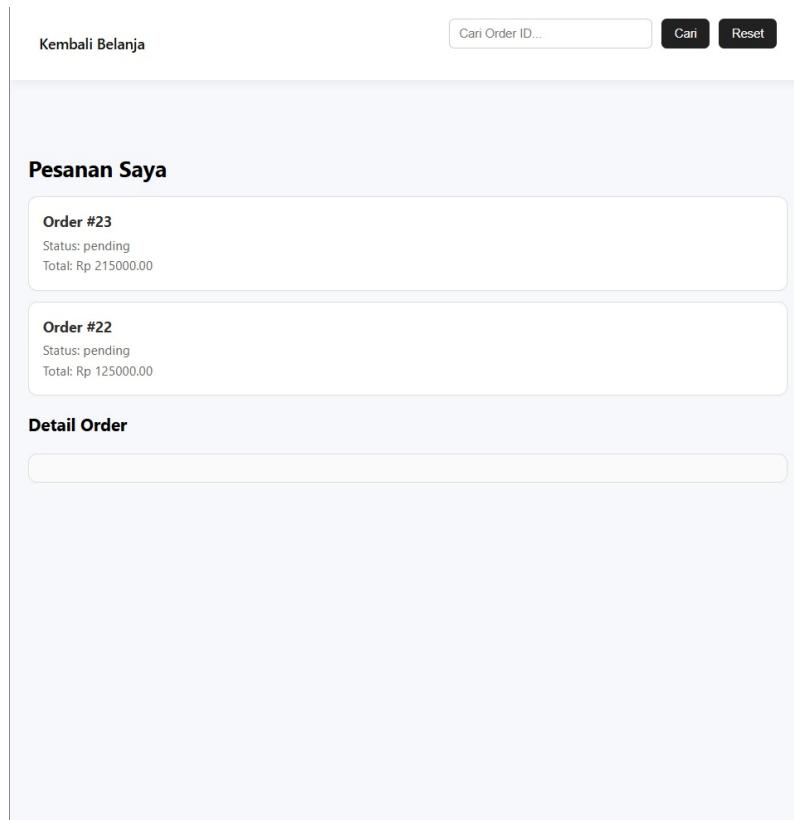
- GET /api/products/:id searching bar masukan nama di konvert ke id



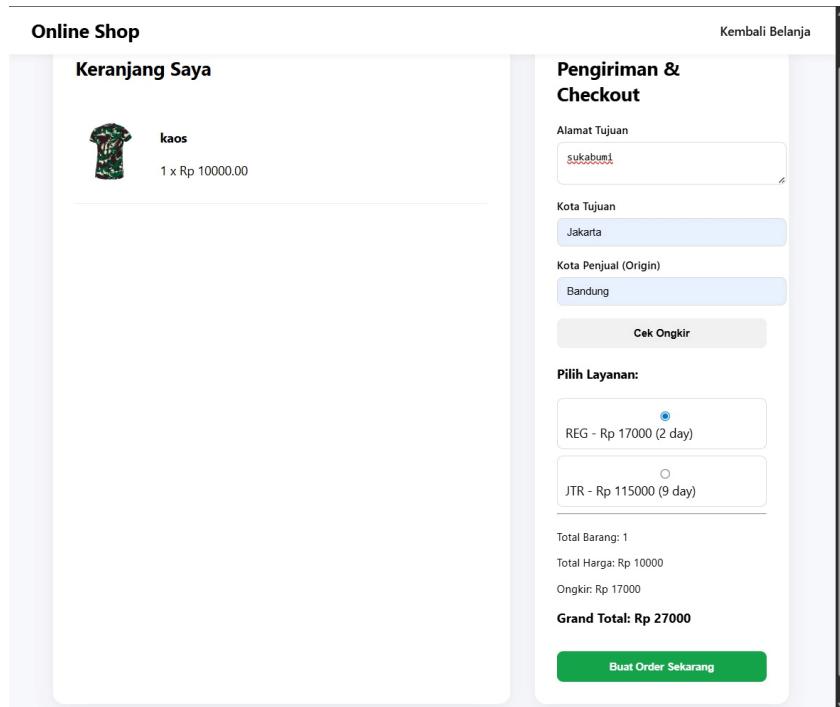
- GET /api/categories/:id untuk melihat categori berdasarkan id



- GET /api/orders/my untuk menampilkan order sediri



- GET /api/shipping/cost untuk menghitung cost menggunakan api publik raja ongkir di hitung berdarkan wilayah antara seller dan buyer



- Transaksi dan Pemesanan (Orders)

Kelompok ini mengatur alur belanja dari mulai pemilihan barang hingga pemrosesan transaksi.

- POST /api/orders: Membuat pesanan baru setelah pembeli memilih produk.
- GET /api/orders: Melihat riwayat transaksi yang pernah dilakukan oleh pembeli.
- GET /api/orders/{id}: Melihat rincian detail dan status pelacakan dari satu pesanan tertentu.

- Integrasi Layanan Pihak Ketiga (Integrations)

API ini berfungsi menghubungkan marketplace dengan penyedia layanan eksternal untuk memperkaya fitur sistem.

- POST /api/payments/process: Mengirim data transaksi ke *payment gateway* untuk diverifikasi.
- GET /api/shipping/cost: Menghitung biaya pengiriman berdasarkan integrasi dengan API kurir logistik.

4.3 Standarisasi Keamanan Endpoint

Setiap *endpoint* yang melibatkan data pribadi atau transaksi finansial wajib melewati lapisan keamanan JWT. Dengan arsitektur ini, server tidak perlu menyimpan *session* (*stateless*), sehingga performa sistem tetap terjaga meski menangani banyak permintaan secara bersamaan.

5. Keamanan dan Autentikasi

Keamanan merupakan aspek krusial dalam perancangan REST API Marketplace, mengingat sistem ini mengelola data pribadi pengguna, kredensial login, serta informasi transaksi keuangan. Untuk menjamin keamanan data tersebut, diterapkan mekanisme autentikasi dan otorisasi yang ketat agar akses hanya diberikan kepada pihak yang berwenang.

5.1 Implementasi JSON Web Token (JWT)

Sesuai dengan konsep arsitektur *stateless* yang dipilih, autentikasi pada sistem ini dilakukan menggunakan JSON Web Token (JWT). Penggunaan JWT memungkinkan server untuk memvalidasi identitas pengguna tanpa harus menyimpan data *session* di sisi server, sehingga meningkatkan efisiensi dan skalabilitas sistem.

Proses kerja JWT dalam sistem ini adalah sebagai berikut:

- Login & Verifikasi: Pengguna mengirimkan email dan kata sandi melalui *endpoint* login.
- Generasi Token: Jika data valid, server akan menghasilkan token unik yang berisi informasi identitas pengguna dan waktu kedaluwarsa.
- Otorisasi: Token tersebut disimpan oleh sisi klien dan harus disertakan dalam *header* setiap permintaan API untuk mengakses fitur-fitur terbatas, seperti mengelola produk atau melakukan pemesanan.

5.2 Manajemen Peran (Role-Based Access Control)

Sistem menerapkan pembatasan akses berdasarkan peran pengguna untuk menjaga integritas data. Terdapat dua peran utama dalam marketplace ini:

- Penjual (Seller): Memiliki hak akses khusus untuk menambah, menyunting, dan menghapus data inventaris produk mereka sendiri.
- Pembeli (Buyer): Memiliki hak akses untuk melakukan pencarian produk, melakukan pemesanan, dan memantau status transaksi mereka.

5.3 Enkripsi dan Validasi Data

Selain autentikasi token, lapisan keamanan tambahan juga diimplementasikan melalui:

- Password Hashing: Kata sandi pengguna tidak pernah disimpan dalam bentuk teks biasa, melainkan dienkripsi menggunakan algoritma *hashing* yang kuat sebelum masuk ke database untuk mencegah kebocoran data.

- Validasi Input: Setiap data yang masuk melalui API divalidasi secara ketat untuk mencegah serangan umum seperti *SQL Injection* atau *Cross-Site Scripting* (XSS).
- HTTPS Encryption: Seluruh komunikasi antara aplikasi *frontend* dan REST API wajib dilakukan melalui protokol HTTPS guna memastikan data yang dikirimkan terenkripsi selama perjalanan di jaringan internet

6. Implementasi dan Pengujian

Bagian ini memaparkan langkah-langkah teknis dalam merealisasikan rancangan REST API menjadi sistem yang fungsional serta serangkaian pengujian untuk memastikan kualitas dan keandalan sistem marketplace tersebut.

6.1 Implementasi Teknologi

Implementasi ini menggunakan arsitektur *backend* modern yang berfokus pada efisiensi pengolahan data dan kemudahan akses lintas platform, baik untuk website maupun aplikasi mobile.

- Bahasa Pemrograman & Framework: Menggunakan teknologi seperti Node.js (Express) atau Python (FastAPI/Django) yang mendukung operasi *asynchronous* untuk performa tinggi.
- Format Pertukaran Data: Seluruh data yang dikirim dan diterima menggunakan format JSON (JavaScript Object Notation) karena sifatnya yang ringan dan standar di industri digital.
- Stateless Architecture: Implementasi tetap berpegang pada prinsip *stateless*, di mana setiap *request* dari klien harus membawa informasi lengkap (token) tanpa bergantung pada data yang disimpan di server.

6.2 Dokumentasi API dengan Postman/Swagger

Agar API dapat diakses dengan mudah oleh pengembang *frontend*, dilakukan dokumentasi secara sistematis.

- Endpoint Catalog: Mencantumkan seluruh daftar URL, metode HTTP (GET, POST, PUT, DELETE), serta parameter yang diperlukan.
- Request & Response Example: Menyediakan contoh data input dan output sehingga meminimalisir kesalahan integrasi antara sisi klien dan server.

6.3 Skenario Pengujian (Testing)

Pengujian dilakukan untuk memastikan bahwa logika bisnis marketplace, seperti pengelolaan produk oleh penjual dan pemesanan oleh pembeli, berjalan sesuai rencana.

- Unit Testing: Menguji fungsi-fungsi terkecil secara mandiri, seperti validasi format email saat registrasi pengguna.
- Functional Testing: Memastikan alur transaksi mulai dari penambahan produk ke keranjang hingga status pesanan dapat dilacak secara akurat.
- Authentication Testing: Mencoba mengakses *endpoint* terbatas tanpa token JWT untuk memastikan sistem keamanan menolak akses yang tidak sah.

- Load Testing: Menguji ketahanan sistem saat menerima lonjakan trafik transaksi guna memastikan skalabilitas sistem tetap stabil.

6.4 Hasil Pengujian

Berdasarkan hasil uji coba, sistem telah mampu menangani proses manajemen akun, pengelolaan inventaris produk secara *real-time*, hingga proses transaksi ekonomi digital secara *end-to-end* dengan respon waktu yang optimal.

7. Penutup

7.1 Kesimpulan

Berdasarkan seluruh tahapan perancangan yang telah dipaparkan, dapat disimpulkan bahwa desain REST API untuk Marketplace Online ini telah memenuhi kebutuhan fundamental dalam ekosistem jual beli digital. Implementasi konsep RESTful API memberikan solusi backend yang terstruktur dan aman, serta memiliki fleksibilitas tinggi untuk diakses oleh berbagai platform seperti website maupun aplikasi mobile.

Penggunaan arsitektur *stateless* dan autentikasi berbasis JWT (JSON Web Token) terbukti menjadi pilihan yang ideal untuk menjaga skalabilitas sistem saat menangani lonjakan transaksi. Selain itu, perancangan database relasional yang matang menjamin integritas data dalam mengelola interaksi antara penjual dan pembeli secara *end-to-end*, mulai dari manajemen produk hingga transparansi status pesanan. Secara keseluruhan, perancangan ini relevan dengan kebutuhan industri saat ini karena mampu menggabungkan efisiensi operasional dengan standar keamanan data yang kuat.

7.2 Saran

Untuk pengembangan sistem lebih lanjut, terdapat beberapa aspek yang dapat ditingkatkan guna memperkaya fungsionalitas marketplace ini:

- Pengembangan Fitur: Menambahkan sistem *rating* dan ulasan produk untuk meningkatkan kepercayaan pembeli terhadap penjual.
- Optimasi Keamanan: Mengimplementasikan fitur *refresh token* untuk memperkuat keamanan sesi pengguna tanpa mengorbankan kenyamanan akses.
- Skalabilitas Lanjut: Mempertimbangkan penggunaan teknologi *caching* (seperti Redis) untuk mempercepat respon API pada data katalog produk yang sering diakses.
- Ekspansi Integrasi: Memperluas integrasi API publik dengan lebih banyak pilihan kurir logistik dan metode pembayaran elektronik lainnya guna meningkatkan pengalaman pengguna.