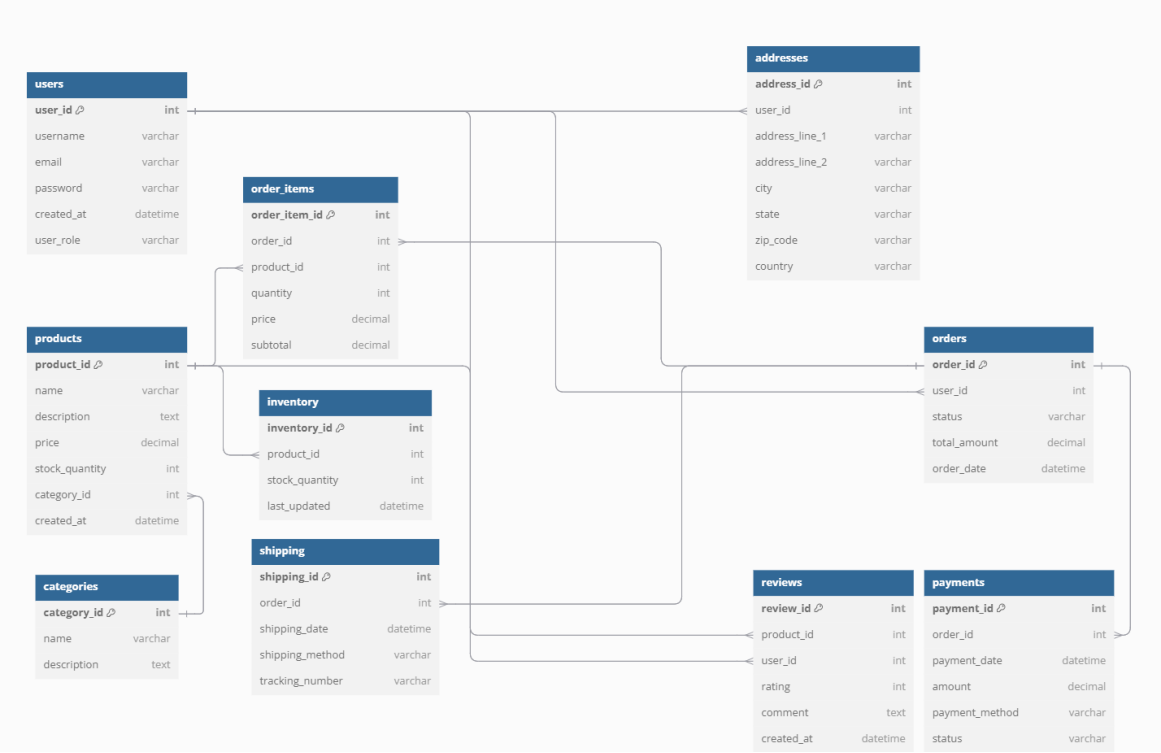
Dokumentasi Web Service E-Commerce

Pendahuluan

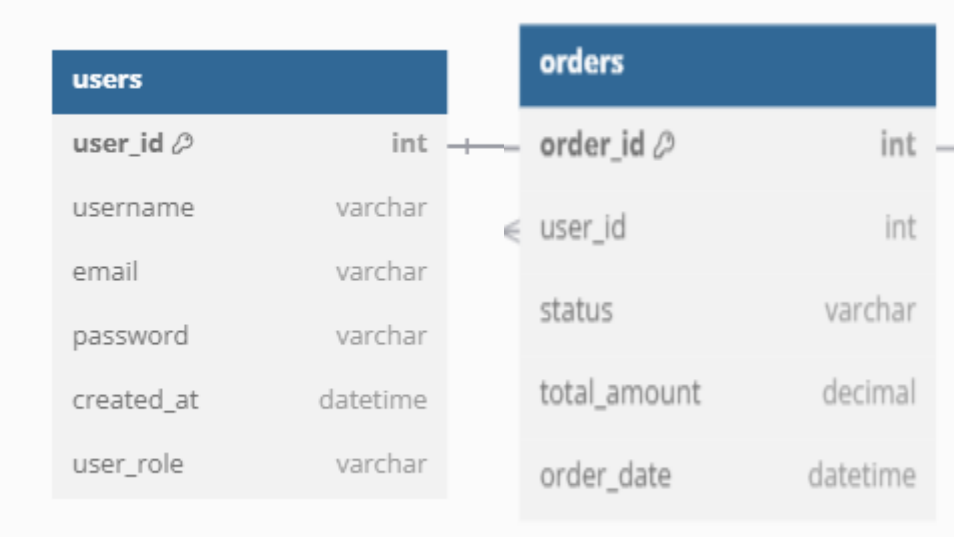
Aplikasi e-commerce yang direncanakan dalam dokumen ini bertujuan untuk menyediakan solusi lengkap bagi pelanggan dan penjual dalam melakukan transaksi online. Sistem ini dilengkapi dengan fitur manajemen pengguna, katalog produk, kategori produk, pengelolaan stok, sistem pesanan, pembayaran, dan pengiriman. Untuk mendukung interaksi antara klien dan server, kami merancang layanan web berbasis RESTful API. API ini menyediakan endpoint-endpoint yang memungkinkan pengguna aplikasi mengakses dan mengelola data secara efisien.

Tujuan dari dokumentasi ini adalah untuk menjelaskan spesifikasi teknis API e-commerce, termasuk metode HTTP, format data, serta contoh permintaan dan respons. Dengan demikian, pengembang front-end dan pihak lain yang berkepentingan dapat memahami cara menggunakan API ini untuk mengintegrasikan fungsionalitas e-commerce ke dalam aplikasi yang mereka kembangkan.

Skema Database

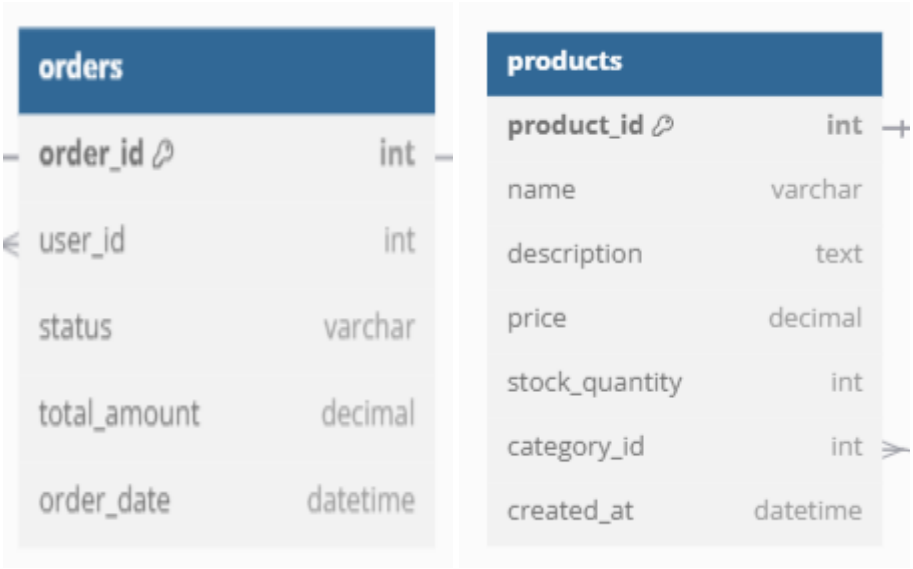


Penjelasan



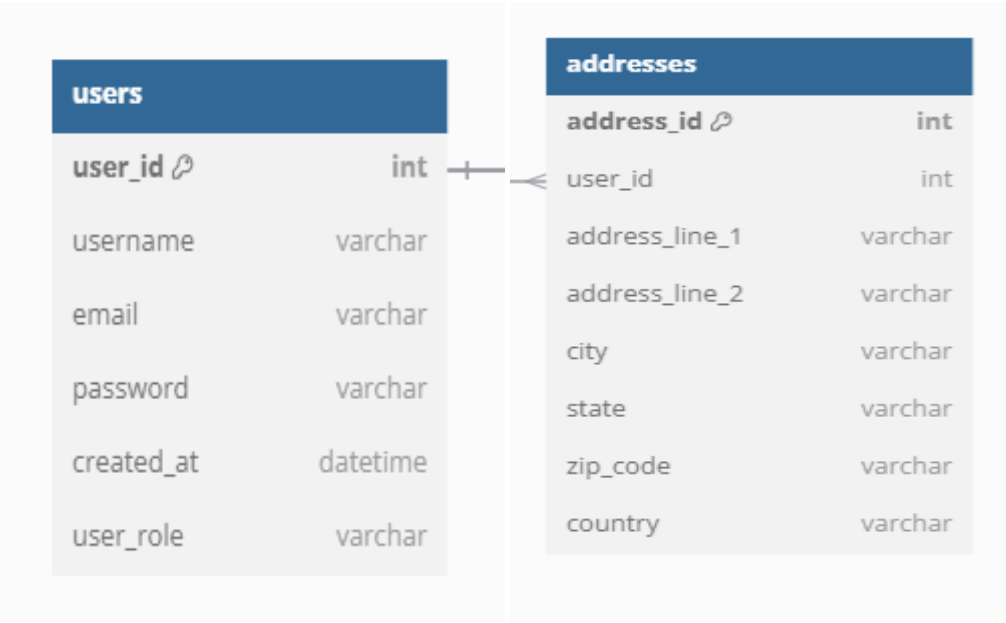
1. Users ↔ Orders (One-to-Many):
    - Penjelasan: Setiap pengguna dapat melakukan beberapa pesanan, tetapi setiap pesanan hanya dikaitkan dengan satu pengguna.
    - Contoh: Seorang pengguna dengan user_id = 1 dapat melakukan beberapa pesanan (misalnya, order_id = 1001, 1002), tetapi setiap pesanan hanya mengarah ke satu pengguna.

2. Orders ↔ Order_Items (One-to-Many):
    - Penjelasan: Setiap pesanan dapat memiliki banyak item, tetapi setiap item pesanan mengacu pada satu pesanan.
    - Contoh: Untuk order_id = 1001, mungkin ada beberapa item dalam Order_Items (misalnya, order_item_id = 5001, 5002), masing-masing mewakili produk tertentu dalam pesanan tersebut.

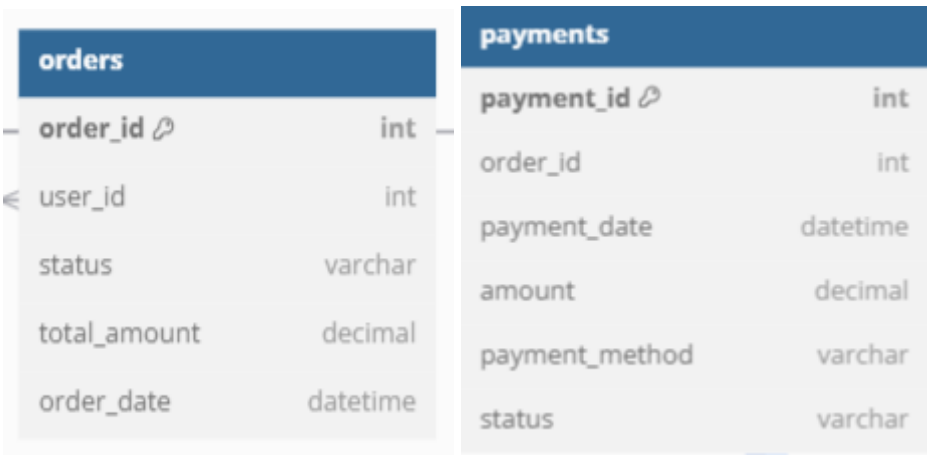


3. Products ↔ Order_Items (One-to-Many):
    - Penjelasan: Setiap produk dapat muncul di beberapa item pesanan (karena beberapa pelanggan dapat memesannya), tetapi setiap item pesanan hanya mengacu pada satu produk.
    - Contoh: Jika sebuah produk dengan product_id = 101 dipesan beberapa kali, produk tersebut akan muncul di baris yang berbeda di Order_Items, setiap kali dengan order_item_id yang unik.

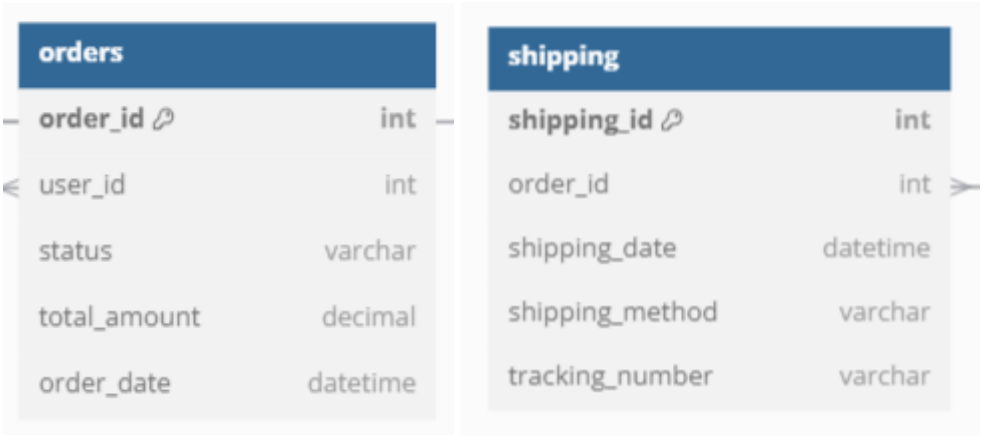
4. Categories ↔ Products (One-to-Many):
    - Penjelasan: Setiap produk termasuk dalam satu kategori, sementara satu kategori dapat mencakup beberapa produk.
    - Contoh: Sebuah kategori dengan category_id = 10 (misalnya, “Elektronik”) dapat mencakup beberapa produk, seperti product_id = 101, 102, dll.



5. Users ↔ Addresses (One-to-Many):
    - Penjelasan: Setiap pengguna dapat memiliki beberapa alamat (misalnya, rumah, kantor), tetapi setiap alamat hanya dimiliki oleh satu pengguna.
    - Contoh: Pengguna dengan user_id = 1 dapat memiliki alamat dengan address_id = 201 (rumah) dan address_id = 202 (kantor).

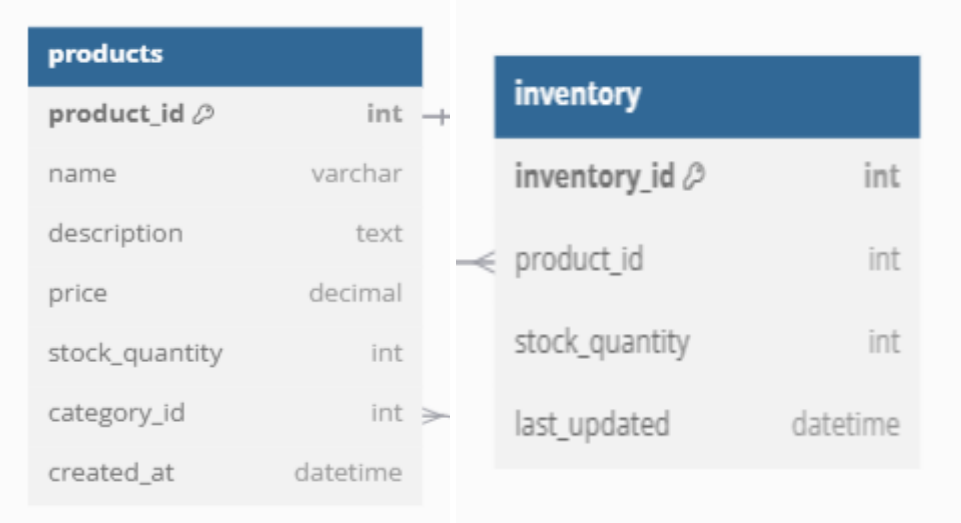


6. Orders ↔ Payments (One-to-One):
    - Penjelasan: Setiap pesanan memiliki satu catatan pembayaran yang merinci jumlah yang dibayarkan, metode, dan status. Setiap catatan pembayaran hanya terkait dengan satu pesanan.
    - Contoh: order_id = 1001 memiliki catatan pembayaran payment_id = 3001 yang menunjukkan bahwa pesanan tersebut dibayar dengan kartu kredit.



7. Orders ↔ Shipping (One-to-One):
    - Penjelasan: Setiap pesanan memiliki catatan pengiriman yang sesuai yang merinci tanggal pengiriman, metode, dan nomor pelacakan.
    - Contoh: order_id = 1001 memiliki shipping_id = 7001 dengan informasi tentang nomor dan tanggal pelacakan pengiriman.

8. Users ↔ Reviews ↔ Products (Many-to-Many via Reviews Table):
    - Penjelasan: Seorang pengguna dapat mengulas beberapa produk, dan setiap produk dapat memiliki ulasan dari beberapa pengguna. Tabel Ulasan bertindak sebagai jembatan antara Pengguna dan Produk.
    - Contoh: Pengguna dengan user_id = 1 mengulas product_id = 101, sehingga menghasilkan review_id = 4001. Pengguna lain juga dapat mengulas produk yang sama atau produk yang berbeda.



9. Products ↔ Inventory (One-to-One):
    - Penjelasan: Setiap produk memiliki satu catatan inventaris yang melacak jumlah stoknya, yang diperbarui setiap kali terjadi perubahan inventaris.
    - Contoh: product_id = 101 memiliki inventory_id = 6001, yang menunjukkan stok 100 unit. Ketika stok berubah, tabel Inventory akan diperbarui.

Restful API Endpoints'

1. Users

- **GET** /api/users:

  Mengambil daftar semua pengguna.

- **POST** /api/users:

  Membuat pengguna baru.

- **GET** /api/users/{user_id}:

  Mengambil detail untuk pengguna tertentu.

- **PUT** /api/users/{user_id}:

  Memperbarui detail pengguna yang sudah ada.

- **DELETE** /api/users/{user_id}:

  Menghapus pengguna tertentu.

2. Products

- **GET** /api/products:

  Mengambil daftar semua produk.

- **POST** /api/products:

  Menambahkan produk baru ke katalog.
- **GET** /api/products/{product_id}:

  Mengambil detail untuk produk tertentu.
- **PUT** /api/products/{product_id}:

  Memperbarui detail produk.
- **DELETE** /api/products/{product_id}:

  Menghapus produk dari katalog.

3. Categories

- **GET** /api/categories:

  Mengambil semua kategori produk.
- **POST** /api/categories:

  Membuat kategori baru.
- **GET** /api/categories/{category_id}:

  Mengambil detail untuk kategori tertentu.
- **PUT** /api/categories/{category_id}:

  Memperbarui sebuah kategori.
- **DELETE** /api/categories/{category_id}:

  Menghapus kategori.

4. Orders

- **GET** /api/orders:

  Mengambil semua pesanan.
- **POST** /api/orders:

  Membuat pesanan baru.
- **GET** /api/orders/{order_id}:

  Mengambil detail pesanan tertentu.
- **PUT** /api/orders/{order_id}:

  Memperbarui detail pesanan.

- **DELETE** /api/orders/{order_id}:

  Menghapus pesanan.

5. Order Items

- **GET** /api/orders/{order_id}/items:

  Mengambil item dalam urutan tertentu.
- **POST** /api/orders/{order_id}/items:

  Menambahkan item ke dalam pesanan.
- **GET** /api/orders/{order_id}/items/{order_item_id}:

  Mengambil item pesanan tertentu.
- **PUT** /api/orders/{order_id}/items/{order_item_id}:

  Memperbarui detail item pesanan.
- **DELETE** /api/orders/{order_id}/items/{order_item_id}:

  Menghapus item pesanan.

6. Addresses

- **GET** /api/users/{user_id}/addresses:

  Mengambil semua alamat untuk seorang pengguna.
- **POST** /api/users/{user_id}/addresses:

  Menambahkan alamat baru untuk pengguna.
- **GET** /api/users/{user_id}/addresses/{address_id}:

  Mengambil alamat tertentu untuk pengguna.
- **PUT** /api/users/{user_id}/addresses/{address_id}:

  Memperbarui alamat pengguna.
- **DELETE** /api/users/{user_id}/addresses/{address_id}:

  Menghapus alamat tertentu.

7. Payments

- **GET** /api/orders/{order_id}/payment:

  Mengambil informasi pembayaran untuk pesanan tertentu.
- **POST** /api/orders/{order_id}/payment:

Membuat catatan pembayaran baru untuk sebuah pesanan.

- **PUT** /api/orders/{order_id}/payment:

  Memperbarui detail pembayaran untuk suatu pesanan.

8. Reviews

- **GET** /api/products/{product_id}/reviews:

  Mengambil semua ulasan untuk produk tertentu.

- **POST** /api/products/{product_id}/reviews:

  Menambahkan ulasan baru untuk suatu produk.

- **GET** /api/products/{product_id}/reviews/{review_id}:

  Mengambil detail ulasan tertentu.

- **PUT** /api/products/{product_id}/reviews/{review_id}:

  Memperbarui ulasan.

- **DELETE** /api/products/{product_id}/reviews/{review_id}:

  Menghapus ulasan.

9. Inventory

- **GET** /api/products/{product_id}/inventory:

  Memeriksa tingkat inventaris untuk produk tertentu.

- **PUT** /api/products/{product_id}/inventory:

  Memperbarui inventaris untuk suatu produk (misalnya, setelah penjualan).

10. Shipping

- **GET** /api/orders/{order_id}/shipping:

  Mengambil detail pengiriman untuk pesanan tertentu.

- **POST** /api/orders/{order_id}/shipping:

  Membuat informasi pengiriman untuk sebuah pesanan.

- **PUT** /api/orders/{order_id}/shipping:

  Memperbarui detail pengiriman (misalnya, nomor pelacakan).

Contoh API Request dan Response

- **Membuat Order Baru (POST /api/orders)**

- **Request**

```
{
  "user_id": 1,
  "status": "pending",
  "total_amount": 150.75,
  "order_date": "2024-11-04"
}
```

- **Response**

```
{
  "order_id": 1001,
  "user_id": 1,
  "status": "pending",
  "total_amount": 150.75,
  "order_date": "2024-11-04"
}
```

- **Memanggil Semua Product (GET /api/products)**
  - **Response**

```
[
  {
    "product_id": 101,
    "name": "Wireless Mouse",
    "description": "A high-quality wireless mouse.",
    "price": 29.99,
    "stock_quantity": 50,
    "category_id": 10,
    "created_at": "2024-01-01"
  },
  {
    "product_id": 102,
    "name": "USB-C Charger",
    "description": "Fast charging USB-C charger.",
    "price": 15.99,
    "stock_quantity": 100,
    "category_id": 10,
    "created_at": "2024-01-05"
  }
]
```