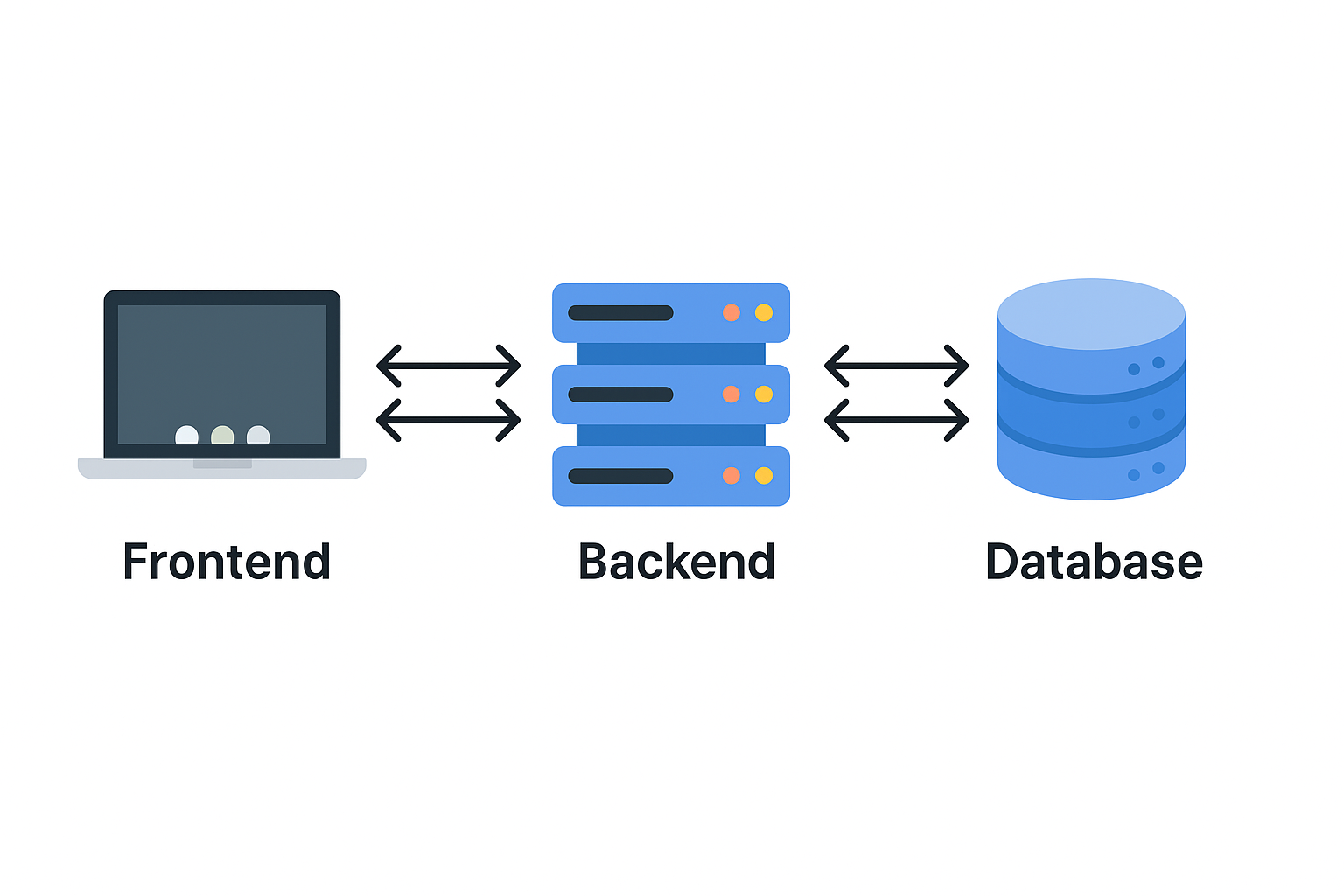
# PENGEMBANGAN SHOPWISE COMMERCE

# Modul Backend ShopWise: Spesifikasi dan Implementasi

# ARSITEKTUR SISTEM



*Gambar 1 Arsitektur Sistem*

# SHOPWISE COMMERCE ENTITY RELATIONSHIP DIAGRAM

**A screenshot of a computer

Description automatically generated***Gambar 2 Entity Relationship Diagram (ERD)*

# 

# LKS MART SISTEM INFORMASI DATA DICTIONARY

*Tabel 1 products*

A screenshot of a computer

Description automatically generated

*Tabel 2 cart\_items*

A screenshot of a computer

AI-generated content may be incorrect.

*Tabel 3 categories*

A screenshot of a black screen

AI-generated content may be incorrect.

*Tabel 4 order\_items*

A screenshot of a computer

AI-generated content may be incorrect.

*Tabel 5 orders*

*A screenshot of a computer

AI-generated content may be incorrect.*

*Tabel 6 users*

A screenshot of a computer

AI-generated content may be incorrect.

*Entity Relationship*

# 

**Tahap 1: Persiapan dan Database**

1. **Pemilihan Teknologi:** Tentukan bahasa pemrograman dan *framework* backend yang akan digunakan (Laravel, Express).
2. **Setup Lingkungan:** Siapkan lingkungan pengembangan (instalasi bahasa, *framework*, database, *tools* seperti Postman).
3. **Implementasi Skema Database:**

* Buat database (MySQL).
* Buat tabel-tabel sesuai ERD yang disempurnakan:
* users
* categories
* products
* product\_variations [optional]
* addresses [optional]
* cart\_items
* orders (Dengan kolom tambahan: shipping\_address\_id, billing\_address\_id, payment\_method, shipping\_method, shipping\_cost, estimated\_delivery\_date, order\_number [opsional])
* order\_items
* Definisikan relasi antar tabel (Foreign Keys).
* *Sangat disarankan:* Gunakan *migration tool* dari *framework* Anda untuk mengelola perubahan skema database.

1. **Pengisian Data Awal (Seeding):** Isi setiap tabel dengan data awal (minimal 5 record per tabel sesuai soal) untuk keperluan pengujian.

**Tahap 2: Modul Inti Backend**

1. **Autentikasi & Autorisasi:**

* API endpoint untuk registrasi pengguna (role: 'customer').
* API endpoint untuk login (menghasilkan token/session untuk 'customer' dan 'admin').
* Implementasi *middleware* untuk melindungi rute/endpoint yang memerlukan login.
* Implementasi *middleware* untuk membatasi akses ke rute admin hanya untuk pengguna dengan role 'admin'.

1. **Manajemen Kategori (Admin):**

* API CRUD (Create, Read, Update, Delete) untuk tabel categories.

1. **Manajemen Produk & Variasi (Admin):**

* API CRUD untuk tabel products (termasuk upload gambar produk).
* API CRUD untuk tabel product\_variations yang terkait dengan produk.
* API untuk menandai/membatalkan produk sebagai *featured* (is\_featured).

1. **Manajemen Alamat (User):**

* API CRUD untuk tabel addresses yang terkait dengan pengguna yang login.
* API untuk menandai alamat sebagai alamat utama (is\_primary).

**Tahap 3: Modul Transaksi**

1. **Katalog Produk (Publik/User):**

* API untuk menampilkan semua produk (dengan fitur pencarian, filter per kategori, pagination).
* API untuk menampilkan produk unggulan (is\_featured = TRUE).
* API untuk menampilkan detail satu produk (termasuk data variasinya).

1. **Manajemen Keranjang Belanja (User):**

* API untuk menambahkan produk/variasi ke cart\_items.
* API untuk melihat isi cart\_items pengguna yang login.
* API untuk mengubah jumlah item di cart\_items.
* API untuk menghapus item dari cart\_items.

1. **Proses Checkout & Pemesanan (User):**

* API untuk membuat pesanan baru:
* Menerima data keranjang, ID alamat pengiriman (dan penagihan jika ada), metode pembayaran, metode pengiriman.
* Membuat record baru di tabel orders.
* Memindahkan item dari cart\_items ke order\_items (menyimpan harga saat itu).
* Menghitung total\_price (termasuk shipping\_cost).
* Mengurangi stok produk/variasi yang relevan.
* Mengosongkan cart\_items pengguna.
* Mengembalikan detail pesanan yang baru dibuat (termasuk nomor pesanan).

**Tahap 4: Modul Pasca-Transaksi & Admin Lanjutan**

1. **Riwayat Pesanan (User):**

* API untuk menampilkan daftar pesanan milik pengguna yang login (dengan status).
* API untuk menampilkan detail lengkap satu pesanan dari riwayat.

1. **Manajemen Pesanan (Admin):**

* API untuk menampilkan semua pesanan (dengan fitur filter berdasarkan status, pencarian, pagination).
* API untuk melihat detail lengkap satu pesanan (termasuk item, alamat, info pelanggan).
* API untuk memperbarui status pesanan (orders.status).
* API untuk menghapus pesanan (pertimbangkan *soft delete* daripada penghapusan permanen).