

Laporan Progres Mingguan - SICATE

Kelompok: 8

Nama Anggota Kelompok:

1. Andini Permata Sari (10231015)
2. Chelsy Olivia (10231025)
3. Jonathan Cristopher Jetro (10231047)
4. Nicholas Christian Samuel Manurung (10231069)

Mitra: Toba Home Catering

Pekan ke-: 11

Tanggal: 22/04/2025

Progress Summary

Progres pada minggu ini dilanjutkan dengan merancang skema database berdasarkan kebutuhan fitur MVP, lalu mengimplementasikannya ke dalam PostgreSQL untuk memastikan data tersimpan dengan baik dan terstruktur. Setelah itu, tim membuat kerangka dasar REST API untuk menangani endpoint utama serta menyiapkan struktur awal antarmuka pengguna pada sisi frontend. Seluruh progres tersebut kemudian didemokan kepada mitra untuk mendapatkan masukan dan validasi pengembangan sistem.

Accomplished Tasks

- Merancang skema database (ERD diagram) dan mengimplementasikan ke dalam PostgreSQL
- Membuat Endpoint dasar (POST & GET)
- Menyiapkan struktur frontend basic

Challenges & Solutions

- **Challenge 1:** Tim kesulitan dalam membuat halaman frontend karena perlu menyesuaikan desain dengan fungsionalitas yang direncanakan.
 - **Solution:** Solusinya adalah mencari error pada website untuk mengetahui letak error nya lalu menelusuri sumber masalahnya agar dapat segera diperbaiki.

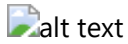
Next Week Plan

Contributions

- **Chelsy Olivia:** Membuat laporan (Md), membuat ERD
- **Andini Permata Sari:** Membantu membuat laporan (Md), membantu membuat ERD
- **Jonathan Cristopher Jetro:** Membuat struktur database, membuat frontend dan backend skeleton
- **Nicholas Christian Samuel Manurung:** Membantu jetro

Screenshots

1. Skema Database (ERD Dia)



Skema database pada diagram tersebut menggambarkan sistem e-commerce yang terdiri dari enam entitas utama: users, products, cart, orders, order_items, dan reviews. Setiap entitas saling terhubung untuk merepresentasikan proses bisnis dari pengguna yang berinteraksi dengan produk, melakukan pembelian, serta memberikan ulasan.

Entitas users menyimpan data pengguna seperti username, email, password, dan role. Seorang user dapat menambahkan banyak produk ke dalam keranjang (cart), yang setiap itemnya mengacu pada satu produk tertentu dari entitas products. Produk sendiri memiliki atribut seperti name, description, price, dan stock. Ketika pengguna melakukan pembelian, data pesanan disimpan dalam entitas orders, yang mencatat total harga, status, dan status pembayaran. Satu order bisa memiliki beberapa item pesanan (order_items), masing-masing menghubungkan produk yang dibeli dalam jumlah tertentu beserta harga pada saat transaksi.

Setelah pembelian, pengguna dapat memberikan penilaian atau ulasan terhadap produk melalui entitas reviews. Setiap review mencatat rating dan comment, serta menghubungkan user dengan produk yang diulas. Seluruh relasi antar entitas menggunakan model one-to-many (1:N), yang berarti satu entitas utama seperti user atau produk bisa terhubung ke banyak entitas terkait seperti cart, orders, order_items, dan reviews.

2. REST API skeleton (endpoint dasar)

Autentikasi

Registrasi

```
POST /api/auth/register
```

Request Body:

```
{
  "username": "string",
  "email": "string",
  "password": "string",
  "role": "customer" // opsional, default: customer
}
```

Response:

```
{
  "token": "string",
  "user": {
    "id": "integer",
    "username": "string",
    "email": "string",
    "role": "string"
  }
}
```

Login

POST /api/auth/login

Request Body:

```
{
  "email": "string",
  "password": "string"
}
```

Response:

```
{
  "token": "string",
  "user": {
    "id": "integer",
    "username": "string",
    "email": "string",
    "role": "string"
  }
}
```

Produk

Mendapatkan Semua Produk

GET /api/products

Response:

```
[
  {
    "id": "integer",
    "name": "string",
    "description": "string",
    "price": "number",
    "image_url": "string",
    "category": "string",
    "stock": "integer"
  }
]
```

Mendapatkan Detail Produk

GET /api/products/{id}

Response:

```
{
  "id": "integer",
  "name": "string",
  "description": "string",
  "price": "number",
  "image_url": "string",
  "category": "string",
  "stock": "integer"
}
```

Membuat Produk Baru (Admin)

POST /api/products

Headers:

Authorization: Bearer {token}

Request Body:

```
{
  "name": "string",
  "description": "string",
}
```

```
"price": "number",  
"image_url": "string",  
"category": "string",  
"stock": "integer"  
}
```

Keranjang

Mendapatkan Keranjang

GET /api/cart

Headers:

Authorization: Bearer {token}

Response:

```
[  
  {  
    "id": "integer",  
    "product": {  
      "id": "integer",  
      "name": "string",  
      "price": "number",  
      "image_url": "string"  
    },  
    "quantity": "integer"  
  }  
]
```

Menambah Item ke Keranjang

POST /api/cart

Headers:

Authorization: Bearer {token}

Request Body:

```
{
  "product_id": "integer",
  "quantity": "integer"
}
```

Pesanan

Mendapatkan Semua Pesanan

GET /api/orders

Headers:

Authorization: Bearer {token}

Response:

```
[
  {
    "id": "integer",
    "total_amount": "number",
    "status": "string",
    "payment_status": "string",
    "created_at": "string"
  }
]
```

Membuat Pesanan Baru

POST /api/orders

Headers:

Authorization: Bearer {token}

Response:

```
{
  "id": "integer",
  "total_amount": "number",
  "status": "pending",
  "payment_status": "unpaid",
  "created_at": "string"
}
```

Ulasan

Membuat Ulasan Baru

POST /api/reviews

Headers:

Authorization: Bearer {token}

Request Body:

```
{
  "product_id": "integer",
  "rating": "integer",
  "comment": "string"
}
```

Response:

```
{
  "id": "integer",
  "product_id": "integer",
  "rating": "integer",
  "comment": "string",
  "created_at": "string"
}
```

Kode Status

- 200: OK
- 201: Created
- 400: Bad Request

- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

Catatan

- Semua endpoint yang memerlukan autentikasi harus menyertakan token JWT di header `Authorization`
- Format token: `Bearer {token}`
- Role yang tersedia: `admin` dan `customer`
- Status pesanan: `pending`, `processing`, `completed`, `cancelled`
- Status pembayaran: `unpaid`, `paid`
- Rating ulasan: 1-5

Untuk lebih jelas bisa dibuka di repositori berikut `TubesProwebA8\ecommerce-backend\docs\API.md`

3. Struktur Frontend Basic

Struktur Folder dan File Ecommerce-Frontend

```
ecommerce-frontend/
├── package.json
├── package-lock.json
├── .env
├── tailwind.config.js
├── vite.config.js
├── eslint.config.js
├── index.html
├── .gitignore
├── README.md
├── public/
│   ├── manifest.json
│   ├── index.html
│   └── vite.svg
├── src/
│   ├── index.js
│   ├── index.css
│   ├── App.js
│   ├── reportWebVitals.js
│   └── components/
│       ├── layout/
│       │   ├── Layout.js
│       │   └── Footer.js
```



```
├── auth/
│   ├── Login.js
│   └── Register.js
├── products/
│   ├── ProductList.js
│   └── ProductDetail.js
├── cart/
│   └── CartList.js
├── orders/
│   ├── OrderList.js
│   └── OrderDetail.js
├── services/
│   ├── authService.js
│   ├── productService.js
│   ├── cartService.js
│   ├── orderService.js
│   └── reviewService.js
├── utils/
│   └── (fungsi-fungsi utilitas)
├── store/
│   └── (state management)
├── pages/
│   └── (komponen halaman)
├── assets/
│   └── (gambar, font, aset statis)
└── node_modules/
    └── (dependensi yang diinstal)
```

Berikut adalah struktur folder frontend basic di local path, Struktur ini menunjukkan hierarki folder dan file dalam proyek, dengan cabang yang menurun ke bawah untuk setiap direktori.

 alt text

Pada gambar diatas merupakan awalan untuk aplikasi web yang bernama SI CATE yang berfungsi untuk melihat produk makanan catering. Bisa dilihat pada gambar, progres awalan sudah terdapat menu navigasi yang terdiri dari Produk, Keranjang, dan Pesanan serta di sisi kanan terdapat opsi Profile dan Logout yang menandakan bahwa pengguna telah masuk ke dalam akun mereka. Secara keseluruhan, halaman ini menampilkan tampilan katalog produk yang berisi nama makanan dan harga.