

**LAPORAN UJIAN AKHIR SEMESTER  
PERANCANGAN BASIS DATA SISTEM PENJUALAN  
DI TOKO LESTARI SUKSES MAKMUR**



Dosen Pengampu Mata Kuliah:

Ridwan Dwi Irawan, M.Kom

Disusun oleh:

Ayu Nurmawati	240103185
Djenar Priya Asoka	240103188
Dian Permata Sari	240103247

**PROGRAM STUDI S1-TEKNIK INFORMATIKA**

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

**KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga laporan Ujian Akhir Semester mata kuliah Pemrograman Basis Data ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu bentuk pemenuhan pertanggungjawaban atas pelaksanaan proyek kelompok yang telah dilakukan selama perkuliahan dengan bimbingan Bapak Ridwan Dwi Irawan, M.Kom. selaku dosen pengampu mata kuliah.

Laporan ini membahas perancangan dan implementasi sebuah sistem basis data penjualan yang disusun berdasarkan studi kasus yang relevan dengan materi perkuliahan. Penyusunan laporan ini mengikuti materi yang terdapat pada Jurnal Kuliah Pemrograman Basis Data, yang meliputi pengantar pemrograman basis data, pengenalan relasi tabel, penyusunan Entity Relationship Diagram (ERD), normalisasi data hingga Third Normal Form (3NF), penerapan ERD ke dalam DBMS menggunakan MySQL Workbench, serta implementasi perintah SQL yang mencakup Data Definition Language (DDL), Data Manipulation Language (DML), Transaction Control Language (TCL), agregasi dan HAVING, GROUP BY, serta JOIN.

Penulis mengucapkan terima kasih kepada Bapak Ridwan Dwi Irawan, M.Kom. atas bimbingan yang telah diberikan sehingga laporan ini dapat disusun dengan baik. Penulis menyadari bahwa laporan ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan sebagai bahan perbaikan di masa mendatang. Akhir kata, penulis berharap laporan ini dapat memberikan manfaat serta menambah wawasan dan pemahaman dalam bidang pemrograman basis data.

Surakarta, 14 November 2025

## DAFTAR ISI

COVER.....	0
KATA PENGANTAR.....	1
DAFTAR ISI .....	2
BAB I PENDAHULUAN .....	4
1.1.LatarBelakang .....	4
1.2. Ruang Lingkup Dan Batasan Masalah.....	5
1.3. Gambaran Umum Sistem.....	6
BAB II LANDASAN TEORI .....	6
2.1. Konsep Basis Data dan DBMS.....	6
2.2. Relasi Tabel Dalam Basis Data.....	7
2.3. Entity Relationship Diagram (ERD).....	7
2.4. Normalisasi Basis Data.....	8
2.4.1 Normalisasi 1NF.....	8
2.4.2 Normalisasi 2NF.....	8
2.4.3 Normalisasi 3NF.....	8
2.5. Structured Query Language (SQL) .....	9
2.5.1 Data Definition Language (DDL).....	9
2.5.2 Data Manipulation Language (DML).....	9
2.5.3 Transaction Control Language (TCL).....	10
2.6. Query Lanjutan .....	10
2.6.1 Agregasi dan HAVING.....	10
2.6.2 GROUP BY.....	11

2.6.3 JOIN .....	11
BAB III PERANCANGAN DAN IMPLEMENTASI .....	11
3.1. Analisis Kebutuhan Sistem .....	11
3.2. Proses Normalisasi Data .....	13
3.2.1. First Normal Form (1NF) .....	13
3.2.2. Second Normal Form (2NF) .....	14
3.2.3. Third Normal Form (3NF) .....	14
3.3. Relasi Antar Tabel .....	15
3.4. Perancangan Entity Relationship Diagram (ERD) .....	16
3.4.1. Entitas dan Atribut .....	17
3.4.2. Relasi Antar Entitas .....	19
3.4.3. Kesimpulan Perancangan ERD .....	19
3.5. Implementasi Basis Data .....	20
3.5.1. Implementasi Data Definition Language (DDL) .....	20
3.5.2. Implementasi Data Manipulation Language (DML) .....	21
3.5.3. Implementasi Transaction Control Language (TCL) .....	22
3.6. Implementasi Query Lanjutan .....	22
3.6.1. Implementasi Query JOIN .....	23
3.6.2. Implementasi Query GROUP BY dan Agregasi .....	23
3.6.3. Implementasi Query HAVING .....	23
BAB IV PENUTUP .....	25
4.1. Hasil Pengujian dan Pembahasan .....	25
4.2. Kendala dan Solusi .....	26
4.3. Kesimpulan .....	27
4.4. Saran .....	27
LAMPIRAN .....	29
DAFTAR PUSTAKA .....	34

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Perkembangan teknologi informasi yang semakin pesat menuntut pengelolaan data yang cepat, akurat, dan terstruktur. Dalam berbagai bidang, khususnya pada sistem penjualan, data transaksi yang terus bertambah memerlukan sistem pengelolaan yang baik agar informasi yang dihasilkan dapat digunakan secara

optimal. Pengelolaan data yang tidak terstruktur berpotensi menimbulkan berbagai permasalahan, seperti kesalahan pencatatan, duplikasi data, serta kesulitan dalam penyusunan laporan.

Basis data merupakan salah satu solusi utama dalam pengelolaan data terstruktur karena mampu menyimpan, mengelola, dan menampilkan data secara efisien (1). Melalui pemrograman basis data, data tidak hanya disimpan, tetapi juga dapat diolah menggunakan perintah Structured Query Language (SQL) untuk menghasilkan informasi yang dibutuhkan. Oleh karena itu, pemahaman terhadap konsep basis data, perancangan Entity Relationship Diagram (ERD), normalisasi data, serta implementasi SQL menjadi hal yang sangat penting (2).

Berdasarkan latar belakang tersebut, pada Ujian Akhir Semester mata kuliah Pemrograman Basis Data disusun sebuah proyek berupa perancangan dan implementasi sistem basis data penjualan. Proyek ini diharapkan mampu merepresentasikan penerapan konsep-konsep pemrograman basis data sesuai dengan materi yang telah dipelajari selama perkuliahan.

1. Menerapkan konsep dasar pemrograman basis data dalam sebuah studi kasus sistem penjualan.
2. Merancang struktur basis data menggunakan Entity Relationship Diagram (ERD).
3. Melakukan proses normalisasi data hingga bentuk Third Normal Form (3NF).
4. Mengimplementasikan hasil perancangan basis data ke dalam DBMS MySQL menggunakan MySQL Workbench.
5. Menggunakan perintah SQL yang meliputi Data Definition Language (DDL), Data Manipulation Language (DML), Transaction Control Language (TCL), serta query agregasi, HAVING, GROUP BY, dan JOIN.
6. Menghasilkan query dan laporan yang dapat menampilkan informasi penjualan secara akurat dan terstruktur.

## **1.2. Ruang Lingkup Dan Batasan Masalah**

1. Sistem yang dibangun difokuskan pada pengelolaan data penjualan, meliputi data produk, pelanggan, transaksi, dan detail transaksi.
2. Basis data yang digunakan adalah MySQL sebagai Database Management System (DBMS).

3. Perancangan database dilakukan menggunakan MySQL Workbench.
4. Implementasi sistem dibatasi pada pembuatan struktur tabel, relasi antar tabel, serta penulisan dan pengujian query SQL.
5. Pengembangan antarmuka aplikasi bersifat opsional dan tidak menjadi fokus utama dalam pembahasan laporan ini.

### **1.3. Gambaran Umum Sistem**

Sistem yang dibangun merupakan sistem basis data penjualan sederhana yang berfungsi untuk mengelola data transaksi penjualan secara terstruktur. Sistem ini terdiri dari beberapa tabel utama, antara lain tabel produk, pelanggan, transaksi, dan detail transaksi yang saling berelasi melalui primary key dan foreign key untuk menjaga konsistensi data.

Melalui sistem ini, pengguna dapat melakukan pencatatan data produk, pencatatan transaksi penjualan, serta pembuatan laporan penjualan menggunakan query SQL. Laporan yang dihasilkan meliputi informasi total penjualan, jumlah produk yang terjual, serta data transaksi berdasarkan kriteria tertentu sesuai kebutuhan pengguna.

## **BAB II LANDASAN TEORI**

### **2.1. Konsep Basis Data dan DBMS**

Basis data (database) merupakan sekumpulan data yang saling terkait dan disimpan secara terorganisir sehingga dapat diakses, dikelola, serta diperbarui dengan mudah. Basis data dirancang untuk meminimalkan terjadinya redundansi data, menjaga konsistensi, dan meningkatkan efisiensi dalam pengolahan informasi (3).

Database Management System (DBMS) hadir sebagai solusi dalam pengelolaan data yang lebih terstruktur dan efektif. DBMS memungkinkan perusahaan untuk menyimpan, mengakses, dan mengelola data secara lebih mudah serta aman. DBMS menyediakan berbagai fasilitas, seperti pendefinisian struktur data, pengolahan data, pengaturan hak akses, serta pengamanan data. Penerapan DBMS memungkinkan data dimanfaatkan oleh banyak pengguna secara bersamaan, mendukung terjaganya integritas data, serta menyediakan mekanisme pencadangan dan pemulihan data guna menjaga keamanan data

## 2.2. Relasi Tabel Dalam Basis Data

Relasi tabel merupakan hubungan logis yang terjalin antara dua atau lebih tabel dalam basis data relasional. Hubungan ini dibentuk melalui pemanfaatan *primary key* (PK) dan *foreign key* (FK). *Primary key* adalah atribut unik yang berfungsi untuk mengidentifikasi setiap baris data dalam suatu tabel, sementara *foreign key* merupakan atribut yang mengacu pada *primary key* di tabel lain (4).

Pada Sistem Penjualan Toko Lestari Sukses Makmur dapat disimpulkan bahwa struktur keterkaitan antar entitas telah dibangun secara terstruktur. Setiap tabel dilengkapi dengan Primary Key (PK) yang bersifat unik sebagai pengenalan utama, serta Foreign Key (FK) yang berperan dalam menjalin koneksi antar tabel dengan menghubungkan data produk dengan transaksi, data produk dengan detail produk, transaksi dengan detail produk dan data transaksi dengan pelanggan. Secara umum, hubungan antar tabel tersebut menciptakan suatu jaringan integrasi yang secara akurat merepresentasikan alur proses bisnis toko.

## 2.3. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model konseptual yang berfungsi untuk merepresentasikan struktur basis data secara visual. ERD menampilkan entitas, atribut, relasi, serta kardinalitas hubungan antar entitas. ERD dimanfaatkan sebagai landasan dalam perancangan basis data sebelum diterapkan ke dalam DBMS. ERD yang dirancang dengan baik akan memudahkan proses normalisasi serta penyusunan skema tabel (5).

ERD pada sistem Sistem Toko Lestari Sukses Makmur terdiri dari beberapa entitas yang mewakili tabel hasil normalisasi, yaitu produk, pelanggan, transaksi, detail produk, toko, kasir, dan owner. Setiap entitas memiliki atribut yang



merepresentasikan data penting, serta relasi yang menunjukkan keterkaitan antar entitas dalam proses Sistem Toko Lestari Sukses Makmur.

## **2.4. Normalisasi Basis Data**

Normalisasi merupakan proses dalam model data relasional yang bertujuan untuk mengorganisasi kumpulan data yang memiliki tingkat ketergantungan dan keterkaitan yang tinggi. Normalisasi digunakan sebagai parameter untuk menghindari terjadinya duplikasi tabel dalam basis data serta sebagai proses dekomposisi terhadap tabel yang masih mengandung anomali atau ketidakwajaran. Melalui proses ini, dihasilkan tabel yang lebih sederhana dan memiliki struktur yang baik, yaitu tabel yang tidak mengandung redundansi data serta memungkinkan pengguna melakukan proses insert, delete, dan update pada setiap baris (record) tanpa menimbulkan inkonsistensi data (6).

### **2.4.1 Normalisasi 1NF**

Tabel dikatakan berada pada bentuk normal pertama (1NF) apabila tidak mengandung kelompok data yang berulang. Untuk mengubah desain tabel yang belum ternormalisasi ke dalam bentuk 1NF, kunci utama tabel perlu diperluas dengan memasukkan kunci utama dari grup yang berulang. Dengan demikian, setiap record dapat diidentifikasi secara unik melalui primary key (7).

### **2.4.2 Normalisasi 2NF**

Second Normal Form (2NF) mensyaratkan bahwa suatu tabel telah memenuhi 1NF dan setiap atribut yang bukan primary key harus bergantung sepenuhnya pada primary key. Oleh karena itu, ketergantungan parsial terhadap sebagian primary key harus dihilangkan. Dengan kata lain, setiap atribut non-primary key harus bergantung secara fungsional pada seluruh atribut key, bukan hanya pada sebagian atribut saja. Penerapan bentuk 2NF bertujuan untuk menghilangkan adanya ketergantungan transitif (8).

### **2.4.3 Normalisasi 3NF**

Suatu relasi dikatakan memenuhi 3NF jika dan hanya jika telah memenuhi 2NF. Setiap atribut yang bukan merupakan kunci memiliki ketergantungan fungsional terhadap atribut bukan kunci lainnya dalam relasi

tersebut. Bentuk normal ketiga (3NF) bertujuan untuk menghilangkan anomali yang timbul akibat adanya ketergantungan fungsional.

Apabila tidak terdapat data yang mengandung nilai koma, maka tabel yang dihasilkan dinyatakan sudah tetap, proses normalisasi dianggap selesai, dan tabel tersebut dapat diimplementasikan ke dalam basis data relasional (9).

## **2.5. Structured Query Language (SQL)**

Structured Query Language (SQL) merupakan bahasa baku yang digunakan untuk berinteraksi dengan basis data relasional. SQL dimanfaatkan untuk mendefinisikan struktur data, mengolah data, serta mengatur transaksi dalam sistem manajemen basis data (DBMS) (10). SQL dikelompokkan ke dalam beberapa kategori berdasarkan fungsinya, seperti DDL, DML, dan TCL. Pengelompokan ini membantu pengembang memahami peran setiap perintah dalam proses pengelolaan basis data (11).

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

Data Definition Language (DDL) digunakan untuk mendefinisikan serta memodifikasi struktur basis data. Perintah DDL mencakup CREATE, ALTER, dan DROP yang berfungsi untuk membuat, mengubah, dan menghapus objek basis data, seperti tabel dan constraint (12). Perintah DDL di antaranya adalah CREATE, ALTER, DROP, TRUNCATE, COMMENT, dan RENAME.

### **2.5.2 Data Manipulation Language (DML)**

Data Manipulation Language (DML) digunakan untuk mengelola data yang tersimpan di dalam tabel basis data, mulai dari mengakses, menambahkan, memperbarui, hingga menghapus data. DML berfokus pada pengolahan isi data, bukan pada struktur basis data. Perintah DML dijalankan terhadap data yang telah didefinisikan sebelumnya melalui DDL dan berkaitan erat dengan konsep operasi CRUD dalam sistem basis data (13).

Jenis-jenis perintah DML mencakup INSERT yang digunakan untuk menambahkan data baru ke dalam tabel, SELECT untuk mengambil atau menampilkan data dari satu maupun beberapa tabel, UPDATE untuk

memperbarui data yang telah ada di dalam tabel, serta DELETE yang digunakan untuk menghapus data dari tabel berdasarkan kondisi tertentu.

### **2.5.3 Transaction Control Language (TCL)**

Transaction Control Language (TCL) digunakan untuk mengatur proses transaksi dalam basis data. Transaksi adalah kumpulan perintah SQL yang dijalankan sebagai satu kesatuan kerja. TCL berfungsi untuk menjaga konsistensi serta keandalan data, khususnya ketika terjadi kegagalan sistem atau kesalahan saat eksekusi (14).

Jenis-jenis perintah TCL meliputi START TRANSACTION untuk memulai transaksi, COMMIT untuk menyimpan perubahan data, ROLLBACK untuk membatalkan perubahan yang telah dilakukan, serta SAVEPOINT untuk membuat titik penyimpanan sementara di dalam suatu transaksi.

## **2.6. Query Lanjutan**

Query lanjutan dimanfaatkan untuk mengolah dan menyajikan data dengan tingkat kompleksitas yang lebih tinggi sesuai dengan kebutuhan sistem. Query ini menggunakan relasi antar tabel serta fungsi-fungsi tertentu guna menghasilkan informasi yang relevan dan bermakna (15).

Beberapa jenis query lanjutan yang diterapkan dalam proyek ini meliputi:

### **2.6.1 Agregasi dan HAVING**

Fungsi agregasi merupakan fungsi yang digunakan untuk mengambil suatu nilai dari sekumpulan data atau beberapa kumpulan data yang telah dikelompokkan. Secara umum, fungsi-fungsi agregasi yang tersedia dalam SQL meliputi SUM yang digunakan untuk menghitung total nilai, AVG untuk menghitung nilai rata-rata, MIN untuk memperoleh nilai paling rendah dari suatu kolom, MAX untuk mendapatkan nilai paling tinggi dari suatu kolom, serta COUNT yang berfungsi menghitung jumlah baris yang dipilih melalui suatu kueri (16).

Fungsi HAVING dalam basis data digunakan untuk menyaring hasil data setelah proses pengelompokan (GROUP BY) dilakukan. Berbeda dengan

klausa WHERE yang menyaring data sebelum proses pengelompokan, HAVING diterapkan pada hasil agregasi, sehingga berperan penting ketika ingin menetapkan kondisi terhadap nilai yang dihasilkan oleh fungsi agregat seperti SUM, COUNT, AVG, MAX, dan MIN.

### **2.6.2 GROUP BY**

Fungsi GROUP BY digunakan untuk mengelompokkan data berdasarkan satu atau beberapa kolom tertentu sehingga memungkinkan penerapan fungsi agregat seperti COUNT, SUM, AVG, MAX, dan MIN. Melalui GROUP BY, data yang memiliki nilai yang sama pada kolom tertentu akan digabungkan ke dalam satu kelompok, kemudian dilakukan perhitungan agregat pada setiap kelompok tersebut (17).

### **2.6.3 JOIN**

Fungsi JOIN digunakan untuk menggabungkan data dari dua atau lebih tabel berdasarkan hubungan antar kolom tertentu. JOIN memungkinkan pengguna memperoleh data yang saling berhubungan meskipun disimpan pada tabel yang berbeda. Dengan JOIN, informasi tersebut dapat ditampilkan dalam satu hasil kueri tanpa perlu menyimpan data secara berulang, sehingga membantu menjaga konsistensi serta efisiensi penyimpanan data (18).

## **BAB III PERANCANGAN DAN IMPLEMENTASI**

### **3.1. Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem dilakukan berdasarkan data transaksi nyata yang diperoleh dari struk penjualan toko CV Lestari Sukses Makmur. Struk tersebut berfungsi sebagai sumber utama dalam menentukan data apa saja yang harus disimpan dan dikelola oleh sistem basis data penjualan.



*Gambar 3.1 Nota Transaksi CV Lestari Sukses Makmur*

Berdasarkan struk transaksi, sistem membutuhkan beberapa jenis data utama, yaitu:

1. Data Toko

Data toko yang tercantum pada struk meliputi nama toko, Nomor Pokok Wajib Pajak (NPWP), dan nomor telepon. Data ini diperlukan untuk mengidentifikasi toko tempat terjadinya transaksi penjualan.

2. Data Transaksi

Data transaksi meliputi nomor struk, tanggal dan waktu transaksi, total pembelian, jumlah pembayaran, serta kembalian. Data ini digunakan sebagai catatan utama setiap proses penjualan yang terjadi.

3. Data Pelanggan

Pada struk tercantum informasi nama pelanggan. Oleh karena itu, sistem memerlukan data pelanggan untuk mencatat pihak yang melakukan transaksi, terutama untuk kebutuhan pelaporan dan riwayat transaksi.

4. Data Produk

Data produk meliputi nama produk, satuan penjualan (misalnya pax), dan harga satuan. Data ini diperlukan agar sistem dapat menampilkan informasi produk secara konsisten pada setiap transaksi.

### 5. Data Detail Transaksi

Detail transaksi memuat hubungan antara transaksi dan produk, yaitu jumlah produk yang dibeli, harga satuan, dan total harga per produk. Data ini penting untuk menghitung subtotal transaksi dan total penjualan.

### 6. Data Kasir

Meskipun tidak ditampilkan secara eksplisit pada struk, keberadaan kasir sebagai pihak yang melayani transaksi diperlukan dalam sistem untuk keperluan pengelolaan pengguna dan tanggung jawab transaksi.

Berdasarkan kebutuhan tersebut, sistem basis data dirancang untuk mampu:

- Menyimpan data transaksi secara terstruktur dan terintegrasi
- Mengelola hubungan antar data toko, pelanggan, kasir, produk, dan transaksi
- Menghasilkan laporan penjualan melalui query SQL seperti JOIN, GROUP BY, dan agregasi
- Menjaga konsistensi dan keakuratan data melalui penerapan primary key dan foreign key

## 3.2. Proses Normalisasi Data

Normalisasi merupakan proses sistematis untuk menyusun data agar lebih efisien, mengurangi duplikasi, dan menjaga konsistensi. Dalam sistem penjualan Toko

Lestari Sukses Makmur, normalisasi menjadi langkah penting untuk memisahkan data

transaksi, data pelanggan, data produk, dan data toko agar lebih terstruktur.

### 3.2.1. First Normal Form (1NF)

Tahap pertama normalisasi adalah memastikan bahwa semua atribut memiliki nilai atomik (tidak ada data yang mengandung lebih dari satu nilai dalam satu kolom) dan bahwa setiap baris data unik.

Setelah disesuaikan, hasil tabel 1NF adalah sebagai berikut:

Banyaknya	Nama Produk	Harga Satuan	Jumlah	NPWP	Telp
2 pax	Boncabe Makaroni	Rp.7.400	Rp.14.800	71.057.920.2-532.000	620479
3 pax	Criscito	Rp.8.200	Rp.24.600	71.057.920.2-532.000	620479
2 pax	Potakrezzz	Rp.7.450	Rp.14.900	71.057.920.2-532.000	620479

NO.Struk	NAMA Pelanggan	TGL	NAMA TOKO	ID TOKO	NO TLP
503837	NURMA	12/06/2025	CV Lestari Sukses Makmur	995	620479
503837	NURMA	12/06/2025	CV Lestari Sukses Makmur	995	620479
503837	NURMA	12/06/2025	CV Lestari Sukses Makmur	995	620479

*Tabel 3.2.1 Tabel Normalisasi 1NF (First Normal Form)*

Tabel ini sudah memenuhi bentuk 1NF karena seluruh data bersifat atomik dan memiliki identitas unik berdasarkan kombinasi No. Struk dan Nama Produk.

### 3.2.2. Second Normal Form (2NF)

Pada tahap 2NF, tabel harus sudah dalam 1NF dan tidak memiliki ketergantungan parsial, yaitu atribut non-kunci tidak boleh hanya bergantung pada sebagian dari primary key.

Karena primary key sementara adalah (No.Struk, Nama Produk), maka atribut seperti Nama Pelanggan, Tanggal, dan Nama Toko hanya bergantung pada No.Struk. Oleh karena itu, data harus dipisahkan menjadi beberapa tabel, yaitu:

Tabel Transaksi				
No.Struk	Tanggal	Nama Pelanggan	Telp Pelanggan	ID Toko
503837	12/06/2025	NURMA	620479	995

Tabel Toko			
ID Toko	Nama Toko	NPWP	No. Tlp
995	CV Lestari Sukses	71.057.920.2-532.000	620479

Tabel Detail Transaksi				
No.Struk	Nama Produk	Harga Satuan	Banyaknya	Jumlah
503837	Boncabe	Rp.7.400	2 pax	Rp.14.800
503837	Criscito	Rp.8.200	3 pax	Rp.24.600
503837	Potakrezzz	Rp.7.450	2 pax	Rp.14.900

*Tabel 3.2.2 Tabel Normalisasi 2NF (Second Normal Form)*

Dengan pemisahan ini, data menjadi lebih efisien dan tidak terjadi pengulangan informasi toko atau pelanggan di setiap baris produk.

### 3.2.3. Third Normal Form (3NF)

Tahap 3NF memastikan bahwa tidak ada ketergantungan transitif, yaitu atribut non-kunci tidak boleh bergantung pada atribut non-kunci lainnya. Pada tabel hasil

2NF, masih terdapat ketergantungan Nama Pelanggan dan Telp Pelanggan. Oleh karena itu, dibuat tabel baru untuk data pelanggan agar lebih terstruktur.

Tabel Pelanggan		
ID Pelanggan	Nama Pelanggan	Telp
P001	NURMA	620479

Tabel Toko			
ID Toko	Nama Toko	NPWP	No Tlp
995	CV Lestari Sukses	71.057.920.2-532.000	620479

Tabel Transaksi			
No.Struk	ID Pelanggan	ID Toko	Tanggal
503837	P001	995	12/06/2025

Tabel Detail Transaksi			
No.Struk	Nama Produk	Harga Satuan	Banyaknya
503837	Boncabe	Rp.7.400	2 pax
503837	Criscito	Rp.8.200	3 pax
503837	Potakrezzz	Rp.7.450	2 pax

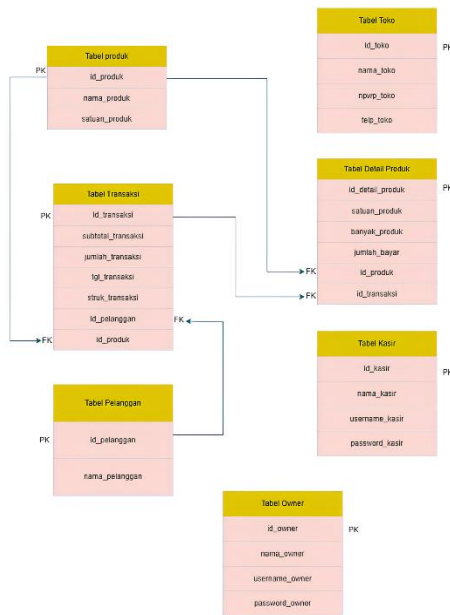
*Tabel 3.2.3 Tabel Normalisasi 3NF (Third Normal Form)*

### 3.3. Relasi Antar Tabel

Relasi antar tabel pada sistem basis data penjualan ini dibentuk berdasarkan hasil analisis kebutuhan sistem serta proses normalisasi data. Perancangan relasi bertujuan untuk memastikan bahwa setiap data saling terhubung dengan baik, tidak terjadi redundansi data, serta menjaga konsistensi dan integritas data di dalam basis data.

Relasi antar tabel ditentukan dengan menggunakan Primary Key (PK) sebagai penanda data unik pada setiap tabel dan Foreign Key (FK) sebagai penghubung antar tabel. Selain itu, relasi juga memperhatikan derajat kardinalitas, yaitu hubungan satu ke satu (one to one), satu ke banyak (one to many), atau banyak ke banyak (many to many).





*Gambar 3.3 Relasi Antar Tabel*

Berdasarkan gambar relasi antar tabel, sistem basis data penjualan ini terdiri dari beberapa tabel utama, yaitu tabel pelanggan, tabel transaksi, tabel detail produk, tabel produk, tabel kasir, tabel toko, dan tabel owner.

Dengan adanya relasi antar tabel tersebut, sistem basis data penjualan dapat berjalan secara terstruktur, konsisten, dan efisien. Setiap data saling terhubung melalui primary key dan foreign key, sehingga mendukung integritas referensial, memudahkan proses pencarian data, serta membantu dalam pembuatan laporan penjualan yang akurat.

### 3.4. Perancangan Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model konseptual yang digunakan untuk menggambarkan struktur basis data secara visual. ERD menunjukkan hubungan antara entitas, atribut yang dimiliki setiap entitas, serta relasi dan kardinalitas antar entitas. Perancangan ERD bertujuan untuk memastikan bahwa data yang disimpan dalam basis data terstruktur dengan baik, tidak redundan, dan mendukung kebutuhan sistem secara optimal.

ERD pada sistem penjualan ini disusun berdasarkan hasil analisis kebutuhan sistem dan telah melalui proses normalisasi hingga bentuk normal ketiga (Third Normal Form/3NF). ERD ini menjadi acuan utama dalam proses implementasi basis data pada DBMS MySQL.

### 3.4.1. Entitas dan Atribut

Berdasarkan ERD yang telah dirancang, terdapat beberapa entitas utama yang membentuk sistem penjualan, yaitu sebagai berikut:

#### 1. Entitas Produk

Entitas produk digunakan untuk menyimpan data barang yang dijual di toko. Entitas ini memiliki atribut:

- id\_produk sebagai *primary key*
- nama\_produk
- satuan\_produk

Setiap produk memiliki identitas unik dan dapat muncul pada banyak transaksi melalui tabel detail produk.

#### 2. Entitas Toko

Entitas toko menyimpan informasi identitas toko tempat transaksi dilakukan. Atribut yang dimiliki antara lain:

- id\_toko sebagai *primary key*
- nama\_toko
- npwp\_toko
- telp\_toko

Data toko digunakan sebagai informasi administratif dan pendukung laporan transaksi.

#### 3. Entitas Pelanggan

Entitas pelanggan berfungsi untuk menyimpan data pelanggan yang melakukan transaksi. Atributnya meliputi:

- id\_pelanggan sebagai *primary key*
- nama\_pelanggan

Satu pelanggan dapat melakukan lebih dari satu transaksi.

#### 4. Entitas Kasir

Entitas kasir digunakan untuk menyimpan data pengguna sistem yang melayani transaksi. Atributnya terdiri dari:

- id\_kasir sebagai *primary key*
- nama\_kasir
- username\_kasir
- password\_kasir

Data kasir diperlukan untuk keperluan autentikasi dan pencatatan siapa yang melayani transaksi.

#### 5. Entitas Owner

Entitas owner menyimpan data pemilik atau pengelola sistem. Atribut yang digunakan adalah:

- id\_owner sebagai *primary key*
- nama\_owner
- username\_owner
- password\_owner

Entitas ini berfungsi sebagai pengelola utama sistem.

#### 6. Entitas Transaksi

Entitas transaksi mencatat setiap proses penjualan yang terjadi. Atribut yang dimiliki antara lain:

- id\_transaksi sebagai *primary key*
- struk\_transaksi
- tgl\_transaksi
- subtotal\_transaksi
- metode\_pembayaran
- id\_pelanggan sebagai *foreign key*

Entitas transaksi menjadi pusat relasi dengan entitas lain dalam sistem.

#### 7. Entitas Detail Produk (Detail Transaksi)

Entitas *detail\_produk* digunakan untuk menyimpan rincian produk yang terlibat dalam setiap transaksi. Atributnya meliputi:

- *id\_detail\_produk* sebagai *primary key*
- *satuan\_produk*
- *banyak\_produk*
- *jumlah\_bayar*
- *id\_produk* sebagai *foreign key*
- *id\_transaksi* sebagai *foreign key*

Entitas ini berfungsi sebagai penghubung antara produk dan transaksi.

### 3.4.2. Relasi Antar Entitas

Relasi antar entitas dalam ERD ini menggambarkan keterkaitan data dalam sistem penjualan, yaitu:

1. Relasi Produk dengan Detail Produk (TERCATAT).  
Satu produk dapat tercatat pada banyak detail produk, sedangkan satu detail produk hanya berkaitan dengan satu produk. Relasi ini memiliki kardinalitas 1 : M.
2. Relasi Transaksi dengan Detail Produk (MEMILIKI).  
Satu transaksi dapat memiliki banyak detail produk, sedangkan satu detail produk hanya berasal dari satu transaksi. Relasi ini juga memiliki kardinalitas 1 : M.
3. Relasi Pelanggan dengan Transaksi (MELAKUKAN).  
Satu pelanggan dapat melakukan banyak transaksi, sedangkan satu transaksi hanya dilakukan oleh satu pelanggan. Kardinalitas relasi ini adalah 1 : M.
4. Relasi Kasir dengan Transaksi (MELAYANI).  
Satu kasir dapat melayani banyak transaksi, sedangkan satu transaksi hanya dilayani oleh satu kasir. Relasi ini memiliki kardinalitas 1 : M.

### 3.4.3. Kesimpulan Perancangan ERD

Berdasarkan perancangan ERD yang telah dibuat, dapat disimpulkan bahwa struktur basis data telah disusun secara terintegrasi dan sistematis. Seluruh entitas

utama saling terhubung melalui relasi yang jelas dengan penggunaan primary key dan foreign key. ERD ini telah memenuhi prinsip normalisasi hingga 3NF sehingga meminimalkan redundansi data dan meningkatkan konsistensi data.

ERD final ini digunakan sebagai dasar dalam proses penerapan basis data ke dalam DBMS MySQL, termasuk pembuatan tabel, penentuan relasi, serta implementasi constraint pada tahap implementasi sistem.

### **3.5. Implementasi Basis Data**

Implementasi basis data dilakukan menggunakan sistem manajemen basis data MySQL. Pada tahap ini, struktur basis data dibangun berdasarkan Entity Relationship Diagram (ERD) yang telah dirancang. Proses implementasi mencakup pembuatan struktur tabel, pengisian data, serta pengelolaan transaksi untuk memastikan konsistensi dan integritas data dalam sistem penjualan.

#### **3.5.1. Implementasi Data Definition Language (DDL)**

Data Definition Language (DDL) digunakan untuk mendefinisikan struktur basis data, meliputi pembuatan database, pembuatan tabel, serta penentuan primary key dan foreign key sesuai dengan hasil perancangan ERD. Penerapan DDL bertujuan untuk membangun struktur basis data yang terorganisir dan mendukung relasi antar tabel.

```

1 • CREATE DATABASE uas_pbd_penjualan;
2
3 • USE uas_pbd_penjualan;
4
5 • CREATE TABLE pelanggan (
6     id_pelanggan INT AUTO_INCREMENT PRIMARY KEY,
7     nama_pelanggan VARCHAR(100)
8 );
9
10 • CREATE TABLE toko (
11     id_toko INT AUTO_INCREMENT PRIMARY KEY,
12     nama_toko VARCHAR(100),
13     npwp VARCHAR(50),
14     telp VARCHAR(20)
15 );
16
17 • CREATE TABLE kasir (
18     id_kasir INT AUTO_INCREMENT PRIMARY KEY,
19     nama_kasir VARCHAR(100),
20     username_kasir VARCHAR(50),
21     password_kasir VARCHAR(100)
22 );
23
24 • CREATE TABLE produk (
25     id_produk INT AUTO_INCREMENT PRIMARY KEY,
26     nama_produk VARCHAR(100),
27     satuan_produk VARCHAR(50)
28 );
29
30 • CREATE TABLE transaksi (
31     id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
32     no_struk INT UNIQUE,
33     tgl_transaksi DATE,
34     id_pelanggan INT,
35     id_toko INT,
36     id_kasir INT,
37
38     FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan),
39     FOREIGN KEY (id_toko) REFERENCES toko(id_toko),
40     FOREIGN KEY (id_kasir) REFERENCES kasir(id_kasir)
41 );
42
43 • CREATE TABLE detail_transaksi (
44     id_detail INT AUTO_INCREMENT PRIMARY KEY,
45     id_transaksi INT,
46     id_produk INT,
47     harga_satuan DECIMAL(10,2),
48     qty INT,
49     subtotal DECIMAL(10,2),
50
51     FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi),
52     FOREIGN KEY (id_produk) REFERENCES produk(id_produk)
53 );

```

### 3.5.2. Implementasi Data Manipulation Language (DML)

Data Manipulation Language (DML) digunakan untuk melakukan manipulasi data pada tabel yang telah dibuat. Pada tahap ini dilakukan proses pengisian data awal serta pengolahan data untuk memastikan data tersimpan dan dapat ditampilkan dengan benar sesuai kebutuhan sistem.

```

55 • INSERT INTO pelanggan (nama_pelanggan)
56 VALUES ('NURMA');
57
58 • INSERT INTO toko (nama_toko, npwp, telp)
59 VALUES ('CV Lestari Sukses Makmur', '71.057.920.2-532.000', '620479');
60
61 • INSERT INTO kasir (nama_kasir, username_kasir, password_kasir)
62 VALUES ('Siti', 'siti01', '12345');
63
64 • INSERT INTO produk (nama_produk, satuan_produk)
65 VALUES
66 ('Boncabe Makaroni', 'pax'),
67 ('Criscito', 'pax'),
68 ('Potakrezzz', 'pax');
69
72 • INSERT INTO transaksi (no_struk, tgl_transaksi, id_pelanggan, id_toko, id_kasir)
73 VALUES (503837, '2025-06-12', 1, 1, 1);
74
77 • INSERT INTO detail_transaksi (id_transaksi, id_produk, harga_satuan, qty, subtotal)
78 VALUES
79 (1, 1, 7400, 2, 14800),
80 (1, 2, 8200, 3, 24600),
81 (1, 3, 7450, 2, 14900);
82

```

### 3.5.3. Implementasi Transaction Control Language (TCL)

Transaction Control Language (TCL) digunakan untuk mengatur proses transaksi dalam basis data. Penerapan TCL bertujuan untuk memastikan bahwa proses penyimpanan data transaksi berjalan secara konsisten dan dapat dikendalikan dengan baik.

```

70 • START TRANSACTION;

75 • COMMIT;

```

## 3.6. Implementasi Query Lanjutan

Implementasi query lanjutan bertujuan untuk mengolah data yang tersimpan dalam basis data menjadi informasi yang berguna. Query yang digunakan disesuaikan dengan materi pada jurnal perkuliahan Pemrograman Basis Data, meliputi JOIN, GROUP BY, HAVING, serta fungsi agregasi.

### 3.6.1. Implementasi Query JOIN

Query JOIN digunakan untuk menggabungkan data dari beberapa tabel yang saling berelasi sehingga informasi transaksi dapat ditampilkan secara lengkap.

```
83 • SELECT * FROM produk;  
84  
85  
86 • SELECT p.nama_pelanggan, t.no_struk, t.tgl_transaksi  
87 FROM transaksi t  
88 JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan;  
89
```

### 3.6.2. Implementasi Query GROUP BY dan Agregasi

Query GROUP BY digunakan untuk mengelompokkan data, sedangkan fungsi agregasi digunakan untuk menghitung data hasil pengelompokan tersebut, seperti total jumlah produk yang terjual.

```
90 • SELECT id_produk, SUM(qty) AS total_terjual  
91 FROM detail_transaksi  
92 GROUP BY id_produk;  
93  
94 • SELECT  
95     p.id_produk,  
96     p.nama_produk,  
97     SUM(d.qty) AS total_terjual  
98 FROM detail_transaksi d  
99 JOIN produk p ON d.id_produk = p.id_produk  
100 GROUP BY p.id_produk, p.nama_produk;
```

### 3.6.3. Implementasi Query HAVING

Query HAVING digunakan untuk memberikan kondisi pada hasil pengelompokan data sehingga hanya data yang memenuhi kriteria tertentu yang ditampilkan.



```
102 • SELECT id_produk, SUM(subtotal) AS total_penjualan
103 FROM detail_transaksi
104 GROUP BY id_produk
105 HAVING total_penjualan > 20000;
106
```

## **BAB IV PENUTUP**

### **4.1. Hasil Pengujian dan Pembahasan**

Pengujian sistem basis data penjualan dilakukan untuk memastikan bahwa seluruh struktur tabel, relasi antar tabel, serta query yang telah diimplementasikan dapat berjalan sesuai dengan perancangan. Pengujian dilakukan dengan mengeksekusi perintah SQL pada DBMS MySQL dan mengamati hasil yang dihasilkan dari setiap proses yang dilakukan.

Hasil pengujian menunjukkan bahwa seluruh tabel berhasil dibuat sesuai dengan perancangan ERD, lengkap dengan penerapan primary key dan foreign key. Relasi antar tabel dapat berjalan dengan baik, yang ditunjukkan oleh keberhasilan penyimpanan data transaksi dan detail transaksi tanpa terjadi pelanggaran constraint. Penggunaan foreign key mampu menjaga konsistensi data, sehingga data transaksi hanya dapat disimpan apabila data pelanggan, toko, kasir, dan produk telah tersedia.

Pengujian terhadap perintah Data Manipulation Language (DML) menunjukkan bahwa proses pengisian data awal berjalan dengan baik. Data pelanggan, toko, kasir, produk, transaksi, dan detail transaksi dapat disimpan dan ditampilkan kembali dengan benar. Selain itu, pengujian Transaction Control Language (TCL) membuktikan bahwa proses transaksi dapat dikendalikan dengan baik, di mana perubahan data hanya tersimpan secara permanen setelah perintah COMMIT dijalankan.

Pengujian query lanjutan menunjukkan bahwa query JOIN berhasil menggabungkan data dari beberapa tabel yang saling berelasi dan menampilkan informasi transaksi secara lengkap. Query GROUP BY dan fungsi agregasi mampu

menghasilkan rekapitulasi data penjualan, seperti total jumlah produk yang terjual dan total nilai penjualan. Penggunaan HAVING juga berjalan dengan baik dalam menyaring hasil agregasi sesuai dengan kriteria yang ditentukan.

Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa sistem basis data penjualan telah berfungsi sesuai dengan tujuan perancangan. Seluruh fitur utama yang berkaitan dengan penyimpanan data, pengolahan data, serta penyajian informasi telah berjalan dengan baik dan sesuai dengan materi yang terdapat pada Jurnal Kuliah Pemrograman Basis Data.

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

Selama proses perancangan dan implementasi sistem basis data penjualan, terdapat beberapa kendala yang dihadapi. Kendala pertama berkaitan dengan penerapan relasi antar tabel menggunakan foreign key. Pada tahap awal, beberapa tabel tidak dapat dibuat karena urutan pembuatan tabel yang belum sesuai, sehingga menyebabkan terjadinya kesalahan pada saat penetapan foreign key. Solusi dari kendala ini adalah dengan memastikan bahwa tabel induk dibuat terlebih dahulu sebelum tabel yang memiliki foreign key, sehingga relasi antar tabel dapat diterapkan dengan benar.

Kendala berikutnya muncul pada proses pengisian data transaksi, khususnya terkait dengan nilai unik pada atribut nomor struk. Kesalahan terjadi ketika data transaksi dimasukkan dengan nomor struk yang sama, sehingga sistem menolak penyimpanan data. Solusi yang dilakukan adalah memastikan bahwa setiap transaksi memiliki nomor struk yang unik serta menyesuaikan data uji agar tidak melanggar constraint yang telah ditetapkan.

Selain itu, kendala juga ditemukan pada pembuatan query lanjutan, terutama pada penggunaan GROUP BY dan HAVING. Pada awal pengujian, hasil query belum sesuai dengan yang diharapkan karena kurang tepat dalam menentukan kolom pengelompokan dan kondisi penyaringan. Solusi yang dilakukan adalah dengan menyesuaikan struktur query sesuai dengan relasi tabel dan menerapkan fungsi agregasi dengan benar.

Berdasarkan solusi yang telah diterapkan, seluruh kendala yang muncul dapat diatasi sehingga sistem basis data penjualan dapat berjalan dengan baik. Kendala-kendala tersebut juga menjadi pembelajaran penting dalam memahami penerapan konsep relasi, constraint, dan query lanjutan dalam sistem basis data.

#### **4.3. Kesimpulan**

Berdasarkan hasil analisis, perancangan, dan pengembangan sistem basis data pada Sistem Penjualan Toko Lestari Sukses Makmur, dapat disimpulkan bahwa proses perancangan basis data telah dilaksanakan secara sistematis dan selaras dengan kebutuhan operasional toko. Tahapan perancangan diawali dengan analisis data transaksi melalui nota penjualan, dilanjutkan dengan penyusunan tabel umum (unnormalized table), hingga penerapan normalisasi data sampai bentuk normal ketiga (Third Normal Form/3NF). Melalui tahapan normalisasi tersebut, struktur basis data menjadi lebih tertata, efisien, serta mampu mengurangi terjadinya redundansi dan anomali data.

Hasil perancangan relasi antar tabel serta penyusunan Entity Relationship Diagram (ERD) telah berhasil merepresentasikan hubungan antar entitas utama, seperti pelanggan, produk, transaksi, detail produk, kasir, dan toko, secara jelas dan terstruktur. Penerapan Primary Key (PK) dan Foreign Key (FK) pada setiap tabel berperan dalam menjaga integritas data serta memastikan keterkaitan antar tabel berjalan sesuai dengan aturan relasi yang telah ditetapkan.

Dengan desain basis data yang dihasilkan, sistem ini mampu menunjang proses pencatatan, pengelolaan, dan pengolahan data transaksi penjualan secara lebih efektif dan konsisten. Oleh karena itu, sistem basis data yang dirancang dinilai telah memenuhi tujuan proyek, yaitu menyediakan fondasi basis data yang terstruktur dan siap dikembangkan guna mendukung kebutuhan operasional serta analisis data penjualan di Toko Lestari Sukses Makmur.

#### **4.4. Saran**

Berdasarkan hasil perancangan sistem basis data yang telah dilakukan, terdapat sejumlah saran yang dapat dipertimbangkan untuk pengembangan ke tahap selanjutnya. Sistem basis data ini dapat dikembangkan dengan menambahkan

antarmuka aplikasi sehingga proses penginputan, pengelolaan, dan pencarian data dapat dilakukan dengan lebih efektif dan mudah oleh pengguna, khususnya kasir dan admin toko. Penerapan mekanisme pencadangan (backup) basis data secara berkala juga perlu dilakukan untuk menjaga keamanan dan keutuhan data dari potensi kehilangan atau kerusakan.

Dengan adanya pengembangan tersebut, sistem basis data yang dirancang diharapkan mampu memberikan dukungan yang lebih optimal terhadap operasional toko dan menjadi landasan bagi pengembangan sistem informasi penjualan yang lebih kompleks di masa mendatang.

## LAMPIRAN

### 1. Tautan Repository Github

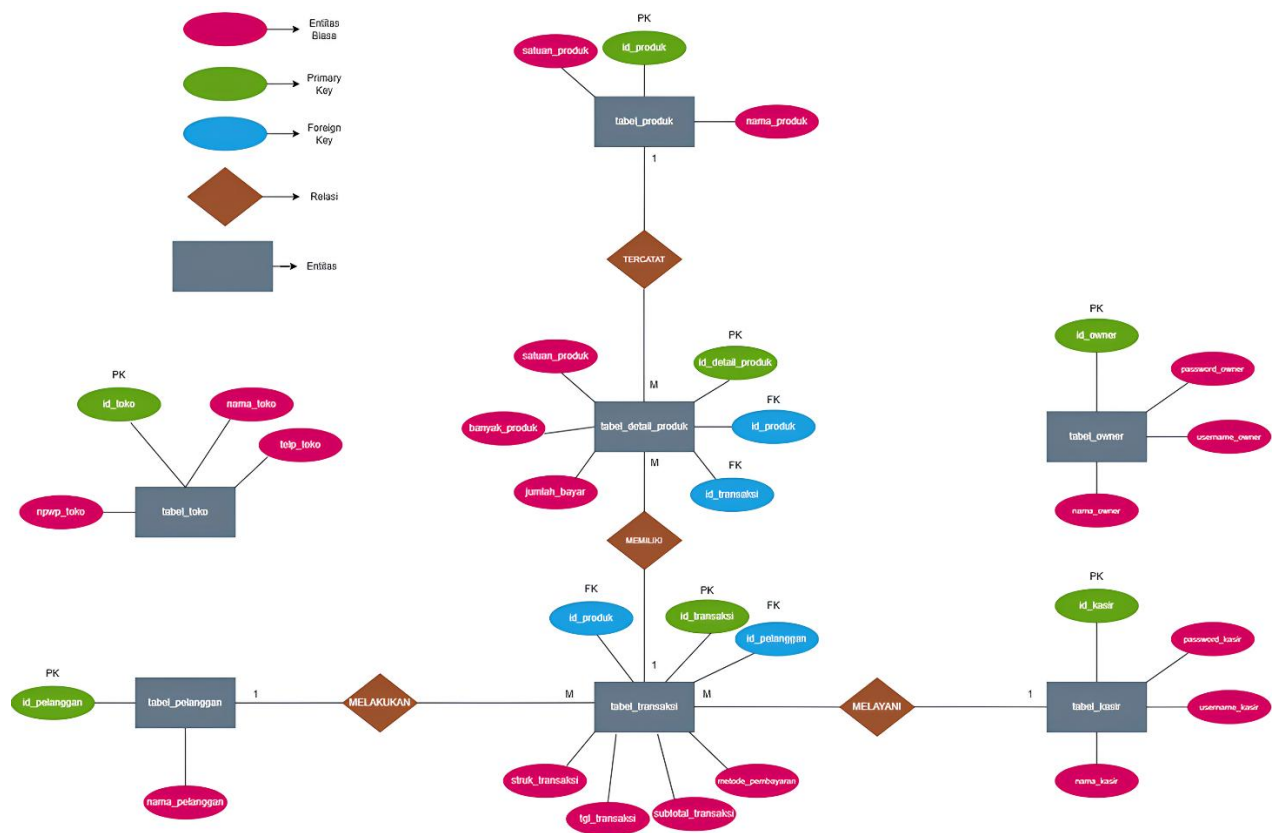
Link: <https://github.com/djenarganteng/UAS-PBD-TI24A6-Kelompok-5>

Ringkasan Struktur Folder

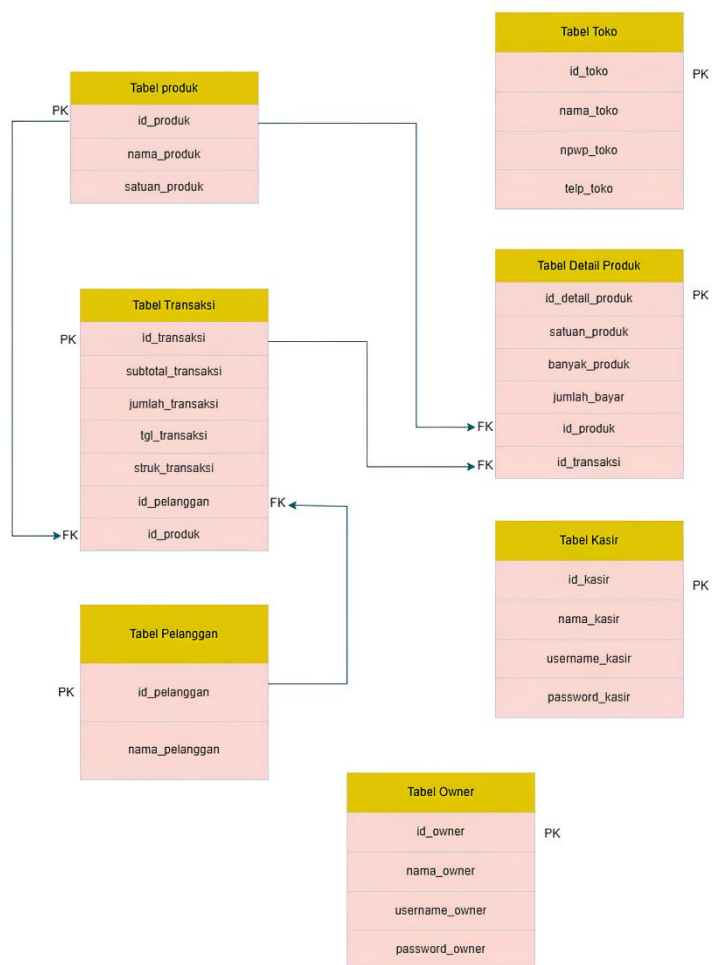
UAS PBD-TI24A6-Kelompok 5/

```
|
|├── Laporan/
|   ├── Laporan UAS PBD.docx
|   └── Laporan UAS PBD.pdf
|
|├── Poster/
|   └── poster.pdf
|
|├── SQL/
|   └── uas_pbd_sistem_penjualan.sql
|
└── README.md
```

## 2. Entity Relationship Diagram (ERD)



### 3. Relasi Antar Tabel





#### 4. Screenshot Implementasi Database

The top screenshot shows the phpMyAdmin interface for the database 'uas\_pbd\_penjualan'. The 'Struktur' (Structure) tab is active, displaying a table list with columns for 'Tabel', 'Tindakan', 'Baris', 'Jenis', 'Penyortiran', 'Ukuran', and 'Beban'. The table 'transaksi' is highlighted. Below the table list, there is a 'Create new table' form with fields for 'Nama tabel' and 'Jumlah kolom' (set to 4).

The bottom screenshot shows the 'Relational' tab in phpMyAdmin, displaying a diagram of the database relationships. The tables and their fields are:

- uas\_pbd\_penjualan.toko**: id\_toko (int(11)), nama\_toko (varchar(100)), npwp (varchar(50)), hlp (varchar(20)).
- uas\_pbd\_penjualan.produk**: id\_produk (int(11)), nama\_produk (varchar(100)), satuan\_produk (varchar(50)).
- uas\_pbd\_penjualan.detail\_transaksi**: id\_detail (int(11)), id\_transaksi (int(11)), id\_produk (int(11)), harga\_satuan (decimal(10,2)), qty (int(11)), subtotal (decimal(10,2)).
- uas\_pbd\_penjualan.transaksi**: id\_transaksi (int(11)), no\_struk (int(11)), tgl\_transaksi (date), id\_pelanggan (int(11)), id\_toko (int(11)), id\_kasir (int(11)).
- uas\_pbd\_penjualan.pelanggan**: id\_pelanggan (int(11)), nama\_pelanggan (varchar(100)).
- uas\_pbd\_penjualan.kasir**: id\_kasir (int(11)), nama\_kasir (varchar(100)), username\_kasir (varchar(50)), password\_kasir (varchar(100)).

Relationships are indicated by lines connecting the tables to their respective fields in the diagram.

#### 5. Hasil Eksekusi Query

83 • `SELECT * FROM produk;`

84

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
id_produk	nama_produk	satuan_produk				
1	Boncabe Makaroni	pax				
2	Crisito	pax				
3	Potakrezzz	pax				
•	NULL	NULL				

produk 7 x

Apply

86 • `SELECT p.nama_pelanggan, t.no_struk, t.tgl_transaksi`

87 `FROM transaksi t`

88 `JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan;`

89

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
nama_pelanggan	no_struk	tgl_transaksi			
NURMA	503837	2025-06-12			

90 • `SELECT id_produk, SUM(qty) AS total_terjual`

91 `FROM detail_transaksi`

92 `GROUP BY id_produk;`

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
id_produk	total_terjual				
1	2				
2	3				
3	2				

94 • `SELECT`

95 `p.id_produk,`

96 `p.nama_produk,`

97 `SUM(d.qty) AS total_terjual`

98 `FROM detail_transaksi d`

99 `JOIN produk p ON d.id_produk = p.id_produk`

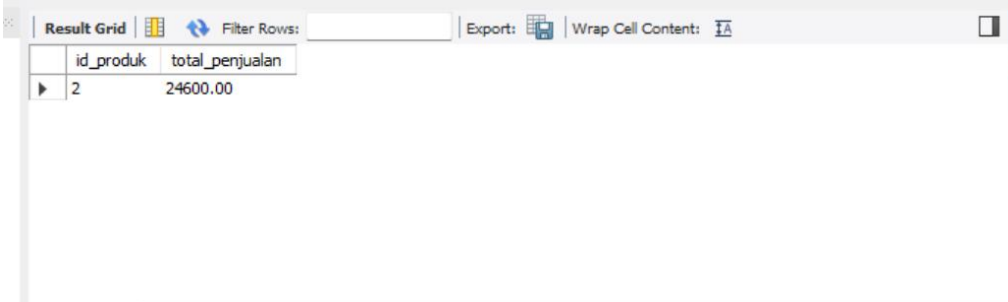
100 `GROUP BY p.id_produk, p.nama_produk;`

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
id_produk	nama_produk	total_terjual			
1	Boncabe Makaroni	2			
2	Crisito	3			
3	Potakrezzz	2			

Result 10

Read Only

```
102 • SELECT id_produk, SUM(subtotal) AS total_penjualan
103 FROM detail_transaksi
104 GROUP BY id_produk
105 HAVING total_penjualan > 20000;
106
```



id_produk	total_penjualan
2	24600.00

## 6. Script SQL Lengkap

Script SQL lengkap tersedia pada folder /sql di repository GitHub

## DAFTAR PUSTAKA

1. Coronel C, Morris S. Database systems: design, implementation, and management. 12e ed. 2015. 791 hlm.
2. Niamilah A, Alfin AA, Kurniasari I. Siklus Hidup Pengembangan Sistem Basis Data Pada Sistem Informasi Buku Tamu di Badan Pusat Statistik Kabupaten Kediri Menggunakan MySQL. J Nas Komputasi Dan Teknol Inf JNKTI [Internet]. 24 Februari 2023 [dikutip 20 Januari 2026];6(1):115–21. Tersedia pada: <https://ojs.serambimekkah.ac.id/jnkti/article/view/5830>
3. Marliana I, Hartanto MB. Perancangan dan Implementasi Basis Data Relasional untuk Sistem Informasi Perpustakaan Berbasis Web.
4. Haryanto B, Ardiansyah A, Kurniasih M. PENGENALAN DATABASE NOSQL DAN PERBANDINGANNYA DENGAN DATABASE RELASIONAL. 2024;12(1).
5. Pulungan SM, Febrianti R, Lestari T, Gurning N, Fitriana N. Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database. J Ekon Manaj Dan Bisnis JEMB [Internet]. 1 Februari 2023 [dikutip 20 Januari 2026];1(2):98–102. Tersedia pada: <https://jurnal.minartis.com/index.php/jemb/article/view/533>
6. Efendy Z. NORMALISASI DALAM DESAIN DATABASE. 2018;
7. Dina DFM. Normalisasi Database Rancangan Sistem Penyewaan Buku Berbayar. Comput Insight J Comput Sci [Internet]. 14 Mei 2024 [dikutip 20 Januari 2026];4(1):56–61. Tersedia pada: <https://journal.um-surabaya.ac.id/CI/article/view/15814>
8. Mulyati S, Sujatmoko BA, Wira TIM, Afif R, Pratama RA. NORMALISASI DATABASE DAN MIGRASI DATABASE UNTUK MEMUDAHKAN MANAJEMEN DATA. Sebatik

- [Internet]. 4 Desember 2018 [dikutip 20 Januari 2026];22(2):124–9. Tersedia pada: <https://jurnal.wicida.ac.id/index.php/sebatik/article/view/319>
9. Firdaus U, Imannudin A, Saputro MCA, Atallah MB, Najmudin MN, Kholifah AS. PERANCANGAN DAN IMPLEMENTASI BASIS DATA RELASIONAL UNTUK SISTEM E-COMMERCE BERBASIS WEB. 2025;1.
  10. Rachmawati R. ANALISIS KESALAHAN MENERAPKAN BAHASA SQL (STRUCTURE QUERY LANGUAGE) MATA KULIAH BASIS DATA. Prism J Pendidik Dan Ris Mat [Internet]. 1 Juni 2019 [dikutip 21 Januari 2026];1(2):27–34. Tersedia pada: <http://ejurnal.budiutomomalang.ac.id/index.php/prismatika/article/view/431>
  11. Fikry M, Yusra Y, Hidayat T. Pembangkitan Formulir Web Berdasarkan Metadata SQL dan Spesifikasi W3C. J CoreIT J Has Penelit Ilmu Komput Dan Teknol Inf [Internet]. 3 Januari 2018 [dikutip 21 Januari 2026];3(2):63. Tersedia pada: <http://ejournal.uin-suska.ac.id/index.php/coreit/article/view/4417>
  12. R S, Salim Y, Hasnawi M. Konversi Bahasa Indonesia ke Perintah Data Manipulation Language pada Structured Query Language menggunakan Natural Language Processing. Bul Sist Inf Dan Teknol Islam [Internet]. 31 Agustus 2022 [dikutip 21 Januari 2026];3(3):181–7. Tersedia pada: <https://jurnal.fikom.umi.ac.id/index.php/BUSITI/article/view/790>
  13. Setiyadi D. Structured Query Language (SQL) Untuk Purchase Order (PO) Menggunakan SQL Server 2008.
  14. Ikhsan A. PERANCANGAN BASIS DATA UNTUK PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEB JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI PADANG. 2013;
  15. Muppala M. SQL Database Mastery: Relational Architectures, Optimization Techniques, and Cloud-Based Applications [Internet]. Deep Science Publishing; 2025 [dikutip 21 Januari 2026]. Tersedia pada: <https://www.deepscienceresearch.com/dsr/catalog/book/231>
  16. Hussein DL. Evaluating Aggregate Functions and Machine Learning Integration: A Comparative Analysis of Performance, Security, and NoSQL Connectivity in Oracle, SQL Server, and MySQL. UHD J Sci Technol [Internet]. 22 September 2024 [dikutip 21 Januari 2026];8(2):7–23. Tersedia pada: <https://journals.uhd.edu.iq/index.php/uhdjst/article/view/1347>
  17. Tm H, K U, Shafiulla M, Dadapeer. An Overview of SQL Optimization Techniques for Enhanced Query Performance. Dalam: 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) [Internet]. Ballar, India: IEEE; 2023 [dikutip 21 Januari 2026]. hlm. 1–5. Tersedia pada: <https://ieeexplore.ieee.org/document/10151265/>
  18. Bhanuprakash C, Nijagunarya YS, Jayaram MA. A Simple Approach to SQL Joins in a Relational Algebraic Notation. Int J Comput Appl [Internet]. 18 Oktober 2014 [dikutip 21 Januari 2026];104(4):18–26. Tersedia pada: <http://research.ijcaonline.org/volume104/number4/pxc3899099.pdf>

