

**LAPORAN PROJECT AKHIR
IMPLEMENTASI SISTEM DATABASE
UNTUK MENGOPTIMALKAN
PROSES TRANSAKSI PADA WARNET**



**DISUSUN OLEH :
KELOMPOK II**

MOCH. NAUFAL A.R	(2341760148)
MUHAMMAD FARUQ B. N	(2341760197)
FITRI CAHYANIATI	(2341760198)

**Dosen Pengampu :
ELOK NUR HAMDANA S.T., M.T**

**JURUSAN TEKNOLOGI INFORMASI
PRODI SISTEM INFORMASI BISNIS
POLITEKNIK NEGERI MALANG**

2024

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi yang semakin maju, telah membuat perubahan besar pada media digital. Perubahan tersebut diantaranya yaitu dalam bidang sistem informasi, sistem informasi sangat diperlukan bagi instansi serta khalayak umum lainnya.

Untuk membuat sistem informasi yang dapat memberikan manfaat besar, harus disusun secara matang agar nantinya sistem informasi tersebut dapat digunakan sebagaimana mestinya.

Tentunya di dalam sebuah sistem informasi berbasis teknologi pasti ada kaitannya dengan basis data, basis data sangat dibutuhkan di sistem informasi untuk menampung data data dari sistem informasi dalam bentuk tabel dan data tersebut nantinya di olah sehingga menjadi sebuah sistem informasi yang mudah digunakan.

1.2 Tujuan

Tujuan pembuatan laporan ini yaitu untuk memenuhi tugas praktikum pembuatan database sistem informasi. Dan tentunya agar mahasiswa dapat membuat database sistem informasi yang berguna.

1.3 Manfaat

1. Dapat mengetahui cara membuat database MySQL
2. Dapat mengetahui cara membuat tabel di database MySQL
3. Dapat menjalankan perintah SQL untuk CRUD

BAB II

LANDASAN TEORI

1.1 Teori

a. Basis Data

Basis data adalah kumpulan informasi atau data yang dikelola secara teratur di dalam suatu sistem komputer. Data yang dimaksud merupakan hasil dari akses yang dilakukan ketika menggunakan perangkat, seperti computer, atau perangkat lainnya. Komponen dasar ini dijadikan sebagai sebuah sistem yang dapat digunakan untuk memudahkan pengguna dalam pengelolaan informasi yang dibutuhkan. Pada dunia pemrograman basis data sering kali digunakan untuk menyimpan konfigurasi, data pemrograman, dan log hasil dari proses running aplikasi program.

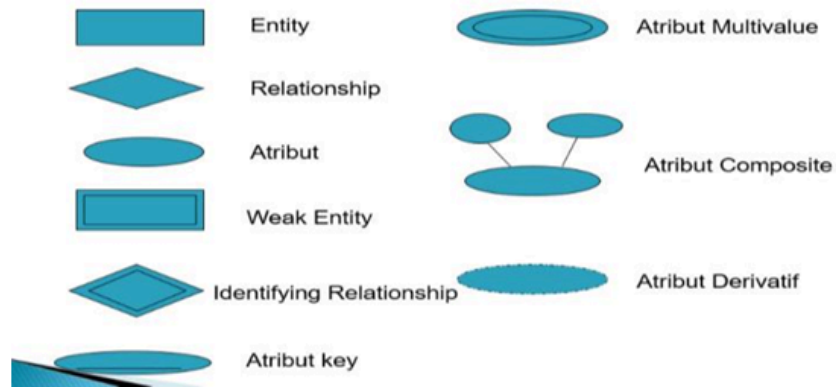
Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data.

Data dalam basis data bersifat integrated dan shared, Terpadu (integrated), berkas-berkas data yang ada pada basis data saling terkait (terjadi dependensi data), lalu untuk Berbagi data (shared), data yang sama dapat dipakai oleh sejumlah pengguna dalam waktu yang bersamaan. Sering dinamakan sebagai sistem multiuser

Beberapa elemen dasar basis data meliputi :

- Tabel: Kumpulan informasi secara logis yang terkait dan diperlakukan sebagai unit. Setiap tabel selalu terdiri atas lajur vertikal yang bisa disebut dengan kolom atribut (column/field) dan lajur horizontal yang bisa disebut dengan baris data (row/record).
- Kunci: Nilai untuk setiap atribut harus berdomain sama, dan setiap baris data harus memiliki kunci utama yang dapat dibedakan atau bersifat unik (tidak kembar).
- Hubungan: Data dalam basis data dapat memiliki hubungan satu-satu, satu-banyak, atau banyak-banyak. Hubungan ini digunakan untuk mengorganisasi data dan menciptakan tahapan dalam proses pengambilan keputusan


b. Entity Relationship Diagram






Entity Relationship Diagram (ERD) adalah sebuah diagram struktural yang digunakan untuk merancang sebuah basis data . ERD akan mendeskripsikan data yang disimpan pada sebuah sistem maupun batasannya. Entity Relationship Diagram adalah suatu alat untuk mempresentasikan model data yang ada pada sistem dimana didalamnya terdapat entity dan relationship yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari dunia nyata yang digambarkan dengan lebih sistematis dengan menggunakan diagram.

ERD memiliki empat konsep utama yaitu :

Tabel 1.1 Simbol *Entity Relationship Diagram*

Nama symbol	Simbol	Deksripsi
Entitas		<p>Simbol entitas digambarkan sebagai persegi panjang. Entitas merupakan objek atau konsep yang ingin disimpan informasinya dalam basis data. Entitas Sebuah entitas dapat berupa orang, tempat, objek, atau kejadian yang dapat dianggap penting bagi sebuah organisasi atau perusahaan. Simbol entitas dalam Entity Relationship Diagram (ERD) digambarkan sebagai persegi panjang. Entitas merupakan objek atau konsep yang ingin disimpan informasinya dalam basis data</p>

Atribut		<p>Simbol atribut digambarkan sebagai oval. Setiap entitas memiliki karakteristik tertentu yang disebut dengan atribut. Atribut berfungsi untuk mendeskripsikan karakteristik yang ada pada entitas yang disimpan dalam basis data . Berdasarkan karakteristik sifatnya, atribut dapat dibedakan menjadi beberapa jenis yaitu simple attribute dan composite attribute, single valued attribute dan multi value attribute, derived attribute, key attribute.</p>
---------	---	--

Relasi		Simbol relasi digambarkan sebagai belah ketupat, Simbol ini digunakan untuk menggambarkan hubungan antara entitas dalam basis data. Simbol ini digunakan untuk menggambarkan hubungan antara entitas dalam basis data. Relasi menunjukkan bagaimana dua entitas berbagi informasi dalam basis data. Relasi ini dapat berupa hubungan satu satu, satu-banyak, atau banyak banyak.
Garis		Simbol garis pada Entity Relationship Diagram (ERD) digunakan untuk menghubungkan antara entitas, atribut, dan relasi. Garis ini menunjukkan hubungan antar elemen-elemen dalam ERD, simbol garis dapat menghubungkan antara entitas dengan atributnya, antara entitas dengan entitas lain, serta antara entitas dengan relasi. Dengan adanya garis ini, ERD dapat memperjelas hubungan dan alur antar elemen-elemen yang ada dalam basis data.

Kardinalitas dalam relasi pada *Entity Relationship Diagram* (ERD) menggambarkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Kardinalitas ini menunjukkan seberapa banyak entitas dari satu himpunan entitas dapat terhubung dengan entitas dari himpunan entitas lain dalam suatu relasi. Kardinalitas relasi dapat dibagi menjadi tiga tipe, yaitu:

1. *One to One (1-1)*

Setiap entitas pada himpunan entitas 1 dapat berelasi dengan paling banyak satu entitas pada himpunan entitas 2, demikian juga sebaliknya.

2. *One to Many (1-N)*

Setiap entitas pada himpunan entitas 1 dapat berelasi dengan banyak entitas pada himpunan entitas 2, tetapi tidak sebaliknya atau setiap entitas pada himpunan entitas 1 hanya dapat berelasi satu entitas pada himpunan entitas 2, tetapi tidak sebaliknya.

3. *Many to Many (N-N)*

Setiap entitas pada himpunan entitas 1 dapat berelasi dengan banyak entitas pada himpunan entitas 2, Demikian juga sebaliknya.

c. Pemetaan ERD ke Model Relasional

□ Relasi Basis Data

Skema Relasi merupakan nama relasi didefinisikan oleh himpunan pasangan atribut dan nama domain, sedangkan Skema Basis Data relasional merupakan himpunan skema relasi dengan nama yang berbeda.

□ Sifat-sifat Relasi

Relasi dalam basis data model relasional memiliki sifat sebagai berikut :

- Nama relasi berbeda satu sama lain dalam skema relasional
- Setiap sel(baris,kolom) dari relasi berisi satu nilai atomik atau nilai tunggal
- Setiap atribut memiliki nama yang berbeda
- Nilai suatu atribut berasal dari domain yang sama
- Setiap tuple adalah berbeda, dan tidak ada duplikasi tuple

□ **Relational Key**

Relational key dalam basis data model relasional terdiri dari:

- Superkey
Sebuah atribut (atau kombinasi atribut) secara unik mengenali setiap entitas dalam sebuah tabel.
- Candidate key
Sebuah superkey minimal, yaitu superkey yang tidak merupakan bagian atribut dari suatu superkey.
- Primary key
Candidate key yang terpilih untuk mengenali secara unik seluruh nilai atribut pada sebuah baris. Tidak boleh kosong.
- Secondary key
Sebuah atribut (atau kombinasi atribut) secara paksa digunakan untuk tujuan pengambilan data.
- Foreign key
Sebuah atribut (atau kombinasi atribut) dalam sebuah tabel dimana nilainya cocok dengan primary key pada tabel lainnya.

BAB III

PEMBAHASAN

1. ERD

ERD Database Kasir Warnet

ATRIBUT DAN ENTITY :

- Pengguna
 - ID_Pengguna 'PRIMARY KEY'
 - Nama
- Komputer
 - ID_Komputer 'PRIMARY KEY'
 - Spesifikasi
 - Status
 - Lokasi
- Transaksi
 - ID_Transaksi 'PRIMARY KEY'
 - ID_Karyawan 'FK'
 - ID_Pengguna 'FOREIGN KEY'
 - ID_Komputer 'FK'
 - ID_Paket 'FK'
 - ID_Penggunaan_Layanan 'FK'
 - Tanggal
 - Metode_Pembayaran
 - Total_Pembayaran
- Paket
 - ID_Paket 'PRIMARY KEY'
 - Jenis_Paket
 - Harga
 - Durasi
- Penggunaan_Paket
 - ID_Penggunaan 'PRIMARY KEY'
 - ID_Pengguna 'FOREIGN KEY'

□ ID_Paket 'FOREIGN KEY'

□ Waktu_Mulai

□ Waktu_Akhir

□ Total Waktu

□ Waktu Tersisa

- Layanan_Tambahan

□ ID_Layanan 'PRIMARY KEY'

□ Jenis_Layanan

□ Harga

- Penggunaan_Layanan

□ ID_Penggunaan_Layanan 'PRIMARY KEY'

□ ID_Pengguna 'FOREIGN KEY'

□ ID_Layanan 'FOREIGN KEY'

□ Jumlah

□ Total Layanan

- Karyawan

□ ID_Karyawan 'PRIMARY KEY'

□ ID_Shif 'FK'

□ Nama

□ Posisi

- Shift_Kerja

□ ID_Shift 'PRIMARY KEY'

□ ID_Karyawan 'FOREIGN KEY'

□ Tanggal

□ Shift

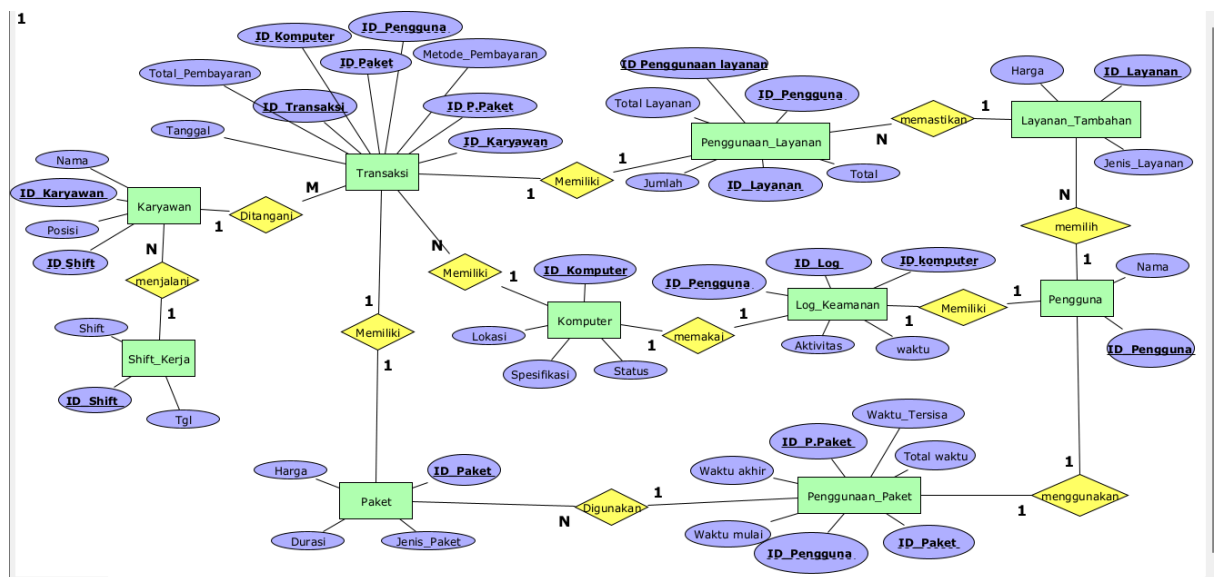
- Log_Keamanan

□ ID_Log 'PRIMARY KEY'

- ID_Pengguna 'FOREIGN KEY'
- ID Komputer 'FK'
- Aktivitas
- Waktu

Relasi Antar Entitas

- Satu pengguna memilih banyak layanan tambahan
- Satu layanan tambahan dapat memastikan banyak penggunaan layanan
- Satu penggunaan layanan mempunyai satu transaksi
- Setiap satu computer memiliki banyak transaksi
- Satu pengguna dapat memiliki banyak log keamanan
- Satu log keamanan memakai Satu komputer
- Satu penggunaan paket dapat digunakan dalam banyak paket
- Satu pengguna dapat menggunakan satu penggunaan paket
- Satu Paket memiliki satu transaksi
- Setiap banyaknya transaksi hanya dapat ditangani oleh satu karyawan
- Banyaknya karyawan hanya bisa menjalani satu shift kerja



2. Normalisasi Data 1NF hingga 3NF

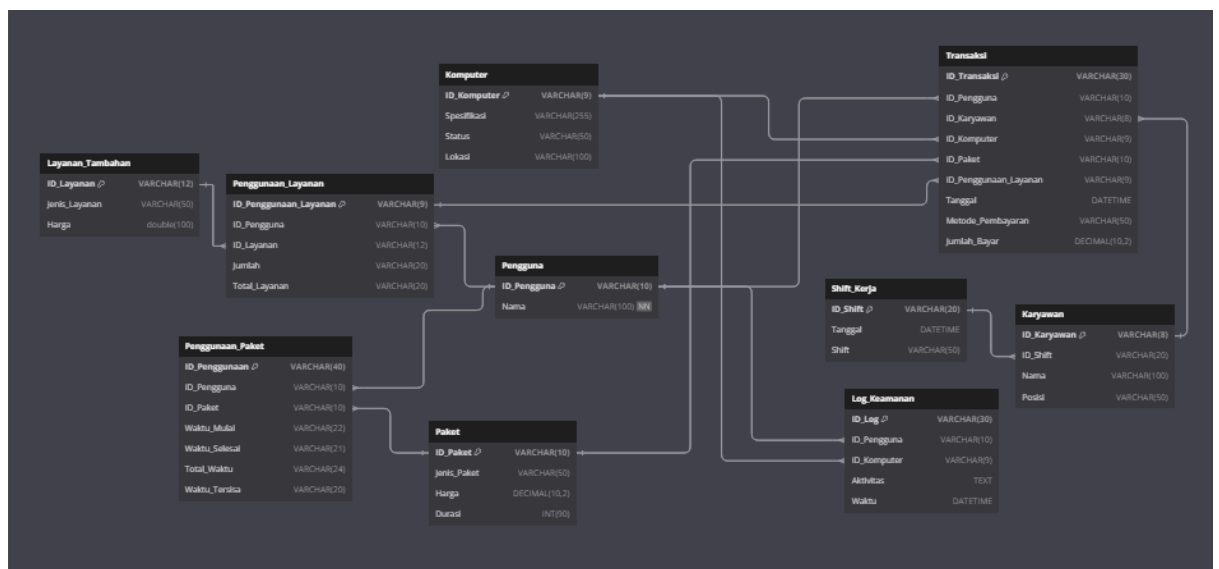
- A. Setiap tabel dalam 1NF (First Normal Form) harus memenuhi tiga kondisi:
 - a. Setiap kolom berisi hanya satu nilai.
 - b. Setiap kolom memiliki tipe data yang sama.
 - c. Setiap kolom memiliki nilai yang unik atau nilai NULL (tidak boleh ada duplikasi baris).
- B. 2NF (Second Normal Form):
 - a. Tabel harus berada dalam 1NF.

- Tabel harus berada dalam 2NF.
- Tidak ada ketergantungan transitif antara atribut non-primer dan kunci primer.
- Tabel lebih lanjut dibagi untuk memastikan bahwa setiap atribut non-primer hanya bergantung pada kunci primer.

Pengunaan_Paket						
ID_Pengunaan	ID_Pengguna	ID_Paket	waktu mulai	waktu akhir	total waktu	Waktu Tersisa
1	P1	1	07.00	10.00	180	60
2	P2	2	10.00	12.00	120	10
3	P3	3	12.00	14.00	120	40
4	P4	1	14.00	17.00	180	25
5	P5	2	17.00	19.00	120	120
6	P6	3	22.00	07.00	540	180

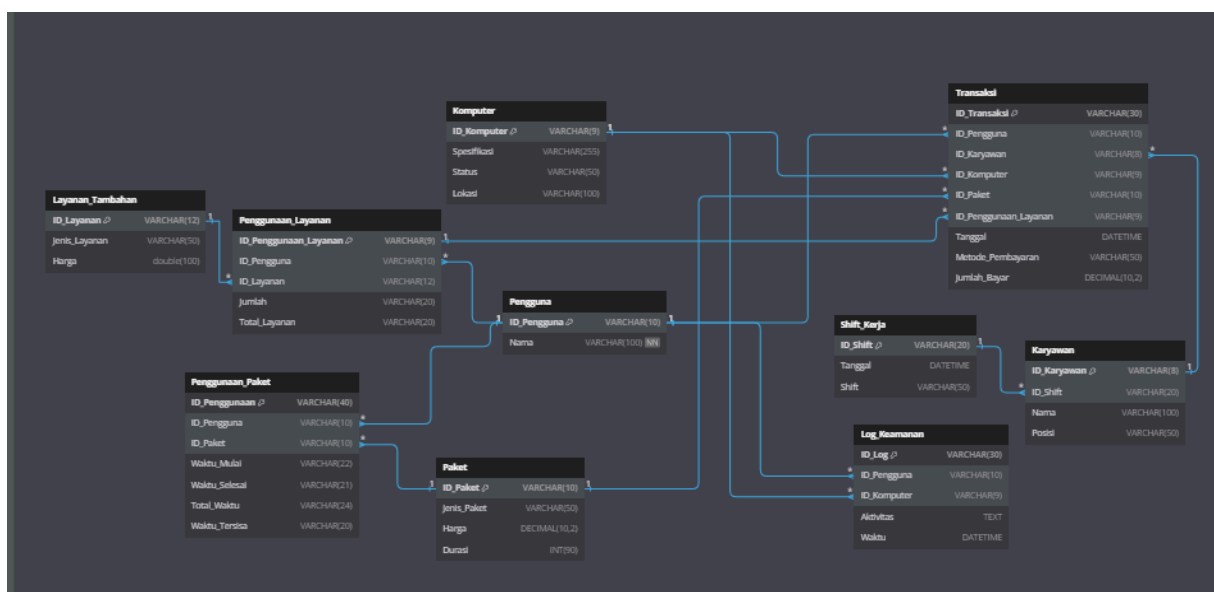
□ CDM (Conceptual Data Model)

- CDM adalah representasi abstrak dari struktur data yang menggambarkan entitas, atribut, dan hubungan di antara entitas tersebut tanpa memperhatikan bagaimana data akan diimplementasikan dalam database.
- Tujuan utama CDM adalah untuk memberikan pemahaman umum tentang data dan bagaimana data tersebut saling berhubungan.
- CDM biasanya digunakan dalam tahap awal perancangan sistem untuk berkomunikasi dengan stakeholder dan untuk mendefinisikan persyaratan data secara keseluruhan.
- CDM menggunakan notasi seperti ERD (Entity-Relationship Diagram) untuk menggambarkan entitas dan hubungan mereka.



□ PDM (Physical Data Model)

- PDM adalah representasi rinci dari struktur data yang menggambarkan bagaimana data akan disimpan dalam sistem database.
- PDM mencakup tabel, kolom, tipe data, indeks, dan constraint yang spesifik untuk DBMS (Database Management System) yang akan digunakan.
- PDM digunakan untuk merancang implementasi fisik dari database, termasuk pengoptimalan kinerja dan penyimpanan data.
- PDM memperhatikan aspek teknis seperti normalisasi, denormalisasi, partisi tabel, dan pengaturan indexing.



3. Penerapan Query Basis Data Terapan

Pembuatan Database

Query ini merupakan langkah awal dalam pembuatan database, yaitu dengan menggunakan perintah CREATE dilanjutkan nama database yang diinginkan yaitu 'database warnet'. Jika database telah dibuat maka selanjutnya menampilkan database yang telah dibuat sebelumnya. DBMS merupakan perantara antara user dengan database.

Penerapan DDL pada :

A. DATABASE

- ❑ **CREATE DATABASE nama_database;** Merupakan perintah untuk membuat database baru, contoh:
CREATE DATABASE nama;
- ❑ **SHOW DATABASES;** Merupakan perintah untuk menampilkan daftar nama database yang terdapat dalam database server.
- ❑ **USE nama_database;** Merupakan perintah untuk masuk kedalam database yang akan digunakan. Sebelum membuat tabel yang digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan satu database sebagai database aktif. Contoh untuk dapat menggunakan/manipulasi database db_polinema, maka gunakan perintah berikut:
USE db_polinema;

- a. Membuat sebuah database dengan nama database warnet

```
MariaDB [(none)]> create database warnet;  
Query OK, 1 row affected (0.009 sec)
```

- b. Menampilkan tabel Untuk menampilkan daftar nama-nama tabel yang terdapat dalam database yang sedang aktif/digunakan, menggunakan perintah : **SHOW TABLES;**

```
MariaDB [warnet]> show tables;  
+-----+  
| Tables_in_warnet |  
+-----+  
| karyawan         |  
| komputer         |  
| layanan_tambahan |  
| log_keamanan     |  
| paket            |  
| pengguna         |  
| penggunaan_layanan |  
| penggunaan_paket |  
| shift_kerja      |  
| transaksi        |  
+-----+  
10 rows in set (0.001 sec)
```

- c. Membuat Tabel Nama tabel tidak boleh mengandung spasi (space). Ketika membuat tabel, ada beberapa yang harus dideklarasikan dalam pembuatannya yaitu antara lain meliputi : nama tabel, nama kolom (field), tipe data dari field dan panjang data. Perintah yang digunakan untuk membuat tabel secara umum adalah sebagai berikut :

CREATE TABLE namatabel (field1 typedata1, field2 typedata2);

- Tabel pengguna

```
CREATE TABLE Pengguna (  
    ID_Pengguna INT AUTO_INCREMENT PRIMARY KEY,  
    Nama VARCHAR(100) NOT NULL,  
    Waktu_Mulai DATETIME,  
    Waktu_Akhir DATETIME,  
    ID_paket INT );
```

```
MariaDB [(none)]> create database warnet;  
Query OK, 1 row affected (0.009 sec)  
  
MariaDB [(none)]> use warnet;  
Database changed  
MariaDB [warnet]> CREATE TABLE `Pengguna` (  
-> `ID_Pengguna` INT PRIMARY KEY AUTO_INCREMENT,  
-> `ID_Komputer` INT,  
-> `Nama` VARCHAR(100) NOT NULL,  
-> `Waktu_Mulai` DATETIME,  
-> `Waktu_Akhir` DATETIME,  
-> `ID_Paket` INT  
-> );  
Query OK, 0 rows affected (0.010 sec)
```

- Tabel Komputer

```
CREATE TABLE Komputer (  
    ID_Komputer VARCHAR(9) PRIMARY KEY,  
    Spesifikasi VARCHAR(255),  
    Status VARCHAR(50),  
    Lokasi VARCHAR(100)  
);
```

```
MariaDB [warnet]>  
MariaDB [warnet]> CREATE TABLE `Komputer` (  
-> `ID_Komputer` VARCHAR(9) PRIMARY KEY,  
-> `Spesifikasi` VARCHAR(255),  
-> `Status` VARCHAR(50),  
-> `Lokasi` VARCHAR(100)  
-> );  
Query OK, 0 rows affected (0.005 sec)
```

- **Tabel Transaksi**

```
CREATE TABLE Transaksi (
    ID_Transaksi VARCHAR(30) PRIMARY KEY,
    ID_Pengguna VARCHAR(10),
    ID_Karyawan VARCHAR(8),
    ID_Komputer VARCHAR(9),
    ID_paket VARCHAR(10),
    ID_Pengguna Layanan VARCHAR(10),
    Tanggal DATETIME,
    Metode_Pembayaran VARCHAR(50),
    Jumlah_Bayar DECIMAL(10, 2),
);
```

```
MariaDB [warnet]>
MariaDB [warnet]> CREATE TABLE `Transaksi` (
  -> `ID_Transaksi` VARCHAR(30) PRIMARY KEY,
  -> `ID_Pengguna` VARCHAR(10),
  -> `ID_Karyawan` VARCHAR(8),
  -> `ID_Komputer` VARCHAR(9),
  -> `ID_Paket` VARCHAR(10),
  -> `ID_Penggunaan_Layanan` VARCHAR(9),
  -> `Tanggal` DATETIME,
  -> `Metode_Pembayaran` VARCHAR(50),
  -> `Jumlah_Bayar` DECIMAL(10,2)
  -> );
Query OK, 0 rows affected (0.006 sec)
```

- **Tabel Paket**

```
CREATE TABLE Paket (
    ID_Paket VARCHAR(10) PRIMARY KEY,
    Jenis_Paket VARCHAR(50),
    Harga DECIMAL(10, 2),
    Durasi INT (90)
);
```

```
MariaDB [warnet]>
MariaDB [warnet]> CREATE TABLE `Paket` (
  -> `ID_Paket` VARCHAR(10) PRIMARY KEY,
  -> `Jenis_Paket` VARCHAR(50),
  -> `Harga` DECIMAL(10,2),
  -> `Durasi` INT(90)
  -> );
Query OK, 0 rows affected (0.006 sec)
```


- **Tabel Penggunaan_Paket**

```
CREATE TABLE Penggunaan_Paket (
  ID_Penggunaan VARCHAR(40) PRIMARY KEY,
  ID_Pengguna VARCHAR(10),
  ID_Paket VARCHAR(10),
  Waktu_Mulai VARCHAR(22),
  Waktu_Selesai VARCHAR(21),
  Total Waktu VARCHAR(24),
  Waktu_Tersisa VARCHAR(20),
);
```

```
MariaDB [warnet]>
MariaDB [warnet]> CREATE TABLE `Penggunaan_Paket` (
  -> `ID_Penggunaan` VARCHAR(40) PRIMARY KEY,
  -> `ID_Pengguna` VARCHAR(10),
  -> `ID_Paket` VARCHAR(10),
  -> `Waktu_Mulai` VARCHAR(22),
  -> `Waktu_Selesai` VARCHAR(21),
  -> `Total_Waktu` VARCHAR(24),
  -> `Waktu_Tersisa` VARCHAR(20)
  -> );
Query OK, 0 rows affected (0.005 sec)
```

- **Tabel Layanan_Tambahan**

```
CREATE TABLE Layanan_Tambahan (
  ID_Layanan INT AUTO_INCREMENT PRIMARY KEY,
  Jenis_Layanan VARCHAR(50),
  Harga DECIMAL(10, 2)
);
```

```
MariaDB [warnet]>
MariaDB [warnet]> CREATE TABLE `Layanan_Tambahan` (
  -> `ID_Layanan` INT PRIMARY KEY AUTO_INCREMENT,
  -> `Jenis_Layanan` VARCHAR(50),
  -> `Harga` DECIMAL(10,2)
  -> );
Query OK, 0 rows affected (0.010 sec)
```

- **Tabel Karyawan**

```
CREATE TABLE Karyawan (
  ID_Karyawan VARCHAR(8) PRIMARY KEY,
  ID_Shift VARCHAR(20),
  Nama VARCHAR(100),
  Posisi VARCHAR(50)
);
```

```

MariaDB [warnet]>
MariaDB [warnet]> CREATE TABLE `Karyawan` (
  -> `ID_Karyawan` VARCHAR(8) PRIMARY KEY,
  -> `ID_Shift` VARCHAR(20),
  -> `Nama` VARCHAR(100),
  -> `Posisi` VARCHAR(50)
  -> );
Query OK, 0 rows affected (0.006 sec)

```

- **Tabel Shift_Kerja**

```

CREATE TABLE Shift_Kerja (
  ID_Shift VARCHAR(20) PRIMARY KEY,
  Tanggal DATETIME,
  Shift VARCHAR(50),
);

```

```

MariaDB [warnet]>
MariaDB [warnet]> CREATE TABLE `Shift_Kerja` (
  -> `ID_Shift` VARCHAR(20) PRIMARY KEY,
  -> `Tanggal` DATETIME,
  -> `Shift` VARCHAR(50)
  -> );
Query OK, 0 rows affected (0.006 sec)

```

- **Tabel Log_Keamanan**

```

CREATE TABLE Log_Keamanan (
  ID_Log VARCHAR(30) PRIMARY KEY,
  ID_Pengguna VARCHAR(10),
  ID_Komputer VARCHAR(9),
  Aktivitas TEXT,
  Waktu DATETIME,
);

```

```

MariaDB [warnet]>
MariaDB [warnet]> CREATE TABLE `Log_Keamanan` (
  -> `ID_Log` VARCHAR(30) PRIMARY KEY,
  -> `ID_Pengguna` VARCHAR(10),
  -> `ID_Komputer` VARCHAR(9),
  -> `Aktivitas` TEXT,
  -> `Waktu` DATETIME
  -> );
Query OK, 0 rows affected (0.005 sec)

```

d. **Perintah SQL untuk menambahkan kunci asing (foreign key) pada tabel dalam database MariaDB**

Primary key dibuat menggunakan alter table, ALTER TABLE. **ALTER TABLE** adalah perintah SQL yang digunakan untuk mengubah struktur tabel yang sudah ada di database. Dengan menggunakan ALTER TABLE, kita bisa melakukan berbagai perubahan pada tabel, seperti menambahkan kolom baru, mengubah tipe data kolom, menghapus kolom, menambahkan atau menghapus constraint, dan lain sebagainya.

- ALTER TABLE Karyawan ADD FOREIGN KEY (ID_Shift) REFERENCES Shift_Kerja (ID_Shift);

Menambahkan kunci asing ID_Shift pada tabel Karyawan yang merujuk pada kolom ID_Shift di tabel Shift_Kerja.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Karyawan' ADD FOREIGN KEY ('ID_Shift') REFERENCES 'Shift_Kerja' ('ID_Shift');
Query OK, 0 rows affected (0.033 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Log_Keamanan ADD FOREIGN KEY (ID_Komputer) REFERENCES Komputer (ID_Komputer);

Menambahkan kunci asing ID_Komputer pada tabel Log_Keamanan yang merujuk pada kolom ID_Komputer di tabel Komputer.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Log_Keamanan' ADD FOREIGN KEY ('ID_Komputer') REFERENCES 'Komputer' ('ID_Komputer');
Query OK, 0 rows affected (0.029 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Transaksi ADD FOREIGN KEY (ID_Pengguna) REFERENCES Pengguna (ID_Pengguna);

Menambahkan kunci asing ID_Pengguna pada tabel Transaksi yang merujuk pada kolom ID_Pengguna di tabel Pengguna.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Transaksi' ADD FOREIGN KEY ('ID_Pengguna') REFERENCES 'Pengguna' ('ID_Pengguna');
Query OK, 0 rows affected (0.027 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Transaksi ADD FOREIGN KEY (ID_Karyawan) REFERENCES Karyawan (ID_Karyawan);

Menambahkan kunci asing ID_Karyawan pada tabel Transaksi yang merujuk pada kolom ID_Karyawan di tabel Karyawan.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Transaksi' ADD FOREIGN KEY ('ID_Karyawan') REFERENCES 'Karyawan' ('ID_Karyawan');
Query OK, 0 rows affected (0.030 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Transaksi ADD FOREIGN KEY (ID_Komputer) REFERENCES Komputer (ID_Komputer);
Menambahkan kunci asing ID_Komputer pada tabel Transaksi yang merujuk pada kolom ID_Komputer di tabel Komputer.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Transaksi' ADD FOREIGN KEY ('ID_Komputer') REFERENCES 'Komputer' ('ID_Komputer');
Query OK, 0 rows affected (0.040 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Transaksi ADD FOREIGN KEY (ID_Paket) REFERENCES Paket (ID_Paket);
Menambahkan kunci asing ID_Paket pada tabel Transaksi yang merujuk pada kolom ID_Paket di tabel Paket.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Transaksi' ADD FOREIGN KEY ('ID_Paket') REFERENCES 'Paket' ('ID_Paket');
Query OK, 0 rows affected (0.027 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Transaksi ADD FOREIGN KEY (ID_Penggunaan_Layanan) REFERENCES Penggunaan_Layanan (ID_Penggunaan_Layanan);
Menambahkan kunci asing ID_Penggunaan_Layanan pada tabel Transaksi yang merujuk pada kolom ID_Penggunaan_Layanan di tabel Penggunaan_Layanan.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Transaksi' ADD FOREIGN KEY ('ID_Penggunaan_Layanan') REFERENCES 'Penggunaan_Layanan' (
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Penggunaan_Paket ADD FOREIGN KEY (ID_Pengguna) REFERENCES Pengguna (ID_Pengguna);
Menambahkan kunci asing ID_Pengguna pada tabel Penggunaan_Paket yang merujuk pada kolom ID_Pengguna di tabel Pengguna.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Penggunaan_Paket' ADD FOREIGN KEY ('ID_Pengguna') REFERENCES 'Pengguna' ('ID_Pengguna'
Query OK, 0 rows affected (0.024 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Penggunaan_Paket ADD FOREIGN KEY (ID_Paket) REFERENCES Paket (ID_Paket);
Menambahkan kunci asing ID_Paket pada tabel Penggunaan_Paket yang merujuk pada kolom ID_Paket di tabel Paket.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Penggunaan_Paket' ADD FOREIGN KEY ('ID_Paket') REFERENCES 'Paket' ('ID_Paket');
Query OK, 0 rows affected (0.023 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Penggunaan_Layanan ADD FOREIGN KEY (ID_Pengguna) REFERENCES Pengguna (ID_Pengguna);

Menambahkan kunci asing ID_Pengguna pada tabel Penggunaan_Layanan yang merujuk pada kolom ID_Pengguna di tabel Pengguna.

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Penggunaan_Layanan' ADD FOREIGN KEY ('ID_Pengguna') REFERENCES 'Pengguna' ('ID_Pengguna');
Query OK, 0 rows affected (0.029 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- ALTER TABLE Log keamanan ADD FOREIGN (ID Pengguna) REFERENCES Pengguna (ID Pengguna);

Menambahkan kunci asing ID Pengguna pada table Log Keamanan yang merujuk pada kolom ID Pengguna di table Log Keamanan

```
MariaDB [warnet]>
MariaDB [warnet]> ALTER TABLE 'Log_Keamanan' ADD FOREIGN KEY ('ID_Pengguna') REFERENCES 'Pengguna' ('ID_Pengguna');
Query OK, 0 rows affected (0.027 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- Mengisi Data / Input Data ke dalam tabel Untuk memasukkan sebuah baris (tuple/record) kedalam tabel MySQL adalah sebagai berikut : INSERT INTO table [(column1, column2,...) VALUES (value1, value2,...)];

- **Tabel Pengguna**

```
MariaDB [warnet]> INSERT INTO Pengguna (ID_Pengguna, Nama) VALUES
-> ('P001', 'John Doe'),
-> ('P002', 'Jane Smith'),
-> ('P003', 'Alice Lee');
Query OK, 3 rows affected (0.008 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Komputer**

```
MariaDB [warnet]> INSERT INTO Komputer (ID_Komputer, Spesifikasi, Status, Lokasi) VALUES
-> ('K0001', 'Intel Core i7, 16GB RAM, 1TB HDD', 'Available', 'Lab 1'),
-> ('K0002', 'AMD Ryzen 5, 8GB RAM, 512GB SSD', 'In Use', 'Lab 2'),
-> ('K0003', 'Intel Core i5, 12GB RAM, 256GB SSD', 'Available', 'Lab 3');
Query OK, 3 rows affected (0.002 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Paket**

```
MariaDB [warnet]> INSERT INTO Paket (ID_Paket, Jenis_Paket, Harga, Durasi) VALUES
-> ('PKT001', 'Basic', 50.00, 1),
-> ('PKT002', 'Premium', 75.00, 2),
-> ('PKT003', 'Pro', 100.00, 3);
Query OK, 3 rows affected (0.008 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Layanan Tambahan**

```
MariaDB [warnet]> INSERT INTO Layanan_Tambahan (ID_Layanan, Jenis_Layanan, Harga) VALUES
-> ('LAY001', 'Print', 10.00),
-> ('LAY002', 'Scan', 5.00),
-> ('LAY003', 'Copy', 3.00);
Query OK, 3 rows affected (0.007 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Penggunaan Layanan**

```
MariaDB [warnet]> INSERT INTO Penggunaan_Layanan (ID_Penggunaan_Layanan, ID_Pengguna, ID_Layanan, Jumlah, Total_Layanan) VALUES
-> ('L001', 'P001', 'LAY001', '5', '50.00'),
-> ('L002', 'P002', 'LAY002', '2', '10.00'),
-> ('L003', 'P003', 'LAY001', '3', '30.00');
Query OK, 3 rows affected (0.004 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Log Keamanan**

```
MariaDB [warnet]> INSERT INTO Log_Keamanan (ID_Log, ID_Pengguna, ID_Komputer, Aktivitas, Waktu) VALUES
-> ('L0001', 'P001', 'K0001', 'Login', '2024-06-01 08:55:00'),
-> ('L0002', 'P002', 'K0002', 'Logout', '2024-06-02 12:35:00'),
-> ('L0003', 'P003', 'K0003', 'File access', '2024-06-03 14:00:00');
Query OK, 3 rows affected (0.002 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Shift Kerja**

```
MariaDB [warnet]> INSERT INTO Shift_Kerja (ID_Shift, Tanggal, Shift) VALUES
-> ('SH001', '2024-06-01', 'Morning'),
-> ('SH002', '2024-06-02', 'Afternoon'),
-> ('SH003', '2024-06-03', 'Evening');
Query OK, 3 rows affected (0.008 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Karyawan**

```
MariaDB [warnet]> INSERT INTO Karyawan (ID_Karyawan, ID_Shift, Nama, Posisi) VALUES
-> ('KAR001', 'SH001', 'Michael', 'Manager'),
-> ('KAR002', 'SH002', 'Emily', 'Assistant'),
-> ('KAR003', 'SH003', 'David', 'Technician');
Query OK, 3 rows affected (0.007 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- **Tabel Penggunaan paket**

```
3 rows in set (0.000 sec)
MariaDB [warnet]> INSERT INTO 'penggunaan_paket' ('ID_Penggunaan', 'ID_Pengguna', 'ID_Paket', 'Waktu_Mulai', 'Waktu_Selesai', 'Total_Waktu', 'Waktu_Tersisa') VALUES
-> ('PENG001', 'P001', 'PKT001', '2024-06-01 08:00:00', '2024-06-01 09:00:00', '1 hour', '0 hours'),
-> ('PENG002', 'P002', 'PKT002', '2024-06-02 10:00:00', '2024-06-02 12:00:00', '2 hours', '0 hours'),
-> ('PENG003', 'P003', 'PKT003', '2024-06-03 13:00:00', '2024-06-03 16:00:00', '3 hours', '0 hours');
Query OK, 3 rows affected (0.003 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- Tabel Transaksi

```
MariaDB [warnet]> INSERT INTO Transaksi (ID_Transaksi, ID_Pengguna, ID_Karyawan, ID_Komputer, ID_Paket, ID_Penggunaan_Layanan, Tanggal, Metode_Pembayaran, Jumlah_Bayar) VALUES
-> ('T0001', 'P001', 'KAR001', 'K0001', 'PKT001', 'L001', '2024-06-01 09:00:00', 'Cash', 50.00),
-> ('T0002', 'P002', 'KAR002', 'K0002', 'PKT002', 'L002', '2024-06-02 10:30:00', 'Credit Card', 75.00),
-> ('T0003', 'P003', 'KAR003', 'K0003', 'PKT001', 'L003', '2024-06-03 13:45:00', 'Debit Card', 100.00);
Query OK, 3 rows affected (0.002 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

- Tabel Penggunaan Layanan

```
MariaDB [warnet]> SELECT * FROM PENGGUNAAN_LAYANAN;
```

ID_Penggunaan_Layanan	ID_Pengguna	ID_Layanan	Jumlah	Total_Layanan
L001	P001	LAY001	5	50.00
L002	P002	LAY002	2	10.00
L003	P003	LAY001	3	30.00

3 rows in set (0.000 sec)

- f. Query merupakan operasi yang melibatkan satu atau lebih tabel untuk melakukan retrieval data. Pengambilan data dilakukan dengan beberapa klausa :

□ **SELECT *** untuk memilih semua kolom Format, **SELECT * FROM**

table_name

SELECT adalah salah satu perintah SQL yang paling dasar dan paling sering digunakan.

FROM : Memilih nama sebuah table

- **SELECT * FROM Transaksi**

```
MariaDB [warnet]> SELECT * FROM TRANSAKSI;
```

ID_Transaksi	ID_Pengguna	ID_Karyawan	ID_Komputer	ID_Paket	ID_Penggunaan_Layanan	Tanggal	Metode_Pembayaran	Jumlah_Bayar
T0001	P001	KAR001	K0001	PKT001	L001	2024-06-01 09:00:00	Cash	50.00
T0002	P002	KAR002	K0002	PKT002	L002	2024-06-02 10:30:00	Credit Card	75.00
T0003	P003	KAR003	K0003	PKT001	L003	2024-06-03 13:45:00	Debit Card	100.00

3 rows in set (0.000 sec)

- **SELECT * FROM LAYANAN_TAMBAHAN**

```
MariaDB [warnet]> SELECT * FROM LAYANAN_TAMBAHAN;
```

ID_Layanan	Jenis_Layanan	Harga
LAY001	Print	10
LAY002	Scan	5
LAY003	Copy	3

3 rows in set (0.000 sec)

- **SELECT * FROM KARYAWAN**

```
MariaDB [warnet]> SELECT * FROM KARYAWAN;
```

ID_Karyawan	ID_Shift	Nama	Posisi
KAR001	SH001	Michael	Manager
KAR002	SH002	Emily	Assistant
KAR003	SH003	David	Technician

3 rows in set (0.000 sec)

- **SELECT * FROM KOMPUTER**

```
MariaDB [warnet]> SELECT * FROM KOMPUTER;
```

ID_Komputer	Spesifikasi	Status	Lokasi
K0001	Intel Core i7, 16GB RAM, 1TB HDD	Available	Lab 1
K0002	AMD Ryzen 5, 8GB RAM, 512GB SSD	In Use	Lab 2
K0003	Intel Core i5, 12GB RAM, 256GB SSD	Available	Lab 3

3 rows in set (0.000 sec)

- **SELECT * FROM PAKET**

```
MariaDB [warnet]> SELECT * FROM PAKET;
```

ID_Paket	Jenis_Paket	Harga	Durasi
PKT001	Basic	50.00	1
PKT002	Premium	75.00	2
PKT003	Pro	100.00	3

3 rows in set (0.000 sec)

- **SELECT * FROM LAYANAN TAMBAHAN**

```
MariaDB [warnet]> SELECT * FROM LAYANAN_TAMBAHAN;
```

ID_Layanan	Jenis_Layanan	Harga
LAY001	Print	10
LAY002	Scan	5
LAY003	Copy	3

3 rows in set (0.000 sec)

- **SELECT * FROM SHIFT KERJA**

```
MariaDB [warnet]> SELECT * FROM SHIFT_KERJA;
```

ID_Shift	Tanggal	Shift
SH001	2024-06-01 00:00:00	Morning
SH002	2024-06-02 00:00:00	Afternoon
SH003	2024-06-03 00:00:00	Evening

3 rows in set (0.000 sec)

- **SELECT * FROM PENGGUNAAN LAYANAN**

```
MariaDB [warnet]> SELECT * FROM PENGGUNAAN_LAYANAN;
```

ID_Penggunaan_Layanan	ID_Pengguna	ID_Layanan	Jumlah	Total_Layanan
L001	P001	LAY001	5	50.00
L002	P002	LAY002	2	10.00
L003	P003	LAY001	3	30.00

3 rows in set (0.000 sec)

- **INNER JOIN UNTUK MENGECEK APAKAH PELANGGAN SUDAH MEMBAYAR ATAU BELUM**

```
MariaDB [warnet]> SELECT pengguna.ID_Pengguna, pengguna>Nama, transaksi.Jumlah_Bayar
-> FROM pengguna
-> LEFT JOIN transaksi ON pengguna.ID_Pengguna = transaksi.ID_Pengguna;
```

ID_Pengguna	Nama	Jumlah_Bayar
P001	Tri	50.00
P002	Awed	75.00
P003	Dito	100.00
P004	Tri	10000.00
P004	Nopal	NULL
P005	Faruq	NULL

6 rows in set (0.000 sec)

```
MariaDB [warnet]> select * from transaksi;
ERROR 1146 (42S02): Table 'warnet.transaksi' doesn't exist
MariaDB [warnet]> select * from transaksi;
```

ID_Transaksi	ID_Pengguna	ID_Karyawan	ID_Komputer	ID_Paket	ID_Penggunaan_Layanan	Tanggal	Metode_Pembayaran	Jumlah_Bayar
T0001	P001	KAR001	K0001	PKT001	L001	2024-06-01 09:00:00	Cash	50.00
T0002	P002	KAR002	K0002	PKT002	L002	2024-06-02 10:30:00	Credit Card	75.00
T0003	P003	KAR003	K0003	PKT001	L003	2024-06-03 13:45:00	Debit Card	100.00
T0004	P001	KAR001	K0001	PKT002	L001	2024-05-09 08:12:05	Cash	10000.00

4 rows in set (0.000 sec)

```
MariaDB [warnet]> SELECT
-> t1.ID_Pengguna,
-> t1.ID_Transaksi AS Transaksi1,
-> t2.ID_Transaksi AS Transaksi2,
-> t1.Tanggal AS Tanggal_Transaksi1,
-> t2.Tanggal AS Tanggal_Transaksi2
-> FROM
-> transaksi t1
-> INNER JOIN
-> transaksi t2 ON t1.ID_Pengguna = t2.ID_Pengguna
-> WHERE
-> t1.Tanggal < t2.Tanggal;
```

ID_Pengguna	Transaksi1	Transaksi2	Tanggal_Transaksi1	Tanggal_Transaksi2
P001	T0004	T0001	2024-05-09 08:12:05	2024-06-01 09:00:00

1 row in set (0.000 sec)

```
MariaDB [warnet]> SELECT pengguna.ID_Pengguna, pengguna>Nama, transaksi.Jumlah_Bayar
-> FROM pengguna
-> RIGHT JOIN transaksi ON pengguna.ID_Pengguna = transaksi.ID_Pengguna;
```

ID_Pengguna	Nama	Jumlah_Bayar
P001	Tri	50.00
P002	Awed	75.00
P003	Dito	100.00
P001	Tri	10000.00

4 rows in set (0.000 sec)

```
MariaDB [warnet]> SELECT pengguna.ID_Pengguna, pengguna>Nama, transaksi.Jumlah_Bayar
-> FROM pengguna
-> LEFT JOIN transaksi ON pengguna.ID_Pengguna = transaksi.ID_Pengguna
-> UNION
-> SELECT pengguna.ID_Pengguna, pengguna>Nama, transaksi.Jumlah_Bayar
-> FROM transaksi
-> LEFT JOIN pengguna ON pengguna.ID_Pengguna = transaksi.ID_Pengguna;
```

ID_Pengguna	Nama	Jumlah_Bayar
P001	Tri	50.00
P002	Awed	75.00
P003	Dito	100.00
P001	Tri	10000.00
P004	Nopal	NULL
P005	Faruq	NULL

6 rows in set (0.001 sec)

```
MariaDB [warnet]> |
```

- INNER JOIN MELIHAT TRANSAKSI PEMBAYARAN

```
MariaDB [warnet]> SELECT pengguna.ID_Pengguna, pengguna>Nama, transaksi.Jumlah_Bayar
-> FROM pengguna
-> INNER JOIN transaksi ON pengguna.ID_Pengguna = transaksi.ID_Pengguna;
```

ID_Pengguna	Nama	Jumlah_Bayar
P001	Tri	50.00
P002	Awed	75.00
P003	Dito	100.00
P001	Tri	10000.00

4 rows in set (0.000 sec)

- INNER JOIN MELIHAT DETAIL TRANSAKSI PEMBAYARAN

```
MariaDB [warnet]> SELECT
-> t1.ID_Transaksi AS Transaksi1,
-> t2.ID_Transaksi AS Transaksi2,
-> t1.ID_Pengguna,
-> t1.Tanggal AS Tanggal_Transaksi1,
-> t2.Tanggal AS Tanggal_Transaksi2
-> FROM
-> transaksi t1
-> INNER JOIN
-> transaksi t2 ON t1.ID_Pengguna = t2.ID_Pengguna
-> WHERE
-> t1.ID_Transaksi <> t2.ID_Transaksi;
```

Transaksi1	Transaksi2	ID_Pengguna	Tanggal_Transaksi1	Tanggal_Transaksi2
T0004	T0001	P001	2024-05-09 08:12:05	2024-06-01 09:00:00
T0001	T0004	P001	2024-06-01 09:00:00	2024-05-09 08:12:05

2 rows in set (0.000 sec)

```
MariaDB [warnet]> SELECT
-> t1.ID_Transaksi AS Transaksi1,
-> t2.ID_Transaksi AS Transaksi2,
-> t1.ID_Karyawan,
-> t1.Tanggal AS Tanggal_Transaksi1,
-> t2.Tanggal AS Tanggal_Transaksi2
-> FROM
-> transaksi t1
-> INNER JOIN
-> transaksi t2 ON t1.ID_Karyawan = t2.ID_Karyawan
-> WHERE
-> t1.ID_Transaksi <> t2.ID_Transaksi;
```

Transaksi1	Transaksi2	ID_Karyawan	Tanggal_Transaksi1	Tanggal_Transaksi2
T0004	T0001	KAR001	2024-05-09 08:12:05	2024-06-01 09:00:00
T0001	T0004	KAR001	2024-06-01 09:00:00	2024-05-09 08:12:05

2 rows in set (0.000 sec)

- **INNER JOIN MELIHAT DETAIL TOTAL PENDAPATAN/BAYAR**

```
MariaDB [warnet]> SELECT ID_Pengguna, SUM(Jumlah_Bayar) AS Total_Jumlah_Bayar
-> FROM transaksi
-> GROUP BY ID_Pengguna;
```

ID_Pengguna	Total_Jumlah_Bayar
P001	10050.00
P002	75.00
P003	100.00

```
3 rows in set (0.006 sec)
```

- **INNER JOIN MELIHAT DETAIL TOTAL PEMBAYARAN**

```
MariaDB [warnet]> SELECT SUM(Jumlah_Bayar) AS Total_Jumlah_Bayar
-> FROM transaksi;
```

Total_Jumlah_Bayar
10225.00

```
1 row in set (0.000 sec)
```

```
MariaDB [warnet]> |
```

