

PERANCANGAN DAN IMPLEMENTASI BASIS DATA SISTEM PEMESANAN HUNGRY!

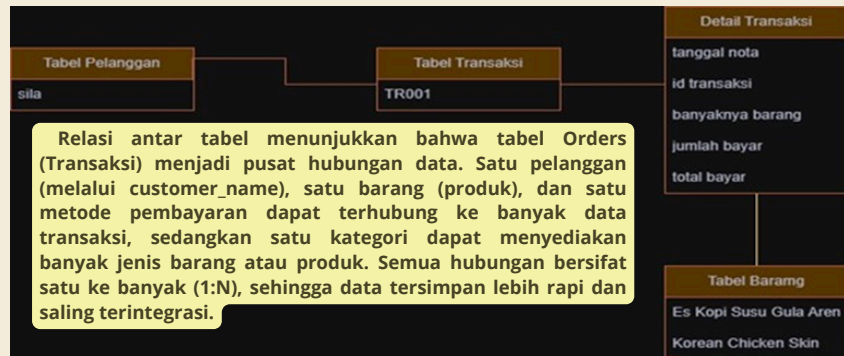
Anggota: Khotijah Naishilla Ariyanto (240103194)| Osama Habib Candranata (240103199)| Samuel Rinaldy (240103202)

1. LATAR BELAKANG

Perkembangan teknologi informasi telah mendorong berbagai sektor usaha kuliner untuk beralih dari sistem manual menuju sistem digital. Sistem penjualan konvensional sering menimbulkan permasalahan seperti duplikasi data, kesalahan perhitungan, dan sulitnya pelacakan stok. Oleh karena itu, diperlukan perancangan basis data relasional yang mampu mengelola integritas data tersebut serta mendukung fitur otomatisasi seperti pengurangan stok otomatis melalui trigger.

2. RELASI ANTAR TABEL

relasi



1. Merancang ERD (Entity Relationship Diagram) untuk sistem restoran.
2. Melakukan normalisasi data dari nota transaksi mentah hingga bentuk 3NF.
3. Mengimplementasikan skema database db_nota_resto beserta Stored Procedure dan Trigger untuk otomasi.
4. Menghasilkan informasi struk belanja yang akurat melalui query SQL.

Ringkasan Masalah

Masalah utama adalah bagaimana membangun basis data transaksi restoran yang terstruktur, terintegrasi, dan mampu mengolah serta menampilkan data transaksi secara akurat menggunakan SQL

PK digunakan untuk membedakan satu record dengan record lainnya sehingga tidak terjadi data ganda.

Foreign Key berfungsi untuk menjaga keterkaitan dan konsistensi data antar tabel, sehingga data yang dimasukkan harus sesuai dengan data yang sudah ada pada tabel yang dirujuk.

DDL

DDL – DATA DEFINITION LANGUAGE

```
-- DATABASE
DROP DATABASE IF EXISTS db_nota_resto;
CREATE DATABASE db_nota_resto
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
USE db_nota_resto;
```

```
SELECT
DATE(o.order_datetime) AS tanggal,
COUNT(DISTINCT o.order_id) AS jumlah_nota,
SUM(o.total) AS total_penjualan
FROM orders o
WHERE o.status <> 'CANCELLED'
GROUP BY DATE(o.order_datetime)
ORDER BY tanggal;
```

```
-- 1. QUERY JOIN
-- Menampilkan detail nota lengkap berdasarkan nomor order
```

```
SELECT
o.order_no,
o.order_datetime,
o.order_type,
dt.table_no,
u.full_name AS kasir,
o.customer_name,
p.product_name,
c.category_name,
oi.qty;
```

```
INSERT INTO roles (role_name)
VALUES ('admin'), ('kasir');
```

```
INSERT INTO users (role_id, full_name, username, password_hash)
VALUES
(1, 'Admin Sistem', 'admin', 'hash_admin'),
(2, 'Kasir 1', 'kasir1', 'hash_kasir1'),
(2, 'Kasir 2', 'kasir2', 'hash_kasir2');
```

```
INSERT INTO dining_tables (table_no)
```

```
START TRANSACTION;
```

```
-- Pembuatan order (insert ke orders)
-- Pembuatan item order (insert ke order_items)
```

```
40
```

```
-- Perhitungan subtotal, diskon, pajak, dan total
```

```
-- Seluruh proses dijalankan secara atomik
```

```
COMMIT;
```

```
SELECT
p.product_name,
SUM(oi.qty) AS total_qty_terjual
FROM order_items oi
JOIN orders o ON o.order_id = oi.order_id
JOIN products p ON p.product_id = oi.product_id
WHERE o.status <> 'CANCELLED'
GROUP BY p.product_name
HAVING SUM(oi.qty) > 2
ORDER BY total_qty_terjual DESC;
```

HAVING

BENTUK NORMAL 1NF

Transaksi	Id_Pelanggan	Nama Pelanggan	Id_Barang	Nama Barang	Satuan	Banyak Barang	Jumlah Barang
TR001	P001	SILA	1	Es Kopi Susu Gula Aren	pcs	1	12.900
TR002	P002	SILA	2	Korean Chicken Skin	pcs	1	30.000

Tabel. 7 1NF

Memastikan setiap kolom bernilai atomik (tidak ada pecahan).

BENTUK NORMAL 2NF

ID_PELANGGAN		NAMA PELANGGAN		
P001		SILA		
ID_BARANG	NAMA	JUMLAH BAYAR	SATUAN	
TR001	Es Kopi Susu Gula Aren	12.900	pcs	
TR002	Korean Chicken Skin	30.000	pcs	
ID_TRANSAKSI	QTY	JUMLAH BAYAR	TANGGAL NOTA	TOTAL
TR001	1	12.900	19/10/2025	42.900

Tahap kedua normalisasi bertujuan untuk menghilangkan ketergantungan parsial terhadap kunci utama. Setelah memenuhi 1NF, setiap atribut dalam tabel harus bergantung sepenuhnya pada kunci utama agar tidak terjadi duplikasi atau redundansi data.

BENTUK NORMAL 3NF

id_pelanggan	nama_customer
P001	Silla

id_transaksi	QTY	JUMLAH BAYAR	TANGGAL NOTA	TOTAL
TR01	1	12.900	19/10/2025	42.900

Kode_barang	Nama_barang	Ukuran
TR001	Es Kopi Susu Gula Aren	-
TR002	Korean Chicken Skin	-

Tahap ini bertujuan untuk menghilangkan ketergantungan antar atribut non-key agar data menjadi lebih efisien dan konsisten.

