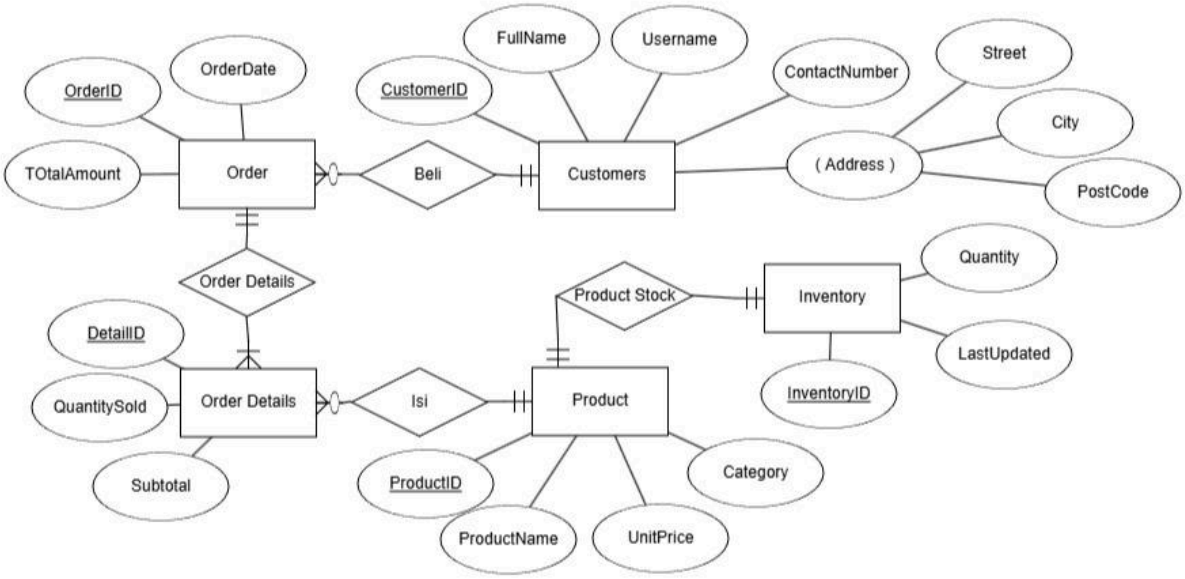
Penjelasan Schema Relational dan Implementasi SQL

Thufail Bahir - Dhimas Sulistio - Maulana Faris



Relational schema dari database **A*** adalah sebagai berikut:

```
Shell
Product(ProductID, ProductName, Category, UnitPrice)

Customers(CustomerID, FullName, Username, ContactNumber, City, PostCode,
Street)

Order(OrderID, OrderDate, TotalAmount, CustomerID)

Order_Details(DetailID, QuantitySold, Subtotal, OrderID, ProductID)

Inventory(InventoryID, Quantity, LastUpdated, ProductID)
```

Dengan *Primary Key (PK)* dan *Foreign Key (FK)* yang terdefinisi secara eksplisit:

- PK(Product) = ProductID
- PK(Customers) = CustomerID
- PK(Order) = OrderID
- PK(Order_Details) = DetailID
- PK(Inventory) = InventoryID

Relasi antar tabel (Foreign Key):

- Order.CustomerID → Customers.CustomerID
- Order_Details.OrderID → Order.OrderID
- Order_Details.ProductID → Product.ProductID
- Inventory.ProductID → Product.ProductID

Schema.sql :

```sql
-- Tabel Product

CREATE TABLE Product

(

  ProductID INT NOT NULL,

  ProductName VARCHAR(100) NOT NULL,

  Category VARCHAR(50) NOT NULL,

  UnitPrice DECIMAL(10, 2) NOT NULL,

  PRIMARY KEY (ProductID)

);


-- Tabel Customers

CREATE TABLE Customers

(

  CustomerID INT NOT NULL,

  FullName VARCHAR(100) NOT NULL,

  Username VARCHAR(50) NOT NULL,

  ContactNumber VARCHAR(20) NOT NULL,

  City VARCHAR(50) NOT NULL,

  PostCode VARCHAR(10) NOT NULL,

  Street VARCHAR(100) NOT NULL,
```

```
  PRIMARY KEY (CustomerID)
);

-- Tabel Order
CREATE TABLE Order
(
  OrderID INT NOT NULL,
  OrderDate DATE NOT NULL,
  TotalAmount DECIMAL(12, 2) NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (OrderID),
  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- Tabel Order_Details
CREATE TABLE Order_Details
(
  DetailID INT NOT NULL,
  QuantitySold INT NOT NULL,
  Subtotal DECIMAL(12, 2) NOT NULL,
  OrderID INT NOT NULL,
  ProductID INT NOT NULL,
  PRIMARY KEY (DetailID),
  FOREIGN KEY (OrderID) REFERENCES Order(OrderID),
  FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
```

```
);


-- Tabel Inventory
CREATE TABLE Inventory
(
  InventoryID INT NOT NULL,
  Quantity INT NOT NULL,
  LastUpdated DATETIME NOT NULL,
  ProductID INT NOT NULL,
  PRIMARY KEY (InventoryID),
  FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);
```

1. Tabel Product

```
SQL
CREATE TABLE Product (
  ProductID INT NOT NULL,
  ProductName VARCHAR(100) NOT NULL,
  Category VARCHAR(50) NOT NULL,
  UnitPrice DECIMAL(10, 2) NOT NULL,
  PRIMARY KEY (ProductID)
);
```

Fungsi dan Peran

Tabel **Product** menyimpan informasi mengenai setiap produk yang dijual oleh UMKM. Struktur ini memisahkan data produk dari data transaksi agar bisa digunakan secara berulang di berbagai order tanpa duplikasi.

Penjelasan Atribut

- **ProductID** → *Primary Key*.
  Berfungsi sebagai identitas unik untuk setiap produk. Tipe INT dipilih karena efisien dan mudah diurutkan.

- **ProductName** → Nama produk (maks. 100 karakter).

- **Category** → Jenis atau kategori produk (maks. 50 karakter).

- **UnitPrice** → Harga per unit barang dengan tipe DECIMAL(10,2) untuk mendukung presisi dua digit desimal (misal Rp 10000.50).

Constraint

PRIMARY KEY (ProductID) memastikan setiap produk unik. Tidak ada *foreign key* di sini karena tabel ini menjadi *entitas utama* yang direferensikan oleh tabel lain.

2. Tabel Customers

```SQL
CREATE TABLE Customers (
  CustomerID INT NOT NULL,
  FullName VARCHAR(100) NOT NULL,
  Username VARCHAR(50) NOT NULL,
  ContactNumber VARCHAR(20) NOT NULL,
  City VARCHAR(50) NOT NULL,
  PostCode VARCHAR(10) NOT NULL,
  Street VARCHAR(100) NOT NULL,
```

```
  PRIMARY KEY (CustomerID)
);
```

Fungsi dan Peran

Menyimpan data pelanggan yang melakukan transaksi. Struktur ini memisahkan data identitas dari aktivitas transaksi agar sistem mudah diperluas ke analisis pelanggan atau promosi di masa depan.

Penjelasan Atribut

- **CustomerID** → *Primary Key*. Identitas unik setiap pelanggan.
- **FullName**, **Username**, **ContactNumber** → Informasi kontak pelanggan.
- **City**, **PostCode**, **Street** → Rincian alamat pelanggan untuk mendukung pengiriman atau analisis wilayah.

Constraint

- PRIMARY KEY (CustomerID) menjaga keunikan pelanggan.
- Tidak ada *foreign key* karena tabel ini berdiri sendiri dan menjadi referensi bagi tabel **Order**.

3. Tabel Order

```SQL
CREATE TABLE Order (
  OrderID INT NOT NULL,
  OrderDate DATE NOT NULL,
  TotalAmount DECIMAL(12, 2) NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (OrderID),
```

```
  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

Fungsi dan Peran

Mewakili transaksi pesanan pelanggan. Setiap baris di tabel ini menggambarkan satu transaksi yang bisa terdiri dari beberapa produk berbeda (melalui tabel Order_Details).

Penjelasan Atribut

- **OrderID** → *Primary Key*. Identitas unik tiap pesanan.
- **OrderDate** → Tanggal pesanan dilakukan.
- **TotalAmount** → Total harga seluruh item dalam pesanan (dalam rupiah, presisi 2 digit desimal).
- **CustomerID** → *Foreign Key* yang menghubungkan pesanan dengan pelanggan di tabel Customers.

Constraint

- PRIMARY KEY (OrderID) memastikan setiap pesanan unik.
- FOREIGN KEY (CustomerID) menjamin bahwa pesanan hanya bisa dibuat oleh pelanggan yang valid (integritas referensial).

Relasi

Customers (1) — (N) Order
→ Satu pelanggan dapat memiliki banyak pesanan.

4. Tabel Order_Details

```
SQL

CREATE TABLE Order_Details (

  DetailID INT NOT NULL,
```

```
  QuantitySold INT NOT NULL,
  Subtotal DECIMAL(12, 2) NOT NULL,
  OrderID INT NOT NULL,
  ProductID INT NOT NULL,
  PRIMARY KEY (DetailID),
  FOREIGN KEY (OrderID) REFERENCES Order(OrderID),
  FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);
```

Fungsi dan Peran

Menjadi *junction table* antara Order dan Product. Setiap record mewakili satu jenis produk dalam satu pesanan yang menjelaskan berapa jumlah barang yang dijual dan nilai subtotalnya.

Penjelasan Atribut

- **DetailID** → *Primary Key*, identitas unik tiap entri.
- **QuantitySold** → Jumlah produk yang dijual pada pesanan tersebut.
- **Subtotal** → Hasil dari QuantitySold × UnitPrice.
- **OrderID**, **ProductID** → *Foreign Key* yang menghubungkan ke pesanan dan produk terkait.

Constraint

- FOREIGN KEY (OrderID) menjaga agar detail pesanan tidak bisa ada tanpa pesanan utama.
- FOREIGN KEY (ProductID) menjaga agar produk yang dicatat di detail hanya bisa berasal dari daftar produk yang valid.

Relasi

- Order (1) — (N) Order_Details

- Product (1) — (N) Order_Details

Tabel ini menyelesaikan relasi *many-to-many* antara produk dan pesanan dengan cara yang sesuai prinsip relasional.

5. Tabel Inventory

```SQL
CREATE TABLE Inventory (
  InventoryID INT NOT NULL,
  Quantity INT NOT NULL,
  LastUpdated DATETIME NOT NULL,
  ProductID INT NOT NULL,
  PRIMARY KEY (InventoryID),
  FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
);
```

Fungsi dan Peran

Tabel Inventory berfungsi memantau jumlah stok yang tersedia untuk setiap produk dan waktu pembaruan terakhir. Diperlukan untuk menjaga sinkronisasi antara transaksi penjualan dengan ketersediaan barang di gudang.

Penjelasan Atribut

- **InventoryID** → *Primary Key*, identitas unik tiap catatan stok.
- **Quantity** → Jumlah barang yang masih tersedia.
- **LastUpdated** → Waktu terakhir stok diperbarui (menggunakan tipe DATETIME).
- **ProductID** → *Foreign Key* yang mereferensikan produk di tabel Product.

Constraint

- PRIMARY KEY (InventoryID) menjamin keunikan stok per record.
- FOREIGN KEY (ProductID) memastikan setiap catatan inventori hanya mengacu pada produk yang sah.

Relasi

Product (1) — (1) Inventory
→ Setiap produk memiliki satu record stok yang terhubung langsung dengannya.

6. Integritas Data dan Validitas Desain

Skema ini menerapkan tiga lapisan integritas 3NF:

Jenis Integritas	Implementasi	Tujuan
Entity Integrity	PRIMARY KEY di setiap tabel	Menjamin setiap entitas unik dan tidak null
Referential Integrity	FOREIGN KEY antar tabel	Menjaga konsistensi relasi (misalnya order tidak mungkin tanpa customer)
Domain Integrity	Tipe data yang tepat (VARCHAR, DECIMAL, DATE, DATETIME, INT)	Menjamin data sesuai dengan domain nilainya