

**PEMOGRAMAN BASIS DATA**  
**PROJECT UTS**  
**NOTA TRANSAKSI BELANJA**



**UNIVERSITAS  
DUTA BANGSA  
SURAKARTA**

Dosen Pengampu: Ridwan Dwi Irawan, M.Kom

Disusun oleh:

Bintang Fajar Mustika Aji (240103186) Dhiannikha  
Wahyu Nur Hidayat (240103187) Nige Lastara  
(240103198)

**PROGRAM STUDI S1-TEKNIK  
INFORMATIKA FAKULTAS ILMU  
KOMPUTER  
UNIVERSITAS DUTA BANGSA  
SURAKARTA TAHUN 2025**

## KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, taufik, serta hidayah-Nya, sehingga kami dapat menyelesaikan laporan proyek Ujian Tengah Semester (UTS) Basis Data Transaksi Belanja ini dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk penilaian dalam mata kuliah Pemrograman Basis Data, di mana mahasiswa dituntut untuk memahami dan menerapkan konsep perancangan basis data secara sistematis dan terstruktur dalam bentuk proyek nyata.

Proyek ini mengangkat studi kasus sistem transaksi belanja, yang diharapkan dapat menggambarkan bagaimana proses penyimpanan, pengelolaan, dan hubungan data antar tabel dilakukan dalam suatu sistem yang terkomputerisasi. Melalui proyek ini, kami berusaha untuk mengimplementasikan konsep-konsep penting seperti normalisasi, relasi antar tabel, serta pembuatan *Entity Relationship Diagram (ERD)* agar dapat menghasilkan desain database yang efisien, bebas redundansi, dan mudah digunakan.

Kami menyadari bahwa tersusunnya laporan ini tidak terlepas dari bimbingan, dukungan, dan bantuan berbagai pihak. Oleh karena itu, kami ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada dosen pengampu mata kuliah Pemrograman Basis Data, Bapak Ridwan Dwi Irawan, M.Kom., yang telah memberikan arahan, ilmu, dan motivasi selama proses pembelajaran berlangsung. Ucapan terima kasih juga kami sampaikan kepada teman-teman satu kelompok yang telah bekerja sama dengan penuh tanggung jawab dan semangat, serta semua pihak yang turut membantu hingga laporan ini dapat diselesaikan dengan baik.

Kami menyadari sepenuhnya bahwa laporan ini masih memiliki berbagai kekurangan, baik dari segi isi maupun tata penyajiannya. Namun, kami telah berupaya maksimal sesuai dengan kemampuan dan pengetahuan yang dimiliki. Untuk itu, kami dengan rendah hati menerima segala bentuk kritik, saran, dan masukan yang membangun demi penyempurnaan laporan ini di masa mendatang.

Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah berpartisipasi dalam penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat, menambah wawasan bagi pembaca, serta menjadi referensi dalam memahami penerapan basis data pada sistem transaksi belanja.

Surakarta, 14 November 2025

Penyusun

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>BAB I.....</b>	<b>4</b>
<b>PENDAHULUAN.....</b>	<b>4</b>
<b>BAB II.....</b>	<b>5</b>
<b>STUDI KASUS.....</b>	<b>5</b>
<b>BAB III.....</b>	<b>8</b>
<b>PERANCANGAN BASIS DATA.....</b>	<b>8</b>
Penjelasan:.....	12
2. DML (Data Manipulation Language).....	13
3. TCL (Transaction Control Language).....	13
<b>BAB IV.....</b>	<b>17</b>
<b>PENUTUP.....</b>	<b>17</b>

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Perkembangan teknologi informasi telah membawa perubahan besar dalam berbagai bidang, termasuk sektor perdagangan. Toko dan perusahaan kini dituntut untuk mampu beradaptasi dengan sistem digital agar kegiatan operasional dapat berjalan lebih efektif dan efisien. Salah satu aspek penting dalam kegiatan bisnis adalah pengelolaan data transaksi belanja, yang mencakup data pelanggan, produk, stok barang, serta proses pembayaran.

Sistem transaksi yang masih dilakukan secara manual sering kali menimbulkan berbagai permasalahan, seperti kesalahan pencatatan, keterlambatan dalam proses pembayaran, dan kesulitan dalam memantau persediaan barang. Oleh karena itu, dibutuhkan sistem basis data yang terstruktur dengan baik untuk mengelola seluruh proses transaksi belanja secara terintegrasi. Dengan adanya sistem basis data, kegiatan penjualan dapat berjalan lebih cepat, akurat, dan transparan, sehingga meningkatkan kepuasan pelanggan serta mempermudah pihak toko dalam pengambilan keputusan.

Selain itu, penerapan sistem transaksi berbasis komputer juga dapat membantu pengelola toko dalam melakukan analisis penjualan, memprediksi kebutuhan stok, dan mengoptimalkan strategi pemasaran. Melalui sistem yang terkomputerisasi, seluruh data transaksi dapat tersimpan dengan aman dan mudah diakses kapan pun dibutuhkan, sehingga mendukung efisiensi serta peningkatan kinerja usaha secara keseluruhan.

#### **B. Rumusan Masalah**

1. Bagaimana merancang sistem basis data yang mampu mencatat setiap transaksi belanja secara terstruktur dan efisien?
2. Bagaimana mengelola data barang, pelanggan, dan nota pembayaran agar saling terhubung dengan baik di dalam sistem?
3. Bagaimana membuat proses pencarian dan pembuatan laporan transaksi menjadi lebih cepat dan akurat?
4. Bagaimana sistem dapat meminimalkan kesalahan dalam pencatatan transaksi serta mempermudah proses pemantauan data penjualan?

#### **C. Tujuan**

1. Merancang struktur basis data untuk sistem transaksi belanja yang mencakup data pelanggan, barang, nota pembayaran, dan detail transaksi agar dapat dikelola secara terintegrasi.
2. Melakukan proses normalisasi hingga Third Normal Form (3NF) untuk memastikan struktur data yang efisien, bebas dari duplikasi, serta mudah dipelihara.
3. Menyusun

Entity Relationship Diagram (ERD) sebagai model konseptual yang menggambarkan hubungan antar entitas dalam sistem transaksi belanja.

## D. Manfaat

Manfaat dari perancangan sistem basis data transaksi belanja ini adalah untuk meningkatkan efisiensi dan keakuratan dalam pengelolaan data transaksi pada suatu toko atau usaha. Rancangan basis data yang dibuat dapat dijadikan dasar dalam pengembangan aplikasi informasi penjualan di masa mendatang. Selain itu, sistem ini memberikan kemudahan bagi pihak pengelola dalam memantau dan mengelola data pelanggan, barang, serta proses pembayaran secara terpusat dan terintegrasi.

Dengan adanya sistem basis data yang terstruktur, proses pencatatan transaksi dapat dilakukan dengan lebih cepat, mengurangi risiko kesalahan input data, serta mempermudah penyusunan laporan penjualan. Hal ini diharapkan dapat membantu pengambilan keputusan bisnis secara lebih tepat dan mendukung peningkatan kinerja operasional toko atau perusahaan.

## BAB II

### STUDI KASUS

#### A. Deskripsi Umum

Toko atau perusahaan yang bergerak di bidang perdagangan memiliki berbagai kegiatan operasional yang berkaitan dengan transaksi penjualan, seperti pencatatan data pelanggan, pengelolaan stok barang, serta proses pembayaran. Seluruh kegiatan tersebut perlu dikelola dengan baik agar data transaksi dapat tersimpan secara rapi, akurat, dan mudah diakses kembali saat dibutuhkan.

Sistem basis data transaksi belanja ini dirancang untuk membantu proses pencatatan dan pengelolaan data transaksi secara terkomputerisasi. Sistem mencatat informasi mengenai pelanggan, barang, jumlah pembelian, harga, serta total pembayaran yang tercantum dalam nota transaksi. Dengan adanya sistem ini, proses administrasi menjadi lebih cepat, mengurangi kesalahan pencatatan, serta memudahkan pihak pengelola dalam memantau data penjualan secara terpusat dan efisien.

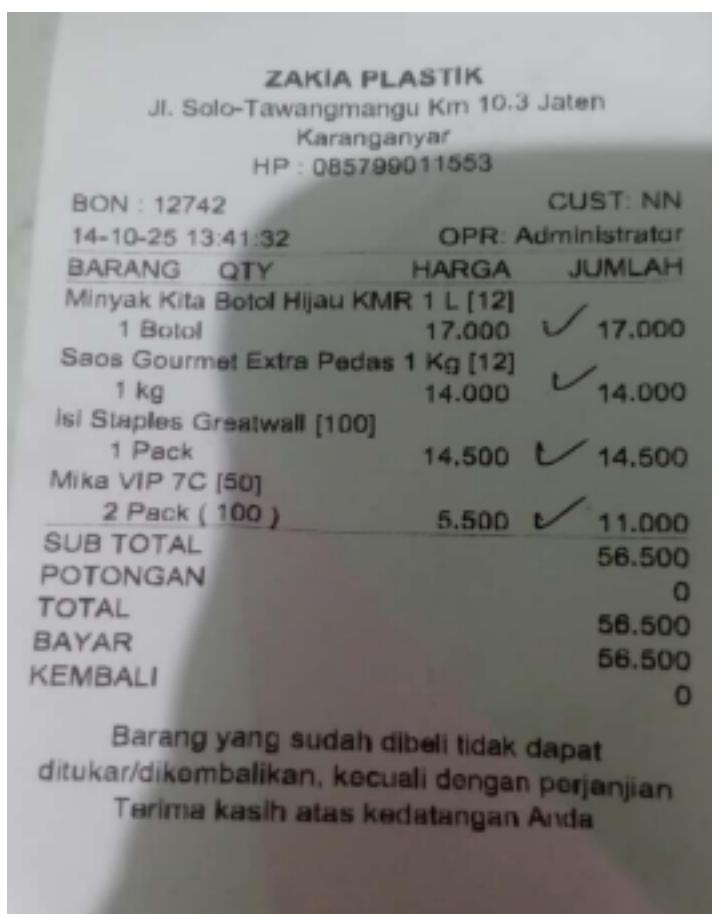
#### B. Alur Transaksi Belanja

Secara umum, alur proses transaksi belanja dalam sistem dapat dijelaskan sebagai berikut:

1. Pelanggan datang ke toko atau melakukan pemesanan barang melalui sistem (baik secara langsung maupun online).
2. Petugas mencatat data pelanggan dan memasukkan barang yang dibeli ke dalam sistem, termasuk jumlah dan harga masing-masing barang.

3. Sistem secara otomatis menghitung total pembayaran, termasuk diskon atau pajak (jika ada).
4. Pelanggan melakukan pembayaran, dan petugas membuat nota transaksi sebagai bukti pembelian.
5. Data transaksi tersimpan ke dalam basis data, sehingga dapat diakses kembali untuk keperluan laporan penjualan, analisis stok barang, atau verifikasi transaksi di kemudian hari

### C. Nota Transaksi Belanja



### D. Analisis Data

Dari hasil pengamatan, data barang, pelanggan, dan transaksi sering kali masih tercampur dalam satu tabel sehingga menimbulkan duplikasi dan kesulitan dalam pengelolaan data. Untuk menghindari pengulangan dan meningkatkan efisiensi penyimpanan, dilakukan proses normalisasi agar setiap tabel hanya berisi data yang relevan dengan kunci utamanya masing-masing. Dengan demikian, struktur basis data sistem transaksi belanja menjadi lebih efisien, konsisten, dan mudah dikelola.



### BAB III

#### PERANCANGAN BASIS DATA

##### **A. Tabel Umum (Unnormalized Table / Tabel Awal)**

Tabel awal yang berisi seluruh data sebelum dilakukan proses normalisasi masih mengandung banyak pengulangan dan belum memiliki struktur data yang terorganisasi.

NAMA BARANG	HARGA BARANG	JUMLAH HARGA	TANGGAL PEMBELIAN	JUMLAH BARANG	SUB TOTAL
minyak goreng	17.000	17.000	14/10/2025	1	
saus	14.000	14.000	14/10/2025	1	56.500
isi staples	14.500	14.500	14/10/2025	1	
mika	5.500	11.000	14/10/2025	2	

##### **B. Normalisasi 1NF (First Normal Form)**

Tabel awal sudah memenuhi ketentuan bentuk normal pertama (1NF) karena setiap kolom berisi satu nilai tunggal, tidak terdapat data yang berulang dalam satu kolom, dan setiap atribut hanya menyimpan satu jenis informasi. Dengan demikian, pada tahap 1NF tidak diperlukan perubahan struktur dari tabel awal.

NAMA BARANG	HARGA BARANG	JUMLAH HARGA	TANGGAL PEMBELIAN	JUMLAH BARANG	SUB TOTAL
minyak goreng	17.000	17.000	14/10/2025	1	
saus	14.000	14.000	14/10/2025	1	56.500
isi staples	14.500	14.500	14/10/2025	1	
mika	5.500	11.000	14/10/2025	2	

##### **C. Normalisasi 2NF (Second Normal Form)**

Pada tahap ini, tabel dipisahkan untuk menghilangkan ketergantungan parsial terhadap kunci utama. Setiap atribut harus bergantung penuh pada primary key. Karena itu, data dipecah menjadi beberapa tabel seperti tabel barang, tabel transaksi, tabel detail transaksi, dan tabel pelanggan. Pemisahan ini membuat struktur data lebih teratur, memudahkan pengelolaan, serta mencegah terjadinya duplikasi data dalam sistem transaksi.

ID PRODUK	NAMA PRODUK	JUMLAH PRODUK	HARGA PRODUK
brg-01	minyak goreng	1	17.000
brg-02	saus	1	14.000
brg-03	isi staples	1	14.500
brg-04	mika	2	11.000

id_transaksi	id_detail	nomor bon	tanggal	total	bayar	kembalian
TRX-01	DTL-001	1	14/10/2025	56.500,00	56.500,00	0
TRX-02	DTL-002	1	15/10/2025			
TRX-03	DTL-003	1	16/10/2025			
TRX-04	DTL-004	1	17/10/2025			
TRX-05	DTL-005	1	18/10/2025			

Id_Pelanggan	nama_Pelanggan
PLGN-001	Hasan

## D. Normalisasi 3NF (Third Normal Form)

Tahap ini bertujuan untuk menghilangkan ketergantungan antar atribut non-key agar data menjadi lebih efisien dan konsisten. Setelah melalui 2NF, masih ditemukan beberapa atribut yang tidak sepenuhnya bergantung pada kunci utama. Oleh karena itu, tabel kembali dipisahkan menjadi tabel barang, tabel pelanggan, tabel transaksi, dan tabel detail transaksi. Dengan pemisahan ini, setiap atribut hanya bergantung pada primary key masing-masing, sehingga struktur data menjadi lebih teratur, bebas duplikasi, dan lebih mudah dikelola dalam sistem transaksi belanja.

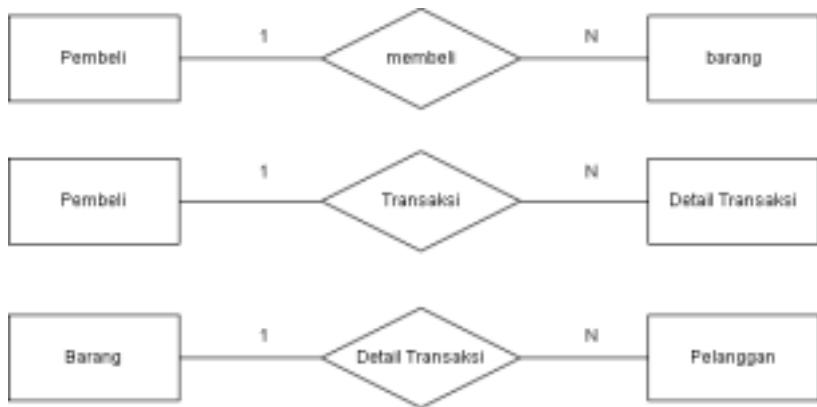
id barang	nama barang	satuan	harga satuan
BRG-001	minyak	botol	17.000
BRG-002	saus gorment	kg	14.000
BRG-003	isi staples	pack	14.500
BRG-004	Mika vip	pack	5.500

ID PRODUK	NAMA PRODUK	JUMLAH PRODUK	HARGA PRODUK
brg-01	minyak goreng	1	17.000
brg-02	saus	1	14.000
brg-03	isi staples	1	14.500
brg-04	mika	2	11.000

id transaksi	id detail	nomor bon	tanggal	total	bayar	kembali
TRX-01	DTL-001		14/10/2025	\$56.500,00	\$56.500,00	0
TRX-02	DTL-002		15/10/2025			
TRX-03	DTL-003		16/10/2025			
TRX-04	DTL-004		17/10/2025			
TRX-05	DTL-005		18/10/2025			

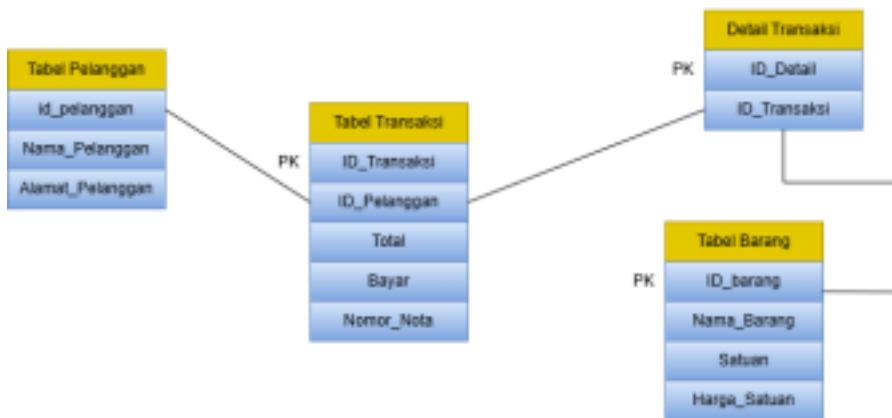
## E. Derajat Kardinalitas

Derajat kardinalitas menunjukkan hubungan antara dua entitas dalam basis data. Pada diagram di atas, setiap hubungan memiliki tipe satu ke banyak (1:N). Contohnya, satu pelanggan dapat melakukan banyak transaksi, satu transaksi dapat memiliki banyak detail pembelian, dan satu barang dapat muncul dalam banyak detail transaksi. Dengan demikian, entitas *Detail Transaksi* berperan sebagai penghubung antara entitas Transaksi dan Barang, sehingga seluruh proses pembelian dapat tercatat secara jelas dan terstruktur.



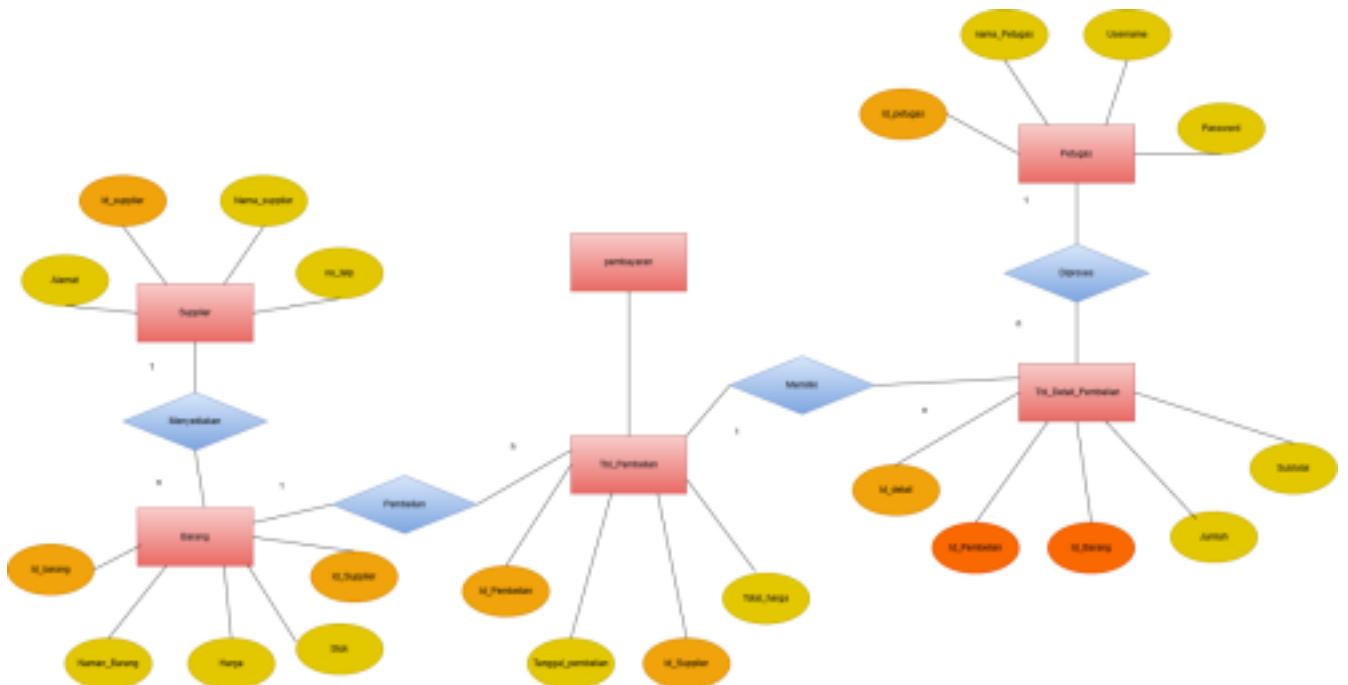
### F. Relasi Antar Tabel

Relasi antar tabel menunjukkan bahwa tabel Transaksi menjadi pusat hubungan data. Satu pelanggan, satu produk, dan satu metode pembayaran dapat terhubung ke banyak data transaksi, sedangkan satu toko dapat menyediakan banyak jenis produk. Semua hubungan bersifat satu ke banyak (1:N), sehingga data tersimpan lebih rapi dan saling terintegrasi.



### G. ERD (Entity Relationship Diagram)

ERD sistem informasi transaksi belanja ini menjelaskan hubungan antar tabel seperti pelanggan, kasir, produk, metode pembayaran, dan toko yang saling terhubung melalui tabel transaksi. Tabel transaksi menjadi pusat data yang mencatat aktivitas antara pelanggan, kasir, serta produk yang dibeli. Setiap pelanggan dapat melakukan beberapa transaksi, satu kasir dapat menangani banyak transaksi, dan satu toko



## H. Implementasi Basis Data

### 1. Implementasi Data Definition Language (DDL)

merupakan tahap awal dalam penerapan basis data yang bertujuan untuk membangun struktur database sesuai dengan hasil perancangan yang telah dibuat sebelumnya. Pada tahap ini, seluruh objek basis data didefinisikan menggunakan perintah SQL agar dapat digunakan oleh sistem transaksi Belanja

```

1
2 • CREATE DATABASE IF NOT EXISTS toko_db;
3 • USE toko_db;
4
5 • CREATE TABLE pelanggan (
6     id_pelanggan INT AUTO_INCREMENT PRIMARY KEY,
7     nama_pelanggan VARCHAR(100) NOT NULL,
8     alamat_pelanggan TEXT
9 ) ENGINE=InnoDB;
10
11 • CREATE TABLE supplier (
12     id_supplier INT AUTO_INCREMENT PRIMARY KEY,
13     nama_supplier VARCHAR(100) NOT NULL,
14     alamat TEXT,
15     telepon VARCHAR(20)

```

```

CREATE TABLE barang (
    id_barang INT AUTO_INCREMENT PRIMARY KEY,
    nama_barang VARCHAR(100) NOT NULL,
    satuan VARCHAR(50),
    harga_satuan DECIMAL(12,2) NOT NULL,
    id_supplier INT,
    FOREIGN KEY (id_supplier) REFERENCES supplier(id_supplier)
        ON UPDATE CASCADE
        ON DELETE SET NULL
) ENGINE=InnoDB;

CREATE TABLE transaksi (
    id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
    id_pelanggan INT NOT NULL,
    total DECIMAL(12,2),
    bayar DECIMAL(12,2),
    nomor_nota VARCHAR(50),
    tanggal_transaksi DATE,
    FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;

```

## Penjelasan:

Data Definition Language (DDL) adalah bagian dari bahasa SQL yang digunakan untuk mendefinisikan dan mengelola struktur basis data. DDL berfungsi untuk membuat, mengubah, dan menghapus objek-objek dalam basis data seperti database, tabel, kolom, dan constraint. DDL tidak digunakan untuk memanipulasi isi data, melainkan hanya mengatur kerangka atau struktur database.

Dalam perancangan sistem basis data, DDL digunakan pada tahap awal untuk menerjemahkan hasil perancangan konseptual seperti Entity Relationship Diagram (ERD) menjadi struktur tabel yang dapat diimplementasikan pada sistem manajemen basis data (DBMS). Dengan DDL, setiap tabel dapat ditentukan atributnya, tipe data, serta hubungan antar tabel melalui primary key dan foreign key.

DDL juga berperan penting dalam menjaga konsistensi dan integritas struktur database. Perintah DDL memungkinkan pengembang untuk menyesuaikan struktur database apabila terjadi perubahan kebutuhan sistem, tanpa harus membangun database dari awal. Oleh karena itu, DDL menjadi dasar utama dalam pembangunan dan pengelolaan database yang terstruktur dan sistematis.

Lalu selanjutnya membuat sebuah tabel dengan cara CREATE TABLE nama tabel yang ingin dibuat (...) didalam kurung di isi kolom kolom yang inign digunakan contohnya seperti pada gambar lalu menentukan tipe datanya apakah varchar, integer, date dll. Setelah di run maka hasilnya akan langsung terbuatlah database di PHP MyAdmin, dan akan ada notifikasi centang warna hijau seperti ini

#	Time	Action
2	00:38:28	USE toko_db
3	00:38:28	CREATE TABLE pelanggan ( id_pelanggan INT AUTO_INCREMENT PRIMARY KEY, nama_pelanggan V...
4	00:38:28	CREATE TABLE supplier ( id_supplier INT AUTO_INCREMENT PRIMARY KEY, nama_supplier VARCHA...
5	00:38:28	CREATE TABLE barang ( id_barang INT AUTO_INCREMENT PRIMARY KEY, nama_barang VARCHAR(100) NOT...
6	00:38:28	CREATE TABLE transaksi ( id_transaksi INT AUTO_INCREMENT PRIMARY KEY, id_pelanggan INT NOT...
7	00:38:28	CREATE TABLE detail_transaksi ( id_detail INT AUTO_INCREMENT PRIMARY KEY, id_transaksi INT NOT...

The screenshot shows the structure of the 'toko\_db' database in PHP MyAdmin. The database contains five tables: barang, detail\_transaksi, pelanggan, supplier, and transaksi. Each table has a green checkmark next to it, indicating successful creation. The 'New' button is also visible.

## 2. DML (Data Manipulation Language)

```
INSERT INTO pelanggan (nama_pelanggan, alamat_pelanggan) VALUES
('Andi Pratama', 'Jl. Merdeka No. 10'),
('Budi Santoso', 'Jl. Sudirman No. 25'),
('Citra Lestari', 'Jl. Diponegoro No. 5');

INSERT INTO supplier (nama_supplier, alamat, telepon) VALUES
('PT Sumber Jaya', 'Jl. Industri No. 1', '081234567890'),
('CV Makmur Sejahtera', 'Jl. Perdagangan No. 12', '082345678901');

INSERT INTO barang (nama_barang, satuan, harga_satuan, id_supplier) VALUES
('Beras 5 Kg', 'Karung', 65000, 1),
('Minyak Goreng 2 L', 'Botol', 32000, 1),
('Gula Pasir 1 Kg', 'Pack', 14000, 2);
```

Implementasi Data Manipulation Language (DML) pada database toko\_db digunakan untuk mengelola data yang tersimpan di dalam tabel setelah struktur database dibuat menggunakan Data Definition Language (DDL). DML diterapkan untuk memasukkan data pelanggan, supplier, barang, serta transaksi penjualan sehingga database dapat digunakan secara nyata dalam sistem administrasi toko. Proses ini memungkinkan penyimpanan data yang terstruktur dan saling terhubung antar tabel.

Pada implementasinya, data pelanggan dan supplier dimasukkan terlebih dahulu karena menjadi data induk bagi tabel lainnya. Data barang kemudian dimasukkan dengan menghubungkannya ke tabel supplier melalui atribut id\_supplier sebagai foreign key. Selanjutnya, data transaksi dicatat dengan mengaitkan pelanggan yang melakukan pembelian, termasuk informasi total pembayaran, jumlah bayar, nomor nota, dan tanggal transaksi. Setiap transaksi yang terjadi kemudian dirinci pada tabel detail\_transaksi, yang mencatat barang apa saja yang dibeli, jumlah barang, serta subtotal harga.

Implementasi DML harus dilakukan secara berurutan untuk menjaga integritas data dan mencegah pelanggaran aturan foreign key. Dengan penerapan DML yang benar, setiap data yang tersimpan dalam database saling berelasi dan dapat digunakan untuk menghasilkan informasi penjualan yang akurat dan terorganisir. Hal ini menjadikan database toko\_db berfungsi secara optimal sebagai sistem pendukung administrasi dan pencatatan transaksi toko.

## 3. TCL (Transaction Control Language)

Transaction Control Language (TCL) merupakan bagian dari SQL yang digunakan untuk mengatur dan mengendalikan transaksi dalam basis data. TCL berfungsi untuk memastikan bahwa setiap perubahan data yang dilakukan melalui perintah Data Manipulation Language (DML) berjalan secara konsisten, aman, dan dapat dikendalikan. Dengan TCL, pengguna dapat menentukan apakah perubahan data akan disimpan secara permanen atau dibatalkan.

```
• CREATE TABLE transaksi (
    id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
    id_pelanggan INT NOT NULL,
    total DECIMAL(12,2),
    bayar DECIMAL(12,2),
    nomor_nota VARCHAR(50),
    tanggal_transaksi DATE,
```

```

• CREATE TABLE detail_transaksi (
    id_detail INT AUTO_INCREMENT PRIMARY KEY,
    id_transaksi INT NOT NULL,
    id_barang INT NOT NULL,
    jumlah INT NOT NULL,
    subtotal DECIMAL(12,2),
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=InnoDB;

INSERT INTO transaksi (id_pelanggan, total, bayar, nomor_nota, tanggal_transaksi) VALUES
(1, 111000, 120000, 'NT-001', '2025-01-15'),
(2, 46000, 50000, 'NT-002', '2025-01-16');

INSERT INTO detail_transaksi (id_transaksi, id_barang, jumlah, subtotal) VALUES
(1, 1, 1, 65000),
(1, 2, 1, 32000),
(1, 3, 1, 14000),
(2, 3, 2, 28000),
(2, 2, 1, 32000);

```

Pada implementasinya dalam database *toko\_db*, TCL digunakan ketika terjadi proses transaksi penjualan yang melibatkan beberapa perintah DML, seperti memasukkan data ke tabel transaksi dan detail\_transaksi. Perintah COMMIT digunakan untuk menyimpan seluruh perubahan data secara permanen ke dalam database setelah transaksi berhasil dilakukan. Sebaliknya, perintah ROLLBACK digunakan untuk membatalkan seluruh perubahan apabila terjadi kesalahan selama proses transaksi, sehingga kondisi database kembali ke keadaan sebelumnya. Selain itu, perintah SAVEPOINT dapat digunakan untuk menandai titik tertentu dalam transaksi agar sebagian perubahan dapat dibatalkan tanpa memengaruhi keseluruhan transaksi.

Penerapan TCL sangat penting untuk menjaga konsistensi dan integritas data, terutama pada sistem administrasi toko yang melakukan banyak transaksi. Dengan menggunakan TCL, database *toko\_db* mampu memastikan bahwa setiap transaksi penjualan diproses secara utuh dan tidak menimbulkan data yang tidak lengkap atau inkonsisten.

#### 4. join

Query JOIN digunakan untuk menggabungkan data dari beberapa tabel yang saling berelasi.

Pada sistem ini, JOIN menghubungkan tabel transaksi dengan tabel pelanggan, kasir, dan barang, sehingga data transaksi dapat ditampilkan secara lengkap dalam satu tampilan, seperti nama pelanggan, nama kasir, jenis layanan, jumlah, dan tanggal transaksi. Dengan JOIN, informasi yang tersebar di beberapa tabel dapat disajikan menjadi satu data yang utuh dan mudah dibaca.

```

SELECT
    t.nomor_nota,
    t.tanggal_transaksi,
    p.nama_pelanggan,
    b.nama_barang,
    dt.jumlah,
    b.harga_satuan,
    dt.subtotal,
    t.total
FROM transaksi t
JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan
JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
JOIN barang b ON dt.id_barang = b.id_barang
ORDER BY t.tanggal_transaksi;

```

Penggabungan antara tabel transaksi dan pelanggan dilakukan untuk mengaitkan setiap transaksi dengan pelanggan yang melakukan pembelian. Tabel transaksi hanya menyimpan *id\_pelanggan* sebagai penanda, sehingga JOIN diperlukan untuk menampilkan informasi nama pelanggan secara nyata. Hal ini sangat penting dalam administrasi toko karena memungkinkan pihak toko mengetahui riwayat pembelian setiap pelanggan dan mempermudah proses pelayanan maupun pelaporan.

Selanjutnya, JOIN antara tabel transaksi dan detail\_transaksi berfungsi untuk menampilkan rincian barang yang dibeli dalam satu transaksi. Tabel transaksi bersifat sebagai data induk yang menyimpan informasi umum seperti nomor nota dan tanggal transaksi, sedangkan tabel detail\_transaksi menyimpan data item pembelian secara rinci. Tanpa JOIN ini, sistem tidak dapat mengetahui barang apa saja yang dibeli dalam suatu transaksi.

JOIN dengan tabel barang diperlukan untuk melengkapi informasi detail transaksi, seperti nama barang, harga satuan, dan satuan barang. Tabel detail\_transaksi hanya menyimpan *id\_barang*, sehingga JOIN diperlukan agar informasi barang dapat ditampilkan secara lengkap dan mudah dipahami. Penggabungan ini juga memungkinkan sistem untuk menghitung subtotal dan melakukan analisis penjualan berdasarkan jenis barang.

Secara keseluruhan, query JOIN ini menghasilkan laporan transaksi yang lengkap, mencakup identitas pelanggan, informasi transaksi, serta rincian barang yang dibeli. Query inilah yang paling sering digunakan dalam pembuatan laporan penjualan, pengecekan transaksi, dan pengambilan keputusan dalam pengelolaan toko. Oleh karena itu, JOIN ini dapat dianggap sebagai bagian paling penting dalam implementasi query JOIN pada database *toko\_db*.

## 5. Group By & Agregasi

Query GROUP BY dan agregasi digunakan untuk melakukan perhitungan data penjualan berdasarkan kelompok tertentu. Pada gambar yang ditampilkan, query menggabungkan tabel tabel\_detail\_transaksi dan tabel\_barang, kemudian mengelompokkan data berdasarkan nama barang. Fungsi agregasi **SUM()** digunakan untuk menghitung total jumlah barang yang terjual dari seluruh transaksi yang tersimpan dalam basis data.

```

SELECT
    t.nomor_nota,
    p.nama_pelanggan,
    SUM(dt.subtotal) AS total_transaksi
FROM transaksi t

```

## 6. Having Query

```
SELECT
    p.nama_pelanggan,
    SUM(dt.subtotal) AS total_belanja
FROM transaksi t
JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan
JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
GROUP BY p.id_pelanggan, p.nama_pelanggan
HAVING SUM(dt.subtotal) > 50000;
```

Pada query di atas, data transaksi dikelompokkan berdasarkan pelanggan sehingga setiap pelanggan hanya menghasilkan satu baris data. Fungsi agregasi SUM digunakan untuk menghitung total belanja masing-masing pelanggan berdasarkan subtotal seluruh barang yang dibeli. Klausula HAVING kemudian digunakan untuk menampilkan hanya pelanggan yang memiliki total belanja lebih dari nilai tertentu. Dengan demikian, query ini mampu menyaring hasil berdasarkan nilai agregasi yang telah dihitung.

Penggunaan HAVING ini sangat berguna dalam sistem administrasi toko, misalnya untuk mengetahui pelanggan dengan tingkat pembelian tinggi, membuat laporan pelanggan prioritas, atau menganalisis pola belanja pelanggan. Oleh karena itu, HAVING menjadi bagian penting dalam pengolahan data lanjutan setelah JOIN, GROUP BY, dan fungsi agregasi diterapkan.

## 7. Query Sub Query

Subquery adalah query yang berada di dalam query lain dan digunakan untuk menghasilkan nilai yang akan dipakai oleh query utama. Pada database *toko\_db*, subquery sangat berguna ketika ingin mengambil data berdasarkan hasil perhitungan atau kondisi tertentu yang tidak bisa diperoleh secara langsung dalam satu query sederhana.

```
-- =====
-- SUBQUERY--
SELECT
    nama_pelanggan
FROM pelanggan
WHERE id_pelanggan IN (
    SELECT id_pelanggan
    FROM transaksi
    GROUP BY id_pelanggan
    HAVING SUM(total) > 100000
);
```

Pada contoh di atas, subquery digunakan untuk mencari *id\_pelanggan* yang memiliki total transaksi lebih dari nilai tertentu. Subquery tersebut bekerja dengan mengelompokkan data transaksi berdasarkan

pelanggan dan menghitung total pembelian menggunakan fungsi agregasi SUM. Hasil dari subquery berupa daftar *id\_pelanggan* yang memenuhi syarat, kemudian digunakan oleh query utama untuk menampilkan nama pelanggan yang sesuai.

Penggunaan subquery ini penting dalam sistem administrasi toko karena memungkinkan analisis data yang lebih mendalam, seperti mengidentifikasi pelanggan dengan pembelian terbesar atau menentukan pelanggan prioritas. Dengan subquery, proses pengolahan data menjadi lebih fleksibel dan terstruktur karena hasil query dapat digunakan kembali sebagai dasar pengambilan data pada query lain.

## **BAB IV**

### **PENUTUP**

#### **A. Pengujian Dan Hasil**

Pengujian dilakukan untuk memastikan database *toko\_db* dapat berjalan dengan baik. Pengujian dimulai dengan pembuatan tabel dan relasi menggunakan perintah DDL, dan hasilnya seluruh tabel berhasil dibuat tanpa error. Selanjutnya dilakukan pengujian DML untuk memasukkan data ke dalam tabel, dan hasilnya data dapat disimpan dengan baik sesuai dengan relasi antar tabel.

Pengujian query JOIN menunjukkan bahwa data transaksi dapat ditampilkan secara lengkap dengan menggabungkan beberapa tabel yang saling berhubungan. Pengujian GROUP BY, fungsi agregasi, HAVING, dan subquery juga berjalan dengan baik dan menghasilkan data yang sesuai dengan kebutuhan laporan. Secara keseluruhan, hasil pengujian menunjukkan bahwa database *toko\_db* berfungsi dengan baik dan siap digunakan sebagai sistem administrasi toko.

#### **B. kendala dan Perbaikan**

Selama proses pembuatan dan pengujian database *toko\_db*, terdapat beberapa kendala yang ditemui, seperti kesalahan saat memasukkan data akibat pelanggaran foreign key dan ketidaksesuaian urutan input data. Kendala ini terjadi karena data pada tabel induk belum tersedia ketika data dimasukkan ke

tabel yang memiliki relasi.

Selain itu, kendala juga muncul pada saat penulisan query JOIN, GROUP BY, dan HAVING, terutama dalam menentukan kolom yang harus dikelompokkan agar hasil query sesuai dan tidak menimbulkan error. Beberapa query awal juga menghasilkan data yang tidak akurat karena kesalahan dalam penggunaan fungsi agregasi.

Perbaikan dilakukan dengan memastikan urutan input data yang benar, yaitu mengisi tabel induk terlebih dahulu sebelum tabel turunan, serta menyesuaikan penulisan query sesuai dengan struktur relasi antar tabel. Setelah dilakukan perbaikan, seluruh query dapat dijalankan dengan baik dan database *toko\_db* berfungsi sesuai dengan tujuan perancangan.

### C. Kesimpulan

Dari hasil perancangan dan analisis yang telah dilakukan, dapat disimpulkan bahwa sistem basis data transaksi belanja mampu meningkatkan efisiensi dan akurasi dalam pencatatan seluruh aktivitas belanja, termasuk data pelanggan, kasir, transaksi, produk, dan metode pembayaran. Dengan menerapkan proses normalisasi hingga bentuk 3NF, struktur data menjadi lebih teratur, minim duplikasi, dan lebih mudah dikelola, sehingga memudahkan pemeliharaan serta pengembangan sistem di masa mendatang. Entity Relationship Diagram (ERD) yang dirancang memperjelas hubungan antar entitas, menegaskan peran tabel transaksi sebagai pusat data yang menghubungkan pelanggan, produk, dan kasir. Dengan struktur ini, sistem informasi transaksi belanja dapat berjalan secara terintegrasi, mendukung pengambilan keputusan, mempercepat proses administrasi, serta memastikan data penjualan tercatat secara akurat dan dapat dipertanggungjawabkan.

### D. Saran

Dalam pengembangan selanjutnya, sistem ini dapat dikembangkan menjadi aplikasi berbasis web atau desktop agar dapat digunakan langsung oleh pihak toko atau kasir. Selain itu, dapat ditambahkan fitur pencatatan pembayaran secara real-time, laporan penjualan dan keuangan, manajemen stok produk, serta keamanan data pelanggan untuk meningkatkan fungsi, keandalan, dan keamanan sistem secara keseluruhan. Dengan pengembangan ini, sistem informasi transaksi belanja akan lebih efektif dalam mendukung operasional toko, mempercepat proses penjualan, serta mempermudah pengambilan keputusan bisnis.