

**LAPORAN PROJECT UJIAN AKHIR SEMESTER
DATABASE KASIR**

Disusun guna memenuhi tugas Ujian Akhir Semeser Pada Mata Kuliah Basis Data.I



**UNIVERSITAS
DUTA BANGSA
SURAKARTA**

Dosen Pengampu :

RIDWAN DWI IRAWAN, S. KOM., M. KOM.

Disusun Oleh :

CATUR PAMUNGKAS 250119005

RIQHAN BIMA SAPUTRA 250119022

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DUTA BANGSA SURAKARTA
2026**

DAFTAR ISI

LAPORAN PROJECT UJIAN AKHIR SEMESTER.....	1
DAFTAR ISI	2
BAB I	4
PENDAHULUAN	4
BAB II.....	8
LANDASAN TEORI.....	8
2.1 Pengantar basis data dan DBMS	8
2.2 Konsep Basis Data Relasional	8
2.3 Relasi Tabel dan Konsep Relasi Basis Data.....	9
2.4 Entity Relationship Diagram (ERD)	9
2.5 Normalisasi Basis Data	10
2.6 Dasar Structured Query Language (SQL).....	10
BAB III	11
3.1 Kebutuhan Sisitem	11
3.2 Perancangan Database.....	12
3.2.1 Entity Relationship Diagram (ERD)	12
3.2.2 Skema Tabel, Primary Key, dan Foreign Key	12
3.2.3 Penerapan ERD ke DBMS (Forward Engineering di MYSQL Workbench)	15
3.3 Implementasi Basis data dan Query SQL	19
3.3.1 Implementasi Data Definition Language (DDL)	19

3.3.2 Implementasi Data Manipulation Language (DML).....	22
3.3.3 Tabel TCL (Transaction)	30
3.3.4 Implementasi (JOIN, GROUP BY, HAVING, Agregasi).....	32
BAB IV	37
4.1 Ringkasan Implementasi Basis Data.....	37
4.2 Ringkasan Manipulasi Data (DML).....	37
4.3 Ringkasan Pengujian Query dan Analisis Data.....	38
4.4 Pengujian Transaksi (Transaction Control Language)	38
4.5 Kendala dan Upaya Perbaikan	39
4.6 Kesimpulan	39
LAMPIRAN.....	41
DAFTAR PUSTAKA.....	46

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi menyebabkan kebutuhan pengelolaan data bisnis menjadi semakin kompleks dan besar, termasuk dalam proses penjualan barang. Basis data relasional merupakan metode yang telah terbukti mampu mengelola data secara efisien, mengurangi redundansi data, dan menjaga konsistensi data antar tabel melalui relasi yang eksplisit antara entitas data. Proses normalisasi menjadi tahapan penting dalam perancangan basis data untuk meningkatkan kualitas struktur data sehingga dapat meminimalkan terjadinya anomali pada operasi insert, update, dan delete serta meningkatkan integritas data(1).

Dalam studi kasus sistem penjualan, data awal transaksi sering kali masih berbentuk dokumen nota yang tidak terstruktur dan mengandung redundansi. Oleh karena itu, perlu dilakukan perancangan basis data yang sistematis melalui pemodelan Entity Relationship Diagram (ERD) dan normalisasi hingga bentuk normal ketiga (3NF) sebelum diterapkan ke dalam sistem manajemen basis data seperti MySQL, sehingga setiap entitas dan relasi antar entitas tersusun dengan baik dan dapat diimplementasikan secara konsisten(2).

1.2 Pengantar Pemrograman Basis Data

Pemrograman basis data merupakan proses perancangan dan pengelolaan data menggunakan sistem manajemen basis data agar data dapat disimpan secara terstruktur, konsisten, dan mudah diakses. Dalam sistem informasi, basis data berperan sebagai komponen utama yang mendukung penyimpanan dan pengolahan data secara terintegrasi sehingga dapat mengurangi kesalahan pencatatan dan duplikasi data. Penerapan basis data

yang terstruktur memungkinkan sistem menghasilkan informasi yang akurat dan relevan untuk mendukung pengambilan Keputusan. (1)

Basis data relasional merupakan model yang paling umum digunakan dalam pemrograman basis data karena mampu merepresentasikan data dalam bentuk tabel-tabel yang saling berelasi. Setiap tabel merepresentasikan satu entitas dengan atribut tertentu, sedangkan hubungan antar tabel dibangun melalui kunci primer dan kunci tamu. Model relasional dinilai efektif karena mampu menjaga integritas data dan memudahkan pengelolaan hubungan antar entitas dalam sistem informasi yang kompleks(3).

Bahasa yang digunakan dalam pemrograman basis data relasional adalah Structured Query Language (SQL). SQL menyediakan perintah untuk mendefinisikan struktur basis data melalui Data Definition Language (DDL), mengelola data menggunakan Data Manipulation Language (DML), serta menjaga konsistensi transaksi melalui Transaction Control Language (TCL). Penggunaan SQL yang terstruktur memungkinkan sistem basis data beroperasi secara konsisten dan aman, terutama pada sistem yang melibatkan banyak transaksi(3).

Dalam implementasinya, MySQL banyak digunakan sebagai DBMS karena bersifat open source, mendukung standar SQL, serta memiliki performa yang baik untuk sistem berskala kecil hingga menengah. MySQL Workbench digunakan sebagai alat bantu visual untuk memodelkan basis data, membuat ERD, serta melakukan proses *forward engineering* dan pengujian query. Kombinasi MySQL dan MySQL Workbench mempermudah proses perancangan hingga implementasi basis data secara menyeluruh(3).

1.3 Latar Belakang Studi Kasus

Studi kasus dalam proyek ini berfokus pada sistem penjualan toko CCTV yang melibatkan berbagai data operasional, seperti data pelanggan,

kasir, produk, transaksi, detail transaksi, dan pembayaran. Dalam praktiknya, pencatatan transaksi masih dilakukan menggunakan nota penjualan yang belum terstruktur dengan baik. Kondisi ini menyebabkan data sulit dikelola secara efisien serta berpotensi menimbulkan kesalahan pencatatan dan kesulitan dalam penyusunan laporan penjualan.

Nota penjualan sebagai dokumen sumber masih mencampurkan berbagai jenis data dalam satu format, seperti informasi pelanggan, daftar produk, jumlah pembelian, dan total pembayaran. Struktur data seperti ini belum memenuhi prinsip basis data relasional sehingga diperlukan proses analisis lebih lanjut untuk memisahkan data ke dalam tabel-tabel sesuai dengan entitasnya masing-masing.

1.4 Tujuan Proyek

Tujuan dari proyek perancangan basis data ini adalah untuk merancang sistem basis data penjualan toko CCTV yang terstruktur dan efisien. Sistem yang dirancang diharapkan mampu mengelola data penjualan secara terintegrasi sehingga dapat mendukung proses transaksi dan pelaporan dengan lebih baik. Secara khusus, tujuan proyek ini meliputi:

- a) Menganalisis kebutuhan data berdasarkan dokumen nota penjualan.
- b) Melakukan normalisasi data hingga bentuk normal ketiga (3NF).
- c) Membuat Entity Relationship Diagram (ERD) sebagai model konseptual basis data.
- d) Mengimplementasikan hasil perancangan ke dalam DBMS MySQL.
- e) Menguji basis data menggunakan query SQL untuk memastikan sistem berjalan sesuai rancangan.

1.5 Ruang Lingkup dan Batasan Masalah

Ruang lingkup proyek ini dibatasi pada perancangan dan implementasi basis data sistem penjualan toko CCTV. Pembahasan difokuskan pada analisis data, normalisasi, pemodelan ERD, serta pembuatan relasi tabel menggunakan DBMS MySQL.

Proyek ini tidak mencakup pengembangan antarmuka pengguna (user interface) maupun aspek keamanan lanjutan seperti enkripsi data dan manajemen hak akses. Data yang digunakan merupakan data contoh yang disesuaikan dengan kebutuhan studi kasus untuk keperluan pembelajaran.

1.6 Gambaran Umum Sistem

Sistem basis data yang dirancang bertujuan untuk mendukung pengelolaan data penjualan toko CCTV secara terintegrasi. Sistem ini mencakup penyimpanan data pelanggan, kasir, produk, transaksi, detail transaksi, dan pembayaran yang saling terhubung melalui relasi tabel.

Pemodelan sistem dilakukan menggunakan Entity Relationship Diagram (ERD) sebagai representasi konseptual hubungan antar entitas. ERD kemudian diimplementasikan ke dalam basis data relasional menggunakan MySQL dan MySQL Workbench. Implementasi ini memungkinkan sistem diuji melalui eksekusi query SQL untuk memastikan bahwa struktur basis data telah sesuai dengan kebutuhan sistem.

BAB II

LANDASAN TEORI

2.1 Pengantar basis data dan DBMS

Basis data merupakan kumpulan data yang saling berhubungan dan disimpan secara sistematis untuk mendukung proses pengolahan data dalam suatu sistem informasi. Basis data dirancang agar data dapat disimpan secara terstruktur sehingga mudah diakses, diperbarui, dan dikelola secara efisien. Penggunaan basis data bertujuan untuk mengurangi redundansi data, menjaga konsistensi, serta meningkatkan kualitas informasi yang dihasilkan oleh sistem.(1)

Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk mengelola basis data, mulai dari proses penyimpanan, pemanggilan, pembaruan, hingga pengamanan data. DBMS menyediakan fasilitas untuk menjaga integritas data serta mengatur hak akses pengguna. Dengan adanya DBMS, pengelolaan basis data dapat dilakukan secara terpusat dan terkontrol sehingga meminimalkan kesalahan pengolahan data.(4)

2.2 Konsep Basis Data Relasional

Basis data relasional merupakan model basis data yang menyimpan data dalam bentuk tabel-tabel yang saling berhubungan. Setiap tabel terdiri atas baris (tuple) dan kolom (atribut) yang merepresentasikan suatu entitas dan karakteristiknya. Model relasional menggunakan konsep kunci primer (*primary key*) untuk mengidentifikasi data secara unik dan kunci tamu (*foreign key*) untuk membangun hubungan antar tabel(1).

Relasi antar tabel dalam basis data relasional memungkinkan data disimpan secara terpisah namun tetap saling terhubung. Pendekatan ini memudahkan pengelolaan data dalam sistem yang kompleks serta mendukung

proses pengambilan informasi melalui penggabungan data dari beberapa tabel. Model relasional dinilai efektif karena mampu menjaga integritas data dan mendukung fleksibilitas dalam pengolahan data. (3)

2.3 Relasi Tabel dan Konsep Relasi Basis Data

Relasi tabel merupakan hubungan logis antar tabel dalam basis data relasional yang dibentuk melalui penggunaan primary key dan foreign key. Relasi ini menggambarkan keterkaitan antar entitas dan memungkinkan data disusun secara terstruktur. Jenis relasi yang umum digunakan meliputi relasi satu ke satu (one-to-one), satu ke banyak (one-to-many), dan banyak ke banyak (many-to-many). (1)

Penerapan relasi basis data bertujuan untuk menjaga konsistensi dan integritas data. Dengan relasi yang tepat, perubahan data pada satu tabel dapat memengaruhi tabel lain secara terkontrol sehingga menghindari terjadinya data yatim (*orphan data*). Konsep ini sangat penting dalam sistem transaksi yang melibatkan banyak entitas seperti pelanggan, produk, dan transaksi. (3)

2.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model konseptual yang digunakan untuk menggambarkan struktur basis data serta hubungan antar entitas secara visual. ERD membantu perancang sistem dalam memahami kebutuhan data dan relasi antar entitas sebelum diimplementasikan ke dalam basis data relasional. Komponen utama ERD meliputi entitas, atribut, dan relasi. (4) (1)

Dalam perancangan basis data, ERD berfungsi sebagai jembatan antara kebutuhan sistem dan implementasi fisik basis data. ERD yang baik akan mempermudah proses normalisasi dan penerjemahan ke dalam struktur tabel pada DBMS. Oleh karena itu, ERD merupakan tahap penting yang tidak dapat dilewatkan dalam proses perancangan basis data. (3)

2.5 Normalisasi Basis Data

Normalisasi adalah proses penyusunan struktur tabel dalam basis data untuk mengurangi redundansi dan mencegah anomali data. Normalisasi dilakukan melalui beberapa tingkatan bentuk normal, yaitu First Normal Form (1NF), Second Normal Form (2NF), dan Third Normal Form (3NF). Proses ini memastikan bahwa setiap tabel memiliki struktur yang efisien dan bebas dari ketergantungan atribut yang tidak perlu.(5)

Normalisasi hingga bentuk normal ketiga (3NF) bertujuan untuk menghilangkan ketergantungan transitif antar atribut non-kunci. Dengan demikian, setiap atribut non-kunci hanya bergantung pada primary key tabel tersebut. Penerapan normalisasi 3NF terbukti meningkatkan kualitas desain basis data serta memudahkan pemeliharaan data dalam jangka Panjang. (3)

2.6 Dasar Structured Query Language (SQL)

Structured Query Language (SQL) merupakan bahasa standar yang digunakan untuk berinteraksi dengan basis data relasional. SQL menyediakan perintah untuk mendefinisikan struktur basis data melalui Data Definition Language (DDL), mengelola data menggunakan Data Manipulation Language (DML), serta mengatur transaksi melalui Transaction Control Language (TCL). Penggunaan SQL memungkinkan pengolahan data dilakukan secara terstruktur dan konsisten. (1)(4)

Selain itu, SQL juga mendukung pengolahan data melalui fungsi agregasi, GROUP BY, HAVING, dan JOIN. Fitur-fitur tersebut memungkinkan pengguna untuk menghasilkan laporan dan analisis data dari beberapa tabel yang saling berelasi. Oleh karena itu, pemahaman dasar SQL merupakan bagian penting dalam pemrograman basis data. (5)

BAB III

PERANCANGAN DAN IMPLEMENTASI

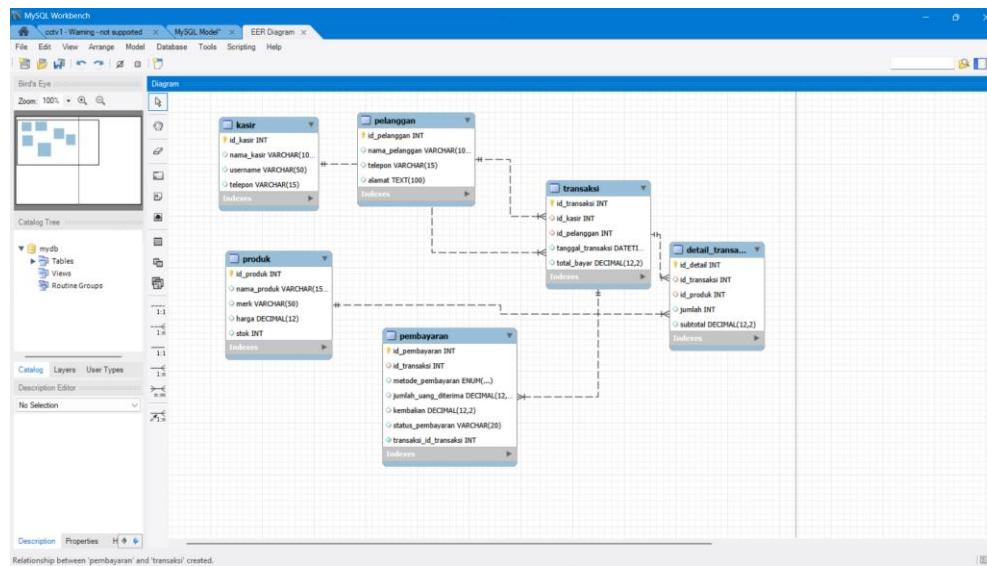
3.1 Kebutuhan Sisitem

Kebutuhan sistem dianalisis untuk menentukan fungsi utama yang harus didukung oleh basis data yang dirancang. Sistem ini ditujukan untuk mengelola data penjualan pada toko CCTV secara terstruktur dan terintegrasi. Berdasarkan analisis kebutuhan, sistem basis data harus mampu menyimpan dan mengelola data pelanggan, kasir, produk CCTV, transaksi penjualan, detail transaksi, serta pembayaran.

Secara fungsional, sistem basis data harus mendukung proses pencatatan transaksi penjualan, penyimpanan data master (pelanggan, kasir, dan produk), serta penyajian informasi melalui query SQL seperti laporan transaksi dan rekapitulasi penjualan. Sistem juga harus mampu menjaga konsistensi data melalui penerapan relasi antar tabel dan mekanisme transaksi.

3.2 Perancangan Database

3.2.1 Entity Relationship Diagram (ERD)



Gambar 3. 1 ERD Kasir

Relasi antar entitas pada sistem penjualan toko CCTV menggunakan relasi one-to-many (1:N), di mana satu entitas induk dapat berelasi dengan banyak entitas turunan. Pemilihan relasi ini sesuai dengan karakteristik sistem transaksi, seperti satu kasir yang dapat melayani banyak transaksi dan satu transaksi yang dapat memiliki banyak detail transaksi.

3.2.2 Skema Tabel, Primary Key, dan Foreign Key

Setelah Entity Relationship Diagram (ERD) dirancang, tahap selanjutnya adalah menerjemahkan ERD tersebut ke dalam skema tabel relasional. Skema tabel ini berfungsi sebagai rancangan struktur database yang akan diimplementasikan ke dalam DBMS MySQL. Setiap tabel memiliki Primary Key (PK) sebagai identitas unik dan Foreign Key (FK) sebagai penghubung antar tabel sesuai relasi yang telah ditentukan pada ERD.

1. Tabel Kasir

Tabel kasir digunakan untuk menyimpan data petugas atau staf yang melayani transaksi penjualan.

Tabel 3. 1 Tabel kasir

Nama Field	Tipe Data	panjang	Keterangan
Id_kasir	INT	-	Primary Key, Auto Increment
nama_kasir	VARCHAR	100	-
username	VARCHAR	50	-
telepon	VARCHAR	15	-

Sebagai referensi petugas yang bertanggung jawab atas setiap transaksi penjualan.

2. Tabel Pelanggan

Tabel pelanggan menyimpan data pelanggan yang melakukan pembelian produk CCTV.

Tabel 3. 2 tabel pelanggan

Nama Field	Tipe Data	panjang	Keterangan
Id_pelanggan	INT	-	Primary Key, Auto Increment
nama_pelanggan	VARCHAR	100	-
telepon	VARCHAR	15	-
alamat	TEXT	-	-

Mencatat identitas pelanggan yang terlibat dalam transaksi penjualan.

3. Table Produk

Tabel produk digunakan untuk menyimpan data barang atau produk CCTV yang dijual.

Tabel 3. 3 Tabel Produk

Nama Field	Tipe Data	panjang	Keterangan
Id_produk	INT	-	Primary Key, Auto Increment

nama_produk	VARCHAR	150	-
merk	VARCHAR	50	Not null
harga	DECIMAL	12,2	-
stok	INT		Not Null

Sebagai master data barang yang akan digunakan dalam transaksi penjualan.

4. Table Transaksi

Tabel transaksi merupakan tabel utama (header transaksi) yang mencatat informasi umum setiap transaksi penjualan.

Tabel 3. 4 tabel Transaksi

Nama Field	Tipe Data	panjang	Keterangan
Id_transaksi	INT	-	Primary Key, Auto Increment
id_kasir	INT	-	FK, NN
id_pelanggan	INT	-	FK, NN
tanggal_transaksi	DATETIME		NN
Total_bayar	DECIMAL	12,2	NN

Menyimpan data transaksi secara umum dan menghubungkan kasir serta pelanggan.

5. Tabel Detail_transaksi

Tabel detail_transaksi berfungsi sebagai tabel rincian item yang dibeli dalam satu transaksi.

Tabel 3. 5 tabel Detail Transaksi

Nama Field	Tipe Data	panjang	Keterangan
Id_detail	INT	-	Primary Key, Auto Increment
id_transaksi	INT	-	FK, NN
Id_produk	INT	-	FK, NN
jumlah	INT	-	FK, NN
subtotal	DECIMAL	12,2	NN

Mengatasi relasi many-to-many antara transaksi dan produk serta menyimpan rincian pembelian.

6. Tabel Pembayaran

Tabel pembayaran menyimpan informasi metode dan status pembayaran dari setiap transaksi.

Tabel 3. 6 tabel Pembayaran

Nama Field	Tipe Data	panjang	Keterangan
Id_pembayaran	INT	-	Primary Key, Auto Increment
Id_transaksi	INT	-	NN
Metode_pembayaran	ENUM	(Tunai, Transfer, Debit, Kredit)	NN
Jumlah_uang_diterima	DECIMAL	12,2	NN
kembalian	DECIMAL	12,2	NN
Status_pembayaran	VARCHAR	20	Lunas

Mencatat detail proses pembayaran dan memastikan konsistensi data transaksi.

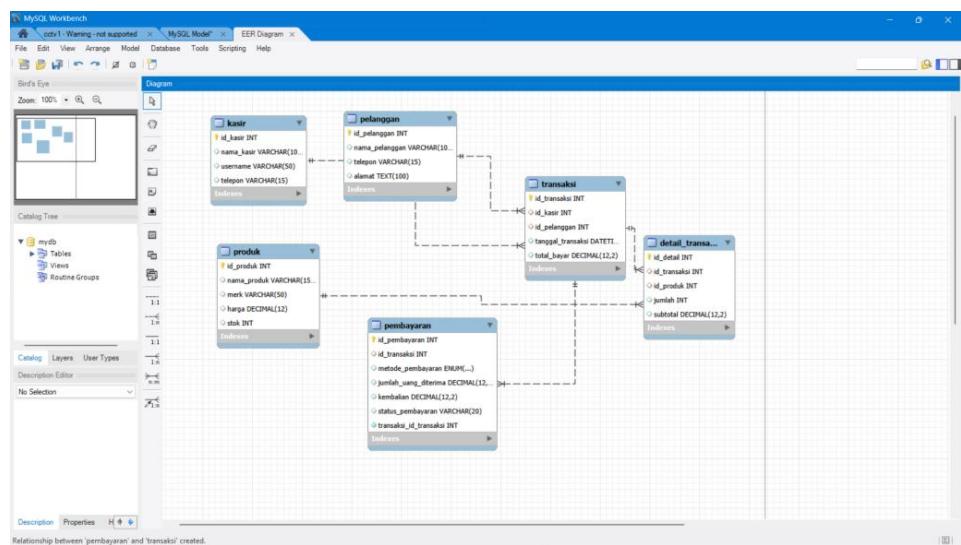
3.2.3 Penerapan ERD ke DBMS (Forward Engineering di MySQL Workbench)

Penerapan Entity Relationship Diagram (ERD) ke dalam Database Management System (DBMS) merupakan tahap implementasi dari perancangan konseptual menjadi perancangan fisik basis data. Pada tahap ini, seluruh entitas, atribut, serta relasi yang telah dirancang pada ERD diterjemahkan menjadi struktur tabel yang dapat dijalankan secara langsung pada DBMS MySQL.

Proses implementasi ERD dilakukan menggunakan MySQL Workbench sebagai alat bantu utama. MySQL Workbench dipilih karena

menyediakan fasilitas perancangan visual EER Diagram serta fitur Forward Engineering yang memungkinkan konversi model ERD menjadi skrip SQL secara otomatis. Dengan adanya fitur ini, risiko kesalahan dalam penulisan struktur tabel dan constraint dapat diminimalkan serta memastikan konsistensi antara desain dan implementasi.

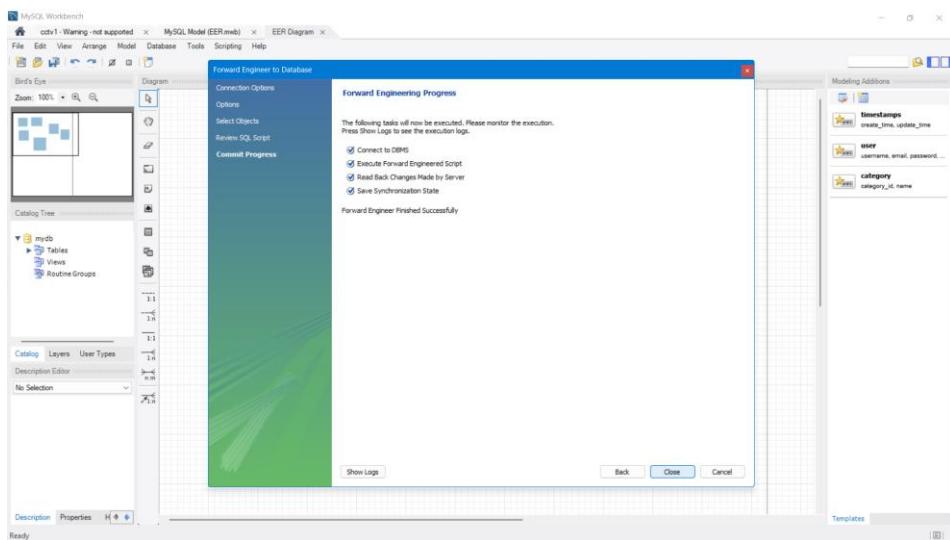
Tahapan penerapan ERD ke DBMS diawali dengan pembuatan dan finalisasi ERD Diagram pada MySQL Workbench. Pada tahap ini, setiap entitas didefinisikan sebagai tabel lengkap dengan atribut, tipe data, penentuan *Primary Key*, serta relasi antar tabel yang merepresentasikan *Foreign Key* sesuai dengan kardinalitas yang telah dirancang sebelumnya.



Gambar 3. 2 EER Diagram

Setelah perancangan ERD selesai, proses dilanjutkan dengan menjalankan fitur Forward Engineering pada MySQL Workbench. Pada tahap ini, pengguna melakukan konfigurasi koneksi ke server MySQL dan menentukan skema database tujuan. MySQL Workbench kemudian menghasilkan skrip SQL berupa perintah *Data Definition Language*

(DDL) yang mencakup pembuatan database, tabel, serta constraint seperti *Primary Key* dan *Foreign Key* secara otomatis berdasarkan ERD yang telah dirancang.



Gambar 3. 3 Konfigurasi Fordward Enineering

Skrip SQL hasil proses Forward Engineering selanjutnya dieksekusi langsung ke DBMS MySQL. Apabila proses eksekusi berjalan tanpa kesalahan, maka seluruh tabel beserta relasinya akan terbentuk secara otomatis sesuai dengan desain ERD. Proses ini menandai bahwa tahap implementasi struktur basis data telah berhasil dilakukan.

Tahap akhir dari penerapan ERD ke DBMS adalah verifikasi struktur database. Verifikasi dilakukan dengan meninjau daftar tabel pada *Schemas* di MySQL Workbench maupun melalui DBMS lain seperti phpMyAdmin. Pada tahap ini dilakukan pemeriksaan terhadap struktur kolom, tipe data, *Primary Key*, *Foreign Key*, serta *index* untuk

memastikan bahwa seluruh struktur database telah sesuai dengan rancangan ERD awal.

The screenshot shows the MySQL Workbench interface with the 'cctv_kasir' schema selected. The 'Tables' tab is active, displaying six tables: detail_transaksi, kasir, produk, pembayaran, transaksi, and transaksi_kasir. Each table has its name, engine (InnoDB), version (10), row format (Dynamic), rows (0), avg row length (0), data length (16.0 KB), max data length (32 KB), index length (0 bytes), data free (0.0 bytes), auto increment (1), create time (2024-01-14 01:40:21), and update time (2024-01-14 01:40:21). The 'Output' tab shows the history of SQL queries executed on the database, including SELECT statements and a database creation command.

Gambar 3. 4 Struktur Tabel Pada Database

The screenshot shows the MySQL Workbench interface with the 'cctv_kasir' schema selected. The 'Indexes' tab is active, displaying 12 indexes across various tables: detail_transaksi, and transaksi. Each index has its name, unique status (Yes or No), index type (BTREE or BTREE), column names, and other properties like comment, seq in index, packed, cardinality, and nullability. The 'Output' tab shows the history of SQL queries executed on the database, including SELECT statements and a database creation command.

Gambar 3. 5 Struktur Index Pada Database

Dengan selesainya tahap penerapan ERD ke DBMS ini, basis data sistem penjualan toko CCTV telah siap digunakan untuk tahap

selanjutnya, yaitu implementasi manipulasi data menggunakan perintah *Data Manipulation Language* (DML), pengelolaan transaksi menggunakan *Transaction Control Language* (TCL), serta pengujian query seperti JOIN, GROUP BY, dan agregasi.

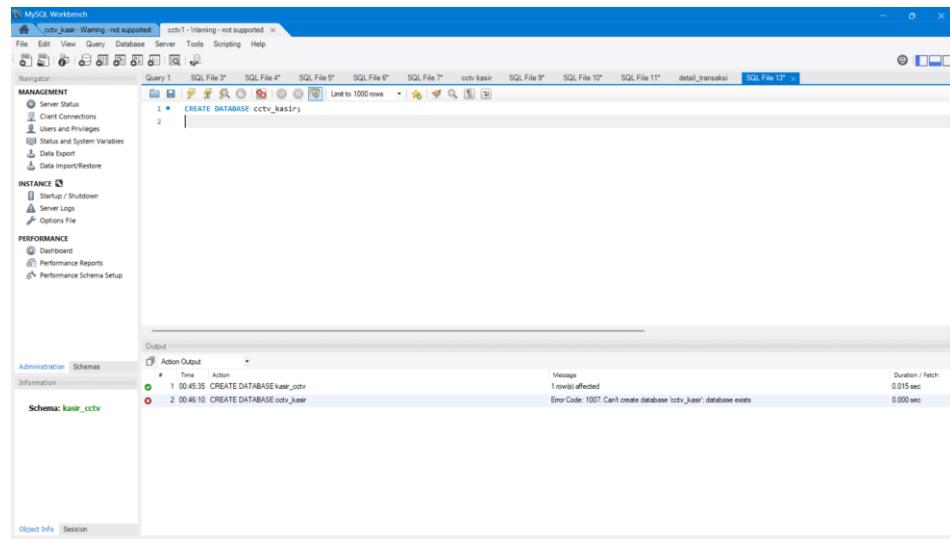
3.3 Implementasi Basis data dan Query SQL

Tahap implementasi basis data merupakan lanjutan dari penerapan Entity Relationship Diagram (ERD) ke dalam Database Management System (DBMS). Pada tahap ini, struktur database yang telah terbentuk digunakan untuk melakukan pembuatan tabel secara fisik, pengisian data awal, pengelolaan transaksi, serta pengujian query SQL untuk memastikan basis data dapat digunakan sesuai dengan kebutuhan sistem penjualan toko CCTV.

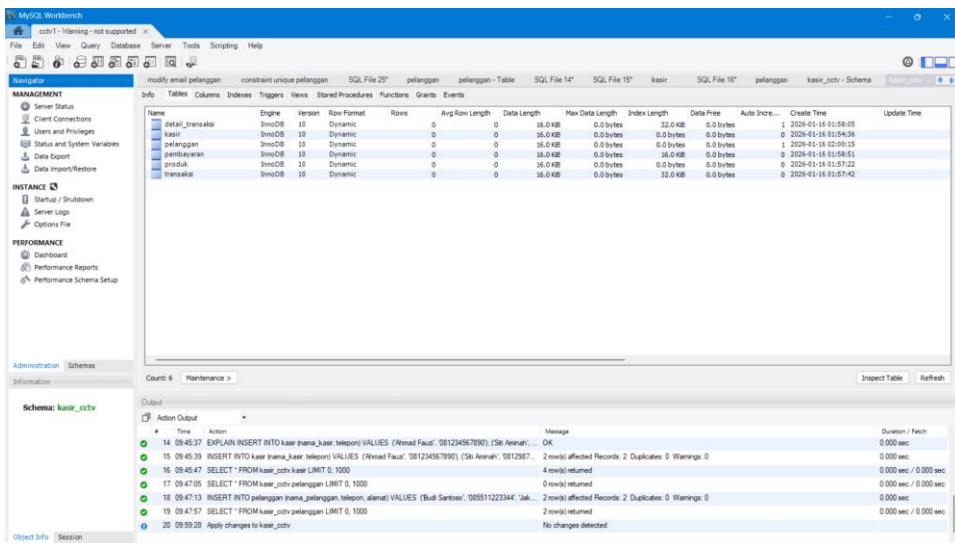
Implementasi dilakukan menggunakan DBMS MySQL dengan bantuan MySQL Workbench sebagai alat pengelolaan database. Implementasi meliputi penggunaan perintah Data Definition Language (DDL), Data Manipulation Language (DML), serta Transaction Control Language (TCL), sesuai dengan materi uliah Pemrograman Basis Data.

3.3.1 Implementasi Data Definition Language (DDL)

Implementasi Data Definition Language (DDL) dilakukan untuk membangun struktur basis data sesuai dengan perancangan Entity Relationship Diagram (ERD) yang telah dibuat sebelumnya. Proses ini bertujuan untuk mendefinisikan skema database, tabel, atribut, serta constraint yang diperlukan agar data dapat tersimpan secara terstruktur dan konsisten.



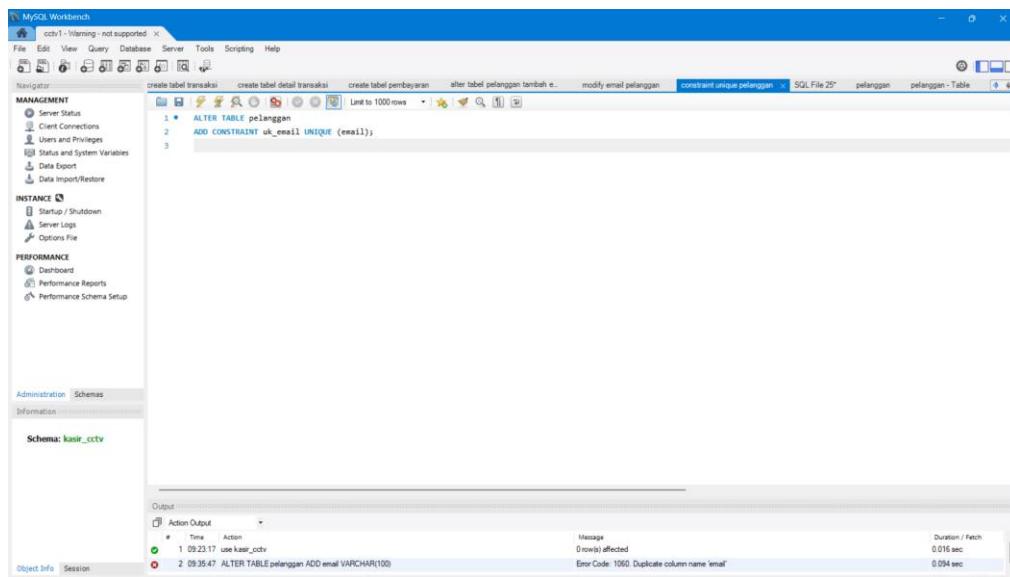
Gambar 3. 6 Create Database



Gambar 3. 7 Struktur Tabel pada Database

Gambar 3. 8 Column Tabel Pada Database

Gambar 3. 9 Index Pada Tabel



Gambar 3. 10 Menambahkan Constraint UNIQUE

3.3.2 Implementasi Data Manipulation Language (DML)

Setelah struktur database berhasil dibentuk melalui penerapan Data Definition Language (DDL), tahap selanjutnya adalah implementasi Data Manipulation Language (DML). DML digunakan untuk melakukan pengelolaan data pada tabel yang telah dibuat, meliputi proses penambahan data (INSERT), pembaruan data (UPDATE), dan penghapusan data (DELETE). Implementasi DML bertujuan untuk memastikan bahwa database dapat menyimpan, mengubah, dan mengelola data transaksi secara dinamis sesuai dengan kebutuhan sistem penjualan toko CCTV.

a. Implementasi Perintah Insert

Perintah INSERT digunakan untuk mengisikan data awal (seed data) ke dalam tabel-tabel utama seperti kasir, pelanggan, produk, transaksi, detail_transaksi, dan pembayaran. Data yang dimasukkan merupakan data contoh yang merepresentasikan kondisi transaksi nyata pada toko CCTV.

The screenshot shows the MySQL Workbench interface. In the SQL Editor pane, there is a single query:

```

1 • INSERT INTO kasir (nama_kasir, telepon)
2 VALUES
3 ('Ahmad Faizal', '081234567890'),
4 ('Siti Aminah', '081298765432');
5

```

In the Output pane, the results of the execution are displayed:

Action	Time	Message	Duration / Fetch
CREATE VIEW v_japon_transaksi AS SELECT t_id_transaksi, p.nama_pelanggan, t.tanggal_transaksi	10.29.45.02	Error Code: 1080: Table 'v_japon_transaksi' already exists	0.000 sec
EXPLAIN ALTER TABLE pelanggan ADD CONSTRAINT uk_email UNIQUE (email)	11.29.44.09	Error Code: 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'CONSTRAINT uk_email UNIQUE (email)' at line 1	
DESCRIBE kasir	12.29.45.07		
INSERT INTO kasir (nama_kasir, telepon) VALUES ('Ahmad Faizal', '081234567890'), ('Siti Aminah', '081298765432')	13.29.45.31	2 rows(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec / 0.000 sec
EXPLAIN INSERT INTO kasir (nama_kasir, telepon) VALUES ('Ahmad Faizal', '081234567890'), ('Siti Aminah', '081298765432')	14.29.45.37	OK	0.000 sec
INSERT INTO kasir (nama_kasir, telepon) VALUES ('Ahmad Faizal', '081234567890'), ('Siti Aminah', '081298765432')	15.29.45.39	2 rows(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec
SELECT * FROM kasir ORDER BY id LIMIT 0, 1000	16.29.45.47	4 rows(s) returned	0.000 sec / 0.000 sec

Gambar 3. 11 Insert DBMS Tabel Kasir

```

MySQL Workbench - cvt1-training - not supported
File Edit View Query Database Server Tools Scripting Help
Navigation: alter tabel pelanggan tambah e... modify email pelanggan constraint unique pelanggan SQL File 25* pelanggan pelanggan - Table SQL File 14* SQL File 15* kasir SQL File 16* pelanggan + +
MANAGEMENT
    ○ Server Status
    ○ Client Connections
    ○ Users and Privileges
    ○ Status and System Variables
    ○ Data Export
    ○ Data Import/Restore
INSTANCE
    ○ Startup / Shutdown
    ○ Server Logs
    ○ Options File
PERFORMANCE
    ○ Dashboard
    ○ Performance Reports
    ○ Performance Schema Setup
Administration Schemas Information
Schema: kasir_cvt
Output
Action Output
# Time Action Message Duration / Fetch
1 12:09:45:07 DESCRIBE transaksi 5 row(s) returned 0.000 sec / 0.000 sec
2 13:09:45:31 INSERT INTO kasir (nama_pelanggan,telepon, alamat) VALUES ('Ahmad Faizal', '081234567890', 'Jl. Ampera, ... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 0.015 sec
3 14:09:45:37 EXPLAIN INSERT INTO kasir (nama_pelanggan,telepon) VALUES ('Rahmad Faizal', '081234567890') (Site Admin)... OK 0.000 sec
4 15:09:45:39 INSERT INTO kasir (nama_pelanggan,telepon) VALUES ('Rahmad Faizal', '081234567890') (Site Admin)... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 0.000 sec
5 16:09:45:47 SELECT * FROM kasir_cvt LIMIT 0,1000 4 row(s) returned 0.000 sec / 0.000 sec
6 17:09:47:05 SELECT * FROM kasir_cvt LIMIT 0,1000 0 row(s) returned 0.000 sec / 0.000 sec
7 18:09:47:13 INSERT INTO pelanggan (nama_pelanggan,telepon, alamat) VALUES ('Budi Santoso', '085511223344', 'Jl. ... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 0.000 sec
8 19:09:47:57 SELECT * FROM kasir_cvt pelanggan LIMIT 0,1000 2 row(s) returned 0.000 sec / 0.000 sec
9 20:09:59:28 Apply changes to kasir_cvt No changes detected 0.000 sec
10 21:10:07:00 INSERT INTO produk (nama_produk, merek, harga, stok) VALUES ('CCTV Indoor 2MP Dome', 'Hikvision', 3500000.00, 50) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.047 sec

```

Gambar 3. 12 Insert DBMS Pelanggan

```

MySQL Workbench - cvt1-training - unconnected
File Edit View Query Database Server Tools Scripting Help
Navigation: pelanggan constraint unique pelanggan SQL File 25* pelanggan pelanggan - Table SQL File 14* SQL File 15* kasir SQL File 16* pelanggan kasir_cvt - Schema kasir_cvt + +
MANAGEMENT
    ○ Server Status
    ○ Client Connections
    ○ Users and Privileges
    ○ Status and System Variables
    ○ Data Export
    ○ Data Import/Restore
INSTANCE
    ○ Startup / Shutdown
    ○ Server Logs
    ○ Options File
PERFORMANCE
    ○ Dashboard
    ○ Performance Reports
    ○ Performance Schema Setup
Administration Schemas Information
Schema: kasir_cvt
Output
Action Output
# Time Action Message Duration / Fetch
1 17:09:47:05 SELECT * FROM kasir_cvt pelanggan LIMIT 0,1000 0 row(s) returned 0.000 sec / 0.000 sec
2 18:09:47:13 INSERT INTO pelanggan (nama_pelanggan,telepon, alamat) VALUES ('Budi Santoso', '085511223344', 'Jl. ... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 0.000 sec
3 19:09:47:57 SELECT * FROM kasir_cvt pelanggan LIMIT 0,1000 2 row(s) returned 0.000 sec / 0.000 sec
4 20:09:59:28 Apply changes to kasir_cvt No changes detected 0.000 sec
5 21:10:07:00 INSERT INTO produk (nama_produk, merek, harga, stok) VALUES ('CCTV Indoor 2MP Dome', 'Hikvision', 3500000.00, 50) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.047 sec

```

Gambar 3. 13 Insert DBMS Produk

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- SQL Editor:** The query window contains the following SQL code:


```
1 * INSERT INTO transaksi (id_kasir, id_pelanggan, tanggal_transaksi, total_bayar)
2 VALUES (1, 1, '2023-10-25 10:30:00', 2200000.00);
3 |
```
- Output Window:** Shows the results of the executed queries, including the insertion of a transaction record and its details.

Gambar 3. 14 Insert DBMS Transaksi 1

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- SQL Editor:** The query window contains the following SQL code:


```
1 * INSERT INTO detail_transaksi (id_transaksi, id_produk, jumlah, subtotal)
2 VALUES (1, 1, 2, 700000.00),
3 (1, 1, 1, 650000.00),
4 (1, 4, 1, 750000.00);
```
- Output Window:** Shows the results of the executed queries, including the insertion of detail transaction records.

Gambar 3. 15 Insert DBMS Detail Transaksi 1

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- Query Editor:** SQL tab, showing the following SQL code:


```
1 • INSERT INTO pembayaran (id_transaksi, metode_pembayaran, jumlah_uang_diterima, kembalian)
2 VALUES (1, 'Total', 2500000.00, 100000.00)
```
- Output Window:** Action Output tab, showing the results of the insert operation:

Action	Time	Message	Duration / Fetch
20 09:59:20	20 09:59:20	Apply changes to kasir_cctv	0.047 sec
21 10:07:00	21 10:07:00	INSERT INTO produk (nama_produk, merek, harga_stok) VALUES ('CCTV Indoor 2MP Dome', 'Hikvision', 350)	0.016 sec
22 10:09:36	22 10:09:36	INSERT INTO transaksi (id_kasir, id_pelanggan, tanggal_transaksi, total_bayar) VALUES (1, 1, 2023-10-25, 1)	0.016 sec
23 10:10:35	23 10:10:35	INSERT INTO detail_transaksi (id_transaksi, id_produk, jumlah, subtotal) VALUES (1, 1, 2, 700000.00)	0.016 sec
24 10:11:23	24 10:11:23	INSERT INTO pembayaran (id_transaksi, metode_pembayaran, jumlah_uang_diterima, kembalian) VALUES (1, 'Total', 2500000.00, 100000.00)	0.016 sec

Gambar 3. 16 Insert DBMS Pembayaran

b. Implementasi Perintah UPDATE

Perintah UPDATE digunakan untuk melakukan perubahan terhadap data yang telah tersimpan di dalam database.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- Query Editor:** SQL tab, showing the following SQL code:


```
1 • UPDATE kasir
2 SET telepon = '081299998888'
3 WHERE id_kasir = 1;
```
- Output Window:** Action Output tab, showing the results of the update operation:

Action	Time	Message	Duration / Fetch
22 10:09:36	22 10:09:36	INSERT INTO transaksi (id_kasir, id_pelanggan, tanggal_transaksi, total_bayar) VALUES (1, 1, 2023-10-25, 1)	0.016 sec
23 10:10:35	23 10:10:35	INSERT INTO detail_transaksi (id_transaksi, id_produk, jumlah, subtotal) VALUES (1, 1, 2, 700000.00)	0.016 sec
24 10:11:23	24 10:11:23	INSERT INTO pembayaran (id_transaksi, metode_pembayaran, jumlah_uang_diterima, kembalian) VALUES (1, 'Total', 2500000.00, 100000.00)	0.016 sec / 0.000 sec
26 10:17:15	26 10:17:15	UPDATE kasir SET telepon = '081299998888' WHERE id_kasir = 1	0.016 sec

Gambar 3. 17 Query Update Data

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- SQL Editor:** Contains the following SQL code:


```

1 UPDATE pelanggan
2 SET alamat = 'Jl. Sudirman No. 25, Jakarta'
3 WHERE id_pelanggan = 1;
      
```
- Output Window:** Shows the execution log with the following entries:

Time	Action	Message	Duration / Fetch
23.10.10:35	INSERT INTO detail_transaksi (id_transaksi, id_produk, jumlah, subtotal) VALUES (1, 1, 2, 700000)	3 rows affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
24.10.11:23	INSERT INTO pembayaran (id_transaksi, metode_pembayaran, jumlah_uang_ditempa, kembalian) VALUES (1, 'Transfer', 1000000, 0)	1 rows affected	0.016 sec
25.10.12:33	SELECT * FROM pelanggan WHERE id_pelanggan = 1	1 rows returned	0.016 sec / 0.000 sec
26.10.17:15	UPDATE kasi SET telepon = '081299998888' WHERE id_kasi = 1	1 rows affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
27.10.19:25	UPDATE pelanggan SET alamat = 'Jl. Sudirman No. 25, Jakarta' WHERE id_pelanggan = 1	1 rows affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
- Status Bar:** Limit to 1000 rows, Unit to 1000 ms.

Gambar 3. 18 Update Alamat pada Data pelanggan

The screenshot shows the MySQL Workbench interface with the following details:

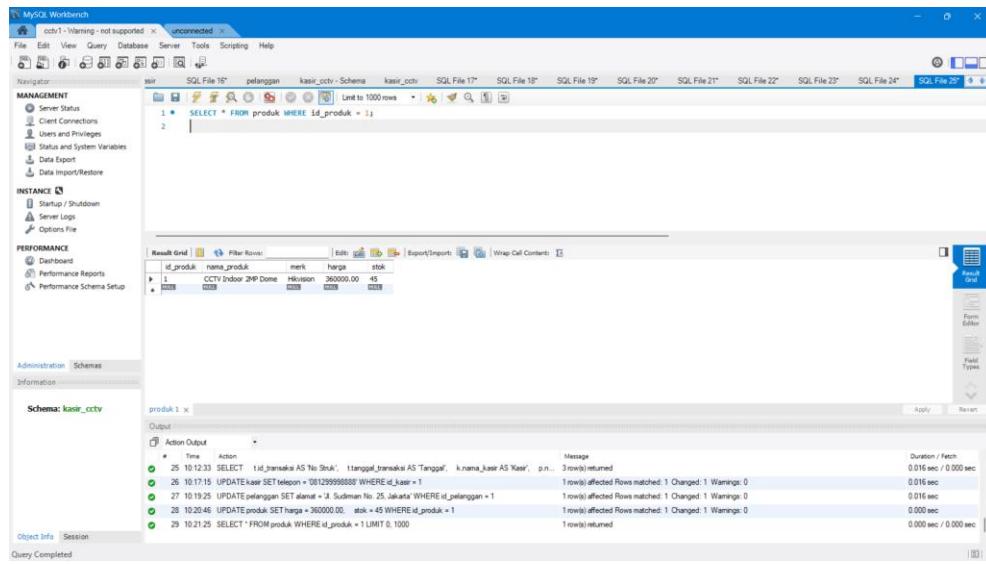
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT, INSTANCE, PERFORMANCE.
- SQL Editor:** Contains the following SQL code:


```

1 UPDATE produk
2 SET harga = 3000000
3 stock = 45
4 WHERE id_produk = 1;
      
```
- Output Window:** Shows the execution log with the following entries:

Time	Action	Message	Duration / Fetch
24.10.11:23	INSERT INTO pembayaran (id_transaksi, metode_pembayaran, jumlah_uang_ditempa, kembalian) VALUES (1, 'Transfer', 1000000, 0)	1 rows affected	0.016 sec
25.10.12:33	SELECT * FROM pelanggan WHERE id_pelanggan = 1	1 rows returned	0.016 sec / 0.000 sec
26.10.17:15	UPDATE kasi SET telepon = '081299998888' WHERE id_kasi = 1	1 rows affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
27.10.19:25	UPDATE pelanggan SET alamat = 'Jl. Sudirman No. 25, Jakarta' WHERE id_pelanggan = 1	1 rows affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
28.10.20:46	UPDATE produk SET harga = 3000000, stock = 45 WHERE id_produk = 1	1 rows affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
- Status Bar:** Limit to 1000 rows, Unit to 1000 ms.

Gambar 3. 19 Update Data Multi Kolom



Gambar 3. 20 Describe Data Yang sudah Diubah Pada Tabel Produk

c. Implementasi Perintah DELETE

Perintah DELETE digunakan untuk menghapus data tertentu dari tabel apabila data tersebut sudah tidak diperlukan atau terjadi kesalahan input. Penghapusan data dilakukan dengan memperhatikan relasi antar tabel agar tidak melanggar constraint Foreign Key yang telah ditetapkan.

Implementasi DELETE pada sistem ini dilakukan secara terbatas untuk menjaga integritas data transaksi.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** A tree view showing various databases like dokumen_qr, fikomdb, flask, gith_220101071, halopet, hngapp, kasir_cctv, live, loketera, mifid, orderpemasok_ranggapalv, otomatis, pesanans_ranggapalv, penjualan_220101071, perceptron, phpmyadmin, gith_220101071, profile_matcher, rangga_220101071, rekomendasi, rong2, si_fkom.
- Current Schema:** kasir_cctv
- Query Editor:**

```

1 • DELETE FROM kasir
  WHERE id_kasir = 3
  
```
- Output Window:**

Action	Time	Action	Message	Duration / Fetch
29	10:21:25	SELECT * FROM produk WHERE id_produk = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
30	10:24:24	SELECT * FROM kasir_cctv.kasir LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
31	10:24:51	Refresh Recordset	There are pending changes. Please commit or rollback first.	
32	10:25:13	Refresh Recordset	There are pending changes. Please commit or rollback first.	
33	10:25:42	DELETE FROM kasir WHERE id_kasir = 3	1 row(s) affected	0.000 sec
- Status:** Object Info, Session, Query Completed.

Gambar 3. 21 Delete Data Kasir ID no 3

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** A tree view showing various databases like dokumen_qr, fikomdb, flask, gith_220101071, halopet, hngapp, kasir_cctv, live, loketera, mifid, orderpemasok_ranggapalv, otomatis, pesanans_ranggapalv, penjualan_220101071, perceptron, phpmyadmin, gith_220101071, profile_matcher, rangga_220101071, rekomendasi, rong2, si_fkom.
- Current Schema:** kasir_cctv
- Query Editor:**

```

1 • DELETE FROM pelanggan
  WHERE id_pelanggan = 3
  
```
- Output Window:**

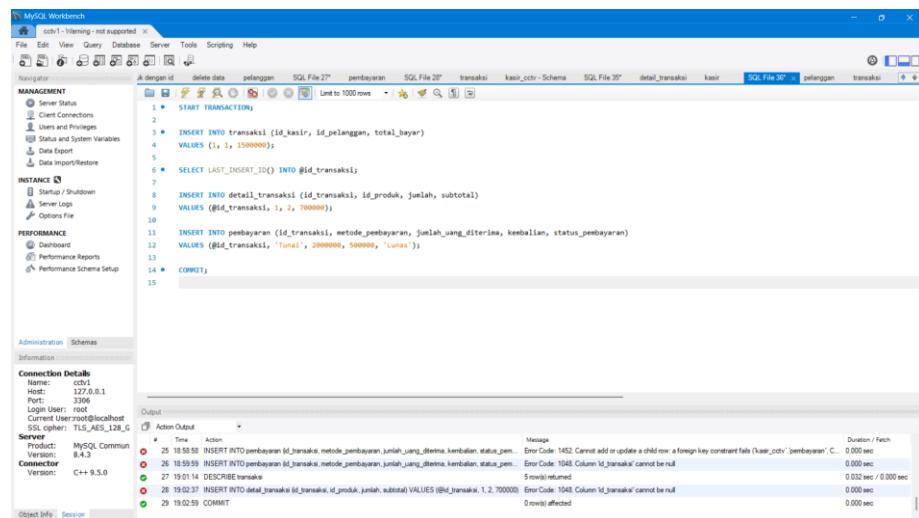
Action	Time	Action	Message	Duration / Fetch
33	10:25:42	DELETE FROM kasir WHERE id_kasir = 3	1 row(s) affected	0.000 sec
34	10:26:45	SELECT * FROM kasir_cctv.pelanggan LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
35	10:27:34	Refresh Recordset	There are pending changes. Please commit or rollback first.	
36	10:27:48	Refresh Recordset	There are pending changes. Please commit or rollback first.	
37	10:28:13	DELETE FROM pelanggan WHERE id_pelanggan = 3	1 row(s) affected	0.000 sec
- Status:** Object Info, Session, Query Completed.

Gambar 3. 22 Delete Data Pelanggan Id No 3

3.3.3 Tabel TCL (Transaction)

Transaction Control Language (TCL) merupakan bagian dari SQL yang digunakan untuk mengatur dan mengendalikan transaksi pada basis data. Penggunaan TCL bertujuan untuk menjaga konsistensi dan integritas data, terutama pada sistem yang melibatkan lebih dari satu tabel yang saling berelasi, seperti sistem penjualan toko CCTV.

Pada sistem yang dibangun, satu proses transaksi penjualan melibatkan beberapa tabel, yaitu tabel transaksi, detail_transaksi, dan pembayaran. Oleh karena itu, seluruh proses tersebut harus dijalankan sebagai satu kesatuan transaksi agar tidak terjadi kondisi data yang tidak lengkap apabila terjadi kesalahan pada salah satu tahap.



The screenshot shows the MySQL Workbench interface with a transaction script in the SQL editor. The script starts with a START TRANSACTION; statement and includes several INSERT statements into tables like transaksi, detail_transaksi, and pembayaran. It also includes a SELECT statement to get the last inserted ID and a COMMIT statement at the end. The connection details in the bottom left show a host of 127.0.0.1 and port 3306.

```
1 • START TRANSACTION;
2
3 • INSERT INTO transaksi (id_kasir, id_pelanggan, total_harga)
4   VALUES (1, 1, 1500000);
5
6 • SELECT LAST_INSERT_ID() INTO @id_transaksi;
7
8   INSERT INTO detail_transaksi (@id_transaksi, id_produk, jumlah, subtotal)
9   VALUES (@id_transaksi, 1, 700000);
10
11  INSERT INTO pembayaran (@id_transaksi, metode_pembayaran, jumlah_uang_diterima, kembalian, status_pembayaran)
12  VALUES (@id_transaksi, 'Transfer', 2000000, 500000, 'Lunas');
13
14 • COMMIT;
15
```

Gambar 3. 23 Start Transaksi

The screenshot shows the MySQL Workbench interface with the following details:

- Connection:** cbv1 - Yeneng - not supported
- Query:** `SELECT * FROM transaksi ORDER BY id_transaksi DESC;`
- Result Grid:**

id_transaksi	id_kasir	id_pelanggan	tanggal_transaksi	total_bayar
9	1	1	2026-01-15 18:58:00	1500000.00
8	1	4	2026-01-15 18:58:00	1500000.00
7	1	6	2026-01-16 11:07:27	3300000.00
5	1	5	2026-01-16 11:04:52	1210000.00
4	2	2	2026-01-16 10:57:46	1080000.00
3	1	4	2026-01-16 10:57:46	1080000.00
2	1	1	2026-01-16 10:33:31	1500000.00
1	1	1	2023-10-25 10:30:00	2200000.00
- Action Output:**

```

27 19:01:14 DESCRIBE transaksi
28 19:02:37 INSERT INTO detail_transaksi(id_transaksi, id_produk, jumlah, subtotal)
VALUES (@id_transaksi, 1, 2, 700000)
29 19:02:59 COMMIT
30 19:03:39 SELECT * FROM transaksi ORDER BY id_transaksi DESC LIMIT 0, 1000
31 19:04:16 SELECT * FROM detail_transaksi ORDER BY id_detail DESC LIMIT 0, 1000

```

Gambar 3. 24 `SELECT * FROM transaksi ORDER BY id_transaksi DESC`

The screenshot shows the MySQL Workbench interface with the following details:

- Connection:** cbv1 - Yeneng - not supported
- Query:** `SELECT * FROM detail_transaksi ORDER BY id_detail DESC;`
- Result Grid:**

id_detail	id_transaksi	id_produk	jumlah	subtotal
12	9	2	2	700000.00
11	2	1	2	700000.00
10	6	5	3	1350000.00
9	6	2	3	1350000.00
8	5	1	1	850000.00
7	5	1	1	360000.00
6	4	1	3	1080000.00
5	3	2	3	1080000.00
4	2	1	2	700000.00
3	1	4	1	750000.00
2	1	3	1	850000.00
1	1	1	2	700000.00
- Action Output:**

```

27 19:01:14 DESCRIBE transaksi
28 19:02:37 INSERT INTO detail_transaksi(id_transaksi, id_produk, jumlah, subtotal)
VALUES (@id_transaksi, 1, 2, 700000)
29 19:02:59 COMMIT
30 19:03:39 SELECT * FROM transaksi ORDER BY id_transaksi DESC LIMIT 0, 1000
31 19:04:16 SELECT * FROM detail_transaksi ORDER BY id_detail DESC LIMIT 0, 1000

```

Gambar 3. 25 `SELECT * FROM detail_transaksi ORDER BY id_detail DESC`

The screenshot shows the MySQL Workbench interface. In the top-left corner, there's a warning message: "cctv1 - Warning - not supported". The main area displays a query result grid and an action history.

Query Editor:

```
SELECT * FROM pembayaran ORDER BY id_pembayaran DESC;
```

Result Grid:

id_pembayaran	id_transaksi	metode_pembayaran	jumlah_uang_diterima	kerabalan	status_pembayaran
7	2	Tunai	2000000.00	500000.00	masuk
6	6	Tunai	3300000.00	0.00	masuk
5	5	Tunai	1200000.00	0.00	masuk
4	4	Tunai	1800000.00	0.00	masuk
3	3	Tunai	1950000.00	0.00	masuk
2	2	Tunai	2000000.00	500000.00	masuk
1	1	Tunai	2500000.00	300000.00	masuk

Action History:

#	Time	Action	Message	Duration / Fetch
28	19:02:37	Action	Error Code: 1043. Column 'id_transaksi' cannot be null	0.000 sec
29	19:02:59	COMMIT	0 row(s) affected	0.000 sec
30	19:03:39	SELECT * FROM transaksi ORDER BY id_transaksi DESC LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
31	19:04:16	SELECT * FROM detail_transaksi ORDER BY id_detail DESC LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
32	19:04:38	SELECT * FROM pembayaran ORDER BY id_pembayaran DESC LIMIT 0, 1000	7 row(s) returned	0.015 sec / 0.000 sec

Gambar 3. 26 `SELECT * FROM pembayaran ORDER BY id_pembayaran DESC`

3.3.4 Implementasi (JOIN, GROUP BY, HAVING, Agregasi)

Setelah proses implementasi database dan pengisian data selesai, tahap selanjutnya adalah penerapan query SQL untuk menampilkan informasi yang dibutuhkan oleh sistem. Query yang digunakan pada sistem penjualan toko CCTV ini meliputi JOIN, GROUP BY, HAVING, dan fungsi agregasi, sesuai dengan materi yang dipelajari pada mata kuliah Pemrograman Basis Data.

a. Penerapan JOIN

Query JOIN digunakan untuk menggabungkan data dari beberapa tabel yang saling berelasikan berdasarkan Primary Key (PK) dan Foreign Key (FK).

The screenshot shows the MySQL Workbench interface with a SQL editor window containing the following query:

```

SELECT
    t.id_transaksi,
    t.tanggal_transaksi,
    k.nama_kasir,
    p.nama_pelanggan,
    t.total_bayar
FROM transaksi t
JOIN kasir k ON t.id_kasir = k.id_kasir
JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan;

```

The results grid displays the following data:

	t.id_transaksi	t.tanggal_transaksi	k.nama_kasir	p.nama_pelanggan	t.total_bayar
1	2	2023-01-16 10:33:31	Ahmad Faiz	Budi Santoso	2200000.00
2	2	2023-01-16 10:33:30	Ahmad Faiz	Budi Santoso	2200000.00
3	4	2023-01-16 18:58:09	Ahmad Faiz	Budi Santoso	1800000.00
4	4	2023-01-16 18:58:08	Ahmad Faiz	Catur Pramugika	1800000.00
5	3	2023-01-16 20:55:14	Ahmad Faiz	Catur Pramugika	1800000.00
6	7	2023-01-16 18:49:45	Ahmad Faiz	Catur Pramugika	1800000.00
7	7	2023-01-16 11:04:52	Ahmad Faiz	hasan m	1200000.00
8	5	2023-01-16 11:07:27	Ahmad Faiz	hasan m	2300000.00

Gambar 3. 27 JOIN

Query ini menampilkan informasi transaksi lengkap yang mencakup tanggal transaksi, nama kasir, nama pelanggan, serta total pembayaran.

b. Penerapan JOIN dengan detail_transaksi

The screenshot shows the MySQL Workbench interface with a SQL editor window containing the following query:

```

SELECT
    t.id_transaksi,
    pr.nama_produk,
    dt.jumlah,
    dt.subtotal
FROM detail_transaksi dt
JOIN transaksi t ON dt.id_transaksi = t.id_transaksi
JOIN produk pr ON dt.id_produk = pr.id_produk;

```

The results grid displays the following data:

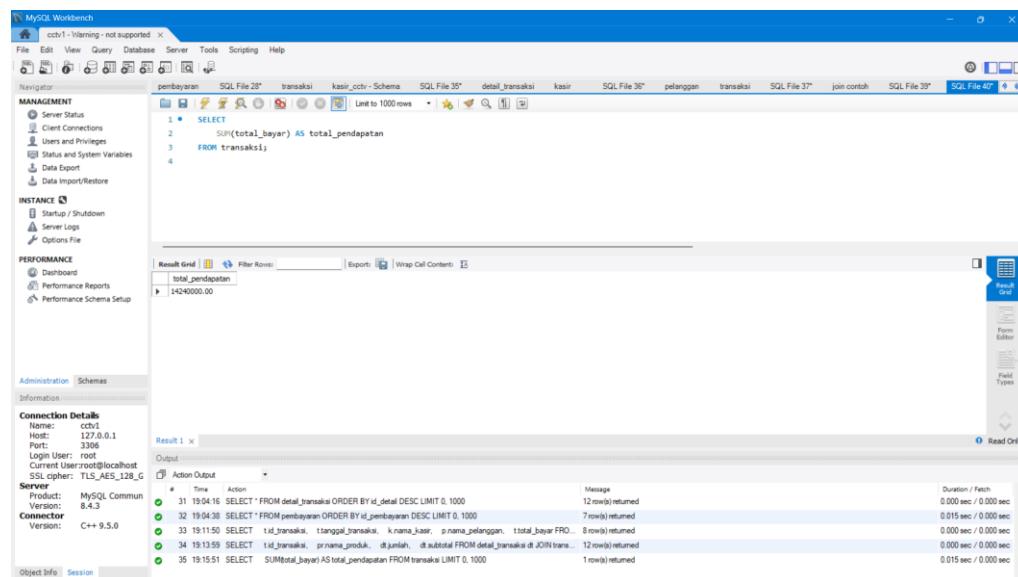
	t.id_transaksi	pr.nama_produk	dt.jumlah	dt.subtotal
1	1	CC TV Indoor 2MP Dome	4	8500000.00
2	1	DVR 4 Channel 8CHP	1	850000.00
3	1	Harddisk 1TB WD Purple	1	750000.00
4	2	CCTV Indoor 2MP Dome	2	7000000.00
5	3	CC TV Indoor 2MP Bullet	3	19500000.00
6	4	CCTV Indoor 2MP Dome	3	18000000.00
7	5	CCTV Indoor 2MP Dome	1	3600000.00
8	5	DVR 4 Channel 8CHP	1	850000.00
9	6	CC TV Indoor 2MP Bullet	3	19500000.00
10	6	Kabel Coaxial RG59 + Power (20m)	3	13500000.00
11	2	CCTV Indoor 2MP Dome	2	7000000.00
12	8	CCTV Indoor 2MP Dome	2	7000000.00

Gambar 3. 28 JOIN dengan detail_transaksi

Menampilkan daftar produk yang dibeli dalam setiap transaksi beserta jumlah dan subtotalnya.

c. Penerapan Agregasi Data (Sum, Count)

Query agregasi digunakan untuk melakukan perhitungan data numerik, seperti total penjualan dan jumlah transaksi.



The screenshot shows the MySQL Workbench interface. In the central SQL editor pane, the following query is displayed:

```
1 SELECT
2     SUM(total_bayar) AS total_pendapatan
3
4 FROM transaksi;
```

The result grid shows one row with the value 14240000.00. Below the result grid, the execution log shows the following entries:

Action	Time	Action	Message	Duration / Fetch
31	19:04:16	SELECT * FROM detail_transaksi ORDER BY id_detail DESC LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
32	19:04:38	SELECT * FROM pembayaran ORDER BY id_pembayaran DESC LIMIT 0, 1000	7 row(s) returned	0.015 sec / 0.000 sec
33	19:11:50	SELECT t.id_transaksi, tanggal_transaksi, k.nama_kasir, p.nama_pelanggan, total_bayar 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 pelanggan p ON t.id_pelanggan = p.id_pelanggan	8 row(s) returned	0.000 sec / 0.000 sec
34	19:13:59	SELECT t.id_transaksi, p.nama_produk, dt.jumlah, dt.subtotal FROM detail_transaksi dt JOIN transaksi t ON dt.id_transaksi = t.id_transaksi JOIN produk p ON dt.id_produk = p.id_produk	12 row(s) returned	0.000 sec / 0.000 sec
35	19:15:51	SELECT SUM(total_bayar) AS total_pendapatan FROM transaksi LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

Gambar 3. 29 Agregasi Data Total pendapatan

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```

1 • SELECT
2   COUNT(id_transaksi) AS jumlah_transaksi
3   FROM transaksi;

```

The result grid shows one row with the value 8.

Below the query editor, the Connection Details section shows the connection is to 'ccv1' with port 3306 and user 'root'. The Server section shows the product is MySQL Community Connector Version 8.4.3, and the Connector section shows it is C++ 9.5.0.

Gambar 3. 30 Agregasi Data Jumlah Transaksi

d. Penerapan GROUP BY

Query GROUP BY digunakan untuk mengelompokkan data berdasarkan kriteria tertentu.

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```

1 • SELECT
2   p.nama_pelanggan,
3   SUM(t.total_bayar) AS total_belanja
4   FROM transaksi t
5   JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan
6   GROUP BY p.nama_pelanggan;

```

The result grid shows the following data:

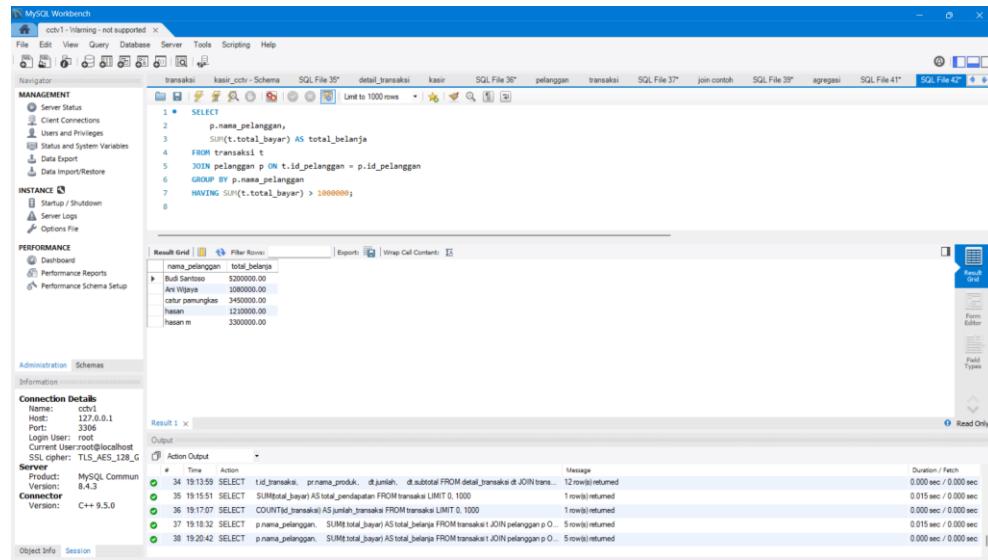
nama_pelanggan	total_belanja
Budi Santoso	5200000.00
Ari Wijaya	1080000.00
catur pamungkas	100000.00
Hasan	1210000.00
Hasan	3300000.00

Below the query editor, the Connection Details section shows the connection is to 'ccv1' with port 3306 and user 'root'. The Server section shows the product is MySQL Community Connector Version 8.4.3, and the Connector section shows it is C++ 9.5.0.

Gambar 3. 31 Group By Total Pelanggan

e. Penerapan HAVING

Having digunakan untuk menyaring hasil dari Group By berdasarkan kondisi tertentu. Contoh pelanggan dengan total belanja Lebih Dari 1.000.000.



The screenshot shows the MySQL Workbench interface with a query editor and results pane. The query is:

```
SELECT p.nama_pelanggan,
       SUM(t.total_bayar) AS total_belanja
  FROM transaksi t
 JOIN pelanggan p
    ON t.id_pelanggan = p.id_pelanggan
 GROUP BY p.nama_pelanggan
 HAVING SUM(t.total_bayar) > 1000000;
```

The results grid shows:

nama_pelanggan	total_belanja
Budi Santoso	3200000.00
Ari Wilaya	1080000.00
catur pemungkas	3450000.00
hassan	1230000.00
hassan m	3300000.00

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
34	19:13:59	SELECT t.id_transaksi, pr.nama_produk, dj.jumlah, dt.autototal FROM detail_transaksi dt JOIN transaksi t ON t.id_transaksi = dt.id_transaksi JOIN produk pr ON pr.id_produk = dt.id_produk JOIN detail_transaksi dj ON dj.id_transaksi = t.id_transaksi JOIN produk dt ON dt.id_produk = dj.id_produk WHERE t.tanggal_transaksi = '2023-09-13' AND pr.id_kategori = 1 AND dt.autototal > 1000000	12 row(s) returned	0.000 sec / 0.000 sec
35	19:15:51	SELECT SUM(total_bayar) AS total_pendapatan FROM transaksi LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
36	19:17:07	SELECT COUNT(id_transaksi) AS jumlah_transaksi FROM transaksi LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
37	19:18:32	SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.tanggal_transaksi = '2023-09-13' AND p.id_kategori = 1	5 row(s) returned	0.015 sec / 0.000 sec
38	19:20:42	SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.tanggal_transaksi = '2023-09-13' AND p.id_kategori = 1	5 row(s) returned	0.000 sec / 0.000 sec

Gambar 3. 32 having contoh

BAB IV

PENUTUP

4.1 Ringkasan Implementasi Basis Data

Implementasi basis data pada sistem penjualan toko CCTV telah dilakukan berdasarkan perancangan Entity Relationship Diagram (ERD) yang disusun pada bab sebelumnya. Penerapan ERD ke dalam Database Management System (DBMS) MySQL dilakukan menggunakan MySQL Workbench melalui proses forward engineering.

Struktur database yang dihasilkan terdiri dari beberapa tabel utama, yaitu tabel kasir, pelanggan, produk, transaksi, detail_transaksi, dan pembayaran. Setiap tabel telah didefinisikan dengan atribut yang sesuai, serta dilengkapi dengan Primary Key (PK) dan Foreign Key (FK) untuk menjaga integritas data dan hubungan antar tabel. Dengan diterapkannya constraint tersebut, database mampu merepresentasikan hubungan satu-ke-banyak dan satu-ke-satu sesuai dengan kebutuhan sistem penjualan.

4.2 Ringkasan Manipulasi Data (DML)

Setelah struktur database berhasil dibuat, tahap selanjutnya adalah manipulasi data menggunakan Data Manipulation Language (DML). Proses ini meliputi pengisian data awal (seed data) ke dalam tabel master seperti kasir, pelanggan, dan produk menggunakan perintah INSERT.

Selain itu, pengujian perintah UPDATE dilakukan untuk memastikan bahwa data dapat diperbarui sesuai kebutuhan, misalnya perubahan stok produk atau status pembayaran. Pengujian DELETE juga dilakukan untuk

memastikan data dapat dihapus dengan tetap memperhatikan keterkaitan antar tabel agar tidak melanggar constraint Foreign Key. Hasil pengujian menunjukkan bahwa proses manipulasi data dapat berjalan dengan baik dan konsisten sesuai dengan rancangan basis data.

4.3 Ringkasan Pengujian Query dan Analisis Data

Pengujian query lanjutan dilakukan untuk membuktikan bahwa basis data mampu menghasilkan informasi yang dibutuhkan oleh sistem penjualan. Query JOIN digunakan untuk menampilkan data transaksi secara lengkap dengan menggabungkan informasi dari tabel transaksi, kasir, pelanggan, detail_transaksi, produk, dan pembayaran.

Selain itu, fungsi agregasi seperti SUM dan COUNT digunakan untuk menghitung total penjualan dan jumlah transaksi. Penggunaan GROUP BY dan HAVING memungkinkan pengelompokan data berdasarkan kriteria tertentu, seperti rekap penjualan per produk atau per merek. Hasil query menunjukkan bahwa database mampu mendukung kebutuhan pelaporan dan analisis data secara efektif.

4.4 Pengujian Transaksi (Transaction Control Language)

Pengujian Transaction Control Language (TCL) dilakukan untuk memastikan konsistensi data dalam proses transaksi. Perintah START TRANSACTION, COMMIT, dan ROLLBACK digunakan untuk mensimulasikan skenario transaksi pembayaran.

Melalui pengujian ini, dapat dibuktikan bahwa apabila terjadi kesalahan pada salah satu proses transaksi, perintah ROLLBACK mampu mengembalikan kondisi database ke keadaan semula sehingga tidak terjadi inkonsistensi data. Sebaliknya, apabila seluruh proses berjalan dengan benar,

perintah COMMIT memastikan bahwa data tersimpan secara permanen di dalam database.

4.5 Kendala dan Upaya Perbaikan

Selama proses perancangan dan implementasi basis data, terdapat beberapa kendala yang dihadapi, antara lain kesalahan urutan pengisian data yang menyebabkan pelanggaran constraint Foreign Key, serta error pada kolom yang tidak boleh bernilai NULL.

Upaya perbaikan dilakukan dengan menyesuaikan urutan eksekusi perintah SQL, memastikan data pada tabel induk tersedia sebelum data pada tabel relasi dimasukkan, serta melakukan pengecekan struktur tabel menggunakan perintah DESCRIBE. Dengan perbaikan tersebut, sistem basis data dapat berjalan sesuai dengan perancangan awal.

4.6 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem basis data penjualan toko CCTV berhasil dibangun dan diimplementasikan dengan baik menggunakan MySQL. Struktur database yang dibuat telah sesuai dengan ERD dan memenuhi prinsip normalisasi hingga bentuk normal ketiga (3NF). Selain itu, penggunaan DDL, DML, TCL, serta query lanjutan seperti JOIN, GROUP BY, HAVING, dan fungsi agregasi menunjukkan bahwa basis data mampu mendukung kebutuhan pengolahan data transaksi secara terstruktur, konsisten, dan efisien.

Untuk pengembangan selanjutnya, sistem basis data ini dapat dikembangkan dengan menambahkan fitur seperti trigger untuk otomatisasi pengelolaan stok, stored procedure untuk meningkatkan efisiensi query, serta

integrasi dengan aplikasi berbasis web atau mobile. Selain itu, pengamanan database dan optimasi performa melalui indexing juga dapat menjadi fokus pengembangan agar sistem lebih siap digunakan dalam skala yang lebih besar.

LAMPIRAN

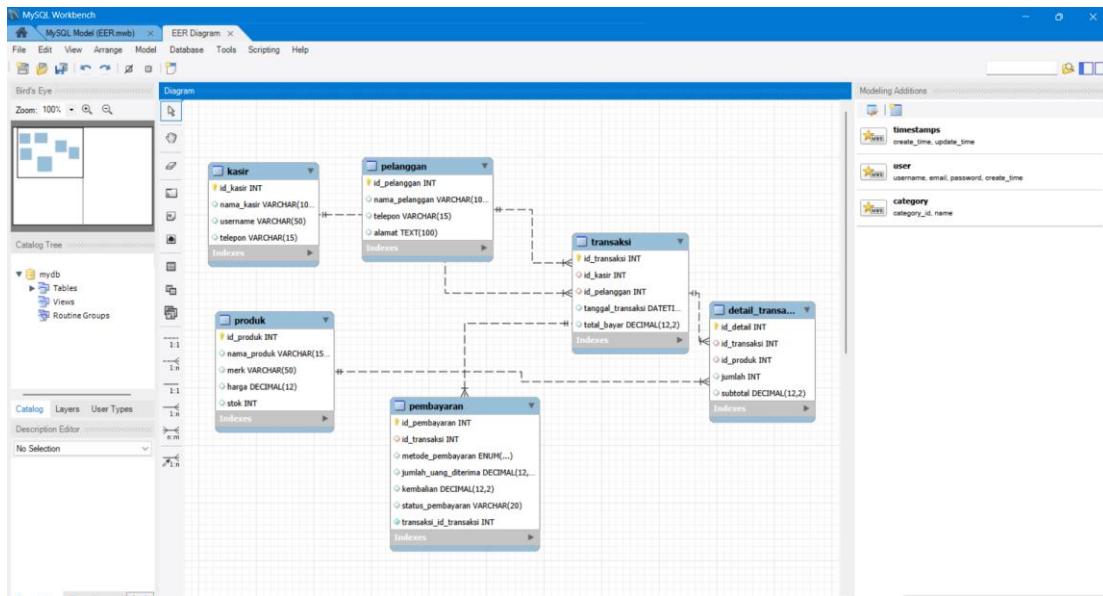
Lampiran 1 : Link Repository Github

<https://github.com/pemungkascat/PROJECT-UAS-KASIR-CCTV-WEB.git>

Lampiran 2 : Struktur Folder Repository

```
/sql
├── schema.sql
└── seed.sql
/query
├── join.sql
├── groupby.sql
└── transaksi.sql
/EER
└── erd_final.png
/kasir_cctv
└── source_code.php
/laporan
└── laporan.pdf
```

Lampiran 3 : ERD FinaL

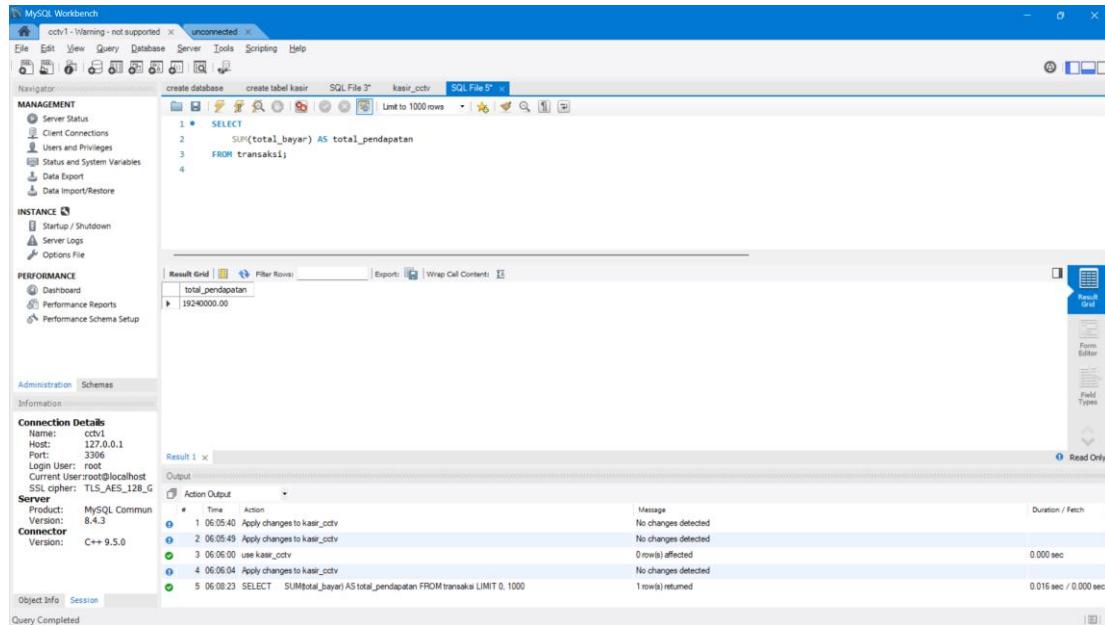


Lampiran 4 : Struktur Tabel

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Incre...	Create Time	Update Time
detail_transaksi	InnoDB	10	Dynamic	16	1024	16.0 KB	0.0 bytes	32.0 KB	0.0 bytes	17	2020-01-16 01:54:05	
kasir	InnoDB	10	Dynamic	4	4096	16.0 KB	0.0 bytes	16.0 KB	0.0 bytes	6	2020-01-16 02:00:36	
pelanggan	InnoDB	10	Dynamic	7	2340	16.0 KB	0.0 bytes	16.0 KB	0.0 bytes	9	2020-01-16 02:00:15	
pembayaran	InnoDB	10	Dynamic	8	2048	16.0 KB	0.0 bytes	16.0 KB	0.0 bytes	11	2020-01-16 01:54:51	
produk	InnoDB	10	Dynamic	6	2730	16.0 KB	0.0 bytes	16.0 KB	0.0 bytes	7	2020-01-16 01:57:22	
transaksi	InnoDB	10	Dynamic	11	1489	16.0 KB	0.0 bytes	32.0 KB	0.0 bytes	12	2020-01-16 01:57:42	

Lampiran 5 : Query

Agregasi Data



MySQL Workbench - cctv1 - Warning - not supported - unconnected

```

SELECT
    SUM(total_bayar) AS total_pendapatan
FROM transaksi;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Field Types Read Only

total_pendapatan
1920000.00

Connection Details

- Name: cctv1
- Host: 127.0.0.1
- Port: 3306
- Login User: root
- Current Userroot@localhost
- SSL cipher: TLS_AES_128_G

Server

- Product: MySQL Commun
- Version: 8.4.3

Connector

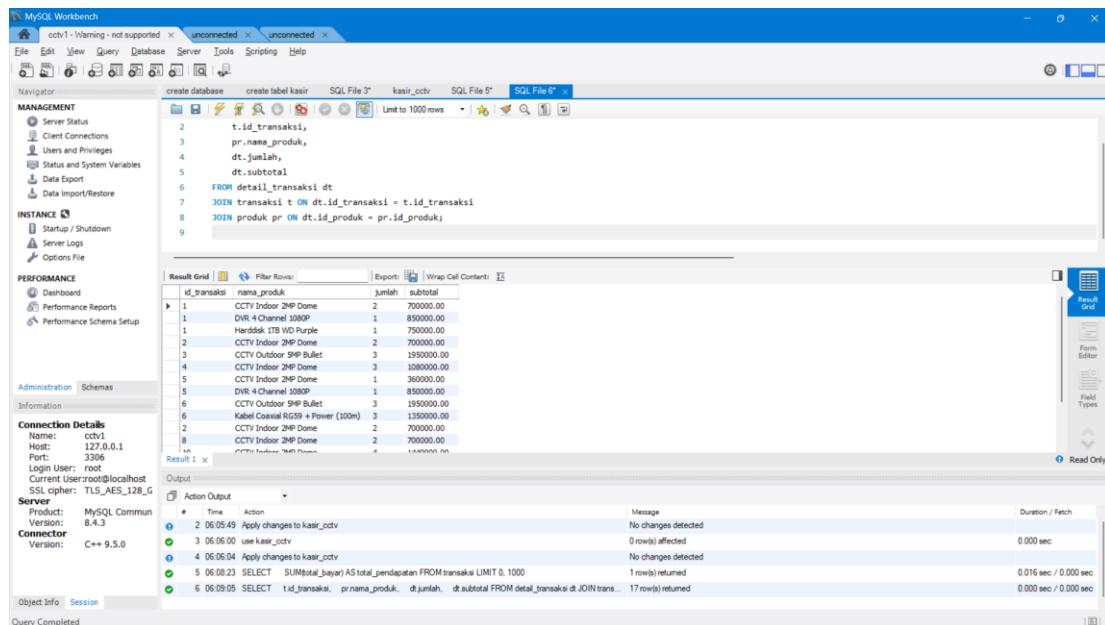
- Version: C++ 9.5.0

Action Output

#	Time	Action	Message	Duration / Fetch
1	06:05:40	Apply changes to kasir_cctv	No changes detected	
2	06:05:49	Apply changes to kasir_cctv	No changes detected	
3	06:06:00	use kasir_cctv	0 row(s) affected	0.000 sec
4	06:06:04	Apply changes to kasir_cctv	No changes detected	
5	06:08:23	SELECT SUM(total_bayar) AS total_pendapatan FROM transaksi LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec

Object Info Session Query Completed

join



MySQL Workbench - cctv1 - Warning - not supported - unconnected

```

SELECT
    t.id_transaksi,
    pr.nama_produk,
    dt.jumlah,
    dt.subtotal
FROM detail_transaksi dt
JOIN transaksi t ON dt.id_transaksi = t.id_transaksi
JOIN produk pr ON dt.id_produk = pr.id_produk;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Field Types Read Only

d_transaksi	nama_produk	Jumlah	subtotal
CCTV Indoor 2MP Dome	2	700000.00	
DVR 4 Channel 1080P	1	850000.00	
Harddisk 1TB WD Purple	1	750000.00	
CCTV Indoor 2MP Dome	2	700000.00	
CCTV Outdoor 5MP Bullet	3	1950000.00	
CCTV Indoor 2MP Dome	3	1080000.00	
CCTV Indoor 2MP Dome	1	360000.00	
DVR 8 Channel 5MP	1	1200000.00	
CCTV Outdoor 5MP Bullet	3	1950000.00	
Kabel Coaxial RG59 + Power (100m)	3	1350000.00	
CCTV Indoor 2MP Dome	2	700000.00	
CCTV Indoor 2MP Dome	2	700000.00	
CCTV Indoor 2MP Dome	1	1400000.00	

Connection Details

- Name: cctv1
- Host: 127.0.0.1
- Port: 3306
- Login User: root
- Current Userroot@localhost
- SSL cipher: TLS_AES_128_G

Server

- Product: MySQL Commun
- Version: 8.4.3

Connector

- Version: C++ 9.5.0

Action Output

#	Time	Action	Message	Duration / Fetch
1	06:05:49	Apply changes to kasir_cctv	No changes detected	
2	06:06:00	use kasir_cctv	0 row(s) affected	0.000 sec
3	06:06:04	Apply changes to kasir_cctv	No changes detected	
4	06:08:23	SELECT SUM(total_bayar) AS total_pendapatan FROM transaksi LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
5	06:09:05	SELECT t.id_transaksi, pr.nama_produk, dt.jumlah, dt.subtotal FROM detail_transaksi dt JOIN transaksi t ON dt.id_transaksi = t.id_transaksi, JOIN produk pr ON dt.id_produk = pr.id_produk	17 row(s) returned	0.000 sec / 0.000 sec

Object Info Session Query Completed

MySQL Workbench

```

    cctv1 - Warning - not supported x unconnected x unconnected x unconnected x
    File Edit View Query Database Server Tools Scripting Help
    Navigator create database create tabel kasir SQL File 3* kasir_cctv SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10*
    MANAGEMENT
        Server Status
        Client Connections
        Users and Privileges
        Status and System Variables
        Data Export
        Data Import/Restore
    INSTANCE
        Startup / Shutdown
        Server Logs
        Options File
    PERFORMANCE
        Dashboard
        Performance Reports
        Performance Schema Setup
    Administration Schemas
    Information
    Connection Details
        Name: cctv1
        Host: 127.0.0.1
        Port: 3306
        Login User: root
        Current User:root@localhost
        SSL cipher: TLS_AES_128_G
    Server
        Product: MySQL Community
        Version: 8.4.3
    Connector
        Version: C++ 9.5.0
    Object Info Session
    Query Completed

```

create tabel kasir

```

3 t.tanggal_transaksi,
4 k.nama_kasir,
5 p.nama_pelanggan,
6 t.total_bayar
7 FROM transaksi t
8 JOIN kasir k ON t.id_kasir = k.id_kasir
9 JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan;
10

```

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

	id_transaksi	tanggal_transaksi	nama_kasir	nama_pelanggan	total_bayar
▶	6	2026-01-16 11:07:27	Ahmed Faizi	heman m	3300000.00
5	2026-01-16 11:04:52	Ahmed Faizi	heman	12100000.00	
7	2026-01-16 10:49:45	Ahmed Faizi	catur pamungkas	1500000.00	
3	2026-01-16 10:49:45	Ahmed Faizi	catur pamungkas	1500000.00	
8	2026-01-16 10:55:09	Ahmed Faizi	Budi Santoso	1500000.00	
2	2026-01-16 10:55:31	Ahmed Faizi	Budi Santoso	1500000.00	
1	2023-10-25 20:30:30	Ahmed Faizi	Budi Santoso	2200000.00	
11	2026-01-16 20:52:39	Siti Aminah	gugupiygy	12100000.00	
10	2026-01-16 20:51:49	Siti Aminah	Ahmad Hasan Mustafa	3790000.00	
4	2026-01-16 20:57:46	Siti Aminah	Ani Wijaya	1080000.00	
9	2026-01-16 20:51:15	Ahmed Faizi	Ahmad Hasan Mustafa	0.00	

Action Output | Duration / Fetch: 0.031 sec / 0.000 sec

```

# Time Action Message
5 06:08:23 SELECT SUM(total_bayar) AS total_pendapatan FROM transaksi LIMIT 0, 1000 1 rows(s) returned
6 06:09:23 SELECT t.id_transaksi, pr.nama_produk, dt.jumlah, dt.subtotal FROM detail_transaksi dt JOIN transaksi t ON t.id_transaksi = dt.id_transaksi JOIN kasir k ON t.id_kasir = k.id_kasir GROUP BY k.nama_kasir;
7 06:09:37 SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.id_kasir = 1
8 06:10:21 SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.id_kasir = 1
9 06:12:35 SELECT t.id_transaksi, t.tanggal_transaksi, k.nama_kasir, p.nama_pelanggan, t.total_bayar FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.id_kasir = 1
10 06:13:04 SELECT k.nama_kasir, SUM(total_bayar) AS total_perjualan FROM transaksi t JOIN kasir k ON id_kasir WHERE t.id_kasir = 1

```

Group By + SUM

MySQL Workbench

```

    cctv1 - Warning - not supported x unconnected x unconnected x unconnected x
    File Edit View Query Database Server Tools Scripting Help
    Navigator create database create tabel kasir SQL File 3* kasir_cctv SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10*
    MANAGEMENT
        Server Status
        Client Connections
        Users and Privileges
        Status and System Variables
        Data Export
        Data Import/Restore
    INSTANCE
        Startup / Shutdown
        Server Logs
        Options File
    PERFORMANCE
        Dashboard
        Performance Reports
        Performance Schema Setup
    Administration Schemas
    Information
    Connection Details
        Name: cctv1
        Host: 127.0.0.1
        Port: 3306
        Login User: root
        Current User:root@localhost
        SSL cipher: TLS_AES_128_G
    Server
        Product: MySQL Community
        Version: 8.4.3
    Connector
        Version: C++ 9.5.0
    Object Info Session
    Query Completed

```

create tabel kasir

```

1 * SELECT
2     k.nama_kasir,
3     SUM(t.total_bayar) AS total_pendapatan
4     FROM transaksi t
5     JOIN kasir k ON t.id_kasir = k.id_kasir
6     GROUP BY k.nama_kasir;
7

```

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

	nama_kasir	total_pendapatan
▶	Ahmed Faizi	13160000.00
	Siti Aminah	6880000.00

Action Output | Duration / Fetch: 0.031 sec / 0.000 sec

```

# Time Action Message
6 06:09:05 SELECT t.id_transaksi, pr.nama_produk, dt.jumlah, dt.subtotal FROM detail_transaksi dt JOIN transaksi t ON t.id_transaksi = dt.id_transaksi JOIN kasir k ON t.id_kasir = k.id_kasir GROUP BY k.nama_kasir;
7 06:09:37 SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.id_kasir = 1
8 06:10:21 SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.id_kasir = 1
9 06:12:35 SELECT t.id_transaksi, t.tanggal_transaksi, k.nama_kasir, p.nama_pelanggan, t.total_bayar FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan WHERE t.id_kasir = 1
10 06:13:04 SELECT k.nama_kasir, SUM(total_bayar) AS total_perjualan FROM transaksi t JOIN kasir k ON id_kasir WHERE t.id_kasir = 1

```

Having

MySQL Workbench

ccv1 - Warning - not supported unconnected unconnected unconnected unconnected

File Edit View Query Database Server Tools Scripting Help

Navigator create database create tabel kasir SQL File 3* kasir_ccv SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12*

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

Connection Details

Name: ccv1
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_128_G

Server

Product: MySQL Community
Version: 8.4.3

Connector

Version: C++ 9.5.0

Object Info Session

Query Completed

```

1 *   SELECT
2       k.nama_kasir,
3       SUM(t.total_bayar) AS total_penjualan
4   FROM transaksi t
5   JOIN kasir k ON t.id_kasir = k.id_kasir
6   GROUP BY k.nama_kasir
7   HAVING SUM(t.total_bayar) > 1000000;
8

```

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

nama_kasir	total_penjualan
Ahmad Faiz	13160000.00
Siti Aminah	6080000.00

Result 1 x

Action Output

#	Time	Action	Message	Duration / Fetch
1	7 06:09:37	SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan	7 rows(s) returned	0.016 sec / 0.000 sec
2	8 06:10:21	SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan	7 rows(s) returned	0.000 sec / 0.000 sec
3	9 06:12:35	SELECT t.id_transaksi, t.tanggal_transaksi, k.nama_kasir, p.nama_pelanggan, t.total_bayar FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan	11 rows(s) returned	0.031 sec / 0.000 sec
4	10 06:13:04	SELECT k.nama_kasir, SUM(total_bayar) AS total_penjualan FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir	2 rows(s) returned	0.000 sec / 0.000 sec
5	11 06:13:31	SELECT k.nama_kasir, SUM(total_bayar) AS total_penjualan FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir	2 rows(s) returned	0.000 sec / 0.016 sec

Read Only

TCL Transaction

MySQL Workbench

ccv1 - Warning - not supported unconnected unconnected unconnected unconnected

File Edit View Query Database Server Tools Scripting Help

Navigator create database create tabel kasir SQL File 3* kasir_ccv SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12*

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

Connection Details

Name: ccv1
Host: 127.0.0.1
Port: 3306
Login User: root
Current User: root@localhost
SSL cipher: TLS_AES_128_G

Server

Product: MySQL Community
Version: 8.4.3

Connector

Version: C++ 9.5.0

Object Info Session

Query Completed

```

1 *   START TRANSACTION;
2
3 *   INSERT INTO transaksi (id_kasir, id_pelanggan, total_bayar)
4     VALUES (1, 1, 2200000);
5
6 *   INSERT INTO pembayaran (id_transaksi, metode_pembayaran, jumlah_uang_diterima, kembalian, status_pembayaran)
7     VALUES (LAST_INSERT_ID(), 'Tunai', 2500000, 300000, 'Lunas');
8
9 *   COMMIT;
10

```

Action Output

#	Time	Action	Message	Duration / Fetch
1	8 06:10:21	SELECT p.nama_pelanggan, SUM(total_bayar) AS total_belanja FROM transaksi t JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan	7 rows(s) returned	0.000 sec / 0.000 sec
2	9 06:12:35	SELECT t.id_transaksi, t.tanggal_transaksi, k.nama_kasir, p.nama_pelanggan, t.total_bayar FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan	11 rows(s) returned	0.031 sec / 0.000 sec
3	10 06:13:04	SELECT k.nama_kasir, SUM(total_bayar) AS total_penjualan FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir	2 rows(s) returned	0.000 sec / 0.000 sec
4	11 06:13:31	SELECT k.nama_kasir, SUM(total_bayar) AS total_penjualan FROM transaksi t JOIN kasir k ON t.id_kasir = k.id_kasir	2 rows(s) returned	0.000 sec / 0.016 sec
5	12 06:13:59	COMMIT	0 row(s) affected	0.000 sec / 0.000 sec

DAFTAR PUSTAKA

1. Efendy Z. Normalisasi dalam desain database. 2018;4(1):34–43.
2. Hardini M. Application of Database Normalization in Increasing Data Storage Efficiency. 2025;3(2).
3. Azuddin M, Yusup M, Setiyowati H, Wibowo S, Suwita J, Astuti ED, et al. Design and Implementation of a Relational Database for an Academic Information System. 2025;3(2):171–80.
4. Karnasyah FP, Akbar M, Islami A, Wibisono OR. PERANCANGAN MODEL BASIS DATA TOKO CERIA MENGGUNAKAN MY STRUCTURED QUERY LANGUAGE (MYSQL). 2025;3(6):412–21.
5. Pane, Ramadhan A. Perancangan Basis Data Menggunakan Normalisasi Tabel pada Perusahaan dagang barokah abadi. J Dinda. 2022;2(2):90–6.