

**PERANCANGAN SISTEM BASIS DATA PENJUALAN RETAIL
BERDASARKAN STUDI KASUS NOTA TRANSAKSI
INDOMARET**



Dosen Pengampu :

Ridwan Dwi Irawan, S. Kom., M. Kom.

Disusun oleh :

Wedyawati 240103179

Heva Maya Oktaviani 240103162

Muhammad Anis Maksum W 240103169

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

DAFTAR ISI

DAFTAR ISI.....	2
A. Pengantar Pemrograman Basis Data	5
B. Latar Belakang Studi Kasus.....	5
C. Tujuan Proyek.....	5
D. Ruang Lingkup dan Batasan	6
E. Gambaran Umum Sistem yang Dibuat.....	6
BAB II.....	7
A. Basis Data dan DBMS	7
B. Relasi dan Tabel.....	8
C. Normalisasi Basis Data	8
D. Entity Relationship Diagram (ERD)	9
E. Structured Query Language (SQL).....	9
F. Primary Key dan Foreign Key pada sistem	10
G. Penerapan ERD ke DBMS dengan MySQL Workbench.....	11
H. Penegasan Materi Bab 2	11
BAB III.....	13
A. Studi Kasus	13
B. Tujuan Utama Sistem	14
C. Kebutuhan Sistem	14
D. Data Awal dan Proses Normalisasi.....	15
E. Hubungan antar Entitas	19
F. Drafar Kardinalitas.....	21
G. Relasi antar Tabel	22
H. ERD (Entity Relation Diagram.....	24
I. Entitas dan atribut utama	24
J. Penerapan ERD KE DBMS	25
K. Implementasi DDL (Data Defition Language)	25
BAB IV.....	32
A. Hasil Pengujian Query	32
B. Ringkasan Hasil Uji	32
C. Kendala dan Perbaikan.....	32
D. Kesimpulan.....	32
E. Saran Pengembangan.....	33

Lampiran	34
DAFTAR PUSAKA	42

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 judul "Perancangan Sistem Basis Data Penjualan Retail Berdasarkan Studi Kasus Nota Transaksi Indomaret".

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 sektor retail, di mana pengelolaan data yang masif dan cepat sangat diperlukan untuk mendukung kegiatan operasional, pelaporan, serta pengambilan keputusan bisnis yang akurat.

Dalam laporan ini, penulis melakukan analisis mendalam terhadap komponen data yang terdapat pada nota transaksi Indomaret, kemudian menerapkan tahapan 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 arsitektur basis data retail 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 Bapak Ridwan Dwi Irawan, S. Kom., M. Kom., selaku 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 retail.

BAB I

PENDAHULUAN

A. Pengantar Pemrograman Basis Data

Pemrograman basis data merupakan disiplin ilmu dalam informatika yang fokus pada pengaturan, pengelolaan, dan manipulasi data yang tersimpan dalam sistem manajemen basis data (DBMS). Dalam era digital saat ini, data merupakan aset yang sangat berharga bagi organisasi. Basis data yang dirancang dengan baik memungkinkan penyimpanan data yang efisien, keamanan yang terjamin, serta kecepatan dalam pengambilan informasi. Sesuai dengan materi pertemuan awal, prinsip utama dalam pemrograman basis data adalah memastikan integritas data melalui mekanisme normalisasi dan penggunaan bahasa kueri terstruktur (*Structured Query Language*) untuk mengelola relasi antar entitas.

B. Latar Belakang Studi Kasus

Sektor retail seperti Indomaret memproses ribuan transaksi setiap harinya. Setiap transaksi menghasilkan bukti fisik berupa nota atau struk belanja yang memuat informasi kompleks, mulai dari data toko, waktu transaksi, rincian barang, harga, pajak (PPN), hingga metode pembayaran. Namun, data pada struk belanja fisik bersifat statis dan tidak terstruktur secara digital.

Untuk dapat mengelola data tersebut menjadi informasi yang berguna bagi manajemen—seperti laporan stok barang, tren penjualan, atau analisis loyalitas pelanggan—maka diperlukan sebuah sistem basis data relasional. Studi kasus ini mengambil satu sampel nota transaksi Indomaret cabang Slamet Riyadi No.100 untuk dianalisis dan ditransformasikan ke dalam bentuk skema basis data yang standar dan optimal.

C. Tujuan Proyek

Tujuan dari penyusunan proyek tugas UAS ini adalah :

1. Menganalisis komponen data yang terdapat pada nota transaksi retail (Indomaret).
2. Menerapkan teknik normalisasi (1NF, 2NF, 3NF) untuk menghindari redundansi data dan menjaga integritas data.
3. Menentukan derajat kardinalitas dan relasi antar tabel secara akurat.

4. Merancang struktur basis data final dalam bentuk *Entity Relationship Diagram* (ERD).
5. Memenuhi syarat penilaian Ujian Akhir Semester mata kuliah Pemrograman Basis Data.

D. Ruang Lingkup dan Batasan

Agar pembahasan lebih terfokus, maka ruang lingkup dan batasan proyek ini adalah:

1. Objek Penelitian : Analisis terbatas pada data yang tertera pada satu lembar struk transaksi Indomaret.
2. Fungsi Sistem : Perancangan hanya berfokus pada struktur basis data (Back-end) untuk mencatat transaksi penjualan, tidak mencakup pembuatan antarmuka aplikasi (Front-end).
3. Data yang Diolah : Meliputi data identitas toko, data produk (nama, harga, jumlah), rincian potongan harga (voucher), data pajak (PPN), dan identitas transaksi (ID Transaksi, Kasir, Waktu).

E. Gambaran Umum Sistem yang Dibuat

Sistem basis data yang dirancang akan mampu menyimpan data transaksi secara relasional. Sistem ini akan memisahkan informasi yang bersifat tetap (seperti nama produk dan lokasi toko) dengan informasi yang bersifat dinamis (seperti detail transaksi setiap pelanggan). Dengan struktur ini, sistem dapat menghasilkan laporan penjualan yang detail tanpa adanya duplikasi data yang tidak perlu, serta memudahkan admin dalam melakukan pembaruan data harga atau stok di masa mendatang.

BAB II

LANDASAN TEORI

A. Basis Data dan DBMS

Basis data (database) merupakan kumpulan data yang saling berhubungan dan disimpan secara sistematis di dalam media penyimpanan komputer sehingga dapat diolah, diakses, dan dikelola dengan mudah. Data yang tersimpan dalam basis data biasanya terstruktur dan merepresentasikan objek atau kejadian tertentu, seperti data barang, data transaksi, data pelanggan, dan data kasir pada sistem penjualan retail. Keberadaan basis data bertujuan untuk menghindari duplikasi data, meningkatkan konsistensi, serta mempermudah proses pencarian dan pengolahan informasi.

Dalam konteks penjualan retail seperti Indomaret, basis data memegang peranan penting karena seluruh aktivitas transaksi dicatat secara terkomputerisasi. Informasi yang terdapat pada nota transaksi seperti nomor transaksi, tanggal transaksi, kode barang, nama barang, jumlah pembelian, harga satuan, total pembayaran, dan metode pembayaran—merupakan contoh data yang harus disimpan secara terstruktur dalam basis data agar dapat digunakan kembali untuk keperluan laporan penjualan, stok barang, dan analisis bisnis.

Sistem Manajemen Basis Data atau Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk membuat, mengelola, dan mengendalikan basis data. DBMS berfungsi sebagai perantara antara pengguna (user) dengan basis data, sehingga pengguna tidak perlu berinteraksi langsung dengan struktur penyimpanan fisik data. Contoh DBMS yang umum digunakan antara lain MySQL, PostgreSQL, Oracle, dan SQL Server.

DBMS menyediakan berbagai fasilitas penting, seperti:

1. Pendefinisian data (Data Definition), yaitu kemampuan untuk mendefinisikan struktur basis data, tabel, atribut, serta relasi antar tabel.
2. Manipulasi data (Data Manipulation), meliputi proses penyimpanan, pengubahan, penghapusan, dan pengambilan data menggunakan perintah SQL.
3. Keamanan data (Data Security), yaitu pengaturan hak akses pengguna agar data hanya dapat diakses oleh pihak yang berwenang.
4. Integritas data (Data Integrity), untuk memastikan bahwa data yang tersimpan selalu valid dan konsisten.

5. Pengendalian transaksi (Transaction Control), yang menjamin bahwa setiap transaksi penjualan diproses secara benar dan aman, terutama saat terjadi kegagalan sistem.

Pada sistem penjualan retail Indomaret, DBMS digunakan untuk mengelola data transaksi harian yang jumlahnya sangat besar. Dengan adanya DBMS, data transaksi dapat disimpan secara terpusat, diolah dengan cepat, serta menghasilkan laporan penjualan yang akurat dan real-time. Oleh karena itu, pemahaman mengenai konsep basis data dan DBMS menjadi landasan utama dalam perancangan sistem basis data penjualan retail yang efektif dan efisien.

B. Relasi dan Tabel

Dalam perancangan basis data relasional sistem ini, data disimpan dalam bentuk tabel yang terstruktur. Setiap tabel terdiri dari Field (kolom) yang mewakili atribut data dan Record (baris) yang mewakili satu satuan data lengkap. tabel-tabel utama yang digunakan adalah:

- a. Tabel Kasir: Menyimpan identitas petugas yang melayani transaksi.
- b. Tabel Member: Menyimpan data pelanggan tetap dan poin yang dikumpulkan.
- c. Tabel Barang: Menyimpan daftar katalog produk beserta harga satuannya.
- d. Tabel Transaksi: Menyimpan data utama nota (header), seperti tanggal dan total bayar.
- e. Tabel Detail_Transaksi: Menyimpan rincian item barang yang dibeli pada setiap nota.

Agar data antar tabel dapat saling terhubung secara logis, digunakan mekanisme relasi. Relasi ini dibentuk menggunakan dua komponen kunci, yaitu:

- a. Primary Key (PK): Sebagai identitas unik pada suatu tabel yang membedakan satu data dengan data lainnya.
- b. Foreign Key (FK): Sebagai kunci tamu yang berfungsi sebagai penghubung ke tabel lain.

C. Normalisasi Basis Data

Proses ini bertujuan untuk menghilangkan redundansi data :

- a. 1NF (First Normal Form)
Memastikan setiap kolom berisi nilai atomik dan tidak ada grup berulang (seperti baris item barang pada struk).
- b. 2NF (Second Normal Form)

Memenuhi 1NF dan memastikan semua atribut non-kunci bergantung penuh pada *Primary Key*.

c. 3NF (Third Normal Form)

Memenuhi 2NF dan menghilangkan ketergantungan transitif (misal: Nama Toko tidak boleh bergantung pada ID Transaksi, melainkan harus pada ID Toko).

D. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model konseptual yang digunakan untuk menggambarkan struktur basis data dalam bentuk entitas, atribut, dan relasi. Komponen utama ERD meliputi:

- a. Entitas: objek yang memiliki data
- b. Atribut: karakteristik dari entitas
- c. Relasi: hubungan antar entitas

ERD digunakan sebagai dasar dalam pembuatan struktur tabel pada DBMS.

E. 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

F. 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 Member (Pelanggan)
 - a. Primary Key: id_member : Digunakan untuk membedakan setiap pelanggan secara unik.
2. Tabel Kasir
 - a. Primary Key: id_kasir : Digunakan sebagai identitas unik setiap kasir.
3. Tabel Barang
 - a. Primary Key: id_barang : Digunakan untuk membedakan setiap jenis barang atau produk.
4. Tabel Transaksi
 - a. Primary Key: id_transaksi
 - b. Foreign Key:
 - a. id_member : Mengacu ke tabel Member.
 - b. id_kasir : Mengacu ke tabel Kasir.
5. Tabel Detail_Transaksi
 - a. Primary Key: id_detail
 - b. Foreign Key:
 - 1. id_transaksi : Mengacu ke tabel Transaksi.
 - 2. id_barang : Mengacu ke tabel Barang.
6. Tabel_pembayaran
 - a. Primary Key: id_pembayaran

b. Foreign Key: id_transaksi

Dengan struktur ini, setiap transaksi dapat diketahui siapa pelanggannya, siapa kasir yang melayani, serta rincian barang apa saja yang dibeli. Relasi antar tabel menjadi lebih terjaga dan data tersimpan secara konsisten tanpa ada pengulangan yang berlebihan.

G. Penerapan ERD ke DBMS dengan MySQL Workbench

Langkah selanjutnya adalah mengimplementasikan desain tersebut ke dalam *Database Management System* (DBMS) menggunakan MySQL Workbench. Proses ini merupakan tahap transformasi dari model logis menjadi struktur fisik basis data yang siap digunakan untuk menyimpan data transaksi retail. Proses penerapan ini mencakup beberapa tahapan teknis sebagai berikut:

1. Transformasi Entitas Menjadi Tabel : Setiap entitas yang telah dirancang dalam ERD, seperti Kasir, Member, Barang, dan Transaksi, diubah menjadi tabel-tabel di MySQL. Atribut pada entitas diatur menjadi kolom (*field*) dengan tipe data yang sesuai, misalnya VARCHAR untuk nama dan DECIMAL untuk nominal harga.
2. Implementasi Kunci (PK dan FK) : Penerapan kunci dilakukan untuk menjamin integritas data. Kolom identitas unik seperti id_transaksi dan id_barang ditetapkan sebagai *Primary Key* (PK). Sedangkan kolom penghubung di tabel Transaksi dan Detail_Transaksi ditetapkan sebagai *Foreign Key* (FK) untuk membentuk relasi antar tabel.
3. Pembuatan EER Diagram : Melalui fitur *Enhanced Entity-Relationship* (EER) di MySQL Workbench, struktur tabel divisualisasikan kembali secara fisik untuk memastikan semua hubungan (kardinalitas) seperti 1:N antara Kasir ke Transaksi atau Barang ke Detail_Transaksi telah terpasang dengan benar.
4. Eksekusi DDL (Forward Engineering): Tahap akhir adalah melakukan *Forward Engineering* untuk menghasilkan *script* SQL otomatis (CREATE TABLE). *Script* ini kemudian dieksekusi di server MySQL sehingga struktur basis data terbentuk secara nyata dan siap dioperasikan menggunakan perintah DML.

H. Penegasan Materi Bab 2

Secara keseluruhan, materi pada Bab 2 ini telah menguraikan landasan teori yang komprehensif mengenai perancangan basis data relasional. Mulai dari pemahaman dasar

tentang DBMS, teknik normalisasi untuk menghilangkan redundansi, hingga perancangan model data menggunakan ERD. Seluruh komponen ini merupakan satu kesatuan yang saling berkaitan guna memastikan bahwa sistem basis data penjualan retail Indomaret yang dibangun memiliki struktur yang efisien, aman, dan mampu menjaga integritas data secara konsisten sesuai dengan prinsip *Structured Query Language* (SQL).

BAB III

PERANCANGAN DAN IMPLEMENTASI BASIS DATA

A. Studi Kasus

Struk Transaksi

PT. INDOMARCO PRISMATAMA
GEDUNG MENARA INDOMARET
BOULEVARD PANTAI INDAH KAPUK
JAKARTA UTARA
NPWP: 001.337.994.6-092.000



SLAMET RIYADI NO.100
JL SLAMET RIYADI NO 100 KEL GAYAM
KEC SUKOHARJO KAB SUKOHARJO, 57514

04.01.26-04:12/4.1.4/TON8-36330/PUGUH/02

STARBUCKS LATTE 220 1 15900 15,900
AQUA AIR MINERAL 600 2 3800 7,600
VOUCHER : (1,700)
SR/RT SBK DUO COK 72 1 8000 8,000

TOTAL BELANJA : 29,800

NON TUNAI : 29,800
ANDA HEMAT : 1,700
PPN : DPP= 26,014 PPN= 3,122
HARGA JUAL : 28,378
QR BRI-TRXID:TON82601040412080102
NO:*****AAAG, PURCHASE:29,800
ID POINKU : xxxxxxxx3207/WExxxxxxx
CEK POIN/STAMP/LUCKY DI APPS POINKU

Gambar 1. Studi Kasus Nota Transaksi

Sistem ini dirancang untuk mengatasi sifat data pada struk belanja fisik yang statis. Dengan menerapkan tahapan Normalisasi (1NF, 2NF, 3NF), data tersebut dipecah ke dalam beberapa tabel yang saling berelasi guna mencapai efisiensi penyimpanan, menghindari pengulangan data (redundansi), serta menjaga integritas data agar tetap konsisten.

B. Tujuan Utama Sistem

Berdasarkan dokumen laporan dan analisis studi kasus yang telah disusun, sistem basis data ini dirancang karena beberapa alasan utama berikut:

1. Transformasi Data Statis ke Digital

Data yang terdapat pada struk belanja fisik (seperti nota Indomaret yang dianalisis) bersifat statis dan sulit untuk diolah kembali. Dengan merancang sistem ini, data yang sebelumnya hanya berupa kertas ditransformasikan ke dalam skema basis data relasional yang standar dan optimal.

2. Efisiensi Pengelolaan Data Masif

Sektor retail seperti Indomaret memproses ribuan transaksi setiap harinya. Sistem ini dirancang untuk:

- a. Menangani volume data yang besar: Memungkinkan penyimpanan data yang masif dengan cepat dan teratur.
- b. Menghindari redundansi: Menggunakan teknik normalisasi (1NF, 2NF, 3NF) untuk mencegah pengulangan data yang tidak perlu, sehingga penyimpanan lebih efisien.

3. Menjaga Integritas dan Keamanan Data

Rancangan ini bertujuan untuk memastikan data tetap konsisten dan valid. Melalui penggunaan *Primary Key* dan *Foreign Key*, hubungan antar entitas (seperti kasir, barang, dan transaksi) terjaga secara akurat. Hal ini mencegah terjadinya kesalahan data, seperti transaksi yang tidak memiliki catatan kasir atau barang yang tidak terdaftar.

4. Mendukung Pengambilan Keputusan Bisnis

Basis data yang terstruktur memungkinkan manajemen untuk mengubah kumpulan data mentah menjadi informasi yang berguna, seperti:

- a. Laporan Penjualan: Menghasilkan laporan yang detail dan tepat waktu (*real-time*).
- b. Analisis Stok: Memudahkan pemantauan persediaan barang.
- c. Tren Pelanggan: Menganalisis loyalitas pelanggan melalui data member.

C. Kebutuhan Sistem

Secara ringkas, sistem ini dirancang untuk mendigitalisasi data transaksi retail agar dapat dikelola secara otomatis. Kebutuhan utama meliputi:

1. Kebutuhan Data: Penyimpanan data master (Kasir, Barang, Member) dan data transaksi (Header dan Detail).
2. Kebutuhan Fungsional: Kemampuan sistem untuk menghitung total belanja, pajak (PPN), dan mengelola relasi antar tabel guna menjaga integritas data.

D. Data Awal dan Proses Normalisasi

1. Tabel Umum

Id_Transaksi	Tanggal	Nama_Kasir	Id_Member	Nama_Barang	Qty	Harga_Satuan	Subtotal	Voucher	Total_Bayar_	Ppn
TON8-36330	04.01.26 04:12	PUGUH	...3207	STARBUCKS LATTE 220	1	15.900	15.900		29.800	3,122
				AQUA AIR MINERAL 600	2	3.800	7.600	1.700		
				SR/RT SBK DUO COK 72	1	8.000	8.000			

Table 1. Tabel Umum

Tabel ini merepresentasikan data mentah yang diambil langsung dari struk Indomaret. Pada tahap ini, data masih memiliki grup berulang (satu nomor nota untuk banyak barang) dan belum atomik.

2. Tabel 1NF (First Normal Form)

Id_Transaksi	Tanggal	Nama_Kasir	Id_Member	Nama_Barang	Qty	Harga_Satuan	Subtotal	Voucher	Total_Bayar	Ppn
TON8-36330	04.01.26 04:12	PUGUH	...3207	STARBUCKS LATTE 220	1	15.900	15.900	1.700	29.800	3.122
TON8-36330	04.01.26 04:12	PUGUH	...3207	AQUA AIR MINERAL 600	2	3.800	7.600	1.700	29.800	3.122
TON8-36330	04.01.26 04:12	PUGUH	...3207	SR/RT SBK DUO COK 72	1	8.000	8.000	1.700	29.800	3.122

Table 2. Tabel 1NF

Tabel ini adalah hasil perbaikan dari Tabel Umum untuk memenuhi syarat normalisasi pertama. Menstandarisasi struktur data sehingga setiap atribut hanya memiliki satu nilai tunggal dalam satu baris. Menghilangkan sel yang digabung dan memastikan semua baris memiliki informasi yang lengkap. Data header (seperti TON8-36330, PUGUH, ...3207) kini diisi secara berulang pada setiap baris item barang. Walaupun terjadi redundansi data, struktur ini wajib dipenuhi sebelum masuk ke tahap 2NF untuk memisahkan entitas.

3. Tabel 2NF (Second Normal From)

a. Tabel Barang

Menyimpan data master produk. Atribut harga satuan dipisahkan ke sini karena melekat pada barang, bukan pada nomor transaksi.

id_barang (PK)	nama_barang	harga_satuan
B01	STARBUCKS LATTE 220	15.900
B02	AQUA AIR MINERAL 600	3.800
B03	SR/RT SBK DUO COK 72	8.000

Table 3. Tabel Barang 2NF

b. Tabel Kasir

Menyimpan identitas kasir secara unik agar nama kasir tidak perlu ditulis berulang kali di setiap nota.

id_kasir (PK)	nama_kasir
K01	PUGUH

Table 4. Tabel Kasir 2NF

c. Tabel Member

Menyimpan data pelanggan tetap (berdasarkan ID Poinku pada struk).

id_member (PK)	nama_member
...3207	Wedy

Table 5. Tabel Member 2NF

d. Tabel Transaksi

Menyimpan informasi umum nota. Data seperti tanggal, total bayar, dan PPN hanya dicatat satu kali per nomor transaksi.

id_transaksi (PK)	tanggal	id_kasir (FK)	id_member (FK)	voucher	total_bayar	ppn
TON8-36330	04.01.26 04:12	K01	...3207	1.700	29.800	3.122

Table 6. Tabel Transaksi 2NF

e. Tabel Detail_Transaksi

Menyimpan rincian item barang yang dibeli. Tabel ini menghubungkan antara transaksi dengan barang beserta jumlah (Qty) yang dibeli.

id_detail (PK)	id_transaksi (FK)	id_barang (FK)	qty	subtotal
1	TON8-36330	B01	1	15.900
2	TON8-36330	B02	2	7.600
3	TON8-36330	B03	1	8.000

Table 7. Tabel Detail Transaksi

f. Tabel Pembayaran

Menyimpan rincian pembayaran untuk setiap transaksi. Satu transaksi hanya memiliki satu catatan rincian pembayaran (Total, PPN, Voucher).

id_pembayaran (PK)	id_transaksi (FK)	ppn	voucher	total_bayar
P01	TRNS-36330	3.122	1.700	29.800

Table 8. Tabel Pembayaran 2NF

4. Tabel 3NF (Third Normal From)

- a. Kamus Basis Data
- b. Nama Database : Penjualan_Indomaret
- c. Nama Table : Tabel_Barang
- d. Fungsi : Menyimpan data barang

Nama field	Tipe	Panjang karakter	Keterangan
id_barang	Varchar	10	Primary key
nama_barang	Varchar	50	Nama produk
harga_satuan	Decimal	15	harga per unit

Table 9. Tabel Barang 3NF

- a. Kamus Basis Data
- b. Nama Database : Penjualan_Indomaret
- c. Nama Table : Tabel_Kasir
- d. Fungsi : Menyimpan identitas petugas kasir yang melayani transaksi.

Nama Field	Tipe Data	Panjang karakter	Keterangan
id_kasir	Varchar	10	Primary Key
nama_kasir	Varchar	50	Nama lengkap petugas kasir

Table 10. Tabel Kasir 3NF

- a. Kamus Basis Data
- b. Nama Database : Penjualan_Indomaret

- c. Nama Table : Tabel Member
- d. Fungsi : Menyimpan data pelanggan yang terdaftar dalam program loyalitas.

Nama Field	Tipe Data	Panjang karakter	Keterangan
id_member	Varchar	15	Primary Key
nama_member	Varchar	50	Nama lengkap pelanggan/ member

Table 11. Tabel Member 3NF

- a. Kamus Basis Data
- b. Nama Database : Penjualan_Indomaret
- c. Nama Table : Tabel_transaksi
- d. Fungsi : Menyimpan data utama (header) setiap nota transaksi

Nama Field	Tipe Data	Panjang karakter	Keterangan
id_transaksi	Varchar	20	Primary Key
tanggal	Datetime	-	tanggal dan waktu transaksi yang dilakukan
id_kasir	Varchar	10	Forign Key
id_member	Varchar	15	Foreign Key

Table 12. Tabel Transaksi 3NF

- a. Kamus Basis Data
- b. Nama Database : Penjualan_Indomaret
- c. Nama Table : Tabel_Detail_Transaksi
- d. Fungsi : Menyimpan rincian item barang yang dibeli pada setiap transaksi

Nama Field	Tipe Data	Panjang karakter	Keterangan
id_detail	Integer	11	Primary Key
id_transaksi	Varchar	20	Foreign Key
id_barang	Varchar	10	Forign Key
qty	Integer	11	Jumlah kuantitas barang yang dibeli

Nama Field	Tipe Data	Panjang karakter	Keterangan
subtotal	decimal	15	Total harga per item (Qty x Harga)

Table 13. Tabel Detail Transaksi 3NF

- a. Kamus Basis Data
- b. Nama Database : Penjualan_Indomaret
- c. Nama Table : Tabel_Pembayaran
- d. Fungsi : Menyimpan rincian finansial akhir, potongan harga, dan pajak (PPN) untuk memenuhi kaidah 3NF.

Nama Field	Tipe Data	Panjang Karakter	Keterangan
id_pembayaran	Varchar	10	Primary Key
id_transaksi	Varchar	20	Foreign Key
voucher	Decimal	15	Nilai potongan harga dari voucher
ppn	Decimal	15	Nilai Pajak Pertambahan Nilai (PPN)
total_bayar	Decimal	15	Total akhir yang dibayarkan pelanggan

Table 14 Tabel Pembayaran 3NF

E. Hubungan antar Entitas

Entitas Asal	Entitas Tujuan	Kardinalitas	Penjelasan
Tabel_Kasir	Tabel_Transaksi	1 : N	Satu kasir dapat melayani banyak transaksi, namun satu transaksi hanya dilayani oleh satu kasir.
Tabel_Member	Tabel_Transaksi	1 : N	Satu member dapat melakukan banyak transaksi, namun satu transaksi hanya tercatat untuk satu member/nonmember
Tabel_Transaksi	Tabel_Detail_Transaksi	1 : N	Satu transaksi dapat memiliki banyak rincian barang, tetapi satu baris detail hanya merujuk pada satu nomor transaksi.

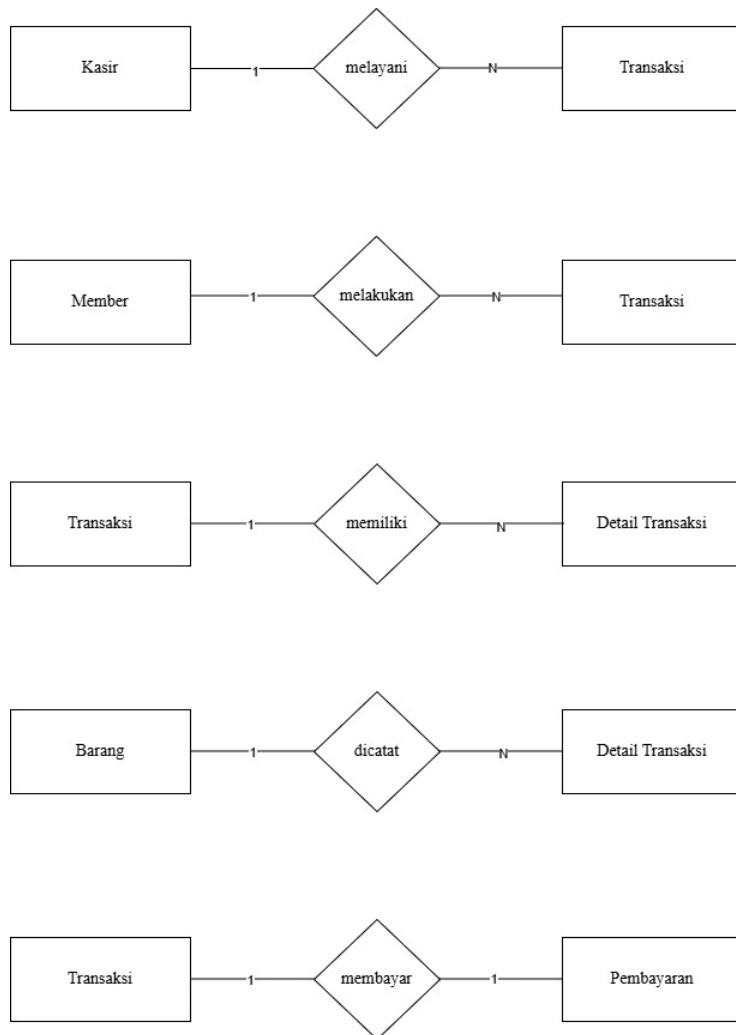
Entitas Asal	Entitas Tujuan	Kardinalitas	Penjelasan
Tabel Barang	Tabel Detail Transaksi	1 : N	Satu jenis barang dapat muncul di banyak detail transaksi yang berbeda.
Tabel Transaksi	Tabel Pembayaran	1 : 1	Satu transaksi hanya memiliki satu catatan rincian pembayaran (Total, PPN, Voucher).

Table 15. Hubungan antar Entitas

Sistem ini dirancang untuk mengelola data operasional di sebuah toko atau ritel, di mana Tabel Transaksi menjadi pusat dari aktivitas utama. Setiap transaksi yang terjadi dicatat oleh seorang Kasir dan dilakukan oleh seorang Member (atau non-member), di mana satu kasir maupun satu member dapat terlibat dalam banyak transaksi yang berbeda (hubungan *One-to-Many*).

Untuk merinci apa saja yang dibeli dalam sebuah transaksi, data dipecah ke dalam Tabel Detail Transaksi. Satu nomor transaksi dapat mencakup banyak item barang, dan setiap jenis Barang dapat muncul di berbagai detail transaksi yang berbeda. Terakhir, setiap transaksi yang telah selesai akan memiliki satu catatan pembayaran yang spesifik di Tabel Pembayaran, yang mencakup rincian finansial seperti total harga, PPN, dan penggunaan voucher melalui hubungan *One-to-One*.

F. Drajar Kardinalitas



Gambar 2. Drajar kardinalitas

Derajat kardinalitas menggambarkan hubungan kuantitas data antar entitas dalam basis data Penjualan_Indomaret. Penentuan kardinalitas ini berfungsi untuk memastikan integritas relasi antar tabel sesuai dengan aturan bisnis retail. Berikut adalah penjelasan rincinya:

a. Kasir Melayani Transaksi (1:N)

Satu orang kasir dapat melayani banyak transaksi di waktu yang berbeda, namun setiap satu nomor transaksi hanya diproses oleh satu orang kasir.

b. Member Melakukan Transaksi (1:N)

Seorang pelanggan atau member dapat melakukan banyak kali transaksi pembelian, sementara satu nota transaksi hanya mencatat identitas dari satu member.

c. Transaksi Memiliki Detail Transaksi (1:N)

Satu nomor transaksi dapat terdiri dari banyak rincian barang yang dibeli (beberapa item), tetapi setiap baris rincian item hanya merujuk pada satu nomor transaksi induk.

d. Barang Dicatat dalam Detail Transaksi (1:N)

Satu jenis produk dapat muncul atau dicatat dalam banyak rincian transaksi milik pelanggan yang berbeda, namun setiap baris detail barang hanya mewakili satu jenis produk tertentu.

e. Transaksi Membayar Pembayaran (1:1)

Satu nomor transaksi hanya memiliki tepat satu catatan rincian pembayaran (yang mencakup total bayar, PPN, dan voucher), dan sebaliknya, satu catatan pembayaran hanya valid untuk satu nomor transaksi tunggal.

G. Relasi antar Tabel

Relasi antar tabel pada sistem basis data Penjualan Indomaret dirancang untuk menghubungkan data barang, kasir, member, transaksi, rincian belanja, dan pembayaran agar tersimpan secara terstruktur dan saling berkaitan. Setiap tabel memiliki peran yang berbeda, namun saling terhubung melalui *primary key* (PK) dan *foreign key* (FK) guna menjaga integritas data.

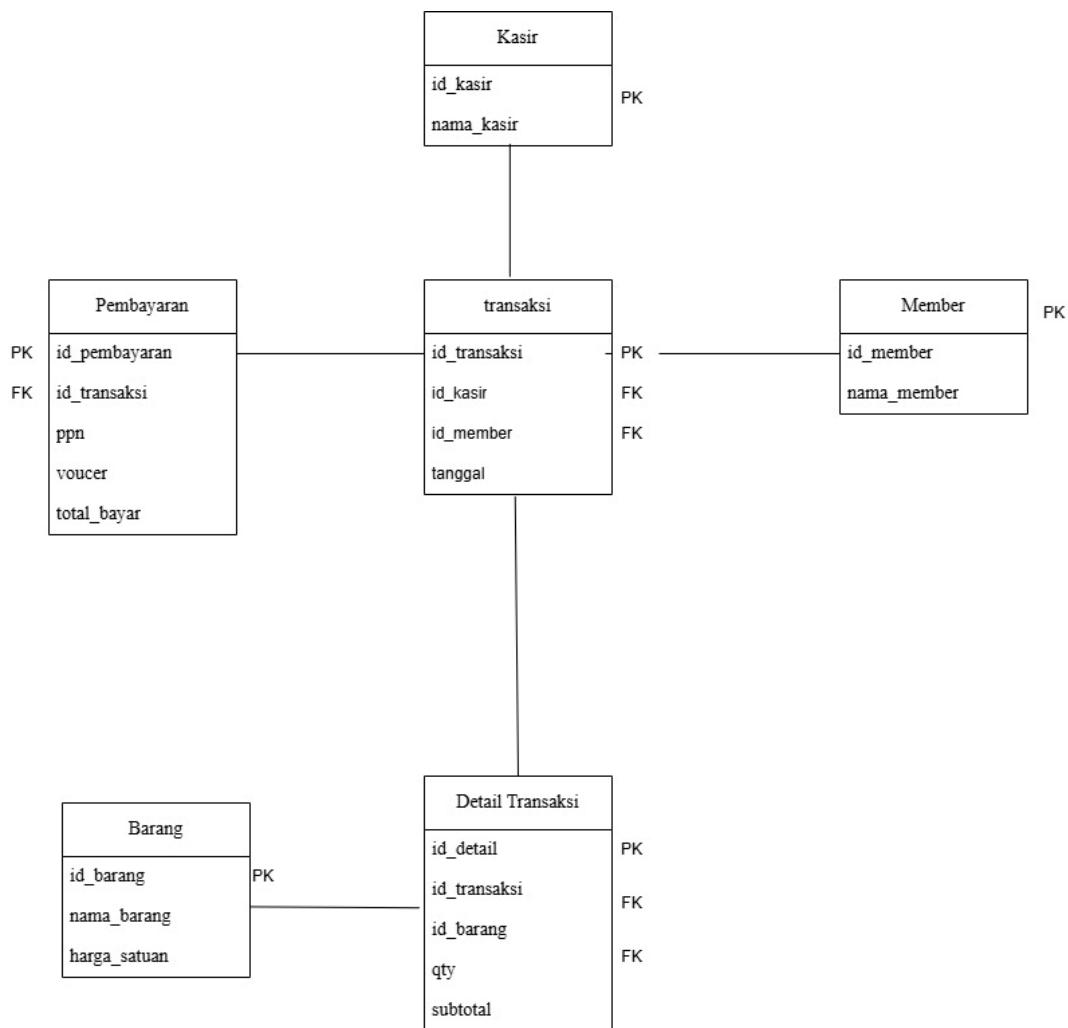
Tabel Kasir terhubung dengan Tabel Transaksi melalui *id_kasir*. Relasi ini menunjukkan bahwa satu kasir dapat melayani banyak transaksi (1:N), sedangkan setiap transaksi hanya dilayani oleh satu kasir pada waktu tertentu.

Tabel Member terhubung dengan Tabel Transaksi melalui *id_member*. Relasi ini menunjukkan bahwa satu member dapat melakukan banyak transaksi (1:N), sementara setiap nota transaksi hanya mencatat satu identitas member (Poinku) atau pelanggan yang bersangkutan.

Tabel Transaksi terhubung dengan Tabel Detail_Transaksi melalui *id_transaksi*. Relasi ini memungkinkan satu nomor transaksi untuk memiliki banyak rincian barang yang dibeli, namun setiap baris detail hanya merujuk pada satu nomor nota induk.

Tabel Barang terhubung dengan Tabel Detail_Transaksi melalui id_barang. Relasi ini menunjukkan bahwa satu jenis produk dapat muncul di banyak transaksi berbeda, sementara setiap baris detail mencatat secara spesifik produk apa yang dibeli oleh pelanggan.

Tabel Transaksi juga terhubung dengan Tabel Pembayaran melalui id_transaksi. Relasi ini bersifat satu-ke-satu (1:1), di mana satu nomor transaksi memiliki tepat satu rincian perhitungan finansial yang mencakup nilai voucher, PPN, dan total bayar guna memenuhi kaidah normalisasi tahap ketiga (3NF).

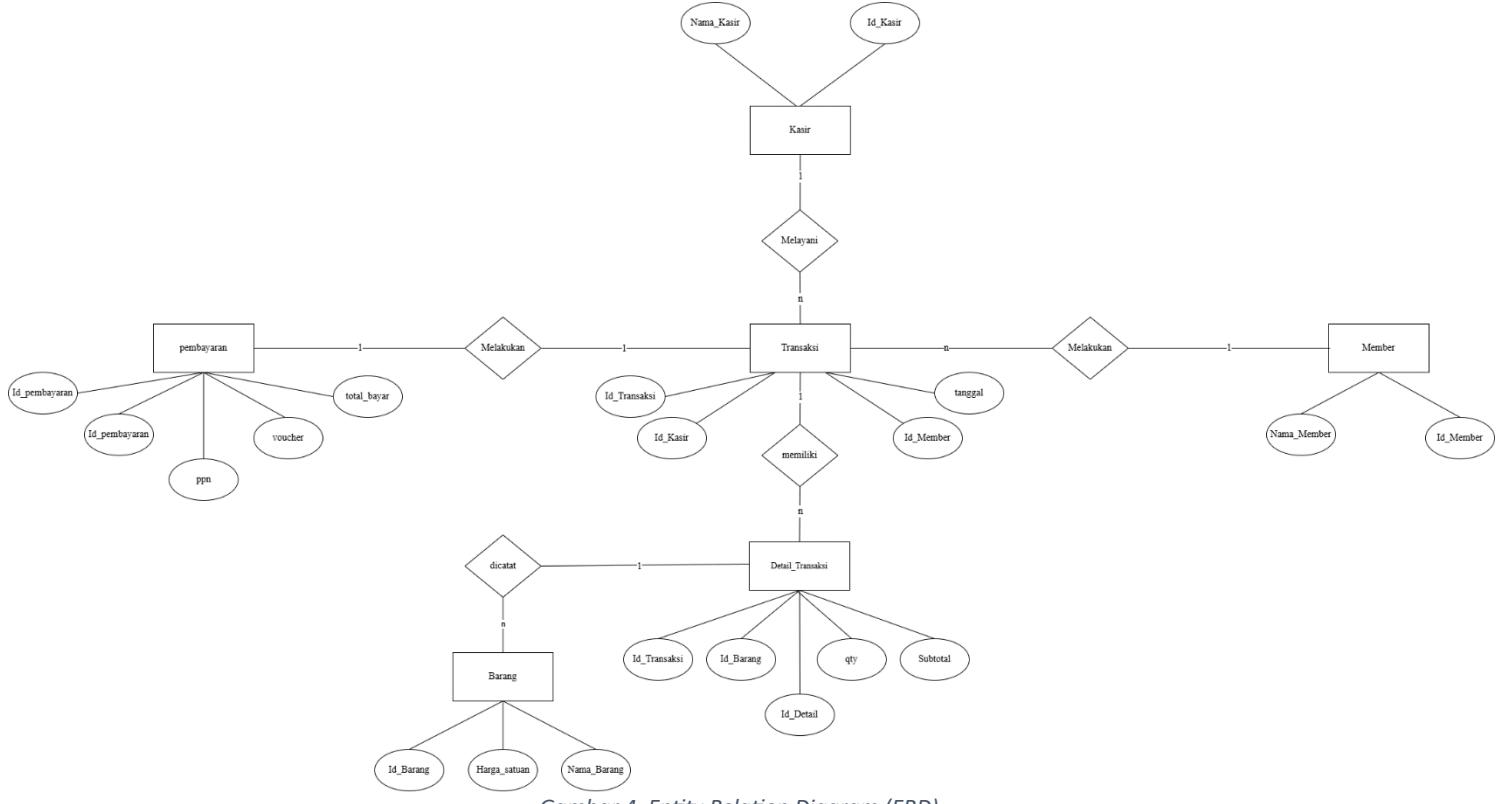


Gambar 3. Relasi antar Tabel

Dengan adanya relasi tersebut, setiap data transaksi dapat diketahui secara jelas siapa membernya, siapa kasir yang melayani, barang apa saja yang dibeli, hingga rincian pajaknya. Relasi ini menjaga konsistensi data, menghindari pengulangan informasi yang tidak perlu, serta memudahkan proses pengolahan laporan penjualan secara akurat.

H. ERD (Entity Relation Diagram)

ERD (Entity Relationship Diagram) ERD merupakan representasi visual dari struktur database Penjualan Indomaret yang menunjukkan hubungan logis antar entitas. Diagram ini menggunakan notasi relasi untuk menggambarkan bagaimana data kasir, member, dan barang saling berinteraksi melalui tabel transaksi.



Gambar 4. Entity Relation Diagram (ERD)

Dalam ERD ini, Tabel_Transaksi bertindak sebagai pusat informasi yang menghubungkan siapa yang melayani (Kasir), siapa yang membeli (Member), apa yang dibeli (Detail_Transaksi), dan bagaimana penyelesaian finansialnya (Pembayaran). Penggunaan relasi *One-to-Many* (1:N) mendominasi struktur ini untuk memastikan efisiensi data, sementara relasi *One-to-One* (1:1) pada tabel pembayaran digunakan untuk menjaga konsistensi nilai pajak dan total bayar sesuai dengan standar normalisasi tahap ketiga (3NF).

I. Entitas dan atribut utama

Entitas merupakan objek unik dalam sistem yang datanya perlu disimpan, sedangkan atribut adalah karakteristik atau detail informasi yang melekat pada setiap entitas tersebut. Berdasarkan hasil normalisasi hingga 3NF, berikut adalah rincian entitas dan atribut utamanya:

1. Entitas Barang : Menyimpan data master produk.

Atribut: id_barang (PK), nama_barang, harga_satuan.

2. Entitas Kasir : Menyimpan data petugas yang melayani transaksi.

Atribut: id_kasir (PK), nama_kasir.

3. Entitas Member : Menyimpan informasi pelanggan yang terdaftar dalam program loyalitas.

Atribut: id_member (PK), nama_member.

4. Entitas Transaksi : Berfungsi sebagai catatan utama (header) setiap nota belanja.

Atribut: id_transaksi (PK), tanggal, id_kasir (FK), id_member (FK).

5. Entitas Detail_Transaksi: Menyimpan rincian setiap item barang yang dibeli dalam satu transaksi.

Atribut: id_detail (PK), id_transaksi (FK), id_barang (FK), qty, subtotal.

6. Entitas Pembayaran : Menyimpan informasi finansial akhir dari transaksi.

Atribut: id_pembayaran (PK), id_transaksi (FK), voucher, ppn, total_bayar.

J. Penerapan ERD KE DBMS

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.

K. 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.

```

Query 1 × penjualan_indomaret
Limit to 300 rows
1 -- Membuat Database
2 CREATE DATABASE Penjualan_Indomaret;
3 USE Penjualan_Indomaret;
4
5 -- Membuat Tabel Kasir
6 CREATE TABLE Tabel_Kasir (
7     id_kasir VARCHAR(10) PRIMARY KEY,
8     nama_kasir VARCHAR(50) NOT NULL
9 );
10
11 -- Membuat Tabel Member
12 CREATE TABLE Tabel_Member (
13     id_member VARCHAR(15) PRIMARY KEY,
14     nama_member VARCHAR(50) NOT NULL
15 );
16
17 -- Membuat Tabel Barang
18 CREATE TABLE Tabel_Barang (
19     id_barang VARCHAR(10) PRIMARY KEY,
20     nama_barang VARCHAR(50) NOT NULL,
21     harga_satuan DECIMAL(15, 2) NOT NULL
22 );
23

```

Gambar 6

```

22 );
23
24 -- Membuat Tabel Transaksi (Header)
25 CREATE TABLE Tabel_Transaksi (
26     id_transaksi VARCHAR(20) PRIMARY KEY,
27     tanggal DATETIME NOT NULL,
28     id_kasir VARCHAR(10),
29     id_member VARCHAR(15),
30     FOREIGN KEY (id_kasir) REFERENCES Tabel_Kasir(id_kasir),
31     FOREIGN KEY (id_member) REFERENCES Tabel_Member(id_member)
32 );
33
34 -- Membuat Tabel Detail_Transaksi
35 CREATE TABLE Tabel_Detail_Transaksi (
36     id_detail INT(11) AUTO_INCREMENT PRIMARY KEY,
37     id_transaksi VARCHAR(20),
38     id_barang VARCHAR(10),
39     qty INT(11) NOT NULL,
40     subtotal DECIMAL(15, 2) NOT NULL,
41     FOREIGN KEY (id_transaksi) REFERENCES Tabel_Transaksi(id_transaksi),
42     FOREIGN KEY (id_barang) REFERENCES Tabel_Barang(id_barang)
43 );
44

```

Gambar 5

```

44
45 -- Membuat Tabel Pembayaran (Relasi 1:1 sesuai 3NF)
46 • CREATE TABLE Tabel_Pembayaran (
47     id_pembayaran VARCHAR(10) PRIMARY KEY,
48     id_transaksi VARCHAR(20) UNIQUE,
49     voucher DECIMAL(15, 2),
50     ppn DECIMAL(15, 2),
51     total_bayar DECIMAL(15, 2),
52     FOREIGN KEY (id_transaksi) REFERENCES Tabel_Transaksi(id_transaksi)
53 );
54

```

Gambar 7

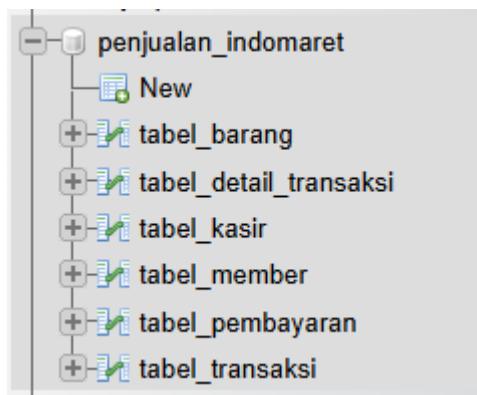
Gambar 5, 6, 7 merupakan

Tahap implementasi dilakukan dengan menerjemahkan rancangan ERD ke dalam DBMS MySQL menggunakan perintah SQL. Pembuatan tabel dilakukan secara berurutan, dimulai dari tabel master (Barang, Kasir, Member) kemudian dilanjutkan dengan tabel transaksi. Penggunaan constraint FOREIGN KEY memastikan integritas referensial terjaga, sehingga tidak ada data transaksi yang merujuk pada kasir atau barang yang tidak terdaftar.

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

Action	Time	Action	Message	Duration / Fetch
84	14:06:01	CREATE DATABASE Perjualan_Indomaret	1 row(s) affected	0.016 sec
85	14:06:05	USE Perjualan_Indomaret	0 row(s) affected	0.000 sec
86	14:06:09	CREATE TABLE Tabel_Kasir (id_kasir VARCHAR(10) PRIMARY KEY, nama_kasir VARCHAR(50) NOT NULL)	0 row(s) affected	0.015 sec
87	14:06:18	CREATE TABLE Tabel_Member (id_member VARCHAR(15) PRIMARY KEY, nama_member VARCHAR(50) NOT NULL)	0 row(s) affected	0.016 sec
88	14:06:22	CREATE TABLE Tabel_Barang (id_barang VARCHAR(10) PRIMARY KEY, nama_barang VARCHAR(50) NOT NULL)	0 row(s) affected	0.016 sec
89	14:06:30	CREATE TABLE Tabel_Transaksi (id_transaksi VARCHAR(20) PRIMARY KEY, tanggal DATETIME NOT NULL)	0 row(s) affected	0.015 sec
90	14:06:34	CREATE TABLE Tabel_Detail_Transaksi (id_detail INT(11) AUTO_INCREMENT PRIMARY KEY, id_transaksi VARCHAR(20) NOT NULL, id_barang VARCHAR(10) NOT NULL, jumlah INT(11) NOT NULL)	0 row(s) affected	0.016 sec
91	14:06:38	CREATE TABLE Tabel_Pembayaran (id_pembayaran VARCHAR(10) PRIMARY KEY, id_transaksi VARCHAR(20) NOT NULL, voucher DECIMAL(15, 2), ppn DECIMAL(15, 2), total_bayar DECIMAL(15, 2))	0 row(s) affected	0.016 sec

Gambar 8 Hasil running Workbench



Gambar 9 Otomatis ke MyPhpAdmin 1

b. DML (Data Manipulation Language) dan TCL (Transaction Control Language)

DML (Data Manipulation Language) adalah kumpulan perintah SQL yang digunakan untuk memanipulasi atau mengelola data di dalam tabel yang telah dibuat sebelumnya. Jika DDL digunakan untuk membangun "rumah" (struktur tabel), maka DML digunakan untuk mengisi "penghuni" (isi data) ke dalam rumah tersebut.

Dalam implementasi database Penjualan_Indomaret, perintah DML yang digunakan adalah INSERT INTO. Perintah ini berfungsi untuk memasukkan data rekaman (*record*) baru ke dalam masing-masing tabel sesuai dengan kolom yang tersedia.

TCL Transaction Control language adalah kategori perintah SQL yang digunakan untuk mengelola transaksi di dalam basis data agar data tetap aman, valid, dan konsisten. Perintah ini menjamin bahwa setiap proses (seperti penjualan) diproses secara benar meskipun terjadi kegagalan sistem. Komponen utamanya adalah:

- a. COMMIT : Digunakan untuk menyimpan semua perubahan data secara permanen ke dalam basis data setelah serangkaian perintah berhasil dijalankan.
- b. ROLLBACK: Digunakan untuk membatalkan perubahan dan mengembalikan data ke kondisi sebelumnya jika terjadi kesalahan atau kegagalan dalam proses transaksi.

TCL memastikan integritas data tetap terjaga, sehingga tidak ada data yang tersimpan "setengah jalan", misalnya ketika total belanja sudah tercatat namun rincian barangnya gagal tersimpan.

```

Query 1 penjualan_indomaret
Limit to 300 rows
55 -- Memulai Transaksi
56 • START TRANSACTION;
57
58 -- Menambah Data Master
59 • INSERT INTO Tabel_Kasir VALUES ('K01', 'PUGUH');
60 • INSERT INTO Tabel_Member VALUES ('...3207', 'Wedyta');
61 • INSERT INTO Tabel_Barang VALUES ('B01', 'STARBUCKS LATTE 220', 15900);
62 • INSERT INTO Tabel_Barang VALUES ('B02', 'AQUA AIR MINERAL 600', 3800);
63 • INSERT INTO Tabel_Barang VALUES ('B03', 'SR/RT SBK DUO COK 72', 8000);
64
65 -- Menambah Data Transaksi
66 • INSERT INTO Tabel_Transaksi VALUES ('TON8-36330', '2026-01-04 04:12:00', 'K01', '...3207');
67
68 -- Menambah Rincian Barang (Detail)
69 • INSERT INTO Tabel_Detail_Transaksi (id_transaksi, id_barang, qty, subtotal) VALUES
70 ('TON8-36330', 'B01', 1, 15900),
71 ('TON8-36330', 'B02', 2, 7600),
72 ('TON8-36330', 'B03', 1, 8000);
73
74 -- Menambah Data Pembayaran Akhir
75 • INSERT INTO Tabel_Pembayaran VALUES ('P01', 'TON8-36330', 1700, 3122, 29800);
76
77 -- Menyimpan Perubahan Secara Permanen
78 • COMMIT;
79

```

Gambar 10. Implementasi DML & TCL

92	14:06:42	START TRANSACTION	0 row(s) affected	0.000 sec
93	14:09:20	INSERT INTO Tabel_Kasir VALUES ('K01', 'PUGUH')	1 row(s) affected	0.016 sec
94	14:09:26	INSERT INTO Tabel_Member VALUES ('...3207', 'Wedyta')	1 row(s) affected	0.016 sec
95	14:09:31	INSERT INTO Tabel_Barang VALUES ('B01', 'STARBUCKS LATTE 220', 15900)	1 row(s) affected	0.000 sec
96	14:09:38	INSERT INTO Tabel_Barang VALUES ('B02', 'AQUA AIR MINERAL 600', 3800)	1 row(s) affected	0.000 sec
97	14:09:43	INSERT INTO Tabel_Barang VALUES ('B03', 'SR/RT SBK DUO COK 72', 8000)	1 row(s) affected	0.000 sec
98	14:10:08	INSERT INTO Tabel_Transaksi VALUES ('TON8-36330', '2026-01-04 04:12:00', 'K01', '...3207')	1 row(s) affected	0.000 sec
99	14:10:26	INSERT INTO Tabel_Detail_Transaksi (id_transaksi, id_barang, qty, subtotal) VALUES ('TON8-36330', 'B01', 1, 15900), ('TON8-36330', 'B02', 2, 7600), ('TON8-36330', 'B03', 1, 8000)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
100	14:10:33	INSERT INTO Tabel_Pembayaran VALUES ('P01', 'TON8-36330', 1700, 3122, 29800)	1 row(s) affected	0.016 sec
101	14:10:37	COMMIT;	0 row(s) affected	0.000 sec

Gambar 11. Running Hasil DML & TCL

c. 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.

```

78 • COMMIT;
79
80 • SELECT
81     t.id_transaksi,
82     t.tanggal,
83     k.nama_kasir,
84     m.nama_member,
85     b.nama_barang,
86     dt.qty,
87     b.harga_satuan,
88     dt.subtotal,
89     p.voucher,
90     p.ppn,
91     p.total_bayar
92 FROM Tabel_Transaksi t
93 JOIN Tabel_Kasir k ON t.id_kasir = k.id_kasir
94 JOIN Tabel_Member m ON t.id_member = m.id_member
95 JOIN Tabel_Detail_Transaksi dt ON t.id_transaksi = dt.id_transaksi
96 JOIN Tabel_Barang b ON dt.id_barang = b.id_barang
97 JOIN Tabel_Pembayaran p ON t.id_transaksi = p.id_transaksi;
98
99 • SELECT b.nama_barang, SUM(dt.qty) AS total_terjual
  FROM Tabel_Barang b

```

Gambar 12. Kode Query Join

	id_transaksi	tanggal	nama_kasir	nama_member	nama_barang	qty	harga_satuan	subtotal	voucher	ppn	total_bayar
▶	TON8-36330	2026-01-04 04:12:00	PUGUH	Wedya	STARBUCKS LATTE 220	1	15900.00	15900.00	1700.00	3122.00	29800.00
	TON8-36330	2026-01-04 04:12:00	PUGUH	Wedya	AQUA AIR MINERAL 600	2	3800.00	7600.00	1700.00	3122.00	29800.00
	TON8-36330	2026-01-04 04:12:00	PUGUH	Wedya	SR/RT SBK DUO COK 72	1	8000.00	8000.00	1700.00	3122.00	29800.00

Gambar 13. Output dari Query Join di Running

b) Group By, Agresi & Having

Dalam pengolahan data transaksi retail, Group By berfungsi untuk mengelompokkan sekumpulan data yang memiliki nilai identik ke dalam satu ringkasan, seperti mengelompokkan ribuan baris penjualan berdasarkan nama produk atau ID kasir. Proses pengelompokan ini kemudian diikuti dengan penggunaan Fungsi Agregasi, yaitu operasi matematika seperti *SUM* untuk menjumlahkan total pendapatan, *COUNT* untuk menghitung frekuensi transaksi, atau *AVG* untuk mencari rata-rata belanja pelanggan.

Terakhir, terdapat klausa HAVING yang berperan sebagai penyaring hasil setelah data dikelompokkan berbeda dengan klausa *WHERE* yang menyaring data mentah, *HAVING* secara spesifik digunakan untuk menyaring hasil kalkulasi agregasi, misalnya hanya menampilkan produk yang total penjualannya telah melampaui target tertentu atau menampilkan kasir yang memiliki performa di atas rata-rata.

Query 1 x penjualan_indomaret

```

99 • SELECT
100     b.id_barang,
101     b.nama_barang,
102     SUM(dt.qty) AS total_terjual,
103     SUM(dt.subtotal) AS total_pendapatan
104   FROM Tabel Barang b
105   JOIN Tabel_Detail_Transaksi dt ON b.id_barang = dt.id_barang
106 GROUP BY b.id_barang, b.nama_barang
107 HAVING SUM(dt.qty) >= 1
108 ORDER BY total_terjual DESC;

```

Kode Query by, Agregasi & Having

	id_barang	nama_barang	total_terjual	total_pendapatan
▶	B02	AQUA AIR MINERAL 600	2	7600.00
▶	B01	STARBUCKS LATTE 220	1	159.7600.00
▶	B03	SR/RT SBK DUO COK 72	1	8000.00

Output Query By, Agresi dan Having

d. Query subquery

Merupakan teknik kueri bersarang di mana sebuah instruksi SELECT ditempatkan di dalam kueri utama. Dalam implementasi sistem ini, subquery berperan penting untuk memfilter data secara dinamis berdasarkan kalkulasi spesifik, seperti mengidentifikasi member yang akumulasi belanjanya melampaui ambang batas rata-rata. Pemanfaatan subquery ini memberikan fleksibilitas tinggi dalam pengolahan data, sehingga informasi yang dihasilkan menjadi lebih akurat, relevan, dan mampu menjawab kebutuhan analisis manajerial yang kompleks.

Query 1 x penjualan_indomaret

```

108 ORDER BY total_terjual DESC;
109
110 -- Query Subquery
111 -- Menampilkan ID dan Nama Member yang belanjanya di atas rata-rata total bayar
112 • SELECT
113     id_member,
114     nama_member
115   FROM Tabel Member
116 WHERE id_member IN (
117     SELECT t.id_member
118     FROM Tabel_Transaksi t
119     JOIN Tabel_Pembayaran p ON t.id_transaksi = p.id_transaksi
120     WHERE p.total_bayar > 20000
121 );
122

```

Kode Query Subquery

	id_member	nama_member
▶	...3207	Wedya
*	NULL	NULL

Output Dari Query Subquery

BAB IV

PENUTUPAN

A. Hasil Pengujian Query

Berdasarkan serangkaian uji coba yang dilakukan melalui MySQL Workbench, seluruh instruksi query yang telah dirancang mampu berjalan dengan akurasi tinggi. Penggunaan kueri *Join* berhasil mengintegrasikan data dari berbagai tabel sehingga informasi transaksi pada nota fisik Indomaret dapat ditampilkan kembali secara digital secara utuh. Selain itu, implementasi fungsi agregasi, *Having*, dan *Subquery* terbukti efektif dalam menyaring data yang kompleks, seperti mengidentifikasi pelanggan prioritas atau menghitung total pendapatan harian secara otomatis dan presisi.

B. Ringkasan Hasil Uji

Ringkasan dari hasil pengujian menunjukkan bahwa sistem basis data ini telah mencapai standar integritas data yang optimal melalui penerapan relasi *Primary Key* dan *Foreign Key* yang konsisten. Dengan struktur yang telah ternormalisasi hingga tahap 3NF, database mampu meminimalkan risiko redundansi data dan mempercepat waktu respon saat pengambilan informasi. Hal ini membuktikan bahwa rancangan teknis yang dibuat telah mampu merepresentasikan kebutuhan bisnis retail secara efisien dan sistematis.

C. Kendala dan Perbaikan

Selama proses implementasi, ditemukan kendala utama berupa *constraint violation* saat pengisian data, di mana sistem menolak penginputan rincian transaksi sebelum data master barang terdaftar. Perbaikan dilakukan dengan menyusun urutan eksekusi DML secara kronologis, yakni mendahulukan data master sebelum data transaksi. Kendala lain terkait ambiguitas nama kolom pada kueri *Join* yang rumit berhasil diatasi dengan menerapkan teknik *aliasing* pada setiap tabel untuk memperjelas referensi atribut.

D. Kesimpulan

Dapat disimpulkan bahwa perancangan basis data penjualan retail ini telah berhasil menjawab tantangan pengolahan data transaksi yang cepat dan terintegrasi. Dengan mengikuti metodologi perancangan yang benar, mulai dari analisis normalisasi hingga implementasi query kompleks, sistem ini mampu menjaga konsistensi data dan menyediakan informasi

manajerial yang akurat. Basis data ini tidak hanya berfungsi sebagai tempat penyimpanan, tetapi juga sebagai alat bantu dalam pengambilan keputusan bisnis yang efektif.

E. Saran Pengembangan

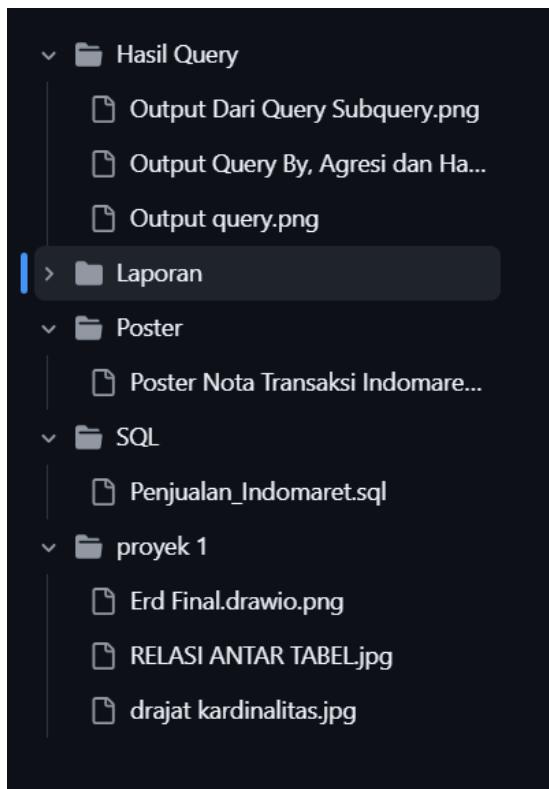
Untuk pengembangan di masa depan, sistem ini disarankan untuk dilengkapi dengan fitur *Trigger* guna melakukan pemutakhiran stok barang secara otomatis setiap kali terjadi penjualan. Selain itu, penggunaan *Stored Procedures* dan *Views* sangat dianjurkan untuk memudahkan pembuatan laporan rutin tanpa harus menulis ulang kueri yang panjang. Penambahan sistem keamanan berbasis hak akses juga perlu dipertimbangkan untuk melindungi data sensitif seperti harga pokok barang dan informasi pribadi pelanggan.

Lampiran

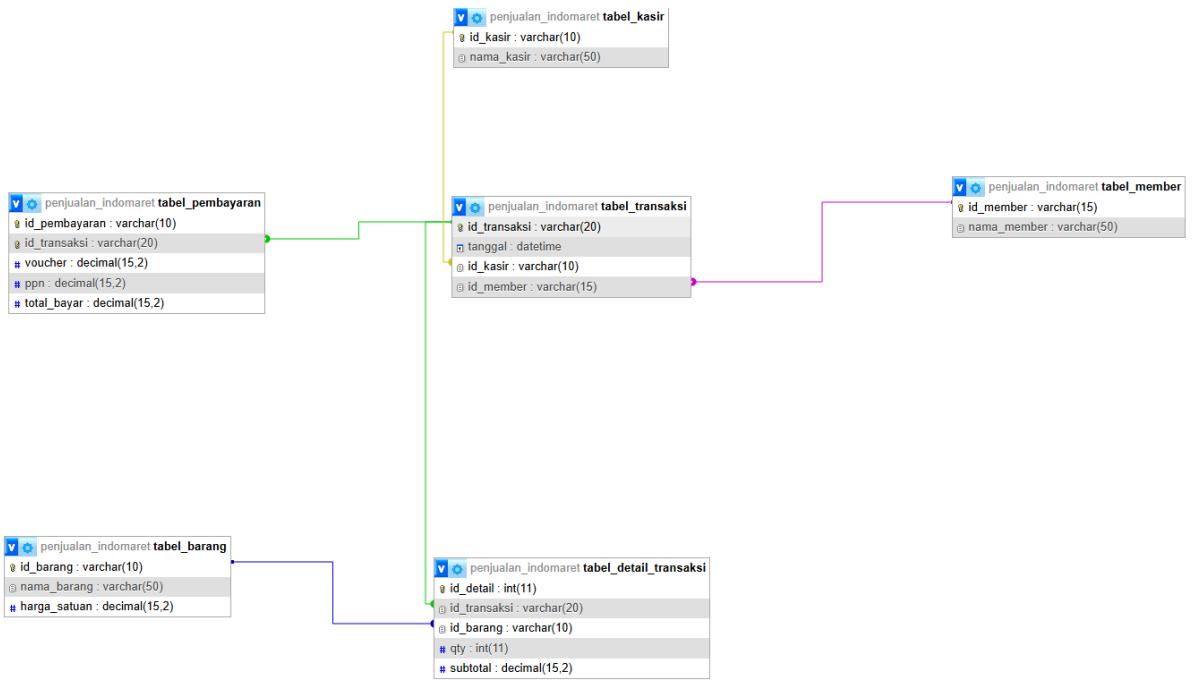
A. Tautan Repository

<https://github.com/nonawedya/Data-penjualan-berdasarkan-studi-kasus-nota-transaksi-indomaret.git>

B. Struktur Folder Repository



C. ERD Final



D. Hasil Eksekusi Query

1. Query Join

Query 1 x penjualan_indomaret

```

78 •   COMMIT;
79
80 •   SELECT
81     t.id_transaksi,
82     t.tanggal,
83     k.nama_kasir,
84     m.nama_member,
85     b.nama_barang,
86     dt.qty,
87     b.harga_satuan,
88     dt.subtotal,
89     p.voucher,
90     p.ppn,
91     p.total_bayar
92     FROM Tabel_Transaksi t
93     JOIN Tabel_Kasir k ON t.id_kasir = k.id_kasir
94     JOIN Tabel_Member m ON t.id_member = m.id_member
95     JOIN Tabel_Detail_Transaksi dt ON t.id_transaksi = dt.id_transaksi
96     JOIN Tabel_Barang b ON dt.id_barang = b.id_barang
97     JOIN Tabel_Pembayaran p ON t.id_transaksi = p.id_transaksi;
98
99 •   SELECT b.nama_barang, SUM(dt.qty) AS total_terjual
100    FROM Tabel_Barang b
  
```

Kode Query Join

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

	id_transaksi	tanggal	nama_kasir	nama_member	nama_barang	qty	harga_satuan	subtotal	voucher	ppn	total_bayar
▶	TQN8-36330	2026-01-04 04:12:00	PUGUH	Wedyo	STARBUCKS LATTE 220	1	15900.00	15900.00	1700.00	3122.00	29800.00
	TQN8-36330	2026-01-04 04:12:00	PUGUH	Wedyo	AQUA AIR MINERAL 600	2	3800.00	7600.00	1700.00	3122.00	29800.00
	TQN8-36330	2026-01-04 04:12:00	PUGUH	Wedyo	SR/RT SBK DUO COK 72	1	8000.00	8000.00	1700.00	3122.00	29800.00

Result Grid Form Editor Field Types

Output Query Join

2. Group By, Agresi & Having

Query 1 × penjualan_indomaret

```

99 •   SELECT
100      b.id_barang,
101      b.nama_barang,
102      SUM(dt.qty) AS total_terjual,
103      SUM(dt.subtotal) AS total_pendapatan
104  FROM Tabel_Barang b
105  JOIN Tabel_Detail_Transaksi dt ON b.id_barang = dt.id_barang
106  GROUP BY b.id_barang, b.nama_barang
107  HAVING SUM(dt.qty) >= 1
108  ORDER BY total_terjual DESC;

```

Kode Group By, Agresi & Having

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

	id_barang	nama_barang	total_terjual	total_pendapatan
▶	B02	AQUA AIR MINERAL 600	2	7600.00
	B01	STARBUCKS LATTE 220	1	15900.00
	B03	SR/RT SBK DUO COK 72	1	8000.00

Result Grid Form Editor

Output Group By, Agresi & Having

Query 1 × penjualan_indomaret

```

108  ORDER BY total_terjual DESC;
109
110  -- Query Subquery
111  -- Menampilkan ID dan Nama Member yang belanjanya di atas rata-rata total bayar
112 •   SELECT
113      id_member,
114      nama_member
115  FROM Tabel_Member
116  WHERE id_member IN (
117      SELECT t.id_member
118      FROM Tabel_Transaksi t
119      JOIN Tabel_Pembayaran p ON t.id_transaksi = p.id_transaksi
120      WHERE p.total_bayar > 20000
121  );
122

```

Kode Query Subquery

	id_member	nama_member
▶	3207	Wedyta
*	NULL	NULL

Output Query Subquery

3. Query Subquery

E. Script SQL lengkap

```

1 -- 1. Membuat Database
2 • CREATE DATABASE Penjualan_Indomaret;
3 • USE Penjualan_Indomaret;
4
5 -- 2. Membuat Tabel Kasir
6 • CREATE TABLE Tabel_Kasir (
7     id_kasir VARCHAR(10) PRIMARY KEY,
8     nama_kasir VARCHAR(50) NOT NULL
9 );
10
11 -- 3. Membuat Tabel Member
12 • CREATE TABLE Tabel_Member (
13     id_member VARCHAR(15) PRIMARY KEY,
14     nama_member VARCHAR(50)
15 );
16

```

```

16
17      -- 4. Membuat Tabel Barang
18 • CREATE TABLE Tabel_Barang (
19         id_barang VARCHAR(10) PRIMARY KEY,
20         nama_barang VARCHAR(100) NOT NULL,
21         harga_satuan DECIMAL(15, 2) NOT NULL
22     );
23
24      -- 5. Membuat Tabel Transaksi (Header)
25 • CREATE TABLE Tabel_Transaksi (
26         id_transaksi VARCHAR(20) PRIMARY KEY,
27         tanggal DATETIME NOT NULL,
28         id_kasir VARCHAR(10),
29         id_member VARCHAR(15),
30         CONSTRAINT fk_kasir FOREIGN KEY (id_kasir) REFERENCES Tabel_Kasir(id_kasir),
31         CONSTRAINT fk_member FOREIGN KEY (id_member) REFERENCES Tabel_Member(id_member)
32     );
33
34      -- 6. Membuat Tabel Detail Transaksi (Relasi M:N yang dipecah)
35 • CREATE TABLE Tabel_Detail_Transaksi (
36         id_detail INT AUTO_INCREMENT PRIMARY KEY,
37         id_transaksi VARCHAR(20),
38         id_barang VARCHAR(10),
39         qty INT NOT NULL,
40         subtotal DECIMAL(15, 2) NOT NULL,
41         CONSTRAINT fk_transaksi_detail FOREIGN KEY (id_transaksi) REFERENCES Tabel_Transaksi(id_transaksi),
42         CONSTRAINT fk_barang_detail FOREIGN KEY (id_barang) REFERENCES Tabel_Barang(id_barang)
43     );
44
45      -- 7. Membuat Tabel Pembayaran (Relasi 1:1 untuk 3NF)
46 • CREATE TABLE Tabel_Pembayaran (
47         id_pembayaran INT AUTO_INCREMENT PRIMARY KEY,
48         id_transaksi VARCHAR(20) UNIQUE,
49         voucher DECIMAL(15, 2) DEFAULT 0,
50         ppn DECIMAL(15, 2) NOT NULL,
51         total_bayar DECIMAL(15, 2) NOT NULL,
52         CONSTRAINT fk_transaksi_bayar FOREIGN KEY (id_transaksi) REFERENCES Tabel_Transaksi(id_transaksi)
53     );

```

```

Query 1 penjualan_indomaret
Limit to 300 rows

55 -- Memulai Transaksi
56 • START TRANSACTION;
57
58 -- Menambah Data Master
59 • INSERT INTO Tabel_Kasir VALUES ('K01', 'PUGUH');
60 • INSERT INTO Tabel_Member VALUES ('...3207', 'Wedya');
61 • INSERT INTO Tabel_Barang VALUES ('B01', 'STARBUCKS LATTE 220', 15900);
62 • INSERT INTO Tabel_Barang VALUES ('B02', 'AQUA AIR MINERAL 600', 3800);
63 • INSERT INTO Tabel_Barang VALUES ('B03', 'SR/RT SBK DUO COK 72', 8000);
64
65 -- Menambah Data Transaksi
66 • INSERT INTO Tabel_Transaksi VALUES ('TON8-36330', '2026-01-04 04:12:00', 'K01', '...3207');
67
68 -- Menambah Rincian Barang (Detail)
69 • INSERT INTO Tabel_Detail_Transaksi (id_transaksi, id_barang, qty, subtotal) VALUES
70 ('TON8-36330', 'B01', 1, 15900),
71 ('TON8-36330', 'B02', 2, 7600),
72 ('TON8-36330', 'B03', 1, 8000);
73
74 -- Menambah Data Pembayaran Akhir
75 • INSERT INTO Tabel_Pembayaran VALUES ('P01', 'TON8-36330', 1700, 3122, 29800);
76
77 -- Menyimpan Perubahan Secara Permanen
78 • COMMIT;
79

80 • SELECT
81     t.id_transaksi,
82     t.tanggal,
83     k.nama_kasir,
84     m.nama_member, |
85     b.nama_barang,
86     dt.qty,
87     b.harga_satuan,
88     dt.subtotal,
89     p.voucher,
90     p.ppn,
91     p.total_bayar
92     FROM Tabel_Transaksi t
93     JOIN Tabel_Kasir k ON t.id_kasir = k.id_kasir
94     JOIN Tabel_Member m ON t.id_member = m.id_member
95     JOIN Tabel_Detail_Transaksi dt ON t.id_transaksi = dt.id_transaksi
96     JOIN Tabel_Barang b ON dt.id_barang = b.id_barang
97     JOIN Tabel_Pembayaran p ON t.id_transaksi = p.id_transaksi;
98

99 • SELECT
100    b.id_barang,
101    b.nama_barang,
102    SUM(dt.qty) AS total_terjual,
103    SUM(dt.subtotal) AS total_pendapatan
104    FROM Tabel_Barang b
105    JOIN Tabel_Detail_Transaksi dt ON b.id_barang = dt.id_barang
106    GROUP BY b.id_barang, b.nama_barang
107    HAVING SUM(dt.qty) >= 1
108    ORDER BY total_terjual DESC;
109

110 -- Query Subquery
111 -- Menampilkan ID dan Nama Member yang belanjanya di atas rata-rata total bayar
112 • SELECT
113     id_member,
114     nama_member
115     FROM Tabel_Member
116     WHERE id_member IN (
117         SELECT t.id_member
118         FROM Tabel_Transaksi t
119         JOIN Tabel_Pembayaran p ON t.id_transaksi = p.id_transaksi
120         WHERE p.total_bayar > 20000
121     );
122

```

F. Database (PHP MyAdmin)

Database dari Sisystem

Table	Action	Rows	Type	Collation	Size	Overhead
tabel_barang	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 Kib	-
tabel_detail_transaksi	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	48.0 Kib	-
tabel_kasir	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 Kib	-
tabel_member	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 Kib	-
tabel_pembayaran	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 Kib	-
tabel_transaksi	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	48.0 Kib	-
6 tables	Sum	10	InnoDB	utf8mb4_general_ci	176.0 Kib	0 B

[Check all](#) [With selected:](#)

Isi dari Id Barang

	Edit	Copy	Delete	id_barang	nama_barang	harga_satuan
<input type="checkbox"/>	Edit	Copy	Delete	B01	STARBUCKS LATTE 220	15900.00
<input type="checkbox"/>	Edit	Copy	Delete	B02	AQUA AIR MINERAL 600	3800.00
<input type="checkbox"/>	Edit	Copy	Delete	B03	SR/RT SBK DUO COK 72	8000.00

[Check all](#) [With selected:](#) [Edit](#) [Copy](#) [Delete](#) [Export](#)

[Show all](#) [Number of rows:](#) 25 [Filter rows:](#) [Search this table](#) [Sort by key:](#) None

Isi dari Tabel Detail Transaksi

	Edit	Copy	Delete	id_detail	id_transaksi	id_barang	qty	subtotal
<input type="checkbox"/>	Edit	Copy	Delete	1	TON8-36330	B01	1	15900.00
<input type="checkbox"/>	Edit	Copy	Delete	2	TON8-36330	B02	2	7600.00
<input type="checkbox"/>	Edit	Copy	Delete	3	TON8-36330	B03	1	8000.00

[Check all](#) [With selected:](#) [Edit](#) [Copy](#) [Delete](#) [Export](#)

[Show all](#) [Number of rows:](#) 25 [Filter rows:](#) [Search this table](#) [Sort by key:](#) None

Isi dari Tabel Kasir

	Edit	Copy	Delete	id_kasir	nama_kasir
<input type="checkbox"/>	Edit	Copy	Delete	K01	PUGUH

[Check all](#) [With selected:](#) [Edit](#) [Copy](#) [Delete](#) [Export](#)

[Show all](#) [Number of rows:](#) 25 [Filter rows:](#) [Search this table](#)

Isi Dari Tabel Member

	Edit	Copy	Delete	id_member	nama_member
<input type="checkbox"/>	Edit	Copy	Delete	...3207	Wedyta

[Check all](#) [With selected:](#) [Edit](#) [Copy](#) [Delete](#) [Export](#)

[Show all](#) [Number of rows:](#) 25 [Filter rows:](#) [Search this table](#)

Extra options								
		← →	▼	id_pembayaran	id_transaksi	voucher	ppn	total_bayar
		<input type="checkbox"/>	<input checked="" type="checkbox"/> Edit	<input checked="" type="checkbox"/> Copy	<input checked="" type="checkbox"/> Delete	P01	TON8-36330	1700.00 3122.00 29800.00
↑	<input type="checkbox"/>	Check all	With selected:	<input checked="" type="checkbox"/> Edit	<input checked="" type="checkbox"/> Copy	<input checked="" type="checkbox"/> Delete	<input checked="" type="checkbox"/> Export	
	<input type="checkbox"/>	Show all	Number of rows:	25	Filter rows:	Search this table		

Isi dari Tabel Pembayaran

Transaksi						
		Data Transaksi		Detail Transaksi		
Header		Detail	Detail	Detail	Detail	Detail
<input type="checkbox"/>	<input checked="" type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	TON8-36330	2026-01-04 04:12:00	K01
						...3207
<input type="checkbox"/>	<input type="checkbox"/> Check all	With selected:		<input checked="" type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete
					<input type="checkbox"/> Export	
<input type="checkbox"/> Show all		Number of rows:		25	Filter rows:	Search this table

Isi dari Tabel Transaksi

DAFTAR PUSAKA

Septian R. Perancangan LRS dan ERD dalam sistem transaksi Indomaret [Internet]. Petulai: Universitas Pat Petulai; 2024 [cited 2026 Jan 18]. Available from:
<https://id.scribd.com/document/814803536/Perancangan-LRS-Dan-Erd-Dalam-Sistem-Transaksi-Indomaret>

Adelia QV. Struktur database perusahaan dagang PT Indomarco Prismatama [Internet]. 2021 [cited 2026 Jan 18]. Available from: <https://id.scribd.com/document/574779588/Struktur-database>

Sitanggang TF. Analisis dan perancangan sistem informasi penjualan minimarket berintegrasi barcode reader menggunakan PHP, MySQL dan JQuery (Studi Kasus: Indomaret Jl. Wijaya Kusuma No. 30 Condong Catur Yogyakarta) [Internet]. Yogyakarta: Universitas AMIKOM Yogyakarta; 2012 [cited 2026 Jan 18]. Available from:

<https://eprints.amikom.ac.id/id/eprint/17576/>

Arimbi. Studi sistem informasi manajemen: Kasus Indomaret [Internet]. Jumbidter. 2024 [cited 2026 Jan 18]. Available from:
<https://ejurnal.arimbi.or.id/index.php/JUMBIDTER/article/download/204/342/1088>

Sumantri RBB, Mahardika F. Perancangan basis data sistem informasi penjualan barang (Studi Kasus: Minimarket “ABC” Sidareja). Jurnal Teknologi dan Bisnis. 2020 Jun 2;2(1):61-68. Available from: <http://e-journal.stmikkomputama.ac.id/index.php/jtb/article/view/16>

Rosihan A. Perancangan basis data (Kasus: Retail) [Internet]. 2022 [cited 2026 Jan 18]. Available from: <https://blog.rosihanari.net/perancangan-basis-data-kasus-retail/>

Prasetya WS. Perancangan model basis data relasional dengan metode database life cycle. Dalam: Seminar Nasional Informatika (SEMNASIF). Yogyakarta: UPN "Veteran" Yogyakarta; 2015. hlm. 91-98.

Sutanta E. Sistem basis data. Yogyakarta: Graha Ilmu; 2004.

Pamungkas CA. Pengantar dan implementasi basis data. Sleman: Deepublish; 2017.

Dybka P. Crow's foot notation [Internet]. Vertabelo; 2016 [cited 2026 Jan 18]. Available from: <https://vertabelo.com/blog/crow-s-foot-notation/>