Praktikum

Basis Data II

PROGRAM STUDI SISTEM INFORMASI

SEMESTER III-24S04

SEKOLAH TINGGI ILMU KOMPUTER TUNAS BANGSA

STIKOM TUNAS BANGSA

TAHUN 2025

TUGAS LAPORAN PRAKTIKUM
PERTEMUAN KE-1 SAMPAI KE-6

Dosen	**:**	**Dr. Dedy Hartama, M.Kom**
Topik	**:**	**DDL, DML, Select Statement, Join, Stored Procedure dan Trigger**
Pertemuan	**:**	**KE-1 SAMPAI KE-6**
CPMK Capaian Pembelajaran Mata Kuliah	**:**	• **Mahasiswa Mampu membuat Query dengan Perintah Membuat Database, Membuat Tabel, Memperbaiki Tabel dan Menghapus Tabel** • **Mahasiswa Mampu membuat Query dengan Perintah Menghubungkan Relasi Tabel dengan Perintah Alter** • **Mahasiswa Mampu membuat Query dengan Perintah Memasukkan Record, Mengedit Record dan Menghapus Record** • **Mahasiswa Mampu menganalisis Persoalan Query Perintah Select Fungsi String dan Tanggal** • **Mahasiswa Mampu menganalisis Persoalan Query Perintah Select dengan Constraint** • **Mahasiswa Mampu menganalisis Persoalan Query Perintah Select dengan Group by dan Having** • **Mahasiswa Mampu menganalisis Persoalan Query Perintah View** • **Mahasiswa mampu membuat join** • **Mahasiswa mampu membuat stored procedure** • **Mahasiswa mampu membuat trigger**

Ketua Kelompok dan Tugas	**:**	**Nama Ketua : Melva Risti Ananta** **Tugas:** • **Mengkoordinasi kelompok** • **Memimpin kelompok** • **Memberikan arahan kelompok** • **Mengawasi proses pengerjaan laporan dari awal hingga akhir** • **Mengetik dan menyatukan laporan** • **Mengecek dan merapikan isi laporan** • **Bagian Pendahuluan, SQL, XAMPP, DDL, DML, DQL, Constraint, Coding, Hasil dan Pembahasan, Join, Stored Procedure, Trigger serta Penutup** • **Memastikan isi laporan agar konsisten** • **Merevisi isi laporan** • **Menjadi penanggungjawab akhir**
Kelas	**:**	**24S04**
Semester	**:**	**Tiga(3)**
Tgl Pertemuan	**:**	**18 September 2025**
Tanggal Selesai dan Revisi	**:**	
Tanggal Acc		
Nilai		**Nilai Kelompok : Rata-Rata Nilai Anggota** **1.** **2.** **3.** **4.** **5.** **6.** **7.** **8.** **9.** **10.**

Nama Anggota dan Tugas	**:**	**Anggota:** **1. Dwi Natasya Bella: Bagian drop ddl, create table, definisi dml, insert dml** **2. Putri Maulidya: Bagian select where union, distinct, group by, and/or/not, aggregate** **3. Elinnah Qori: Bagian definisi dql, select from, select as** **4. Suci Maharani: Bagian definisi ddl, create database, alter ddl** **5. Bunga Ummi Kurniati: Bagian constaint, primary key, foreign key** **6. Nurul Azzahra Sukamto: Bagian update, delete, select dml** **7. Nurlaila: Bagian DML–Delete.** **8. Febri Irwansyah: Bagian select where in, select any, select where betwen, select like** **9. M. Ilham Fajar: Bagian Coding SQL, select order by, limit, where**

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, serta hidayah-Nya sehingga penulis dapat menyelesaikan makalah yang berjudul Structured Query Language (SQL) tepat pada waktunya. Makalah ini disusun sebagai salah satu tugas pada mata kuliah Basis Data di Program Studi Sistem Informasi STIKOM Tunas Bangsa Pematangsiantar.

Adapun tujuan dari penyusunan makalah ini adalah untuk menambah pengetahuan dan pemahaman mengenai konsep dasar Structured Query Language (SQL), termasuk Data Definition Language (DDL), Data Manipulation Language (DML) dan Data Query Language (DQL), serta berbagai perintah penting yang digunakan dalam pengelolaan basis data. Penulis berharap makalah ini dapat menjadi referensi yang bermanfaat bagi mahasiswa maupun pembaca yang ingin mempelajari lebih dalam mengenai SQL. Dalam penyusunan makalah ini, penulis menyadari masih terdapat banyak kekurangan, baik dari segi isi maupun penyajian. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan guna perbaikan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, Dr. Dedy Hartama, S.T., M.Kom., serta semua pihak yang telah memberikan arahan, dukungan, dan motivasi dalam penyelesaian makalah ini. Semoga makalah ini dapat memberikan manfaat dan menambah wawasan bagi kita semua.

Pematangsiantar, 18 September 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR iv
DAFTAR ISI v
BAB 1 2
PENDAHULUAN 2
1.1 Latar Belakang 2
1.2 Tujuan Penelitian 2
1.3 Manfaat Penelitian 2
1.4 Structured Query Language (SQL) 3
1.5 XAMPP 3
1.5.1 Fungsi Utama XAMPP 3
1.5.2 Komponen Penting dalam XAMPP 4
1.5.3 Kelebihan XAMPP 4
1.5.4 Kekurangan XAMPP 4
1.5.5 Langkah-Langkah Menggunakan XAMPP 4
BAB II 7
DEFINITION DATA LANGUAGE (DDL) 7
2.1 Definisi DDL 7
2.2.1 Create 7
2.2.2 Alter 9
2.2.3 Drop 9
BAB III 10
DATA MANIPULATION LANGUAGE (DML) 10
3.1 Definisi DML 10
3.2 Perintah DML 10
3.2.1 Insert 10
3.2.2 Update 11
3.2.3 Delete 12
3.2.4 Select 12

BAB IV 14
DATA QUERY LANGUAGE (DQL) 14
4.1 Definisi DQL 14
4.2 Jenis-Jenis DQL 14
4.2.1 SELECT … FROM 14
4.2.2 SELECT … AS 14
4.2.3 SELECT … ORDER BY 15
4.2.4 SELECT … LIMIT 16
4.2.5 SELECT … WHERE 16
4.2.6 SELECT … WHERE tabel1 = tabel2 17
4.2.7 SELECT … WHERE ... IN 17
4.2.8 SELECT … WHERE ... ANY 18
4.2.9 SELECT … WHERE ... SOME 18
4.2.10 SELECT … WHERE ... BETWEEN 19
4.2.11 SELECT … WHERE ... ALL 19
4.2.12 SELECT … WHERE ... LIKE % DIAKHIR 20
4.2.13 SELECT … WHERE ... LIKE % DIAWAL 20
4.2.14 SELECT … WHERE ... LIKE UNDERSCORE (_) 21
4.2.15 SELECT … WHERE ... LIKE KOMBINASI % DAN UNDERSCORE (_) 21
4.2.17 SELECT … UNION 22
4.2.18 SELECT … LEFT 22
4.2.19 SELECT … FULL JOIN 23
4.2.20 SELECT … DISTINCT 23
4.2.21 SELECT … GROUP BY 23
4.2.22 SELECT … AND / OR / NOT 24
4.2.23 SELECT … AGGREGATE 25
4.2.24 SELECT … AGGREGATE.. COUNT 26
4.2.25 SELECT … AGGREGATE..SUM 26
4.2.26 SELECT … AGGREGATE..AVG 27
4.2.27 SELECT … AGGREGATE..MAX 27
4.2.28 SELECT … AGGREGATE.. MIN 28

BAB V 29
CONSTRAINTS 29
5.1 Definisi Constraints 29
5.2 Jenis-Jenis Constraint 29
5.2.1 Primary Key 29
5.2.2 Foreign Key 31
5.2.3 Unique 31
5.2.4 Not Null 31
5.2.5 Check 32
5.2.6 Default 32
BAB VI 33
HASIL DAN PEMBAHASAN 33
6.1 CPMK DDL dan DML 33
6.2 CPMK DML dan Select Statement 56
BAB VII 73
JOIN 73
7.1 Definisi Join 73
7.2 Tujuan Utama Join 73
BAB VIII 76
STORED PROCEDURE 76
8.1. Definisi Stored Procedure 76
BAB IX 80
TRIGGER 80
9.1. Definisi Trigger 80
BAB X 73
PENUTUP 73
10.1. Kesimpulan 73
10.2. Saran 73
10.3. Dokumentasi 74
DAFTAR PUSTAKA 87

DAFTAR GAMBAR

Gambar 1. 1 Cara Membuka Aplikasi XAMPP 4
Gambar 1. 2 Cara Mengaktifkan Apache 5
Gambar 1. 3 Cara Mengaktifkan MySQL 5
Gambar 1. 4 Cara Mengakses ke phpMyAdmin 6

Gambar 2. 1 Cara Membuat Database 7
Gambar 2. 2 Cara Menggunakan Database 8
Gambar 2. 3 Cara Membuat Tabel 8
Gambar 2. 4 Cara Menampilkan Struktur Tabel 8
Gambar 2. 5 Cara Mengubah Tabel 9
Gambar 2. 6 Cara Menghapus Tabel 9

Gambar 3. 1 Cara Mengisi Tabel 11
Gambar 3. 2 Cara Menampilkan isi Tabel 11
Gambar 3. 3 Cara Memperbarui Data 12
Gambar 3. 4 Cara Menghapus Data 12
Gambar 3. 5 Cara Menampilkan isi Tabel 13

Gambar 4. 1 SELECT…FROM 14
Gambar 4. 2 SELECT…AS 15
Gambar 4. 3 SELECT…ORDER BY 15
Gambar 4. 4 SELECT…ORDER BY (ASC) 16
Gambar 4. 5 SELECT…LIMIT 16
Gambar 4. 6 SELECT…WHERE 17
Gambar 4. 7 SELECT…WHERE TABEL1 = TABEL2 17
Gambar 4. 8 SELECT…WHERE…IN 18
Gambar 4. 9 SELECT…WHERE…ANY 18
Gambar 4. 10 SELECT…WHERE…SOME 19
Gambar 4. 11 SELECT…WHERE…BETWEEN 19
Gambar 4. 12 SELECT…WHERE…ALL 20
Gambar 4. 13 SELECT…WHERE…LIKE…% 20
Gambar 4. 14 SELECT…WHERE…LIKE% 21
Gambar 4. 15 SELECT…WHERE…LIKE_ 21
Gambar 4. 16 SELECT…WHERE…LIKE_% 22
Gambar 4. 17 SELECT…UNION 22
Gambar 4. 18 SELECT…DISTINCT 23
Gambar 4. 19 SELECT…GROUP BY 24
Gambar 4. 20 SELECT…AND 24
Gambar 4. 21 SELECT…OR 25
Gambar 4. 22 SELECT…AGGREGATE 26
Gambar 4. 23 SELECT…AGGREGATE…COUNT 26
Gambar 4. 24 SELECT…AGGREGATE…SUM 27
Gambar 4. 25 SELECT…AGGREGATE…AVG 27

Gambar 4. 26 SELECT…AGGREGATE…MAX 28
Gambar 4. 27 SELECT…AGGREGATE…MIN 28

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat menuntut adanya sistem pengelolaan data yang cepat, tepat, dan akurat. Hampir semua bidang, baik pendidikan, bisnis, kesehatan, maupun pemerintahan, bergantung pada data dalam jumlah besar yang harus diolah secara sistematis. Untuk itu, diperlukan suatu sistem basis data yang mampu menyimpan, mengelola, dan mengolah data dengan baik.

Dalam pengelolaan basis data, dibutuhkan sebuah bahasa standar yang dapat digunakan untuk berinteraksi dengan sistem manajemen basis data relasional. Structured Query Language (SQL) hadir sebagai solusi karena menyediakan berbagai perintah yang memungkinkan pengguna untuk mendefinisikan struktur database, memasukkan data, memperbarui, menghapus, hingga menampilkan data sesuai kebutuhan.

SQL memiliki tiga komponen utama yang sangat penting, yaitu Data Definition Language (DDL), Data Manipulation Language (DML), dan Data Query Language (DQL). DDL digunakan untuk membangun dan mengubah struktur database, DML berfungsi untuk memanipulasi data di dalam tabel, dan DQL berfungsi untuk menampilkan semua data yang ada di dalam tabel maupun kolom. Selain itu, adanya constraint juga memastikan integritas data tetap terjaga.

Dengan memahami SQL secara menyeluruh, mahasiswa tidak hanya mampu mengelola data, tetapi juga dapat menerapkan prinsip-prinsip pengelolaan informasi yang baik. Hal ini menjadi bekal penting dalam menghadapi tantangan era digital yang semakin kompetitif, di mana penguasaan basis data merupakan salah satu keterampilan inti yang dibutuhkan di berbagai bidang.

1.2 Tujuan Penelitian

a. Menjelaskan pengertian dan fungsi SQL dalam basis data.
b. Menyajikan penjelasan serta contoh perintah DDL dan DML.
c. Memahami penerapan constraint dalam database.
d. Memberikan gambaran penggunaan SQL sebagai dasar pengelolaan data.

1.3 Manfaat Penelitian

a. Melatih mahasiswa dalam menuliskan dan memahami perintah-perintah dasar SQL.
b. Menjadi referensi praktis untuk penerapan SQL dalam sistem manajemen basis data.
c. Membantu mahasiswa mempersiapkan diri menghadapi praktik maupun proyek yang berkaitan dengan basis data di dunia kerja.y

1.4 Structured Query Language (SQL)

SQL adalah bahasa pemrograman yang nantinya akan merealisasikan pada beberapa database ataupun tabel dengan database. SQL (Structured Query Language) yang dibaca "sequel" adalah bahasa standar yang digunakan untuk berkomunikasi dengan database relasional, dengan kata lain, SQL merupakan bahasa yang digunakan user untuk berinteraksi dengan database. Secara umum, terdapat komponen utama bahasa SQL. Yang pertama adalah DDL atau Data Definition Language. Seperti arti "Data Definition" yaitu mendefinisikan data, modul ini memungkinkan kita membuat atau memodifikasi tabel dan database, misal statemen CREATE, DROP, atau ALTER. Yang kedua adalah DML atau Data Manipulation Language. Seperti arti "Data Manipulation" yaitu manipulasi data, modul ini memungkinkan kita untuk memanipulasi data, baik mengambil, menambah, mengubah, dan menghapus data pada database, seperti penggunaan statemen SELECT, UPDATE, INSERT, dan DELETE. Yang ketiga adalah DQL atau Data Query Language yang berfungsi untuk menampilkan dan mengambil semua data yang ada di dalam tabel maupun kolom yaitu SELECT saja.

1.5 XAMPP

XAMPP adalah sebuah software package (paket perangkat lunak) yang digunakan untuk membuat server lokal di komputer/laptop. Dengan XAMPP, kita bisa menjalankan website atau aplikasi berbasis web tanpa harus terkoneksi ke internet terlebih dahulu. Dengan adanya XAMPP, pengguna tidak perlu menginstal satu per satu komponen server, karena semuanya sudah terintegrasi dalam satu paket. Hal ini membuat XAMPP sangat populer di kalangan mahasiswa, dosen, maupun developer web untuk proses pembelajaran dan pengembangan aplikasi berbasis web.
Kepanjangan XAMPP adalah:
X = Cross (bisa digunakan di berbagai sistem operasi: Windows, Linux, macOS)
A = Apache (web server utama untuk menjalankan website)
M = MySQL/MariaDB (database server untuk menyimpan data)
P = PHP (bahasa pemrograman server-side)
P = Perl (bahasa pemrograman lain yang juga didukung, meskipun jarang dipakai sekarang)

1.5.1 Fungsi Utama XAMPP

8.1. Membuat server lokal (localhost) → agar kita bisa menguji website sebelum di-upload ke internet.
8.2. Menyediakan database (MySQL/MariaDB) → untuk menyimpan data aplikasi/website.
8.3. Menjalankan bahasa pemrograman server-side (PHP, Perl).
8.4. Memudahkan pengembangan aplikasi web tanpa harus menyewa hosting.

1.5.2 Komponen Penting dalam XAMPP

8.1. Apache → sebagai web server.
8.2. MySQL/MariaDB → sistem manajemen database.
8.3. PHP → bahasa untuk membuat website dinamis.
8.4. phpMyAdmin → aplikasi berbasis web untuk mengelola database dengan mudah.
8.5. FileZilla, Mercury, Tomcat → tambahan untuk keperluan tertentu (FTP, mail server, Java server).

1.5.3 Kelebihan XAMPP

8.1. Gratis dan open source.
8.2. Mudah dipasang dan digunakan.
8.3. Mendukung banyak sistem operasi.
8.4. Praktis untuk belajar pemrograman web (PHP, database, dll).

1.5.4 Kekurangan XAMPP

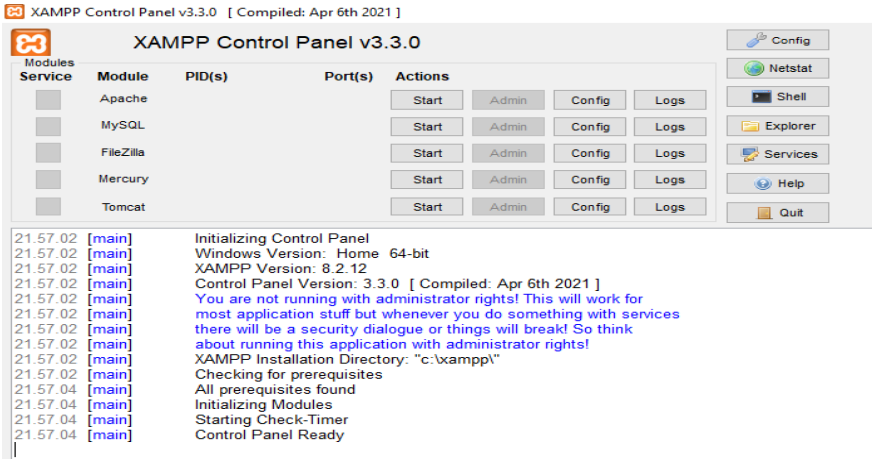
8.1. Tidak seaman server produksi (karena dibuat untuk kebutuhan pengembangan/ujicoba).
8.2. Bisa agak berat di komputer dengan spesifikasi rendah.

1.5.5 Langkah-Langkah Menggunakan XAMPP

Adapun langkah-langkah menggunakan XAMPP, yaitu sebagai berikut:

a. Membuka Aplikasi XAMPP

i. Klik ikon XAMPP Control Panel di desktop atau cari di menu Start.
ii. Setelah terbuka, akan muncul daftar layanan seperti Apache, MySQL, FileZilla, Mercury, dan Tomcat.



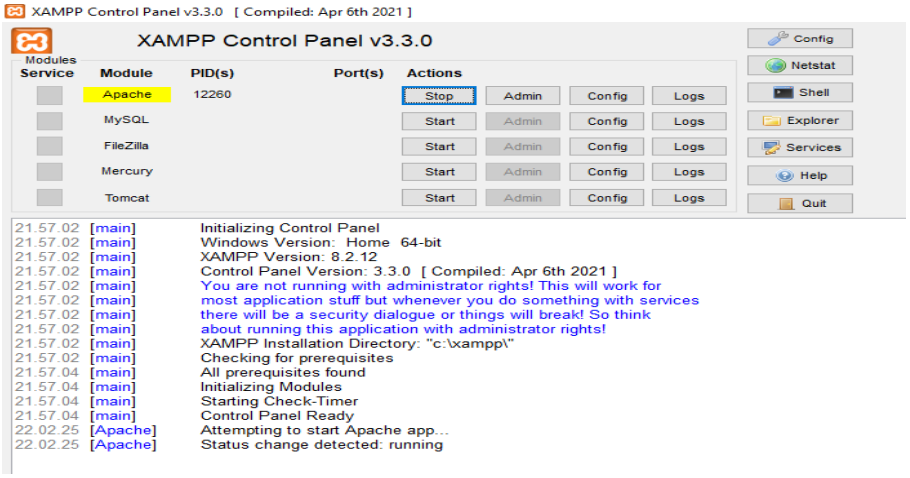
Gambar 1. 1 Cara Membuka Aplikasi XAMPP

b. Mengaktifkan Apache

i. Klik tombol Start pada baris *Apache*.

ii. Jika berhasil, warna background akan berubah hijau dan muncul keterangan *Running*.

iii. Apache ini berfungsi sebagai web server untuk menjalankan file PHP/HTML.



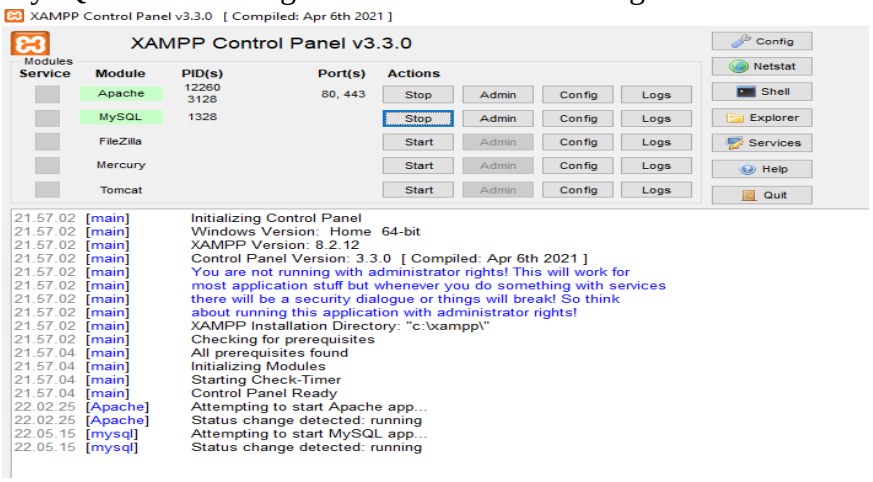
Gambar 1. 2 Cara Mengaktifkan Apache

c. Mengaktifkan MySQL

i. Klik tombol **Start** pada baris *MySQL*.

ii. Sama seperti Apache, indikator akan berubah hijau jika berhasil dijalankan.

iii. MySQL berfungsi untuk mengelola database.



Gambar 1. 3 Cara Mengaktifkan MySQL

d. Mengakses Server Lokal

i. Buka browser (Chrome/Edge/Firefox).

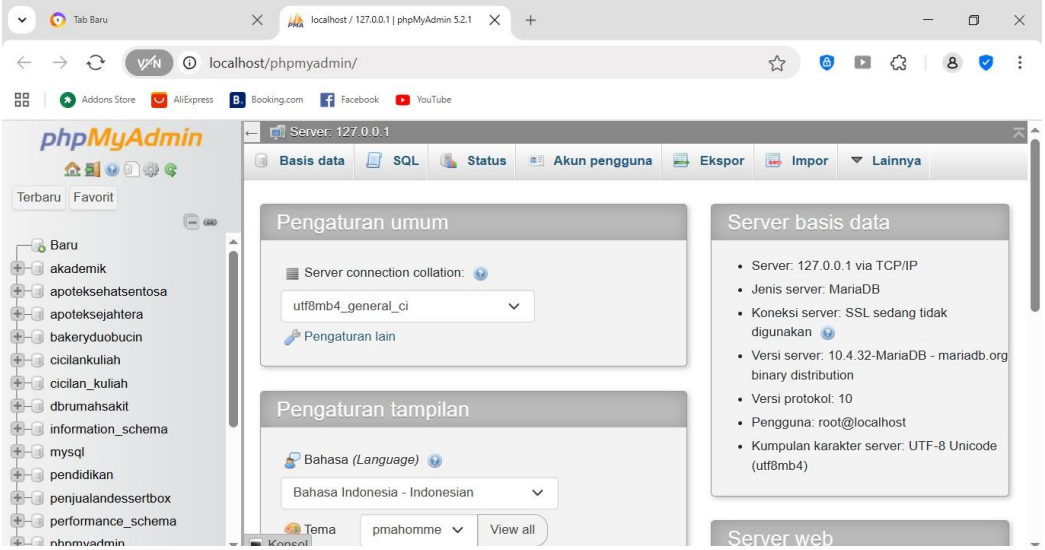
ii. Ketik http://localhost lalu tekan Enter.

iii. Jika server berjalan normal, akan muncul halaman dashboard XAMPP.

e. Mengakses phpMyAdmin

i. Untuk mengelola database, buka http://localhost/phpmyadmin.

ii. Di sana bisa membuat, mengedit, dan menghapus database maupun tabel.



Gambar 1. 4 Cara Mengakses ke phpMyAdmin

f. Menambahkan File Website

i. Simpan file PHP/HTML ke dalam folder htdocs yang ada di direktori instalasi XAMPP (misalnya C:\xampp\htdocs).

ii. Contoh: jika kamu membuat file latihan.php, maka bisa diakses melalui http://localhost/latihan.php.

g. Menutup atau Mematikan Layanan

i. Jika sudah selesai, kembali ke **XAMPP Control Panel**.

ii. Klik **Stop** pada Apache dan MySQL.

iii. Klik tombol **Quit** untuk keluar dari aplikasi.

BAB II

DEFINITION DATA LANGUAGE (DDL)

2.1 Definisi DDL

Data Definition Language adalah bahasa pemrograman yang ada di dalam komputer sehingga bisa dimanfaatkan guna membuat ataupun memodifikasi struktur pada suatu objek pada database yang paling utama untuk pembuatan skema. Data Definition Language merupakan kumpulan perintah yang tersedia dalam SQL (Structure Query Language) yang memberikan gambaran terkait desain database dengan menyeluruh. SQL ini juga disebut dengan MySQL. MySQL merupakan database management system memakai bahasa SQL untuk bahasa yang menghubungkan antara software aplikasi beserta database server. Namun jika kembali pada pembahasan data definition language maka merupakan sekumpulan perintah SQL yang terhubung dengan definisi dari struktur database. Jadi DDL adalah suatu bentuk dari SQL yang dapat dipergunakan untuk menciptakan ataupun membuat suatu database, struktur tabel, tabel, dan lainnya.

2.2 Perintah DDL

Adapun perintah dasar Data Definition Language (DDL) yaitu:

2.2.1 Create

Create adalah salah satu perintah SQL yang termasuk kategori DDL(Data Definition Language). DDL dipakai untuk mendefinisikan dan mengatur struktur database (bukan isi datanya). Jadi, create merupakan pondasi sebelum data bisa dimanipulasi dengan DML. Create terbagi menjadi 2, yaitu:

a. Create Database

Create database digunakan untuk membuat database baru.

Bentuk Umum:

CREATE DATABASE nama_database;
USE nama_database;

Contoh:

CREATE DATABASE cicilan_kuliah;

```
MariaDB [(none)]> CREATE DATABASE Cicilan_Kuliah;
Query OK, 1 row affected (0.021 sec)
```

Gambar 2. 1 Cara Membuat Database

USE cicilan_kuliah;

```
MariaDB [CicilanKuliah]> USE Cicilan_Kuliah;
Database changed
```

Gambar 2. 2 Cara Menggunakan Database

b. Create Table

Create table digunakan untuk membuat table baru di dalam database yang sudah ada.

Bentuk Umum:

CREATE TABLE nama_table (
nama_kolom1 tipe_data1 [constraints1],
nama_kolom2 tipe_data2 [constraints2],
dst...
nama_kolomN tipe_dataN [constraintsN]
);

Contoh:

CREATE TABLE tbprodi (
kdprodi INT(11) PRIMARY KEY,
Namaprodi VARCHAR(50) NOT NULL,
Prog_Pendidikan VARCHAR(10)
);

```
MariaDB [Cicilan_Kuliah]> CREATE TABLE tbprodi (
    ->      kdprodi INT(11) PRIMARY KEY,
    ->      Namaprodi VARCHAR(50) NOT NULL,
    ->      Prog_Pendidikan VARCHAR(10)
    -> );
Query OK, 0 rows affected (0.222 sec)
```

Gambar 2. 3 Cara Membuat Tabel

DESC tbprodi;

```
MariaDB [Cicilan_Kuliah]> DESC tbprodi;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| kdprodi         | int(11)     | NO   | PRI | NULL    |       |
| Namaprodi       | varchar(50) | NO   |     | NULL    |       |
| Prog_Pendidikan | varchar(10) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
3 rows in set (0.087 sec)
```

Gambar 2. 4 Cara Menampilkan Struktur Tabel

2.2.2 Alter

Alter Table dipakai untuk memodifikasi struktur tabel tanpa harus menghapus tabelnya. Dengan ALTER, kita bisa menambah kolom, menghapus kolom, mengganti tipe data, mengganti nama kolom, bahkan menghapus primary key.

Bentuk Umum:

ALTER TABLE nama_tabel

ADD kolom tipe data;

Contoh:

ALTER TABLE tbmahasiswa

ADD Hobbi ENUM(‘membaca’,’menulis’,’olahraga’);

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE tbmahasiswa ADD Hobbi ENUM('membaca','menulis','olahraga');
Query OK, 0 rows affected (0.138 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Gambar 2. 5 Cara Mengubah Tabel

2.2.3 Drop

DROP adalah perintah yang digunakan untuk menghapus objek dalam basis data, seperti tabel, database, view, atau index secara permanen.

Bentuk Umum:

DROP TABLE nama tabel;

Contoh:

DROP TABLE ManajemenUser;

```
MariaDB [Cicilan_Kuliah]> DROP TABLE ManajemenUser;
Query OK, 0 rows affected (0.137 sec)
```

Gambar 2. 6 Cara Menghapus Tabel

BAB III

DATA MANIPULATION LANGUAGE (DML)

3.1 Definisi DML

Data Manipulation Language (DML) adalah salah satu bagian penting dari SQL yang berfungsi untuk mengelola data yang tersimpan di dalam tabel pada sebuah database. Jika diibaratkan, struktur tabel dan database sudah dibangun dengan DDL (Data Definition Language), maka DML hadir sebagai "alat" untuk mengisi, merawat, dan menjaga isi dari tabel tersebut agar selalu relevan dan sesuai kebutuhan pengguna.

Keberadaan DML membuat database menjadi lebih dinamis karena memungkinkan data yang ada di dalamnya terus diperbarui. Hal ini penting mengingat data dalam sistem informasi tidak pernah statis; selalu ada informasi baru yang harus dicatat, informasi lama yang perlu diubah, atau bahkan informasi yang sudah tidak relevan dan harus dihapus.

Dalam praktiknya, DML juga menjadi jembatan utama antara pengguna dengan sistem basis data. Melalui DML, pengguna dapat melakukan berbagai interaksi dengan data tanpa harus mengutak-atik struktur tabel. Karena itulah DML sering disebut sebagai inti dari penggunaan SQL sehari-hari, terutama dalam aplikasi bisnis, pendidikan, kesehatan, maupun pemerintahan yang membutuhkan data up-to-date.

Selain itu, DML juga berperan dalam mendukung analisis data. Dengan data yang dapat diolah secara fleksibel, organisasi dapat menghasilkan laporan, mengevaluasi kinerja, dan mengambil keputusan yang lebih tepat berdasarkan informasi terbaru. Oleh karena itu, pemahaman tentang DML menjadi dasar yang sangat penting bagi mahasiswa maupun praktisi sistem informasi agar mampu bekerja secara efektif dengan data yang mereka kelola.

3.2 Perintah DML

Adapun perintah dasar Data Manipulation Language (DML) yaitu:

3.2.1 Insert

INSERT adalah perintah yang digunakan untuk menambahkan data baru ke dalam tabel pada basis data. Dengan perintah ini, kita bisa memasukkan satu atau beberapa baris data sesuai dengan struktur kolom yang ada pada tabel. **Bentuk Umum:**

INSERT INTO nama_tabel (kolom1, kolom2, kolom3, ...)
VALUES (nilai1, nilai2, nilai3, ...);

Contoh:

INSERT INTO tbprodi (kdprodi, Namaprodi, Prog_Pendidikan) VALUES

(1,'Manajemen Informatika','Diploma 3'),

(2,'Sistem Informasi(SI)','Strata 1'),

(3,'Teknik Informatika','Strata 1'),

(4,'Komputerisasi Akuntansi','Diploma 3');

```
MariaDB [Cicilan_Kuliah]> INSERT INTO tbprodi (kdprodi, Namaprodi, Prog_Pendidikan) VALUES
    -> (1,'Manajemen Informatika','Diploma 3'),
    -> (2,'Sistem Informasi(SI)','Strata 1'),
    -> (3,'Teknik Informatika','Strata 1'),
    -> (4,'Komputerisasi Akuntansi','Diploma 3');
Query OK, 4 rows affected (0.087 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Gambar 3. 1 Cara Mengisi Tabel

SELECT *FROM tbprodi;

```
MariaDB [cicilankuliah]> select *from tbprodi;
+---------+-------------------------+-----------------+
| kdprodi | Namaprodi               | Prog_Pendidikan |
+---------+-------------------------+-----------------+
|       1 | Manajemen Informatika   | Diploma 3       |
|       2 | Sistem Informasi (SI)   | Strata 1        |
|       3 | Teknik Informatika      | Strata 1        |
|       4 | Komputerisasi Akuntansi | Diploma 3       |
+---------+-------------------------+-----------------+
4 rows in set (0.060 sec)
```

Gambar 3. 2 Cara Menampilkan isi Tabel

3.2.2 Update

UPDATE adalah perintah yang digunakan untuk mengubah atau memperbarui data pada tabel yang sudah ada, tanpa harus menambahkan baris baru. Perintah ini biasanya dipakai ketika ada kesalahan data atau terjadi.

Bentuk Umum:
UPDATE nama_tabel
SET kolom1 = nilai_baru1, kolom2 = nilai_baru2, ..., kolomN = nilai_baruN
WHERE Kondisi;

Contoh:
UPDATE transaksi_cicilan
SET UangSejumlah=200000
WHERE KdJenisBayar='JN01';

```
MariaDB [Cicilan_Kuliah]> UPDATE transaksi_cicilan SET UangSejumlah=200000 WHERE KdJenisBayar='JN01';
Query OK, 3 rows affected (0.095 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

Gambar 3. 3 Cara Memperbarui Data

3.2.3 Delete

DELETE adalah perintah yang digunakan untuk menghapus data/baris tertentu dari tabel, sesuai dengan kondisi yang diberikan. Struktur tabelnya tetap ada, hanya isi datanya yang terhapus.

Bentuk Umum:
DELETE FROM nama_tabel
WHERE kondisi;

Contoh:
DELETE FROM transaksi_cicilan
WHERE NoKwitansi='00001';

```
MariaDB [Cicilan_Kuliah]> DELETE FROM transaksi_cicilan WHERE NoKwitansi='00001';
Query OK, 1 row affected (0.104 sec)
```

Gambar 3. 4 Cara Menghapus Data

3.2.4 Select

SELECT adalah perintah yang digunakan untuk mengambil atau meminta data spesifik dari tabel.

Bentuk Umum:
SELECT *FROM nama tabel;

Contoh:
SELECT * FROM transaksi_cicilan;

```
MariaDB [Cicilan_Kuliah]> SELECT * FROM transaksi_cicilan;
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar   | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| 01205      | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |      2000000 | 10000 | JN04         |
| 01422      | 2303007 |       1 | 201     | 2024-02-12 | KS2     | S      |      2000000 | 10000 | JN04         |
| 03851      | 2303016 |       1 | 201     | 2023-02-15 | KS3     | M      |       200000 | 10000 | JN01         |
| 07706      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |       200000 | 10000 | JN01         |
| 07707      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 07710      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 09027      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |       200000 | 10000 | JN01         |
| 09028      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |      1000000 | 10000 | JN02         |
| 09355      | 2303007 |       1 | 201     | 2023-08-08 | KS2     | S      |      4000000 | 10000 | JN03         |
| 14886      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      1000000 | 10000 | JN02         |
| 14887      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      4000000 | 10000 | JN03         |
| 14888      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      2000000 | 10000 | JN04         |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
12 rows in set (0.000 sec)
```

Gambar 3. 5 Cara Menampilkan isi Tabel

BAB IV

DATA QUERY LANGUAGE (DQL)

4.1 Definisi DQL

DQL adalah bagian dari SQL yang digunakan untuk mengambil atau menampilkan data dari tabel dalam database. Perintah utama yang digunakan adalah SELECT. Tujuannya untuk menampilkan informasi sesuai kebutuhan, baik seluruh isi tabel maupun sebagian data berdasarkan kriteria tertentu.

4.2 Jenis-Jenis DQL

4.2.1 SELECT … FROM

Digunakan untuk mengambil data dari tabel tertentu.

Bentuk umum:

SELECT nama_kolom1, nama_kolom2
FROM nama_tabel;

Contoh:

SELECT nama, kota
FROM pelanggan;

```
MariaDB [TokoRoti]> SELECT nama, kota FROM Pelanggan;
+-------+---------+
| nama  | kota    |
+-------+---------+
| Andi  | Jakarta |
| Budi  | Medan   |
| Citra | Jakarta |
| Dewi  | Bandung |
| Eko   | Medan   |
+-------+---------+
5 rows in set (0.069 sec)
```

Gambar 4. 1 SELECT…FROM

4.2.2 SELECT … AS

Memberikan alias/nama lain pada kolom atau tabel agar lebih mudah dibaca.

Bentuk umum:

SELECT nama_kolom AS alias
FROM nama_tabel;

Contoh:
SELECT nama AS Nama_Pelanggan, kota AS Kota Asal
FROM pelanggan;

```
MariaDB [TokoRoti]> SELECT nama AS Nama_Pelanggan, kota AS Kota_Asal FROM Pelanggan;
+----------------+-----------+
| Nama_Pelanggan | Kota_Asal |
+----------------+-----------+
| Andi           | Jakarta   |
| Budi           | Medan     |
| Citra          | Jakarta   |
| Dewi           | Bandung   |
| Eko            | Medan     |
+----------------+-----------+
5 rows in set (0.004 sec)
```

Gambar 4. 2 SELECT…AS

4.2.3 SELECT … ORDER BY

Mengurutkan hasil query berdasarkan kolom tertentu, baik ascending (ASC) maupun descending (DESC).

Sorting Descending artinya menyusun data mulai dari nilai tertinggi ke nilai terendah.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
ORDER BY nama_kolom DESC;

Contoh:
SELECT nama, umur
FROM pelanggan
ORDER BY umur DESC;

```
MariaDB [TokoRoti]> SELECT nama, umur FROM Pelanggan ORDER BY umur DESC;
+-------+------+
| nama  | umur |
+-------+------+
| Eko   |   35 |
| Budi  |   32 |
| Citra |   28 |
| Andi  |   25 |
| Dewi  |   22 |
+-------+------+
5 rows in set (0.034 sec)
```

Gambar 4. 3 SELECT…ORDER BY

Sorting Ascending artinya menyusun data mulai dari nilai terendah ke tertinggi.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
ORDER BY nama_kolom ASC;

Contoh:
SELECT nama, umur
FROM pelanggan
ORDER BY umur ASC;

```
MariaDB [tokoroti]> SELECT nama, umur
    -> FROM pelanggan
    -> ORDER BY umur ASC;
+-------+------+
| nama  | umur |
+-------+------+
| Dewi  |   22 |
| Andi  |   25 |
| Citra |   28 |
| Budi  |   32 |
| Eko   |   35 |
+-------+------+
5 rows in set (0.113 sec)
```

Gambar 4. 4 SELECT…ORDER BY (ASC)

4.2.4 SELECT … LIMIT

Membatasi jumlah baris yang ditampilkan.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
LIMIT jumlah;

Contoh:
SELECT *
FROM pelanggan
LIMIT 3;

```
MariaDB [TokoRoti]> SELECT * FROM Pelanggan LIMIT 3;
+--------------+-------+---------+------+
| id_pelanggan | nama  | kota    | umur |
+--------------+-------+---------+------+
|            1 | Andi  | Jakarta |   25 |
|            2 | Budi  | Medan   |   32 |
|            3 | Citra | Jakarta |   28 |
+--------------+-------+---------+------+
3 rows in set (0.002 sec)
```

Gambar 4. 5 SELECT…LIMIT

4.2.5 SELECT … WHERE

Mengambil data dengan kondisi tertentu.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE kondisi;

Contoh:
SELECT nama, kota
FROM pelanggan
WHERE kota = 'Jakarta';

```
MariaDB [TokoRoti]> SELECT nama, kota FROM Pelanggan WHERE kota = 'Jakarta';
+-------+---------+
| nama  | kota    |
+-------+---------+
| Andi  | Jakarta |
| Citra | Jakarta |
+-------+---------+
2 rows in set (0.057 sec)
```

Gambar 4. 6 SELECT…WHERE

4.2.6 SELECT … WHERE tabel1 = tabel2

Menggabungkan data dari dua tabel berdasarkan kolom yang berhubungan.

Bentuk umum:
SELECT t1.kolom, t2.kolom
FROM tabel1 t1, tabel2 t2
WHERE t1.id = t2.id;

Contoh:
SELECT pelanggan.nama, pesanan.tanggal, pesanan.total
FROM pelanggan, pesanan
WHERE pelanggan.id_pelanggan = pesanan.id_pelanggan;

```
MariaDB [TokoRoti]> SELECT Pelanggan.nama, Pesanan.tanggal, Pesanan.total
    -> FROM Pelanggan, Pesanan
    -> WHERE Pelanggan.id_pelanggan = Pesanan.id_pelanggan;
+-------+------------+--------+
| nama  | tanggal    | total  |
+-------+------------+--------+
| Andi  | 2025-09-01 |  50000 |
| Budi  | 2025-09-02 |  75000 |
| Citra | 2025-09-02 |  60000 |
| Andi  | 2025-09-03 |  90000 |
| Eko   | 2025-09-04 | 120000 |
+-------+------------+--------+
5 rows in set (0.040 sec)
```

Gambar 4. 7 SELECT…WHERE TABEL1 = TABEL2

4.2.7 SELECT … WHERE ... IN

Menyeleksi data dengan mencocokkan nilai dalam suatu daftar.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE nama_kolom IN (nilai1, nilai2, …);

Contoh:
SELECT nama
FROM pelanggan
WHERE kota IN ('Medan', 'Jakarta');

```
MariaDB [TokoRoti]> SELECT nama FROM Pelanggan WHERE kota IN ('Medan','Jakarta');
+-------+
| nama  |
+-------+
| Andi  |
| Budi  |
| Citra |
| Eko   |
+-------+
4 rows in set (0.002 sec)
```

Gambar 4. 8 SELECT…WHERE…IN

4.2.8 SELECT … WHERE ... ANY

Membandingkan nilai kolom dengan himpunan nilai.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE kolom_operator ANY (subquery);

Contoh:
SELECT nama
FROM pelanggan
WHERE umur > ANY (SELECT umur FROM pelanggan WHERE kota = 'Medan');

```
MariaDB [TokoRoti]> SELECT nama FROM Pelanggan
    -> WHERE umur > ANY (SELECT umur FROM Pelanggan WHERE kota = 'Jakarta');
+-------+
| nama  |
+-------+
| Budi  |
| Citra |
| Eko   |
+-------+
3 rows in set (0.025 sec)
```

Gambar 4. 9 SELECT…WHERE…ANY

4.2.9 SELECT … WHERE ... SOME

Minimal ada satu nilai di subquery yang memenuhi kondisi.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE kolom_operator
SOME (subquery);

Contoh:
SELECT nama, umur
FROM Pelanggan
WHERE umur > SOME (
SELECT umur FROM Pelanggan WHERE kota = 'Jakarta'
);

```
MariaDB [tokoroti]> SELECT nama, umur
    -> FROM Pelanggan
    -> WHERE umur > SOME (
    ->     SELECT umur FROM Pelanggan WHERE kota = 'Jakarta'
    -> );
+-------+------+
| nama  | umur |
+-------+------+
| Budi  |   32 |
| Citra |   28 |
| Eko   |   35 |
+-------+------+
3 rows in set (0.123 sec)
```

Gambar 4. 10 SELECT…WHERE…SOME

4.2.10 SELECT … WHERE ... BETWEEN

Memilih data dalam rentang nilai.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE kolom BETWEEN nilai1 AND nilai2;

Contoh:
SELECT nama, umur
FROM pelanggan
WHERE umur BETWEEN 20 AND 30;

```
MariaDB [TokoRoti]> SELECT nama, umur FROM Pelanggan WHERE umur BETWEEN 25 AND 30;
+-------+------+
| nama  | umur |
+-------+------+
| Andi  |   25 |
| Citra |   28 |
+-------+------+
2 rows in set (0.050 sec)
```

Gambar 4. 11 SELECT…WHERE…BETWEEN

4.2.11 SELECT … WHERE ... ALL

Untuk membandingkan suatu nilai dengan semua nilai subquery.

Bentuk umum:
SELECT nama_kolom

FROM nama_tabel
WHERE kolom_operator ALL (subquery);

Contoh:
SELECT nama, umur
FROM Pelanggan
WHERE umur > ALL (
SELECT umur FROM Pelanggan WHERE kota = 'Jakarta'
);

```
MariaDB [tokoroti]> SELECT nama, umur
    -> FROM Pelanggan
    -> WHERE umur > ALL (
    ->     SELECT umur FROM Pelanggan WHERE kota = 'Jakarta'
    -> );
+------+------+
| nama | umur |
+------+------+
| Budi |   32 |
| Eko  |   35 |
+------+------+
2 rows in set (0.001 sec)
```

Gambar 4. 12 SELECT…WHERE…ALL

4.2.12 SELECT … WHERE ... LIKE % DIAKHIR

Mencari data dengan pola tertentu (wildcard: % untuk banyak karakter, _ untuk satu karakter).

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE nama_kolom LIKE 'pola';

Contoh:
SELECT nama
FROM pelanggan
WHERE nama LIKE 'A%';

```
MariaDB [TokoRoti]> SELECT nama FROM Pelanggan WHERE nama LIKE 'A%';
+------+
| nama |
+------+
| Andi |
+------+
1 row in set (0.002 sec)
```

Gambar 4. 13 SELECT…WHERE…LIKE…%

4.2.13 SELECT … WHERE ... LIKE % DIAWAL

Digunakan untuk mencari data yang berakhir dengan pola terrtentu.

Bentuk umum:

SELECT nama_kolom
FROM nama_tabel
WHERE nama_kolom ‘pola’ LIKE;

Contoh:

SELECT nama
FROM Pelanggan
WHERE nama LIKE '%i';

```
MariaDB [tokoroti]> SELECT nama
    -> FROM Pelanggan
    -> WHERE nama LIKE '%i';
+------+
| nama |
+------+
| Andi |
| Budi |
| Dewi |
+------+
3 rows in set (0.002 sec)
```

Gambar 4. 14 SELECT…WHERE…LIKE%

4.2.14 SELECT … WHERE ... LIKE UNDERSCORE (_)

Digunakan untuk mewakili satu karakter

Bentuk Umum:

SELECT nama_kolom FROM nama_tabel WHERE nama_kolom LIKE ‘a__’;

Contoh:

SELECT nama
FROM Pelanggan
WHERE nama LIKE '_u%';

```
MariaDB [tokoroti]> SELECT nama
    -> FROM Pelanggan
    -> WHERE nama LIKE '_u%';
+------+
| nama |
+------+
| Budi |
+------+
1 row in set (0.002 sec)
```

Gambar 4. 15 SELECT…WHERE…LIKE_

4.2.15 SELECT … WHERE ... LIKE KOMBINASI % DAN UNDERSCORE (_)

Digunakan bersama untuk bikin pola lebih spesifik.

Bentuk Umum:
SELECT nama_kolom FROM nama_tabel WHERE nama_kolom LIKE ...

Contoh:
SELECT nama FROM mahasiswa WHERE nama LIKE '_u%i';

```
MariaDB [tokoroti]> SELECT nama
    -> FROM Pelanggan
    -> WHERE nama LIKE '_u%i';
+------+
| nama |
+------+
| Budi |
+------+
1 row in set (0.001 sec)
```

Gambar 4. 16 SELECT…WHERE…LIKE_%

4.2.16 SELECT … UNION

Menggabungkan hasil query dari dua tabel dengan struktur kolom yang sama.

Bentuk umum:
SELECT kolom FROM tabel1
UNION
SELECT kolom FROM tabel2;

Contoh:
SELECT nama FROM pelanggan
UNION
SELECT nama FROM pelanggan WHERE kota = 'Medan';

```
MariaDB [TokoRoti]> SELECT nama FROM Pelanggan
    -> UNION
    -> SELECT nama FROM Pelanggan WHERE kota = 'Medan';
+-------+
| nama  |
+-------+
| Andi  |
| Budi  |
| Citra |
| Dewi  |
| Eko   |
+-------+
5 rows in set (0.033 sec)
```

Gambar 4. 17 SELECT…UNION

4.2.18 SELECT … LEFT

Mengambil semua data dari tabel kiri(left), dan data dari tabel kanan (right) jika ada yang cocok.

Bentuk umum:
SELECT kolom FROM tabel1
LEFT JOIN
SELECT kolom FROM tabel2;

Contoh:
SELECT nama FROM pelanggan
LEFT JOIN
SELECT nama FROM pelanggan WHERE kota = 'Medan';

4.2.19 SELECT … FULL JOIN

Mengambil semua data dari tabel kiri dan kanan, baik yang cocok maupun yang tidak cocok. Jika tidak cocok, akan muncul NULL.

Bentuk umum:
SELECT kolom FROM tabel1
FULL JOIN
SELECT kolom FROM tabel2;

Contoh:
SELECT nama FROM pelanggan
FULL JOIN
SELECT nama FROM pelanggan WHERE kota = 'Medan';

4.2.20 SELECT … DISTINCT

Menghapus duplikasi pada hasil query.
Bentuk umum:
SELECT DISTINCT nama_kolom
FROM nama_tabel;

Contoh:
SELECT DISTINCT kota
FROM pelanggan;

```
MariaDB [TokoRoti]> SELECT DISTINCT kota FROM Pelanggan;
+---------+
| kota    |
+---------+
| Jakarta |
| Medan   |
| Bandung |
+---------+
3 rows in set (0.002 sec)
```

Gambar 4. 18 SELECT…DISTINCT

4.2.21 SELECT … GROUP BY

Mengelompokkan data berdasarkan nilai kolom tertentu. Biasanya digunakan dengan fungsi agregat.
Bentuk umum:
SELECT kolom_agregat, kolom
FROM nama_tabel

GROUP BY kolom;

Contoh:
SELECT kota, COUNT(*) AS Jumlah_Pelanggan
FROM pelanggan
GROUP BY kota;

```
MariaDB [TokoRoti]> SELECT kota, COUNT(*) AS Jumlah_Pelanggan
    -> FROM Pelanggan
    -> GROUP BY kota;
+---------+-----------------+
| kota    | Jumlah_Pelanggan |
+---------+-----------------+
| Bandung |               1 |
| Jakarta |               2 |
| Medan   |               2 |
+---------+-----------------+
3 rows in set (0.009 sec)
```

Gambar 4. 19 SELECT…GROUP BY

4.2.22 SELECT … AND / OR / NOT

AND menggabungkan dua atau lebih kondisi, semua kondisi harus benar agar data ditampilkan.

Menggabungkan beberapa kondisi dalam WHERE.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE kondisi1 AND kondisi2;

Contoh:
SELECT nama
FROM pelanggan
WHERE kota = 'Jakarta' AND umur > 25;

```
MariaDB [TokoRoti]> SELECT nama FROM Pelanggan
    -> WHERE kota = 'Jakarta' AND umur > 25;
+-------+
| nama  |
+-------+
| Citra |
+-------+
1 row in set (0.097 sec)
```

Gambar 4. 20 SELECT…AND

OR menggabungkan dua atau lebih kondisi, cukup salah satu kondisi benar agar data ditampilkan.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE kondisi1 OR kondisi26;

Contoh:
SELECT nama
FROM pelanggan
WHERE kota = 'Jakarta' OR umur > 25;

```
MariaDB [tokoroti]> SELECT nama
    -> FROM pelanggan
    -> WHERE kota = 'Jakarta' OR umur > 25;
+-------+
| nama  |
+-------+
| Andi  |
| Budi  |
| Citra |
| Eko   |
+-------+
4 rows in set (0.091 sec)
```

Gambar 4. 21 SELECT…OR

NOT menyaring data yang tidak memenuhi kondisi tertentu.

Bentuk umum:
SELECT nama_kolom
FROM nama_tabel
WHERE kondisi1 NOT kondisi26;

Contoh:
SELECT nama
FROM pelanggan
WHERE kota = 'Jakarta' NOT umur > 25;

4.2.23 SELECT … AGGREGATE

Menggunakan fungsi agregat seperti COUNT, SUM, AVG, MAX, MIN.

Bentuk umum:
SELECT fungsi_agregat(nama_kolom)
FROM nama_tabel;

Contoh:
SELECT AVG(umur) AS Rata_Umur, MAX(umur) AS Umur_Tertua
FROM Pelanggan;

```
MariaDB [TokoRoti]> SELECT AVG(umur) AS Rata_Umur, MAX(umur) AS Umur_Tertua FROM Pelanggan;
+-----------+-------------+
| Rata_Umur | Umur_Tertua |
+-----------+-------------+
|   28.4000 |          35 |
+-----------+-------------+
1 row in set (0.002 sec)
```

Gambar 4. 22 SELECT…AGGREGATE

4.2.24 SELECT … AGGREGATE.. COUNTS

Menghitung jumlah baris(rows) dalam tabel atau kelompok data. Bisa menghitung (COUNT(*)) atau hanya kolom tertentu.

Bentuk umum:
SELECT COUNT(nama_kolom)
FROM nama_tabel;

Contoh:
SELECT COUNT(*) AS Jumlah_Jakarta
FROM Pelanggan
WHERE kota = 'Jakarta';

```
MariaDB [tokoroti]> SELECT COUNT(*) AS Jumlah_Jakarta
    -> FROM Pelanggan
    -> WHERE kota = 'Jakarta';
+----------------+
| Jumlah_Jakarta |
+----------------+
|              2 |
+----------------+
1 row in set (0.052 sec)
```

Gambar 4. 23 SELECT…AGGREGATE…COUNT

4.2.25 SELECT … AGGREGATE..SUM

Menjumlahkan nilai dari sebuah kolom bertope numerik

Bentuk umum:
SELECT SUM(nama_kolom)
FROM nama_tabel;

Contoh:
SELECT SUM(total) AS Total_Pendapatan
FROM Pesanan;

```
MariaDB [tokoroti]> SELECT SUM(total) AS Total_Pendapatan
    -> FROM Pesanan;
+------------------+
| Total_Pendapatan |
+------------------+
|           395000 |
+------------------+
1 row in set (0.055 sec)
```

Gambar 4. 24 SELECT…AGGREGATE…SUM

4.2.26 SELECT … AGGREGATE..AVG

Menghitung nilai rata-rata dari sebuah kolom numerik.

Bentuk umum:
SELECT AVG (nama_kolom)
FROM nama_tabel;

Contoh:
SELECT AVG(umur) AS Rata_Umur
FROM Pelanggan;

```
MariaDB [tokoroti]> SELECT AVG(umur) AS Rata_Umur
    -> FROM Pelanggan;
+-----------+
| Rata_Umur |
+-----------+
|   28.4000 |
+-----------+
1 row in set (0.001 sec)
```

Gambar 4. 25 SELECT…AGGREGATE…AVG

4.2.27 SELECT … AGGREGATE..MAX

Menghasilkan nilai terbesar (maksimum) dalam kolom tertentu.

Bentuk umum:
SELECT MAX (nama_kolom)
FROM nama_tabel;

Contoh:
SELECT MAX(total) AS Pesanan_Terbesar
FROM Pesanan;

```
MariaDB [tokoroti]> SELECT MAX(total) AS Pesanan_Terbesar
    -> FROM Pesanan;
+------------------+
| Pesanan_Terbesar |
+------------------+
|           120000 |
+------------------+
1 row in set (0.069 sec)
```

Gambar 4. 26 SELECT…AGGREGATE…MAX

4.2.28 SELECT … AGGREGATE.. MIN

Menghasilkan nilai terkecil (minmum) dari kolom tertentu..

Bentuk umum:
SELECT Min(nama_kolom)
FROM nama_tabel;

Contoh:
SELECT MIN(total) AS Pesanan_Terkecil
FROM Pesanan;

```
MariaDB [tokoroti]> SELECT MIN(total) AS Pesanan_Terkecil
    -> FROM Pesanan;
+------------------+
| Pesanan_Terkecil |
+------------------+
|            50000 |
+------------------+
1 row in set (0.001 sec)
```

Gambar 4. 27 SELECT…AGGREGATE…MIN

BAB V

CONSTRAINTS

5.1 Definisi Constraints

Constrain merupakan aturan yang diterapkan pada kolom suatu tabel sehingga nilai pada tabel tersebut memenuhi kriteria/aturan yang telah ditetapkan. Dengan constrain ini, kita dapat membatasi data apa saja yang bisa dimasukkan ke dalam tabel, dengan demikian akurasi dan integritas data tetap dapat terjaga. Misal: kolom jenis kelamin harus diisi dengan nilai L atau P, jika diisi dengan selain itu, maka data akan ditolak. Bentuk constrain ini sangat banyak tidak terbatas pada bentuk tertentu, jika memenuhi kriteria diatas, maka bisa disebut constrain meskipun bentuknya stored program seperti trigger. Untuk membatasi pembahasan, pada bab ini, kita hanya akan membahas beberapa constrain eksplisit yang didukung oleh MySQL, yaitu NOT NULL, PRIMARY KEY, UNIQUE, FOREIGN KEY, ENUM, SET, dan CHECK. Umumnya, constrain didefinisikan bersamaan dengan pembuatan tabel, constrain terebut dapat didefinisikan pada level kolom maupun pada level tabel. Pada level kolom, constrain hanya berlaku pada kolom tersebut, jika pada level tabel, constrain tersebut dapat diterapkan pada beberapa kolom.

5.2 Jenis-Jenis Constraint

Ada beberapa jenis constraints di dalam bahasa query, antara lain:

5.2.1 Primary Key

Primary Key adalah constraint yang berfungsi sebagai identitas unik dari setiap baris data di dalam tabel.
Ciri-cirinya:

a. Nilai harus unik (tidak boleh ada data ganda).
b. Nilai tidak boleh NULL.
c. Satu tabel hanya boleh memiliki satu PRIMARY KEY, tetapi bisa terdiri dari satu kolom atau gabungan beberapa kolom (composite key).

CARA MENAMBAHKAN PRIMARY KEY

Ada 3 cara utama untuk membuat/menambahkan Primary Key, yaitu:

a. Menambahkan PRIMARY KEY Langsung di Kolom Saat CREATE TABLE

Cara paling sederhana adalah mendefinisikan PRIMARY KEY langsung setelah tipe data kolom.

Bentuk Umum:

```
CREATE TABLE nama_tabel (
nama_kolom tipe_data PRIMARY KEY,
kolom_lain tipe_data
);
```

Contoh:

```
CREATE TABLE tbprodi (
kdprodi INT(11) PRIMARY KEY,
Namaprodi VARCHAR(50) NOT NULL,
Prog_Pendidikan VARCHAR(10)
);
```

b. Menambahkan PRIMARY KEY dengan CONSTRAINT di CREATE TABLE

PRIMARY KEY bisa juga ditulis terpisah menggunakan CONSTRAINT. Biasanya dipakai jika kolom kunci lebih dari satu (composite key).

Bentuk Umum:

```
CREATE TABLE nama_tabel (
kolom1 tipe_data,
kolom2 tipe_data,
kolom3 tipe_data,
CONSTRAINT nama_pk PRIMARY KEY (kolom1, kolom2)
);
```

Contoh:

```
CREATE TABLE transaksi_cicilan (
NoKwitansi VARCHAR(5),
Nim VARCHAR(10),
CONSTRAINT pk_transaksi PRIMARY KEY (NoKwitansi, Nim)
);
```

c. Menambahkan PRIMARY KEY dengan ALTER TABLE (Setelah Tabel Dibuat)

Kalau tabel sudah terlanjur dibuat tanpa PRIMARY KEY, bisa ditambahkan belakangan dengan ALTER TABLE.

Bentuk Umum:

```
ALTER TABLE nama_tabel
ADD CONSTRAINT nama_pk PRIMARY KEY (nama_kolom);
```

Contoh:

```
ALTER TABLE tbmahasiswa
ADD CONSTRAINT pk_mahasiswa PRIMARY KEY (Nim);
```

5.2.2 Foreign Key

Digunakan untuk menghubungkan tabel satu dengan tabel lain. Nilai pada kolom dengan FOREIGN KEY harus ada pada kolom PRIMARY KEY tabel yang direferensikan.

Bentuk Umum:

```
CONSTRAINT nama_fk FOREIGN KEY (kolom_di_tabel_ini)
REFERENCES nama_tabel_lain(kolom_pk);
```

Contoh:

```
CREATE TABLE transaksi_cicilan (
NoKwitansi VARCHAR(5) PRIMARY KEY,
Nim VARCHAR(10) NOT NULL,
CONSTRAINT fk_mahasiswa FOREIGN KEY (Nim)
REFERENCES tbmahasiswa(Nim)
);
```

5.2.3 Unique

Menjamin bahwa setiap nilai dalam kolom tidak boleh ada yang sama. Beda dengan PRIMARY KEY, kolom UNIQUE masih boleh berisi NULL (jika tidak ada NOT NULL).

Bentuk Umum:

```
nama_kolom tipe_data UNIQUE
Atau
CONSTRAINT nama_unik UNIQUE (nama_kolom)
```

Contoh:

```
CREATE TABLE tbkasir (
kdkasir VARCHAR(5) PRIMARY KEY,
Namakasir VARCHAR(30) UNIQUE
);
```

5.2.4 Not Null

Membatasi kolom agar **tidak boleh kosong** (NULL).

Sintax:

```
nama_kolom tipe_data NOT NULL
```

Contoh:

```
CREATE TABLE tbsesi (
kdsesi VARCHAR(5) PRIMARY KEY,
Namasesi VARCHAR(7) NOT NULL
);
```

5.2.5 Check

Digunakan untuk membatasi nilai kolom agar sesuai dengan kondisi tertentu.

Sintax:

nama_kolom tipe_data CHECK (kondisi)

Atau

CONSTRAINT nama_check CHECK (kondisi)

Contoh:

```
CREATE TABLE tbjenisbayar (
kdjenis VARCHAR(5) PRIMARY KEY,
TahunAjaran VARCHAR(10) NOT NULL,
UangSejumlah INT CHECK (UangSejumlah >= 0)
);
```

5.2.6 Default

Memberikan nilai bawaan (default) ketika user tidak mengisi kolom tersebut saat
melakukan INSERT.

Sintax:

nama_kolom tipe_data DEFAULT nilai

Contoh:

```
CREATE TABLE tbmahasiswa (
Nim VARCHAR(10) PRIMARY KEY,
NamaMahasiswa VARCHAR(30),
Status VARCHAR(15) DEFAULT 'Aktif'
);
```

BAB VI

HASIL DAN PEMBAHASAN

6.1 CPMK DDL dan DML

1. Membuat Database

Sintaks	CREATE DATABASE CicilanKuliahh;
Penjelasan	Membuat database dengan Nama : CicilanKuliah
Hasil	`MariaDB [(none)]> CREATE DATABASE CicilanKuliah;` `Query OK, 1 row affected (0.001 sec)`

2. Tampilkan database

Sintaks	SHOW DATABASES;
Penjelasan	Tampilkan database

Hasil	`MariaDB [(none)]> SHOW DATABASES;`
	`+--------------------+`
	`| Database |`
	`+--------------------+`
	`| cicilankuliah |`
	`| information_schema |`
	`| mysql |`
	`| performance_schema |`
	`| phpmyadmin |`
	`| test |`
	`+--------------------+`
	`6 rows in set (0.001 sec)`

3. Masuk ke dalam database

Sintaks	USE CicilanKuliahh;
Penjelasan	Masuk Ke dalam Data base

Hasil	`MariaDB [(none)]> USE CicilanKuliah;` `Database changed`

4. Buat tabel

a. Tabel tbprodi

Sintaks	CREATE TABLE tbprodi (kdprodi INT(11) PRIMARY KEY, Namaprodi VARCHAR(50) NOT NULL, Prog_Pendidikan VARCHAR(10));
Penjelasan	Buat Tabel tbprodi

Hasil	`MariaDB [CicilanKuliah]> CREATE TABLE tbprodi (`
	`-> kdprodi INT(11) PRIMARY KEY,`
	`-> Namaprodi VARCHAR(50) NOT NULL,`
	`-> Prog_Pendidikan VARCHAR(10)`
	`-> );`
	`Query OK, 0 rows affected (0.388 sec)`

b. Tabel tbsesi

Sintaks	CREATE TABLE tbsesi (kdsesi VARCHAR(5) PRIMARY KEY, Namasesi VARCHAR(7) NOT NULL);
Penjelasan	Buat Tabel tbsesi

Hasil	`MariaDB [CicilanKuliah]> CREATE TABLE tbsesi (`
	`-> kdsesi VARCHAR(5) PRIMARY KEY,`
	`-> Namasesi VARCHAR(7) NOT NULL`
	`-> );`
	`Query OK, 0 rows affected (0.158 sec)`

c. Tabel tbkelas

Sintaks	CREATE TABLE tbkelas (kdkelas VARCHAR(5) PRIMARY KEY, Namakelas VARCHAR(6) NOT NULL);
Penjelasan	Buat Tabel tbkelas

Hasil	MariaDB [CicilanKuliah]> CREATE TABLE tbkelas (-> kdkelas VARCHAR(5) PRIMARY KEY, -> Namakelas VARCHAR(6) NOT NULL ->); Query OK, 0 rows affected (0.140 sec)

d. Tabel tbkasir

Sintaks	CREATE TABLE tbkasir (kdkasir VARCHAR(5) PRIMARY KEY, Namakasir VARCHAR(30) NOT NULL);
Penjelasan	Buat Tabel tbkasir
Hasil	MariaDB [CicilanKuliah]> CREATE TABLE tbkasir (-> kdkasir VARCHAR(5) PRIMARY KEY, -> Namakasir VARCHAR(30) NOT NULL ->); Query OK, 0 rows affected (0.160 sec)

e. Tabel tbjenisbayar

Sintaks	CREATE TABLE tbjenisbayar (kdjenis VARCHAR(5) PRIMARY KEY, jenisbayar ENUM('pendaftaran','uang kuliah','Perlengkapan','ICT'), TahunAjaran VARCHAR(10) NOT NULL);
Penjelas an	Buat Tabel tbjenisbayar
Hasil	MariaDB [CicilanKuliah]> CREATE TABLE tbjenisbayar (-> kdjenis VARCHAR(5) PRIMARY KEY, -> jenisbayar ENUM('pendaftaran','uang kuliah','Perleng -> TahunAjaran VARCHAR(10) NOT NULL ->); Query OK, 0 rows affected (0.423 sec)

f. Tabel tbmahasiswa

Sintaks	CREATE TABLE tbmahasiswa (Nim VARCHAR(10) PRIMARY KEY, NamaMahasiswa VARCHAR(30), TempatLahir VARCHAR(30) NOT NULL, TglLahir DATE NOT NULL);
Penjelasan	Buat Tabel tbmahasiswa

Hasil	`MariaDB [CicilanKuliah]> CREATE TABLE tbmahasiswa (`
	`->  Nim VARCHAR(10) PRIMARY KEY,`
	`->  NamaMahasiswa VARCHAR(30),`
	`->  TempatLahir VARCHAR(30) NOT NULL,`
	`->  TglLahir DATE NOT NULL`
	`-> );`
	`Query OK, 0 rows affected (0.228 sec)`

g. Tabel transaksi_cicilan

Sintaks	CREATE TABLE transaksi_cicilan (NoKwitansi VARCHAR(5) PRIMARY KEY, Nim VARCHAR(10) NOT NULL, KdProdi INT(11), KdKelas VARCHAR(5) NOT NULL, TglBayar DATE NOT NULL, KdKasir VARCHAR(5) NOT NULL, KdSesi VARCHAR(5) NOT NULL, UangSejumlah BIGINT(10) NOT NULL, Denda INT(10) NOT NULL, KdJenisBayar VARCHAR(5) NOT NULL);
Penjelasan	Buat Tabel transaksi_cicilan

Hasil	`MariaDB [CicilanKuliah]> CREATE TABLE transaksi_cicilan (`
	`->  NoKwitansi VARCHAR(5) PRIMARY KEY,`
	`->  Nim VARCHAR(10) NOT NULL,`
	`->  KdProdi INT(11),`
	`->  KdKelas VARCHAR(5) NOT NULL,`
	`->  TglBayar DATE NOT NULL,`
	`->  KdKasir VARCHAR(5) NOT NULL,`
	`->  KdSesi VARCHAR(5) NOT NULL,`
	`->  UangSejumlah BIGINT(10) NOT NULL,`
	`->  Denda INT(10) NOT NULL,`
	`->  KdJenisBayar VARCHAR(5) NOT NULL`
	`-> );`
	`Query OK, 0 rows affected (0.135 sec)`

5. Tampilkan Struktur Tabel

Sintaks	SHOW TABLES;
Penjelasan	Tampilkan Struktur Table yang baru di buat

Hasil	MariaDB [CicilanKuliah]> SHOW TABLES; +----------+ | Tables_in_cicilankuliah | +----------+ | tbjenisbayar | | tbkasir | | tbkelas | | tbmahasiswa | | tbprodi | | tbsesi | | transaksi_cicilan | +----------+ 7 rows in set (0.002 sec)

Sintaks	DESC tbprodi;
Penjelasan	Tampilkan Struktur Table yang baru di buat
Hasil	MariaDB [CicilanKuliah]> DESC tbprodi; +----------+----------+------+-----+---------+----- | Field | Type | Null | Key | Default | Extr +----------+----------+------+-----+---------+----- | kdprodi | int(11) | NO | PRI | NULL | | Namaprodi | varchar(50) | NO | | NULL | | Prog_Pendidikan | varchar(10) | YES | | NULL | +----------+----------+------+-----+---------+----- 3 rows in set (0.049 sec)

Sintaks	DESC tbsesi;
Penjelasan	Tampilkan Struktur Table yang baru di buat
Hasil	MariaDB [CicilanKuliah]> DESC tbsesi; +----------+----------+------+-----+---------+-------+ | Field | Type | Null | Key | Default | Extra | +----------+----------+------+-----+---------+-------+ | kdsesi | varchar(5) | NO | PRI | NULL | | | Namasesi | varchar(7) | NO | | NULL | | +----------+----------+------+-----+---------+-------+ 2 rows in set (0.016 sec)

Sintaks	DESC tbkelas;
Penjelasan	Tampilkan Struktur Table yang baru di buat
Hasil	

```
MariaDB [CicilanKuliah]> DESC tbkelas;
+-----------+------------+------+-----+---------+-------+
| Field     | Type       | Null | Key | Default | Extra |
+-----------+------------+------+-----+---------+-------+
| kdkelas   | varchar(5) | NO   | PRI | NULL    |       |
| Namakelas | varchar(6) | NO   |     | NULL    |       |
+-----------+------------+------+-----+---------+-------+
2 rows in set (0.014 sec)
```

Sintaks	DESC tbkasir;
Penjelasan	Tampilkan Struktur Table yang baru di buat
Hasil	

```
MariaDB [CicilanKuliah]> DESC tbkasir;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| kdkasir   | varchar(5)  | NO   | PRI | NULL    |       |
| Namakasir | varchar(30) | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0.005 sec)
```

Sintaks	DESC tbjenisbayar;
Penjelasan	Tampilkan Struktur Table yang baru di buat
Hasil	

```
MariaDB [CicilanKuliah]> DESC tbjenisbayar;
+------------+-----------------------------------------------------+------+-----+---------+-------+
| Field      | Type                                                | Null | Key | Default | Extra |
+------------+-----------------------------------------------------+------+-----+---------+-------+
| kdjenis    | varchar(5)                                          | NO   | PRI | NULL    |       |
| jenisbayar | enum('pendaftaran','uang kuliah','Perlengkapan','ICT') | YES  |     | NULL    |       |
| TahunAjaran | varchar(10)                                         | NO   |     | NULL    |       |
+------------+-----------------------------------------------------+------+-----+---------+-------+
3 rows in set (0.006 sec)
```

Sintaks	DESC tbmahasiswa;
Penjelasan	Tampilkan Struktur Table yang baru di buat
Hasil	

```
MariaDB [CicilanKuliah]> DESC tbmahasiswa;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Nim           | varchar(10) | NO   | PRI | NULL    |       |
| NamaMahasiswa | varchar(30) | YES  |     | NULL    |       |
| TempatLahir   | varchar(30) | NO   |     | NULL    |       |
| TglLahir      | date        | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.005 sec)
```

Sintaks	DESC transaksi_cicilan;
Penjelasan	Tampilkan Struktur Table yang baru di buat
Hasil	

```
MariaDB [CicilanKuliah]> DESC transaksi_cicilan;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| NoKwitansi   | varchar(5)  | NO   | PRI | NULL    |       |
| Nim          | varchar(10) | NO   |     | NULL    |       |
| KdProdi      | int(11)     | YES  |     | NULL    |       |
| KdKelas      | varchar(5)  | NO   |     | NULL    |       |
| TglBayar     | date        | NO   |     | NULL    |       |
| KdKasir      | varchar(5)  | NO   |     | NULL    |       |
| KdSesi       | varchar(5)  | NO   |     | NULL    |       |
| UangSejumlah | bigint(10)  | NO   |     | NULL    |       |
| Denda        | int(10)     | NO   |     | NULL    |       |
| KdJenisBayar | varchar(5)  | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
10 rows in set (0.005 sec)
```

6. Tambahkan Foreign Key di transaksi_cicilan

Sintaks	ALTER TABLE transaksi_cicilan ADD CONSTRAINT fk_mahasiswa FOREIGN KEY (Nim) REFERENCES tbmahasiswa(Nim), ADD CONSTRAINT fk_prodi FOREIGN KEY (KdProdi)

	REFERENCES tbprodi(kdprodi), ADD CONSTRAINT fk_kelas FOREIGN KEY (KdKelas) REFERENCES tbkelas(kdkelas), ADD CONSTRAINT fk_kasir FOREIGN KEY (KdKasir) REFERENCES tbkasir(kdkasir), ADD CONSTRAINT fk_sesi FOREIGN KEY (KdSesi) REFERENCES tbsesi(kdsesi), ADD CONSTRAINT fk_jenis FOREIGN KEY (KdJenisBayar) REFERENCES tbjenisbayar(kdjenis);
Penjelasan	Dengan Perintah Alter Berikan Foreign Key pada Tabel transaksi_cicilan Untuk Kolom : NIM, KdProdi, KdKelas, KdKasir, KdSesi, KdJenisBayar
Hasil	

```
MariaDB [CicilanKuliah]> ALTER TABLE transaksi_cicilan
    -> ADD CONSTRAINT fk_mahasiswa FOREIGN KEY (Nim) REFERENCES tbmahasiswa(Nim),
    -> ADD CONSTRAINT fk_prodi FOREIGN KEY (KdProdi) REFERENCES tbprodi(kdprodi),
    -> ADD CONSTRAINT fk_kelas FOREIGN KEY (KdKelas) REFERENCES tbkelas(kdkelas),
    -> ADD CONSTRAINT fk_kasir FOREIGN KEY (KdKasir) REFERENCES tbkasir(kdkasir),
    -> ADD CONSTRAINT fk_sesi FOREIGN KEY (KdSesi) REFERENCES tbsesi(kdsesi),
    -> ADD CONSTRAINT fk_jenis FOREIGN KEY (KdJenisBayar) REFERENCES tbjenisbayar(kdjenis);
Query OK, 0 rows affected (0.893 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [Cicilan_Kuliah]> desc  transaksi_cicilan;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| NoKwitansi   | varchar(5)  | NO   | PRI | NULL    |       |
| Nim          | varchar(10) | NO   | MUL | NULL    |       |
| KdProdi      | int(11)     | YES  | MUL | NULL    |       |
| KdKelas      | varchar(5)  | NO   | MUL | NULL    |       |
| TglBayar     | date        | NO   |     | NULL    |       |
| KdKasir      | varchar(5)  | NO   | MUL | NULL    |       |
| KdSesi       | varchar(5)  | NO   | MUL | NULL    |       |
| UangSejumlah | bigint(10)  | NO   |     | NULL    |       |
| Denda        | int(10)     | NO   |     | NULL    |       |
| KdJenisBayar | varchar(5)  | NO   | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
10 rows in set (0.010 sec)
```

7. Tambahkan kolom jenis_kelamin

Sintaks	ALTER TABLE tbmahasiswa ADD jenis_kelamin VARCHAR(1) NOT NULL AFTER NamaMahasiswa;
Penjelasan	Dengan Perintah Alter Tambahkan Kolom jenis_kelamin Varchar(1) Not Null di TbMahasiswa di bawah kolom NamaMahasiswa

Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE tbmahasiswa ADD jenis_kelamin VARCHAR(1) NOT NULL AFTER NamaMahasiswa;
Query OK, 0 rows affected (0.309 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]> desc tbmahasiswa;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Nim           | varchar(10) | NO   | PRI | NULL    |       |
| NamaMahasiswa | varchar(30) | YES  |     | NULL    |       |
| jenis_kelamin | varchar(1)  | NO   |     | NULL    |       |
| TempatLahir   | varchar(30) | NO   |     | NULL    |       |
| TglLahir      | date        | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.012 sec)
```

8. Tambahkan kolom Hobbi

Sintaks	ALTER TABLE tbmahasiswa ADD Hobbi ENUM('membaca','menulis','olahraga');
Penjelasan	Dengan Perintah Alter Tambahkan Kolom Hobbi dengan type enum Dimana hobbi nya : membaca, menulis, olaraga
Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE tbmahasiswa ADD Hobbi ENUM('membaca','menulis','olahraga');
Query OK, 0 rows affected (0.181 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]> desc tbmahasiswa;
+---------------+-----------------------------------+------+-----+---------+-------+
| Field         | Type                              | Null | Key | Default | Extra |
+---------------+-----------------------------------+------+-----+---------+-------+
| Nim           | varchar(10)                       | NO   | PRI | NULL    |       |
| NamaMahasiswa | varchar(30)                       | YES  |     | NULL    |       |
| jenis_kelamin | varchar(1)                        | NO   |     | NULL    |       |
| TempatLahir   | varchar(30)                       | NO   |     | NULL    |       |
| TglLahir      | date                              | NO   |     | NULL    |       |
| Hobbi         | enum('membaca','menulis','olahraga') | YES  |     | NULL    |       |
+---------------+-----------------------------------+------+-----+---------+-------+
6 rows in set (0.008 sec)
```

9. Ubah size NamaMahasiswa

Sintaks	ALTER TABLE tbmahasiswa MODIFY NamaMahasiswa VARCHAR(50);
Penjelasan	Dengan Perintah Alter Ubah size Kolom NamaMahasiswa menjadi 50
Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE tbmahasiswa MODIFY NamaMahasiswa VARCHAR(50);
Query OK, 0 rows affected (0.144 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]> desc tbmahasiswa;
+---------------+-------------------------------------+------+-----+---------+-------+
| Field         | Type                                | Null | Key | Default | Extra |
+---------------+-------------------------------------+------+-----+---------+-------+
| Nim           | varchar(10)                         | NO   | PRI | NULL    |       |
| NamaMahasiswa | varchar(50)                         | YES  |     | NULL    |       |
| jenis_kelamin | varchar(1)                          | NO   |     | NULL    |       |
| TempatLahir   | varchar(30)                         | NO   |     | NULL    |       |
| TglLahir      | date                                | NO   |     | NULL    |       |
| Hobbi         | enum('membaca','menulis','olahraga') | YES  |     | NULL    |       |
+---------------+-------------------------------------+------+-----+---------+-------+
6 rows in set (0.009 sec)
```

10. Ubah nama kolom Hobbi

Sintaks	ALTER TABLE tbmahasiswa CHANGE Hobbi Hobbi_mahasiswa ENUM('membaca','menulis','olahraga');
Penjelasan	Dengan Perintah Alter Ubah Kolom Hobi menjadi Hobbi_mahasiswa
Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE tbmahasiswa CHANGE Hobbi Hobbi_mahasiswa ENUM('membaca','menulis','olahraga');
Query OK, 0 rows affected (0.187 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]>
MariaDB [Cicilan_Kuliah]>  desc tbmahasiswa;
+----------------+-------------------------------------+------+-----+---------+-------+
| Field          | Type                                | Null | Key | Default | Extra |
+----------------+-------------------------------------+------+-----+---------+-------+
| Nim            | varchar(10)                         | NO   | PRI | NULL    |       |
| NamaMahasiswa  | varchar(50)                         | YES  |     | NULL    |       |
| jenis_kelamin  | varchar(1)                          | NO   |     | NULL    |       |
| TempatLahir    | varchar(30)                         | NO   |     | NULL    |       |
| TglLahir       | date                                | NO   |     | NULL    |       |
| Hobbi_mahasiswa | enum('membaca','menulis','olahraga') | YES  |     | NULL    |       |
+----------------+-------------------------------------+------+-----+---------+-------+
6 rows in set (0.009 sec)
```

11. Hapus kolom jenis_kelamin

Sintaks	ALTER TABLE tbmahasiswa DROP COLUMN jenis_kelamin;
Penjelasan	Dengan Perintah Alter Hapus Kolom Jenis_kelamin

Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE tbmahasiswa DROP COLUMN jenis_kelamin;
Query OK, 0 rows affected (0.111 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]>  desc tbmahasiswa;
+-----------------+-----------------------------------+------+-----+---------+-------+
| Field           | Type                              | Null | Key | Default | Extra |
+-----------------+-----------------------------------+------+-----+---------+-------+
| Nim             | varchar(10)                       | NO   | PRI | NULL    |       |
| NamaMahasiswa   | varchar(50)                       | YES  |     | NULL    |       |
| TempatLahir     | varchar(30)                       | NO   |     | NULL    |       |
| TglLahir        | date                              | NO   |     | NULL    |       |
| Hobbi_mahasiswa | enum('membaca','menulis','olahraga') | YES  |     | NULL    |       |
+-----------------+-----------------------------------+------+-----+---------+-------+
5 rows in set (0.004 sec)
```

12. Hapus kolom Hobbi_mahasiswa

Sintaks	ALTER TABLE tbmahasiswa DROP COLUMN Hobbi_mahasiswa;
Penjelasan	Dengan Perintah Alter Hapus Kolom Hobbi
Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE tbmahasiswa DROP COLUMN Hobbi_mahasiswa;
Query OK, 0 rows affected (0.100 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]> desc tbmahasiswa;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Nim           | varchar(10) | NO   | PRI | NULL    |       |
| NamaMahasiswa | varchar(50) | YES  |     | NULL    |       |
| TempatLahir   | varchar(30) | NO   |     | NULL    |       |
| TglLahir      | date        | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.008 sec)
```

13. Buat tabel Login

Sintaks	CREATE TABLE Login (username VARCHAR(5) NOT NULL, Password VARCHAR(10) NOT NULL);
Penjelasan	Dengan perintah Create table, Buat Table Login diatas

Hasil	MariaDB [Cicilan_Kuliah]> CREATE TABLE Login (-> username VARCHAR(5) NOT NULL, -> Password VARCHAR(10) NOT NULL ->); Query OK, 0 rows affected (0.287 sec) MariaDB [Cicilan_Kuliah]> desc login; +----------+----------+------+-----+---------+-------+ | Field | Type | Null | Key | Default | Extra | +----------+----------+------+-----+---------+-------+ | username | varchar(5) | NO | | NULL | | | Password | varchar(10) | NO | | NULL | | +----------+----------+------+-----+---------+-------+ 2 rows in set (0.010 sec)

14. Tambah kolom id

Sintaks	ALTER TABLE Login ADD id INT(11) AUTO_INCREMENT PRIMARY KEY FIRST;
Penjelasan	Dengan Perintah Alter Tambahkan Kolom id dengan type int(11) Auto_Increment dengan key Primary Key
Hasil	MariaDB [Cicilan_Kuliah]> ALTER TABLE Login ADD id INT(11) AUTO_INCREMENT PRIMARY KEY FIRST; Query OK, 0 rows affected (0.490 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [Cicilan_Kuliah]> desc login; +----------+----------+------+-----+---------+----------+ | Field | Type | Null | Key | Default | Extra | +----------+----------+------+-----+---------+----------+ | id | int(11) | NO | PRI | NULL | auto_increment | | username | varchar(5) | NO | | NULL | | | Password | varchar(10) | NO | | NULL | | +----------+----------+------+-----+---------+----------+ 3 rows in set (0.010 sec)

15. Tambah kolom firstname & lastname

Sintaks	ALTER TABLE Login ADD firstname VARCHAR(25), ADD lastname VARCHAR(25);
Penjelasan	Dengan Perintah Alter Tambahkan Kolom firstname dan lastname dengan type varchar(25)

Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE Login ADD firstname VARCHAR(25), ADD lastname VARCHAR(25);
Query OK, 0 rows affected (0.108 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]> desc login;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| id       | int(11)     | NO   | PRI | NULL    | auto_increment |
| username | varchar(5)  | NO   |     | NULL    |                |
| Password | varchar(10) | NO   |     | NULL    |                |
| firstname | varchar(25) | YES  |     | NULL    |                |
| lastname | varchar(25) | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
5 rows in set (0.010 sec)
```

16. Tambah kolom email di bawah Password

Sintaks	ALTER TABLE Login ADD email VARCHAR(100) AFTER Password;
Penjelasan	Dengan Perintah Alter Tambahkan Kolom email dengan type varchar(100) dibawah Kolom Password
Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE Login ADD email VARCHAR(100) AFTER Password;
Query OK, 0 rows affected (0.145 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]>  desc login;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| username | varchar(5)   | NO   |     | NULL    |                |
| Password | varchar(10)  | NO   |     | NULL    |                |
| email    | varchar(100) | YES  |     | NULL    |                |
| firstname | varchar(25)  | YES  |     | NULL    |                |
| lastname | varchar(25)  | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
6 rows in set (0.010 sec)
```

17. Ubah size username

Sintaks	ALTER TABLE Login MODIFY username VARCHAR(8);
Penjelasan	Dengan Perintah Alter Rubah Size username menjadi varchar(8)

Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE Login MODIFY username VARCHAR(8);
Query OK, 0 rows affected (0.528 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]> desc login;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| id        | int(11)      | NO   | PRI | NULL    | auto_increment |
| username  | varchar(8)   | YES  |     | NULL    |                |
| Password  | varchar(10)  | NO   |     | NULL    |                |
| email     | varchar(100) | YES  |     | NULL    |                |
| firstname | varchar(25)  | YES  |     | NULL    |                |
| lastname  | varchar(25)  | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
6 rows in set (0.009 sec)
```

18. Ganti nama kolom firstname jadi nama_pertama

Sintaks	ALTER TABLE Login CHANGE firstname nama_pertama VARCHAR(25);
Penjelasan	Dengan Perintah Alter Ganti kolom Firstname menjadi nama_pertama
Hasil	

```
MariaDB [Cicilan_Kuliah]> ALTER TABLE Login CHANGE firstname nama_pertama VARCHAR(25);
Query OK, 0 rows affected (0.206 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Cicilan_Kuliah]> desc login;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int(11)      | NO   | PRI | NULL    | auto_increment |
| username     | varchar(8)   | YES  |     | NULL    |                |
| Password     | varchar(10)  | NO   |     | NULL    |                |
| email        | varchar(100) | YES  |     | NULL    |                |
| nama_pertama | varchar(25)  | YES  |     | NULL    |                |
| lastname     | varchar(25)  | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
6 rows in set (0.009 sec)
```

19. Ganti nama tabel Login jadi ManajemenUser

Sintaks	RENAME TABLE Login TO ManajemenUser;
Penjelasan	Dengan Perintah Alter Ganti nama Tabel Login menjadi ManajemenUser

Hasil	MariaDB [Cicilan_Kuliah]> RENAME TABLE Login TO ManajemenUser; Query OK, 0 rows affected (0.296 sec) MariaDB [Cicilan_Kuliah]> desc ManajemenUser; +---+---+---+---+---+---+ | Field | Type | Null | Key | Default | Extra | +---+---+---+---+---+---+ | id | int(11) | NO | PRI | NULL | auto_increment | | username | varchar(8) | YES | | NULL | | | Password | varchar(10) | NO | | NULL | | | email | varchar(100) | YES | | NULL | | | nama_pertama | varchar(25) | YES | | NULL | | | lastname | varchar(25) | YES | | NULL | | +---+---+---+---+---+---+ 6 rows in set (0.050 sec)

20. Hapus tabel ManajemenUser

Sintaks	DROP TABLE ManajemenUser;
Penjelasan	Dengan Perintah Drop Hapus Tabel ManajemenUser
Hasil	MariaDB [Cicilan_Kuliah]> DROP TABLE ManajemenUser; Query OK, 0 rows affected (0.196 sec)

21. Insert tbprodi

Sintaks	INSERT INTO tbprodi (kdprodi, Namaprodi, Prog_Pendidikan) VALUES (1,'Manajemen Informatika','Diploma 3'), (2,'Sistem Informasi(SI)','Strata 1'), (3,'Teknik Informatika','Strata 1'), (4,'Komputerisasi Akuntansi','Diploma 3'); SELECT * FROM tbprodi;
Penjelasan	Dengan perintah Insert Into dengan Kolom Tambahkan 4 Record di tbprodi diatas dan Tampilkan tbProdi

Hasil	

```
MariaDB [cicilan_kuliah]> INSERT INTO tbprodi (kdprodi, Namaprodi, Prog_Pendidikan) VALUES
    -> (1,'Manajemen Informatika','Diploma 3'),
    -> (2,'Sistem Informasi(SI)','Strata 1'),
    -> (3,'Teknik Informatika','Strata 1'),
    -> (4,'Komputerisasi Akuntansi','Diploma 3');
Query OK, 4 rows affected (0.067 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [cicilan_kuliah]>
MariaDB [cicilan_kuliah]> SELECT * FROM tbprodi;
+---------+------------------------+-----------------+
| kdprodi | Namaprodi              | Prog_Pendidikan |
+---------+------------------------+-----------------+
|       1 | Manajemen Informatika  | Diploma 3       |
|       2 | Sistem Informasi(SI)   | Strata 1        |
|       3 | Teknik Informatika     | Strata 1        |
|       4 | Komputerisasi Akuntansi | Diploma 3       |
+---------+------------------------+-----------------+
4 rows in set (0.028 sec)
```

22. Insert tbsesi

Sintaks	INSERT INTO tbsesi VALUES ('M','Malam'),('P','PAGI'),('S','Siang'); SELECT * FROM tbsesi;
Penjelasan	Dengan perintah Insert Into tanpa Kolom Tambahkan 3 Record di tbsesi diatas dan Tampilkan tbsesi
Hasil	

```
MariaDB [cicilan_kuliah]> INSERT INTO tbsesi VALUES ('M','Malam'),('P','PAGI'),('S','Siang');
Query OK, 3 rows affected (0.070 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [cicilan_kuliah]> SELECT * FROM tbsesi;
+--------+----------+
| kdsesi | Namasesi |
+--------+----------+
| M      | Malam    |
| P      | PAGI     |
| S      | Siang    |
+--------+----------+
3 rows in set (0.035 sec)
```

23. Insert tbkelas

Sintaks	INSERT INTO tbkelas VALUES ('201','20S01'),('202','20S02'),('203','20S03'), ('211','21S01'),('212','21S02'),('213','21S03'), ('214','21S04'),('215','21S05'),('216','21S06'), ('221','22S01'); SELECT * FROM tbkelas;
Penjelasan	Dengan perintah Insert Into Multi Values Tambahkan 10 Record di tbkelas diatas dan Tampilkan tbkelas

Hasil	MariaDB [cicilan_kuliah]> INSERT INTO tbkelas VALUES -> ('201','20S01'),('202','20S02'),('203','20S03'), -> ('211','21S01'),('212','21S02'),('213','21S03'), -> ('214','21S04'),('215','21S05'),('216','21S06'), -> ('221','22S01'); Query OK, 10 rows affected (0.115 sec) Records: 10 Duplicates: 0 Warnings: 0 MariaDB [cicilan_kuliah]> SELECT * FROM tbkelas; +---------+-----------+ | kdkelas | Namakelas | +---------+-----------+ | 201 | 20S01 | | 202 | 20S02 | | 203 | 20S03 | | 211 | 21S01 | | 212 | 21S02 | | 213 | 21S03 | | 214 | 21S04 | | 215 | 21S05 | | 216 | 21S06 | | 221 | 22S01 | +---------+-----------+ 10 rows in set (0.068 sec)

24. Insert tbkasir

Sintaks	INSERT INTO tbkasir SET kdkasir='KS1',Namakasir='Fahmi Firzada, A.MD'; INSERT INTO tbkasir SET kdkasir='KS2',Namakasir='Legina, A.MD'; INSERT INTO tbkasir SET kdkasir='KS3',Namakasir='Sri Rahayu Purba, A.MD'; INSERT INTO tbkasir SET kdkasir='KS4',Namakasir='Elisa, A.MD'; SELECT * FROM tbkasir;
Penjelasan	Dengan perintah Insert Into Set Tambahkan 4 Record di tbkasir diatas dan Tampilkan tbkasir
Hasil	MariaDB [cicilan_kuliah]>MariaDB [cicilan_kuliah]> INSERT INTO tbkasir SET kdkasir='KS1',Namakasir='Fahmi Firzada, A.MD'; Query OK, 1 row affected (0.032 sec) MariaDB [cicilan_kuliah]> INSERT INTO tbkasir SET kdkasir='KS2',Namakasir='Legina, A.MD'; Query OK, 1 row affected (0.036 sec) MariaDB [cicilan_kuliah]> INSERT INTO tbkasir SET kdkasir='KS3',Namakasir='Sri Rahayu Purba, A.MD'; Query OK, 1 row affected (0.036 sec) MariaDB [cicilan_kuliah]> INSERT INTO tbkasir SET kdkasir='KS4',Namakasir='Elisa, A.MD'; Query OK, 1 row affected (0.034 sec) MariaDB [cicilan_kuliah]> SELECT * FROM tbkasir; +---------+-----------+ | kdkasir | Namakasir | +---------+-----------+ | KS1 | Fahmi Firzada, A.MD | | KS2 | Legina, A.MD | | KS3 | Sri Rahayu Purba, A.MD | | KS4 | Elisa, A.MD | +---------+-----------+ 4 rows in set (0.001 sec)

25. Insert jenisbayar

Sintaks	INSERT INTO tbjenisbayar SET kdjenis='JN01',jenisbayar='pendaftaran',TahunAjaran='2023/2024'; INSERT INTO tbjenisbayar SET kdjenis='JN02',jenisbayar='Perlengkapan',TahunAjaran='2023/2024'; INSERT INTO tbjenisbayar SET kdjenis='JN03',jenisbayar='uang kuliah',TahunAjaran='2023/2024'; INSERT INTO tbjenisbayar SET kdjenis='JN04',jenisbayar='ICT',TahunAjaran='2023/2024'; SELECT * FROM tbjenisbayar;
Penjelasan	Dengan perintah Insert Into Set Tambahkan 4 Record di tbjenisbayar diatas dan Tampilkan tbjenisbayar
Hasil	MariaDB [cicilan_kuliah]> INSERT INTO tbjenisbayar SET kdjenis='JN01',jenisbayar='pendaftaran',TahunAjaran='2023/2024'; Query OK, 1 row affected (0.034 sec) MariaDB [cicilan_kuliah]> INSERT INTO tbjenisbayar SET kdjenis='JN02',jenisbayar='Perlengkapan',TahunAjaran='2023/2024'; Query OK, 1 row affected (0.029 sec) MariaDB [cicilan_kuliah]> INSERT INTO tbjenisbayar SET kdjenis='JN03',jenisbayar='uang kuliah',TahunAjaran='2023/2024'; Query OK, 1 row affected (0.073 sec) MariaDB [cicilan_kuliah]> INSERT INTO tbjenisbayar SET kdjenis='JN04',jenisbayar='ICT',TahunAjaran='2023/2024'; Query OK, 1 row affected (0.029 sec) MariaDB [cicilan_kuliah]> SELECT * FROM tbjenisbayar; +---------+--------------+-------------+ | kdjenis | jenisbayar | TahunAjaran | +---------+--------------+-------------+ | JN01 | pendaftaran | 2023/2024 | | JN02 | Perlengkapan | 2023/2024 | | JN03 | uang kuliah | 2023/2024 | | JN04 | ICT | 2023/2024 | +---------+--------------+-------------+ 4 rows in set (0.001 sec)

26. Insert tbmahasiswa

Sintaks	INSERT INTO tbmahasiswa (Nim,NamaMahasiswa,TempatLahir,TglLahir) VALUES ('2102161','Indah F Sari','P.Siantar','2003-07-15'), ('2303006','Najwa Mutia Armayani Nasuton','Laras','2005-11-13'), ('2303007','Novralina Anggraini Purba','P.Siantar','2002-11-07'), ('2303016','Hafizah Rahmi Lubis','P.Siantar','2004-04-22'), ('2303017','Zaskia Aulia Zahra','P.Siantar','2006-05-23'); SELECT * FROM tbmahasiswa;
Penjelasan	Dengan perintah Insert Into dengan Kolom Tambahkan 5 Record di tbmahasiswa diatas dan Tampilkan tbmahasiswa

Hasil	

```
MariaDB [cicilan_kuliah]>
MariaDB [cicilan_kuliah]> INSERT INTO tbmahasiswa (Nim,NamaMahasiswa,TempatLah
    -> ('2102161','Indah F Sari','P.Siantar','2003-07-15'),
    -> ('2303006','Najwa Mutia Armayani Nasuton','Laras','2005-11-13'),
    -> ('2303007','Novralina Anggraini Purba','P.Siantar','2002-11-07'),
    -> ('2303016','Hafizah Rahmi Lubis','P.Siantar','2004-04-22'),
    -> ('2303017','Zaskia Aulia Zahra','P.Siantar','2006-05-23');
Query OK, 5 rows affected (0.151 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [cicilan_kuliah]> SELECT * FROM tbmahasiswa;
+---------+-----------------------------+-------------+------------+
| Nim     | NamaMahasiswa               | TempatLahir | TglLahir   |
+---------+-----------------------------+-------------+------------+
| 2102161 | Indah F Sari                | P.Siantar   | 2003-07-15 |
| 2303006 | Najwa Mutia Armayani Nasuton | Laras       | 2005-11-13 |
| 2303007 | Novralina Anggraini Purba   | P.Siantar   | 2002-11-07 |
| 2303016 | Hafizah Rahmi Lubis         | P.Siantar   | 2004-04-22 |
| 2303017 | Zaskia Aulia Zahra          | P.Siantar   | 2006-05-23 |
+---------+-----------------------------+-------------+------------+
5 rows in set (0.001 sec)
```

27. Insert transaksi_cicilan

Sintaks	INSERT INTO transaksi_cicilan VALUES ('00001','2303017',1,'M231','2023-12-15','KS1','P',0,0,'JN03'), ('00002','2303017',1,'M231','2023-12-15','KS1','P',0,0,'JN04'), ('00003','2303017',1,'M231','2023-12-15','KS1','P',0,0,'JN02'), ('01205','2303006',1,'M231','2024-02-06','KS1','P',0,0,'JN04'), ('01422','2303007',1,'M231','2024-02-12','KS2','P',0,0,'JN04'), ('03851','2303016',1,'M231','2023-02-15','KS3','P',0,0,'JN01'), ('07706','2303006',1,'M231','2023-06-15','KS1','P',0,0,'JN01'), ('07707','2303006',1,'M231','2023-06-15','KS1','P',0,0,'JN02'), ('07710','2303006',1,'M231','2023-06-15','KS1','P',0,0,'JN03'), ('09027','2303007',1,'M231','2023-08-02','KS4','P',0,0,'JN01'), ('09028','2303007',1,'M231','2023-08-02','KS4','P',0,0,'JN02'), ('09355','2303007',1,'M231','2023-08-08','KS2','P',0,0,'JN03'), ('14886','2303016',1,'M231','2023-12-15','KS3','P',0,0,'JN02'), ('14887','2303016',1,'M231','2023-12-15','KS3','P',0,0,'JN03'), ('14888','2303016',1,'M231','2023-12-15','KS3','P',0,0,'JN04'); SELECT * FROM transaksi_cicilan;
Penjelasan	Dengan perintah Insert Into Multi Values Tambahkan 15 Record di transaksi_cicilan diatas dan Tampilkan transaksi_cicilan

Hasil

```
MariaDB [cicilan_kuliah]> INSERT INTO transaksi_cicilan VALUES
    -> ('00001','2303017',1,'201','2023-12-15','KS1','P',0,0,'JN03'),
    -> ('00002','2303017',1,'201','2023-12-15','KS1','P',0,0,'JN04'),
    -> ('00003','2303017',1,'201','2023-12-15','KS1','P',0,0,'JN02'),
    -> ('01205','2303006',1,'201','2024-02-06','KS1','P',0,0,'JN04'),
    -> ('01422','2303007',1,'201','2024-02-12','KS2','P',0,0,'JN04'),
    -> ('03851','2303016',1,'201','2023-02-15','KS3','P',0,0,'JN01'),
    -> ('07706','2303006',1,'201','2023-06-15','KS1','P',0,0,'JN01'),
    -> ('07707','2303006',1,'201','2023-06-15','KS1','P',0,0,'JN02'),
    -> ('07710','2303006',1,'201','2023-06-15','KS1','P',0,0,'JN03'),
    -> ('09027','2303007',1,'201','2023-08-02','KS4','P',0,0,'JN01'),
    -> ('09028','2303007',1,'201','2023-08-02','KS4','P',0,0,'JN02'),
    -> ('09355','2303007',1,'201','2023-08-08','KS2','P',0,0,'JN03'),
    -> ('14886','2303016',1,'201','2023-12-15','KS3','P',0,0,'JN02'),
    -> ('14887','2303016',1,'201','2023-12-15','KS3','P',0,0,'JN03'),
    -> ('14888','2303016',1,'201','2023-12-15','KS3','P',0,0,'JN04');
Query OK, 15 rows affected (0.130 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

```
MariaDB [cicilan_kuliah]> SELECT * FROM transaksi_cicilan;
+-----------+---------+---------+---------+-----------+---------+--------+-------------+-------+-------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar  | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+-----------+---------+---------+---------+-----------+---------+--------+-------------+-------+-------------+
| 00001     | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |           0 |     0 | JN03        |
| 00002     | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |           0 |     0 | JN04        |
| 00003     | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |           0 |     0 | JN02        |
| 01205     | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |           0 |     0 | JN04        |
| 01422     | 2303007 |       1 | 201     | 2024-02-12 | KS2     | P      |           0 |     0 | JN04        |
| 03851     | 2303016 |       1 | 201     | 2023-02-15 | KS3     | P      |           0 |     0 | JN01        |
| 07706     | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |           0 |     0 | JN01        |
| 07707     | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |           0 |     0 | JN02        |
| 07710     | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |           0 |     0 | JN03        |
| 09027     | 2303007 |       1 | 201     | 2023-08-02 | KS4     | P      |           0 |     0 | JN01        |
| 09028     | 2303007 |       1 | 201     | 2023-08-02 | KS4     | P      |           0 |     0 | JN02        |
| 09355     | 2303007 |       1 | 201     | 2023-08-08 | KS2     | P      |           0 |     0 | JN03        |
| 14886     | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |           0 |     0 | JN02        |
| 14887     | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |           0 |     0 | JN03        |
| 14888     | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |           0 |     0 | JN04        |
+-----------+---------+---------+---------+-----------+---------+--------+-------------+-------+-------------+
15 rows in set (0.002 sec)
```

28. Update UangSejumlah sesuai jenis bayar

Sintaks	UPDATE transaksi_cicilan SET UangSejumlah=200000 WHERE KdJenisBayar='JN01'; UPDATE transaksi_cicilan SET UangSejumlah=1000000 WHERE KdJenisBayar='JN02'; UPDATE transaksi_cicilan SET UangSejumlah=4000000 WHERE KdJenisBayar='JN03'; UPDATE transaksi_cicilan SET UangSejumlah=2000000 WHERE KdJenisBayar='JN04';
Penjelasan	Dengan perintah Update Perbaiki data Transaksi_cicilan Untuk JN01 UangSejumlah=200000 Untuk JN02 UangSejumlah=1000000 Untuk JN03 UangSejumlah=4000000 Untuk JN04 UangSejumlah=2000000

Hasil	

```
MariaDB [CicilanKuliah]> UPDATE transaksi_cicilan SET UangSejumlah=200000 WHERE KdJenisBayar='JN01';
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [CicilanKuliah]> UPDATE transaksi_cicilan SET UangSejumlah=1000000 WHERE KdJenisBayar='JN02';
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [CicilanKuliah]> UPDATE transaksi_cicilan SET UangSejumlah=4000000 WHERE KdJenisBayar='JN03';
Query OK, 0 rows affected (0.100 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [CicilanKuliah]> UPDATE transaksi_cicilan SET UangSejumlah=2000000 WHERE KdJenisBayar='JN04';
Query OK, 0 rows affected (0.000 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [cicilan_kuliah]> select *from transaksi_cicilan;
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar   | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| 00001      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      4000000 |     0 | JN03         |
| 00002      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      2000000 |     0 | JN04         |
| 00003      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      1000000 |     0 | JN02         |
| 01205      | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |      2000000 |     0 | JN04         |
| 01422      | 2303007 |       1 | 201     | 2024-02-12 | KS2     | P      |      2000000 |     0 | JN04         |
| 03851      | 2303016 |       1 | 201     | 2023-02-15 | KS3     | P      |       200000 |     0 | JN01         |
| 07706      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |       200000 |     0 | JN01         |
| 07707      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      1000000 |     0 | JN02         |
| 07710      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      4000000 |     0 | JN03         |
| 09027      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | P      |       200000 |     0 | JN01         |
| 09028      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | P      |      1000000 |     0 | JN02         |
| 09355      | 2303007 |       1 | 201     | 2023-08-08 | KS2     | P      |      4000000 |     0 | JN03         |
| 14886      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |      1000000 |     0 | JN02         |
| 14887      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |      4000000 |     0 | JN03         |
| 14888      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |      2000000 |     0 | JN04         |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
15 rows in set (0.001 sec)
```

29. Update Denda semua jadi 10000

Sintaks	UPDATE transaksi_cicilan SET Denda=10000;
Penjelasan	Dengan perintah Update Perbaiki data Transaksi_cicilan Untuk denda seluruhnya adalah 10000
Hasil	

```
MariaDB [CicilanKuliah]> UPDATE transaksi_cicilan SET Denda=10000;
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [cicilan_kuliah]> select *from transaksi_cicilan;
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar   | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| 00001      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 00002      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      2000000 | 10000 | JN04         |
| 00003      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 01205      | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |      2000000 | 10000 | JN04         |
| 01422      | 2303007 |       1 | 201     | 2024-02-12 | KS2     | P      |      2000000 | 10000 | JN04         |
| 03851      | 2303016 |       1 | 201     | 2023-02-15 | KS3     | P      |       200000 | 10000 | JN01         |
| 07706      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |       200000 | 10000 | JN01         |
| 07707      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 07710      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 09027      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | P      |       200000 | 10000 | JN01         |
| 09028      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | P      |      1000000 | 10000 | JN02         |
| 09355      | 2303007 |       1 | 201     | 2023-08-08 | KS2     | P      |      4000000 | 10000 | JN03         |
| 14886      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |      1000000 | 10000 | JN02         |
| 14887      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |      4000000 | 10000 | JN03         |
| 14888      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | P      |      2000000 | 10000 | JN04         |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
15 rows in set (0.001 sec)
```

30. Update sesi mahasiswa tertentu

Sintaks	UPDATE transaksi_cicilan SET KdSesi='S' WHERE Nim='2303007'; UPDATE transaksi_cicilan SET KdSesi='M' WHERE Nim='2303016';
Penjelasan	Dengan perintah Update Perbaiki data Transaksi_cicilan Untuk NIM 2303007 KDSESI =S Untuk NIM 2303016 KDSESI =M
Hasil	

```
MariaDB [CicilanKuliah]> UPDATE transaksi_cicilan SET KdSesi='S' WHERE Nim='2303007';
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [CicilanKuliah]> UPDATE transaksi_cicilan SET KdSesi='M' WHERE Nim='2303016';
Query OK, 0 rows affected (0.002 sec)
Rows matched: 0  Changed: 0  Warnings: 0
MariaDB [cicilan_kuliah]> select *from transaksi_cicilan;
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar   | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| 00001      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 00002      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      2000000 | 10000 | JN04         |
| 00003      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 01205      | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |      2000000 | 10000 | JN04         |
| 01422      | 2303007 |       1 | 201     | 2024-02-12 | KS2     | S      |      2000000 | 10000 | JN04         |
| 03851      | 2303016 |       1 | 201     | 2023-02-15 | KS3     | M      |       200000 | 10000 | JN01         |
| 07706      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |       200000 | 10000 | JN01         |
| 07707      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 07710      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 09027      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |       200000 | 10000 | JN01         |
| 09028      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |      1000000 | 10000 | JN02         |
| 09355      | 2303007 |       1 | 201     | 2023-08-08 | KS2     | S      |      4000000 | 10000 | JN03         |
| 14886      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      1000000 | 10000 | JN02         |
| 14887      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      4000000 | 10000 | JN03         |
| 14888      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      2000000 | 10000 | JN04         |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
15 rows in set (0.001 sec)
```

31. Lihat tabel

Sintaks	SELECT * FROM transaksi_cicilan;
Penjelasan	Tampilkan transaksi_cicilan
Hasil	

```
MariaDB [cicilan_kuliah]> select *from transaksi_cicilan;
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar   | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| 00001      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 00002      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      2000000 | 10000 | JN04         |
| 00003      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 01205      | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |      2000000 | 10000 | JN04         |
| 01422      | 2303007 |       1 | 201     | 2024-02-12 | KS2     | S      |      2000000 | 10000 | JN04         |
| 03851      | 2303016 |       1 | 201     | 2023-02-15 | KS3     | M      |       200000 | 10000 | JN01         |
| 07706      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |       200000 | 10000 | JN01         |
| 07707      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 07710      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 09027      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |       200000 | 10000 | JN01         |
| 09028      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |      1000000 | 10000 | JN02         |
| 09355      | 2303007 |       1 | 201     | 2023-08-08 | KS2     | S      |      4000000 | 10000 | JN03         |
| 14886      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      1000000 | 10000 | JN02         |
| 14887      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      4000000 | 10000 | JN03         |
| 14888      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      2000000 | 10000 | JN04         |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
15 rows in set (0.001 sec)
```

32. Delete record dengan NoKwitansi = 00001

Sintaks	DELETE FROM transaksi_cicilan WHERE NoKwitansi='00001';
Penjelasan	Dengan perintah DELETE Hapus Record dengan NoKwitansi = 00001
Hasil	

```
MariaDB [CicilanKuliah]> DELETE FROM transaksi_cicilan WHERE NoKwitansi='00001';
Query OK, 0 rows affected (0.045 sec)

MariaDB [cicilan_kuliah]>  select *from transaksi_cicilan;
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar   | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| 00002      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      2000000 | 10000 | JN04         |
| 00003      | 2303017 |       1 | 201     | 2023-12-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 01205      | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |      2000000 | 10000 | JN04         |
| 01422      | 2303007 |       1 | 201     | 2024-02-12 | KS2     | S      |      2000000 | 10000 | JN04         |
| 03851      | 2303016 |       1 | 201     | 2023-02-15 | KS3     | M      |       200000 | 10000 | JN01         |
| 07706      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |       200000 | 10000 | JN01         |
| 07707      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 07710      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 09027      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |       200000 | 10000 | JN01         |
| 09028      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |      1000000 | 10000 | JN02         |
| 09355      | 2303007 |       1 | 201     | 2023-08-08 | KS2     | S      |      4000000 | 10000 | JN03         |
| 14886      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      1000000 | 10000 | JN02         |
| 14887      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      4000000 | 10000 | JN03         |
| 14888      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      2000000 | 10000 | JN04         |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
14 rows in set (0.001 sec)
```

33. Delete record dengan NoKwitansi 00002 dan 00003

Sintaks	DELETE FROM transaksi_cicilan WHERE NoKwitansi IN ('00002','00003');
Penjelasan	Dengan perintah DELETE Hapus Record dengan NoKwitansi = 00002 dan 00003 dengan perintah IN
Hasil	

```
MariaDB [CicilanKuliah]> DELETE FROM transaksi_cicilan WHERE NoKwitansi IN ('00002','00003');
Query OK, 0 rows affected (0.015 sec)
MariaDB [cicilan_kuliah]> SELECT * FROM transaksi_cicilan;
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| NoKwitansi | Nim     | KdProdi | KdKelas | TglBayar   | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
| 01205      | 2303006 |       1 | 201     | 2024-02-06 | KS1     | P      |      2000000 | 10000 | JN04         |
| 01422      | 2303007 |       1 | 201     | 2024-02-12 | KS2     | S      |      2000000 | 10000 | JN04         |
| 03851      | 2303016 |       1 | 201     | 2023-02-15 | KS3     | M      |       200000 | 10000 | JN01         |
| 07706      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |       200000 | 10000 | JN01         |
| 07707      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      1000000 | 10000 | JN02         |
| 07710      | 2303006 |       1 | 201     | 2023-06-15 | KS1     | P      |      4000000 | 10000 | JN03         |
| 09027      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |       200000 | 10000 | JN01         |
| 09028      | 2303007 |       1 | 201     | 2023-08-02 | KS4     | S      |      1000000 | 10000 | JN02         |
| 09355      | 2303007 |       1 | 201     | 2023-08-08 | KS2     | S      |      4000000 | 10000 | JN03         |
| 14886      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      1000000 | 10000 | JN02         |
| 14887      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      4000000 | 10000 | JN03         |
| 14888      | 2303016 |       1 | 201     | 2023-12-15 | KS3     | M      |      2000000 | 10000 | JN04         |
+------------+---------+---------+---------+------------+---------+--------+--------------+-------+--------------+
12 rows in set (0.054 sec)
```

34. Tampilkan hasil

Sintaks	SELECT * FROM transaksi_cicilan;

Penjelasan	Tampilkan Tabel transaksi_cicilan;
Hasil	MariaDB [cicilan_kuliah]> SELECT * FROM transaksi_cicilan;

```
+-----------+--------+--------+--------+-----------+--------+-------+-------------+------+-------------+
| NoKwitansi | Nim    | KdProdi | KdKelas | TglBayar  | KdKasir | KdSesi | UangSejumlah | Denda | KdJenisBayar |
+-----------+--------+--------+--------+-----------+--------+-------+-------------+------+-------------+
| 01205     | 2303006 |      1 | 201    | 2024-02-06 | KS1    | P     |     2000000 | 10000 | JN04        |
| 01422     | 2303007 |      1 | 201    | 2024-02-12 | KS2    | S     |     2000000 | 10000 | JN04        |
| 03851     | 2303016 |      1 | 201    | 2023-02-15 | KS3    | M     |      200000 | 10000 | JN01        |
| 07706     | 2303006 |      1 | 201    | 2023-06-15 | KS1    | P     |      200000 | 10000 | JN01        |
| 07707     | 2303006 |      1 | 201    | 2023-06-15 | KS1    | P     |     1000000 | 10000 | JN02        |
| 07710     | 2303006 |      1 | 201    | 2023-06-15 | KS1    | P     |     4000000 | 10000 | JN03        |
| 09027     | 2303007 |      1 | 201    | 2023-08-02 | KS4    | S     |      200000 | 10000 | JN01        |
| 09028     | 2303007 |      1 | 201    | 2023-08-02 | KS4    | S     |     1000000 | 10000 | JN02        |
| 09355     | 2303007 |      1 | 201    | 2023-08-08 | KS2    | S     |     4000000 | 10000 | JN03        |
| 14886     | 2303016 |      1 | 201    | 2023-12-15 | KS3    | M     |     1000000 | 10000 | JN02        |
| 14887     | 2303016 |      1 | 201    | 2023-12-15 | KS3    | M     |     4000000 | 10000 | JN03        |
| 14888     | 2303016 |      1 | 201    | 2023-12-15 | KS3    | M     |     2000000 | 10000 | JN04        |
+-----------+--------+--------+--------+-----------+--------+-------+-------------+------+-------------+
12 rows in set (0.054 sec)
```

6.2 CPMK DML dan Select Statement

1. Membuat Database

Syntax:	CREATE DATABASE UjiKompetensi2;
Penjelasan:	Membuat database dengan nama Ujikompetensi2
Hasil:	MariaDB [CicilanKuliah]> CREATE DATABASE UJIKompetensi2; Query OK, 1 row affected (0.002 sec)

2. Tampilkan database

Syntax:	SHOW DATABASES;
Penjelasan:	Untuk menampilkan daftar semua database yang ada di server database tersebut

Hasil:	`MariaDB [CicilanKuliah]> SHOW DATABASES;`
	`+--------------------+`
	`| Database |`
	`+--------------------+`
	`| cicilankuliah |`
	`| information_schema |`
	`| mysql |`
	`| performance_schema |`
	`| phpmyadmin |`
	`| test |`
	`| ujikompetensi2 |`
	`+--------------------+`
	`7 rows in set (0.001 sec)`

3. Masuk ke dalam database

Syntax:	USE UjiKompetensi2;
Penjelasan:	Masuk kedalam database Ujikompetensi2
Hasil;	`MariaDB [CicilanKuliah]> USE UJIKompetensi2;` `Database changed`

4. Membuat tabel

Syntax:	CREATE TABLE tbmahasiswa (-> NimMahasiswa VARCHAR(10) PRIMARY KEY, ->NamaMahasiswa VARCHAR(50), ->TglLahir DATE);
Penjelasan:	Buat tabel tbmahasiswa tahap pertama diatas

Hasil:

```
MariaDB [UJIKompetensi2]> CREATE TABLE tbmahasiswa (
    ->     Nim CHAR(10) PRIMARY KEY,
    ->     NamaMahasiswa VARCHAR(50),
    ->     TglLahir DATE
    -> );
Query OK, 0 rows affected (0.249 sec)

MariaDB [UJIKompetensi_2]> desc tbmahasiswa;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Nim           | char(10)    | NO   | PRI | NULL    |       |
| NamaMahasiswa | varchar(50) | YES  |     | NULL    |       |
| TglLahir      | date        | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.055 sec)
```

5. Perintah Alter Table

Syntax:	ALTER TABLE tbmahasiswa ADD Agama VARCHAR(20) AFTER NamaMahasiswa, ADD Jenis_Kelamin VARCHAR(10) AFTER Agama, ADD Tempat_Tinggal VARCHAR(30) AFTER TglLahir, ADD UTS INT AFTER TglLahir, ADD Gaji_Ortu INT;
Penjelasan:	Dengan Perintah Alter table Tambahkan Kolom sebagai berikut : Agama dan Jenis Kelamin di bawah NamaMahasiswa Tempat_Tinggal dibawah TglLahir UTS dan Gaji_Ortu dibawah TglLahir di kolom terakhir
Hasil:	MariaDB [UJIKompetensi2]> ALTER TABLE tbmahasiswa -> ADD Agama VARCHAR(20) AFTER NamaMahasiswa, -> ADD Jenis_Kelamin VARCHAR(15) AFTER Agama, -> ADD Tempat_Tinggal VARCHAR(30) AFTER TglLahir, -> ADD UTS INT AFTER Tempat_Tinggal, -> ADD Gaji_Ortu INT AFTER UTS; Query OK, 0 rows affected (0.106 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [UJIKompetensi_2]> desc tbmahasiswa; +----------+----------+----+----+----+----+ | Field | Type | Null | Key | Default | Extra | +----------+----------+----+----+----+----+ | Nim | char(10) | NO | PRI | NULL | | | NamaMahasiswa | varchar(50) | YES | | NULL | | | Agama | varchar(20) | YES | | NULL | | | Jenis_Kelamin | varchar(10) | YES | | NULL | | | TglLahir | date | YES | | NULL | | | UTS | int(11) | YES | | NULL | | | Tempat_Tinggal | varchar(30) | YES | | NULL | | | Gaji_Ortu | int(11) | YES | | NULL | | +----------+----------+----+----+----+----+ 8 rows in set (0.009 sec)

6. Mengisi data

Syntax:	INSERT INTO tbmahasiswa (NimMahasiswa, NamaMahasiswa, Agama, Jenis_Kelamin, TglLahir, Tempat_Tinggal, UTS, Gaji_Ortu) VALUES ('2302001','ABDULLAH MUJADID AL QAWI','Islam','Laki-laki','2003-07-15','Medan',72,3000000), ('2302002','ADINDA NABILA','Islam','Perempuan','2005-11-13','Medan',90,7500000), ('2302005','ANGGUN BUNGA LESTARI LUBIS','Islam','Perempuan','2002-11-07','Bogor',80,6200000), ('2302006','ANNISA TRINABILA','Islam','Perempuan','2004-04-22','Bogor',72,4700000), ('2302008','BAGAS TRY ATMAJA','Islam','Laki-laki','2004-04-22','Medan',91,7100000),

	('2302009','BUNGA SABILA','Islam','Perempuan','2006-05-23','Medan',75,4400000), ('2302364','ALMAH WIJAYA ALHAKIM','Islam','Laki-laki','2002-11-07','Jakarta',90,4800000);
Penjelasan	Isikan Record Tabel tbmahasiswa yang sudah ditambah dengan Alter (Bebas dengan insert apapun)
Hasil:	

```
MariaDB [UJIKompetensi2]> INSERT INTO tbmahasiswa VALUES
    -> ('2302001','ABDULLAH MUJADID AL QAWI','ISLAM','LAKI-LAKI','2003-07-15','MEDAN',72,3000000),
    -> ('2302002','ADINDA NABILA','ISLAM','PEREMPUAN','2005-11-13','MEDAN',90,7500000),
    -> ('2302005','ANGGUN BUNGA LESTARI LUBIS','ISLAM','PEREMPUAN','2002-11-07','BOGOR',80,6200000),
    -> ('2302006','ANNISA TRINABILA','ISLAM','PEREMPUAN','2004-04-22','BOGOR',72,4700000),
    -> ('2302008','BAGAS TRY ATMAJA','ISLAM','LAKI-LAKI','2004-04-22','MEDAN',91,7100000),
    -> ('2302009','BUNGA SABILA','ISLAM','PEREMPUAN','2006-05-23','MEDAN',75,4400000),
    -> ('2302364','ALMAH WIJAYA ALHAKIM','ISLAM','LAKI-LAKI','2002-11-07','JAKARTA',90,4800000);
Query OK, 7 rows affected (0.062 sec)
Records: 7  Duplicates: 0  Warnings: 0

MariaDB [UJIKompetensi_2]> select *from tbmahasiswa;
+---------+---------------------------+-------+--------------+------------+------+---------------+-----------+
| Nim     | NamaMahasiswa             | Agama | Jenis_Kelamin | TglLahir   | UTS  | Tempat_Tinggal | Gaji_Ortu |
+---------+---------------------------+-------+--------------+------------+------+---------------+-----------+
| 2302001 | ABDULLAH MUJADID AL QAWI  | ISLAM | LAKI-LAKI    | 2003-07-15 |    0 | 72            |   3000000 |
| 2302002 | ADINDA NABILA             | ISLAM | PEREMPUAN    | 2005-11-13 |    0 | 90            |   7500000 |
| 2302005 | ANGGUN BUNGA LESTARI LUBIS | ISLAM | PEREMPUAN    | 2002-11-07 |    0 | 80            |   6200000 |
| 2302006 | ANNISA TRINABILA          | ISLAM | PEREMPUAN    | 2004-04-22 |    0 | 72            |   4700000 |
| 2302008 | BAGAS TRY ATMAJA          | ISLAM | LAKI-LAKI    | 2004-04-22 |    0 | 91            |   7100000 |
| 2302009 | BUNGA SABILA              | ISLAM | PEREMPUAN    | 2006-05-23 |    0 | 75            |   4400000 |
| 2302364 | ALMAH WIJAYA ALHAKIM      | ISLAM | LAKI-LAKI    | 2002-11-07 |    0 | 90            |   4800000 |
+---------+---------------------------+-------+--------------+------------+------+---------------+-----------+
7 rows in set (0.001 sec)
```

7. Menampilkan data

Syntax:	SELECT NimMahasiswa AS 'Nim Mahasiswa', NamaMahasiswa AS 'Nama Mahasiswa',Tempat_Tinggal AS 'Tempat Tinggal' FROM tbmahasiswa;
Penjelasan:	Tampilkan data dibawah ini dengan Fungsi As String

Hasil:	

```
MariaDB [UJIKompetensi2]> SELECT Nim AS "Nim Mahasiswa",
    ->        NamaMahasiswa AS "Nama Mahasiswa",
    ->        Tempat_Tinggal AS "Tempat Tinggal"
    -> FROM tbmahasiswa;
+---------------+----------------------------+----------------+
| Nim Mahasiswa | Nama Mahasiswa             | Tempat Tinggal |
+---------------+----------------------------+----------------+
| 2302001       | ABDULLAH MUJADID AL QAWI   | MEDAN          |
| 2302002       | ADINDA NABILA              | MEDAN          |
| 2302005       | ANGGUN BUNGA LESTARI LUBIS | BOGOR          |
| 2302006       | ANNISA TRINABILA           | BOGOR          |
| 2302008       | BAGAS TRY ATMAJA           | MEDAN          |
| 2302009       | BUNGA SABILA               | MEDAN          |
| 2302364       | ALMAH WIJAYA ALHAKIM       | JAKARTA        |
+---------------+----------------------------+----------------+
7 rows in set (0.033 sec)
```

8. Menampilkan data

Syntax:	SELECT NimMahasiswa AS NIM, NamaMahasiswa, TglLahir, TRUNCATE(DATEDIFF(CURDATE(),TglLahir)/365,0) AS Umur FROM tbmahasiswa;
Penjelasan:	Tampilkan data dibawah ini dengan Fungsi TRUNCATE(DATEDIFF())
Hasil:	

```
MariaDB [UJIKompetensi2]> SELECT Nim, NamaMahasiswa, TglLahir,
    ->        TRUNCATE(DATEDIFF(CURDATE(), TglLahir)/365,0) AS Umur
    -> FROM tbmahasiswa;
+---------+----------------------------+------------+------+
| Nim     | NamaMahasiswa              | TglLahir   | Umur |
+---------+----------------------------+------------+------+
| 2302001 | ABDULLAH MUJADID AL QAWI   | 2003-07-15 |   22 |
| 2302002 | ADINDA NABILA              | 2005-11-13 |   19 |
| 2302005 | ANGGUN BUNGA LESTARI LUBIS | 2002-11-07 |   22 |
| 2302006 | ANNISA TRINABILA           | 2004-04-22 |   21 |
| 2302008 | BAGAS TRY ATMAJA           | 2004-04-22 |   21 |
| 2302009 | BUNGA SABILA               | 2006-05-23 |   19 |
| 2302364 | ALMAH WIJAYA ALHAKIM       | 2002-11-07 |   22 |
+---------+----------------------------+------------+------+
7 rows in set (0.092 sec)
```

9. Menampilkan Data

Syntax:	SELECT NimMahasiswa AS NIM, NamaMahasiswa, Gaji_Ortu AS 'Gaji Orang Tua', (Gaji_Ortu*12) AS 'Gaji Orang Tua Satu Tahun' FROM tbmahasiswa;
Penjerlasan:	Mengambil data Gaji_Ortu bulanan, Menghitung gaji dalam 1 tahun dengan cara Gaji_Ortu * 12.
Hasil:	

```
MariaDB [UJIKompetensi2]> SELECT Nim, NamaMahasiswa,
    ->        Gaji_Ortu AS "Gaji Orang Tua",
    ->        (Gaji_Ortu*12) AS "Gaji Orang Tua Satu Tahun"
    -> FROM tbmahasiswa;
+---------+---------------------------+----------------+---------------------------+
| Nim     | NamaMahasiswa             | Gaji Orang Tua | Gaji Orang Tua Satu Tahun |
+---------+---------------------------+----------------+---------------------------+
| 2302001 | ABDULLAH MUJADID AL QAWI  |        3000000 |                  36000000 |
| 2302002 | ADINDA NABILA             |        7500000 |                  90000000 |
| 2302005 | ANGGUN BUNGA LESTARI LUBIS |        6200000 |                  74400000 |
| 2302006 | ANNISA TRINABILA          |        4700000 |                  56400000 |
| 2302008 | BAGAS TRY ATMAJA          |        7100000 |                  85200000 |
| 2302009 | BUNGA SABILA              |        4400000 |                  52800000 |
| 2302364 | ALMAH WIJAYA ALHAKIM      |        4800000 |                  57600000 |
+---------+---------------------------+----------------+---------------------------+
7 rows in set (0.001 sec)
```

10. Tampilkan data dibawah ini dengan fungsi string

Syntax :	SELECT NIM, NamaMahasiswa, 'MEMILIKI GAJI ORANG TUA PERBULAN' AS Keterangan, Gaji AS "GAJI BULANAN ORANG TUA" FROM Mahasiswa;
Penjelasan	Tampilkan data dibawah ini dengan fungsi string
Hasil :	

```
MariaDB [UJIKompetensi2]> SELECT Nim, NamaMahasiswa,
    ->        CONCAT('MEMILIKI GAJI ORANG TUA PERBULAN') AS "Memiliki Gaji Orang Tua Perbulan",
    ->        Gaji_Ortu AS "Gaji Bulanan Orang Tua"
    -> FROM tbmahasiswa;
+---------+---------------------------+---------------------------------+-----------------------+
| Nim     | NamaMahasiswa             | Memiliki Gaji Orang Tua Perbulan | Gaji Bulanan Orang Tua |
+---------+---------------------------+---------------------------------+-----------------------+
| 2302001 | ABDULLAH MUJADID AL QAWI  | MEMILIKI GAJI ORANG TUA PERBULAN |                3000000 |
| 2302002 | ADINDA NABILA             | MEMILIKI GAJI ORANG TUA PERBULAN |                7500000 |
| 2302005 | ANGGUN BUNGA LESTARI LUBIS | MEMILIKI GAJI ORANG TUA PERBULAN |                6200000 |
| 2302006 | ANNISA TRINABILA          | MEMILIKI GAJI ORANG TUA PERBULAN |                4700000 |
| 2302008 | BAGAS TRY ATMAJA          | MEMILIKI GAJI ORANG TUA PERBULAN |                7100000 |
| 2302009 | BUNGA SABILA              | MEMILIKI GAJI ORANG TUA PERBULAN |                4400000 |
| 2302364 | ALMAH WIJAYA ALHAKIM      | MEMILIKI GAJI ORANG TUA PERBULAN |                4800000 |
+---------+---------------------------+---------------------------------+-----------------------+
7 rows in set (0.001 sec)
```

11. Tampilkan fungsi dibawah ini dengan fungsi case when

Syntax :	SELECT NIM, NamaMahasiswa, UTS, CASE WHEN UTS >= 85 THEN 'A' WHEN UTS >= 75 THEN 'B' WHEN UTS >= 65 THEN 'C' WHEN UTS >= 55 THEN 'D' ELSE 'E' END AS NilaiHuruf FROM Mahasiswa;
Penjelasan :	Tampilkan fungsi di bawah ini dengan fungsi case when
Hasil :	+----------+----------+----+----+ | NimMahasiswa | NamaMahasiswa | UTS | NilaiHuruf | +----------+----------+----+----+ | 2302001 | ABDULLAH MUJADID AL QAWI | 72 | B | | 2302002 | ADINDA NABILA | 90 | A | | 2302005 | ANGGUN BUNGA LESTARI LUBIS | 80 | B | | 2302006 | ANNISA TRINABILA | 72 | B | | 2302008 | BAGAS TRY ATMAJA | 91 | A | | 2302009 | BUNGA SABILA | 75 | B | | 2302364 | ALMAH WIJAYA ALHAKIM | 90 | A | +----------+----------+----+----+

12. Menampilkan semua data yang memiliki huruf ‘A' tidak menampilkan ‘AN'

Syntax :	SELECT * FROM tbmahasiswa -> WHERE NamaMahasiswa LIKE 'A%' -> AND NamaMahasiswa NOT LIKE 'AN%';
Penjelasan	Tampilkan Semua data TbMahasiswaYang memiliki NamaMahasiswa Huruf ‘A’ tetapi Tidak menampilkan NamaMahasiswa ‘AN’
Hasil :	+----------+----------+----+----+----+----+----+----+ | NimMahasiswa | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | UTS | Tempat_Tinggal | Gaji_Ortu | +----------+----------+----+----+----+----+----+----+ | 2302001 | ABDULLAH MUJADID AL QAWI | Islam | Laki-laki | 2003-07-15 | 72 | Medan | 3000000 | | 2302002 | ADINDA NABILA | Islam | Perempuan | 2005-11-13 | 90 | Medan | 7500000 | | 2302364 | ALMAH WIJAYA ALHAKIM | Islam | Laki-laki | 2002-11-07 | 90 | Jakarta | 4800000 | +----------+----------+----+----+----+----+----+----+

13. Menampilkan semua data yang memiliki Huruf ke-4 nya ‘A' huruf 1,2,3 dst. bebas

Syntax :	SELECT * FROM tbmahasiswa -> WHERE NamaMahasiswa LIKE '___A%';
Penjelasan	Tampilkan Semua data TbMahasiswaYang memiliki NamaMahasiswa Hurufke 4nya adalah ‘A’huruf ke 1,2,3bebas dan huruf terakhir Bebas.
Hasil :	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302364	ALMAH WIJAYA ALHAKIM	Islam	Laki-laki	2002-11-07	90	Jakarta	4800000

14. NamaMahasiswa huruf terakhir “A”

Syntax	SELECT * FROM tbmahasiswa WHERE NamaMahasiswa LIKE '%A
Penjelasan	Tampilkan nama mahasiswa yang berawalan huruf A di tabel TbMahasiswa
Hasil :	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302002	ADINDA NABILA	Islam	Perempuan	2005-11-13	90	Medan	7500000
2302006	ANNISA TRINABILA	Islam	Perempuan	2004-04-22	72	Bogor	4700000
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302009	BUNGA SABILA	Islam	Perempuan	2006-05-23	75	Medan	4400000

15. NamaMahasiswa diawali “B”

Syntax :	SELECT * FROM tbmahasiswa WHERE NamaMahasiswa LIKE 'B%';
Penjelasan :	Tampilkan nama mahasiswa yang berawalan huruf B di tabel Tbmahasiswa
Hasil :	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302009	BUNGA SABILA	Islam	Perempuan	2006-05-23	75	Medan	4400000

16. Menampilkan data yang memiliki nilai UTS diantara 85 sd 95 jenis_kelamin Perempuan (Perintah Between)

Syntax :	SELECT * FROM tbmahasiswa -> WHERE UTS BETWEEN 85 AND 95 -> AND Jenis_Kelamin = 'Perempuan';
Penjelasan	Tampilkan Semua data TbMahasiswaYang memiliki Nilai UTS diantara 85 sd 95untuk jenis_kelamin adalah Perempuan. (dengan Perintah Between).
Hasil :	+----------+----------+-------+----------+----------+-----+----------+----------+ | NimMahasiswa | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | UTS | Tempat_Tinggal | Gaji_Ortu | +----------+----------+-------+----------+----------+-----+----------+----------+ | 2302002 | ADINDA NABILA | Islam | Perempuan | 2005-11-13 | 90 | Medan | 7500000 | +----------+----------+-------+----------+----------+-----+----------+----------+

17. Menampilkan data yang memiliki nilai UTS diantara 70 sd 80 jenis_kelamin Perempuan (Perintah AND)

Syntax :	SELECT * FROM tbmahasiswa -> WHERE UTS >= 70 AND UTS <= 80 -> AND Jenis_Kelamin = 'Perempuan';
Penjelasan	Tampilkan Semua data TbMahasiswaYang memiliki Nilai UTS diantara 70 sd 80untuk jenis_kelaminadalah Perempuan. (dengan Perintah AND).
Hasil :	+----------+----------+-------+----------+----------+-----+----------+----------+ | NimMahasiswa | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | UTS | Tempat_Tinggal | Gaji_Ortu | +----------+----------+-------+----------+----------+-----+----------+----------+ | 2302005 | ANGGUN BUNGA LESTARI LUBIS | Islam | Perempuan | 2002-11-07 | 80 | Bogor | 6200000 | | 2302006 | ANNISA TRINABILA | Islam | Perempuan | 2004-04-22 | 72 | Bogor | 4700000 | | 2302009 | BUNGA SABILA | Islam | Perempuan | 2006-05-23 | 75 | Medan | 4400000 | +----------+----------+-------+----------+----------+-----+----------+----------+

18. Menampilkan semua data yang memiliki Tempat_Tinggal di Medan dan Jakarta (perintah IN)

Sintaks	SELECT * FROM tbmahasiswa Where Tempat_Tinggal IN ('Medan','Jakarta');
Penjelasan	Tampilkan Semua data tbmahasiswa yang memiliki Tempat_Tinggal di Medan dan Jakarta menggunakan Perintah IN.

Hasil	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302001	ABDULLAH MUJADID AL QAWI	Islam	Laki-laki	2003-07-15	72	Medan	3000000
2302002	ADINDA NABILA	Islam	Perempuan	2005-11-13	90	Medan	7500000
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302009	BUNGA SABILA	Islam	Perempuan	2006-05-23	75	Medan	4400000
2302364	ALMAH WIJAYA ALHAKIM	Islam	Laki-laki	2002-11-07	90	Jakarta	4800000

19. Menampilkan semua data yang memiliki Tempat_Tinggal di Medan dan Bogor (Perintah OR).

Sintaks	SELECT * FROM tbmahasiswa WHERE Tempat_Tinggal = 'Medan' OR Tempat_Tinggal = 'Bogor';
Penjelasan	Tampilkan semua data tbmahasiswa yang memiliki Tempat_Tinggal di Medan dan Bogor menggunakan Perintah OR
Hasil	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302001	ABDULLAH MUJADID AL QAWI	Islam	Laki-laki	2003-07-15	72	Medan	3000000
2302002	ADINDA NABILA	Islam	Perempuan	2005-11-13	90	Medan	7500000
2302005	ANGGUN BUNGA LESTARI LUBIS	Islam	Perempuan	2002-11-07	80	Bogor	6200000
2302006	ANNISA TRINABILA	Islam	Perempuan	2004-04-22	72	Bogor	4700000
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302009	BUNGA SABILA	Islam	Perempuan	2006-05-23	75	Medan	4400000

20. Tampilkan semua data Tbmahasiswa yang memiliki nama mahasiswa dimana huruf ke 2 adalah U dan huruf ke 4 adalah G yang lain bebas.

Syntax	SELECT * FROM TbMahasiswa WHERE NamaMahasiswa LIKE '_U_G%';
Penjelasan	Menampilkan semua data dari tabel TbMahasiswa di mana nama mahasiswa memiliki huruf ke-2 adalah 'U' dan huruf ke-4 adalah 'G'
Hasil	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302009	BUNGA SABILA	Islam	Perempuan	2006-05-23	75	Medan	4400000

21. Tampilkan semua data TbMahasiswa yang memiliki jenis kelamin adalah Tidak Perempuan

Syntax	SELECT * FROM TbMahasiswa WHERE Jenis_Kelamin != 'PEREMPUAN';
Penjelasan	Menampilkan semua data dari tabel TbMahasiswa di mana jenis kelamin bukan 'Perempuan'
Hasil	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302001	ABDULLAH MUJADID AL QAWI	Islam	Laki-laki	2003-07-15	72	Medan	3000000
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302364	ALMAH WIJAYA ALHAKIM	Islam	Laki-laki	2002-11-07	90	Jakarta	4800000

22. Tampilkan semua data Tbmahasiswa yang diurutkan berdasarkan namamahasiswa berdasarkan descending

Syntax	SQL SELECT * FROM TbMahasiswa ORDER BY NamaMahasiswa DESC;
Penjelasan	Menampilkan semua data dari tabel TbMahasiswa diurutkan berdasarkan nama mahasiswa secara descending.
Hasil	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302009	BUNGA SABILA	Islam	Perempuan	2006-05-23	75	Medan	4400000
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302006	ANNISA TRINABILA	Islam	Perempuan	2004-04-22	72	Bogor	4700000
2302005	ANGGUN BUNGA LESTARI LUBIS	Islam	Perempuan	2002-11-07	80	Bogor	6200000
2302364	ALMAH WIJAYA ALHAKIM	Islam	Laki-laki	2002-11-07	90	Jakarta	4800000
2302002	ADINDA NABILA	Islam	Perempuan	2005-11-13	90	Medan	7500000
2302001	ABDULLAH MUJADID AL QAWI	Islam	Laki-laki	2003-07-15	72	Medan	3000000

23. Menampilkan rata-rata gaji_ortu

Syntax :	SELECT AVG(Gaji_Ortu) AS 'Rata-rata Gaji Ortu' -> FROM tbmahasiswa;
Penjelasan :	Tampilkan Rata-rata Gaji_Ortu dari Table TbMahasiswa
Hasil :	+----------+ | Rata-rata Gaji Ortu | +----------+ | 5385714.2857 | +----------+

24. Menampilkan semua data tbmahasiswa gaji_ortu diatas rata-rata

Syntax :	SELECT * FROM tbmahasiswa -> WHERE Gaji_Ortu > (SELECT AVG(Gaji_Ortu) FROM tbmahasiswa);
Penjelasan :	Tampilkan Semua data TbMahasiswaYang memiliki GAJI_ORTU Diatas Rata-Rata
Hasil :	+----------+----------+----------+----------+----------+----------+----------+----------+ | NimMahasiswa | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | UTS | Tempat_Tinggal | Gaji_Ortu | +----------+----------+----------+----------+----------+----------+----------+----------+ | 2302002 | ADINDA NABILA | Islam | Perempuan | 2005-11-13 | 90 | Medan | 7500000 | | 2302005 | ANGGUN BUNGA LESTARI LUBIS | Islam | Perempuan | 2002-11-07 | 80 | Bogor | 6200000 | | 2302008 | BAGAS TRY ATMAJA | Islam | Laki-laki | 2004-04-22 | 91 | Medan | 7100000 | +----------+----------+----------+----------+----------+----------+----------+----------+

25. Menampilkan semua data tbmahasiswa gaji_ortu dibawah rata-rata

Syntax :	SELECT * FROM tbmahasiswa -> WHERE Gaji_Ortu < (SELECT AVG(Gaji_Ortu) FROM tbmahasiswa);
Penjelasan :	Tampilkan Semua data TbMahasiswaYang memiliki GAJI_ORTU Dibawah Rata-Rata
Hasil :	+----------+----------+----------+----------+----------+----------+----------+----------+ | NimMahasiswa | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | UTS | Tempat_Tinggal | Gaji_Ortu | +----------+----------+----------+----------+----------+----------+----------+----------+ | 2302001 | ABDULLAH MUJADID AL QAWI | Islam | Laki-laki | 2003-07-15 | 72 | Medan | 3000000 | | 2302006 | ANNISA TRINABILA | Islam | Perempuan | 2004-04-22 | 72 | Bogor | 4700000 | | 2302009 | BUNGA SABILA | Islam | Perempuan | 2006-05-23 | 75 | Medan | 4400000 | | 2302364 | ALMAH WIJAYA ALHAKIM | Islam | Laki-laki | 2002-11-07 | 90 | Jakarta | 4800000 | +----------+----------+----------+----------+----------+----------+----------+----------+

26. Tampilkan Semua data TbMahasiswa Yang memiliki Nilai UTS Paling Tinggi

Syntax :	SELECT * FROM tbmahasiswa WHERE UTS = (SELECT MAX(UTS) FROM tbmahasiswa);
Penjelasan :	Tampilkan Semua data TbMahasiswa Yang memiliki Nilai UTS Paling Tinggi

Hasil :	MariaDB [UJIKompetensi2]> SELECT * FROM tbmahasiswa -> WHERE UTS = (SELECT MAX(UTS) FROM tbmahasiswa); +---------+----------+-------+----------+----------+----------+-----+----------+ | Nim | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | Tempat_Tinggal | UTS | Gaji_Ortu | +---------+----------+-------+----------+----------+----------+-----+----------+ | 2302008 | BAGAS TRY ATMAJA | ISLAM | LAKI-LAKI | 2004-04-22 | MEDAN | 91 | 7100000 | +---------+----------+-------+----------+----------+----------+-----+----------+ 1 row in set (0.002 sec)

27. Tampilkan Semua data TbMahasiswa Yang memiliki Nilai UTS Paling Rendah

Syntax :	SELECT * FROM tbmahasiswa WHERE UTS = (SELECT MIN(UTS) FROM tbmahasiswa);
Penjelasan :	Tampilkan Semua data TbMahasiswaYang memiliki GAJI_ORTU Dibawah Rata-Rata
Hasil :	MariaDB [UJIKompetensi2]> SELECT * FROM tbmahasiswa -> WHERE UTS = (SELECT MIN(UTS) FROM tbmahasiswa); +---------+----------+-------+----------+----------+----------+-----+----------+ | Nim | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | Tempat_Tinggal | UTS | Gaji_Ortu | +---------+----------+-------+----------+----------+----------+-----+----------+ | 2302001 | ABDULLAH MUJADID AL QAWI | ISLAM | LAKI-LAKI | 2003-07-15 | MEDAN | 72 | 3000000 | | 2302006 | ANNISA TRINABILA | ISLAM | PEREMPUAN | 2004-04-22 | BOGOR | 72 | 4700000 | +---------+----------+-------+----------+----------+----------+-----+----------+ 2 rows in set (0.003 sec)

28. Tampilkan data

Syntax :	SELECT 'Total Gaji Orang Tua Kelas 23S01 adalah Rp.' AS Keterangan, SUM(Gaji_Ortu) AS TotalGaji FROM tbmahasiswa WHERE NimMahasiswa LIKE '23%';
Penjelasan :	Tampilkan data gaji
Hasil :	+----------+ | Keterangan | TotalGaji | +----------+ | Total Gaji Orang Tua Kelas 23S01 adalah Rp. | 37700000 | +----------+

29. Tampilkan jumlah data

Syntax :	SELECT COUNT(*) AS 'Jumlah Data Jenis Kelamin Perempuan' FROM tbmahasiswa WHERE Jenis_Kelamin = 'Perempuan';
Penjelasan :	Tampilkan jumlah data yang berjenis kelamin perempuan
Hasil :	+----------+ | Jumlah Data Jenis Kelamin Perempuan | +----------+ | 4 | +----------+

30. Data semua laki-laki

Syntax :	SELECT * FROM tbmahasiswa WHERE Jenis_Kelamin = 'Laki-laki';
Penjelasan :	Tampilkan Semuan Data dari Tabel TbMahasiswa Yang memiliki JENIS_KELAMIN adalah LAKI-LAKI
Hasil :	

NimMahasiswa	NamaMahasiswa	Agama	Jenis_Kelamin	TglLahir	UTS	Tempat_Tinggal	Gaji_Ortu
2302001	ABDULLAH MUJADID AL QAWI	Islam	Laki-laki	2003-07-15	72	Medan	3000000
2302008	BAGAS TRY ATMAJA	Islam	Laki-laki	2004-04-22	91	Medan	7100000
2302364	ALMAH WIJAYA ALHAKIM	Islam	Laki-laki	2002-11-07	90	Jakarta	4800000

31. Rata-rata UTS per jenis kelamin

Syntax :	SELECT Jenis_Kelamin, AVG(UTS) AS 'Rata-rata Nilai UTS' FROM tbmahasiswa GROUP BY Jenis_Kelamin;
Penjelasan :	Tampikan Berdasarkan Group Data Nilai rata-Rata UTS berdasarkan Jenis_kelamin
Hasil :	

Jenis_Kelamin	Rata-rata Nilai UTS
Laki-laki	84.3333
Perempuan	79.2500

32. Rata-rata gaji ortu per tempat tinggal

Syntax :	SELECT Tempat_Tinggal, AVG(Gaji_Ortu) AS 'Rata-rata Gaji Orang Tua' FROM tbmahasiswa GROUP BY Tempat_Tinggal;
Penjelasan :	Tampikan Berdasarkan Group Data Rata-rata Gaji Orang Tua berdasarkan Tempat_Tinggal
Hasil :	

Tempat_Tinggal	Rata-rata Gaji Orang Tua
Bogor	5450000.0000
Jakarta	4800000.0000
Medan	5500000.0000

33. Rata-rata gaji ortu per tempat tinggal (HAVING ≥ 5.000.000)

Syntax :	SELECT Tempat_Tinggal, AVG(Gaji_Ortu) AS 'Rata-rata Gaji Orang Tua' FROM tbmahasiswa GROUP BY Tempat_Tinggal HAVING AVG(Gaji_Ortu) >= 5000000;
Penjelasan :	Tampikan Berdasarkan Group Data Rata-Rata Gaji Orang Tua berdasarkan Tempat_Tinggal, dimana Gaji Orang Tua >=500000
Hasil :	+----------+----------+ | Tempat_Tinggal | Rata-rata Gaji Orang Tua | +----------+----------+ | Bogor | 5450000.0000 | | Medan | 5500000.0000 | +----------+----------+

34. Total gaji ortu berdasarkan jenis kelamin

Syntax :	SELECT Jenis_Kelamin, SUM(Gaji_Ortu) AS 'Total Gaji Orang Tua' FROM tbmahasiswa GROUP BY Jenis_Kelamin;
Penjelasan :	Tampikan Berdasarkan Group Data Jumlah Total Gaji Orang Tua berdasarkan Jenis_Kelamin
Hasil :	+----------+----------+ | Jenis_Kelamin | Total Gaji Orang Tua | +----------+----------+ | Laki-laki | 14900000 | | Perempuan | 22800000 | +----------+----------+

35. Membuat View VMhsMedan

Syntax :	CREATE VIEW VMhsMedan AS SELECT * FROM tbmahasiswa WHERE Tempat_Tinggal = 'Medan';
Penjelasan :	Buat View dengan nama VMhsMedan dengan menampilkan Semua Data TbMahasiswa dengan Tempat Tinggal di Medan
Hasil :	MariaDB [UJIKompetensi2]> MariaDB [UJIKompetensi2]> CREATE VIEW VMhsMedan AS -> SELECT * FROM tbmahasiswa WHERE Tempat_Tinggal='MEDAN'; Query OK, 0 rows affected (0.045 sec)

36. Menampilkan daftar View di database

Syntax :	SHOW FULL TABLES WHERE Table_type = 'VIEW';
Penjelasan :	Buat Perintah Menampilkan View di Database

Hasil :	+----------+----------+ | Tables_in_ujikompetensi_2 | Table_type | +----------+----------+ | vmhsmedan | VIEW | +----------+----------+

37. Menampilkan isi View VMhsMedan

Syntax :	SELECT * FROM VMhsMedan;
Penjelasan :	Buat Perintah Menampilkan View VMhsMedan
Hasil :	+----------+----------+----------+----------+----------+----------+----------+----------+ | NimMahasiswa | NamaMahasiswa | Agama | Jenis_Kelamin | TglLahir | UTS | Tempat_Tinggal | Gaji_Ortu | +----------+----------+----------+----------+----------+----------+----------+----------+ | 2302001 | ABDULLAH MUJADID AL QAWI | Islam | Laki-laki | 2003-07-15 | 72 | Medan | 3000000 | | 2302002 | ADINDA NABILA | Islam | Perempuan | 2005-11-13 | 90 | Medan | 7500000 | | 2302008 | BAGAS TRY ATMAJA | Islam | Laki-laki | 2004-04-22 | 91 | Medan | 7100000 | | 2302009 | BUNGA SABILA | Islam | Perempuan | 2006-05-23 | 75 | Medan | 4400000 | +----------+----------+----------+----------+----------+----------+----------+----------+

BAB VII

JOIN

7.1. Definisi Join

Join dalam basis data (database) adalah sebuah operasi yang digunakan untuk menggabungkan baris-baris dari dua tabel atau lebih berdasarkan pada kolom-kolom yang terkait di antara mereka.

7.2. Tujuan Utama Join

Tujuan utama dari operasi join adalah untuk mengambil data yang terdistribusi di beberapa tabel dan menyajikannya sebagai satu set hasil yang koheren.

```
MariaDB [(none)]> create database JOINN;
Query OK, 1 row affected (0.001 sec)
MariaDB [(none)]> USE JOINN;
Database changed
MariaDB [JOINN]> CREATE TABLE CUSTOMERS (
    -> ID INT PRIMARY KEY,
    -> NAME VARCHAR(50),
    -> AGE INT,
    -> ADDRESS VARCHAR(50),
    -> SALARY DECIMAL(10,2));
Query OK, 0 rows affected (0.020 sec)

MariaDB [JOINN]> INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS,
SALARY) VALUES
    -> (1, 'Ramesh', 32, 'Ahmedabad', 2000.00),
    -> (2, 'Khilan', 25, 'Delhi', 1500.00),
    -> (3, 'Kaushik', 23, 'Kota', 2000.00),
    -> (4, 'Chaitali', 25, 'Mumbai', 6500.00),
    -> (5, 'Hardik', 27, 'Bhopal', 8500.00),
    -> (6, 'Komal', 22, 'MP', 4500.00),
    -> (7, 'Muffy', 24, 'Indore', 10000.00);
Query OK, 7 rows affected (0.018 sec)
Records: 7  Duplicates: 0  Warnings: 0

MariaDB [JOINN]> CREATE TABLE ORDERS (
    -> OID INT PRIMARY KEY,
    -> DATE DATETIME,
    -> CUSTOMERS_ID INT,
    -> AMOUNT DECIMAL(10,2),
    -> FOREIGN KEY (CUSTOMERS_ID) REFERENCES CUSTOMERS (ID));
Query OK, 0 rows affected (0.051 sec)

MariaDB [JOINN]> INSERT INTO ORDERS (OID, DATE, CUSTOMERS_ID, AMOUNT)
VALUES
    -> (102, '2009-10-08 00:00:00', 3, 3000.00),
    -> (100, '2009-10-08 00:00:00', 3, 1500.00),
```

```
    -> (101, '2009-11-20 00:00:00', 2, 1560.00),
    -> (103, '2008-05-20 00:00:00', 4, 2060.00);
```
Query OK, 4 rows affected (0.019 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [JOINN]> SELECT * FROM CUSTOMERS;

```
+----+----------+------+-----------+----------+
| ID | NAME     | AGE  | ADDRESS   | SALARY   |
+----+----------+------+-----------+----------+
|  1 | Ramesh   |   32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |   25 | Delhi     |  1500.00 |
|  3 | Kaushik  |   23 | Kota      |  2000.00 |
|  4 | Chaitali |   25 | Mumbai    |  6500.00 |
|  5 | Hardik   |   27 | Bhopal    |  8500.00 |
|  6 | Komal    |   22 | MP        |  4500.00 |
|  7 | Muffy    |   24 | Indore    | 10000.00 |
+----+----------+------+-----------+----------+
7 rows in set (0.001 sec)
```

MariaDB [JOINN]> SELECT * FROM ORDERS;

```
+-----+---------------------+--------------+---------+
| OID | DATE                | CUSTOMERS_ID | AMOUNT  |
+-----+---------------------+--------------+---------+
| 100 | 2009-10-08 00:00:00 |            3 | 1500.00 |
| 101 | 2009-11-20 00:00:00 |            2 | 1560.00 |
| 102 | 2009-10-08 00:00:00 |            3 | 3000.00 |
| 103 | 2008-05-20 00:00:00 |            4 | 2060.00 |
+-----+---------------------+--------------+---------+
4 rows in set (0.001 sec)
```

MariaDB [JOINN]> DESC CUSTOMERS;

```
+---------+---------------+------+-----+---------+-------+
| Field   | Type          | Null | Key | Default | Extra |
+---------+---------------+------+-----+---------+-------+
| ID      | int(11)       | NO   | PRI | NULL    |       |
| NAME    | varchar(50)   | YES  |     | NULL    |       |
| AGE     | int(11)       | YES  |     | NULL    |       |
| ADDRESS | varchar(50)   | YES  |     | NULL    |       |
| SALARY  | decimal(10,2) | YES  |     | NULL    |       |
+---------+---------------+------+-----+---------+-------+
5 rows in set (0.036 sec)
```

MariaDB [JOINN]> DESC ORDERS;

```
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| OID          | int(11)       | NO   | PRI | NULL    |       |
| DATE         | datetime      | YES  |     | NULL    |       |
| CUSTOMERS_ID | int(11)       | YES  | MUL | NULL    |       |
| AMOUNT       | decimal(10,2) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
4 rows in set (0.030 sec)
```

BAB VIII

STORED

PROCEDUR

8.1. Definisi Stored Procedure

Stored procedure adalah kumpulan perintah SQL yang telah disimpan di dalam database dan dapat dipanggil oleh aplikasi atau pengguna untuk menjalankan tugas-tugas tertentu. Mereka digunakan untuk menyederhanakan pengelolaan kode, meningkatkan keamanan, dan meningkatkan kinerja database dengan mengurangi lalu lintas jaringan. Stored Procedure hanya bisa digunakan untuk operasi-operasi kompleks seperti manipulasi data, validasi, dan logika bisnis.
Procedure dibagi 2:

a. Tidak menggunakan variabel
b. Dengan menggunakan variabel

Ciri khas procedure:

```
DELIMITER $$ ** // ##
CREATE PROCEDURE Nama_Prosedur()
BEGIN
Perintah SQL
END$$ ** // ##
DELIMITER  ;
```

Untuk menjalankan atau memanggil Stored Procedure sintaksnya adalah :

```
CALL Nama_Procedure();
```

Membuat dan Menjalankan Stored Procedure

Untuk membuat Stored Procedure tidak lepas dari perintah-perintah DML atau statement SQL. Sebelum membuat stored procedure, adapun langkah-langkah nya sebagai berikut:

1. Buat database terlebih dahulu
2. Buat tabel
3. Tampilkan tabel
4. Gunakan database tersebut
5. Masuk ke perintah stored procedure

Perintah

```
DELIMITER $$
CREATE PROCEDURE Nama_Prosedure()
BEGIN
SELECT nama_field FROM nama_tabel;
END$$
DELIMITER  ;
```

Untuk memanggil stored procedure sintaksnya adalah :

```
CALL Nama_Prosedur ();
```

Contoh

```
MariaDB [(none)]> create database dbprocedure;
MariaDB [(none)]> use dbprocedure;
MariaDB [dbprocedure]> create table barang(
->kd_barang int Primary key,
-> Nama_barang varchar (100),
```

```
-> qty int,
-> unit varchar (10),
-> harga varchar (10));

MariaDB [dbprocedure]> desc barang;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| kd_barang   | int(11)      | NO   | PRI | NULL    |       |
| Nama_barang | varchar(100) | YES  |     | NULL    |       |
| qty         | int(11)      | YES  |     | NULL    |       |
| unit        | varchar(10)  | YES  |     | NULL    |       |
| harga       | varchar(10)  | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+

MariaDB [dbprocedure]> insert into barang values
    -> (101,'Kertas HVS-A4 70 GRAM',5,'RIM',80000),
    -> (102,'Kertas HVS-A4 80 GRAM',5,'RIM',90000);
MariaDB [dbprocedure]> select*from barang;
+-----------+-----------------------+------+------+-------+
| kd_barang | Nama_barang           | qty  | unit | harga |
+-----------+-----------------------+------+------+-------+
|       101 | Kertas HVS-A4 70 GRAM |    5 | RIM  | 80000 |
|       102 | Kertas HVS-A4 80 GRAM |    5 | RIM  | 90000 |
+-----------+-----------------------+------+------+-------+
MariaDB [dbprocedure]> DELIMITER $$
MariaDB [dbprocedure]> create procedure SELECTbarang()
    -> begin
    -> select kd_barang,Nama_barang,harga from barang;
    -> END$$
MariaDB [dbprocedure]> delimiter ;
MariaDB [dbprocedure]> call selectbarang();
+-----------+-----------------------+-------+
| kd_barang | Nama_barang           | harga |
+-----------+-----------------------+-------+
|       101 | Kertas HVS-A4 70 GRAM | 80000 |
|       102 | Kertas HVS-A4 80 GRAM | 90000 |
+-----------+-----------------------+-------+
```

MENCARI BARANG 'RIM'

```
MariaDB [dbprocedure]> DELIMITER $$
MariaDB [dbprocedure]> create procedure caribarang(unitbarang varchar (10))
    -> BEGIN
    -> select*from barang
    ->  WHERE unit=unitbarang;
    -> END $$
MariaDB [dbprocedure]> DELIMITER ;
MariaDB [dbprocedure]> call caribarang('RIM');
+-----------+-----------------------+------+------+-------+
| kd_barang | Nama_barang           | qty  | unit | harga |
+-----------+-----------------------+------+------+-------+
|       101 | Kertas HVS-A4 70 GRAM |    5 | RIM  | 80000 |
|       102 | Kertas HVS-A4 80 GRAM |    5 | RIM  | 90000 |
+-----------+-----------------------+------+------+-------+
```

MENAMBAHKAN BARANG

```
MariaDB [dbprocedure]> DELIMITER $$
MariaDB [dbprocedure]> create procedure tambahbarang(
    -> Vkd_barang int,
    -> VNama_barang varchar(100),
    -> Vqty int,
    -> Vunit varchar(10),
    -> Vharga varchar(10))
    -> BEGIN
    -> INSERT INTO barang
    -> VALUES (kd_barang,Nama_barang,qty,unit,harga);
    -> END $$
MariaDB [dbprocedure]>  DELIMITER ;
MariaDB [dbprocedure]>  call tambahbarang (103,'Buku tulis folio 50
LBR',2,'BOX',500000);
MariaDB [dbprocedure]>  call tambahbarang (104,'Buku tulis folio 100
LBR',2,'BOX',900000);
MariaDB [dbprocedure]> call tambahbarang  (105,'SKRETCBOOK',1,'BOX',900000);
MariaDB [dbprocedure]> call tambahbarang  (205,'SKRETCBOOK',1,'BOX',900000);
MariaDB [dbprocedure]> select* from barang;
+-----------+-------------------------+------+------+--------+
| kd_barang | Nama_barang             | qty  | unit | harga  |
+-----------+-------------------------+------+------+--------+
|       101 | Kertas HVS-A4 70 GRAM   |    5 | RIM  | 80000  |
|       102 | Kertas HVS-A4 80 GRAM   |    5 | RIM  | 90000  |
|       103 | Buku tulis folio 50 LBR |    2 | BOX  | 500000 |
|       104 | Buku tulis folio 100 LBR |    2 | BOX  | 900000 |
|       105 | SKRETCBOOK              |    1 | BOX  | 900000 |
|       205 | SKRETCBOOK              |    1 | BOX  | 900000 |
+-----------+-------------------------+------+------+--------+
```

MENGHAPUS BARANG

```
MariaDB [dbprocedure]> DELIMITER $$
MariaDB [dbprocedure]> CREATE PROCEDURE hapusbarang(Vkd_barang INT)
    -> BEGIN
    -> DELETE FROM barang WHERE kd_barang = Vkd_barang;
    -> END$$
MariaDB [dbprocedure]>  DELIMITER ;
MariaDB [dbprocedure]>  CALL hapusbarang(205);
MariaDB [dbprocedure]>  CALL hapusbarang(102);
MariaDB [dbprocedure]> select *from barang;
+-----------+-------------------------+------+------+--------+
| kd_barang | Nama_barang             | qty  | unit | harga  |
+-----------+-------------------------+------+------+--------+
|       101 | Kertas HVS-A4 70 GRAM   |    5 | RIM  | 80000  |
|       103 | Buku tulis folio 50 LBR |    2 | BOX  | 500000 |
|       104 | Buku tulis folio 100 LBR |    2 | BOX  | 900000 |
|       105 | SKRETCBOOK              |    1 | BOX  | 900000 |
+-----------+-------------------------+------+------+--------+
```

MENCARI DATA BERDASARKAN NAMA

```
MariaDB [dbprocedure]> DELIMITER $$
MariaDB [dbprocedure]> CREATE PROCEDURE caribarangnama(VCariNama
VARCHAR(100))
    -> BEGIN
    -> SELECT * FROM barang WHERE Nama_barang LIKE CONCAT('%', VCariNama,
```

```
'%');
    -> END$$
MariaDB [dbprocedure]> DELIMITER ;
MariaDB [dbprocedure]> CALL caribarangnama('Kertas');
+-----------+-----------------------+------+------+-------+
| kd_barang | Nama_barang           | qty  | unit | harga |
+-----------+-----------------------+------+------+-------+
|       101 | Kertas HVS-A4 70 GRAM |    5 | RIM  | 80000 |
+-----------+-----------------------+------+------+-------+
```

MENGUBAH/MEMPERBARUI DATA

```
MariaDB [dbprocedure]> DELIMITER $$
MariaDB [dbprocedure]> CREATE PROCEDURE ubahbarang(
    -> Vkd_barang INT,
    -> VNama_barang VARCHAR(100),
    -> Vqty INT,
    -> Vunit VARCHAR(10),
    -> Vharga VARCHAR(10))
    -> BEGIN
    -> UPDATE barang
    -> SET Nama_barang = VNama_barang,
    ->     qty = Vqty,
    ->     unit = Vunit,
    ->     harga = Vharga
    -> WHERE kd_barang = Vkd_barang;
    -> END$$
MariaDB [dbprocedure]> DELIMITER ;
MariaDB [dbprocedure]> CALL ubahbarang(101,'Kertas HVS-A4 70 GRAM
(Baru)',10,'RIM','85000');
MariaDB [dbprocedure]> select *from barang;
+-----------+-------------------------------+------+------+--------+
| kd_barang | Nama_barang                   | qty  | unit | harga  |
+-----------+-------------------------------+------+------+--------+
|       101 | Kertas HVS-A4 70 GRAM (Baru)  |   10 | RIM  | 85000  |
|       103 | Buku tulis folio 50 LBR       |    2 | BOX  | 500000 |
|       104 | Buku tulis folio 100 LBR      |    2 | BOX  | 900000 |
|       105 | SKRETCBOOK                    |    1 | BOX  | 900000 |
+-----------+-------------------------------+------+------+--------+
```

BAB IX

TRIGGER

9.1. Definisi Trigger

Trigger adalah serangkain Tindakan yang dijalankan secara otomatis saat mengoperasiksan perubahan tertentu yang dilakukan pada tabel tertentu. Kejadian (event) yang dapat membangkitkan trigger umumnya berupa pernyataan INSERT , UPDATE, atau DELETE. Berdasarkan ruang lingkupnya, trigger dikklasifikasikan menjadi dua jenis : row trigger dan statement.

Membuat dan Menjalankan Trigger

CONTOH : BEFORE INSERT

1. Kita memiliki tabel dengan nama petugas

```
MariaDB [dbtrigger24s04]> create table petugas(
    -> kd_petugas int primary key,
    -> nama_petugas varchar(50),
    -> username varchar(50),
    -> password varchar(50),
    -> alamat varchar(50));
Query OK, 0 rows affected (7.66 sec)
```

2. Setelah` itu kita buat tabel Trigger dengan nama petugas_log

```
MariaDB [dbtrigger24s04]> create table petugas_log(
    -> no int auto_increment primary key,
    -> nama varchar(50),
    -> waktu timestamp,
    -> keterangan text);
Query OK, 0 rows affected (10.33 sec)
```

3. Kita tidak perlu input data ke dalam tabel, kita akan isi data sambil menjalankan Trigger

4. Buat sintaks Trigger

```
MariaDB [dbtrigger24s04]> DELIMITER $$
MariaDB [dbtrigger24s04]> create trigger before_tugas_insert
    -> before insert on petugas
    -> for each row
    -> begin
    -> insert into petugas_log values
    -> (0,NEW.username,NOW(),'PETUGAS BARU');
    -> END $$
Query OK, 0 rows affected (9.48 sec)
```

5. Lihat tabel petugas dan tabel petugas_log

```
MariaDB [dbtrigger24s04]> SELECT *FROM PETUGAS_LOG;
+----+--------------+---------------------+--------------+
| no | nama         | waktu               | keterangan   |
+----+--------------+---------------------+--------------+
|  1 | @umeenyzwaaz | 2025-10-09 08:35:57 | PETUGAS BARU |
+----+--------------+---------------------+--------------+
1 row in set (0.42 sec)
```

CONTOH : AFTER DELETE

1. Buat sintaks Trigger

```
MariaDB [dbtrigger24s04]> DELIMITER $$
MariaDB [dbtrigger24s04]> create trigger after_tugas_delete
    -> after delete on petugas
    -> for each row
    -> begin
    -> insert into petugas_log values
    -> (0,OLD.username,NOW(),'PENGHAPUSAN PETUGAS');
    -> END $$
Query OK, 0 rows affected (0.12 sec)
```

2. Lihat table petugas dan petugas_log

```
MariaDB [dbtrigger24s04]> select *from petugas_log;
+----+--------------+---------------------+---------------------+
| no | nama         | waktu               | keterangan          |
+----+--------------+---------------------+---------------------+
|  1 | @umeenyzwaaz | 2025-10-09 08:35:57 | PETUGAS BARU        |
|  2 | @umeenyzwaaz | 2025-10-09 08:51:21 | PENGHAPUSAN PETUGAS |
+----+--------------+---------------------+---------------------+
2 rows in set (0.00 sec)
```

CONTOH : ATER UPDATE

1. Buat sintaks Trigger

```
MariaDB [dbtrigger24s04]> DELIMITER $$
MariaDB [dbtrigger24s04]> CREATE TRIGGER after_tugas_update
    -> after update on petugas
    -> for each row
    -> begin
    -> DECLARE keterangan text;
    -> SET keterangan = CONCAT(OLD.username,'->',NEW.username);
    -> insert into petugas_log values
   (0,NEW.username,NOW(),keterangan);
    -> END $$
Query OK, 0 rows affected (5.82 sec)
MariaDB [dbtrigger24s04]> DELIMITER ;
MariaDB [dbtrigger24s04]> INSERT INTO petugas values
    -> (102,'SYIFA ANDINI PUTRI','@syifaandini','syifaa','JL.MATARAM');
Query OK, 1 row affected (0.04 sec)
```

2. Kita perbarui data menggunakan sintaks UPDATE

```
MariaDB [dbtrigger24s04]> UPDATE petugas set username ='@syifaandn'
       where kd_petugas ='102';
Query OK, 1 row affected (0.63 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

3. Lihat table petugas dan petugas_log

```
MariaDB [dbtrigger24s04]> select *from petugas;
+------------+--------------------+------------+----------+------------+
| kd_petugas | nama_petugas       | username   | password | alamat     |
+------------+--------------------+------------+----------+------------+
|        102 | SYIFA ANDINI PUTRI | @syifaandn | syifaa   | JL.MATARAM |
+------------+--------------------+------------+----------+------------+
1 row in set (0.00 sec)
```

```
MariaDB [dbtrigger24s04]> select *from petugas_log;
+----+--------------+---------------------+--------------------------+
| no | nama         | waktu               | keterangan               |
+----+--------------+---------------------+--------------------------+
|  1 | @umeenyzwaaz | 2025-10-09 08:35:57 | PETUGAS BARU             |
|  2 | @umeenyzwaaz | 2025-10-09 08:51:21 | PENGHAPUSAN PETUGAS      |
|  3 | @syifaandini | 2025-10-09 08:59:44 | PETUGAS BARU             |
|  4 | @syifaandn   | 2025-10-09 09:01:10 | @syifaandini->@syifaandn |
+----+--------------+---------------------+--------------------------+
4 rows in set (0.00 sec)
```

CONTOH : AFTER INSERT

1. Buat sintaks Trigger

```
MariaDB [dbtrigger24s04]> DELIMITER $$
MariaDB [dbtrigger24s04]> create trigger after_tugas1_insert
    -> after insert on petugas
    -> for each row
    -> begin
    -> insert into petugas_log values
    -> (0,NEW.username,NOW(),'PETUGAS BARU');
    -> END $$
Query OK, 0 rows affected (1.76 sec)
MariaDB [dbtrigger24s04]> DELIMITER ;
MariaDB [dbtrigger24s04]> INSERT INTO PETUGAS VALUES
    -> (103,'ROSA FITRIA','@rosafitria','rosaa','JL.KARTINI');
Query OK, 1 row affected (1.88 sec)
```

2. Lihat table petugas dan table petugas_log

```
MariaDB [dbtrigger24s04]> SELECT *FROM PETUGAS;
+------------+--------------------+-------------+----------+------------+
| kd_petugas | nama_petugas       | username    | password | alamat     |
+------------+--------------------+-------------+----------+------------+
|        102 | SYIFA ANDINI PUTRI | @syifaandn  | syifaa   | JL.MATARAM |
|        103 | ROSA FITRIA        | @rosafitria | rosaa    | JL.KARTINI |
+------------+--------------------+-------------+----------+------------+
2 rows in set (0.00 sec)
```

```
MariaDB [dbtrigger24s04]> SELECT *FROM PETUGAS_LOG;
+----+--------------+---------------------+--------------------------+
| no | nama         | waktu               | keterangan               |
+----+--------------+---------------------+--------------------------+
|  1 | @umeenyzwaaz | 2025-10-09 08:35:57 | PETUGAS BARU             |
|  2 | @umeenyzwaaz | 2025-10-09 08:51:21 | PENGHAPUSAN PETUGAS      |
|  3 | @syifaandini | 2025-10-09 08:59:44 | PETUGAS BARU             |
|  4 | @syifaandn   | 2025-10-09 09:01:10 | @syifaandini->@syifaandn |
|  5 | @rosafitria  | 2025-10-09 09:09:59 | PETUGAS BARU             |
|  6 | @rosafitria  | 2025-10-09 09:09:59 | PETUGAS BARU             |
+----+--------------+---------------------+--------------------------+
6 rows in set (0.00 sec)
```

CONTOH : BEFORE DELETE

1. Buat sintaks Trigger

```
MariaDB [dbtrigger24s04]> DELIMITER $$
MariaDB [dbtrigger24s04]> CREATE TRIGGER before_tugas_delete
    -> before delete on petugas
    -> for each row
    -> begin
    -> insert into petugas_log values
    -> (0,OLD.username,NOW(),'PENGHAPUSAN PETUGAS');
    -> END $$
Query OK, 0 rows affected (0.10 sec)

MariaDB [dbtrigger24s04]> DELIMITER ;
MariaDB [dbtrigger24s04]> DELETE FROM PETUGAS WHERE kd_petugas ='102';
Query OK, 1 row affected (0.06 sec)
```

2. Lihat table petugas dan petugas_log

```
MariaDB [dbtrigger24s04]> select *from petugas;
+------------+--------------+-------------+----------+------------+
| kd_petugas | nama_petugas | username    | password | alamat     |
+------------+--------------+-------------+----------+------------+
|        103 | ROSA FITRIA  | @rosafitria | rosaa    | JL.KARTINI |
+------------+--------------+-------------+----------+------------+
1 row in set (0.00 sec)
```

```
MariaDB [dbtrigger24s04]> select *from petugas_log;
+----+--------------+---------------------+--------------------------+
| no | nama         | waktu               | keterangan               |
+----+--------------+---------------------+--------------------------+
|  1 | @umeenyzwaaz | 2025-10-09 08:35:57 | PETUGAS BARU             |
|  2 | @umeenyzwaaz | 2025-10-09 08:51:21 | PENGHAPUSAN PETUGAS      |
|  3 | @syifaandini | 2025-10-09 08:59:44 | PETUGAS BARU             |
|  4 | @syifaandn   | 2025-10-09 09:01:10 | @syifaandini->@syifaandn |
|  5 | @rosafitria  | 2025-10-09 09:09:59 | PETUGAS BARU             |
|  6 | @rosafitria  | 2025-10-09 09:09:59 | PETUGAS BARU             |
|  7 | @syifaandn   | 2025-10-09 09:15:48 | PENGHAPUSAN PETUGAS      |
|  8 | @syifaandn   | 2025-10-09 09:15:48 | PENGHAPUSAN PETUGAS      |
+----+--------------+---------------------+--------------------------+
8 rows in set (0.00 sec)
```

CONTOH : BEFORE UPDATE

1. Buat sintaks Trigger

```
MariaDB [dbtrigger24s04]> DELIMITER $$
MariaDB [dbtrigger24s04]> CREATE TRIGGER before_tugas_update
    -> before update on petugas
    -> for each row
    -> begin
    -> DECLARE keterangan text;
    -> set keterangan =CONCAT(OLD.username,'->',NEW.username);
    -> insert into petugas_log values (0,NEW.username,NOW(),keterangan);
    -> END $$
Query OK, 0 rows affected (0.10 sec)

MariaDB [dbtrigger24s04]> DELIMITER ;
MariaDB [dbtrigger24s04]> INSERT INTO petugas values
    -> (105,'MURI SINAGA','@murisinaga01','muri00','JL.LANGKAT');
Query OK, 1 row affected (0.04 sec)
```

2. Kita perbarui data dengan menggunakan sintaks UPDATE

```
MariaDB [dbtrigger24s04]> UPDATE petugas SET username = '@murisinaga'
where kd_petugas ='105';
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

3. Menampilkan table petugas dan petugas_log

```
MariaDB [dbtrigger24s04]> select *from petugas;
+------------+--------------+-------------+----------+------------+
| kd_petugas | nama_petugas | username    | password | alamat     |
+------------+--------------+-------------+----------+------------+
|        103 | ROSA FITRIA  | @rosafitria | rosaa    | JL.KARTINI |
|        105 | MURI SINAGA  | @murisinaga | muri00   | JL.LANGKAT |
+------------+--------------+-------------+----------+------------+
2 rows in set (0.00 sec)

MariaDB [dbtrigger24s04]> select *from petugas_log;
+----+---------------+---------------------+---------------------------+
| no | nama          | waktu               | keterangan                |
+----+---------------+---------------------+---------------------------+
|  1 | @umeenyzwaaz  | 2025-10-09 08:35:57 | PETUGAS BARU              |
|  2 | @umeenyzwaaz  | 2025-10-09 08:51:21 | PENGHAPUSAN PETUGAS       |
|  3 | @syifaandini  | 2025-10-09 08:59:44 | PETUGAS BARU              |
|  4 | @syifaandn    | 2025-10-09 09:01:10 | @syifaandini->@syifaandn  |
|  5 | @rosafitria   | 2025-10-09 09:09:59 | PETUGAS BARU              |
|  6 | @rosafitria   | 2025-10-09 09:09:59 | PETUGAS BARU              |
|  7 | @syifaandn    | 2025-10-09 09:15:48 | PENGHAPUSAN PETUGAS       |
|  8 | @syifaandn    | 2025-10-09 09:15:48 | PENGHAPUSAN PETUGAS       |
|  9 | @murisinaga01 | 2025-10-09 09:22:20 | PETUGAS BARU              |
| 10 | @murisinaga01 | 2025-10-09 09:22:20 | PETUGAS BARU              |
| 11 | @murisinaga   | 2025-10-09 09:24:31 | @murisinaga01->@murisinaga |
| 12 | @murisinaga   | 2025-10-09 09:24:31 | @murisinaga01->@murisinaga |
+----+---------------+---------------------+---------------------------+
12 rows in set (0.00 sec)
```

BAB X

PENUTUP

10.1. Kesimpulan

Dari pembahasan dalam makalah ini dapat disimpulkan bahwa Structured Query Language (SQL) merupakan bahasa standar yang digunakan untuk berkomunikasi dengan database relasional. SQL terbagi menjadi Tiga komponen utama, yaitu: Data Definition Language (DDL), yang berfungsi untuk mendefinisikan, membuat, mengubah, dan menghapus struktur database maupun tabel, Data Manipulation Language (DML), yang berfungsi untuk memanipulasi data di dalam tabel, seperti menambahkan, mengambil, memperbarui, dan menghapus data dan Data Query Language (DQL) yang berfungsi untuk mengambil data di dalam tabel maupun kolom.

Selain itu, SQL juga menyediakan berbagai constraint seperti primary key, foreign key, unique, not null, check, dan default yang berguna untuk menjaga integritas serta konsistensi data dalam database. Dengan demikian, SQL tidak hanya penting sebagai alat teknis dalam pengelolaan data, tetapi juga berperan besar dalam mendukung sistem informasi agar berjalan dengan baik, terstruktur, dan efisien.

10.2. Saran

1. Mahasiswa atau pembaca yang baru mempelajari SQL sebaiknya menguasai terlebih dahulu konsep dasar DDL dan DML sebelum masuk ke perintah yang lebih kompleks.
2. Pembelajaran SQL akan lebih efektif jika dibarengi dengan praktik langsung menggunakan DBMS seperti MySQL, PostgreSQL, atau SQL Server.
3. Gunakan constraint secara optimal agar data tetap terjaga keakuratannya dan terhindar dari inkonsistensi.
4. Setelah memahami dasar SQL, disarankan untuk mempelajari konsep lanjutan seperti stored procedure, trigger, view, serta optimasi query agar keterampilan semakin meningkat.
5. Ilmu SQL sebaiknya diterapkan pada proyek atau studi kasus nyata sehingga pemahaman tidak hanya bersifat teori, tetapi juga aplikatif.

10.3. Dokumentasi

DAFTAR PUSTAKA

[1][1] J. Adflin, “Bahasa Query,” *Univ. Palangka Raya*, vol. 21, no. 1, pp. 1–9, 2020, [Online]. Available: http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203%0Ahttp://mpoc.org.my/malaysian-palm-oil-industry/

[2] M. Amin, “Bahasa Query Menggunakan MySQL,” 2022.

[3] Anggoro Dimas Aryo, Supriyanti Wiwit, and Putri Devri Afriyantari Puspa, “Konsep Dasar Sistem Basis Data Dengan Mysql,” 2021.

[4] Canggih Ajika Pamungkas, “Pengantar dan Implementasi Basis Data - Google Books,” 2017. [Online]. Available: https://www.google.co.id/books/edition/Pengantar_dan_Implementasi_Basis_Data/hKdADwAAQBAJ?hl=id&gbpv=1&dq=Basis+data+pamungkas&printsec=frontcover%0Ahttps://www.google.co.id/books/edition/Pengantar_dan_Implementasi_Basis_Data/hKdADwAAQBAJ?hl=id&gbpv=1&dq=

[5] S. P. Collins *et al.*, “No Title 済無No Title No Title No Title,” pp. 167–186, 2021.

[6] Jamaluddin *et al.*, *BUKU (Book Chapter)-Sistem Basis Data (Elmi Devia)_oke*. 2022. [Online]. Available: https://repository.unkris.ac.id/id/eprint/518/1/BUKU (Book Chapter)-Sistem Basis Data (Elmi Devia)_oke.pdf

[7] I. D. Kurniati *et al.*, *Buku Ajar*. 2015.