

**LAPORAN PROJECT AKHIR
MATA KULIAH BASIS DATA**

SISTEM MANAJEMEN OBAT



Dosen:

Imam Adi Nata, M.Kom.

Disusun Oleh:

2420506022	Faiz Difa Kurniawan
2420506023	Azzahra Febia Mufida
2420506027	Muhammad Ramdhan
2430506058	Nabila Syafiqah Zahran Firlina

**PROGRAM STUDI S-1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR
TAHUN 2025**

DAFTAR ISI

DESKRIPSI APLIKASI	3
SISTEM MANAJEMEN OBAT (APOTEK MEDIKA)	3
ANALISIS KEBUTUHAN DATA.....	3
DIAGRAM ERD	4
1) Before normalization	4
2) After normalization.....	5
NORMALISASI TABEL.....	5
1) Sebelum Normalisasi	5
2) Setelah Normalisasi	5
a. 1NF (First Normal Form).....	5
b. 2NF (Second Normal Form).....	6
c. 3NF (Third Normal Form).....	7
DESAIN SKEMA BASIS DATA RELASIONAL.....	7
1) Before normalization	7
2) After normalization.....	8
IMPLEMENTASI SQL	8
1. CREATE TABLE	8
2. INSERT TABLE	10
3. SELECT DATA.....	12
4. UPDATE DATA	15
5. DELETE DATA	15
6. JOIN	15
7. VIEW	16
8. INDEX	16
9. STORED PROCEDURE	17
10. TRIGGER	17
SIMULASI DATA	17
PEMBAGIAN TUGAS TIAP ANGGOTA.....	20
DOKUMENTASI SCREENSHOOT SQL DAN OUTPUT	21
KESIMPULAN DAN REFLEKSI TIM	21
LAMPIRAN KODE SQL.....	22

DESKRIPSI APLIKASI

SISTEM MANAJEMEN OBAT (APOTEK MEDIKA)

Apotek Medika adalah sistem basis data yang dirancang untuk mendukung aplikasi pemesanan obat secara online, memungkinkan pengguna membeli obat dengan mudah dari rumah. Sistem ini mencakup fitur pencarian obat berdasarkan nama atau kategori, penambahan ke keranjang, pemesanan, pembayaran, dan pelacakan pengiriman. Untuk obat yang memerlukan resep, pengguna dapat mengunggah foto resep yang nantinya akan diverifikasi oleh petugas apotek. Sistem ini juga menyimpan data pengguna, data obat, kategori, resep, transaksi pemesanan, pembayaran, dan pengiriman.

Sistem ini bertujuan untuk mempermudah akses masyarakat terhadap layanan apotek dengan cara yang lebih cepat, aman, dan efisien. Selain membantu pengguna, Apotek Medika juga mendukung operasional apotek melalui digitalisasi penjualan dan pengelolaan data yang terpusat, seperti pengelolaan stok obat dan laporan transaksi. Dengan demikian, sistem ini memberikan solusi praktis dalam mendukung layanan kesehatan modern berbasis teknologi.

ANALISIS KEBUTUHAN DATA

Terdapat beberapa data yang dibutuhkan supaya sistem dapat berjalan sebagaimana mestinya.

1. Data Pelanggan : Berisi informasi pelanggan yang berupa id pelanggan, nama, alamat, dan nomor hp.
2. Data Admin : Berisi informasi mengenai admin yang sedang jaga dimana berupa id admin dan nama admin.
3. Data Obat : Berisi informasi mengenai obat yang tersedia dalam apotek yang berupa id obat, nama obat, jenis obat, stok, dan harga.
4. Data Pemesanan : Berisi informasi mengenai pesanan obat pelanggan yang berupa id pemesanan, id pelanggan, alamat, id admin, tanggal pesan, dan total harga.

5. Data Detail Pemesanan : Berisi informasi lebih detail mengenai pemesanan obat yang berupa id pemesanan, id obat, jumlah obat, dan sub total harga.
6. Data Resep : Berisi informasi mengenai resep yang dikirimkan oleh pelanggan yang nantinya akan divalidasi oleh petugas apotek yang berupa id resep, id pemesanan, nama dokter, dan keterangan.
7. Data Detail Resep : Berisi informasi lebih detail mengenai resep yang dikirimkan pelanggan yang berupa id resep, id obat, dan jumlah obat.
8. Data Pembayaran : Berisi informasi pembayaran yang berupa id pembayaran, id pemesanan, id pelanggan, metode pembayaran, tanggal bayar, dan status bayar.
9. Data Pengiriman : Berisi informasi mengenai pengiriman obat yang berupa id pengiriman, id pemesanan, tanggal pengiriman, status pengiriman (terkirim/belum terkirim), dan alamat pelanggan.

DIAGRAM ERD

1) Before normalization



Diagram ERD di atas adalah diagram ketika belum dilakukan normalisasi, sehingga dapat terlihat masih banyak redundansi data dan anomali yang membuat sistem tidak berjalan dengan baik.

2) After normalization

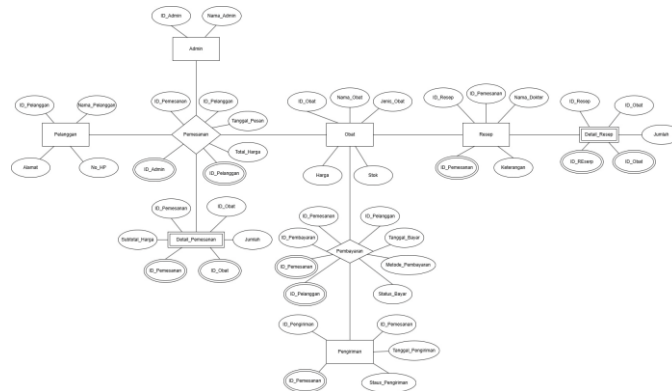


Diagram ERD di atas merupakan hasil dari proses normalisasi **yang telah dilakukan secara optimal**, sehingga struktur data menjadi lebih efisien dan bebas dari anomali.

NORMALISASI TABEL

1) Sebelum Normalisasi

a. Tabel Transaksi Apotek

ID Pemesanan	Tanggal	Nama Pelanggan	Alamat	Nama Obat	Jumlah	Harga	Stok	Resep	Dokter	Jenis Obat	Total	Bayar	Status

Keterangan: Semua data masih dijadikan satu tabel sehingga sangat rawan terjadinya redundansi data dan inkonsistensi data.

2) Setelah Normalisasi

a. 1NF (First Normal Form)

- Tidak ada perubahan pada tabel karena data sudah dipisah per baris (tidak ada multi-value).
- Tetapi, pada tahap 1NF masih memungkinkan terjadinya redundansi data karena masih banyak data yang berulang.
- Tahap 1NF masih rawan terjadi error ketika kita melakukan update data.

b. 2NF (Second Normal Form)

- Pada tahap 2NF, kita akan memisahkan entitas yang bergantung sebagian pada kunci utama menjadi beberapa tabel.
- Pada tahap ini, sudah tidak ada partial dependency sehingga sudah tidak memungkinkan terjadinya redundansi dan inkonsistensi data.
- Namun, pada tahap ini masih ada transitive dependency.
- Pada tahap 2NF, kita membuat beberapa tabel yaitu Tabel Pelanggan, Obat, Admin, Pemesanan, dan Resep.

a. Tabel Pelanggan

ID_Pelanggan	Nama_Pelanggan	Alamat	No_Hp

PK: ID_Pelanggan

b. Tabel Obat

ID_Obat	Nama_Obat	Jenis_Obat	Stok	Harga

PK: ID_Obat

c. Tabel Admin

ID_Admin	Nama_Admin

PK: ID_Admin

d. Tabel Pemesanan

ID_Pemesanan	ID_Obat	Jumlah	Subtotal_Harga

PK: ID_Pemesanan, FK: ID_Obat

e. Tabel Resep

ID_Resep	ID_Pemesanan	Nama_Dokter	Keterangan

PK: ID_Resep, FK: ID_Pemesanan

c. 3NF (Third Normal Form)

- Pada tahap 3NF, kita akan memisahkan entitas yang tergantung tidak langsung pada PK.
- Kita akan menambahkan beberapa tabel yaitu Tabel Detail_Pemesanan, Detail_Resep, Pembayaran, dan Pengiriman.
- Pada tahap ini juga sudah tidak memungkinkan terjadinya redundansi data.
- Pada tahap ini juga, semua kolom bergantung langsung pada primary key masing-masing tabel.
- Untuk lebih detailnya, silakan lihat tabel di bawah ini:

a. Tabel Detail_Pemesanan

ID_Pemesanan	ID_Obat	Jumlah	Subtotal Harga

PK: ID_Pemesanan, FK: ID_Obat

b. Tabel Detail_Resep

ID_Resep	ID_Obat	Jumlah

PK: ID_Resep, FK: ID_Obat

c. Tabel Pembayaran

ID_Pembayaran	ID_Pemesanan	ID_Pelanggan	Metode_Pembayaran	Tanggal_Bayar	Status_Bayar

PK: ID_Pembayaran, FK: ID_Pemesanan, ID_Pelanggan

d. Tabel Pengiriman

ID_Pengiriman	ID_Pemesanan	Tanggal_Pengiriman	Status_Pengiriman	Alamat

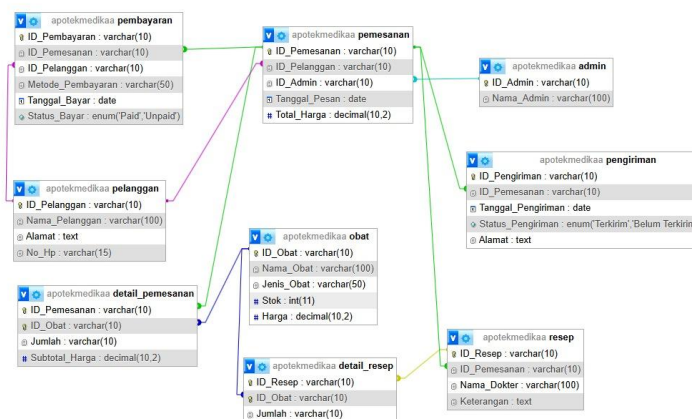
PK: ID_Pengiriman, FK: ID_Pemesanan

DESAIN SKEMA BASIS DATA RELASIONAL

1) Before normalization



2) After normalization



IMPLEMENTASI SQL

1. CREATE TABLE

1) Create Table Admin

```

MariaDB [apotekmedikaa]> CREATE TABLE Admin (
  -> ID_Admin VARCHAR(10) PRIMARY KEY,
  -> Nama_Admin VARCHAR(100)
  -> );
Query OK, 0 rows affected (0.022 sec)

```

2) Create Table Obat

```

MariaDB [apotekmedikaa]> CREATE TABLE Obat (
  -> ID_Obat VARCHAR(10) PRIMARY KEY,
  -> Nama_Obat VARCHAR(100),
  -> Jenis_Obat VARCHAR(50),
  -> Stok INT,
  -> Harga DECIMAL(10,2)
  -> );
Query OK, 0 rows affected (0.011 sec)

```


3) Create Table Detail Pemesanan

```
MariaDB [apotekmedikaa]> CREATE TABLE Detail_Pemesanan (  
  -> ID_Pemesanan VARCHAR(10),  
  -> ID_Obat VARCHAR(10),  
  -> Jumlah VARCHAR(10),  
  -> Subtotal_Harga DECIMAL(10,2),  
  -> PRIMARY KEY (ID_Pemesanan, ID_Obat),  
  -> FOREIGN KEY (ID_Pemesanan) REFERENCES Pemesanan(ID_Pemesanan),  
  -> FOREIGN KEY (ID_Obat) REFERENCES Obat(ID_Obat)  
  -> );  
Query OK, 0 rows affected (0.011 sec)
```

4) Create Table Pembayaran

```
MariaDB [apotekmedikaa]> CREATE TABLE Pembayaran (  
  -> ID_Pembayaran VARCHAR(10) PRIMARY KEY,  
  -> ID_Pemesanan VARCHAR(10),  
  -> ID_Pelanggan VARCHAR(10),  
  -> Metode_Pembayaran VARCHAR(50),  
  -> Tanggal_Bayar DATE,  
  -> Status_Bayar ENUM('Paid', 'Unpaid'),  
  -> FOREIGN KEY (ID_Pemesanan) REFERENCES Pemesanan(ID_Pemesanan),  
  -> FOREIGN KEY (ID_Pelanggan) REFERENCES Pelanggan(ID_Pelanggan)  
  -> );  
Query OK, 0 rows affected (0.018 sec)
```

5) Create Table Pengiriman

```
MariaDB [apotekmedikaa]> CREATE TABLE Pengiriman (  
  -> ID_Pengiriman VARCHAR(10) PRIMARY KEY,  
  -> ID_Pemesanan VARCHAR(10),  
  -> Tanggal_Pengiriman DATE,  
  -> Status_Pengiriman ENUM('Ter kirim', 'Belum Ter kirim'),  
  -> Alamat TEXT,  
  -> FOREIGN KEY (ID_Pemesanan) REFERENCES Pemesanan(ID_Pemesanan)  
  -> );  
Query OK, 0 rows affected (0.013 sec)
```

6) Create Table Resep

```
MariaDB [apotekmedikaa]> CREATE TABLE Resep (  
  -> ID_Resep VARCHAR(10) PRIMARY KEY,  
  -> ID_Pemesanan VARCHAR(10),  
  -> Nama_Dokter VARCHAR(100),  
  -> Keterangan TEXT,  
  -> FOREIGN KEY (ID_Pemesanan) REFERENCES Pemesanan(ID_Pemesanan)  
  -> );  
Query OK, 0 rows affected (0.011 sec)
```

7) Create Table Detail Resep

```

MariaDB [apotekmedikaa]> CREATE TABLE Detail_Resep (
  -> ID_Resep VARCHAR(10),
  -> ID_Obat VARCHAR(10),
  -> Jumlah VARCHAR(10),
  -> PRIMARY KEY (ID_Resep, ID_Obat),
  -> FOREIGN KEY (ID_Resep) REFERENCES Resep(ID_Resep),
  -> FOREIGN KEY (ID_Obat) REFERENCES Obat(ID_Obat)
  -> );
Query OK, 0 rows affected (0.016 sec)

```

8) Create Table Pemesanan

```

MariaDB [apotekmedikaa]> CREATE TABLE Pemesanan (
  -> ID_Pemesanan VARCHAR(10) PRIMARY KEY,
  -> ID_Pelanggan VARCHAR(10),
  -> ID_Admin VARCHAR(10),
  -> Tanggal_Pesan DATE,
  -> Total_Harga DECIMAL(10,2),
  -> FOREIGN KEY (ID_Pelanggan) REFERENCES Pelanggan(ID_Pelanggan),
  -> FOREIGN KEY (ID_Admin) REFERENCES Admin(ID_Admin)
  -> );
Query OK, 0 rows affected (0.011 sec)

```

9) Create Table Pelanggan

```

MariaDB [apotekmedikaa]> CREATE TABLE Pelanggan (
  -> ID_Pelanggan varchar(10) NOT NULL Primary Key,
  -> Nama_Pelanggan varchar(100),
  -> Alamat text,
  -> No_HP varchar(15)
  -> );
Query OK, 0 rows affected (0.040 sec)

```

2. INSERT TABLE

1) Insert Data Pemesanan

```

MariaDB [apotekmedikaa]> INSERT INTO Pemesanan VALUES
  -> ('P1', '2422', 'A1', CURDATE(), '9800'),
  -> ('P2', '2423', 'A2', CURDATE(), '5762'),
  -> ('P3', '2427', 'A2', CURDATE(), '1700'),
  -> ('P4', '2458', 'A1', CURDATE(), '9800');
Query OK, 4 rows affected (0.010 sec)
Records: 4 Duplicates: 0 Warnings: 0

```

2) Insert Data Pelanggan

```

C 2423, Zara, Mertoyudan, ... at line 2
MariaDB [apotekmedikaa]> INSERT INTO Pelanggan VALUES
  -> ('2422', 'Faiz', 'Ngadimulyo', '082420506022'),
  -> ('2423', 'Zara', 'Mertoyudan', '082420506023'),
  -> ('2427', 'Rama', 'Pringsurat', '082420506027'),
  -> ('2458', 'Bila', 'Krasak', '082430506058');
Query OK, 4 rows affected (0.004 sec)
Records: 4 Duplicates: 0 Warnings: 0

```

3) Insert Data Obat

```

MariaDB [apotekmedikaa]> INSERT INTO Obat VALUES
  -> ('01', 'Paracetamol', 'Obat Bebas', 10, '5762'),
  -> ('02', 'Amoxicillin', 'Obat Keras', 10, '9800'),
  -> ('03', 'Morfin', 'Obat Keras Khusus', 10, '1700');
Query OK, 3 rows affected (0.012 sec)
Records: 3 Duplicates: 0 Warnings: 0

```

4) Insert Data Admin

```

MariaDB [apotekmedikaa]> INSERT INTO Admin VALUES
  -> ('A1', 'Agus'),
  -> ('A2', 'Bambang');
Query OK, 2 rows affected (0.009 sec)
Records: 2 Duplicates: 0 Warnings: 0

```

5) Insert Data Detail Pemesanan

```

MariaDB [apotekmedikaa]> INSERT INTO Detail_Pemesanan VALUES
  -> ('P1', '02', '1', '9800'),
  -> ('P2', '01', '1', '5762'),
  -> ('P3', '03', '1', '1700'),
  -> ('P4', '02', '1', '9800');
Query OK, 4 rows affected (0.011 sec)
Records: 4 Duplicates: 0 Warnings: 0

```

6) Insert Data Pembayaran

```

MariaDB [apotekmedikaa]> INSERT INTO Pembayaran VALUES
  -> ('B1', 'P1', '2422', 'Transfer', CURDATE(), 'Paid'),
  -> ('B2', 'P2', '2423', 'Transfer', CURDATE(), 'Paid'),
  -> ('B3', 'P3', '2427', 'Transfer', CURDATE(), 'Paid'),
  -> ('B4', 'P4', '2458', 'Transfer', CURDATE(), 'Paid');
Query OK, 4 rows affected (0.002 sec)
Records: 4 Duplicates: 0 Warnings: 0

```

7) Insert Data Pengiriman

```

MariaDB [apotekmedikaa]> INSERT INTO Pengiriman VALUES
  -> ('K1', 'P1', CURDATE(), 'Ter kirim', 'Ngadimulyo'),
  -> ('K2', 'P2', CURDATE(), 'Ter kirim', 'Mertoyudan'),
  -> ('K3', 'P3', CURDATE(), 'Ter kirim', 'Pringsurat'),
  -> ('K4', 'P4', CURDATE(), 'Ter kirim', 'Krasak');
Query OK, 4 rows affected (0.002 sec)
Records: 4 Duplicates: 0 Warnings: 0

```

8) Insert Data Resep

```
MariaDB [apotekmedikaa]> INSERT INTO Resep VALUES
-> ('R1', 'P1', 'dr. Richard', 'Obat harus habis.'),
-> ('R2', 'P2', 'dr. Ayman', 'Obat harus habis.'),
-> ('R3', 'P3', 'dr. Ikhsan', 'Obat harus habis.'),
-> ('R4', 'P4', 'dr. Farhan', 'Obat harus habis.');
```

Query OK, 4 rows affected (0.010 sec)
Records: 4 Duplicates: 0 Warnings: 0

9) Insert Data Detail Resep

```
MariaDB [apotekmedikaa]> INSERT INTO Detail_Resep VALUES
-> ('R1', 'O2', '1'),
-> ('R2', 'O1', '1'),
-> ('R3', 'O3', '1'),
-> ('R4', 'O2', '1');
```

Query OK, 4 rows affected (0.010 sec)
Records: 4 Duplicates: 0 Warnings: 0

3. SELECT DATA

1) Select Table Obat

```
MariaDB [apotekmedikaa]> SELECT * FROM Obat;
```

ID_Obat	Nama_Obat	Jenis_Obat	Stok	Harga
01	Paracetamol	Obat Bebas	10	5762.00
02	Amoxicillin	Obat Keras	10	9800.00
03	Morfin	Obat Keras Khusus	10	1700.00

3 rows in set (0.000 sec)

2) Select Table Pelanggan

```
MariaDB [apotekmedikaa]> SELECT * FROM Pelanggan;
```

ID_Pelanggan	Nama_Pelanggan	Alamat	No_Hp
2422	Faiz	Ngadimulyo	082420506022
2423	Zara	Mertoyudan	082420506023
2427	Rama	Pringsurat	082420506027
2458	Bila	Krasak	082430506058

4 rows in set (0.000 sec)

3) Select Table Admin

```
MariaDB [apotekmedikaa]> SELECT * FROM Admin;
```

ID_Admin	Nama_Admin
A1	Agus
A2	Bambang
A3	Cahya
A4	Danu
A5	Eka

```
5 rows in set (0.001 sec)
```

4) Select Table Pembayaran

```
MariaDB [apotekmedikaa]> SELECT * FROM pembayaran;
```

ID_Pembayaran	ID_Pemesanan	ID_Pelanggan	Metode_Pembayaran	Tanggal_Bayar	Status_Bayar
B1	P1	2422	Transfer	2025-06-02	Paid
B10	P10	2422	Cash On Delivery	2025-06-04	Paid
B2	P2	2423	Transfer	2025-06-02	Paid
B3	P3	2427	Transfer	2025-06-02	Paid
B4	P4	2458	Transfer	2025-06-02	Paid
B5	P5	2423	Transfer	2025-06-04	Paid
B6	P6	2458	Cash On Delivery	2025-06-04	Paid
B7	P7	2482	Cash On Delivery	2025-06-04	Paid
B8	P8	2481	Transfer	2025-06-04	Paid
B9	P9	2423	Cash On Delivery	2025-06-04	Paid

```
10 rows in set (0.001 sec)
```

5) Select Table Pemesanan

```
MariaDB [apotekmedikaa]> SELECT * FROM pemesanan;
```

ID_Pemesanan	ID_Pelanggan	ID_Admin	Tanggal_Pesan	Total_Harga
P1	2422	A1	2025-06-02	9800.00
P10	2422	A5	2025-06-04	5870.00
P2	2423	A2	2025-06-02	5762.00
P3	2427	A2	2025-06-02	1700.00
P4	2458	A1	2025-06-02	9800.00
P5	2423	A4	2025-06-04	5870.00
P6	2458	A4	2025-06-04	9800.00
P7	2482	A4	2025-06-04	5762.00
P8	2481	A5	2025-06-04	5510.00
P9	2423	A5	2025-06-04	4530.00

```
10 rows in set (0.001 sec)
```

6) Select Table Pengiriman

```
MariaDB [apotekmedikaa]> SELECT * FROM pengiriman;
```

ID_Pengiriman	ID_Pemesanan	Tanggal_Pengiriman	Status_Pengiriman	Alamat
K1	P1	2025-06-02	Terkirim	Ngadimulyo
K10	P10	2025-06-04	Terkirim	Ngadimulyo
K2	P2	2025-06-02	Terkirim	Mertoyudan
K3	P3	2025-06-02	Terkirim	Pringsurat
K4	P4	2025-06-02	Terkirim	Krasak
K5	P5	2025-06-04	Terkirim	Mertoyudan
K6	P6	2025-06-04	Terkirim	Krasak
K7	P7	2025-06-04	Terkirim	Jambesari
K8	P8	2025-06-04	Terkirim	Rejowinangun
K9	P9	2025-06-04	Terkirim	Mertoyudan

```
10 rows in set (0.001 sec)
```

7) Select Table Resep

```
MariaDB [apotekmedikaa]> SELECT * FROM resep;
```

ID_Resep	ID_Pemesanan	Nama_Dokter	Keterangan
R1	P1	dr. Richard	Obat harus habis.
R10	P10	dr. Richard	Obat harus habis.
R2	P2	dr. Ayman	Obat harus habis.
R3	P3	dr. Ikhsan	Obat harus habis.
R4	P4	dr. Farhan	Obat harus habis.
R5	P5	dr. Ayman	-
R6	P6	dr. Ayman	-
R7	P7	dr. Ikhsan	Obat harus habis.
R8	P8	dr. Farhan	Obat harus habis.
R9	P9	dr. Farhan	Obat harus habis.

```
10 rows in set (0.001 sec)
```

8) Select Table Detail Pemesanan

```
MariaDB [apotekmedikaa]> SELECT * FROM detail_pemesanan;
```

ID_Pemesanan	ID_Obat	Jumlah	Subtotal_Harga
P1	02	1	9800.00
P10	05	1	5870.00
P2	01	1	5762.00
P3	03	1	1700.00
P4	02	1	9800.00
P5	05	1	5870.00
P6	02	1	9800.00
P7	01	1	5762.00
P8	01	1	5510.00
P9	04	1	4530.00

```
10 rows in set (0.001 sec)
```

9) Select Table Detail Resep

```
MariaDB [apotekmedikaa]> SELECT * FROM detail_resep;
```

ID_Resep	ID_Obat	Jumlah
R1	02	1
R10	05	1
R2	01	1
R3	03	1
R4	02	1
R5	05	1
R6	02	1
R7	01	1
R8	01	1
R9	04	1

```
10 rows in set (0.001 sec)
```

4. UPDATE DATA

1) Update Data Pelanggan

```
MariaDB [apotekmedikaa]> UPDATE Pelanggan
-> SET Alamat = 'Jambesari'
-> WHERE Nama_Pelanggan = 'Jennie';
Query OK, 1 row affected (0.019 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [apotekmedikaa]> SELECT * FROM Pelanggan;
```

ID_Pelanggan	Nama_Pelanggan	Alamat	No_Hp
2422	Faiz	Ngadimulyo	082420506022
2423	Zara	Mertoyudan	082420506023
2427	Rama	Pringsurat	082420506027
2458	Bila	Krasak	082430506058
2481	Rara	Rejowinangun	2440506081
2482	Jennie	Jambesari	2440506082

```
6 rows in set (0.001 sec)
```

5. DELETE DATA

```
MariaDB [apotekmedikaa]> DELETE FROM pembayaran WHERE metode_pembayaran = 'COD';
Query OK, 0 rows affected (0.012 sec)
```

```
MariaDB [apotekmedikaa]> SELECT * FROM pembayaran WHERE metode_pembayaran = 'COD';
Empty set (0.001 sec)
```

6. JOIN

1) Pelanggan yang membeli Paracetamol

```
MariaDB [apotekmedikaa]> SELECT DISTINCT p>Nama_Pelanggan
-> FROM pelanggan p
-> JOIN pemesanan pe ON p.ID_Pelanggan = pe.ID_Pelanggan
-> JOIN detail_pemesanan dp ON pe.ID_Pemesanan = dp.ID_Pemesanan
-> JOIN obat o ON dp.ID_Obat = o.ID_Obat
-> WHERE o>Nama_Obat = 'Morfin';
+-----+
| Nama_Pelanggan |
+-----+
| Rama           |
+-----+
1 row in set (0.001 sec)
```

2) Pelanggan yang membeli Morfin

```
MariaDB [apotekmedikaa]> SELECT DISTINCT p>Nama_Pelanggan
-> FROM pelanggan p
-> JOIN pemesanan pe ON p.ID_Pelanggan = pe.ID_Pelanggan
-> JOIN detail_pemesanan dp ON pe.ID_Pemesanan = dp.ID_Pemesanan
-> JOIN obat o ON dp.ID_Obat = o.ID_Obat
-> WHERE o>Nama_Obat = 'Morfin';
+-----+
| Nama_Pelanggan |
+-----+
| Rama           |
+-----+
1 row in set (0.001 sec)
```

7. VIEW

```
MariaDB [apotekmedikaa]> CREATE VIEW view_obat_stok AS
-> SELECT id_obat, nama_obat, jenis_obat, harga, stok,
-> CASE
-> WHEN stok = 0 THEN 'Stok Habis'
-> WHEN stok < 10 THEN 'Stok Hampir Habis'
-> ELSE 'Stok Aman'
-> END AS status_stok
-> FROM obat;
Query OK, 0 rows affected (0.012 sec)

MariaDB [apotekmedikaa]> SELECT * FROM view_obat_stok;
+-----+-----+-----+-----+-----+-----+
| id_obat | nama_obat | jenis_obat | harga | stok | status_stok |
+-----+-----+-----+-----+-----+-----+
| 01      | Paracetamol | Obat Bebas | 5762.00 | 10 | Stok Aman |
| 02      | Amoxicillin | Obat Keras | 9800.00 | 10 | Stok Aman |
| 03      | Morfin      | Obat Keras Khusus | 1700.00 | 10 | Stok Aman |
| 04      | Cetirizine  | Obat Bebas Terbatas | 4530.00 | 25 | Stok Aman |
| 05      | Omeprazole  | Obat Keras | 5870.00 | 18 | Stok Aman |
| 06      | Antasida Doen | Obat Bebas | 5510.00 | 15 | Stok Aman |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.002 sec)
```

8. INDEX

```
MariaDB [apotekmedikaa]> CREATE INDEX idx_nama_obat ON obat(nama_obat);
Query OK, 0 rows affected (0.024 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [apotekmedikaa]> SELECT * FROM obat WHERE nama_obat = 'Paracetamol';
+-----+-----+-----+-----+-----+
| ID_Obat | Nama_Obat | Jenis_Obat | Stok | Harga |
+-----+-----+-----+-----+-----+
| 01      | Paracetamol | Obat Bebas | 10 | 5762.00 |
+-----+-----+-----+-----+-----+
1 row in set (0.003 sec)
```


9. STORED PROCEDURE

```
MariaDB [apotekmedikaa]>
MariaDB [apotekmedikaa]> DELIMITER //
MariaDB [apotekmedikaa]>
MariaDB [apotekmedikaa]> CREATE PROCEDURE tambah_pelanggan(
->     IN p_id_pelanggan VARCHAR(10),
->     IN p_nama VARCHAR(100),
->     IN p_alamat VARCHAR(200),
->     IN p_no_hp VARCHAR(15)
-> )
-> BEGIN
->     INSERT INTO pelanggan (id_pelanggan, nama_pelanggan, alamat, no_hp)
->     VALUES (p_id_pelanggan, p_nama, p_alamat, p_no_hp);
-> END //
Query OK, 0 rows affected (0.012 sec)
```

10. TRIGGER

1) Trigger untuk mengurangi stok secara otomatis ketika ada transaksi

```
MariaDB [apotekmedikaa]> DELIMITER //
MariaDB [apotekmedikaa]>
MariaDB [apotekmedikaa]> CREATE TRIGGER kurangi_stok_obat
-> AFTER INSERT ON detail_pemesanan
-> FOR EACH ROW
-> BEGIN
->     UPDATE obat
->     SET stok = stok - NEW.jumlah
->     WHERE id_obat = NEW.id_obat;
-> END //
Query OK, 0 rows affected (0.021 sec)
```

SIMULASI DATA

1. DATA ADMIN

```
MariaDB [apotekmedikaa]> SELECT * FROM Admin;
+-----+-----+
| ID_Admin | Nama_Admin |
+-----+-----+
| A1      | Agus      |
| A2      | Bambang   |
| A3      | Cahya     |
| A4      | Danu      |
| A5      | Eka       |
+-----+-----+
5 rows in set (0.001 sec)
```

2. DATA PELANGGAN

```

MariaDB [apotekmedikaa]> UPDATE Pelanggan
-> SET Alamat = 'Jambesari'
-> WHERE Nama_Pelanggan = 'Jennie';
Query OK, 1 row affected (0.019 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [apotekmedikaa]> SELECT * FROM Pelanggan;
+-----+-----+-----+-----+
| ID_Pelanggan | Nama_Pelanggan | Alamat      | No_Hp      |
+-----+-----+-----+-----+
| 2422         | Faiz           | Ngadimulyo  | 082420506022 |
| 2423         | Zara           | Mertoyudan  | 082420506023 |
| 2427         | Rama           | Pringsurat  | 082420506027 |
| 2458         | Bila           | Krasak      | 082430506058 |
| 2481         | Rara           | Rejowinangun | 2440506081  |
| 2482         | Jennie         | Jambesari   | 2440506082  |
+-----+-----+-----+-----+
6 rows in set (0.001 sec)

```

3. DATA OBAT

```

MariaDB [apotekmedikaa]> SELECT * FROM obat;
+-----+-----+-----+-----+-----+
| ID_Obat | Nama_Obat      | Jenis_Obat      | Stok | Harga |
+-----+-----+-----+-----+-----+
| 01      | Paracetamol    | Obat Bebas      | 10   | 5762.00 |
| 02      | Amoxicillin    | Obat Keras      | 10   | 9800.00 |
| 03      | Morfin         | Obat Keras Khusus | 10   | 1700.00 |
| 04      | Cetirizine     | Obat Bebas Terbatas | 25   | 4530.00 |
| 05      | Omeprazole     | Obat Keras      | 18   | 5870.00 |
| 06      | Antasida Doen  | Obat Bebas      | 15   | 5510.00 |
+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

```

4. DATA RESEP

```

MariaDB [apotekmedikaa]> SELECT * FROM resep;
+-----+-----+-----+-----+
| ID_Resep | ID_Pemesanan | Nama_Dokter | Keterangan |
+-----+-----+-----+-----+
| R1       | P1           | dr. Richard | Obat harus habis. |
| R10      | P10          | dr. Richard | Obat harus habis. |
| R2       | P2           | dr. Ayman   | Obat harus habis. |
| R3       | P3           | dr. Ikhsan  | Obat harus habis. |
| R4       | P4           | dr. Farhan  | Obat harus habis. |
| R5       | P5           | dr. Ayman   | -           |
| R6       | P6           | dr. Ayman   | -           |
| R7       | P7           | dr. Ikhsan  | Obat harus habis. |
| R8       | P8           | dr. Farhan  | Obat harus habis. |
| R9       | P9           | dr. Farhan  | Obat harus habis. |
+-----+-----+-----+-----+
10 rows in set (0.001 sec)

```

5. DATA DETAIL RESEP

```
MariaDB [apotekmedikaa]> SELECT * FROM detail_resep;
```

ID_Resep	ID_Obat	Jumlah
R1	02	1
R10	05	1
R2	01	1
R3	03	1
R4	02	1
R5	05	1
R6	02	1
R7	01	1
R8	01	1
R9	04	1

```
10 rows in set (0.001 sec)
```

6. DATA PEMESANAN

```
MariaDB [apotekmedikaa]> SELECT * FROM pemesanan;
```

ID_Pemesanan	ID_Pelanggan	ID_Admin	Tanggal_Pesan	Total_Harga
P1	2422	A1	2025-06-02	9800.00
P10	2422	A5	2025-06-04	5870.00
P2	2423	A2	2025-06-02	5762.00
P3	2427	A2	2025-06-02	1700.00
P4	2458	A1	2025-06-02	9800.00
P5	2423	A4	2025-06-04	5870.00
P6	2458	A4	2025-06-04	9800.00
P7	2482	A4	2025-06-04	5762.00
P8	2481	A5	2025-06-04	5510.00
P9	2423	A5	2025-06-04	4530.00

```
10 rows in set (0.001 sec)
```

7. DATA DETAIL PEMESANAN

```
MariaDB [apotekmedikaa]> SELECT * FROM detail_pemesanan;
```

ID_Pemesanan	ID_Obat	Jumlah	Subtotal_Harga
P1	02	1	9800.00
P10	05	1	5870.00
P2	01	1	5762.00
P3	03	1	1700.00
P4	02	1	9800.00
P5	05	1	5870.00
P6	02	1	9800.00
P7	01	1	5762.00
P8	01	1	5510.00
P9	04	1	4530.00

```
10 rows in set (0.001 sec)
```

8. DATA PEMBAYARAN

```
MariaDB [apotekmedikaa]> SELECT * FROM pembayaran;
```

ID_Pembayaran	ID_Pemesanan	ID_Pelanggan	Metode_Pembayaran	Tanggal_Bayar	Status_Bayar
B1	P1	2422	Transfer	2025-06-02	Paid
B10	P10	2422	Cash On Delivery	2025-06-04	Paid
B2	P2	2423	Transfer	2025-06-02	Paid
B3	P3	2427	Transfer	2025-06-02	Paid
B4	P4	2458	Transfer	2025-06-02	Paid
B5	P5	2423	Transfer	2025-06-04	Paid
B6	P6	2458	Cash On Delivery	2025-06-04	Paid
B7	P7	2482	Cash On Delivery	2025-06-04	Paid
B8	P8	2481	Transfer	2025-06-04	Paid
B9	P9	2423	Cash On Delivery	2025-06-04	Paid

```
10 rows in set (0.001 sec)
```

9. DATA PENGIRIMAN

```
MariaDB [apotekmedikaa]> SELECT * FROM pengiriman;
```

ID_Pengiriman	ID_Pemesanan	Tanggal_Pengiriman	Status_Pengiriman	Alamat
K1	P1	2025-06-02	Terkirim	Ngadimulyo
K10	P10	2025-06-04	Terkirim	Ngadimulyo
K2	P2	2025-06-02	Terkirim	Mertoyudan
K3	P3	2025-06-02	Terkirim	Pringsurat
K4	P4	2025-06-02	Terkirim	Krasak
K5	P5	2025-06-04	Terkirim	Mertoyudan
K6	P6	2025-06-04	Terkirim	Krasak
K7	P7	2025-06-04	Terkirim	Jambesari
K8	P8	2025-06-04	Terkirim	Rejowinangun
K9	P9	2025-06-04	Terkirim	Mertoyudan

```
10 rows in set (0.001 sec)
```

PEMBAGIAN TUGAS TIAP ANGGOTA

No	Nama	NPM	Keterangan
1.	Faiz Difa Kurniawan	2420506022	Membuat perencanaan, membuat ERD sebelum normalisasi, menyusun laporan, membuat ppt
2.	Azzahra Febia Mufida	2420506023	Membuat perencanaan, membuat ERD sebelum normalisasi, membuat PPT, menyusun laporan
3.	Muhammad Ramdhan	2420506027	Membuat ERD setelah normalisasi, mengunggah dokumentasi screenshot SQL
4.	Nabila Syafiqah Zahran Firlina	2430506058	Membuat database setelah proses normalisasi data, mengimplementasikan SQL, dan menyusun laporan

DOKUMENTASI SCREENSHOOT SQL DAN OUTPUT

<https://drive.google.com/drive/folders/1dYpjkadckQWLjnW4rp1ONN5B5fGd1AD5?usp=sharing>

KESIMPULAN DAN REFLEKSI TIM

- **KESIMPULAN**

Proyek akhir ini bertujuan untuk merancang dan membangun sistem manajemen obat berbasis basis data yang dapat membantu pengelolaan data obat di apotek secara lebih terstruktur dan efisien. Melalui penerapan konsep ERD, normalisasi, serta penggunaan query SQL, sistem ini mampu menangani pencatatan data obat, pengelolaan stok, dan transaksi dengan baik.

Implementasi sistem ini menunjukkan bahwa teknologi basis data berperan penting dalam mendukung akurasi dan efisiensi pengelolaan informasi di bidang farmasi. Dengan pengembangan lebih lanjut, sistem ini berpotensi menjadi solusi digital yang bermanfaat dalam operasional apotek secara menyeluruh.

- **REFLEKSI TIM**

Selama mengerjakan proyek ini, tim kami belajar banyak tentang pentingnya kerja sama, pembagian tugas, dan komunikasi yang baik. Setiap anggota berperan sesuai kemampuan masing-masing, mulai dari perancangan database, penulisan query, hingga pengujian sistem.

Kami juga menyadari bahwa membangun sistem tidak hanya soal teknis, tetapi juga memahami kebutuhan pengguna dan memastikan data dikelola dengan benar. Proyek ini menjadi pengalaman berharga yang meningkatkan pemahaman kami terhadap konsep basis data dan penerapannya dalam dunia nyata.

LAMPIRAN KODE SQL

Github: <https://github.com/nabilaszf/Project-Akhir-Basis-Data/blob/78fd20a3d9fa341994cf34dfe633a2b049cbda35/ApotekMedika.sql>