

# LAPORAN TUGAS

## “SISTEM BASIS DATA”



UNIVERSITAS  
**DUTA BANGSA**  
SURAKARTA

Disusun oleh:

NAMA	: ATHAYA DAFFA (250119004)
	: VYKA AMANUL ALAM (250119024)
	: DANAR IKHSAN SM (250119006)
	: ANDREAN VIERY (210102041)

PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DUTA BANGSA SURAKARTA 2025

## **DAFTAR ISI**

<b>BAB I .....</b>	<b>3</b>
<b>PENDAHULUAN .....</b>	<b>3</b>
<b>1.1 Latar Belakang Masalah.....</b>	<b>3</b>
<b>1.2 Rumusan Masalah dan Batasan .....</b>	<b>3</b>
<b>1.3 Tujuan Penulisan .....</b>	<b>3</b>
<b>BAB II.....</b>	<b>4</b>
<b>PERANCANGAN DAN SKEMA DATA.....</b>	<b>4</b>
<b>2.1 Analisis Kebutuhan Data .....</b>	<b>4</b>
<b>2.2 Penjelasan Struktur Tabel .....</b>	<b>4</b>
<b>2.3 Skema Relasi (Entity Relationship).....</b>	<b>4</b>
<b>BAB III.....</b>	<b>5</b>
<b>IMPLEMENTASI KODE QUERY.....</b>	<b>5</b>
<b>3.1 Data Definition Language (DDL).....</b>	<b>5</b>
<b>3.2 Data Manipulation Language (DML) .....</b>	<b>6</b>
<b>3.3 Penjelasan Logika Insert .....</b>	<b>6</b>
<b>BAB IV .....</b>	<b>7</b>
<b>ANALISIS HASIL QUERY.....</b>	<b>7</b>
<b>4.1 Menampilkan Nota Digital (JOIN).....</b>	<b>7</b>
<b>4.2 Laporan Kinerja Kasir (GROUP BY).....</b>	<b>8</b>
<b>4.3 Manajemen Stok Kritis (HAVING) .....</b>	<b>8</b>
<b>BAB V .....</b>	<b>9</b>
<b>PENUTUP.....</b>	<b>9</b>
<b>5.1 Kesimpulan.....</b>	<b>9</b>
<b>5.2 Hambatan yang Ditemui.....</b>	<b>9</b>
<b>5.3 Saran Pengembangan .....</b>	<b>9</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Di zaman sekarang, belanja di gerai ritel seperti Indomaret sudah menjadi kebutuhan harian. Namun, pernahkah kita terpikir bagaimana ribuan transaksi yang terjadi setiap detik bisa tercatat dengan rapi? Tanpa sistem database yang kuat, kasir akan kesulitan menghitung total belanjaan, stok barang akan berantakan, dan pemilik toko tidak akan tahu berapa keuntungan mereka. Oleh karena itu, simulasi pembuatan sistem nota ini sangat penting untuk memahami dasar dari aplikasi kasir yang nyata.

### **1.2 Rumusan Masalah dan Batasan**

Masalah utama yang ingin diselesaikan adalah bagaimana menyusun data yang saling berhubungan agar informasi transaksi tidak tertukar. Batasan dalam studi kasus ini mencakup pengelolaan data karyawan (kasir), daftar produk (items), serta pencatatan nota transaksi beserta rincian item yang dibeli. Kami tidak membahas sistem pembayaran digital secara mendalam, melainkan fokus pada logika penyimpanan datanya.

### **1.3 Tujuan Penulisan**

Tujuan dari pembuatan makalah ini adalah:

1. Membangun struktur tabel yang rapi dan terhubung (relasional).
2. Mempraktikkan cara memasukkan dan memanipulasi data tanpa terjadi error.
3. Menggali informasi penting dari database, seperti siapa kasir yang paling rajin atau barang apa yang paling cepat habis menggunakan query tingkat lanjut.

# BAB II

## PERANCANGAN DAN SKEMA DATA

### 2.1 Analisis Kebutuhan Data

Sebelum mengetik kode di MySQL Workbench, kita perlu membedah apa saja yang ada di selembar nota Indomaret. Ada nama kasir, tanggal belanja, daftar barang, harga satuan, jumlah barang, diskon, pajak (PPN), dan total bayar. Data ini tidak bisa dimasukkan ke dalam satu tabel besar karena akan menyebabkan data double (redundansi). Maka, kita pecah menjadi empat bagian logis.

### 2.2 Penjelasan Struktur Tabel

Berikut adalah detail dari tabel yang kita buat:

- **Tabel Kasir:** Berisi data pribadi orang yang bekerja di balik meja kasir.
- **Tabel Items:** Berisi katalog produk. Di sini "stok" menjadi kunci penting untuk kontrol gudang.
- **Tabel Transaksi:** Ibarat "kepala nota". Isinya informasi umum seperti tanggal dan total uang.
- **Tabel Detail Transaksi:** Ibarat "isi nota". Isinya rincian barang apa saja yang dibeli di satu nomor nota tertentu.

### 2.3 Skema Relasi (Entity Relationship)

Relasi antartabel dijaga menggunakan *Foreign Key*. Tabel transaksi bergantung pada id\_kasir. Sementara tabel detail\_transaksi adalah jembatan yang menghubungkan nota (id\_transaksi) dengan produknya (id\_items). Tanpa relasi ini, data akan menjadi "yatim piatu" dan tidak bisa dilacak asalnya.

The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code area contains the following SQL statements:

```
4 • CREATE TABLE kasir (
5     id_ Execute the statement under the keyboard cursor
6         VARCHAR(100) NOT NULL,
7         no_telp VARCHAR(15),
8         alamat TEXT
9 );
10
11 • CREATE TABLE items (
12     id_items INT AUTO_INCREMENT PRIMARY KEY,
13     nama_items VARCHAR(100) NOT NULL,
14     harga INT NOT NULL,
15     stok INT NOT NULL
16 );
17
18 • CREATE TABLE transaksi (
19     id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
```

The code is being typed into the editor, with the cursor currently at line 5, column 12, under the 'id\_' keyword. A tooltip 'Execute the statement under the keyboard cursor' is visible above the cursor. The interface includes standard MySQL Workbench icons for file operations, search, and refresh.

## **BAB III**

### **IMPLEMENTASI KODE QUERY**

#### **3.1 Data Definition Language (DDL)**

Tahap pertama adalah membuat "rumah" bagi data kita. Kita mendefinisikan tipe data seperti INT untuk angka dan VARCHAR untuk teks.

```
CREATE DATABASE nota_indomaret;
USE nota_indomaret;

-- Membuat tabel utama
CREATE TABLE kasir (
    id_kasir INT AUTO_INCREMENT PRIMARY KEY,
    nama_kasir VARCHAR(100) NOT NULL,
    no_telp VARCHAR(15),
    alamat TEXT
);

CREATE TABLE items (
    id_items INT AUTO_INCREMENT PRIMARY KEY,
    nama_items VARCHAR(100) NOT NULL,
    harga INT NOT NULL,
    stok INT NOT NULL
);

CREATE TABLE transaksi (
    id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
    id_kasir INT NOT NULL,
```

```

tanggal DATETIME NOT NULL,
disc INT DEFAULT 0,
ppn INT DEFAULT 0,
total INT NOT NULL,
FOREIGN KEY (id_kasir) REFERENCES kasir(id_kasir)
);

CREATE TABLE detail_transaksi (
id_detail INT AUTO_INCREMENT PRIMARY KEY,
id_transaksi INT NOT NULL,
id_items INT NOT NULL,
kuantitas INT NOT NULL,
subtotal INT NOT NULL,
FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi),
FOREIGN KEY (id_items) REFERENCES items(id_items)
);

```

### 3.2 Data Manipulation Language (DML)

Setelah tabel jadi, kita isi dengan data dummy agar sistem bisa diuji coba.

```

-- Memasukkan data contoh
INSERT INTO kasir (nama_kasir, no_telp, alamat) VALUES
('Abdi', '081234567890', 'Jl. semen romo No.1'),
('damar', '081298765432', 'Jl. raden said No.10'),
('dinda', '081377788899', 'Jl. Ahmad Yani No.5');

INSERT INTO items (nama_items, harga, stok) VALUES
('Indomie Goreng', 3500, 100),
('Aqua 600ml', 4000, 80),
('Chitato', 7500, 5); -- Contoh stok tipis

```

### 3.3 Penjelasan Logika Insert

Proses input data harus berurutan. Kita tidak bisa membuat data di detail\_transaksi jika id\_transaksi-nya belum dibuat. Begitu juga transaksi tidak bisa dibuat kalau id\_kasir-nya tidak terdaftar. Ini yang disebut dengan menjaga integritas data.

```
17
18 • CREATE TABLE transaksi (
19     id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
20     id_kasir INT NOT NULL,
21     tanggal DATETIME NOT NULL,
22     disc INT DEFAULT 0,
23     ppn INT DEFAULT 0,
24     total INT NOT NULL,
25     FOREIGN KEY (id_kasir) REFERENCES kasir(id_kasir)
26 );
27
28 • CREATE TABLE detail_transaksi (
29     id_detail INT AUTO_INCREMENT PRIMARY KEY,
30     id_transaksi INT NOT NULL,
31     id_items INT NOT NULL,
32     kuantitas INT NOT NULL,
```

## BAB IV

## ANALISIS HASIL QUERY

### 4.1 Menampilkan Nota Digital (JOIN)

Query ini bertujuan untuk menggabungkan potongan-potongan data dari berbagai tabel menjadi satu tampilan nota yang utuh.

SELECT

```
t.id_transaksi AS 'No Nota',
k.nama_kasir AS 'Kasir',
i.nama_items AS 'Produk',
d.kuantitas AS 'Jumlah',
```

```

d.subtotal AS 'Subtotal',
t.tanggal AS 'Waktu'
FROM detail_transaksi d
JOIN transaksi t ON d.id_transaksi = t.id_transaksi
JOIN kasir k ON t.id_kasir = k.id_kasir
JOIN items i ON d.id_items = i.id_items;

```

## 4.2 Laporan Kinerja Kasir (GROUP BY)

Dengan query ini, pemilik toko bisa melihat siapa kasir yang paling banyak menghasilkan omzet. Ini sangat berguna untuk pemberian bonus atau evaluasi kerja.

```

SELECT
k.nama_kasir,
SUM(t.total) AS total_penjualan
FROM transaksi t
JOIN kasir k ON t.id_kasir = k.id_kasir
GROUP BY k.nama_kasir;

```

## 4.3 Manajemen Stok Kritis (HAVING)

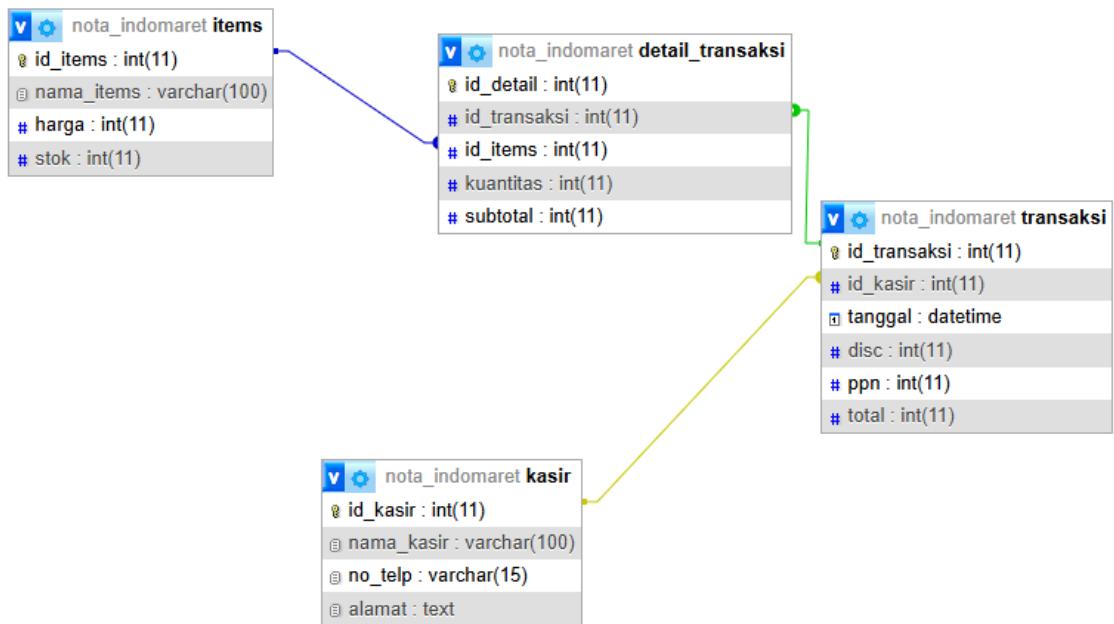
Toko tidak boleh kehabisan barang. Query ini bertindak sebagai alarm untuk memberi tahu barang apa saja yang stoknya di bawah batas aman (misal: di bawah 20 pcs).

```

SELECT nama_items, stok
FROM items
GROUP BY id_items, nama_items
HAVING stok < 20;

```

# Screenshot Skema Database



## **BAB V**

## **PENUTUP**

### **5.1 Kesimpulan**

Melalui tugas ini, kita belajar bahwa database bukan sekadar kumpulan tabel, tapi sebuah sistem yang saling mengunci. Penggunaan MySQL Workbench mempermudah kita dalam merancang nota Indomaret yang rapi, mulai dari urusan input data hingga penarikan laporan penjualan yang kompleks melalui fitur Join dan Grouping.

### **5.2 Hambatan yang Ditemui**

Beberapa kendala yang biasanya muncul adalah kesalahan urutan saat menghapus data (*Foreign Key constraint*) atau kesalahan pengetikan nama kolom. Namun, dengan pemahaman relasi yang kuat, masalah ini bisa diatasi dengan melakukan pengecekan pada skema ERD.

### **5.3 Saran Pengembangan**

Untuk ke depannya, sistem ini bisa ditambahkan fitur "Trigger". Jadi, setiap kali ada transaksi, stok di tabel items akan otomatis berkurang tanpa perlu kita update manual. Selain itu, penambahan tabel diskon member akan membuat sistem ini semakin mirip dengan yang ada di toko aslinya.