



Basis Data

2025

SISTEM BASIS DATA UNTUK OLSHOP "QUICKMART"

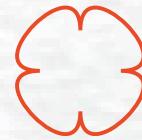
Disusun Oleh:

Anisa Intania Putri - 2024071010

Karine Olivia Permana - 2024071025



Deskripsi Sistem



Sistem yang dibuat adalah sistem basis data untuk mendukung operasional online shop (QUICKMART) yang tujuannya untuk mengelola katalog dan stok produk, mencatat pesanan, mencatat status order, dll.

Pengguna Sistem:

- Pelanggan (yang melakukan pembelian)
- Penjual (yang mengelola produk, pesanan, dan sistem).

Proses Bisnis: Pelanggan memesan produk dan jumlah produknya, Sistem mencatat pesanan, Pelanggan melakukan pembayaran, Status pembayaran “selesai”, Status pesanan diperbarui “proses”, Stok berkurang otomatis.

Identifikasi Entitas dan Atribut

Entitas:

- User
- Kategori
- Produk
- Pesanan
- Pembayaran

Tabel 1: User

Field Name	Data Type	Constraints
User_id	INT	PRIMARY KEY, AUTO_INCREMENT
Nama	VARCHAR(100)	NOT NULL
Email	VARCHAR(100)	UNIQUE, NOT NULL
Role	ENUM('Penjual','Pembeli')	NOT NULL

Tabel 2 : Kategori

Field Name	Data Type	Constraints
Kategori_id	INT	PRIMARY KEY, AUTO_INCREMENT
Nama_kategori	VARCHAR(100)	NOT NULL
Deskripsi	TEXT	NULL
Tanggal_input	DATETIME	DEFAULT CURRENT_TIMESTAMP

Identifikasi Entitas dan Atribut

Tabel 4: Pesanan

Field Name	Data Type	Constraints
Pesanan_id	INT	PRIMARY KEY, AUTO_INCREMENT
User_id	INT	FOREIGN KEY, NOT NULL
Produk_id	INT	FOREIGN KEY, NOT NULL
Jumlah_barang	INT	NOT NULL
Total_harga	DECIMAL (12,2)	NOT NULL
Status	ENUM('Menunggu', 'Diproses', 'Dikirim', 'Selesai', 'Dibatalkan')	DEFAULT 'Menunggu'
Tanggal_pesanan	DATETIME	DEFAULT CURRENT_TIMESTAMP

Tabel 3: Produk

Field Name	Data Type	Constraints
Produk_id	INT	PRIMARY KEY, AUTO_INCREMENT
User_id	INT	FOREIGN KEY, NOT NULL
Kategori_id	INT	FOREIGN KEY, NOT NULL
Nama_produk	VARCHAR(150)	NOT NULL
Harga	DECIMAL (12,2)	NOT NULL
Stok	INT	NOT NULL
Tanggal_upload	DATETIME	DEFAULT CURRENT_TIMESTAMP

Identifikasi Entitas dan Atribut

Tabel 5 : Pembayaran

Field Name	Data Type	Constraints
Pembayaran_id	INT	PRIMARY KEY, AUTO_INCREMENT
Pesanan_id	INT	FOREIGN KEY
Metode	ENUM('Transfer Bank', 'E-Wallet', 'COD')	NOT NULL
Harga	DECIMAL (12,2)	NOT NULL
Status	ENUM('Menunggu', 'Berhasil', 'Gagal')	DEFAULT 'Menunggu'
TanggalBayar	DATETIME	DEFAULT CURRENT_TIMESTAMP



Kardinalitas Relasi



- **User → Pesanan = 1 : N**

Satu User dapat membuat Banyak Pesanan.

- **Kategori → Produk = 1 : N**

Satu Kategori (misalnya, Elektronik) dapat memuat Banyak Produk.

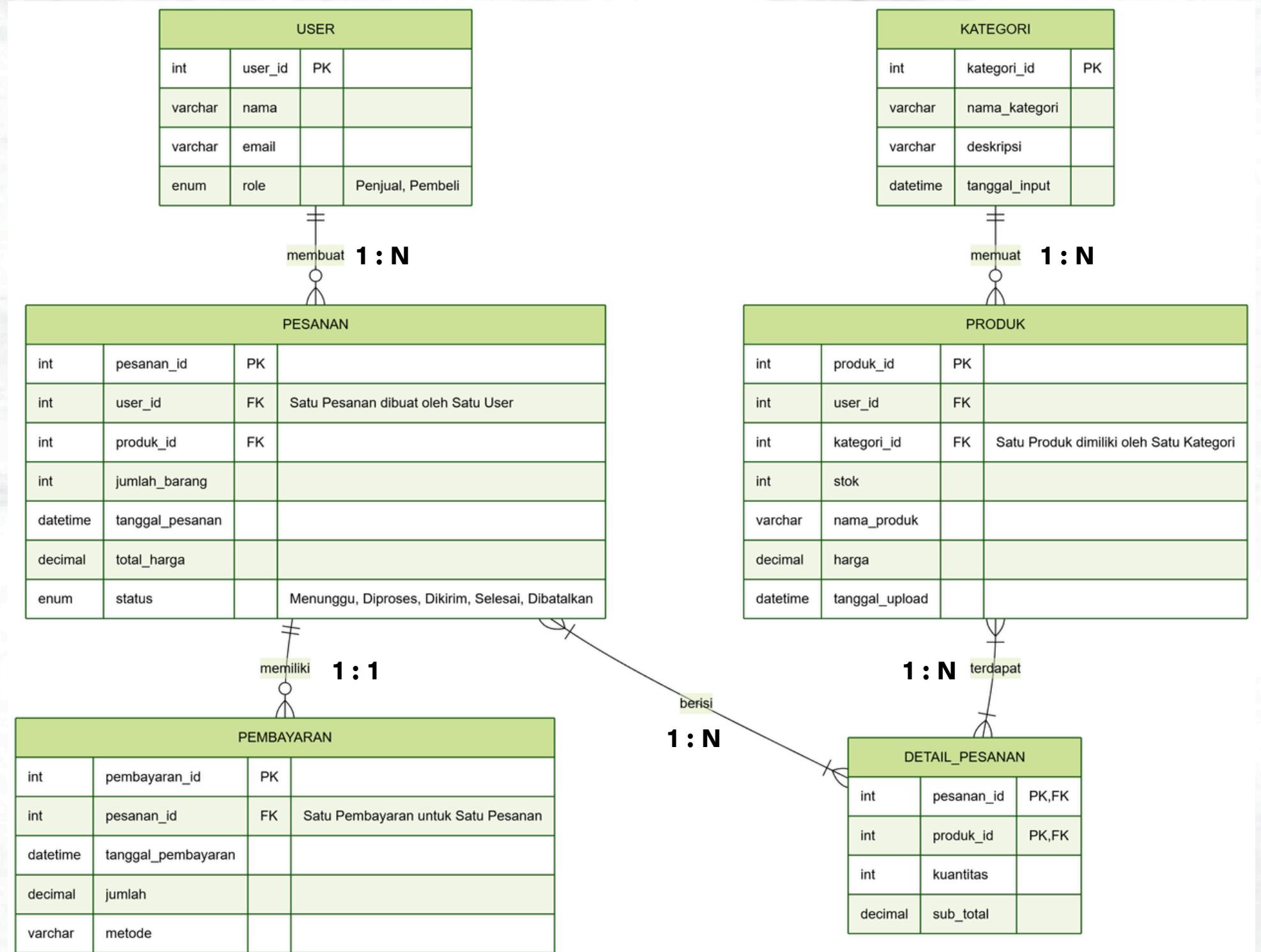
- **Produk → Pesanan = N : M**

1 produk bisa muncul di banyak pesanan dan 1 pesanan punya banyak produk

- **Pesanan → Pembayaran = 1 : 1**

Satu Pesanan idealnya memiliki Satu Pembayaran yang berhasil.

Entity Relationship Diagram (ERD)



1. User → Pesanan = 1 : N

Satu pengguna bisa membuat banyak pesanan.

2. Kategori → Produk = 1 : N

Satu kategori bisa punya banyak produk.

3. Pesanan → Detail_Pesanan = 1 : N

Satu pesanan punya banyak item (detail pesanan).

4. Produk → Detail_Pesanan = 1 : N

Satu produk bisa muncul di banyak detail pesanan (karena dibeli banyak orang).

5. Pesanan → Pembayaran = 1 : 1

Satu pesanan punya satu pembayaran final (idealnya).

Normalisasi

Sebelum Normalisasi

Tabel 4: Pesanan

Field Name	Data Type	Constraints
Pesanan_id	INT	PRIMARY KEY, AUTO_INCREMENT
User_id	INT	FOREIGN KEY, NOT NULL
Produk_id	INT	FOREIGN KEY, NOT NULL
Jumlah_barang	INT	NOT NULL
Total_harga	DECIMAL (12,2)	NOT NULL
Status	ENUM('Menunggu', 'Diproses', 'Dikirim', 'Selesai', 'Dibatalkan')	DEFAULT 'Menunggu'
Tanggal_pesanan	DATETIME	DEFAULT CURRENT_TIMESTAMP

Alasan Normalisasi

Tabel Pesanan awal menyimpan data pesanan dan data produk sekaligus dalam satu tabel. Ini menimbulkan masalah karena satu pesanan bisa berisi banyak produk, tetapi tabel tetap memakai satu kunci (Pesanan_id). Akibatnya, data pesanan harus diulang setiap kali produk baru ditambahkan dalam pesanan yang sama. Tabel ini melanggar 1NF, karena satu pesanan bisa memiliki banyak item, tapi masih disimpan dalam satu tabel. Seharusnya, data pesanan dan data item dipisahkan.

Tabel ini juga melanggar 2NF yaitu menyebabkan redundansi dan bisa menimbulkan update anomaly, misalnya jika status pesanan berubah, kita harus mengupdate banyak baris sekaligus.

Dengan menormalkan tabel menjadi dua bagian, yaitu tabel Pesanan (sebagai header) dan tabel Detail_Pesanan (penyimpanan item pesanan), data menjadi lebih terstruktur, bebas duplikasi, dan mengikuti kaidah 3NF, sehingga lebih konsisten dan stabil untuk digunakan.

Normalisasi

Sesudah Normalisasi

Tabel 4: Pesanan

Field Name	Data Type	Constraints
Pesanan_id	INT	PRIMARY KEY, AUTO_INCREMENT
User_id	INT	FOREIGN KEY, NOT NULL
Status	ENUM('Menunggu', 'Diproses', 'Dikirim', 'Selesai', 'Dibatalkan')	DEFAULT 'Menunggu'
Tanggal_pesanan	DATETIME	DEFAULT CURRENT_TIMESTAMP

Tabel 6: Detail Pesanan

Field Name	Data Type	Constraints
Detail_id	INT	PRIMARY KEY, AUTO_INCREMENT
Pesanan_id	INT	FOREIGN KEY, NOT NULL
Produk_id	INT	FOREIGN KEY, NOT NULL
Jumlah_barang	INT	NOT NULL
Harga_satuan	INT	NOT NULL
Total_harga	DECIMAL (12,2)	NOT NULL



Preview Query



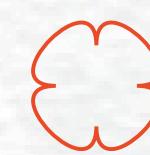
```
SELECT p.Nama_produk, p.Harga, p.Stok, k>Nama_kategori  
FROM Produk p  
JOIN Kategori k ON p.Kategori_id = k.Kategori_id  
WHERE k>Nama_kategori = 'Elektronik';
```

Nama_produk	Harga	Stok	Nama_kategori
Laptop ASUS VivoBook	8500000.00	5	Elektronik
Smartphone Samsung A14	3200000.00	10	Elektronik

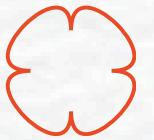
Relasi tabel Produk → Kategori

Query ini menampilkan data produk dan kategorinya.

Digunakan untuk mencari produk mana saja yang termasuk kategori tertentu.



Kesulitan Dan Pembelajaran



Kesulitan:

- Menangani relasi N : M (Many-to-Many) tabel Pesanan dan Produk
- Normalisasi table Pesanan sampai dipecah menjadi table Pesanan dan Detail Pesanan

Pembelajaran:

- Memahami berbagai jenis relasi ERD (1:1, 1:N, N:M)
- Memahami normalisasi untuk menghindari data ganda
- Merancang database yang lebih realistik



Basis Data

2025

Terima Kasih!

Disusun Oleh:

Anisa Intania Putri - 2024071010

Karine Olivia Permana - 2024071025

