

**LAPORAN TUGAS UAS**  
**PEMROGRAMAN BASIS DATA**



**UNIVERSITAS  
DUTA BANGSA  
SURAKARTA**

Dosen pengampu: Ridwan Dwi Irawan, S.Kom, M.Kom.

Disusun oleh:

1. Afif abdilah M.S (240103152)
2. Maulana Dhigjay F.A (240103166)
3. Shandy Satria W.P (240103178)

**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS DUTA BANGSA SURAKARTA**  
**2025/2026**

## **Kata Pengantar**

Puji syukur ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tugas Ujian Akhir Semester (UAS) mata kuliah Pemrograman Basis Data dengan menganalisis Studi Kasus Nota Transaksi Penjualan Toko Fotocopy, hingga perumusan struktur basis data final

Laporan ini disusun sebagai salah satu bentuk penerapan teori dan praktik dalam merancang sistem basis data yang efisien dan terintegrasi. Studi kasus yang diangkat berfokus pada proses pengolahan data transaksi penjualan pada sebuah toko fotocopy dan print, di mana pengelolaan data yang baik sangat diperlukan untuk mendukung kegiatan operasional, pelaporan, serta pengambilan keputusan.

Dalam laporan ini, penulis melakukan analisis terhadap data transaksi, kemudian menerapkan normalisasi data untuk mencapai efisiensi penyimpanan serta menjaga integritas data. Selanjutnya dilakukan penentuan derajat kardinalitas guna menggambarkan hubungan antar entitas secara tepat, serta perumusan relasi antar tabel sebagai dasar pembentukan struktur basis data yang terorganisir. Sebagai tahap akhir, hasil perancangan divisualisasikan dalam bentuk Entity Relationship Diagram (ERD) untuk memberikan gambaran menyeluruh tentang rancangan sistem basis data toko fotocopy dan print tersebut. Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah Pemrograman Basis Data serta semua pihak yang telah memberikan dukungan dan bimbingan dalam proses penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi dalam memahami penerapan konsep basis data pada sistem transaksi penjualan.

Hormat Kami,

Tim Penyusun

## DAFTAR ISI:

<b>Kata Pengantar .....</b>	<b>2</b>
<b>BAB I .....</b>	<b>5</b>
<b>PENDAHULUAN .....</b>	<b>5</b>
1.1 Latar Belakang.....	5
1.2 Tujuan laporan .....	5
1.3 Batasan / Ruang Lingkup.....	6
1.4 Pengantar Pemrograman Basis Data .....	6
1.5 Gambaran Umum Sistem.....	7
1.6 Pembagian Tugas .....	7
<b>BAB II.....</b>	<b>8</b>
<b>LANDASAN TEORI .....</b>	<b>8</b>
2.1 Basis Data dan DBMS .....	8
2.3 Tabel dan Relasi .....	8
2.3 Normalisasi Basis data.....	9
2.4 Entity Relationship Diagram (ERD) .....	9
2.5 Structured Query Language (SQL).....	10
2.7 Primary Key dan Foreign Key pada Sistem .....	11
2.8 Penerapan ERD ke DBMS dengan MySQL Workbench .....	11
2.9 Penegasan Materi BAB II .....	12
<b>BAB III .....</b>	<b>13</b>
<b>PERANCANGAN DAN IMPLEMENTASI BASIS DATA .....</b>	<b>13</b>
3.1 Studi Kasus / Nota Fotokopi .....	13
3.2 Tujuan Sistem .....	13
3.3 Kebutuhan Sistem.....	14
3.4 Data Awal dan Proses Normalisasi .....	14
3.5 Hubungan Antar Entitas .....	21
3.6 Derajat kardinalitas.....	22
3.7 Relasi Antar Tabel.....	22

<b>3.8 ERD (Entity Relation Diagram) .....</b>	<b>24</b>
<b>3.9 Penerapan ERD ke DBMS (MySQL Workbench) .....</b>	<b>26</b>
<b>3.10 Implementasi DDL (Data Definition Language) .....</b>	<b>27</b>
<b>BAB IV.....</b>	<b>36</b>
<b>PENUTUP .....</b>	<b>36</b>
<b>    4.1 Hasil Pengujian Query .....</b>	<b>36</b>
<b>    4.2 Ringkasan Hasil Uji .....</b>	<b>37</b>
<b>    4.3 Kendala dan Perbaikan .....</b>	<b>37</b>
<b>    4.4 Kesimpulan.....</b>	<b>38</b>
<b>    4.5 Saran Pengembangan .....</b>	<b>38</b>
<b>    4.6 Lampiran .....</b>	<b>39</b>
<b>DAFTAR PUSTAKA .....</b>	<b>48</b>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Laporan proyek ini dibuat sebagai pemenuhan tugas **Ujian Akhir Semester (UAS)** pada mata kuliah Pemrograman Basis Data. Penyusunan laporan ini berawal dari kebutuhan untuk memahami dan menerapkan cara merancang basis data yang rapi, terstruktur, serta mudah digunakan, dimulai dari data mentah yang masih sederhana. Studi kasus yang digunakan adalah sistem transaksi pada usaha Fotocopy & Print “**Focus**”. Data transaksi yang ada masih disimpan secara sederhana dan belum tersusun dengan baik. Akibatnya, data sering berulang, kurang rapi, dan berpotensi menimbulkan kesalahan saat pencatatan maupun saat membuat laporan. Kondisi tersebut dapat menghambat proses pengelolaan data dan menyulitkan pemilik usaha dalam memantau transaksi.

Untuk mengatasi hal tersebut, dilakukan beberapa tahapan, mulai dari menganalisis data transaksi, melakukan normalisasi hingga mencapai bentuk 1NF, 2NF, dan 3NF, kemudian merancang model basis data dalam bentuk Entity Relationship Diagram (ERD). Tahapan ini bertujuan mengubah data yang awalnya tidak teratur menjadi struktur basis data yang lebih rapi, konsisten, dan mudah dikelola. Dengan perancangan yang baik, sistem diharapkan dapat membantu proses pencatatan transaksi menjadi lebih akurat dan efisien.

### 1.2 Tujuan laporan

Laporan ini disusun untuk mengetahui permasalahan yang terdapat pada data transaksi mentah di usaha Fotocopy & Print “Focus”, terutama yang berkaitan dengan pengulangan data dan struktur yang belum tertata dengan baik. Selain itu, laporan ini bertujuan untuk menerapkan proses normalisasi basis data hingga bentuk 1NF, 2NF, dan 3NF agar data tersusun lebih rapi dan terhindar dari kesalahan saat penyimpanan maupun pengolahan.

Tujuan lainnya adalah merancang model relasional dan Entity Relationship Diagram (ERD) sebagai gambaran visual dari struktur basis data yang dibangun. Secara keseluruhan, laporan ini diharapkan dapat mendokumentasikan proses perancangan basis data dengan baik serta menghasilkan rancangan yang mampu mendukung pengelolaan transaksi secara lebih teratur dan mudah dipahami.

### **1.3 Batasan / Ruang Lingkup**

Ruang lingkup proyek ini dibatasi pada perancangan dan penerapan basis data untuk sistem transaksi pada usaha Fotocopy & Print “Focus”. Pembahasan difokuskan pada pengelolaan data pelanggan, kasir, layanan atau barang, serta data transaksi penjualan. Proyek ini meliputi proses analisis data mentah, normalisasi hingga bentuk normal ketiga (3NF), perancangan struktur tabel, penentuan relasi antar tabel, pembuatan ERD, serta implementasi basis data menggunakan DBMS dan perintah SQL.

Pengembangan aplikasi secara menyeluruh, seperti pembuatan tampilan antarmuka berbasis web atau mobile, sistem login pengguna, pengelolaan stok secara detail, serta fitur pembayaran digital, tidak dibahas dalam proyek ini. Fokus utama laporan adalah pada perancangan dan pengelolaan basis data sebagai dasar sistem informasi transaksi

### **1.4 Pengantar Pemrograman Basis Data**

Pemrograman basis data merupakan proses perancangan dan pengelolaan data menggunakan sistem basis data agar informasi dapat disimpan secara rapi, konsisten, dan mudah diakses. Dalam sebuah sistem informasi, basis data berperan sebagai pusat penyimpanan seluruh data penting. Dengan bantuan DBMS dan bahasa SQL, data dapat dikelola melalui proses pembuatan tabel, pengisian data, pembaruan, serta penarikan informasi dalam bentuk laporan.

Penerapan pemrograman basis data yang baik akan meminimalkan kesalahan data, menghindari pengulangan informasi, dan memudahkan proses pengolahan data. Pada usaha Fotocopy & Print “Focus”, pemrograman basis data digunakan untuk mengelola data pelanggan, kasir, layanan, dan transaksi penjualan agar tersusun lebih teratur dan mudah dikelola.

## **1.5 Gambaran Umum Sistem**

Sistem yang dibangun dalam proyek ini merupakan sistem basis data transaksi untuk usaha Fotocopy & Print “Focus”. Sistem ini dirancang untuk mencatat dan mengelola data pelanggan, kasir, layanan, serta transaksi penjualan secara terintegrasi. Setiap transaksi akan menyimpan informasi mengenai pelanggan yang melakukan transaksi, kasir yang melayani, jenis layanan yang digunakan, jumlah layanan, serta nilai pembayaran. Seluruh data tersebut disimpan dalam basis data yang saling terhubung melalui relasi antar tabel.

Dengan adanya sistem ini, proses pencatatan transaksi menjadi lebih rapi, mudah ditelusuri, dan dapat digunakan sebagai dasar dalam pembuatan laporan serta pengambilan keputusan.

## **1.6 Pembagian Tugas**

1. Afif Abdilah M.S (240103152)

Bertugas mencari studi kasus, membuat poster, membuat Normalisasi 1NF - 3NF, membuat ERD

2. Maulana Dhigjay F.A (240103166)

Bertugas membuat laporan projek, membuat repository di GitHub, membuat database serta kode sql

3. Shandy Satria W.P (240103178)

Bertugas membuat Hubungan Antar Entitas, Derajat Kardinalitas, serta Relasi Antar Tabel

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Basis Data dan DBMS**

Basis data adalah 8ambaran data yang tersimpan secara terstruktur di dalam sistem komputer sehingga mudah diakses, dikelola, dan diperbarui. Dalam sebuah usaha seperti fotocopy dan print, basis data sangat penting untuk menyimpan informasi pelanggan, kasir, layanan, serta Riwayat transaksi. Untuk mengelola basis data diperlukan sebuah perangkat lunak yang disebut DBMS (Database Management System). DBMS berfungsi sebagai pengelola data agar dapat:

- a. menyimpan data dengan rapi,
- b. menjaga keamanan dan keutuhan data,
- c. serta mempermudah proses pencarian dan pengelolaan data.

Contoh DBMS yang sering digunakan adalah MySql, PostgreSQL, dan SQL Server. Dalam projek ini, DBMS digunakan untuk menyimpan dan mengelola data transaksi toko fotocopy secara terpusat dan terstruktur.

#### **2.3 Tabel dan Relasi**

Dalam basis data relasional, data disimpan dalam bentuk tabel. Setiap tabel terdiri dari:

1. Field (kolom) yang mewakili atribut,
2. Record (baris) yang mewakili satu data.

Pada sistem fotocopy ini, contoh tabel yang digunakan adalah:

- a. Tabel Pelanggan
- b. Tabel Kasir
- c. Tabel barang/Layanan
- d. Tabel Transaksi

Agar data antar tabel dapat terhubung, digunakan relasi, relasi dibentuk menggunakan Primary Key (PK) sebagai identitas unik pada suatu tabel dan juga Foreign Key (FK) yaitu sebagai

penghubung ke tabel lain. Dengan adanya relasi, data menjadi lebih terorganisir dan tidak terjadi pengulangan yang berlebihan.

### 2.3 Normalisasi Basis data

Normalisasi adalah proses penyusunan struktur tabel agar data tersimpan dengan efisien dan tidak terjadi duplikasi. Tujuan utama normalisasi adalah mengurangi redundansi data, mencegah kesalahan saat insert, update, dan delete serta menjadikan struktur basis data lebih rapi dan konsisten. Tahapan normalisasi yang digunakan dalam projek ini adalah:

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

Pada tahap ini, setiap kolom hanya berisi satu nilai dan tidak ada data yang bersifat ganda dalam satu sel.

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

Pada tahap ini, data yang tidak bergantung sepenuhnya pada primary key dipisahkan ke tabel lain. Tujuannya agar setiap atribut benar-benar menggambarkan entitasnya.

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

Pada tahap ini, ketergantungan tidak langsung antar atribut dihilangkan. Setiap kolom non-kunci hanya bergantung pada primary key, bukan pada kolom lain.

Dengan normalisasi sampai tahap 3NF, struktur basis data menjadi lebih stabil dan mudah dikelola.

### 2.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang digunakan untuk menggambarkan

- a. Entitas (tabel),
- b. Artibut (field),
- c. Hubungan antar entitas.

ERD membantu dalam memahami alur data dan hubungan antar tabel sebelum sistem diimplementasikan ke dalam DBMS. Pada sistem fotocopy ini, ERD menggambarkan hubungan antara Pelangan, Kasir, Barang/Layanan, dan Transaksi, ERD menjadi blueprint utama dalam pembuatan struktur basis data.

## 2.5 Structured Query Language (SQL)

SQL adalah bahasa yang digunakan untuk berinteraksi dengan basis data, SQL digunakan untuk membuat tabel, mengisi data, serta mengambil informasi dari database. SQL dibagi menjadi beberapa bagian utama:

### 1. DDL (Data Definition Language)

Digunakan untuk membuat dan mengubah struktur database, seperti:

- a. CREATE DATABASE
- b. CREATE TABLE
- c. ALTER TABLE

### 2. DML (Data Manipulation Language)

Digunakan untuk mengelola isi data, seperti:

- a. INSERT
- b. UPDATE
- c. DELETE

### 3. TCL (Transaction Control Language)

Digunakan untuk mengatur transaksi agar data tetap aman, seperti:

- a. COMMIT
- b. ROLLBACK

Selain itu, SQL juga digunakan untuk melakukan:

- a. JOIN : Menggabungkan data dari beberapa tabel
- b. GROUP BY & Agregasi : mengelompokkan dan menghitung data
- c. HAVING : menyaring hasil agregasi
- d. Subquery : query di dalam query

## 2.7 Primary Key dan Foreign Key pada Sistem

Dalam sistem basis data ini, setiap tabel memiliki Primary Key (PK) sebagai identitas unik, serta Foreign Key (FK) sebagai penghubung antar tabel. Penggunaan PK dan FK bertujuan untuk menjaga keunikan data dan membentuk relasi yang konsisten antar tabel.

Struktur kunci pada setiap tabel adalah sebagai berikut:

- 1) Tabel Pelanggan
  - a. Primary Key: id\_pelanggan : Digunakan untuk membedakan setiap pelanggan secara unik..
- 2) Tabel Kasir
  - b. Primary Key: id\_kasir : Digunakan sebagai identitas unik setiap kasir.
- 3) Tabel Layanan/Barang
  - c. Primary Key: id\_barang : Digunakan untuk membedakan setiap jenis layanan atau barang.
- 4) Tabel Transaksi
  - d. Primary Key: id\_transaksi
  - e. Foreign key:
    - a) id\_pelanggan : mengacu ke tabel Pelanggan,
    - b) id\_kasir : mengacu ke tabel Kasir
    - c) id\_barang : mengacu ke tabel barang/Layanan

Dengan struktur ini, setiap transaksi dapat diketahui siapa pelanggannya, siapa kasir yang melayani, serta layanan apa yang digunakan. Relasi antar tabel menjadi lebih terjaga dan data tersimpan secara konsisten.

## 2.8 Penerapan ERD ke DBMS dengan MySQL Workbench

Pada proyek ini, penerapan ERD ke dalam sistem basis data dilakukan menggunakan **MySQL Workbench**. ERD yang telah dirancang sebelumnya diimplementasikan menjadi struktur tabel pada DBMS melalui fitur *forward engineering*. Proses ini memungkinkan rancangan visual ERD diubah secara langsung menjadi perintah SQL untuk membentuk database dan tabel-tabel yang saling berelasi. Dengan cara ini, struktur basis data yang dihasilkan sesuai dengan rancangan awal, baik dari sisi tabel, atribut, maupun hubungan antar tabel.

Penggunaan MySQL Workbench membantu mempercepat proses pembuatan database, meminimalkan kesalahan, serta memastikan bahwa relasi antar tabel terbentuk sesuai dengan ERD yang telah dibuat.

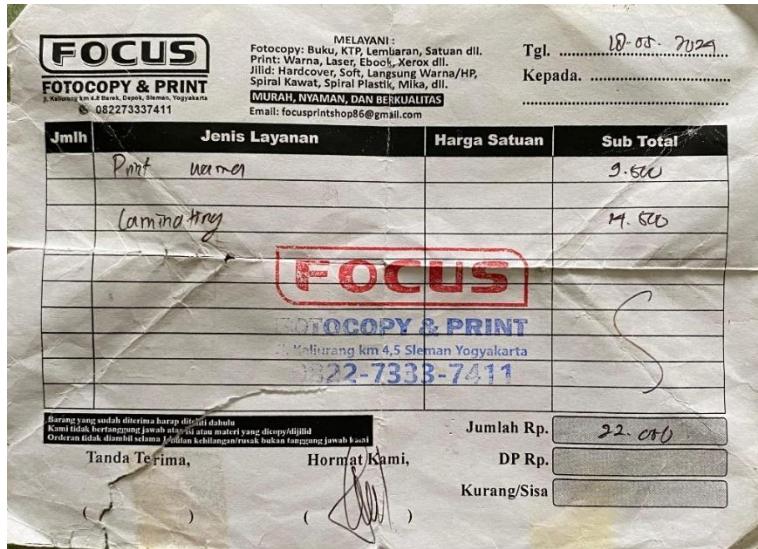
## **2.9 Penegasan Materi BAB II**

Seluruh isi pada bab ini disusun dengan mengacu pada materi yang dibahas dalam Jurnal Kuliah Pemrograman Basis Data. Pembahasan dimulai dari pengantar basis data, relasi antar tabel, penyusunan ERD, hingga proses normalisasi dari 1NF sampai 3NF. Selain itu, bab ini juga memuat konsep relasi basis data, penerapan ERD ke dalam DBMS menggunakan MySQL Workbench, serta penggunaan SQL yang mencakup DDL, DML, TCL, agregasi dan HAVING, GROUP BY, serta JOIN. Dengan adanya pembahasan tersebut, BAB II telah tersusun secara lengkap dan sesuai dengan ketentuan yang ditetapkan oleh dosen.

## BAB III

### PERANCANGAN DAN IMPLEMENTASI BASIS DATA

#### 3.1 Studi Kasus / Nota Fotokopi



Sistem ini dirancang untuk mengelola transaksi pada sebuah Fotocopy & Print yang Bernama “Focus” mencakup pencatatan data pembelian, jumlah yang dibeli/layanan, harga satuan, serta subtotal dan jumlah keseluruhan.

#### 3.2 Tujuan Sistem

- Mencatat setiap transaksi layanan fotokopi dan print secara lengkap, termasuk jenis layanan, harga satuan, subtotal, total bayar, uang muka (DP), dan sisa pembayaran.
- Mengelola informasi dasar layanan yang tersedia, seperti nama layanan dan harga satuan.
- Menyimpan data dasar pelanggan dan kasir.
- Menyediakan data dasar untuk laporan transaksi harian/bulanan.

### 3.3 Kebutuhan Sistem

Sistem yang dibangun bertujuan untuk mengelola data transaksi pada usaha Fotocopy & Print “Focus”. Sistem ini harus mampu menyimpan data pelanggan, kasir, layanan atau barang, serta transaksi penjualan secara terstruktur.

Selain itu, sistem harus dapat mencatat setiap transaksi yang terjadi, menampilkan kembali data transaksi, serta menjaga keterkaitan antar data agar informasi yang tersimpan tetap rapi dan konsisten. Dengan adanya sistem ini, proses pencatatan dan pengelolaan transaksi diharapkan menjadi lebih mudah, akurat, dan terorganisir.

### 3.4 Data Awal dan Proses Normalisasi

#### 3.4.1 Tabel Umum

JUMLAH_BARANG	JENIS_LAYANAN	HARGA_SATUAN	NAMA_KASIR	SUB_TOTAL	TANGGAL	NAMA_PELANGGAN	TOTAL_BAYAR	DP	KURANG/SISA
1	Print Warna	Rp 9,500.00	Albert	Rp9,500.00	18/05/2025	Daniel	Rp 22,000.00	0	Rp0.00
1	Laminating	Rp 14,500.00		Rp14,500.00					

#### 3.4.2 Tabel 1NF (First Normal Form)

JUMLAH_BARANG	JENIS_LAYANAN	HARGA_SATUAN	NAMA_KASIR	SUB_TOTAL
1	Print Warna	Rp 9,500.00	Albert	Rp9,500.00
1	Laminating	Rp 14,500.00	Albert	Rp14,500.00

TANGGAL	NAMA_PELANGGAN	TOTAL_BAYAR	DP	KURANG/SISA
18/05/2025	Daniel	Rp 22,000.00	0	Rp0.00
18/05/2025	Daniel	Rp 22,000.01	0	Rp0.00

### Fungsi & Tujuan:

Tahap 1NF berfungsi untuk menyederhanakan struktur data mentah dengan menerapkan dua aturan dasar: pertama, setiap sel hanya berisi satu nilai atomik (tidak boleh ada multi-value); kedua, menghilangkan kelompok data yang berulang. Tujuan utama normalisasi 1NF adalah menciptakan fondasi database yang terstruktur dan konsisten, di mana data dapat diidentifikasi dan diakses secara individual.

### Penjelasan Tabel:

- Tabel Item Transaksi:** Tabel ini berfungsi untuk mendaftarkan setiap item atau layanan yang dibeli dalam satu transaksi. Isinya mencakup Jumlah Barang, Jenis Layanan, Harga Satuan, Nama Kasir, dan Sub Total. Tujuannya adalah memecah transaksi yang kompleks menjadi unit-unit terkecil sehingga memudahkan pelacakan detail penjualan.
- Tabel Data Transaksi:** Tabel ini berfungsi sebagai ringkasan atau header dari sebuah transaksi. Isinya meliputi Tanggal, Nama Pelanggan, Total Bayar, DP, dan Kurang/Sisa. Tujuannya adalah memberikan gambaran menyeluruh tentang satu kejadian transaksi.

### Manfaat:

Dengan menerapkan 1NF, data yang sebelumnya berantakan dan mungkin digabung dalam satu sel (seperti beberapa layanan dalam satu baris) kini tersusun rapi. Ini memudahkan proses query dasar dan menghilangkan ambiguitas dalam penyimpanan data.

### 3.4.3 Tabel 2NF (Second Normal Form)

#### 1. Tabel Transaksi:

ID_TRANSAKSI	JUMLAH_BARANG	SUB_TOTAL	TANGGAL	TOTAL_BAYAR	DP	KURANG / SISA
120	1	Rp9,500.00	18/05/2025	Rp 22,000.00	0	Rp0.00
121	1	Rp14,500.00	18/05/2025	Rp 22,000.01	0	Rp0.00

#### 2. Tabel Barang:

ID_BARANG	JENIS_LAYANAN	HARGA_SATUAN
76348001	Print Warna	Rp 9,500.00
23591002	Laminating	Rp 14,500.00

#### 3. Tabel Pelanggan:

ID_PELANGGAN	NAMA_PELANGGAN
98765456	Daniel
98765456	Daniel

#### 4. Tabel Kasir

ID_KASIR	NAMA_KASIR
k0001	Albert
k0001	Albert

#### Fungsi & Tujuan:

Fungsi tahap 2NF adalah menghilangkan *partial dependency*, yaitu kondisi dimana suatu atribut non-kunci hanya bergantung pada sebagian primary key (biasanya pada key komposit). Tujuannya adalah memastikan bahwa setiap atribut dalam sebuah tabel benar-benar hanya menggambarkan entitas utamanya, sehingga meningkatkan integritas data dan menghindari redundansi.

#### Penjelasan Tabel:

##### a. Tabel Transaksi:

Fungsinya sebagai pusat rekaman inti setiap transaksi. ID\_Transaksi sebagai primary key-nya memastikan setiap transaksi unik. Tujuannya adalah menyimpan informasi yang melekat pada satu transaksi secara keseluruhan, seperti Tanggal dan Total\_Bayar.

##### b. Tabel Detail\_Transaksi:

Fungsinya adalah sebagai *junction table* yang menghubungkan transaksi dengan barang yang dibeli. Tujuannya adalah mencatat Jumlah\_Barang dan Sub\_Total untuk setiap item dalam suatu transaksi, sehingga satu transaksi bisa terdiri dari banyak barang.

##### c. Tabel Barang/Jasa, Pelanggan, dan Kasir:

Fungsinya sebagai tabel master atau referensi. Tujuannya adalah menyimpan data tetap tentang layanan, pelanggan, dan karyawan di satu tempat terpusat, sehingga menghindari penulisan berulang dan memastikan konsistensi.

## **Manfaat:**

Pemecahan tabel ini menghilangkan redudansi data yang signifikan. Misalnya, nama pelanggan dan harga layanan tidak perlu ditulis berulang kali untuk setiap transaksi yang sama, sehingga menghemat ruang penyimpanan dan mencegah ketidakkonsistensi data (*update anomalies*).

### **3.4.4 Tabel 3NF (Third Normal Form)**

1. Kamus Basis Data
2. Nama\_Database: Fotocopy
3. Nama\_Tabel: Tabel\_Transaksi
4. Fungsi: Menyimpan data Transaksi

Nama field	Tipe	Panjang Karakter	Keterangan
id_tarnsaksi	integer	11	Primary key
id_pelanggan	integer	11	Foreign Key
id_kasir	varchar	10	Foreign Key
id_barang	integer	11	Foreign Key
jumlah_barang	integer	11	jumlah layanan
tanggal	Date	11	tanggal transaksi
dp	integer	11	uang muka
total_bayar	Integer	11	total pembayaran

sub_total	integer	11	harga x jumlah
kurang_sisa	integer	11	sisa pembayaran

1. Kamus Basis Data
2. Nama\_Database: Fotocopy
3. Nama\_Tabel: Tabel\_Barang/Layanan
4. Fungsi: Menyimpan data Barang/Layanan

Nama Field	Tipe	Panjang Karakter	Keterangan
id_barang	integer	11	Primary key
jenis_layanan	varchar	50	Nama layanan
harga_satuan	integer	11	Harga per layanan

1. Kamus Basis Data
2. Nama\_Database: Fotocopy
3. Nama\_Tabel: Tabel\_Pelanggan
4. Fungsi: Menyimpan data Pelanggan

Nama Field	Tipe	Panjang Karakter	Keterangan
id_pelanggan	integer	1	Primary key
nama_pelanggan	varchar	100	Nama Pelanggan

1. Kamus Basis Data
2. Nama\_Database: Fotocopy
3. Nama\_Tabel: Tabel\_Kasir
4. Fungsi: Menyimpan data Kasir

Nama Field	Tipe	Panjang Karakter	Keterangan
id_kasir	integer	10	Primary Key
nama_kasir	varchar	100	Nama Kasir

### Fungsi & Tujuan:

Fungsi 3NF adalah menghilangkan *transitive dependency*, yaitu ketergantungan tidak langsung di mana suatu atribut non-kunci bergantung pada atribut non-kunci lainnya, bukan langsung pada primary key. Tujuan akhirnya adalah menciptakan struktur database yang sangat efisien, logis, dan bebas dari anomali data baik saat penyisipan, pembaruan, maupun penghapusan.

### Penjelasan:

Pada tahap ini, struktur tabel dari 2NF sudah umumnya memenuhi syarat 3NF. Yang dilakukan adalah penyempurnaan akhir dan pendefinisian kamus data yang tepat. Tujuannya adalah memastikan bahwa dalam sebuah tabel, hanya ada ketergantungan fungsional langsung pada primary key. Contohnya, dalam tabel Transaksi, Total\_Bayar bergantung langsung pada ID\_Transaksi, bukan pada Sub\_Total dari tabel detail.

## **Manfaat:**

Database yang telah dinormalisasi hingga 3NF menjadi sangat stabil dan mudah dikelola (*Maintain*). Perubahan pada satu data (misalnya, mengganti harga suatu layanan) hanya perlu dilakukan di satu tempat (tabel BARANG), dan perubahan itu akan secara akurat tercermin di seluruh laporan dan history transaksi tanpa harus mengubah data masa lalu.

### **3.5 Hubungan Antar Entitas**

Model basis data 3NF ini memiliki 4 entitas utama, yaitu: Tabel Pelanggan, Tabel Kasir, Tabel Layanan, dan Tabel Transaksi (Header & Detail).

Entitas 1	Entitas 2	Hubungan	Keterangan
Pelanggan	Transaksi (Header)	1 : N	Satu pelanggan dapat memiliki banyak transaksi.
Kasir	Transaksi (Header)	1 : N	Satu kasir dapat melayani banyak transaksi.
Transaksi (Header)	Transaksi (Detail)	1 : N	Satu transaksi utama memiliki banyak detail layanan.
Barang/Layanan	Transaksi (Detail)	N : N	Satu layanan dapat muncul di banyak transaksi.

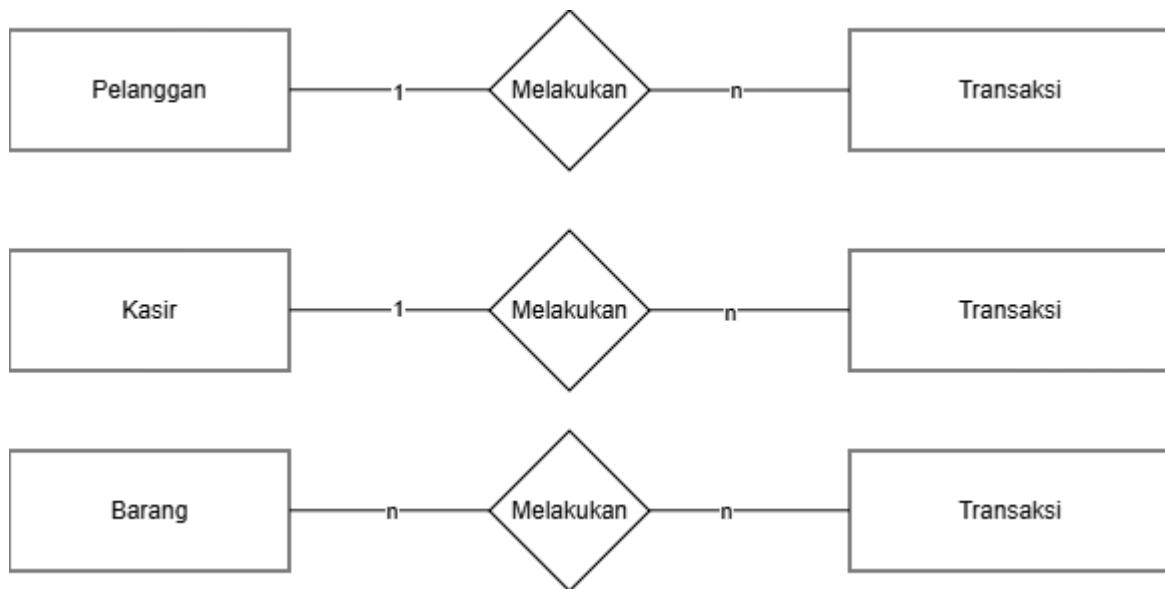
### **Hubungan Tidak Langsung (M:N)**

Hubungan many-to-many antara pelanggan dan layanan dipecah menjadi dua hubungan 1:N melalui Tabel Transaksi Detail, yaitu:

- a. Pelanggan → Transaksi Header (1:N)
- b. Transaksi Header → Transaksi Detail (1:N)
- c. Layanan → Transaksi Detail (1:N)

Struktur ini menjamin integritas data, mengurangi redundansi, dan memudahkan pembuatan laporan transaksi.

### 3.6 Derajat kardinalitas



Deskripsi :

1. Seorang pelanggan dapat melakukan banyak transaksi, tetapi banyak pelanggan tidak boleh memiliki satu transaksi,
2. Seorang kasir boleh melakukan banyak transaksi, satu transaksi tidak boleh dilakukan oleh banyak kasir.
3. Barang dan transaksi bisa dilakukan banyak kali.

### 3.7 Relasi Antar Tabel

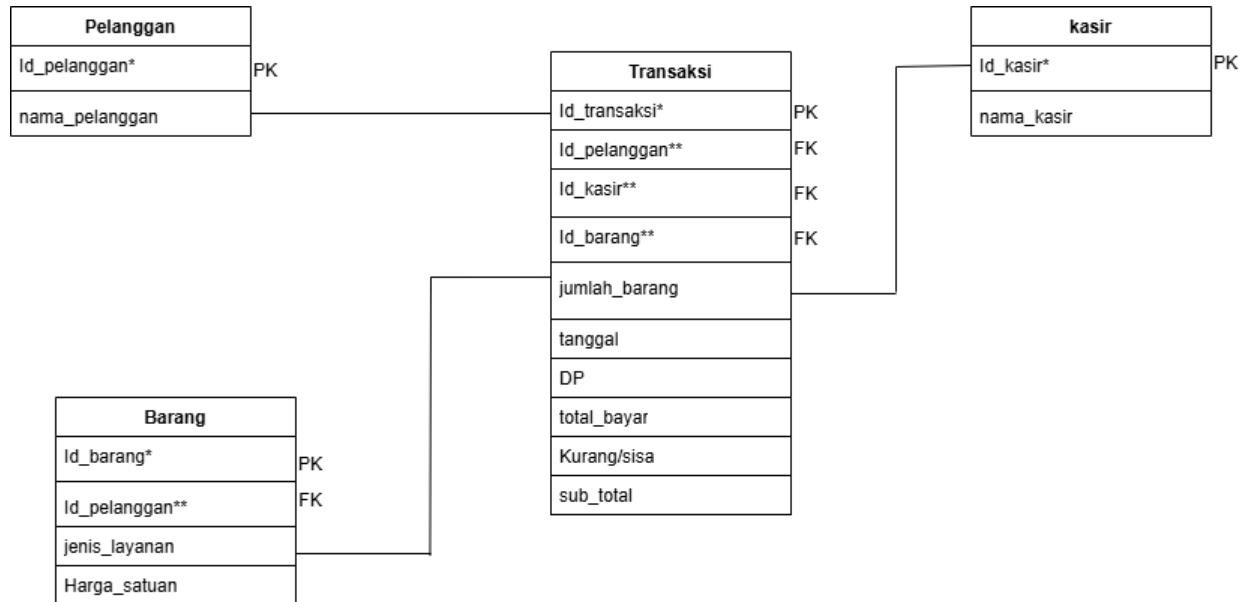
Relasi antar tabel pada sistem Fotocopy & Print “Focus” dirancang untuk menghubungkan data pelanggan, kasir, layanan, dan transaksi agar tersimpan secara terstruktur dan saling berkaitan. Setiap tabel memiliki peran yang berbeda, namun saling terhubung melalui primary key (PK) dan foreign key (FK).

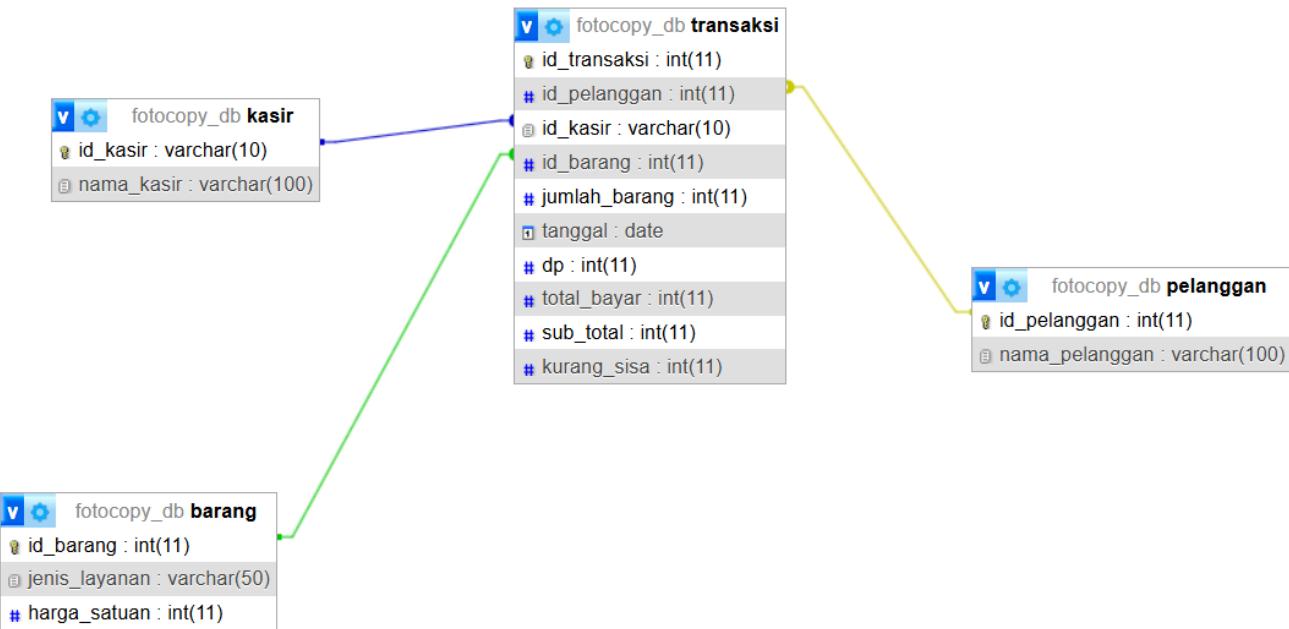
Tabel **pelanggan** terhubung dengan tabel **transaksi** melalui id\_pelanggan. Relasi ini menunjukkan bahwa satu pelanggan dapat melakukan banyak transaksi (1:N), sedangkan setiap transaksi hanya dimiliki oleh satu pelanggan.

Tabel **kasir** juga terhubung dengan tabel **transaksi** melalui id\_kasir. Relasi ini menunjukkan bahwa satu kasir dapat melayani banyak transaksi, namun satu transaksi hanya dilayani oleh satu kasir.

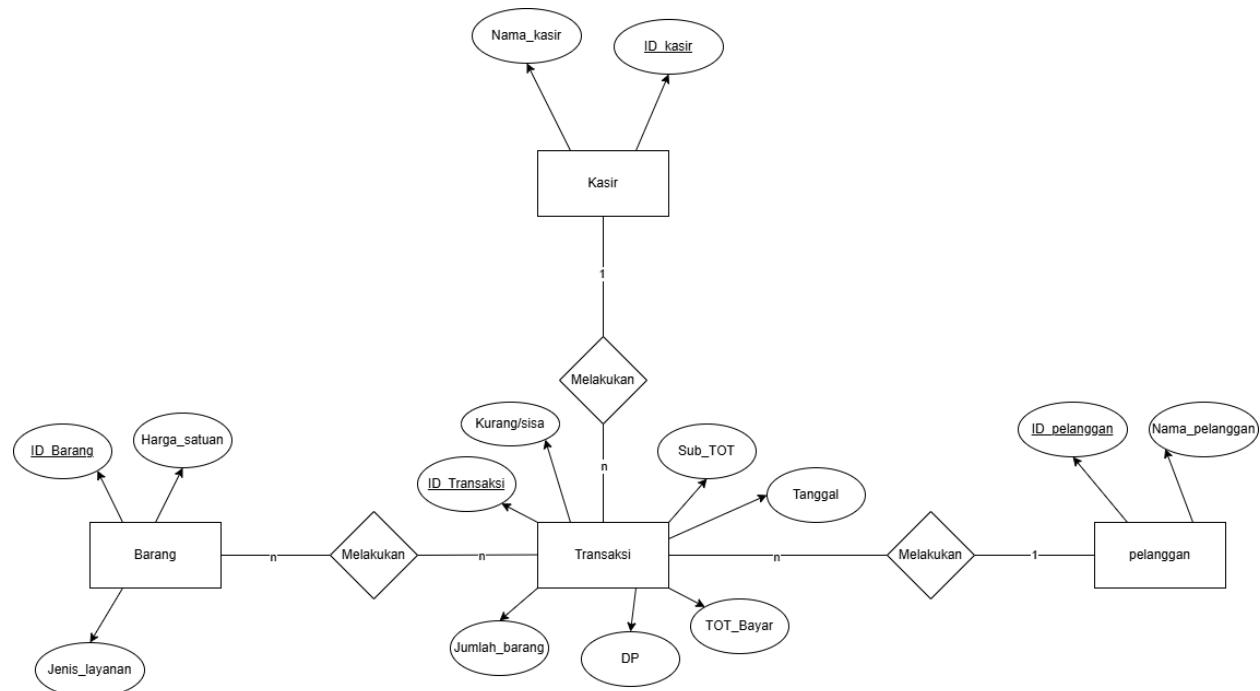
Tabel **barang/layanan** terhubung dengan tabel **transaksi** melalui id\_barang. Relasi ini menunjukkan bahwa satu jenis layanan dapat muncul pada banyak transaksi, sedangkan setiap transaksi mencatat layanan yang digunakan. Dengan adanya relasi tersebut, setiap data transaksi dapat diketahui secara jelas siapa pelanggannya, siapa kasir yang melayani, serta layanan apa yang digunakan. Relasi ini menjaga konsistensi data, menghindari pengulangan informasi, dan memudahkan proses pengolahan serta pelaporan transaksi.

Untuk desain seperti dibawah ini:





### 3.8 ERD (Entity Relation Diagram)



Entity Relationship Diagram (ERD) di atas menjelaskan rancangan hubungan antar tabel pada sistem transaksi fotokopi dan print di toko *Focus*. Sistem ini berfungsi untuk mencatat setiap transaksi layanan fotokopi, print, atau laminating secara terintegrasi, mencakup data pelanggan, kasir, barang/layanan, serta rincian transaksi seperti subtotal, total bayar, dan sisa pembayaran.

### **3.8.1 Entitas dan atribut utama**

#### **1. Entitas Kasir**

Entitas ini menyimpan informasi tentang pegawai yang melayani transaksi.

##### **a. Atribut:**

- 1) ID\_Kasir → Primary Key, kode unik untuk setiap kasir.
- 2) Nama\_Kasir → nama kasir yang bertugas.

##### **b. Relasi:**

- 1) Satu kasir dapat menangani banyak transaksi (1 : N).

#### **2. Entitas Pelanggan**

Entitas ini berisi data pelanggan yang melakukan transaksi di toko.

##### **a. Atribut:**

- 1) ID\_Pelanggan → Primary Key, kode unik setiap pelanggan.
- 2) Nama\_Pelanggan → nama pelanggan yang terdaftar.

##### **b. Relasi**

- 1) Satu pelanggan dapat melakukan banyak transaksi (1 : N).

#### **3. Entitas Barang / Layanan**

Entitas ini merepresentasikan jenis layanan atau barang yang disediakan oleh toko.

##### **a. Atribut:**

- 1) ID\_Barang → Primary Key, kode unik untuk tiap jenis layanan.

- 2) Jenis\_Layanan → nama layanan seperti “Print Warna”, “Laminating”, dll.
- 3) Harga\_Satuan → tarif untuk tiap layanan.

b. Relasi:

- 1) Satu jenis layanan/barang dapat muncul di banyak transaksi (1 : N).

#### 4. Entitas Transaksi

Entitas ini merupakan inti sistem yang mencatat setiap transaksi antara pelanggan, kasir, dan layanan.

a. Atribut:

- 1) ID\_Transaksi → Primary Key, nomor unik setiap transaksi.
- 2) Tanggal → waktu transaksi dilakukan.
- 3) Jumlah\_Barang → jumlah layanan/barang yang digunakan.
- 4) Sub\_Total → hasil perhitungan harga satuan × jumlah barang.
- 5) Total\_Bayar → total pembayaran seluruh layanan dalam transaksi.
- 6) DP → jumlah uang muka yang dibayarkan pelanggan.
- 7) Kurang/Sisa → sisa pembayaran jika ada kekurangan.

b. Relasi:

- 1) 1 Kasir → banyak Transaksi (1:N)
- 2) 1 Pelanggan → banyak Transaksi (1:N)
- 3) 1 Barang → banyak Transaksi (1:N)

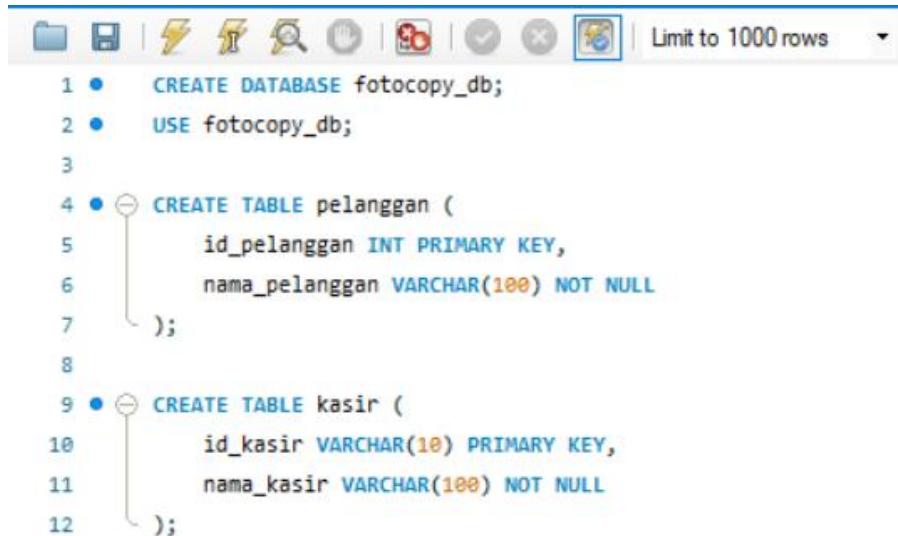
### 3.9 Penerapan ERD ke DBMS (MySQL Workbench)

Penerapan ERD ke dalam DBMS pada proyek ini dilakukan menggunakan MySQL Workbench. ERD yang telah dirancang diimplementasikan melalui fitur forward engineering, sehingga rancangan visual dapat diubah secara otomatis menjadi perintah SQL. Proses ini menghasilkan database beserta tabel-tabel yang saling berelasi sesuai dengan ERD, lengkap dengan penentuan

primary key dan foreign key. Dengan demikian, struktur basis data yang terbentuk pada DBMS telah sesuai dengan rancangan konseptual yang dibuat sebelumnya.

### 3.10 Implementasi DDL (Data Definition Language)

- a. **DDL (Data Definition Language)** digunakan untuk membentuk dan mengatur struktur database. Perintah ini berfungsi membuat database, membuat tabel, serta menentukan aturan seperti primary key dan foreign key. Melalui DDL, struktur basis data dibangun sesuai rancangan yang telah dibuat.



```
1 • CREATE DATABASE photocopy_db;
2 • USE photocopy_db;
3
4 • CREATE TABLE pelanggan (
    id_pelanggan INT PRIMARY KEY,
    nama_pelanggan VARCHAR(100) NOT NULL
);
5
6
7
8
9 • CREATE TABLE kasir (
    id_kasir VARCHAR(10) PRIMARY KEY,
    nama_kasir VARCHAR(100) NOT NULL
);
10
11
12
```

```

13
14 • CREATE TABLE barang (
15     id_barang INT PRIMARY KEY,
16     jenis_layanan VARCHAR(50) NOT NULL,
17     harga_satuan INT NOT NULL
18 );
19
20 • CREATE TABLE transaksi (
21     id_transaksi INT PRIMARY KEY,
22     id_pelanggan INT,
23     id_kasir VARCHAR(10),
24     id_barang INT,
25     jumlah_barang INT,
26     tanggal DATE,
27     dp INT,
28     total_bayar INT,
29     sub_total INT,
30     kurang_sisa INT,
31     CONSTRAINT fk_pelanggan FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan),
32     CONSTRAINT fk_kasir FOREIGN KEY (id_kasir) REFERENCES kasir(id_kasir),
33     CONSTRAINT fk_barang FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
34 );

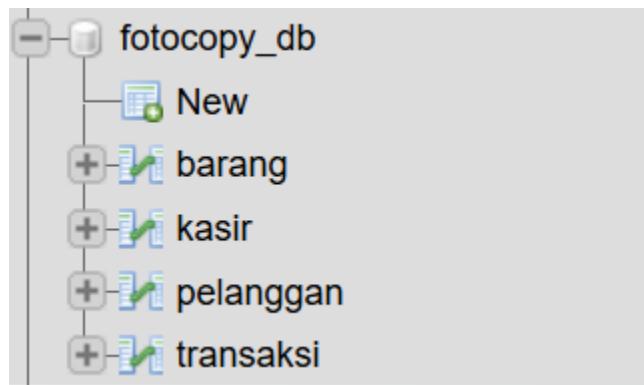
```

Penjelasan :

Ini adalah kode program untuk DDL yang mana untuk membuat sebuah database baru dengan menggunakan kode “CREATE DATABASE (nama database yang ingin dibuat)”; jika sudah maka akan ada seperti notifikasi yang centang hijau di Action Output, dan fyi database itu Cuma bisa dibuat 1 kali setelah dibuat maka jika mau run lagi maka tidak akan bisa, setelah membuat database maka kita gunakan perintah “USE (nama database kamu)”; yang mana perintah tersebut bertujuan untuk menggunakan databasenya.

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

15	14:39:16	CREATE DATABASE fotocopy_db_	1 row(s) affected	0.000 sec
16	14:39:20	USE fotocopy_db_	0 row(s) affected	0.000 sec
17	14:39:22	CREATE TABLE pelanggan ( id_pelanggan INT PRIMARY KEY, nama_pelanggan VARCHAR(100) NOT NULL )	0 row(s) affected	0.062 sec
18	14:39:29	CREATE TABLE kasir ( id_kasir VARCHAR(10) PRIMARY KEY, nama_kasir VARCHAR(100) NOT NULL )	0 row(s) affected	0.031 sec
19	14:39:32	CREATE TABLE barang ( id_barang INT PRIMARY KEY, jenis_layanan VARCHAR(50) NOT NULL, harga_satuan INT NOT NULL )	0 row(s) affected	0.047 sec
20	14:39:35	CREATE TABLE transaksi ( id_transaksi INT PRIMARY KEY, id_pelanggan INT, id_kasir VARCHAR(10), id_barang INT, jumlah_barang INT, tanggal DATE, dp INT, total_bayar INT, sub_total INT, kurang_sisa INT )	0 row(s) affected	0.062 sec



*Ini adalah hasil running dari DDL*

- b. **DML (Data Manipulation Language)** digunakan untuk mengelola isi data di dalam tabel. Perintah ini berfungsi untuk menambahkan, mengubah, menghapus, dan menampilkan data. Dengan DML, tabel yang sudah dibuat dapat diisi dengan data pelanggan, kasir, layanan, dan transaksi.

```
INSERT INTO pelanggan VALUES
(98765456, 'Daniel'),
(98765457, 'Rina'),
(98765458, 'Budi');
```

```
INSERT INTO kasir VALUES
('k0001', 'Albert'),
('k0002', 'Sinta');
```

- ```
INSERT INTO barang VALUES
(76348001, 'Print Warna', 9500),
(23591002, 'Laminating', 14500),
(99112003, 'Fotocopy Hitam Putih', 500),
(88123004, 'Jilid Spiral', 7000);
```
- ```
INSERT INTO transaksi VALUES
(120, 98765456, 'k0001', 76348001, 1, '2025-05-18', 0, 9500, 9500, 0),
(121, 98765456, 'k0001', 23591002, 1, '2025-05-18', 0, 14500, 14500, 0),
(122, 98765457, 'k0002', 99112003, 10, '2025-05-19', 0, 5000, 5000, 0),
(123, 98765458, 'k0001', 88123004, 1, '2025-05-19', 0, 7000, 7000, 0),
(124, 98765457, 'k0002', 76348001, 2, '2025-05-20', 0, 19000, 19000, 0);
```

*Gambar ini merupakan kode program untuk DML*

Penjelasan :

Bagian DML (Data Manipulation Language) digunakan untuk mengisi data awal (*seed data*) ke dalam tabel yang telah dibuat agar database memiliki isi dan dapat diuji fungsinya. Perintah INSERT INTO pelanggan digunakan untuk menambahkan data pelanggan bernama Daniel, INSERT INTO kasir untuk menambahkan data kasir Albert, serta INSERT INTO barang untuk mengisi data layanan seperti Print Warna dan Laminating beserta harga satunya. Selanjutnya, perintah INSERT INTO transaksi digunakan untuk mencatat transaksi yang terjadi, yang memuat informasi pelanggan, kasir, layanan yang digunakan, jumlah, tanggal, serta nilai pembayaran. Perintah SELECT \* FROM transaksi digunakan untuk menampilkan kembali seluruh data transaksi sebagai bukti bahwa data berhasil disimpan dan dapat diakses oleh sistem. Dengan penggunaan DML ini, tabel yang semula kosong menjadi terisi sehingga sistem basis data siap digunakan untuk pengujian dan pengolahan data.

✓	21 15:00:06	INSERT INTO pelanggan VALUES (98765456, 'Daniel')	1 row(s) affected	
✓	21 15:00:06	INSERT INTO pelanggan VALUES (98765456, 'Daniel')	1 row(s) affected	
✓	22 15:00:08	INSERT INTO kasir VALUES ('k0001', 'Albert')	1 row(s) affected	
✓	23 15:00:10	INSERT INTO barang VALUES (76348001, 'Print Warna', 9500), (23591002, 'Laminating', 14500)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 : 0	
✓	24 15:00:12	INSERT INTO transaksi VALUES (120, 98765456, 'k0001', 76348001, 2, '2025-05-18', 0, 22000, 9500, 0), (12...)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 : 0	

- c. **TCL (Transaction Control Language)** digunakan untuk mengatur proses transaksi agar perubahan data berlangsung dengan aman. TCL memastikan bahwa data hanya disimpan jika proses berjalan dengan benar, dan dapat dibatalkan jika terjadi kesalahan, sehingga menjaga konsistensi data.

```
| START TRANSACTION;  
| INSERT INTO transaksi VALUES  
| (125, 98765456, 'k0001', 76348001, 2, '2025-05-21', 0, 19000, 19000, 0);  
| COMMIT;  
-- Gunakan ROLLBACK jika terjadi kesalahan
```

*Gambar ini adalah kode program untuk TCL*

Penjelasan:

Bagian TCL (Transaction Control Language) digunakan untuk mengatur proses transaksi agar perubahan data pada database berlangsung dengan aman dan terkontrol. Pada kode yang digunakan, perintah START TRANSACTION menandai awal sebuah proses transaksi, kemudian dilakukan perintah INSERT INTO transaksi untuk menambahkan data transaksi baru. Setelah proses berjalan

dengan benar, perintah COMMIT digunakan untuk menyimpan perubahan tersebut secara permanen ke dalam database. Apabila terjadi kesalahan pada proses penyimpanan data, perintah ROLLBACK dapat digunakan untuk membatalkan seluruh perubahan yang telah dilakukan selama transaksi berlangsung. Dengan penerapan TCL ini, sistem mampu menjaga konsistensi data dan mencegah terjadinya data yang tidak lengkap atau rusak akibat kegagalan proses.

25	16:05:50	START TRANSACTION	0 row(s) affected
26	16:06:18	COMMIT	0 row(s) affected

#### d. Query

Query adalah perintah yang digunakan untuk mengambil, menampilkan, dan mengolah data yang tersimpan di dalam basis data. Melalui query, pengguna dapat menentukan data apa yang ingin ditampilkan, dari tabel mana data tersebut diambil, serta bagaimana data tersebut diproses. Query memungkinkan data yang tersimpan dalam banyak tabel diolah menjadi informasi yang berguna, seperti daftar transaksi, laporan pendapatan, atau ringkasan data tertentu. Dengan adanya query, basis data tidak hanya berfungsi sebagai tempat penyimpanan, tetapi juga sebagai sumber informasi yang dapat diakses dan dianalisis sesuai kebutuhan.

##### a) Join

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

```
SELECT t.id_transaksi, p.nama_pelanggan, k.nama_kasir,
       b.jenis_layanan, t.jumlah_barang, t.sub_total, t.tanggal
FROM transaksi t
JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan
JOIN kasir k ON t.id_kasir = k.id_kasir
JOIN barang b ON t.id_barang = b.id_barang;
```

*Ini adalah kode query Join*

Penjelasan:

Query ini menampilkan data transaksi secara lengkap dengan menggabungkan tabel transaksi, pelanggan, kasir, dan barang. Hasilnya bukan lagi berupa ID, tetapi informasi yang mudah dibaca seperti nama pelanggan, nama kasir, dan jenis layanan. Query ini digunakan untuk melihat laporan transaksi dalam bentuk yang lebih informatif.

Result Grid							
	id_transaksi	nama_pelanggan	nama_kasir	jenis_layanan	jumlah_barang	sub_total	tanggal
▶	120	Daniel	Albert	Print Warna	1	9500	2025-05-18
	121	Daniel	Albert	Laminating	1	14500	2025-05-18
	122	Rina	Sinta	Fotocopy Hitam Putih	10	5000	2025-05-19
	123	Budi	Albert	Jilid Spiral	1	7000	2025-05-19
	124	Rina	Sinta	Print Warna	2	19000	2025-05-20

*Dan ini adalah hasil query Join setelah di run*

b) Group By & Agregasi

GROUP BY digunakan untuk mengelompokkan data berdasarkan kolom tertentu, sedangkan fungsi agregasi seperti SUM() digunakan untuk melakukan perhitungan. Pada sistem ini, GROUP BY digunakan untuk mengelompokkan transaksi berdasarkan jenis layanan, kemudian dihitung total pendapatan setiap layanan. Hasilnya dapat digunakan sebagai dasar laporan, misalnya untuk mengetahui layanan mana yang paling banyak menghasilkan pendapatan.

```
67 •   SELECT b.jenis_layanan,  
68           SUM(t.sub_total) AS total_pendapatan  
69     FROM transaksi t  
70    JOIN barang b ON t.id_barang = b.id_barang  
71    GROUP BY b.jenis_layanan;
```

*Ini adalah kode query Group By & Agregasi*

Penjelasan:

Query ini digunakan untuk menghitung total pendapatan berdasarkan jenis layanan. Data transaksi dikelompokkan per layanan, kemudian nilai sub\_total dijumlahkan menggunakan SUM(). Hasilnya menunjukkan berapa total pemasukan dari setiap layanan, seperti Print Warna dan Laminating.

	jenis_layanan	total_pendapatan
▶	Fotocopy Hitam Putih	5000
	Jilid Spiral	7000
	Laminating	14500
	Print Warna	28500

*Ini adalah output dari query Group By & Agregasi*

e. Having

HAVING digunakan untuk menyaring hasil dari proses pengelompokan yang dilakukan oleh GROUP BY. Berbeda dengan WHERE yang menyaring data sebelum pengelompokan, HAVING menyaring data setelah proses agregasi. Dalam sistem ini, HAVING digunakan untuk menampilkan hanya layanan yang memiliki total pendapatan di atas nilai tertentu, sehingga informasi yang ditampilkan menjadi lebih spesifik.

```
73 •   SELECT b.jenis_layanan,  
74           SUM(t.sub_total) AS total_pendapatan  
75     FROM transaksi t  
76   JOIN barang b ON t.id_barang = b.id_barang  
77   GROUP BY b.jenis_layanan  
78   HAVING SUM(t.sub_total) > 10000;
```

*Ini adalah kode query Having*

Penjelasan:

Query ini merupakan lanjutan dari pengelompokan sebelumnya, tetapi ditambahkan kondisi HAVING. Hanya layanan yang memiliki total pendapatan lebih dari 10.000 yang akan ditampilkan. Dengan query ini, data yang muncul menjadi lebih spesifik dan terfokus.

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	jenis_layanan	total_pendapatan		
▶	Laminating	14500		
	Print Warna	28500		

*Ini adalah hasil output dari query having*

f. Query Subquery

Subquery adalah query yang berada di dalam query lain. Pada sistem ini, subquery digunakan untuk menyeleksi data berdasarkan hasil perhitungan tertentu, misalnya menampilkan pelanggan yang memiliki total transaksi di atas batas tertentu. Dengan subquery, proses pencarian data menjadi lebih fleksibel dan mampu menghasilkan informasi yang lebih terarah.

```
80 •   SELECT nama_pelanggan
81     FROM pelanggan
82     WHERE id_pelanggan IN (
83         SELECT id_pelanggan
84         FROM transaksi
85         GROUP BY id_pelanggan
86         HAVING SUM(total_bayar) >= 20000
87     );
```

*Ini adalah kode dari query subquery*

Penjelasan:

Query ini digunakan untuk menampilkan nama pelanggan yang memiliki total pembayaran minimal 20.000. Subquery di dalam tanda kurung menghitung total pembayaran setiap pelanggan, lalu memilih id\_pelanggan yang memenuhi syarat. Hasil dari subquery tersebut kemudian digunakan oleh query utama untuk menampilkan nama pelanggan yang sesuai.

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

	nama_pelanggan
▶	Daniel
	Rina

*Ini adalah hasil output dari query Subquery*

## BAB IV

### PENUTUP

#### 4.1 Hasil Pengujian Query

Pengujian dilakukan terhadap seluruh query yang telah dibuat pada sistem basis data Fotocopy & Print “Focus”, meliputi query JOIN, GROUP BY dengan agregasi, HAVING, serta skenario transaksi menggunakan TCL. Pengujian bertujuan untuk memastikan bahwa struktur tabel, relasi, dan perintah SQL yang digunakan berjalan sesuai dengan rancangan.

Hasil pengujian menunjukkan bahwa query **JOIN** berhasil menampilkan data transaksi secara lengkap dengan menggabungkan tabel pelanggan, kasir, barang, dan transaksi. Data yang sebelumnya berupa kode atau ID dapat ditampilkan dalam bentuk informasi yang mudah dipahami, seperti nama pelanggan, nama kasir, jenis layanan, jumlah, dan tanggal transaksi.

Query **GROUP BY** dengan fungsi agregasi SUM() berhasil menghitung total pendapatan berdasarkan jenis layanan. Dari hasil yang diperoleh, sistem mampu menampilkan ringkasan pendapatan untuk setiap layanan, sehingga memudahkan analisis layanan yang paling sering digunakan atau paling menghasilkan.

Query **HAVING** berjalan dengan baik dalam menyaring hasil agregasi. Hanya layanan yang memiliki total pendapatan di atas nilai tertentu yang ditampilkan, sehingga informasi yang dihasilkan menjadi lebih spesifik dan relevan.

Selain itu, pengujian skenario transaksi menggunakan **TCL** menunjukkan bahwa proses penyimpanan data berjalan aman. Perintah START TRANSACTION dan COMMIT berhasil menyimpan data transaksi baru secara konsisten, serta memastikan bahwa perubahan data hanya tersimpan ketika proses berjalan tanpa kesalahan. Berdasarkan hasil tersebut, seluruh query yang digunakan pada sistem ini dapat dijalankan dengan baik dan mampu menghasilkan informasi sesuai dengan kebutuhan pengelolaan transaksi pada usaha Fotocopy & Print “Focus”.

## **4.2 Ringkasan Hasil Uji**

Berdasarkan seluruh pengujian yang telah dilakukan, sistem basis data Fotocopy & Print “Focus” dapat berjalan sesuai dengan perancangan. Seluruh tabel berhasil dibuat berdasarkan ERD, data dapat dimasukkan tanpa kesalahan, dan relasi antar tabel berfungsi dengan baik. Query JOIN mampu menampilkan data transaksi secara lengkap, sedangkan GROUP BY, agregasi, dan HAVING berhasil menghasilkan ringkasan informasi sesuai kebutuhan. Skenario transaksi menggunakan TCL juga berjalan aman, sehingga perubahan data tersimpan secara konsisten. Hasil ini menunjukkan bahwa struktur basis data dan perintah SQL yang digunakan telah memenuhi kebutuhan pengelolaan transaksi secara terstruktur dan akurat.

## **4.3 Kendala dan Perbaikan**

Selama proses pengerjaan, kendala yang paling sering muncul adalah kesalahan saat membuat foreign key. Beberapa data transaksi tidak dapat dimasukkan karena tipe data antara primary key dan foreign key tidak sama. Masalah ini diperbaiki dengan menyamakan tipe data pada kolom yang saling berhubungan serta memastikan tabel induk dibuat terlebih dahulu sebelum tabel transaksi.

Kendala lainnya adalah kesalahan penulisan query, seperti nama kolom yang tidak sesuai atau urutan perintah yang keliru, sehingga query tidak dapat dijalankan. Perbaikan dilakukan dengan memeriksa kembali struktur tabel dan menyesuaikan penulisan query sesuai skema yang ada.

Dari sisi sistem, tabel transaksi saat ini hanya dapat menyimpan satu layanan dalam satu baris. Untuk pengembangan ke depan, sistem dapat diperbaiki dengan memisahkan transaksi menjadi tabel utama dan detail transaksi agar satu nota dapat memuat banyak layanan. Selain itu, perhitungan nilai seperti sub\_total dan total\_bayar masih dilakukan secara manual, sehingga berpotensi menimbulkan kesalahan. Perbaikan yang disarankan adalah menambahkan perhitungan otomatis melalui sistem atau aplikasi agar data lebih akurat dan mudah digunakan.

#### **4.4 Kesimpulan**

Berdasarkan seluruh tahapan yang telah dilakukan, mulai dari analisis data mentah, proses normalisasi hingga 3NF, perancangan ERD, hingga implementasi basis data ke dalam DBMS, dapat disimpulkan bahwa sistem basis data transaksi untuk Fotocopy & Print “Focus” berhasil dirancang dengan baik. Data yang sebelumnya tidak terstruktur kini tersusun rapi dalam beberapa tabel yang saling berelasi, sehingga mengurangi pengulangan data dan mempermudah pengelolaan informasi.

Implementasi database menggunakan MySQL menunjukkan bahwa struktur tabel, relasi, serta perintah SQL yang digunakan dapat berjalan sesuai dengan rancangan. Proses pembuatan tabel, pengisian data, pengaturan transaksi, serta penggunaan query JOIN, GROUP BY, dan HAVING berhasil menghasilkan informasi yang dibutuhkan, seperti laporan transaksi dan ringkasan pendapatan.

Dengan demikian, sistem yang dibangun mampu menjadi dasar yang kuat untuk pengelolaan data transaksi secara lebih teratur dan akurat. Meskipun masih sederhana, rancangan ini sudah cukup menggambarkan bagaimana sebuah basis data dapat digunakan untuk mendukung kegiatan operasional usaha fotocopy secara lebih efektif.

#### **4.5 Saran Pengembangan**

Sistem basis data yang dirancang pada proyek ini masih berfokus pada pengelolaan data transaksi. Ke depannya, sistem ini dapat dikembangkan menjadi aplikasi sederhana berbasis web atau desktop agar proses input data menjadi lebih mudah dan tidak perlu dilakukan langsung melalui perintah SQL. Selain itu, struktur transaksi dapat diperbaiki dengan memisahkan data transaksi menjadi tabel utama dan tabel detail, sehingga satu transaksi dapat memuat lebih dari satu layanan dalam satu nota. Hal ini akan membuat sistem lebih fleksibel dan mendekati kondisi nyata di lapangan.

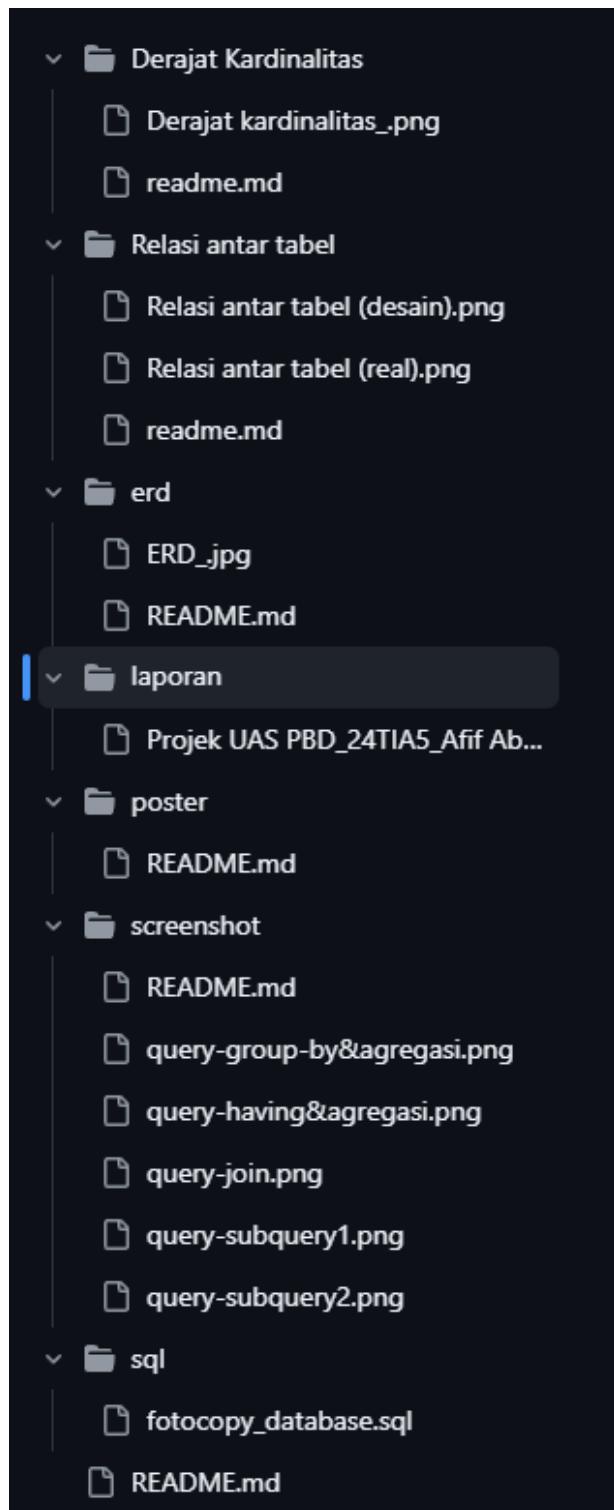
Pengembangan lain yang dapat dilakukan adalah menambahkan perhitungan otomatis untuk nilai sub\_total, total\_bayar, dan kurang\_sisa, sehingga kasir tidak perlu menghitung secara manual. Dengan pengembangan tersebut, sistem akan menjadi lebih praktis, mengurangi kesalahan input, dan lebih siap digunakan dalam kegiatan operasional sehari-hari.

## 4.6 Lampiran

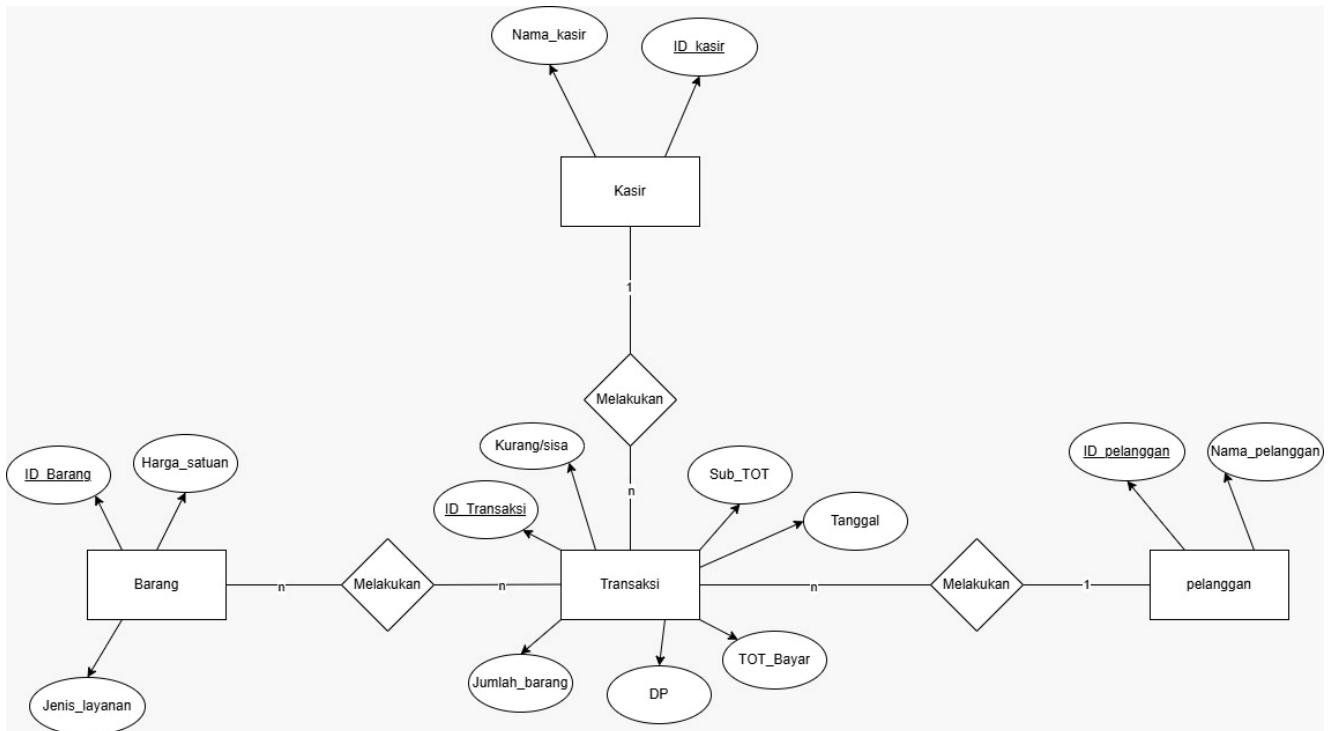
### A. Tautan Repository

<https://github.com/Dhigjay/uas-pbd-fotocopy-focus>

### B. Struktur Folder Repository



### C. ERD Final Sistem



Gambar ini adalah ERD final

### D. Hasil Eksekusi Query

#### 1 Query Join

```

SELECT t.id_transaksi, p.nama_pelanggan, k.nama_kasir,
       b.jenis_layanan, t.jumlah_barang, t.sub_total, t.tanggal
FROM transaksi t
JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan
JOIN kasir k ON t.id_kasir = k.id_kasir
JOIN barang b ON t.id_barang = b.id_barang;
    
```

Gambar ini adalah kode Program Query Join

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

	id_transaksi	nama_pelanggan	nama_kasir	jenis_layanan	jumlah_barang	sub_total	tanggal
▶	120	Daniel	Albert	Print Warna	1	9500	2025-05-18
	121	Daniel	Albert	Laminating	1	14500	2025-05-18
	122	Rina	Sinta	Fotocopy Hitam Putih	10	5000	2025-05-19
	123	Budi	Albert	Jilid Spiral	1	7000	2025-05-19
	124	Rina	Sinta	Print Warna	2	19000	2025-05-20

Gambar ini adalah hasil running dari kode query join

## 2 Query Group By & Agregasi

```
SELECT b.jenis_layanan,
       SUM(t.sub_total) AS total_pendapatan
  FROM transaksi t
 JOIN barang b ON t.id_barang = b.id_barang
 GROUP BY b.jenis_layanan;
```

Gambar ini adalah kode program Query Group By & Agregasi

Result Grid | Filter Rows: Export

	jenis_layanan	total_pendapatan
▶	Fotocopy Hitam Putih	5000
	Jilid Spiral	7000
	Laminating	14500
	Print Warna	28500

Gambar ini adalah hasil running dari kode program query Group By & Agregasi

## 3 Query Having

```
SELECT b.jenis_layanan,
       SUM(t.sub_total) AS total_pendapatan
  FROM transaksi t
 JOIN barang b ON t.id_barang = b.id_barang
 GROUP BY b.jenis_layanan
 HAVING SUM(t.sub_total) > 10000;
```

Gambar ini adalah kode program Query Having

Result Grid | Filter Rows: Export:

	jenis_layanan	total_pendapatan
▶	Laminating	14500
	Print Warna	28500

Gambar ini adalah hasil running dari kode Query Having

#### 4 Query Subquery

```

SELECT nama_pelanggan
FROM pelanggan
WHERE id_pelanggan IN (
    SELECT id_pelanggan
    FROM transaksi
    GROUP BY id_pelanggan
    HAVING SUM(total_bayar) >= 20000
);
  
```

Gambar ini adalah kode program dari Query Subquery

Result Grid | Filter Rows:

	nama_pelanggan
▶	Daniel
	Rina

Gambar ini adalah hasil running dari kode program Query Subquery

```
select * from transaksi;
```

Kode tersebut berfungsi untuk menampilkan tabel transaksi

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id_transaksi	id_pelanggan	id_kasir	id_barang	jumlah_barang	tanggal	dp	total_bayar	sub_total	kurang_sisa
▶	120	98765456	k0001	76348001	1	2025-05-18	0	22000	9500	0
	121	98765456	k0001	23591002	1	2025-05-18	0	22000	14500	0
	122	98765457	k0002	99112003	10	2025-05-19	0	5000	5000	0
	123	98765458	k0001	88123004	1	2025-05-19	0	7000	7000	0
*	124	98765457	k0002	76348001	2	2025-05-20	0	19000	19000	0
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Gambar ini adalah hasil running dari kode di atas yaitu menampilkan tabel transaksi

## E. Script SQL Lengkap

```
1 •  CREATE DATABASE fotocopy_db;
2 •  USE fotocopy_db;
3
4 •  CREATE TABLE pelanggan (
5      id_pelanggan INT PRIMARY KEY,
6      nama_pelanggan VARCHAR(100) NOT NULL
7 );
8
9 •  CREATE TABLE kasir (
10     id_kasir VARCHAR(10) PRIMARY KEY,
11     nama_kasir VARCHAR(100) NOT NULL
12 );
13
14 •  CREATE TABLE barang (
15     id_barang INT PRIMARY KEY,
16     jenis_layanan VARCHAR(50) NOT NULL,
17     harga_satuan INT NOT NULL
18 );
19
```

```
20 • CREATE TABLE transaksi (
21     id_transaksi INT PRIMARY KEY,
22     id_pelanggan INT,
23     id_kasir VARCHAR(10),
24     id_barang INT,
25     jumlah_barang INT,
26     tanggal DATE,
27     dp INT,
28     total_bayar INT,
29     sub_total INT,
30     kurang_sisa INT,
31     CONSTRAINT fk_pelanggan FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan),
32     CONSTRAINT fk_kasir FOREIGN KEY (id_kasir) REFERENCES kasir(id_kasir),
33     CONSTRAINT fk_barang FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
34 );
35
36
37 • INSERT INTO pelanggan VALUES
38     (98765456, 'Daniel'),
39     (98765457, 'Rina'),
40     (98765458, 'Budi');
41
42 • INSERT INTO kasir VALUES
43     ('k0001', 'Albert'),
44     ('k0002', 'Sinta');
45
46 • INSERT INTO barang VALUES
47     (76348001, 'Print Warna', 9500),
48     (23591002, 'Laminating', 14500),
49     (99112003, 'Fotocopy Hitam Putih', 500),
50     (88123004, 'Jilid Spiral', 7000);
51
52 • INSERT INTO transaksi VALUES
53     (120, 98765456, 'k0001', 76348001, 1, '2025-05-18', 0, 9500, 9500, 0),
54     (121, 98765456, 'k0001', 23591002, 1, '2025-05-18', 0, 14500, 14500, 0),
55     (122, 98765457, 'k0002', 99112003, 10, '2025-05-19', 0, 5000, 5000, 0),
56     (123, 98765458, 'k0001', 88123004, 1, '2025-05-19', 0, 7000, 7000, 0),
57     (124, 98765457, 'k0002', 76348001, 2, '2025-05-20', 0, 19000, 19000, 0);
58
```

```
63 •    START TRANSACTION;
64 •    INSERT INTO transaksi VALUES
65      (125, 98765456, 'k0001', 76348001, 2, '2025-05-21', 0, 19000, 19000, 0);
66 •    COMMIT;
67
68 •    SELECT * FROM transaksi;
69
70
71 •    SELECT t.id_transaksi, p.nama_pelanggan, k.nama_kasir,
72          b.jenis_layanan, t.jumlah_barang, t.sub_total, t.tanggal
73    FROM transaksi t
74    JOIN pelanggan p ON t.id_pelanggan = p.id_pelanggan
75    JOIN kasir k ON t.id_kasir = k.id_kasir
76    JOIN barang b ON t.id_barang = b.id_barang;
77
78
79 •    SELECT b.jenis_layanan,
80          SUM(t.sub_total) AS total_pendapatan
81    FROM transaksi t
82    JOIN barang b ON t.id_barang = b.id_barang
83    GROUP BY b.jenis_layanan;
84
85
86 •    SELECT b.jenis_layanan,
87          SUM(t.sub_total) AS total_pendapatan
88    FROM transaksi t
89    JOIN barang b ON t.id_barang = b.id_barang
90    GROUP BY b.jenis_layanan
91    HAVING SUM(t.sub_total) > 10000;
```

```

94 •     SELECT nama_pelanggan
95      FROM pelanggan
96      WHERE id_pelanggan IN (
97          SELECT id_pelanggan
98          FROM transaksi
99          GROUP BY id_pelanggan
100         HAVING SUM(total_bayar) >= 20000
101     );
102
103
104 •     SELECT nama_pelanggan
105      FROM pelanggan
106      WHERE id_pelanggan IN (
107          SELECT id_pelanggan
108          FROM transaksi
109          GROUP BY id_pelanggan
110         HAVING SUM(total_bayar) <= 20000
111     );

```

## F. Database (PHP MyAdmin)

Table	Action	Rows	Type	Collation	Size	Overhead
barang		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
kasir		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
pelanggan		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
transaksi		2	InnoDB	utf8mb4_general_ci	64.0 KiB	-
4 tables		6	InnoDB utf8mb4_general_ci		112.0 KiB	0 B

Gambar ini adalah database dari sistem kami

			id_barang	jenis_layanan	harga_satuan	
<input type="checkbox"/>	Edit	Copy	Delete	23591002	Laminating	14500
<input type="checkbox"/>	Edit	Copy	Delete	76348001	Print Warna	9500
<input type="checkbox"/>	Edit	Copy	Delete	88123004	Jilid Spiral	7000
<input type="checkbox"/>	Edit	Copy	Delete	99112003	Fotocopy Hitam Putih	500

Gambar ini adalah isi dari tabel barang

	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	id_kasir	nama_kasir
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	k0001	Albert
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	k0002	Sinta

Gambar ini adalah isi dari tabel kasir

	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	id_pelanggan	nama_pelanggan
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	98765456	Daniel
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	98765457	Rina
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	98765458	Budi

Gambar ini adalah isi dari tabel pelanggan

	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	id_transaksi	id_pelanggan	id_kasir	id_barang	jumlah_barang	tanggal	dp	total_bayar	sub_total	kurang_sisa
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	120	98765456	k0001	76348001	1	2025-05-18	0	22000	9500	0
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	121	98765456	k0001	23591002	1	2025-05-18	0	22000	14500	0
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	122	98765457	k0002	99112003	10	2025-05-19	0	5000	5000	0
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	123	98765458	k0001	88123004	1	2025-05-19	0	7000	7000	0
	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	124	98765457	k0002	76348001	2	2025-05-20	0	19000	19000	0

Gambar ini adalah isi dari tabel transaksi

## DAFTAR PUSTAKA

- 1 Connolly T, Begg C. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. Harlow: Pearson Education; 2015.
- 2 Elmasri R, Navathe SB. *Fundamentals of Database Systems*. 7th ed. Boston: Pearson; 2016.
- 3 Silberschatz A, Korth HF, Sudarshan S. *Database System Concepts*. 7th ed. New York: McGraw-Hill Education; 2020.
- 4 Kroenke DM, Auer DJ. *Database Processing: Fundamentals, Design, and Implementation*. 14th ed. Boston: Pearson; 2015.
- 5 Ramakrishnan R, Gehrke J. *Database Management Systems*. 3rd ed. New York: McGraw-Hill; 2014.
- 6 Hoffer JA, Ramesh V, Topi H. *Modern Database Management*. 12th ed. Boston: Pearson; 2016.
- 7 Kadir A. *Pengenalan Sistem Informasi*. Yogyakarta: Andi Offset; 2014.
- 8 Oracle Corporation. *MySQL 8.0 Reference Manual* [Internet]. 2024 [cited 2026 Jan 17]. Available from: <https://dev.mysql.com/doc/>
- 9 Oracle Corporation. *MySQL Workbench Manual* [Internet]. 2024 [cited 2026 Jan 17]. Available from: <https://dev.mysql.com/doc/workbench/en/>