

Apotek Sehat

LAPORAN PROYEK AKHIR

COMP6852004 – Database Technology

KELAS BB20



Oleh:

2602181875 – Putri Febiyani

2602059670 – Regina Celine Adiwinata

2602155574 – Nazhira Dewi Aqmarina

Semester Ganjil 2023/2024

MALANG

Daftar Isi

BAB I.....	3
PENDAHULUAN	3
1. 1 Latar Belakang	3
1.2 Rumusan Masalah	3
1. 3 Tujuan	4
1.4 Manfaat	4
BAB II.....	5
LANDASAN TEORI.....	5
2.1 Database.....	5
2.2 Database Management System	5
2.3 Relational DBMS	5
2.4 Entity Relational Diagram	5
2.5 Data Definition Language.....	6
2.6 Data Manipulation Language.....	6
BAB III	7
HASIL & PEMBAHASAN	7
3.1 Entity Relational Diagram	7
3.2 List of Attributes	7
3.3 Database.....	9
BAB IV	41
KESIMPULAN.....	41
REFERENSI	41

BAB I

PENDAHULUAN

1. 1 Latar Belakang

Pertumbuhan industri farmasi dan pelayanan kesehatan memperlihatkan peningkatan signifikan dalam permintaan dan penawaran obat-obatan. Apotek, sebagai elemen integral dalam sistem kesehatan, menjadi perantara yang menghubungkan kebutuhan masyarakat dengan pasokan obat. Dalam menghadapi dinamika ini, penting untuk memperkuat manajemen pembelian obat di apotek.

Latar belakang penelitian ini didorong oleh kesadaran akan kompleksitas yang terlibat dalam manajemen stok obat dan transaksi pembelian di apotek. Dalam menghadapi tantangan seperti fluktuasi permintaan pelanggan, keberagaman produk, dan perubahan regulasi, sebuah sistem database yang terstruktur dapat memberikan solusi integral.

Apotek yang efisien dan responsif perlu mengadopsi teknologi informasi yang canggih untuk memberdayakan proses pembelian obat. Pencatatan manual yang rentan terhadap kesalahan dan kurangnya pemahaman yang mendalam tentang tren pembelian dapat memberikan dampak negatif terhadap efisiensi operasional dan pelayanan pelanggan.

1.2 Rumusan Masalah

Beberapa masalah krusial yang diidentifikasi meliputi:

1. Tidak optimalnya pengelolaan stok obat yang dapat mengakibatkan kekurangan atau kelebihan persediaan.
2. Keterbatasan dalam melacak dan menganalisis data pembelian obat untuk perencanaan kebutuhan masa depan.
3. Kehadiran lapisan birokrasi dan waktu yang dibutuhkan dalam proses pembelian yang dapat menghambat respons cepat terhadap perubahan kebutuhan pasar.

Dalam kerangka ini, rumusan masalah menjadi landasan untuk merancang sistem database yang tidak hanya mengatasi tantangan ini tetapi juga memberikan solusi inovatif untuk meningkatkan kualitas manajemen pembelian obat di apotek.

1.3 Tujuan

Penelitian ini memiliki beberapa tujuan yang jelas, yaitu:

1. Membangun sistem database yang dapat mengelola pembelian obat dengan efisien dan efektif.
2. Mengoptimalkan manajemen stok obat untuk menghindari kerugian akibat kekurangan atau kelebihan persediaan.
3. Menciptakan platform yang memungkinkan analisis data pembelian obat untuk perencanaan dan pengambilan keputusan yang lebih baik.

1.4 Manfaat

Penerapan sistem database ini diharapkan dapat memberikan manfaat yang signifikan, seperti:

1. Peningkatan efisiensi operasional apotek dalam proses pembelian dan manajemen stok.
2. Pemahaman yang lebih baik tentang tren pembelian obat melalui analisis data yang terstruktur.
3. Respons yang lebih cepat terhadap perubahan kebutuhan pasar dan regulasi.
4. Peningkatan pelayanan pelanggan melalui ketersediaan obat yang lebih baik dan pilihan produk yang optimal.

Dengan demikian, penelitian ini diarahkan untuk memberikan kontribusi positif terhadap efektivitas dan kualitas manajemen pembelian obat di apotek, menciptakan fondasi untuk pelayanan kesehatan yang lebih baik dan responsif terhadap kebutuhan masyarakat.

BAB II

LANDASAN TEORI

2.1 Database

Database (basis data) merupakan kumpulan informasi yang terstruktur dan tersimpan dalam sistem komputer, memungkinkan akses, pengelolaan, serta pengambilan informasi dengan bantuan program komputer sehingga memfasilitasi pengguna untuk mendapatkan data yang diperlukan untuk keperluan analisis, pelaporan, dan pengembangan aplikasi. Konseptualnya, sebuah basis data dapat dianggap sebagai pondasi atau pangkalan data yang memberikan landasan yang kokoh bagi pengelolaan informasi dalam berbagai konteks organisasi maupun aplikasi.

2.2 Database Management System

Database Management System (DBMS), yang umumnya dikenal sebagai perangkat lunak, mengatur serta melaksanakan perintah-perintah yang terkait dengan basis data, memainkan peran penting dalam pengelolaan informasi. Perangkat lunak DBMS ini berfungsi untuk mengurus semua aspek yang terkait dengan basis data, mulai dari tahap pembuatan awal hingga operasi-operasi yang terjadi di dalamnya, seperti penambahan, pengubahan, atau penghapusan data, penggunaan query untuk mengekstrak informasi, serta pembuatan laporan.

2.3 Relational DBMS

Relational Database Management System (RDBMS), menggambarkan informasi dalam format tabel yang terkait satu sama lain, termasuk di antara jenis DBMS yang paling populer saat ini. Dalam konteks ini, tabel terdiri dari baris yang mencakup rekaman data dan kolom yang berisi bidang informasi. Terdapat sejumlah perangkat lunak RDBMS yang telah dikembangkan, seperti MySQL, Oracle, Sybase, dBase, MS. SQL, Microsoft Access (MS. Access), dan beberapa lainnya.

2.4 Entity Relational Diagram

Entity Relational Diagram (ERD) adalah representasi grafis yang sering digunakan dalam perancangan basis data untuk menampilkan secara visual hubungan atau keterkaitan antara entitas atau objek beserta atribut-atributnya.

2.5 Data Definition Language

Data Definition Language (DDL), adalah bagian integral dari bahasa query SQL yang secara khusus ditujukan untuk menetapkan struktur data dalam suatu basis data. Tujuan utamanya adalah untuk menyediakan query yang meliputi pembuatan (CREATE) untuk basis data dan tabel, penghapusan (DROP) tabel atau basis data, serta perubahan (ALTER) struktur tabel yang sudah ada dengan kemampuan menambah, mengubah nama, atau menghapus kolom.

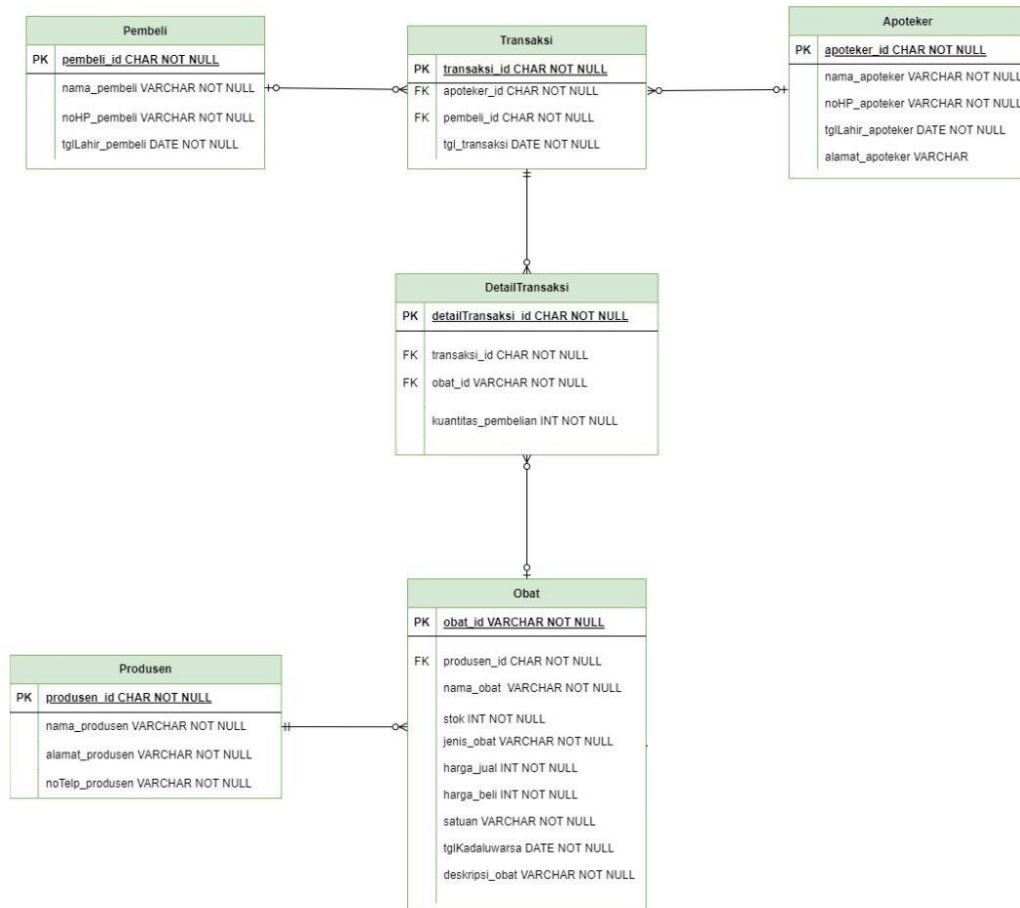
2.6 Data Manipulation Language

Data Manipulation Language (DML) merupakan segmen dari bahasa query SQL yang diterapkan setelah penggunaan DDL, fokus pada proses manipulasi terhadap basis data yang telah terbentuk sebelumnya. Operasi-operasi dalam DML mencakup INSERT (untuk menambahkan data ke dalam tabel basis data), UPDATE (untuk melakukan perubahan pada data yang telah ada dalam tabel basis data), dan DELETE (untuk menghapus data dari tabel basis data).

BAB III

HASIL & PEMBAHASAN

3.1 Entity Relational Diagram



3.2 List of Attributes

1. Entity MsCustomer

Attribute	Description	Type	Length	Constraint
customerID	Nomor unik sebagai representasi tabel MsCustomer, PRIMARY KEY	CHAR	5	PRIMARY KEY, dimulai dengan CU diikuti 3 digit angka
customerName	Nama customer	VARCHAR	50	NOT NULL
customerPhone	Nomor telepon customer	VARCHAR	14	Dimulai dengan 62, semua digit merupakan angka, jumlah digit di antara 11-14

customerDOB	Tanggal lahir customer	DATE		NOT NULL
-------------	------------------------	------	--	----------

2. Entity MsStaff

Attribute	Description	Type	Length	Constraint
customerID	Nomor unik sebagai representasi tabel MsStaff, PRIMARY KEY	CHAR	5	PRIMARY KEY, dimulai dengan ST diikuti 3 digit angka
staffName	Nama staff	VARCHAR	50	NOT NULL
staffPhone	Nomor telepon staff	VARCHAR	14	Dimulai dengan 62, semua digit merupakan angka, jumlah digit di antara 11-14
staffDOB	Tanggal lahir staff	DATE		NOT NULL
staffAddress	Alamat staff	VARCHAR		

3. Entity TransactionHeader

Attribute	Description	Type	Length	Constraint
transactionID	Nomor unik sebagai representasi tabel TransactionHeader	CHAR	5	PRIMARY KEY, dimulai dengan TR diikuti 3 digit angka
staffID	ID Staff yang menangani transaksi	CHAR	5	FOREIGN KEY, REFERENCES MsStaff(staffID)
customerID	ID Customer yang melakukan transaksi	CHAR	5	FOREIGN KEY, REFERENCES MsCustomer(customerID)
transactionDate	Tanggal transaksi	DATE		NOT NULL

4. Entity TransactionDetail

Attribute	Description	Type	Length	Constraint
transDetailID	Nomor unik sebagai representasi tabel TransactionDetail, PRIMARY KEY	CHAR	5	PRIMARY KEY, dimulai dengan TD diikuti 3 digit angka
transactionID	ID transaksi pembelian obat	CHAR	5	FOREIGN KEY, REFERENCES transactionDetail(transactionID)
obatID	ID obat yang dibeli	CHAR	6	FOREIGN KEY, REFERENCES MsObat (obatID)
jumlahBeli	Jumlah obat yang dibeli	INT		NOT NULL

5. Entity MsProdusen

Attribute	Description	Type	Length	Constraint
-----------	-------------	------	--------	------------

produsenID	Nomor unik sebagai representasi tabel MsProdusen	CHAR	5	PRIMARY KEY, dimulai dengan PR diikuti 3 digit angka
produsenName	Nama produsen	VARCHAR	50	NOT NULL
produsenPhone	Nomor telepon produsen	VARCHAR	14	Dimulai dengan 62, semua digit merupakan angka, jumlah digit di antara 11-14
produsenAddress	Alamat produsen	VARCHAR		NOT NULL

6. Entity MsObat

Attribute	Description	Type	Length	Constraint
obatID	Nomor unik sebagai representasi tabel MsStaff, PRIMARY KEY	CHAR	5	PRIMARY KEY, dimulai dengan MED diikuti 3 digit angka
produsenID	Nama produsen yang memproduksi obat	CHAR	50	FOREIGN KEY, REFERENCES MsProdusen(produsenID)
hargaBeli	Harga beli obat dari produsen	INT	14	NOT NULL
obatName	Nama obat	VARCHAR	50	NOT NULL
obatStock	Stok obat	INT		NOT NULL
hargaJual	Harga jual obat ke customer	INT		NOT NULL
expDate	Tanggal kadaluwarsa obat	DATE		NOT NULL
jenisObat	Jenis obat	VARCHAR	30	NOT NULL
satuan	Berat bersih satuan obat	VARCHAR	20	NOT NULL
deskripsiObat	Deskripsi obat	VARCHAR	70	NOT NULL

3.3 Database

1. Create and Insert Table

a) MsCustomer

CREATE TABLE MsCustomer (

customerID CHAR(5) PRIMARY KEY NOT NULL,

customerName VARCHAR(50) NOT NULL,

customerPhone VARCHAR(14) NOT NULL,

customerDOB DATE NOT NULL

CHECK (

```

        LENGTH(customerPhone) BETWEEN 12 AND 14

        AND SUBSTRING(customerPhone, 1, 2) = '62'

        AND customerPhone REGEXP '^[0-9]+$'

        AND customerID REGEXP '^CU[0-9]{3}$'

    )

);

INSERT INTO mscustomer

VALUES ('CU001','Budi Santoso','6281334455678','1990-05-15'),

('CU002','Gina Hermawan','6289876543210','1997-01-10'),

('CU003','William Tan','6285678901234','1999-02-14'),

('CU004','Putri Febiyani','6282345678901','2004-02-10'),

('CU005','Regina Celine','6283456789012','2004-06-09'),

('CU006','Nazhira Dewi','6281112345678','2005-01-11'),

('CU007','Sean Putra','6288887654321','2001-01-05'),

('CU008','Andrew Manurung','6287771234567','2002-08-14'),

('CU009','Sangkara Pratama','6289998765432','2002-12-31'),

('CU010','Asad Alkatiri','6286669876543','2004-01-28')

```

customerID	customerName	customerPhone	customerDOB
CU001	Budi Santoso	6281334455678	1990-05-15
CU002	Gina Hermawan	6289876543210	1997-01-10
CU003	William Tan	6285678901234	1999-02-14
CU004	Putri Febiyani	6282345678901	2004-02-10
CU005	Regina Celine	6283456789012	2004-06-09
CU006	Nazhira Dewi	6281112345678	2005-01-11
CU007	Sean Putra	6288887654321	2001-01-05
CU008	Andrew Manurung	6287771234567	2002-08-14
CU009	Sangkara Pratama	6289998765432	2002-12-31
CU010	Asad Alkatiri	6286669876543	2004-01-28
CU011	Budi Utomo	621231231322	2004-02-13

b) MsStaff

```

CREATE TABLE MsStaff (
    staffID CHAR(5) PRIMARY KEY NOT NULL,
    staffName VARCHAR(50) NOT NULL,
    staffPhone VARCHAR(14) NOT NULL,
    staffDOB DATE NOT NULL,
    staffAddress VARCHAR(70)

    CHECK (
        LENGTH(staffPhone) BETWEEN 12 AND 14
        AND SUBSTRING(staffPhone, 1, 2) = '62'
        AND staffPhone REGEXP '^([0-9])+$'
        AND staffID REGEXP '^ST([0-9]){3}$'
    )
)

```

);

```
INSERT INTO MsStaff (staffID, staffName, staffPhone, staffDOB, staffAddress)
VALUES
```

```
('ST001', 'Adi Nugroho', '6281234567890', '1990-05-15', 'Jl. Merdeka No. 10,
Malang'),
```

```
('ST002', 'Putri Indah', '6282345678901', '1985-08-20', 'Jl. Sudirman No. 15, Batu'),
```

```
('ST003', 'Budi Santoso', '6283456789012', '1992-11-10', 'Jl. Pahlawan No. 5,
Blitar'),
```

```
('ST004', 'Eka Surya', '6284567890123', '1987-04-25', 'Jl. Diponegoro No. 25,
Probolinggo'),
```

```
('ST005', 'Lina Setiawati', '6285678901234', '1995-09-30', 'Jl. Gajah Mada No. 7,
Kota Batu'),
```

```
('ST006', 'Putra Wirawan', '6286789012345', '1988-12-05', 'Jl. Mawar No. 12,
Pasuruan'),
```

```
('ST007', 'Dewi Wulandari', '6287890123456', '1991-07-12', 'Jl. Surya Sumantri No.
17, Probolinggo'),
```

```
('ST008', 'Rudi Hidayat', '6288901234567', '1983-02-28', 'Jl. Ganesha No. 3,
Blitar'),
```

```
('ST009', 'Ani Cahyani', '6289012345678', '1997-03-18', 'Jl. Panglima Sudirman
No. 9, Kota Batu'),
```

```
('ST010', 'Sinta Riyanti', '6280123456789', '1986-06-22', 'Jl. Diponegoro No. 8,
Pasuruan'),
```

```
('ST011', 'Bagus Kusuma', '6281234567890', '1994-01-09', 'Jl. Hayam Wuruk No.
16, Malang'),
```

```
('ST012', 'Sari Utami', '6282345678901', '1989-10-14', 'Jl. Pahlawan No. 20, Kota
Batu'),
```

('ST013', 'Fajar Perdana', '6283456789012', '1993-12-01', 'Jl. Medan Merdeka No. 30, Blitar'),

('ST014', 'Rina Susanti', '6284567890123', '1984-07-27', 'Jl. Pemuda No. 50, Malang'),

('ST015', 'Doni Firmansyah', '6285678901234', '1998-08-03', 'Jl. Gajah Mada No. 40, Pasuruan'),

('ST016', 'Larasati Widya', '6286789012345', '1982-11-19', 'Jl. Thamrin No. 23, Probolinggo'),

('ST017', 'Reza Aditya', '6287890123456', '1996-04-17', 'Jl. Sudirman No. 35, Malang'),

('ST018', 'Nina Permata', '6288901234567', '1981-05-08', 'Jl. Pahlawan No. 10, Kota Batu'),

('ST019', 'Andi Kurniawan', '6289012345678', '1990-02-25', 'Jl. Diponegoro No. 5, Blitar'),

('ST020', 'Dini Ariska', '6280123456789', '1987-06-11', 'Jl. Ganesha No. 9, Probolinggo');

staffID	staffName	staffPhone	staffDOB	staffAddress
ST001	Adi Nugroho	6281234567890	1990-05-15	Jl. Merdeka No. 10, Malang
ST002	Putri Indah	6282345678901	1985-08-20	Jl. Sudirman No. 15, Batu
ST003	Budi Santoso	6283456789012	1992-11-10	Jl. Pahlawan No. 5, Blitar
ST004	Eka Surya	6284567890123	1987-04-25	Jl. Diponegoro No. 25, Probolinggo
ST005	Lina Setiawati	6285678901234	1995-09-30	Jl. Gajah Mada No. 7, Kota Batu
ST006	Putra Wirawan	6286789012345	1988-12-05	Jl. Mawar No. 12, Pasuruan
ST007	Dewi Wulandari	6287890123456	1991-07-12	Jl. Surya Sumantri No. 17, Probolinggo
ST008	Rudi Hidayat	6288901234567	1983-02-28	Jl. Ganesha No. 3, Blitar
ST009	Ani Cahyani	6289012345678	1997-03-18	Jl. Panglima Sudirman No. 9, Kota Batu
ST010	Sinta Riyanti	6280123456789	1986-06-22	Jl. Diponegoro No. 8, Pasuruan

c) TransactionHeader

CREATE TABLE TransactionHeader(

```

transactionID CHAR(5) PRIMARY KEY NOT NULL,

staffID CHAR(5) NOT NULL,

customerID CHAR(5) NOT NULL,

transactionDate DATE NOT NULL,


FOREIGN KEY (staffID) REFERENCES msstaff(staffID) ON UPDATE
CASCADE ON DELETE CASCADE,


FOREIGN KEY (customerID) REFERENCES mscustomer(customerID) ON
UPDATE CASCADE ON DELETE CASCADE


);


INSERT INTO TransactionHeader (transactionID, staffID, customerID,
transactionDate)

VALUES

('TR001', 'ST001', 'CU001', '2023-01-15'),

('TR002', 'ST002', 'CU002', '2023-02-20'),

('TR003', 'ST003', 'CU001', '2023-03-10'),

('TR004', 'ST002', 'CU004', '2023-04-25'),

('TR005', 'ST005', 'CU005', '2023-05-30'),

('TR006', 'ST003', 'CU006', '2023-06-12'),

('TR007', 'ST007', 'CU007', '2023-07-12'),

('TR008', 'ST008', 'CU008', '2023-08-28'),

('TR009', 'ST007', 'CU006', '2023-09-18'),

('TR010', 'ST010', 'CU010', '2023-10-22')

```

transactionID	staffID	customerID	transactionDate
TR001	ST001	CU001	2023-01-15
TR002	ST002	CU002	2023-02-20
TR003	ST003	CU001	2023-03-10
TR004	ST002	CU004	2023-05-25
TR005	ST005	CU005	2023-05-30
TR006	ST003	CU006	2023-06-12
TR007	ST007	CU007	2023-07-12
TR008	ST008	CU008	2023-08-28
TR009	ST007	CU006	2023-09-18
TR010	ST010	CU010	2023-10-22
TR011	ST003	CU011	2023-12-02

d) TransactionHeader

```

CREATE TABLE TransactionDetail(

    transDetailID CHAR(5) PRIMARY KEY NOT NULL,

    transactionID CHAR(5) NOT NULL,

    obatID CHAR(6) NOT NULL,

    jumlahBeli INT NOT NULL

    CHECK (

        transDetailID REGEXP '^TD[0-9]{3}$'

    ),

    FOREIGN KEY (transactionID) REFERENCES
transactionheader(transactionID) ON UPDATE CASCADE ON DELETE
CASCADE,

    FOREIGN KEY (obatID) REFERENCES msobat(obatID) ON UPDATE
CASCADE ON DELETE CASCADE

);

```

```
INSERT INTO transactiondetail(transDetailID, transactionID, obatID, jumlahBeli)
VALUES
('TD001', 'TR001', 'MED006', 2),
('TD002', 'TR001', 'MED002', 1),
('TD003', 'TR002', 'MED011', 3),
('TD004', 'TR003', 'MED001', 1),
('TD005', 'TR004', 'MED009', 5),
('TD006', 'TR005', 'MED003', 2),
('TD007', 'TR006', 'MED008', 1),
('TD008', 'TR007', 'MED016', 4),
('TD009', 'TR007', 'MED020', 2),
('TD010', 'TR008', 'MED004', 6),
('TD011', 'TR009', 'MED007', 1),
('TD012', 'TR009', 'MED001', 2),
('TD013', 'TR010', 'MED013', 3)
```


transDetailID	transactionID	obatID	jumlahBeli
TD001	TR001	MED006	2
TD002	TR001	MED002	1
TD003	TR002	MED011	3
TD004	TR003	MED001	1
TD005	TR004	MED009	5
TD006	TR005	MED003	2
TD007	TR006	MED008	1
TD008	TR007	MED016	4
TD009	TR007	MED020	2
TD010	TR008	MED004	6
TD011	TR009	MED007	1
TD012	TR009	MED001	2
TD013	TR010	MED013	3
TD014	TR011	MED007	4

e) MsProdusen

```

CREATE TABLE MsProdusen (
    produsenID CHAR(5) PRIMARY KEY NOT NULL,
    produsenName VARCHAR(50) NOT NULL,
    produsenPhone VARCHAR(14) NOT NULL,
    produsenAddress VARCHAR(70) NOT NULL
    CHECK (
        LENGTH(produsenPhone) BETWEEN 12 AND 14
        AND SUBSTRING(produsenPhone, 1, 2) = '62'
        AND produsenPhone REGEXP '^[0-9]+$'
        AND produsenID REGEXP '^PR[0-9]{3}$'
    )
)

```

);

```
INSERT INTO msprodusen (produsenID, produsenName, produsenPhone,
produsenAddress)
```

```
VALUES
```

```
('PR001', 'PT. Jaya Makmur', '628111111111', 'Jl. Ikan Mas, Surabaya'),
```

```
('PR002', 'PT. Aman Sentosa', '628222222222', 'Jl. Ikan Patin, Surabaya'),
```

```
('PR003', 'PT. Maju Jaya', '6283333333333', 'Jl. Ikan Lele, Malang'),
```

```
('PR004', 'PT. Cahaya Nur', '6284444444444', 'Jl. Kemanggisan, Jakarta'),
```

```
('PR005', 'PT. Anak Agung', '6285555555555', 'Jl. Bulungcangkring, Jakarta'),
```

```
('PR006', 'PT. Bakti', '6286666666666', 'Jl. Tembalang, Jakarta'),
```

```
('PR007', 'PT. Subur Indah', '6287777777777', 'Jl. Pattimura, Jakarta'),
```

```
('PR008', 'PT. Kimia Farma', '6288888888888', 'Jl. Sayonara, Bandung'),
```

```
('PR009', 'PT. Kalbe Farma', '6289999999999', 'Jl. Cut Nyak Dien, Jakarta'),
```

```
('PR010', 'PT. Dexa Medica', '6281010101010', 'Jl. Kemenangan, Jakarta'),
```

```
('PR011', 'PT. Fahrenheit', '6281414141414', 'Jl. Blimbing, Surabaya'),
```

```
('PR012', 'PT. Hidup Sehat', '6281212121212', 'Jl. Lowokwaru, Bandung')
```

produsenID	produsenName	produsenPhone	produsenAddress
PR001	PT. Jaya Makmur	628111111111	Jl. Ikan Mas, Surabaya
PR002	PT. Aman Sentosa	628222222222	Jl. Ikan Patin, Surabaya
PR003	PT. Maju Jaya	628333333333	Jl. Ikan Lele, Malang
PR004	PT. Cahaya Nur	628444444444	Jl. Kemanggisan, Jakarta
PR005	PT. Anak Agung	628555555555	Jl. Bulungcangkring, Jakart
PR006	PT. Bakti	628666666666	Jl. Tembalang, Jakarta
PR007	PT. Subur Indah	628777777777	Jl. Pattimura, Jakarta
PR008	PT. Kimia Farma	628888888888	Jl. Sayonara, Bandung
PR009	PT. Kalbe Farma	628999999999	Jl. Cut Nyak Dien, Jakarta
PR010	PT. Dexa Medica	628101010101	Jl. Kemenangan, Jakarta
PR011	PT. Fahrenheit	628141414141	Jl. Blimbing, Surabaya
PR012	PT. Hidup Sehat	628121212121	Jl. Lowokwaru, Bandung

f) MsObat

```

CREATE TABLE MsObat (

    obatID CHAR(6) PRIMARY KEY NOT NULL,

    produsenID CHAR(5) NOT NULL,

    hargaBeli INT NOT NULL,

    obatName VARCHAR(50) NOT NULL,

    obatStock INT NOT NULL,

    hargaJual INT NOT NULL,

    expDate DATE NOT NULL,

    jenisObat VARCHAR(30) NOT NULL,

    satuan VARCHAR(20) NOT NULL,

    deskripsiObat VARCHAR(70) NOT NULL

CHECK (

```

```

obatID REGEXP '^MED[0-9]{3}$'

),

FOREIGN KEY (produsenID) REFERENCES msprodusen(produsenID) ON
UPDATE CASCADE ON DELETE CASCADE

);

INSERT INTO msobat (obatID, produsenID, hargaBeli, obatName, obatStock,
hargaJual, expDate,
jenisObat, satuan, deskripsiObat)
VALUES

('MED001', 'PR001', 3500, 'Oskadon', 30, 6000, '2025-05-18', 'Tablet', '300 g',
'Menyembuhkan sakit kepala dan pegal-pegal'),

('MED002', 'PR001', 12000, 'Paramex', 50, 15000, '2024-12-15', 'Tablet', '250 g',
'Menyembuhkan sakit kepala'),

('MED003', 'PR001', 4000, 'Antangin', 40, 8000, '2026-02-24', 'Tablet', '350 g',
'Menyembuhkan masuk angin'),

('MED004', 'PR002', 1500, 'Tolak Angin', 60, 3000, '2025-01-08', 'Sirup', '15 ml',
'Menyembuhkan masuk angin'),

('MED005', 'PR002', 20000, 'Oralit', 35, 25000, '2025-05-18', 'Sirup', '200 ml',
'Menggantikan cairan tubuh saat diare/muntah/demam'),

('MED006', 'PR002', 10000, 'Promag', 45, 12000, '2025-05-18', 'Tablet', '350 g',
'Menyembuhkan sakit maag'),

('MED007', 'PR003', 5000, 'Mylanta', 20, 7500, '2025-05-18', 'Tablet', '300 g',
'Menyembuhkan sakit maag'),

('MED008', 'PR004', 7000, 'Paracetamol', 30, 10000, '2026-03-06', 'Tablet', '200 g',
'Menyembuhkan sakit kepala dan nyeri'),

```

('MED009', 'PR005', 13000, 'Sangobion', 50, 15000, '2025-10-14', 'Kapsul', '320 g',
'Suplemen penambah darah'),

('MED010', 'PR006', 9000, 'Ester C', 60, 12000, '2027-01-11', 'Tablet', '250 g',
'Suplemen vitamin'),

('MED011', 'PR006', 6000, 'UC 1000', 30, 8500, '2025-05-18', 'Sirup', '30 ml',
'Suplemen vitamin'),

('MED012', 'PR007', 34000, 'Vitacimin', 40, 38000, '2025-05-18', 'Kapsul', '500 g',
'Suplemen vitamin'),

('MED013', 'PR007', 8500, 'Paratusin', 65, 11000, '2025-05-18', 'Tablet', '600 g',
'Menyembuhkan pilek'),

('MED014', 'PR008', 23000, 'Sakatonik ABC', 25, 27000, '2025-05-18', 'Tablet',
'550 g', 'Suplemen vitamin'),

('MED015', 'PR009', 10000, 'Amoxilin', 45, 14000, '2025-05-18', 'Tablet', '300 g',
'Obat antibiotik'),

('MED016', 'PR009', 8000, 'Panadol', 30, 10000, '2025-05-18', 'Kapsul', '350 g',
'Menyembuhkan demam'),

('MED017', 'PR009', 17000, 'Ambeven', 20, 21000, '2025-05-18', 'Tablet', '370 g',
'Menyembuhkan wasir'),

('MED018', 'PR010', 13000, 'Caladine', 55, 15000, '2025-05-18', 'Salep', '200 g',
'Menyembuhkan gatal-gatal'),

('MED019', 'PR011', 16000, 'Hydrocortisone', 50, 19000, '2025-05-18', 'Salep', '250
g', 'Menyembuhkan alergi'),

('MED020', 'PR012', 21000, 'OB Herbal', 30, 23500, '2025-05-18', 'Sirup', '50 ml',
'Menyembuhkan sakit batuk')

obatID	produsenID	hargaBeli	obatName	obatStock	hargaJual	expDate	jenisObat	satuan	deskripsiObat
MED001	PR001	3500	Oskadon	30	6000	2025-05-18	Tablet	300 g	Menyembuhkan sakit kepala dan pegal-pegal
MED002	PR001	12000	Paramex	50	15000	2024-12-15	Tablet	250 g	Menyembuhkan sakit kepala
MED003	PR001	4000	Antangin	40	8000	2026-02-24	Tablet	350 g	Menyembuhkan masuk angin
MED004	PR002	1500	Tolak Angin	60	3000	2025-01-08	Sirup	15 ml	Menyembuhkan masuk angin
MED005	PR002	20000	Oralit	35	25000	2025-05-18	Sirup	200 ml	Menggantikan cairan tubuh saat diare/muntah/demam
MED006	PR002	10000	Promag	45	12000	2025-05-18	Tablet	350 g	Menyembuhkan sakit maag
MED007	PR003	5000	Mylanta	20	7500	2025-05-18	Tablet	300 g	Menyembuhkan sakit maag
MED008	PR004	7000	Paracetamol	30	10000	2026-03-06	Tablet	200 g	Menyembuhkan sakit kepala dan nyeri
MED009	PR005	13000	Sangobion	50	15000	2025-10-14	Kapsul	320 g	Suplemen penambah darah
MED010	PR006	9000	Ester C	60	12000	2027-01-11	Tablet	250 g	Suplemen vitamin
MED011	PR006	6000	UC 1000	30	8500	2025-05-18	Sirup	30 ml	Suplemen vitamin
MED012	PR007	34000	Vitacimin	40	38000	2025-05-18	Kapsul	500 g	Suplemen vitamin
MED013	PR007	8500	Paratusin	65	11000	2025-05-18	Tablet	600 g	Menyembuhkan pilek
MED014	PR008	23000	Sakatonik ABC	25	27000	2025-05-18	Tablet	550 g	Suplemen vitamin
MED015	PR009	10000	Amoxilin	45	14000	2025-05-18	Tablet	300 g	Obat antibiotik
MED016	PR009	8000	Panadol	30	10000	2025-05-18	Kapsul	350 g	Menyembuhkan demam
MED017	PR009	17000	Ambeven	20	21000	2025-05-18	Tablet	370 g	Menyembuhkan wasir
MED018	PR010	13000	Caladine	55	15000	2025-05-18	Salep	200 g	Menyembuhkan gatal-gatal
MED019	PR011	16000	Hydrocortisone	50	19000	2025-05-18	Salep	250 g	Menyembuhkan alergi
MED020	PR012	21000	OB Herbal	30	23500	2025-05-18	Sirup	50 ml	Menyembuhkan sakit batuk

2. Create View

a) informasi_produsen

CREATE VIEW Informasi_Produsen AS

SELECT P.produsenName, P.produsenPhone, P.produsenAddress,
GROUP_CONCAT(O.obatName) AS `Nama Obat`

FROM msobat AS O

INNER JOIN msprodusen AS P

ON P.produsenID = O.produsenID

GROUP BY P.produsenName

produsenName	produsenPhone	produsenAddress	Nama Obat
PT. Aman Sentosa	628222222222	Jl. Ikan Patin, Surabaya	Tolak Angin,Oralit,Promag
PT. Anak Agung	628555555555	Jl. Bulungcangkring, Jakarta	Sangobion
PT. Bakti	628666666666	Jl. Tembalang, Jakarta	Ester C,UC 1000
PT. Cahaya Nur	628444444444	Jl. Kemanggisan, Jakarta	Paracetamol
PT. Dexa Medica	6281010101010	Jl. Kemenangan, Jakarta	Caladine
PT. Fahrenheit	6281414141414	Jl. Blimbing, Surabaya	Hydrocortisone
PT. Hidup Sehat	6281212121212	Jl. Lowokwaru, Bandung	OB Herbal
PT. Jaya Makmur	6281111111111	Jl. Ikan Mas, Surabaya	Oskadon,Paramex,Antangin
PT. Kalbe Farma	6289999999999	Jl. Cut Nyak Dien, Jakarta	Amoxilin,Panadol,Ambeven
PT. Kimia Farma	6288888888888	Jl. Sayonara, Bandung	Sakatonik ABC
PT. Maju Jaya	6283333333333	Jl. Ikan Lele, Malang	Mylanta
PT. Subur Indah	6287777777777	Jl. Pattimura, Jakarta	Vitacimin,Paratusin

b) nota_pembelian

CREATE VIEW Nota_Pembelian AS

SELECT TH.transactionDate, C.customerName, S.staffName,

O.obatName, TD.jumlahBeli, O.hargaJual, O.hargaJual*TD.jumlahBeli AS Total

FROM transactionheader AS TH

INNER JOIN mscustomer AS C

ON C.customerID = TH.customerID

INNER JOIN msstaff AS S

ON S.staffID = TH.staffID

INNER JOIN transactiondetail AS TD

ON TD.transactionID = TH.transactionID

INNER JOIN msobat AS O

ON O.obatID = TD.obatID

transactionDate	customerName	TotalPembelian
2023-01-15	Budi Santoso	39000
2023-02-20	Gina Hermawan	25500
2023-03-10	Budi Santoso	6000
2023-05-25	Putri Febiyani	75000
2023-05-30	Regina Celine	16000
2023-06-12	Nazhira Dewi	10000
2023-07-12	Sean Putra	87000
2023-08-28	Andrew Manurung	18000
2023-09-18	Nazhira Dewi	19500
2023-10-22	Asad Alkatiri	33000
2023-12-04	Budi Utomo	30000

3. Function

a) Calculate_Income()

DELIMITER //

CREATE FUNCTION Calculate_Income()

RETURNS INT

BEGIN

 DECLARE totalJual INT;

 DECLARE totalBeli INT;

 DECLARE income INT;

 SELECT SUM(O.hargaJual*TD.jumlahBeli) INTO totalJual

 FROM transactiondetail AS TD

 INNER JOIN msobat AS O

 ON O.obatID=TD.obatID

 INNER JOIN transactionheader AS TH

 ON TH.transactionID = TD.transactionID;

```

SELECT SUM(O.hargaBeli*TD.jumlahBeli) INTO totalBeli

FROM transactiondetail AS TD

INNER JOIN msobat AS O

ON O.obatID=TD.obatID

INNER JOIN transactionheader AS TH

ON TH.transactionID = TD.transactionID;

SET income = totalJual-totalBeli;

RETURN income;

END

//

SELECT Calculate_Income()

```

```

Calculate_Income()
85000

```

4. Procedures

a) **Update_Stock()**

```

DELIMITER //

CREATE PROCEDURE Update_Stock()

BEGIN

    UPDATE msobat AS O

    INNER JOIN transactiondetail AS TD

    ON TD.obatID = O.obatID

    INNER JOIN transactionheader AS TH

    ON TH.transactionID = TD.transactionID

```

SET O.obatStock = O.obatStock - TD.jumlahBeli

WHERE TH.transactionDate = CURDATE();

END //

DELIMITER ;

TR010	ST010	CU010	2023-12-05
-------	-------	-------	------------

TD013	TR010	MED013	3
-------	-------	--------	---

MED013 PR007	8500 Paratusin	65	11000 2025-05-18 Tablet	600 g	Menyembuhkan pilek
--------------	----------------	----	-------------------------	-------	--------------------

CALL Update_Stock()

MED013 PR007	8500 Paratusin	62	11000 2025-05-18 Tablet	600 g	Menyembuhkan pilek
--------------	----------------	----	-------------------------	-------	--------------------

b) Update_AddStock(IN o_id CHARACTER(6), IN addInput INT)

DELIMITER //

CREATE PROCEDURE Update_AddStock(IN o_id CHARACTER(6), IN
addInput INT)

BEGIN

UPDATE msobat AS O

SET O.obatStock = O.obatStock + addInput

WHERE O.obatID = o_id;

END

//

DELIMITER ;

MED021 PR004	6500 Lapifed	15	9000 2024-07-23 Sirup	55 ml	Menyembuhkan pilek
--------------	--------------	----	-----------------------	-------	--------------------

CALL Update_AddStock('MED021', 20)

MED021 PR004	6500 Lapifed	35	9000 2024-07-23 Sirup	55 ml	Menyembuhkan pilek
--------------	--------------	----	-----------------------	-------	--------------------

c) Insert_MsObat()

DELIMITER //

```
CREATE PROCEDURE Insert_MsObat(IN o_id CHARACTER(6), IN p_id  
CHARACTER(5), IN hargaBeli INT, IN o_name VARCHAR(50), IN stock INT,  
IN hargaJual INT, IN exp DATE, IN jenis VARCHAR(30), IN satuan  
VARCHAR(20), deskripsi VARCHAR(70))
```

BEGIN

```
INSERT INTO msobat(obatID, produsenID, hargaBeli, obatName,  
obatStock, hargaJual, expDate, jenisObat, satuan, deskripsiObat) VALUES(o_id,  
p_id, hargaBeli, o_name, stock, hargaJual, exp, jenis, satuan, deskripsi);
```

END //

DELIMITER ;

```
CALL Insert_MsObat('MED022', 'PR009', 8000, 'Cetirizine', 35, 10000, '2024-05-  
27', 'Tablet', '10 mg', 'Menyembuhkan gatal')
```

MED021	PR004	6500	Lapifed	35	9000	2024-07-23	Sirup	55 ml	Menyembuhkan pilek
MED022	PR009	8000	Cetirizine	35	10000	2024-05-27	Tablet	10 mg	Menyembuhkan gatal

d) Insert_MsCustomer()

DELIMITER //

```
CREATE PROCEDURE Insert_MsCustomer(IN c_id CHARACTER(5), IN  
c_name VARCHAR(50), IN c_phone VARCHAR(14), c_DOB DATE)
```

BEGIN

```
INSERT INTO mscustomer(customerID, customerName, customerPhone,  
customerDOB) VALUES(c_id, c_name, c_phone, c_DOB);
```

END //

DELIMITER ;

```
CALL Insert_MsCustomer('CU013', 'Karina Aespa', '6287836463743', '2000-08-  
15')
```

CU012	Stefani Maia	628398474747	2000-03-30
CU013	Karina Aespa	6287836463743	2000-08-15

e) Search_Transaction_ByID()

DELIMITER //

```
CREATE PROCEDURE Search_Transaction_ByID(IN transID CHAR(5), OUT
name VARCHAR(50),OUT phone VARCHAR(50), OUT list_obat
VARCHAR(100), OUT totalBayar INT)
```

BEGIN

```
SELECT C.customerName INTO name
```

```
FROM mscustomer AS C
```

```
INNER JOIN transactionheader AS TH ON C.customerID = TH.customerID
```

```
WHERE TH.transactionID = transID;
```

```
SELECT C.customerPhone INTO phone
```

```
FROM mscustomer AS C
```

```
INNER JOIN transactionheader AS TH ON C.customerID = TH.customerID
```

```
WHERE TH.transactionID = transID;
```

```
SELECT GROUP_CONCAT(O.obatName SEPARATOR ',') INTO list_obat
```

```
FROM transactiondetail AS TD
```

```
INNER JOIN msobat AS O ON TD.obatID = O.obatID
```

```
GROUP BY TD.transactionID
```

```
HAVING TD.transactionID = transID;
```

```
SELECT SUM(O.hargaJual * TD.jumlahBeli) INTO totalBayar
```

```
FROM transactiondetail AS TD
```

```
INNER JOIN msobat AS O ON O.obatID = TD.obatID
```

```

WHERE TD.transactionID = transID;

END

//

DELIMITER ;

CALL Search_Transaction_ByID('TR005', @name, @phone, @list_obat,
@totalBayar);

SELECT @name AS nama, @phone as no_telp, @list_obat AS obat, @totalBayar
AS total

```

nama	no_telp	obat	total
Regina Celine	6283456789012	Antangin	16000

f) Calculate_Income_ByMonth()

```

DELIMITER //

CREATE PROCEDURE Calculate_Income_ByMonth(IN tahun YEAR)

BEGIN

    SELECT MONTHNAME(TH.transactionDate) AS Month,

    SUM(O.hargaJual * TD.jumlahBeli) - SUM(O.hargaBeli * TD.jumlahBeli) AS
Income

    FROM transactiondetail AS TD

    INNER JOIN msobat AS O ON O.obatID = TD.obatID

    INNER JOIN transactionheader AS TH ON TH.transactionID =
TD.transactionID

    WHERE YEAR(TH.transactionDate) = tahun

    GROUP BY MONTHNAME(TH.transactionDate)

    ORDER BY MONTH(TH.transactionDate);

END

```

//

DELIMITER ;

CALL Calculate_Income_ByMonth(2023)

Month	Income
January	7000
February	7500
March	2500
May	18000
June	3000
July	13000
August	9000
September	7500
December	17500

g) BiggestTransaction_ByMonth()

CREATE PROCEDURE BiggestTransaction_ByMonth(IN bulan INT, OUT
namaBulan VARCHAR(15), OUT nama VARCHAR(50), OUT transaksi INT)

BEGIN

DECLARE tempNamaBulan VARCHAR(15);

DECLARE tempNama VARCHAR(50);

DECLARE tempTransaksi INT;

SELECT MONTHNAME(TH.transactionDate), C.customerName,
SUM(O.hargaJual * TD.jumlahBeli)

INTO tempNamaBulan, tempNama, tempTransaksi

FROM transactionheader AS TH

INNER JOIN mscustomer AS C ON C.customerID = TH.customerID

INNER JOIN transactiondetail AS TD ON TH.transactionID = TD.transactionID

INNER JOIN msobat AS O ON O.obatID = TD.obatID

WHERE MONTH(TH.transactionDate) = bulan

GROUP BY TH.transactionDate, C.customerName

ORDER BY SUM(O.hargaJual * TD.jumlahBeli) DESC

LIMIT 1;

SET namaBulan = tempNamaBulan;

SET nama = tempNama;

SET transaksi = tempTransaksi;

END//

DELIMITER ;

CALL BiggestTransaction_ByMonth(3, @namaBulan, @nama, @transaksi);

SELECT @namaBulan as Bulan, @nama as Nama, @transaksi as totalTransaksi

Bulan	Nama	totalTransaksi
March	Budi Santoso	6000

3.4 Pertanyaan dan Jawaban Quiz

1. Buat query untuk menampilkan apoteker yang paling tinggi dan paling rendah pada jumlah pelayanan yang dilakukan

Hasil :

The screenshot shows the phpMyAdmin interface with a SQL query executed. The query is a complex join between the 'Pharmacist' and 'TotalServices' tables. The results table shows two rows: Budi Santoso with 3 total services and Rina Susanti with 0 total services.

Pharmacist	TotalServices
Budi Santoso	3
Rina Susanti	0

Query :


```

(
  SELECT
    A.staffName AS Pharmacist,
    A.TotalServices AS TotalServices
  FROM (
    SELECT
      S.staffName,
      COUNT(TH.transactionID) AS TotalServices
    FROM
      msstaff S
    LEFT JOIN
      transactionheader TH ON S.staffID = TH.staffID
    GROUP BY
      S.staffID, S.staffName
  ) A
  ORDER BY
    TotalServices DESC
  LIMIT 1
)
UNION ALL
(
  SELECT
    B.staffName AS Pharmacist,
    B.TotalServices AS TotalServices
  FROM (
    SELECT
      S.staffName,
      COUNT(TH.transactionID) AS TotalServices
    FROM
      msstaff S
    LEFT JOIN
      transactionheader TH ON S.staffID = TH.staffID
    GROUP BY
      S.staffID, S.staffName
  ) B
  ORDER BY
    TotalServices ASC
  LIMIT 1
);

```

2. Buat query untuk menampilkan data obat yang sering dibeli
Hasil :

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain](#)]

Extra options

obatName	total_pembelian
Tolak Angin	6

Query results operations

Query :

```
SELECT O.obatName, SUM(TD.jumlahBeli) AS total_pembelian
FROM msobat AS O
INNER JOIN transactiondetail AS TD ON TD.obatID = O.obatID
GROUP BY O.obatID, O.obatName
ORDER BY total_pembelian DESC
LIMIT 1;
```

3. Buat query untuk menampilkan daftar obat yang diproduksi oleh masing-masing produsen

Hasil :

produsenName	daftar_obat
PT. Jaya Makmur	Oskadon, Paramex, Antangin
PT. Aman Sentosa	Tolak Angin, Oralit, Promag
PT. Maju Jaya	Mylanta
PT. Cahaya Nur	Paracetamol, Lapifed
PT. Anak Agung	Sangobion
PT. Bakti	Ester C, UC 1000
PT. Subur Indah	Vitacimin, Paratusin
PT. Kimia Farma	Sakatonik ABC
PT. Kalbe Farma	Amoxilin, Panadol, Ambeven, Cetirizine
PT. Dexa Medica	Caladine
PT. Fahrenheit	Hydrocortisone
PT. Hidup Sehat	OB Herbal

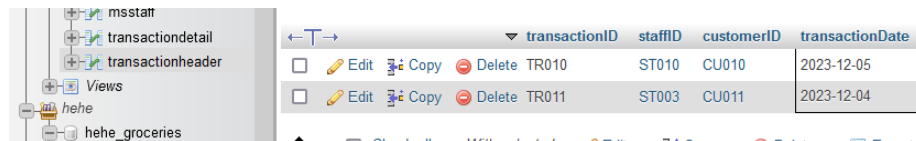
Query :

```
SELECT      P.produsenName,      GROUP_CONCAT(O.obatName
SEPARATOR ', ') AS      daftar_obat
FROM msobat AS O
INNER JOIN msprodusen AS P ON O.produsenID = P.produsenID
```

GROUP BY P.produkenID, P.produkenName;

4. Buat query untuk menampilkan transaksi pada 1 bulan terbaru

Hasil :



transactionID	staffID	customerID	transactionDate
TR010	ST010	CU010	2023-12-05
TR011	ST003	CU011	2023-12-04

Query :

SELECT *

FROM transactionheader

WHERE MONTH(transactionDate) = MONTH(CURRENT_DATE())

AND YEAR(transactionDate) = YEAR(CURRENT_DATE());

5. Buat query untuk menampilkan data obat yang jarang dibeli

Hasil :

obatName	JumlahPembelian
Sakatonik ABC	0
Amoxilin	0
Ambeven	0
Caladine	0
Hydrocortisone	0
Lapifed	0
Cetirizine	0
Oralit	0
Ester C	0
Vitacimin	0
Paramex	1
Paracetamol	1
OB Herbal	2
Antangin	2
Promag	2

Query :

```

SELECT O.obatName, COALESCE(SUM(TD.jumlahBeli), 0) AS
    JumlahPembelian
FROM msobat AS O
LEFT JOIN transactiondetail AS TD ON O.obatID = TD.obatID
GROUP BY O.obatID, O.obatName
ORDER BY SUM(TD.jumlahBeli)
LIMIT 15

```

6. Buat query untuk menampilkan data transaksi dan detail transaksi secara keseluruhan

Hasil :

transactionID	customerName	customerPhone	obatName	jumlahBeli	staffName
TR001	Budi Santoso	6281334455678	Promag	2	Adi Nugroho
TR001	Budi Santoso	6281334455678	Paramex	1	Adi Nugroho
TR002	Gina Hermawan	6289876543210	UC 1000	3	Putri Indah
TR003	Budi Santoso	6281334455678	Oskadon	1	Budi Santoso
TR004	Putri Febiyani	6282345678901	Sangobion	5	Putri Indah
TR005	Regina Celine	6283456789012	Antangin	2	Lina Setiawati
TR006	Nazhira Dewi	6281112345678	Paracetamol	1	Budi Santoso
TR007	Sean Putra	6288887654321	Panadol	4	Dewi Wulandari
TR007	Sean Putra	6288887654321	OB Herbal	2	Dewi Wulandari
TR008	Andrew Manurung	6287771234567	Tolak Angin	6	Rudi Hidayat
TR009	Nazhira Dewi	6281112345678	Mylanta	1	Dewi Wulandari
TR009	Nazhira Dewi	6281112345678	Oskadon	2	Dewi Wulandari
TR010	Asad Alkatiri	6286669876543	Paratusin	3	Sinta Riyanti
TR011	Budi Utomo	621231231322	Mylanta	4	Budi Santoso

Query :

```

SELECT TH.transactionID, C.customerName, C.customerPhone, O.obatName,
    TD.jumlahBeli, S.staffName FROM transactionheader AS TH
INNER JOIN mscustomer AS C
ON C.customerID = TH.customerID
JOIN transactiondetail AS TD
ON TH.transactionID = TD.transactionID
INNER JOIN msobat AS O
ON O.obatID = TD.obatID
INNER JOIN msstaff AS S
ON S.staffID = TH.staffID

```

7. Buat query untuk menampilkan data transaksi yang memiliki paling banyak jenis obat yang dibeli

Hasil :

transactionID	customerName	customerPhone	GROUP_CONCAT(O.obatName)	staffName
TR001	Budi Santoso	6281334455678	Promag,Paramex	Adi Nugroho
TR007	Sean Putra	6288887654321	Panadol,OB Herbal	Dewi Wulandari
TR009	Nazhira Dewi	6281112345678	Mylanta,Oskadon	Dewi Wulandari

Query :

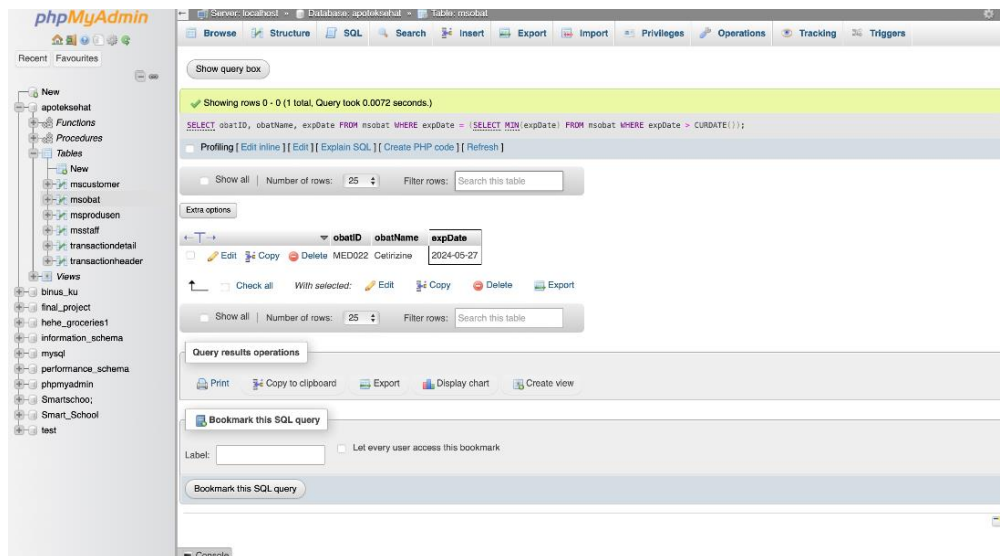
```

SELECT TH.transactionID, C.customerName, C.customerPhone,
      GROUP_CONCAT(O.obatName) AS obat, S.staffName FROM
transactionheader AS TH
INNER JOIN mscustomer AS C
ON C.customerID = TH.customerID
JOIN transactiondetail AS TD
ON TH.transactionID = TD.transactionID
INNER JOIN msobat AS O
ON O.obatID = TD.obatID
INNER JOIN msstaff AS S
ON S.staffID = TH.staffID
GROUP BY TH.transactionID
HAVING COUNT(O.obatID) > 1
ORDER BY COUNT(O.obatID) DESC

```

8. Buat query untuk menampilkan obat yang memiliki tanggal kadaluarsa terdekat

Hasil :



Query :

```

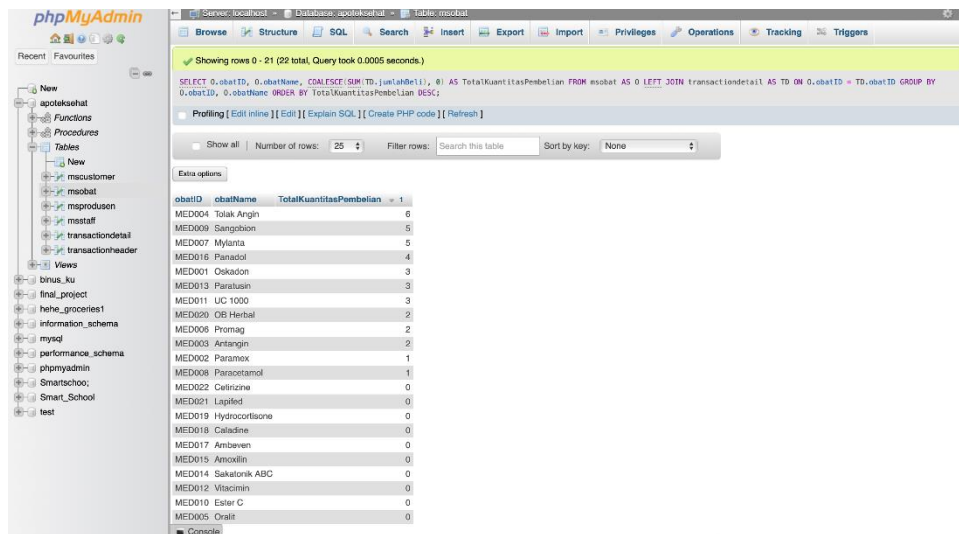
SELECT obatID, obatName, expDate
FROM msobat

```

WHERE expDate = (SELECT MIN(expDate) FROM msobat WHERE
expDate > CURDATE());

- Buat query untuk menampilkan total kuantitas pembelian pada masing-masing obat

Hasil :



Showing rows 0 - 21 (22 total, Query took 0.0005 seconds)

```
SELECT O.obatID, O.obatName, COALESCE(SUM(TD.jumlahBeli), 0) AS TotalKuantitasPembelian FROM msobat AS O LEFT JOIN transactiondetail AS TD ON O.obatID = TD.obatID GROUP BY O.obatID, O.obatName ORDER BY TotalKuantitasPembelian DESC;
```

obatID	obatName	TotalKuantitasPembelian
MED004	Tolak Angin	6
MED009	Sengoblon	5
MED007	Mykanta	5
MED016	Pianadol	4
MED001	Oskadon	3
MED013	Paralusin	3
MED011	UC 1000	3
MED020	OB Herbal	2
MED005	Promag	2
MED003	Antangin	2
MED002	Paracetamol	1
MED008	Paracetamol	1
MED022	Cefixime	0
MED021	Lapifed	0
MED019	Hydrocortisone	0
MED018	Caladine	0
MED017	Ambeven	0
MED015	Amoxilin	0
MED014	Sakatonik ABC	0
MED012	Miscimin	0
MED010	Ester C	0
MED005	Oralit	0

Query :

SELECT

O.obatID,

O.obatName,

COALESCE(SUM(TD.jumlahBeli), 0) AS TotalKuantitasPembelian

FROM

msobat AS O

LEFT JOIN

transactiondetail AS TD ON O.obatID = TD.obatID

GROUP BY

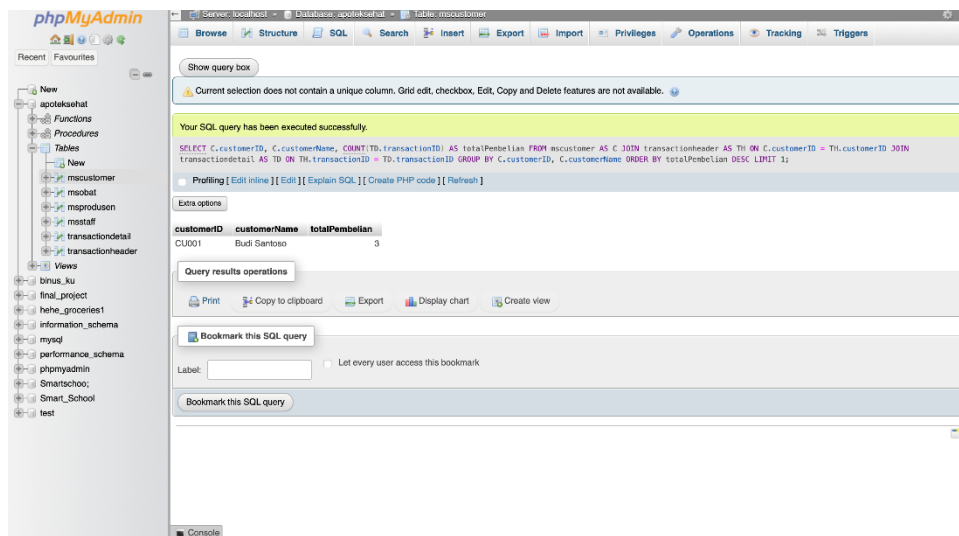
O.obatID, O.obatName

ORDER BY

TotalKuantitasPembelian DESC;

10. Buat query untuk menampilkan pembeli yang paling sering melakukan pembelian

Hasil :



Query :

SELECT

C.customerID,

C.customerName,

COUNT(TD.transactionID) AS totalPembelian

FROM

mscustomer AS C

JOIN

transactionheader AS TH ON C.customerID = TH.customerID

JOIN

transactiondetail AS TD ON TH.transactionID = TD.transactionID

GROUP BY

C.customerID, C.customerName

ORDER BY

totalPembelian DESC

LIMIT 1;

BAB IV

KESIMPULAN

Kesimpulan penelitian ini menunjukkan bahwa pembangunan sistem database untuk manajemen pembelian obat di apotek sangat penting mengingat pertumbuhan cepat dalam industri farmasi dan pelayanan kesehatan. Dengan mengatasi kendala seperti pengelolaan stok obat yang kurang optimal, keterbatasan analisis data pembelian, dan birokrasi dalam proses pembelian, sistem ini diharapkan memberikan manfaat berupa peningkatan efisiensi operasional apotek, pemahaman yang lebih baik tentang tren pembelian obat, respons yang lebih cepat terhadap perubahan pasar, dan peningkatan pelayanan pelanggan. Keseluruhan, penelitian ini memberikan kontribusi positif untuk meningkatkan efektivitas manajemen pembelian obat di apotek, menciptakan dasar untuk pelayanan kesehatan yang lebih baik dan responsif terhadap kebutuhan masyarakat.

REFERENSI

- [1] “View of MySQL database management system (DBMS) on FTP site LAPAN Bandung,” *Iaic-publisher.org*. Available: <https://iiast.iaic-publisher.org/ijcitsm/index.php/IJCITSM/article/view/47/16>. [05-12-2023].
- [2] M. E. Murniati and S. Sudarti, “Analisis potensi energi angin sebagai pembangkit energi listrik tenaga angin Di daerah Banyuwangi Kota menggunakan database online-BMKG,” *JURNAL SURYA ENERGY*, vol. 6, no. 1, pp. 9–16, 2023.
- [3] J. Frantiska Jr, “Entity-relationship diagrams,” in *Visualization Tools for Learning Environment Development*, Cham: Springer International Publishing, 2018, pp. 21–30.