

Studi Kasus 1 : Sistem Penjualan Online

Anda ditugaskan untuk merancang sistem penjualan online untuk sebuah toko retail. Sistem ini harus memungkinkan pelanggan untuk menelusuri produk, menambahkan produk ke keranjang belanja, dan menyelesaikan pembelian.

- a. Identifikasi objek-objek yang diperlukan dalam sistem penjualan online.

JAWAB :

Pelanggan: Individu yang melakukan pembelian.

Produk: Barang atau jasa yang dijual.

Pembayaran: Proses transaksi pembayaran.

Pesanan: Rekapitulasi produk yang dibeli oleh pelanggan dalam satu transaksi.

Keranjang Belanja: Tempat sementara untuk menyimpan produk yang dipilih pelanggan sebelum checkout.

- b. Bagaimana Anda akan mengorganisasi objek-objek tersebut ke dalam kelas-kelas dalam model berorientasi objek?

JAWAB:

Customer: Mewakili pelanggan yang menggunakan sistem.

Product: Mewakili produk yang dijual di toko.

ShoppingCart: Menyimpan produk yang dipilih oleh pelanggan.

Order: Mewakili pesanan yang dibuat setelah pelanggan melakukan checkout.

Payment: Mengelola pembayaran dari pelanggan.

Shipping: Mengelola pengiriman produk.

ProductCategory: Mewakili kategori dari produk-produk yang tersedia di toko.

- c. Apa saja atribut yang dapat dimiliki oleh kelas-kelas tersebut, seperti atribut produk atau keranjang belanja?

JAWAB:

Kelas Pelanggan:

- idPelanggan
- nama
- alamat
- email

- nomorTelepon
- password

Kelas Produk:

- idProduk
- namaProduk
- harga
- deskripsi
- stok
- kategori

Kelas Keranjang Belanja:

- idKeranjang
- daftarProduk
- totalHarga

Kelas Pesanan:

- idPesanan
- tanggalPesanan
- totalHarga
- statusPesanan
- daftarProduk

Kelas Pembayaran:

- idPembayaran
- metodePembayaran
- statusPembayaran

d. Buatlah beberapa contoh metode atau operasi yang bisa dilakukan oleh kelas-kelas dalam sistem penjualan online.

JAWAB:

Kelas Pelanggan:

- register
- login
- ubahProfil

Kelas Produk:

- cariProduk
- tampilkanDetailProduk

Kelas Keranjang Belanja:

- tambahProduk
- hapusProduk
- hitungTotalHarga

Kelas Pesanan:

- buatPesanan
- bayarPesanan
- lihatStatusPesanan

Kelas Pembayaran:

- prosesPembayaran
- pemilihan pembayaran

e. Bagaimana Anda akan menghubungkan objek-objek ini melalui relasi atau asosiasi, seperti hubungan antara pelanggan, produk, dan keranjang belanja?

JAWAB:

1. Pelanggan dan Keranjang Belanja:

- **Hubungan:** Satu pelanggan dapat memiliki banyak keranjang belanja (misalnya, keranjang belanja untuk sesi belanja yang berbeda).
- **Implementasi:** Dalam ERD, hubungan ini digambarkan sebagai **satu-ke-banyak**. Kita akan menambahkan atribut "idPelanggan" pada entitas Keranjang Belanja sebagai foreign key yang mengacu pada entitas Pelanggan.

2. Keranjang Belanja dan Produk:

- **Hubungan:** Satu keranjang belanja dapat berisi banyak produk, dan satu produk dapat berada di banyak keranjang belanja.
- **Implementasi:** Karena ini adalah hubungan banyak-ke-banyak, kita perlu menambahkan entitas tambahan yang disebut **Detail Keranjang** (atau bisa juga disebut "Item Keranjang"). Entitas ini akan memiliki atribut:

- idDetailKeranjang (sebagai primary key)
- idKeranjang (foreign key mengacu ke Keranjang Belanja)
- idProduk (foreign key mengacu ke Produk)
- jumlah (jumlah produk dalam keranjang)
- **ERD:** Kita akan membuat hubungan satu-ke-banyak antara Keranjang Belanja dan Detail Keranjang, serta antara Produk dan Detail Keranjang.

3. Pesanan dan Pelanggan, Produk:

- **Hubungan:** Satu pesanan dibuat oleh satu pelanggan dan berisi banyak produk.
- **Implementasi:** Sama seperti hubungan keranjang belanja dan produk, kita akan menambahkan entitas tambahan yang disebut **Detail Pesanan**. Entitas ini akan memiliki atribut:
 - idDetailPesanan (sebagai primary key)
 - idPesanan (foreign key mengacu ke Pesanan)
 - idProduk (foreign key mengacu ke Produk)
 - jumlah (jumlah produk dalam pesanan)
 - hargaSatuan (harga produk saat pemesanan)
- **ERD:** Kita akan membuat hubungan satu-ke-banyak antara Pelanggan dan Pesanan, serta antara Pesanan dan Detail Pesanan, dan antara Produk dan Detail Pesanan.

4. Pesanan dan Pembayaran:

- **Hubungan:** Satu pesanan hanya bisa memiliki satu pembayaran, dan satu pembayaran hanya terkait dengan satu pesanan.
- **Implementasi:** Hubungan ini adalah **satu-ke-satu**. Kita akan menambahkan atribut "idPesanan" pada entitas Pembayaran sebagai foreign key yang mengacu ke entitas Pesanan.

Customer → Satu ke Banyak → Order

Customer → Satu ke Satu → ShoppingCart

ShoppingCart → Banyak ke Banyak → Product

Order → Banyak ke Banyak → Product

Order → Satu ke Satu → Payment

Order → Satu ke Satu → Shipping

Product → Satu ke Banyak → ProductCategory