

**KETENTUAN PENGUMPULAN UJIAN AKHIR  
SEMESTER SISTEM BASIS DATA PENJUALAN KASIR  
MINIMARKET INDOMARET**



**UNIVERSITAS  
DUTA BANGSA  
SURAKARTA**

Dosen Pengampu:

Ridwan Dwi Irawan,S.Kom,M.Kom

Dibuat oleh :

DIVA ANGGARA

240103159

FALIH NABIL MALIK

240103161

**PROGRAM STUDI TEKNIK INFORMATIKA  
UNIVERSITAS DUTA BANGSA  
SURAKARTA  
2025/2026**

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan proyek mata kuliah Pemrograman Basis Data dengan judul **“Perancangan Sistem Basis Data Transaksi Kasir pada Minimarket Indomaret”** ini dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat Ujian Tengah Semester mata kuliah Pemrograman Basis Data di Program Studi S1 Teknik Informatika, Fakultas Ilmu Komputer, Universitas Duta Bangsa Surakarta. Proyek ini bertujuan untuk merancang sebuah sistem basis data yang terstruktur dan terintegrasi guna mendukung proses transaksi penjualan di Minimarket Indomaret.

Dalam penyusunan laporan ini, penulis mendapatkan bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak **Ridwan Dwi Irawan, S.Kom., M.Kom**, selaku dosen pengampu mata kuliah Pemrograman Basis Data.
2. Manajemen Indomaret sebagai referensi studi kasus sistem transaksi minimarket.
3. Rekan-rekan mahasiswa, khususnya anggota kelompok, yang telah bekerja sama dalam penyelesaian proyek ini.
4. Keluarga yang selalu memberikan semangat, doa, dan dukungan penuh.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini bermanfaat bagi pembaca dan dapat menjadi referensi pengembangan sistem basis data di bidang ritel.

Surakarta, 19 Januari 2026

Hormat Kami,  
Tim Penyusun

## DAFTAR ISI

HALAMAN JUDUL.....	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang.....	4
1.2 Rumusan Masalah.....	5
1.3 Tujuan.....	5
1.4 Manfaat Sistem.....	5
BAB II LANDASAN TEORI.....	7
2.1 Teori Dasar.....	7
2.1.1 Konsep Basis Data dan DBMS.....	7
2.1.2 Teori ERD (Entity Relationship Diagram).....	7
2.1.3 Teori Normalisasi.....	7
2.1.4 Dasar SQL (DDL, DML, TCL).....	7
2.2 Studi Kasus (Nota Transaksi).....	8
2.2.1 Skenario Sistem).....	8
2.2.2 Tabel Umum (Unnormalized Table / Tabel Awal).....	9
2.2.3 Normalisasi 1NF (First Normal Form).....	9
2.2.4 Normalisasi 2NF (Second Normal Form).....	10
2.2.5 Normalisasi 3NF (Third Normal Form).....	11
2.3 Desain Akhir dan Relasi.....	11
2.3.1 dan 2.3.2 Kamus Data dan Kardinalitas.....	15
2.3.3 ERD Final (Entity Relationship Diagram).....	17
BAB III IMPLEMENTASI BASIS DATA.....	19
3.1 Implementasi DDL (Pembuatan Tabel).....	19
3.2 Implementasi DML (Pengisian Data).....	21
3.3 Implementasi TCL (Skenario Transaksi).....	22
3.4 Query dan Pelaporan Data.....	23
3.4.1 Query Join (Struk Belanja).....	23
3.4.2 Query Aggregation (Laporan Omset).....	24
3.4.3 Subquery dan View.....	24
3.5 Fitur Lanjutan (Trigger Update Stok).....	25
BAB IV PENUTUP.....	26
4.1 Kesimpulan.....	26
4.2 Saran Pengembangan.....	27
DAFTAR PUSTAKA.....	28
LAMPIRAN.....	29
Lampiran 1: Link Repository GitHub.....	29
Lampiran 2: Dokumentasi / Screenshot.....	30

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar belakang**

Dalam era digital saat ini, perkembangan teknologi informasi telah memberikan dampak besar terhadap berbagai bidang, termasuk sektor usaha ritel modern seperti minimarket. Proses pengelolaan transaksi yang semula dilakukan secara manual kini mulai beralih ke sistem terkomputerisasi agar lebih efisien, cepat, dan akurat. Salah satu aspek penting dalam pengelolaan usaha ritel adalah sistem pencatatan transaksi penjualan yang dilakukan di bagian kasir.

Minimarket Indomaret merupakan salah satu usaha ritel modern yang melayani penjualan berbagai jenis barang dan jasa secara langsung kepada pelanggan. Dalam studi kasus ini, proses pencatatan transaksi penjualan menjadi fokus utama. Sistem kasir yang ada sudah terkomputerisasi, namun data dari nota transaksi akan dijadikan dasar untuk perancangan basis data yang optimal. Proses pencatatan transaksi secara manual, yang menjadi fokus studi kasus ini, sering menimbulkan berbagai kendala, seperti kesalahan pencatatan, kesulitan dalam pelacakan data transaksi, dan keterlambatan dalam proses pembayaran.

Untuk mengatasi permasalahan tersebut, diperlukan sebuah Sistem Basis Data Penjualan Kasir (Point of Sale) yang mampu mengelola data transaksi secara terintegrasi dan otomatis. Sistem ini dirancang untuk membantu kasir dalam mencatat setiap transaksi penjualan mulai dari pemindaian barang, penginputan data, perhitungan total pembayaran, hingga pembuatan nota transaksi sebagai bukti pembayaran. Dengan adanya sistem ini, diharapkan proses bisnis di Minimarket Indomaret dapat berjalan lebih efektif, efisien, serta mengurangi risiko kesalahan dalam pencatatan transaksi. Selain itu, data yang dihasilkan dari setiap nota transaksi akan menjadi sumber utama dalam proses normalisasi dan perancangan basis data, sehingga struktur penyimpanan data menjadi lebih teratur dan mudah dikelola. Sistem ini tidak hanya bermanfaat dalam proses operasional harian, tetapi juga membantu pemilik usaha dalam menganalisis penjualan dan mengambil keputusan yang lebih tepat berdasarkan data yang tersimpan.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana merancang sistem basis data yang dapat mencatat dan mengelola transaksi penjualan di Minimarket Indomaret secara efisien?
2. Bagaimana proses normalisasi data dilakukan agar struktur basis data lebih optimal dan mudah digunakan?
3. Bagaimana sistem dapat menghasilkan nota transaksi secara otomatis sebagai bukti pembayaran pelanggan?
4. Bagaimana sistem ini dapat membantu kasir dalam mempercepat proses pelayanan dan mengurangi kesalahan pencatatan transaksi?

## **1.3 Tujuan**

Tujuan dari perancangan Sistem Basis Data Penjualan Kasir ini antara lain adalah:

1. Membuat sistem yang mampu mengelola data pelanggan, data barang/produk, dan data transaksi secara terintegrasi.
2. Meningkatkan efisiensi dan akurasi dalam proses pencatatan transaksi penjualan.
3. Menghasilkan nota transaksi otomatis yang dapat digunakan sebagai bukti pembayaran pelanggan.
4. Membangun struktur basis data yang terorganisasi melalui proses normalisasi agar mudah dikembangkan di masa mendatang.
5. Membantu pemilik Minimarket Indomaret dalam memantau laporan penjualan dan menganalisis data untuk

## **1.4 Manfaat Sistem**

Dengan diterapkannya sistem ini, diharapkan dapat memberikan beberapa manfaat, di antaranya:

1. Mempermudah kasir dalam melakukan transaksi dan pencatatan penjualan.
2. Mengurangi kesalahan dalam penginputan data dan perhitungan total pembayaran.
3. Menyediakan data transaksi yang tersimpan secara rapi dan aman dalam basis data.
4. Mempercepat pelayanan kepada pelanggan dan meningkatkan kepuasan pelanggan.

5. Menyediakan informasi laporan penjualan yang berguna bagi pemilik dalam pengambilan keputusan bisnis.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Teori Dasar**

##### **2.1.1 Konsep Basis Data dan DBMS**

Basis data adalah kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari sistem komputer. Untuk mengelola basis data, diperlukan Database Management System (DBMS). DBMS adalah perangkat lunak yang berinteraksi dengan pengguna akhir, aplikasi, dan basis data itu sendiri untuk menangkap dan menganalisis data. Dalam proyek ini, DBMS yang digunakan adalah MySQL dengan antarmuka MySQL Workbench.

##### **2.1.2 Entity Relationship Diagram (ERD)**

ERD adalah diagram yang digunakan untuk merancang struktur basis data dengan menggambarkan entitas-entitas yang terlibat beserta hubungan (relasi) antar entitas tersebut. Komponen utama ERD meliputi:

- Entitas: Objek dunia nyata yang dapat dibedakan (contoh: Pelanggan, Barang).
- Atribut: Karakteristik dari entitas (contoh: Nama, Harga).
- Primary Key (PK): Atribut unik yang membedakan setiap rekaman.
- Foreign Key (FK): Atribut yang menghubungkan satu tabel dengan tabel lain.

##### **2.1.3 Normalisasi**

Normalisasi adalah proses pengelompokan atribut data yang membentuk entitas sederhana, non-redundant, fleksibel, dan mudah beradaptasi. Tahapan normalisasi meliputi:

1. 1NF (First Normal Form): Menghilangkan grup berulang, memastikan setiap kolom bernilai atomik.
2. 2NF (Second Normal Form): Memastikan tidak ada ketergantungan parsial (atribut non-kunci harus bergantung penuh pada Primary Key).
3. 3NF (Third Normal Form): Memastikan tidak ada ketergantungan transitif (atribut non-kunci tidak boleh bergantung pada atribut non-kunci lainnya).

##### **2.1.4 Dasar SQL (Structured Query Language)**

SQL adalah bahasa standar untuk mengakses dan memanipulasi basis data. SQL dibagi menjadi beberapa kategori:

- DDL (Data Definition Language): Perintah untuk mendefinisikan struktur database (contoh: CREATE, DROP).
- DML (Data Manipulation Language): Perintah untuk mengelola data (contoh: INSERT, UPDATE, SELECT).
- TCL (Transaction Control Language): Perintah untuk mengelola transaksi (contoh: COMMIT, ROLLBACK).

## 2.2 Perancangan Basis Data Transaksi Kasir pada Minimarket Indomaret

### 2.2.1 Skenario Sistem

Studi kasus ini mengambil proses bisnis di "Minimarket Indomaret", sebuah usaha ritel modern yang melayani penjualan barang dan jasa. Skenario sistem yang ingin dibuat adalah "Sistem Basis Data Penjualan Kasir (Point of Sale)". Sistem ini bertujuan untuk mengelola dan mencatat setiap transaksi penjualan yang terjadi di minimarket tersebut.

Alur skenario adalah sebagai berikut:

1. Seorang Pelanggan (misalnya 'Rani') datang dan membawa barang yang ingin dibeli.
2. Kasir (Admin) mengidentifikasi pelanggan (jika memiliki kartu member atau untuk layanan khusus) dan membuka Transaksi baru.
3. Pelanggan membeli beberapa Barang (item produk).
4. Kasir memasukkan setiap item pesanan ke dalam sistem, yang akan dicatat dalam Detail Transaksi.
5. Sistem menghitung total pembayaran.
6. Setelah pembayaran (misalnya via 'Tunai'), sistem akan menghasilkan Nota sebagai bukti transaksi. Data dari nota inilah yang akan dijadikan dasar (Tabel Umum/Unnormalized) untuk proses normalisasi dan perancangan basis data.

Berikut adalah contoh data transaksi yang diambil dari satu nota penjualan di Minimarket Indomaret, yang menjadi dasar studi kasus ini:





### 2.2.2 Tabel Umum (Unnormalized Table / Tabel Awal)

Tabel ini adalah Tabel Umum (Unnormalized Table / Tabel Awal) yang mewakili data mentah dari satu nota transaksi penjualan di Indomaret. Tabel ini mengumpulkan semua informasi dari nota, yang akan menjadi dasar untuk proses normalisasi basis data.

Studi Kasus Warung Makan SE/Tabel Umum								
No Nota	Waktu	Jenis_order	Nama_order	Jumlah_Barang	Nama_Barang	Harga_Barang	Total_Bayar	Transfer
CS/11/510/010	13.10.2025	direct purchase	diva	1	Nivea Drk	43500	46500	46500
CS/11/510/010	13.10.2025	direct purchase	diva	1	Le Mineral	3000	46500	46500

Tabel ini adalah Tabel Umum (Unnormalized Table / Tabel Awal) yang mewakili data mentah dari satu nota transaksi penjualan di Indomaret. Tabel ini mencakup detail kepala transaksi seperti No Nota, Waktu, dan Nama\_Kasir, serta detail item yang dibeli. Meskipun semua data dikumpulkan, tabel ini bersifat unnormalized karena data kepala transaksi (No\_Nota, Waktu, Total\_Belanja, Pembayaran\_Tunai, dll.) terulang di setiap baris (jika ada lebih dari satu item yang dibeli), yang menunjukkan adanya redundansi..

### 2.2.3 Normalisasi 1NF (First Normal Form)

Tabel ini menangkap informasi lengkap mengenai sebuah transaksi. Karena setiap kolom sudah berisi nilai tunggal dan tidak ada grup berulang, tabel ini sudah memenuhi syarat 1NF. Kunci Utama Komposit (Composite Primary Key) ditetapkan sebagai (No\_Nota, Id\_Barang).

Tabel_Normalis									
Id_barang	No_Nota	Waktu	Jenis_order	Nama_order	Nama_Barang	Jumlah_Barang	Harga_Barang	Total_Bay	Transfer
BRG01	CS/11/510/010	13.10.2025	direct purchase	diva	Nivea Drk	1	43500	46500	46500
BRG02	CS/11/510/010	13.10.2025	direct purchase	diva	Le Mineral	1	3000	46500	46500

Tabel di atas sudah memenuhi 1NF, namun dapat diamati adanya redundansi data atau pengulangan informasi pada beberapa kolom. Data seperti No\_Nota, Waktu, Nama\_Kasir, Nama\_Pelanggan, dan Total\_Belanja memiliki nilai yang sama untuk setiap baris. Redundansi ini mengindikasikan adanya ketergantungan parsial (sebagian atribut non-kunci hanya bergantung pada sebagian kunci utama), yang harus dihilangkan pada normalisasi 2NF.

## 2.2.4 Normalisasi 2NF (Second Normal Form)

### A. Tabel Barang

Table\_Barang(2NF)

Table_Barang(2NF)		
Id_barang (PK)	Nama_Barang	Harga_Barang
BRG01	Nivea Drk	43500
BRG02	Le Mineral	3000

Tabel Barang di atas berfungsi untuk menyimpan data master atau referensi dari setiap item produk yang dijual. Dalam tabel ini, Id\_Barang bertindak sebagai Primary Key (PK) yang mengidentifikasi setiap barang secara unik. Atribut lainnya sepenuhnya bergantung pada kunci utama ini, sehingga telah memenuhi syarat 2NF dengan menghilangkan ketergantungan parsial.

### B. Tabel Detail Transaksi

Tabel\_Detail\_Transaksi(2NF)

Tabel_Transaksi(3NF)				
No_Nota (PK)	Waktu	Total_Bayar	Transfer	ID_PELANGGAN (FK)
CS/11/510/010	13.10.2025	46500	46500	PLG06

Tabel Detail Transaksi berfungsi sebagai tabel penghubung (bridge table) untuk merekam setiap item barang yang dipesan dalam suatu transaksi. Tabel ini memiliki kunci utama komposit yang terdiri dari No\_Nota dan Id\_Barang, di mana keduanya juga berperan sebagai Foreign Key (FK). Struktur ini berhasil memecahkan hubungan many-to-many antara entitas Transaksi dan Barang.pemesanannya, sehingga hubungan many-to-many antara entitas transaksi dan

## 2.2.5 Normalisasi 3NF (Third Normal Form)

### A. Tabel Pelanggan

Tabel\_Pelanggan(3NF)

Tabel_Pelanggan(3NF)		
ID_PELANGGAN (PK)	Nama_pelanggan	Jenis_Pelanggan
PLG06	diva	direct purchase

Keterangan Tabel Pelanggan: Tabel ini berfungsi sebagai Tabel Master yang menyimpan informasi dasar tentang setiap pelanggan yang melakukan transaksi di Indomaret. Pemisahan data pelanggan (Nama, Jenis) dari tabel transaksi utama (seperti yang dilakukan pada 3NF) bertujuan untuk menghilangkan ketergantungan transitif dan redundansi (pengulangan data). Jika seorang pelanggan melakukan 100 transaksi, datanya (nama dan jenis) hanya perlu disimpan satu kali di tabel ini, dan di tabel transaksi cukup diwakili oleh ID\_PELANGGAN sebagai Foreign Key.

## 2.3 Desain Akhir dan Relasi

### 2.3.1 Kamus basis data

#### A. Kamus Basis Data Tabel Pelanggan

Nama Database : **DB\_Minimarket**

Nama Tabel : **Pelanggan**

Fungsi Tabel : Menyimpan informasi pelanggan yang melakukan transaksi, terutama untuk pelanggan member.

Nama Field	Tipe Data	Panjang	Keterangan
ID_Pelanggan	CHAR	6	PRIMARY KEY, kode unik untuk setiap pelanggan
Nama_Pelanggan	VARCHAR	100	Nama lengkap pelanggan
No_Telepon	VARCHAR	15	Nomor telepon aktif pelanggan
Tipe_Pelanggan	VARCHAR	20	Status pelanggan (Member / Non-member)

Tabel Pelanggan berfungsi sebagai tempat penyimpanan data identitas para pelanggan minimarket. Keberadaan tabel ini sangat penting terutama untuk mendukung fitur membership yang biasanya dimiliki oleh minimarket modern seperti Indomaret maupun Alfamart. Dengan keberadaan ID\_Pelanggan sebagai kunci utama, sistem dapat mengidentifikasi setiap pelanggan secara unik sehingga riwayat pembelian mereka dapat ditelusuri dengan mudah. Informasi Nama\_Pelanggan dan No\_Telepon disediakan untuk

kebutuhan komunikasi, seperti pengiriman notifikasi promo atau penawaran khusus bagi pelanggan loyal. Kolom Tipe\_Pelanggan membantu sistem membedakan pelanggan biasa dengan pelanggan member yang berhak mendapatkan diskon, poin belanja, atau reward tertentu. Data ini juga mendukung analisis perilaku konsumen, membantu pihak minimarket dalam mengambil keputusan pemasaran yang lebih tepat sasaran.

B. Kamus Basis Data Tabel Barang

**Nama Database:** DB\_Minimarket  
**Nama Tabel:** Barang  
**Fungsi Tabel:** Menyimpan data inventaris barang yang tersedia di minimarket.

Nama Field	Tipe Data	Panjang	Keterangan
ID_Barang	CHAR	8	PRIMARY KEY, kode unik untuk setiap barang
Nama_Barang	VARCHAR	150	Nama produk atau item
Kategori	VARCHAR	50	Kategori barang (makanan, minuman, dll.)
Harga_Jual	INT	-	Harga jual barang
Stok	INT	-	Jumlah barang yang tersedia

Tabel Barang merupakan tabel yang berfungsi untuk mengelola informasi seluruh barang yang dijual di minimarket. Setiap barang diberi kode unik (ID\_Barang) untuk mempermudah proses pendataan, pemindaian barcode, dan transaksi di kasir. Kolom Nama\_Barang digunakan untuk mencatat nama lengkap produk yang dijual, sehingga memudahkan proses pelaporan dan pencarian pada sistem. Sementara itu, kolom Kategori sangat membantu dalam pengelompokan barang berdasarkan jenisnya, seperti makanan, minuman, kebutuhan rumah tangga, hingga produk kebersihan. Harga\_Jual disimpan dalam tipe data integer untuk memfasilitasi perhitungan total transaksi. Sedangkan kolom Stok digunakan untuk mencatat jumlah persediaan barang sehingga manajemen inventaris dapat dilakukan dengan lebih akurat. Keberadaan tabel ini sangat krusial karena menjadi pusat informasi untuk operasional penjualan, pengadaan barang, dan pengelolaan stok.

### C. Kamus Basis Data Tabel Transaksi

Nama Database:  
**DB\_Minimarket**

Nama Tabel: **Transaksi**

Fungsi Tabel: **Mencatat seluruh data transaksi yang terjadi di minimarket.**

Struktur Tabel

Nama Field	Tipe Data	Panjang	Keterangan
No_Transaksi	VARCHAR	20	PRIMARY KEY, identitas unik transaksi
Waktu	DATETIME	-	Tanggal dan waktu terjadinya transaksi
Total_Bayar	INT	-	Total harga yang harus dibayar pelanggan
Metode_Bayar	VARCHAR	20	Cash, QRIS, Debit, E-wallet, atau Transfer
ID_Pelanggan	CHAR	6	FOREIGN KEY ke tabel Pelanggan (boleh NULL jika non-member)
ID_Kasir	CHAR	5	FOREIGN KEY ke tabel Kasir

Tabel **Transaksi** merupakan tabel inti dalam sistem minimarket karena berfungsi sebagai pusat pencatatan semua aktivitas pembelian yang dilakukan pelanggan. No\_Transaksi digunakan sebagai identitas unik untuk membedakan satu transaksi dengan transaksi lainnya, dan biasanya digenerate otomatis berdasarkan tanggal dan waktu. Kolom Waktu mencatat detail waktu transaksi, yang penting untuk kebutuhan audit, pelaporan penjualan harian, dan analisis waktu pembelian pelanggan. Total\_Bayar adalah penjumlahan seluruh barang yang dibeli pelanggan pada satu transaksi. Metode\_Bayar mencatat jenis pembayaran, memungkinkan analisis preferensi pembayaran pelanggan seperti tren penggunaan QRIS atau e-wallet. Relasi dengan tabel Pelanggan melalui ID\_Pelanggan membantu menghubungkan transaksi dengan akun member tertentu, sementara relasi dengan tabel Kasir membantu melacak siapa pegawai yang melayani transaksi tersebut. Data ini penting untuk kontrol operasional, pelacakan kesalahan input, hingga evaluasi performa pegawai.

#### D. Kamus Basis Data Tabel Detail Transaksi

Nama Database: DB\_Minimarket

Nama Tabel: Detail\_Transaksi

Fungsi Tabel: Mencatat rincian item dan jumlah barang yang dibeli dalam sebuah transaksi.

Nama Field	Tipe Data	Panjang	Keterangan
No_Transaksi	VARCHAR	20	PRIMARY KEY, FOREIGN KEY ke Tabel Transaksi
ID_Barang	CHAR	8	PRIMARY KEY, FOREIGN KEY ke Tabel Barang
Jumlah	INT	-	Jumlah barang yang dibeli
Subtotal	INT	-	Jumlah × Harga_Jual barang

Tabel Detail\_Transaksi adalah tabel yang berfungsi sebagai penghubung antara tabel Transaksi dan tabel Barang. Tabel ini sering disebut sebagai tabel jembatan (bridge table) atau tabel many-to-many karena memungkinkan pencatatan banyak barang di dalam satu transaksi. Dengan menetapkan kombinasi antara No\_Transaksi dan ID\_Barang sebagai composite primary key, sistem dapat memastikan bahwa setiap barang hanya tercatat satu kali per transaksi, meskipun jumlahnya lebih dari satu. Kolom Jumlah digunakan untuk mencatat kuantitas barang yang dibeli, sementara Subtotal mengakomodasi perhitungan jumlah barang dikalikan harga satuan. Keberadaan tabel ini memudahkan sistem untuk menghitung total pembelian, menganalisis barang yang paling sering dibeli, memantau stok keluar, dan melakukan pelaporan penjualan berdasarkan kategori atau item tertentu.

#### E. Kamus Basis Data Tabel Pelanggan

Nama Database: DB\_Minimarket

Nama Tabel: Kasir

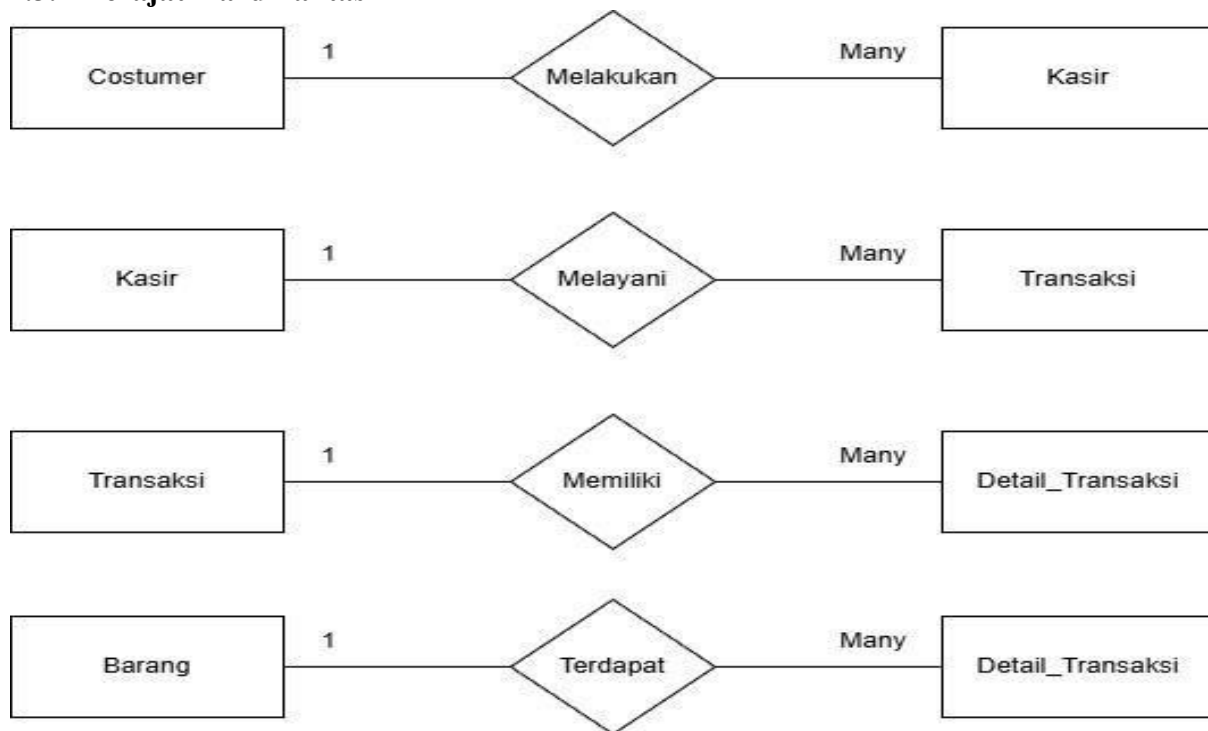
Fungsi Tabel: Menyimpan data pegawai yang bertugas sebagai kasir di minimarket.

Nama Field	Tipe Data	Panjang	Keterangan
ID_Kasir	CHAR	5	PRIMARY KEY
Nama_Kasir	VARCHAR	100	Nama lengkap kasir
No_Telepon	VARCHAR	15	Nomor telepon pegawai kasir

Tabel **Kasir** digunakan untuk mengelola data pegawai yang bertugas melayani pelanggan di bagian kasir. Setiap kasir diberikan ID\_Kasir sebagai identitas unik untuk memastikan bahwa setiap transaksi dapat ditelusuri siapa pegawai yang melayaninya. Penyimpanan Nama\_Kasir diperlukan untuk memudahkan pengenalan dalam sistem dan laporan internal. Kolom No\_Telepon juga dicatat untuk keperluan komunikasi antar

pegawai atau informasi kontak yang dibutuhkan oleh manajemen. Dengan adanya relasi antara tabel Kasir dan tabel Transaksi, minimarket dapat melacak performa penjualan masing-masing kasir, mengetahui kasir yang bertugas pada waktu tertentu, hingga memeriksa transaksi apabila ada kesalahan input atau selisih kas.

### 2.3.2 Derajat Kardinalitas



Penjelasan Kardinalitas pada Setiap Relasi

#### 1. Customer — Melakukan — Kasir

Kardinalitas: 1 Customer : Many Kasir Artinya:

- Seorang Customer (pelanggan) dapat melakukan interaksi dengan banyak Kasir (dalam beberapa transaksi berbeda).
- Namun setiap interaksi/transaksi dilayani oleh tepat 1 Kasir.

#### 2. Kasir — Melayani — Transaksi

Kardinalitas: 1 Kasir : Many Transaksi Artinya:

- Satu Kasir dapat melayani banyak Transaksi.
- Setiap Transaksi hanya dilayani oleh satu Kasir.

### 3. Transaksi — Memiliki — Detail\_Transaksi

Kardinalitas: 1 Transaksi : Many Detail\_Transaksi Artinya:

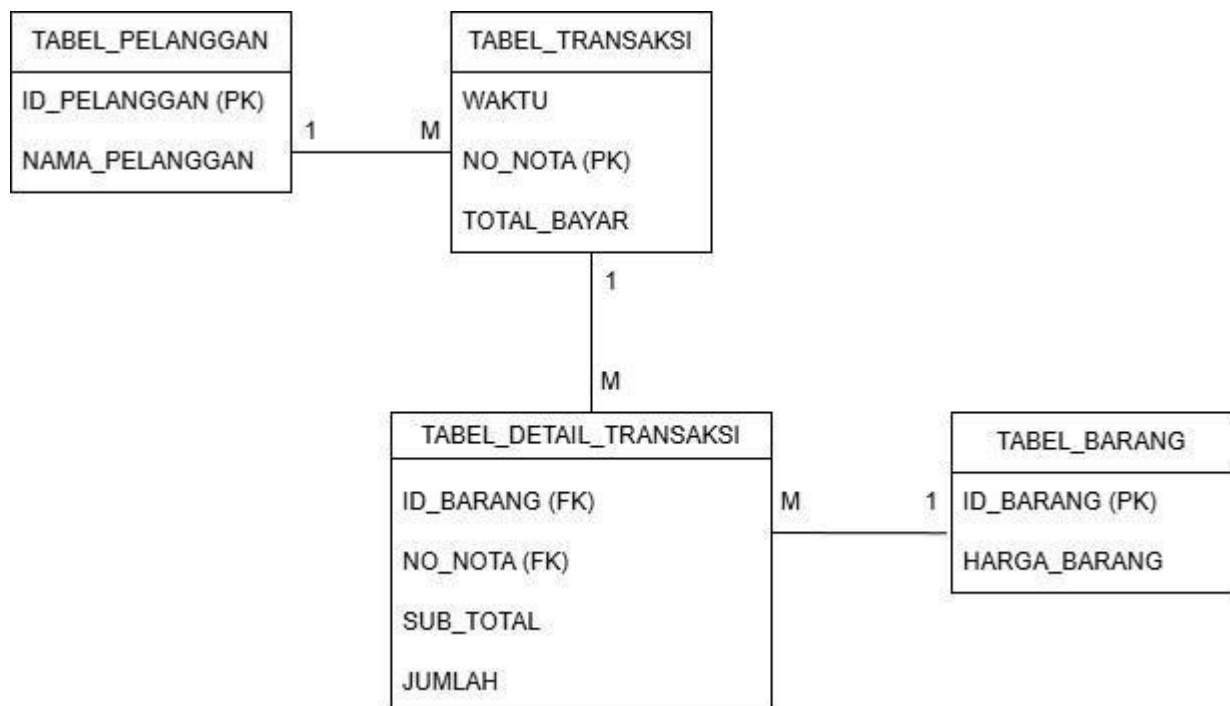
- Satu Transaksi memiliki banyak Detail Transaksi (item belanja).
- Setiap Detail\_Transaksi hanya terhubung dengan satu Transaksi.

### 4. Barang — Terdapat — Detail\_Transaksi

Kardinalitas: 1 Barang : Many Detail\_Transaksi Artinya:

- Satu Barang dapat muncul di banyak Detail\_Transaksi (dalam banyak transaksi yang berbeda).
- Setiap Detail\_Transaksi hanya mewakili 1 Barang.

## 2.3.3 RELASI ANTAR TABEL



Model database ini mendeskripsikan sebuah sistem transaksi sederhana yang terdiri dari empat tabel utama: TABEL\_PELANGGAN, TABEL\_TRANSAKSI,



TABEL\_DETAIL\_TRANSAKSI, dan TABEL\_BARANG. TABEL\_PELANGGAN berelasi One-to-Many (1:M) dengan TABEL\_TRANSAKSI, menandakan bahwa satu pelanggan dapat melakukan banyak transaksi, dan relasi ini dihubungkan melalui atribut ID\_PELANGGAN.

### 2.3.3 ERD Final

Model ini menggambarkan sistem penjualan toko melalui lima entitas utama: Pelanggan, Kasir, Transaksi, Barang, dan Detail Transaksi. Entitas Pelanggan dan Kasir berinteraksi melalui relasi Melayani untuk menghasilkan Transaksi. Setiap Transaksi dicatat dengan atribut seperti No-nota dan Waktu. Karena satu transaksi dapat mencakup banyak barang dan satu barang dapat dijual dalam banyak transaksi, maka relasi Many-to-Many ini dipecahkan oleh entitas perantara yaitu Detail Transaksi. Entitas Transaksi memiliki relasi Memiliki dengan Detail Transaksi, dan entitas Barang memiliki relasi Terdapat dengan Detail

Transaksi. Detail Transaksi berfungsi untuk menyimpan data kuantitas (jumlah) dan harga per item (sub-total) untuk setiap barang dalam setiap transaksi, menggunakan No-nota dan id-barang sebagai kunci gabungan. Dengan demikian, model ini secara lengkap memetakan proses penjualan dari layanan kasir hingga perincian setiap item yang terjual.

## BAB III

### IMPLEMENTASI BASIS DATA

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

Pada tahap ini, dilakukan implementasi rancangan fisik database menggunakan perintah DDL. Database yang dibuat bernama db\_indomaret, yang terdiri dari lima tabel utama yang saling berelasi.

Berikut adalah perintah SQL untuk pembuatan struktur tabel beserta *constraint* Primary Key (PK) dan Foreign Key (FK):

##### 1. Pembuatan Database dan Tabel

```
CREATE DATABASE db_indomaret;  
USE db_indomaret;
```

-- 1. Tabel Pelanggan

```
CREATE TABLE Pelanggan (  
    Id_Pelanggan CHAR(5) PRIMARY KEY,  
    Nama_Pelanggan VARCHAR(50),  
    No_Telepon VARCHAR(15),  
    Tipe_Pelanggan VARCHAR(20)  
);
```

-- 2. Tabel Kasir

```
CREATE TABLE Kasir (  
    Id_Kasir CHAR(5) PRIMARY KEY,  
    Nama_Kasir VARCHAR(50),  
    No_Telepon VARCHAR(15)  
);
```

-- 3. Tabel Barang

```
CREATE TABLE Barang (  
    Id_Barang CHAR(5) PRIMARY KEY,  
    Nama_Barang VARCHAR(100),  
    Kategori VARCHAR(30),  
    Harga_Jual INT,  
    Stok INT  
);
```

-- 4. Tabel Transaksi

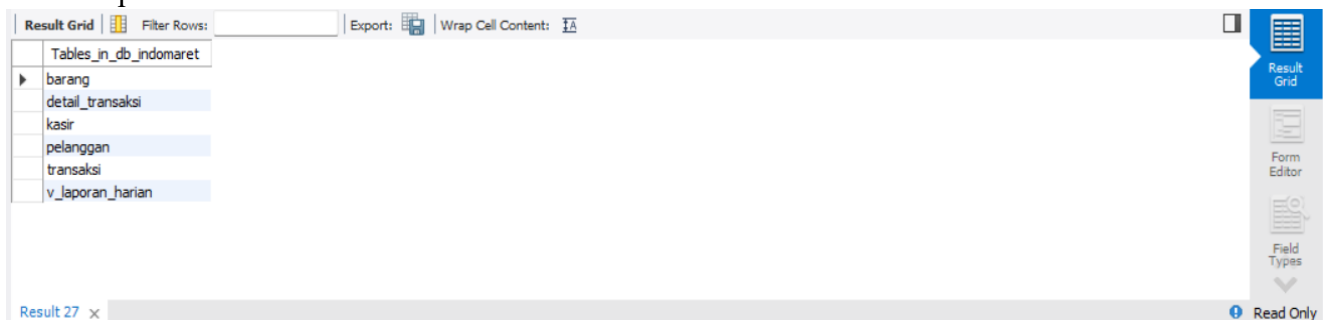
```
CREATE TABLE Transaksi (  
    No_Nota CHAR(10) PRIMARY KEY,  
    Waktu DATETIME,  
    Total_Bayar INT,  
    Metode_Bayar VARCHAR(20),  
    Id_Pelanggan CHAR(5),  
    Id_Kasir CHAR(5),  
    FOREIGN KEY (Id_Pelanggan) REFERENCES Pelanggan(Id_Pelanggan),  
    FOREIGN KEY (Id_Kasir) REFERENCES Kasir(Id_Kasir)  
);
```

-- 5. Tabel Detail Transaksi

```
CREATE TABLE Detail_Transaksi (  
    Id_Barang CHAR(5) PRIMARY KEY,  
    Id_Transaksi CHAR(10) PRIMARY KEY,  
    Jumlah INT  
);
```

No\_Nota CHAR(10),  
 Id\_Barang CHAR(5),  
 Jumlah INT,  
 Subtotal INT,  
 PRIMARY KEY (No\_Nota, Id\_Barang),  
 FOREIGN KEY (No\_Nota) REFERENCES Transaksi(No\_Nota),  
 FOREIGN KEY (Id\_Barang) REFERENCES Barang(Id\_Barang)  
 );

### Hasil Implementasi Tabel:



The screenshot shows a database management interface with a table list on the left. The table 'Tables\_in\_db\_indomaret' is expanded, showing a list of tables: barang, detail\_transaksi, kasir, pelanggan, transaksi, and v\_laporan\_harian. The right sidebar contains icons for 'Result Grid', 'Form Editor', and 'Field Types'.

### Hasil Implementasi Tabel Barang:



The screenshot shows a table structure view for the 'Barang' table. The table has the following columns: Id\_Barang (char(5), NO, PRI, NULL), Nama\_Barang (varchar(100), YES, NULL), Kategori (varchar(30), YES, NULL), Harga\_Jual (int(11), YES, NULL), and Stok (int(11), YES, NULL). The right sidebar contains icons for 'Result Grid', 'Form Editor', and 'Field Types'.

### Hasil Implementasi Tabel Detail\_Transaksi:



The screenshot shows a table structure view for the 'Detail\_Transaksi' table. The table has the following columns: No\_Nota (char(10), NO, PRI, NULL), Id\_Barang (char(5), NO, PRI, NULL), Jumlah (int(11), YES, NULL), and Subtotal (int(11), YES, NULL). The right sidebar contains icons for 'Result Grid', 'Form Editor', and 'Field Types'.

### Hasil Implementasi Tabel Transaksi:



The screenshot shows a table structure view for the 'Transaksi' table. The table has the following columns: No\_Nota (char(10), NO, PRI, NULL), Waktu (datetime, YES, NULL), Total\_Bayar (int(11), YES, NULL), Metode\_Bayar (varchar(20), YES, NULL), Id\_Pelanggan (char(5), YES, MUL, NULL), and Id\_Kasir (char(5), YES, MUL, NULL). The right sidebar contains icons for 'Result Grid', 'Form Editor', and 'Field Types'.

### Hasil Implementasi Tabel Kasir:

Field	Type	Null	Key	Default	Extra
Id_Kasir	char(5)	NO	PRI	NULL	
Nama_Kasir	varchar(50)	YES		NULL	
No_Telepon	varchar(15)	YES		NULL	

### Hasil Implementasi Tabel Pelanggan:

Field	Type	Null	Key	Default	Extra
Id_Pelanggan	char(5)	NO	PRI	NULL	
Nama_Pelanggan	varchar(50)	YES		NULL	
No_Telepon	varchar(15)	YES		NULL	
Tipe_Pelanggan	varchar(20)	YES		NULL	

## 3.2 Implementasi DML (Data Manipulation Language)

Setelah struktur tabel terbentuk, dilakukan pengisian data awal (*dummy data*) untuk keperluan pengujian sistem. Data yang dimasukkan meliputi data master (Pelanggan, Kasir, Barang).

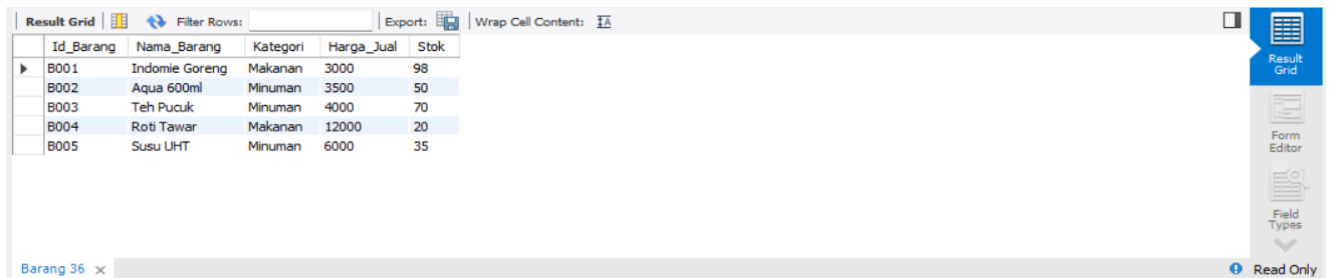
#### Kode SQL Pengisian Data:

```
-- Isi Data Pelanggan (PENTING: Diisi dulu sebelum Transaksi)
INSERT INTO Pelanggan VALUES ('P01', 'Budi Santoso', '08111222333', 'Member');
INSERT INTO Pelanggan VALUES ('P02', 'Siti Aminah', '08555666777', 'Non-Member');

-- Isi Data Kasir
INSERT INTO Kasir VALUES ('K01', 'Diva Anggara', '08123456789');
INSERT INTO Kasir VALUES ('K02', 'Falih Nabil', '08198765432');

-- Isi Data Barang
INSERT INTO Barang VALUES ('B001', 'Indomie Goreng', 'Makanan', 3000, 100);
INSERT INTO Barang VALUES ('B002', 'Aqua 600ml', 'Minuman', 3500, 50);
INSERT INTO Barang VALUES ('B003', 'Teh Pucuk', 'Minuman', 4000, 70);
INSERT INTO Barang VALUES ('B004', 'Roti Tawar', 'Makanan', 12000, 20);
INSERT INTO Barang VALUES ('B005', 'Susu UHT', 'Minuman', 6000, 40);
```

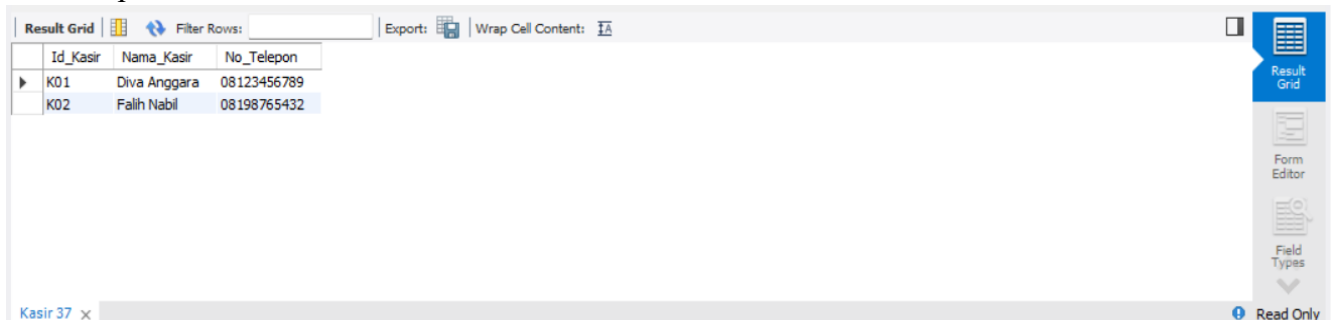
### Hasil Implementasi Isi Data Barang:



The screenshot shows a data grid interface with a toolbar at the top containing 'Filter Rows', 'Export', and 'Wrap Cell Content' options. The grid displays five rows of barang data. On the right side, there is a vertical toolbar with 'Result Grid', 'Form Editor', and 'Field Types' buttons. At the bottom, a tab labeled 'Barang 36' is visible, and a 'Read Only' status indicator is present.

Id_Barang	Nama_Barang	Kategori	Harga_Jual	Stok
B001	Indomie Goreng	Makanan	3000	98
B002	Aqua 600ml	Minuman	3500	50
B003	Teh Pucuk	Minuman	4000	70
B004	Roti Tawar	Makanan	12000	20
B005	Susu UHT	Minuman	6000	35

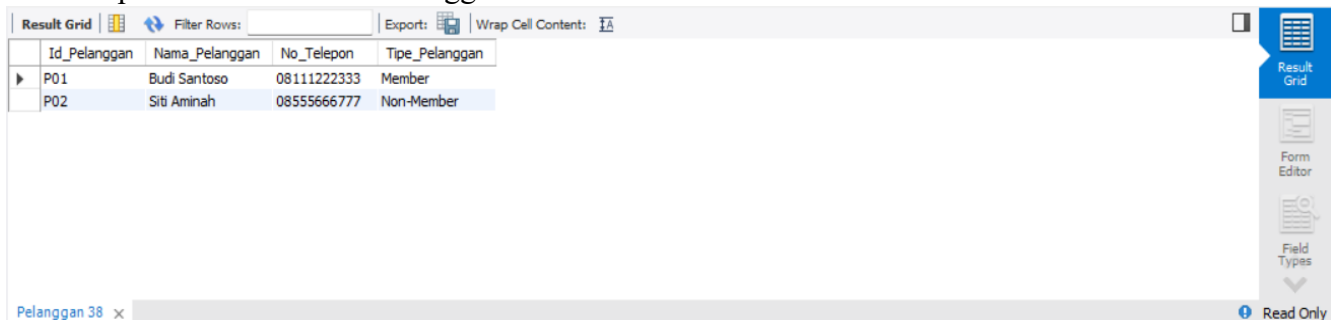
### Hasil Implementasi Isi Data Kasir:



The screenshot shows a data grid interface with a toolbar at the top containing 'Filter Rows', 'Export', and 'Wrap Cell Content' options. The grid displays two rows of kasir data. On the right side, there is a vertical toolbar with 'Result Grid', 'Form Editor', and 'Field Types' buttons. At the bottom, a tab labeled 'Kasir 37' is visible, and a 'Read Only' status indicator is present.

Id_Kasir	Nama_Kasir	No_Telepon
K01	Diva Anggara	08123456789
K02	Falih Nabil	08198765432

### Hasil Implementasi Isi Data Pelanggan:



The screenshot shows a data grid interface with a toolbar at the top containing 'Filter Rows', 'Export', and 'Wrap Cell Content' options. The grid displays two rows of pelanggan data. On the right side, there is a vertical toolbar with 'Result Grid', 'Form Editor', and 'Field Types' buttons. At the bottom, a tab labeled 'Pelanggan 38' is visible, and a 'Read Only' status indicator is present.

Id_Pelanggan	Nama_Pelanggan	No_Telepon	Tipe_Pelanggan
P01	Budi Santoso	08111222333	Member
P02	Siti Aminah	08555666777	Non-Member

## 3.3 Implementasi TCL (Transaction Control Language)

Sistem menerapkan mekanisme transaksi (Transaction) untuk menjamin integritas data. Transaksi memastikan bahwa sekumpulan operasi database (Insert Header dan Detail) berhasil disimpan secara utuh (Commit) atau dibatalkan semua jika terjadi kesalahan (Rollback).

### Skenario Transaksi Manual:

-- Skenario 1: Transaksi Manual dengan COMMIT (Syarat TCL)

START TRANSACTION;

-- Header Transaksi

INSERT INTO Transaksi VALUES ('TRX001', '2025-11-14 10:00:00', 6000, 'Tunai', NULL, 'K01');

-- Detail Transaksi

INSERT INTO Detail\_Transaksi VALUES ('TRX001', 'B001', 2, 6000);

-- Update Stok Manual (Sebelum ada Trigger)

UPDATE Barang SET Stok = Stok - 2 WHERE Id\_Barang = 'B001';

COMMIT;

-- Skenario 2: Transaksi Besar (Untuk Data Dummy Laporan)

```
INSERT INTO Transaksi VALUES ('TRX-BIG', '2025-01-17 14:00:00', 180000, 'Tunai', NULL, 'K01');
```

```
INSERT INTO Detail_Transaksi VALUES ('TRX-BIG', 'B001', 60, 180000);
```

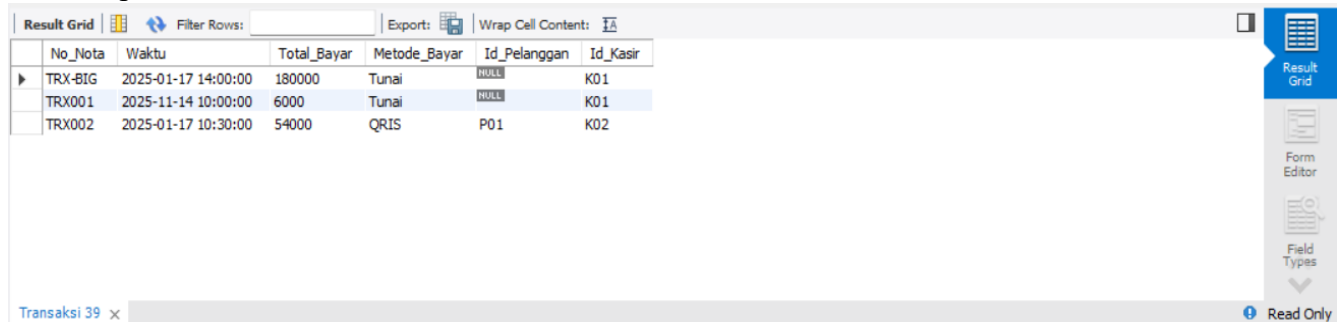
-- Skenario 3: Transaksi Pelanggan Member (Data Awal TRX002)

```
INSERT INTO Transaksi VALUES ('TRX002', '2025-01-17 10:30:00', 54000, 'QRIS', 'P01', 'K02');
```

```
INSERT INTO Detail_Transaksi VALUES ('TRX002', 'B004', 2, 24000);
```

-- (Catatan: TRX002 totalnya 54rb, rinciannya 24rb Roti + 30rb Susu yg akan diinput nanti saat tes Trigger)

### Hasil Implementasi TCL:



No_Nota	Waktu	Total_Bayar	Metode_Bayar	Id_Pelanggan	Id_Kasir
TRX-BIG	2025-01-17 14:00:00	180000	Tunai	NULL	K01
TRX001	2025-11-14 10:00:00	6000	Tunai	NULL	K01
TRX002	2025-01-17 10:30:00	54000	QRIS	P01	K02

## 3.4 Query dan Pelaporan Data

Bagian ini menyajikan informasi strategis yang dihasilkan melalui perintah SQL kompleks, mencakup Join, Aggregation, dan Filtering.

### 3.4.1 Query Join (Cetak Struk Belanja)

Query ini menghubungkan tabel Transaksi, Detail, dan Barang untuk menampilkan rincian belanjaan pelanggan secara lengkap.

Kode SQL:

-- 5.1 Query JOIN (Struk Belanja)

```
SELECT t.No_Nota, t.Waktu, b>Nama_Barang, d.Jumlah, d.Subtotal
```

```
FROM Transaksi t
```

```
JOIN Detail_Transaksi d ON t.No_Nota = d.No_Nota
```

```
JOIN Barang b ON d.Id_Barang = b.Id_Barang
```

```
WHERE t.No_Nota = 'TRX001';
```

Hasil Implementasi:

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

No_Nota	Waktu	Nama_Barang	Jumlah	Subtotal
TRX001	2025-11-14 10:00:00	Indomie Goreng	2	6000

Result 42 x | Read Only

### 3.4.2 Query Aggregation & Group By (Laporan Omset Kasir)

Query ini digunakan untuk menghitung total pendapatan yang dihasilkan oleh masing-masing kasir.

Kode SQL:

```
-- 5.2 Query GROUP BY (Laporan Omset Kasir)
SELECT k>Nama_Kasir, COUNT(t.No_Nota) AS Jumlah_Transaksi, SUM(t.Total_Bayar) AS Total_Pendapatan
FROM Kasir k
JOIN Transaksi t ON k.Id_Kasir = t.Id_Kasir
GROUP BY k>Nama_Kasir;
```

Hasil Implementasi:

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Nama_Kasir	Jumlah_Transaksi	Total_Pendapatan
Diva Anggara	2	186000
Falih Nabil	1	54000

Result 43 x | Read Only

### 3.4.3 Query Having (Analisis Barang Terlaris)

Query ini menyaring barang-barang yang penjualannya melebihi jumlah tertentu (Contoh: > 50 unit) untuk menentukan Best Seller.

Kode SQL:

```
-- 5.3 Query HAVING (Barang Terlaris > 50)
SELECT b>Nama_Barang, SUM(d.Jumlah) AS Total_Terjual
FROM Barang b
JOIN Detail_Transaksi d ON b.Id_Barang = d.Id_Barang
GROUP BY b>Nama_Barang
HAVING SUM(d.Jumlah) > 50;
```

Hasil Implementasi:



Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Nama_Barang	Total_Terjual		
Indomie Goreng	62		

Result 44 x Read Only

### 3.4.4 Subquery dan View

Sistem menggunakan *View* untuk menyederhanakan laporan harian yang sering diakses, serta *Subquery* untuk mencari barang dengan harga di atas rata-rata.

Kode SQL Subquery:

```
-- 5.4 Subquery (Mencari Barang di atas Harga Rata-rata)
SELECT Nama_Barang, Harga_Jual
FROM Barang
WHERE Harga_Jual > (SELECT AVG(Harga_Jual) FROM Barang);
```

Hasil Implementasi:

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Nama_Barang	Harga_Jual		
Roti Tawar	12000		
Susu UHT	6000		

Barang 47 x Read Only

Kode SQL View:

```
-- 5.5 VIEW (Laporan Virtual)
CREATE OR REPLACE VIEW V_Laporan_Harian AS
SELECT t.No_Nota, t.Waktu, p>Nama_Pelanggan, t.Total_Bayar, t.Metode_Bayar
FROM Transaksi t
LEFT JOIN Pelanggan p ON t.Id_Pelanggan = p.Id_Pelanggan;
```

Hasil Implementasi:

No_Nota	Waktu	Nama_Pelanggan	Total_Bayar	Metode_Bayar
TRX-BIG	2025-01-17 14:00:00	NULL	1 180000	Tunai
TRX001	2025-11-14 10:00:00	NULL	6000	Tunai
TRX002	2025-01-17 10:30:00	Budi Santoso	54000	QRIS

### 3.5 Fitur Lanjutan (Otomatisasi Trigger)

Sebagai nilai tambah, sistem dilengkapi dengan *Trigger* database. Fitur ini berfungsi memotong stok barang secara otomatis di tabel Barang setiap kali ada input baru di tabel Detail\_Transaksi.

Kode Trigger:

```
-- 6.1 Membuat Trigger Otomatis Potong Stok
```

```
DELIMITER $$
```

```
CREATE TRIGGER Update_Stok_Otomatis
```

```
AFTER INSERT ON Detail_Transaksi
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Barang
```

```
    SET Stok = Stok - NEW.Jumlah
```

```
    WHERE Id_Barang = NEW.Id_Barang;
```

```
END $$
```

```
DELIMITER ;
```

```
-- Step A: Cek Stok Awal Susu UHT (B005) - Harusnya 40
```

```
SELECT Id_Barang, Nama_Barang, Stok FROM Barang WHERE Id_Barang = 'B005';
```

```
-- Step B: Tambah Item Susu ke TRX002 (Beli 5 pcs)
```

```
-- Saat script ini jalan, Trigger akan aktif otomatis
```

```
INSERT INTO Detail_Transaksi VALUES ('TRX002', 'B005', 5, 30000);
```

```
-- Step C: Cek Stok Akhir Susu UHT (B005) - Harusnya jadi 35
```

```
SELECT Id_Barang, Nama_Barang, Stok FROM Barang WHERE Id_Barang = 'B005';
```

Hasil Implementasi Step A:

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	
	Id_Barang	Nama_Barang	Stok			
▶	B005	Susu UHT	35			

Barang 49 x

Read Only

### Hasil Implementasi Step B:

✓ 87 18:34:20 INSERT INTO Detail\_Transaksi VALUES (TRX002, 'B005', 5, 30000) 1 row(s) affected 0.000 sec

### Hasil Implementasi Step C:

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	
	Id_Barang	Nama_Barang	Stok			
▶	B005	Susu UHT	35			

Barang 50 x

Read Only

### Pengujian Trigger:

1. Stok Awal: Sebelum transaksi, stok Susu UHT adalah 40. [SCREENSHOT STOK AWAL 40]
2. Proses Transaksi: Dilakukan pembelian 5 unit Susu UHT.
3. Stok Akhir: Setelah transaksi, stok berkurang otomatis menjadi 35. [SCREENSHOT STOK AKHIR 35]

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Berdasarkan tahapan perancangan dan implementasi yang telah dilakukan pada bab-bab sebelumnya, dapat ditarik beberapa kesimpulan mengenai sistem basis data transaksi kasir Minimarket Indomaret ini:

1. Keberhasilan Perancangan: Sistem basis data telah berhasil dirancang mulai dari tahap konseptual (ERD) hingga tahap fisik (Tabel SQL). Struktur tabel telah memenuhi kaidah Normalisasi tingkat ketiga (3NF), sehingga redundansi data dapat diminimalisir dan integritas data terjaga.
2. Integritas Data: Penerapan Constraint berupa Primary Key dan Foreign Key pada setiap tabel memastikan relasi antar data (Pelanggan, Kasir, Barang, dan Transaksi) tetap konsisten dan valid.
3. Fungsionalitas Query: Pengujian menggunakan bahasa SQL membuktikan bahwa sistem mampu menangani operasi data yang kompleks, antara lain:
  - Join: Mampu mencetak struk belanja lengkap dengan menggabungkan data dari tiga tabel berbeda.
  - Agregasi: Mampu menghitung total omset pendapatan per kasir secara akurat.
  - View: Mampu menyajikan laporan harian virtual untuk kebutuhan manajerial.
4. Otomatisasi Sistem: Implementasi fitur Trigger pada database berhasil melakukan pemotongan stok barang secara otomatis dan real-time setiap kali transaksi penjualan terjadi, sehingga mengurangi risiko kesalahan pencatatan manual.

#### **4.2 Kendala dan Solusi**

Selama proses pengerjaan tugas akhir ini, terdapat beberapa kendala teknis yang dihadapi beserta solusinya:

1. Kendala Foreign Key: Pada awal implementasi, terjadi error saat pengisian data transaksi karena data referensi (Pelanggan/Kasir) belum tersedia.
  - Solusi: Memperbaiki urutan eksekusi kode SQL dengan mendahulukan pengisian tabel induk (Master Data) sebelum tabel transaksi.
2. Logika Trigger: Kesulitan dalam menentukan logika pengurangan stok agar berjalan otomatis setelah insert data detail.
  - Solusi: Menggunakan fungsi NEW.Jumlah dan NEW.Id\_Barang dalam blok algoritma Trigger untuk menangkap data yang baru masuk secara dinamis.

#### **4.3 Saran**

Sistem basis data yang telah dirancang ini masih bersifat back-end (pengelolaan data) dan memiliki potensi untuk dikembangkan lebih lanjut. Beberapa saran untuk pengembangan di masa depan adalah:

1. Pembuatan Antarmuka (Interface): Sistem basis data ini sebaiknya dihubungkan dengan aplikasi berbasis Web atau Desktop (menggunakan bahasa pemrograman seperti PHP, Java, atau Python) agar kasir dapat menginput data melalui formulir visual yang user-friendly tanpa harus mengetik perintah SQL.
2. Keamanan Data: Perlu ditambahkan manajemen hak akses (User Privileges) yang lebih ketat untuk membedakan hak akses antara Kasir (hanya Insert transaksi) dan Manajer Toko (Bisa melihat laporan View).
3. Fitur Analitik Lanjutan: Sistem dapat dikembangkan dengan menambahkan query analitik yang lebih mendalam, seperti prediksi kebutuhan stok barang berdasarkan tren penjualan bulanan atau analisis kebiasaan belanja pelanggan member.

## DAFTAR PUSTAKA

1. Irawan RD. Modul Mata Kuliah Pemrograman Basis Data. Surakarta: Universitas Duta Bangsa; 2024.
2. Connolly T, Begg C. Database Systems: A Practical Approach to Design, Implementation, and Management. 6th ed. London: Pearson Education; 2015.
3. Elmasri R, Navathe SB. Fundamentals of Database Systems. 7th ed. Boston: Pearson Education; 2017.
4. Kroenke DM, Auer DJ. Database Concepts. 8th ed. New York: Pearson Education; 2016.
5. Oracle Corporation. MySQL 8.0 Reference Manual [Internet]. 2024 [dikutip 18 Januari 2025]. Tersedia dari: <https://dev.mysql.com/doc/refman/8.0/en/>

## LAMPIRAN

### Lampiran 1: Link Repository Proyek

Seluruh kode sumber (source code), skrip SQL, dan dokumentasi proyek ini dapat diakses secara daring melalui tautan GitHub berikut:

- URL Repository: <https://github.com/falihnbilm576-lgtm/UAS-BasisData-Indomaret-2026.git>
- Nama Repository: UAS-BasisData-Indomaret-2026