



LAPORAN PRAKTIKUM BASIS DATA II

Prodi : Sistem Informasi



Disusun Oleh:

› Arif Silaban
› Rafael Sinaga
› Ondo Rajagukguk
› Diaz Al Latif
› Febi Sofia

Ayuhana Vannesya <
Gadis Fitria <
Yosi Manurung <
Azura Intan <
Dea Cindiasyahwa <

-- 24S04 --

**STIKOM TUNAS BANGSA
TAHUN 2025**

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas limpahan rahmat dan karunia-Nya, buku praktikum ini dapat kami selesaikan dengan baik. Laporan praktikum ini disusun dengan tujuan untuk memberikan pemahaman yang lebih mendalam tentang materi-materi penting dalam Basis Data II.

Kami menyadari bahwa materi Basis Data II merupakan salah satu pilar utama dalam pengelolaan informasi yang sangat penting di era digital saat ini. Oleh karena itu, laporan praktikum ini dirancang agar dapat menjadi panduan praktis bagi mahasiswa, dosen, dan pembaca lainnya yang ingin mengembangkan kompetensinya dalam memahami konsep-konsep tersebut secara komprehensif. Dengan penjelasan yang terstruktur, diharapkan pembaca dapat lebih mudah mengikuti setiap langkah dalam praktikum dan mampu mengimplementasikan pengetahuan yang diperoleh ke dalam dunia nyata.

Kami menyadari bahwa dalam proses penyusunan laporan praktikum ini masih terdapat kekurangan dan keterbatasan, baik dari segi materi maupun penyajiannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang konstruktif dari para pembaca. Masukan tersebut akan menjadi bekal berharga bagi kami untuk terus memperbaiki dan menyempurnakan laporan praktikum ini dimasa mendatang.

Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan laporan praktikum ini, baik secara langsung maupun tidak langsung. Semoga laporan praktikum ini dapat memberikan manfaat yang sebesar-besarnya bagi pembaca, khususnya meningkatkan wawasan dan keterampilan dalam bidang Basis Data.

Pematangsiantar, September 2025

Kelompok Penyusun

DAFTAR ISI

Kata Pengantar	i
Daftar Isi.....	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum	1
1.3 Pemahaman Dasar Basis Data	1
BAB II XAMPP CONTROL PANEL	3
(Praktikum 1).....	3
1. Memulai Xampp Control Panel.....	3
2. Membuat Database	4
3. Membuat Tabel	5
4. Mengisi Tabel	5
BAB III Data Definition Language (DDL)	6
1. CREATE	6
2. ALTER.....	6
3. DROP.....	7
4. RENAME	7
(Praktikum 2).....	8
Sub CMK (1). DDL Database : CREATE	9
Sub CMK (2). DDL Database : ALTER	11
Gambar ERD	12
Sub CMK (3). DDL Database : Create :: Alter :: Drop	12
BAB IV Data Manipulation Language (DML)	16
1. INSERT	16
2. UPDATE.....	17
3. DELETE	17
4. SELECT.....	17
Sub CMK (4). DML Database: INSERT	18
Sub CMK (5). DML Database: UPDATE	22
Sub CMK (6). DML Database: DELETE	24
(Praktikum 3).....	25

Sub CMK (1) . Review DDL dan DML : Create, ALTER, INSERT	26
Sub CMK (2) . DQL Database : Select Statement Fungsi String	27
Sub CMK (3) . Select With Filter Where, Between, Like, In, AND, OR, ORDER BY	31
Sub CMK(4) . DML Database : AGGREGATE, Group By	34
Sub CMK (5). VIEW	37
BAB V STORED PROCEDURE	39
A. Membuat dan Menjalankan Stored Procedure (Praktikum 5)	39
BAB VI TRIGGER	46
A. Membuat dan Menjalankan Trigger (Praktikum 6)	47
DAFTAR PUSTAKA	51
DOKUMENTASI	52

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring perkembangan teknologi informasi, basis data menjadi komponen krusial dalam berbagai sistem informasi dan aplikasi. Kemampuan mengelola dan memanipulasi data secara efisien sangat dibutuhkan, khususnya bagi mahasiswa jurusan Sistem Informasi. Dalam pembelajaran dasar, biasanya hanya diperkenalkan konsep dasar basis data, biasanya hanya diperkenalkan konsep dasar basis data, tetapi pada tingkat lanjutan, perlu dipelajari topik-topik seperti DDL dan DML agar mahasiswa mampu menangani data dengan lebih kompleks dan efisien. Praktikum ini dirancang untuk membekali mahasiswa dan keterampilan yang lebih mendalam dalam menggunakan SQL dan MariaDB untuk membangun sistem yang berdaya guna dan dapat diandalkan.

1.2 Tujuan Praktikum

Tujuan dari praktikum ini adalah:

1. Membantu mahasiswa memahami konsep lanjutan basis data yang sering digunakan dalam pengembangan aplikasi nyata.
2. Memberikan wawasan bagaimana cara memulai Xampp Control Panel.
3. Memberikan pengalaman praktis dalam menerapkan operasi basis data yang melibatkan banyak table, seperti *ddl* dan *dml*.
4. Mengajarkan cara membuat dan menggunakan prosedur tersimpan, *ddl*, serta *dml* untuk meningkatkan efisiensi pengolahan data.
5. Mengembangkan kemampuan mahasiswa dalam merancang basis data yang efisien dan aman melalui praktik langsung.

1.3 Pemahaman dasar Basis Data

Sebelum masuk ke pembahasan umum, seharusnya kita sudah tau apa itu SQL. SQL adalah bahasa pemrograman yang efektif untuk mengelola database. Fungsinya meliputi eksekusi query, pengaturan hak akses pengguna, serta manipulasi dan

pengaksesan database. Penggunaan SQL mempermudah pekerjaan dengan basis data secara efisien dan terstruktur yang meliputi; membuat, mengambil, menambah, merubah, menghapus.

BAB II

XAMPP CONTROL PANEL

XAMPP Control Panel adalah antarmuka utama yang digunakan untuk mengelola layanan (server) yang ada di paket XAMPP. XAMPP sendiri adalah paket software yang berisi beberapa aplikasi server penting, seperti:

Apache → web server untuk menjalankan aplikasi berbasis web.

MySQL/MariaDB → database server untuk menyimpan data.

PHP → bahasa pemrograman server-side.

Perl → bahasa pemrograman tambahan. (Gunawan et al., 2023)

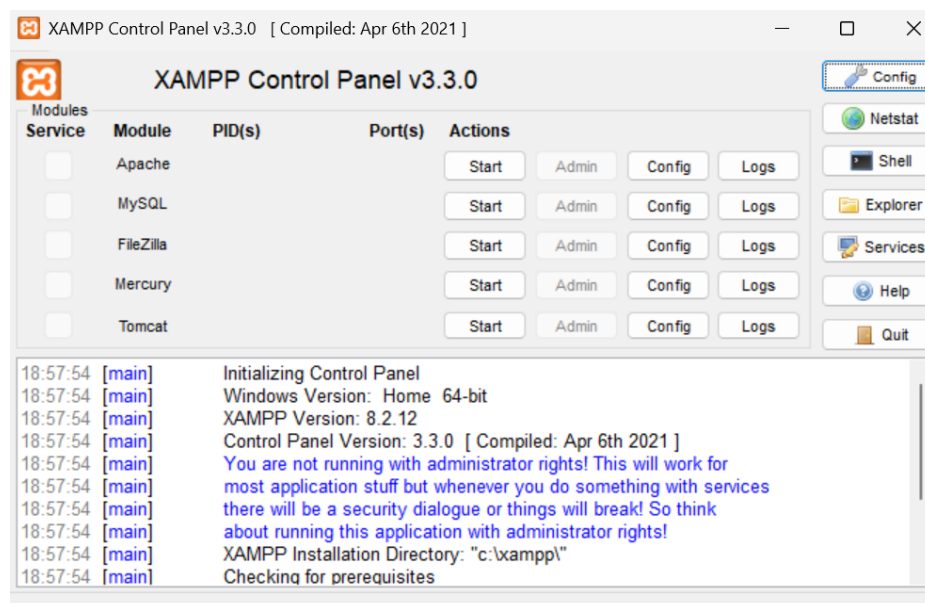
Control Panel berfungsi sebagai pusat kendali untuk:

- Menyalakan / mematikan layanan (Apache, MySQL, FileZilla, Mercury, Tomcat).
- Melihat status layanan apakah sedang berjalan atau berhenti.
- Mengakses konfigurasi setiap layanan dengan mudah.
- Melihat log/error jika ada masalah saat menjalankan server.

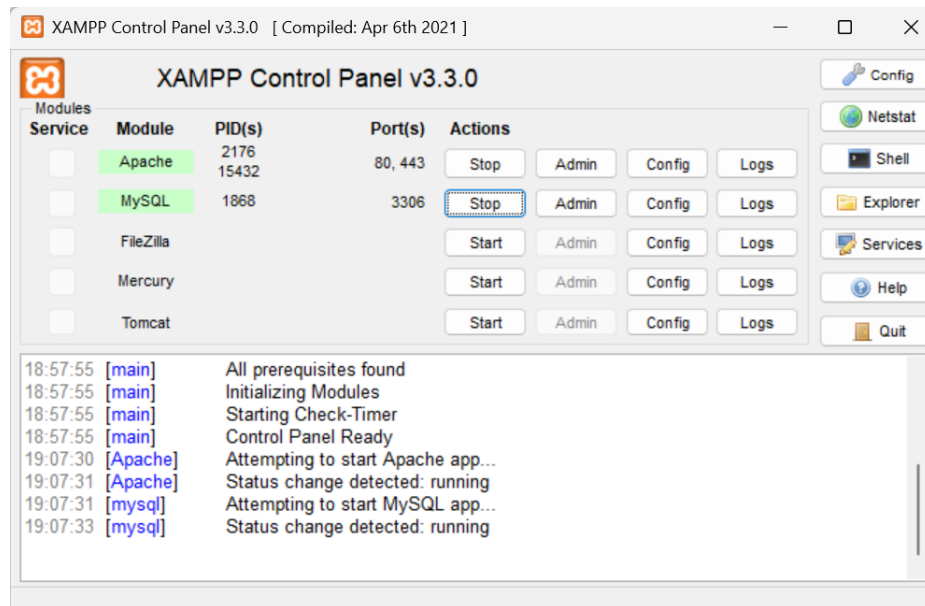
(Ismail, 2021)

1. Memulai XAMPP Control Panel (Praktikum 1)

Contoh tampilannya biasanya ada tombol Start / Stop untuk masing-masing layanan, juga tombol Config, Logs, dan Admin.



Apabila ingin memulai Xampp Control Panel, klik Start pada Apache dan MySQL



Apabila module pada Apache dan MySQL sudah hijau, klik Shell untuk memulai XAMPP

Tampilan Awal:

```

Setting environment for using XAMPP for Windows.
MSI MODERN 14@MSI c:\xampp
#
  
```

Ketik `mysql -u root` untuk masuk ke MariaDB

```

Setting environment for using XAMPP for Windows.
MSI MODERN 14@MSI c:\xampp
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
  
```

2. Membuat Database

Ketik Perintah: `CREATE DATABASE [nama_database];`

Contoh: `CREATE DATABASE cicilan_kuliah;`

```

MariaDB [(none)]> create database CicilanKuliah;
Query OK, 1 row affected (0.008 sec)
  
```

Jangan lupa untuk masuk ke dalam database nya dengan perintah use [nama_db];

Tampilan:

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

3. Membuat Tabel

Setelah membuat database kita bisa membuat tabel seperti berikut:

Ketik Perintah: CREATE TABLE [nama_tabel]([kolom 1]
tipe_date(size));

Contoh: CREATE TABLE tbprodi(Varchar(50));

Tampilan:

```
MariaDB [CicilanKuliah]> create table tbprodi (kdprodi INT(11) PRIMARY KEY, Namaprodi Varchar(50) NOT NULL, Prog_Pendidikan Varchar(10) NULL);  
Query OK, 0 rows affected (0.025 sec)
```

4. Mengisi Data

Apabila sudah membuat tabel kita dapat mengisi tabel tersebut.

Ketik Perintah:

Insert Into tbprodi ([Kolom1], [Kolom2], [kolom n]...) VALUES
('[Record1]', '[Record2]',));

Contoh:

```
MariaDB [cicilankuliah]> 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 (0.006 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```

BAB III

DATA DEFINITION LANGUAGE (DDL)

DDL (Data Definition Language) adalah bagian dari bahasa SQL (Structured Query Language) yang digunakan untuk mendefinisikan struktur dan skema basis data. DDL fokus pada operasi-operasi yang memengaruhi struktur database, seperti membuat tabel, mengubah struktur tabel, atau menghapus objek database. Beberapa perintah DDL yang umum digunakan melibatkan operasi pembuatan, perubahan, dan penghapusan objek dalam database. (Kurniati et al., 2015)

DDL bertujuan untuk membangun dan memodifikasi struktur objek yang disimpan dalam database. Definisi dan modifikasi ini mengontrol deskripsi skema basis data. Mereka berbeda dari perintah Data Manipulation Language (DML) yang digunakan untuk tujuan modifikasi data karena perintah DDL digunakan untuk mengubah struktur database. Contohnya adalah membuat tabel atau objek baru bersama dengan semua atributnya seperti tipe data, nama tabel, dll. (Silalahi, 2022)

Perintah DDL biasanya digunakan saat awal pembuatan database, dan jarang digunakan oleh pengguna umum. Perintah DDL dimaksudkan untuk memanipulasi dan memodifikasi objek, seperti pengguna, basis data, skema, tabel, tampilan, kolom, fungsi, dan membuat prosedur tersimpan. (Budayawan et al., 2023)

1. CREATE

Create bertujuan untuk membuat objek dalam database, seperti tabel, indeks, atau prosedur tersimpan

Bentuk Umum:

- CREATE DATABASE [nama_database]
- CREATE TABLE [nama_tabel] ([kolom] tipe_data(size));

Contoh syntax:

- CREATE DATABASE cicilan_kuliah;
- CREATE TABLE tbprodi>NamaMahasiswa Varchar(50));

2. ALTER

Alter bertujuan untuk mengubah struktur objek yang sudah ada, seperti menambahkan kolom atau mengubah tipe data kolom.

Penggunaan alter:

1. Alter Table Add Column → Menambahkan kolom baru pada tabel.
2. Alter Table Add Foreign Key → Menambahkan Foreign key sebagai penghubung Primary Key dengan syarat tipe dan ukuran data yang sama.
3. Alter Table Add Primary Key → Menambahkan Primary key pada kolom.
4. Alter Table Add.....First → Menambahkan kolom baru pada tabel yang posisinya berada di paling atas.
5. Alter Table Add.....After → Menambahkan kolom baru pada tabel yang posisinya berada setelah kolom yang ada.
6. Alter Table Add.....Before → Menambahkan kolom baru pada tabel yang posisinya berada sebelum di bawah kolom yang ada.
7. Alter Table Modify → Mengubah size dari tipe data pada tabel tertentu.
8. Alter Table Rename → Mengubah nama tabel tertentu.
9. Alter Table Change → Mengubah kolom tertentu menjadi kolom yang berbeda.
10. Alter Table Drop → Menghapus kolom tertentu di dalam tabel.(Samsoni et al., 2021)

3. DROP

Rename bertujuan menghapus objek dari database, seperti tabel, indeks, atau basis data

Bentuk Umum:

- Drop database [nama_database];
- Drop table [nama_tabel];

Contoh Syntax:

- Drop database cicilan_kuliah;
- Drop table tbprodi;

4. RENAME

Rename bertujuan mengubah nama suatu objek dalam dataset, seperti tabel.

Bentuk Umum:

- Rename table [nama_tabel_lama] to [nama_tabel_baru];

Contoh Syntax:

- Rename table login to managementuser; (Canggih Ajika Pamungkas, 2017)

PERTEMUAN KE 2

Dosen	:	Dr. Dedy Hartama, M.Kom
Topik	:	DDL DATABASE DAN TABEL & DML
Pertemuan	:	Ke 2
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.
Ketua Kelompok dan Tugas	:	Nama Ketua : Arif Hasudungan Silaban Tugas : Membimbing Anggota sekaligus Pengerjaan Laporan
Kelas	:	24S04
Semester	:	3 (GANJIL)
Tgl Pertemuan	:	11 September 2025
Tanggal Selesai dan Revisi	:	Selesai pada 14 September 2025 dan direvisi I pada 18 September 2025
Tanggal Acc		18 September 2025 (Revisi I)
Nilai		Nilai Kelompok : Rata-Rata Nilai Anggota 1. 2. 3. 4. 5. 6. 7. 8. 9. 10.
Nama Anggota dan Tugas	:	Anggota 1. Arif Hasudungan Silaban (CMK 3 + 5) 2. Dea Cindiasyahwa (CMK 1 + 4) 3. Rafael Sinaga (CMK 2 + 4) 4. Diaz Al Latif (CMK 2 + 5) 5. Ondo Rajagukguk (CMK 2 + 5)

		6. Febi Sofia Azzura (CMK 3 + 5) 7. Gadis Fitria (CMK 3 + 5) 8. Ayuhana Vannesya (CMK 2 + 4) 9. Yosi Anggraini (CMK 2 + 4) 10. Azura Intan (CMK 1 + 6)
--	--	--

HASIL DAN PEMBAHASAN (PRAKTIKUM 2)

➤ SUB CMK (1). DDL DATABASE: **CREATE**

1. Membuat database dengan nama: CicilanKuliah

Syntax :	Create database cicilankuliah;
Penjelasan :	Membuat database dengan nama cicilankuliah
Hasil	<pre>MariaDB [(none)]> create database CicilanKuliah; Query OK, 1 row affected (0.008 sec)</pre>

2. Tampilkan Database

Syntax :	show databases;
Penjelasan :	Menampilkan database cicilankuliah
Hasil	<pre>MariaDB [cicilankuliah]> show databases; +-----+ Database +-----+ cicilankuliah </pre>

3. Masuk ke dalam Database

Syntax :	use CicilanKuliah;
Penjelasan :	Masuk ke dalam database untuk digunakan
Hasil	<pre>MariaDB [(none)]> use CicilanKuliah; Database changed</pre>

4. Buat table tbprodi, tbsesi, tbkelas, tbkasir, tbjenisbayar, tbmahasiswa, transaksi_cicilan

Syntax :	Create table tbprodi (kdprodi int(11) PRIMARY KEY, Namaprodi Varchar(50) NOT NULL, Prog_Pendidikan varchar(10) NULL);
Penjelasan :	Membuat table tbprodi

Hasil	<pre>MariaDB [CicilanKuliah]> create table tbprodi (kdprodi INT(11) PRIMARY KEY, Namaprodi Varchar(50) NOT NULL, Prog_Pendidikan Varchar(10) NULL); Query OK, 0 rows affected (0.025 sec)</pre>
Syntax :	<pre>Create table tbsesi (kdsesi varchar(5) PRIMARY KEY, Namasesi Varchar(7) NOT NULL);</pre>
Penjelasan :	Membuat table tbsesi
Hasil	<pre>MariaDB [CicilanKuliah]> create table tbsesi (kdsesi varchar(5) PRIMARY KEY, Namasesi Varchar(7) NOT NULL); Query OK, 0 rows affected (0.032 sec)</pre>
Syntax :	<pre>Create table tbkelas (kdkelas varchar(5) PRIMARY KEY, Namakelas Varchar(6) NOT NULL);</pre>
Penjelasan :	Membuat table tbkelas
Hasil	<pre>MariaDB [CicilanKuliah]> create table tbkelas (kdkelas varchar(5) PRIMARY KEY, Namakelas varchar(6) NOT NULL); Query OK, 0 rows affected (0.028 sec)</pre>
Syntax :	<pre>Create table tbkasir (kdkasir varchar(5) PRIMARY KEY, Namakasir Varchar(30) NOT NULL);</pre>
Penjelasan :	Membuat table tbkasir
Hasil	<pre>MariaDB [CicilanKuliah]> create table tbkelas (kdkelas varchar(5) PRIMARY KEY, Namakelas varchar(6) NOT NULL); Query OK, 0 rows affected (0.028 sec)</pre>
Syntax :	<pre>Create table tbjenisbayar (kdjenis varchar(5) PRIMARY KEY, jenisbayar enum('pendaftaran','uang kuliah','perlengkapan','ICT') NULL, TahunAjaran Varchar(10) NOT NULL);</pre>
Penjelasan :	Membuat table tbjenisbayar
Hasil	<pre>MariaDB [CicilanKuliah]> create table tbjenisbayar (kdjenis varchar(5) PRIMARY KEY, jenisbayar enum('pendaftaran','uang kuliah','Perlengkapan','ICT'), TahunAjaran Varchar(10) NOT NULL); Query OK, 0 rows affected (0.028 sec)</pre>
Syntax :	<pre>Create table tbmahasiswa (Nim Varchar(10) PRIMARY KEY, NamaMahasiswa Varchar(30), TempatLahir Varchar(30) NOT NULL, TglLahir DATE NOT NULL);</pre>
Penjelasan :	Membuat table tbmahasiswa
Hasil	<pre>MariaDB [CicilanKuliah]> create table tbmahasiswa (Nim Varchar(10) PRIMARY KEY, NamaMahasiswa Varchar(30) NOT NULL, TempatLahir Varchar(30) NOT NULL, TglLahir DATE NOT NULL); Query OK, 0 rows affected (0.028 sec)</pre>
Syntax :	<pre>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, UangSejumlah bigint(10) NOT NULL, Denda int(10) NOT NULL, KdJenisBayar Varchar(5) NOT NULL);</pre>
Penjelasan :	Membuat table transaksi_cicilan
Hasil	<pre>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 (0.024 sec)</pre>

5. Tampilkan Struktur Table yang baru dibuat

Syntax :	<pre>describe [nama_tabel];</pre>
-----------------	-----------------------------------

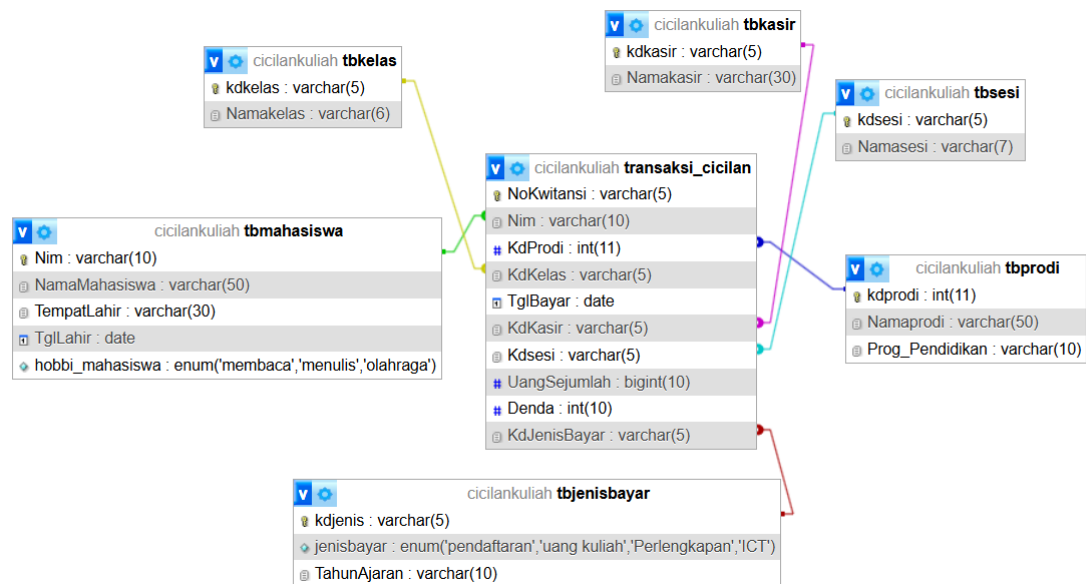
Penjelasan :	Menampilkan struktur tabel
Hasil	<pre> MariaDB [CicilanKuliah]> describe tbprodi; +-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+ kdprodi int(11) NO PRI NULL Namaprodi varchar(50) NO NULL Prog_Pendidikan varchar(10) YES NULL +-----+-----+-----+-----+-----+ 3 rows in set (0.030 sec) MariaDB [CicilanKuliah]> describe tbsesi; +-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+ kdsesi varchar(5) NO PRI NULL Namasesi varchar(7) NO NULL +-----+-----+-----+-----+-----+ 2 rows in set (0.030 sec) MariaDB [CicilanKuliah]> describe tbkelas; +-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+ kdkelas varchar(5) NO PRI NULL Namakelas varchar(6) NO NULL +-----+-----+-----+-----+-----+ 2 rows in set (0.001 sec) MariaDB [CicilanKuliah]> describe 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 (0.031 sec) MariaDB [CicilanKuliah]> describe tbkasir; +-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+ kdkasir varchar(5) NO PRI NULL Namakasir varchar(10) NO NULL +-----+-----+-----+-----+-----+ 2 rows in set (0.030 sec) MariaDB [CicilanKuliah]> describe tbjenisbayar; +-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+ kjenis varchar(5) NO PRI NULL jenisbayar enum('pendaftaran','uang kuliah','Perlempangan','ICT') YES NULL TabungJaran varchar(10) NO NULL +-----+-----+-----+-----+-----+ 3 rows in set (0.030 sec) MariaDB [CicilanKuliah]> describe tbmahasiswa; +-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+ Nim varchar(10) NO PRI NULL NamaMahasiswa varchar(30) YES NULL TanggalLahir varchar(30) NO NULL TglLahir date NO NULL +-----+-----+-----+-----+-----+ 4 rows in set (0.030 sec) </pre>

➤ Sub CMK (2) . DDL Database : **Alter**

- Dengan Perintah Alter Berikan Foreign Key pada Tabel transaksi_cicilan untuk kolom: NIM, KdProdi, KdKelas, KdKasir, KdSesi, KdJenisBayar

Syntax :	Alter table [nama_tabel] add Foreign Key(kolom_target) REFERENCES [tabel_asal]([kolom yang memiliki Primary Key dan tipe data yg sama])
Penjelasan :	Menambahkan foreign key pada kolom yang ada pada transaksi cicilan
Hasil	<pre> MariaDB [CicilanKuliah]> alter table transaksi_cicilan add foreign key (Nim) references tbmahasiswa(Nim); Query OK, 0 rows affected (0.129 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [CicilanKuliah]> alter table transaksi_cicilan add foreign key (KdProdi) references tbprodi(KdProdi); Query OK, 0 rows affected (0.125 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [CicilanKuliah]> alter table transaksi_cicilan add foreign key (KdKelas) references tbkelas(KdKelas); Query OK, 0 rows affected (0.128 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [CicilanKuliah]> alter table transaksi_cicilan add foreign key (KdKasir) references tbkasir(KdKasir); Query OK, 0 rows affected (0.124 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [CicilanKuliah]> alter table transaksi_cicilan add foreign key (KdSesi) references tbsesi(KdSesi); Query OK, 0 rows affected (0.104 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [CicilanKuliah]> alter table transaksi_cicilan add foreign key (KdJenisBayar) references tbjenisbayar(KdJenis); Query OK, 0 rows affected (0.137 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [CicilanKuliah]> 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 (0.035 sec) </pre>

Dengan Memberikan FOREIGN KEY ke PRIMARY KEY yang sudah ada maka akan membentuk Relasi antar Tabel sebagai berikut:



GAMBAR 2.1 RELASI ANTAR TABEL (ERD) CICILAN KULIAH

7. Dengan Perintah Alter Tambahkan Kolom jenis_kelamin varchar(1) Not Null di TbMahasiswa di bawah kolom NamaMahasiswa

Syntax :	Alter table tbmahasiswa add jenis_kelamin varchar(1) NOT NULL after NamaMahasiswa;
Penjelasan :	Menambahkan Kolom jenis_kelamin varchar(1) Not Null di TbMahasiswa di bawah kolom NamaMahasiswa
Hasil	<pre> MariaDB [CicilanKuliah]> alter table tbmahasiswa add jenis_kelamin Varchar(1) NOT NULL after NamaMahasiswa; Query OK, 0 rows affected (0.022 sec) Records: 0 Duplicates: 0 Warnings: 0 </pre>

8. Dengan Perintah Alter Tambahkan Kolom Hobbi dengan type enum dimana hobbi nya: membaca, menulis, olahraga

Syntax :	Alter table tbmahasiswa add Hobbi enum('membaca','menulis','olahraga');
Penjelasan :	Menambahkan Kolom Hobbi dengan type enum dimana hobbi nya: membaca, menulis, olahraga
Hasil	<pre> MariaDB [CicilanKuliah]> alter table tbmahasiswa add Hobbi enum('membaca','menulis','olahraga'); Query OK, 0 rows affected (0.015 sec) Records: 0 Duplicates: 0 Warnings: 0 </pre>

9. Dengan Perintah Alter ubah size Kolom NamaMahasiswa menjadi 50

Syntax :	Alter table tbmahasiswa MODIFY NamaMahasiswa varchar(50);
Penjelasan :	Mengubah size kolom NamaMahasiswa menjadi 50
Hasil	<pre>MariaDB [CicilanKuliah]> alter table tbmahasiswa MODIFY NamaMahasiswa Varchar(50); Query OK, 0 rows affected (0.018 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

10. Dengan Perintah Alter ubah Kolom hobi menjadi Hobbi_mahasiswa

Syntax :	Alter table tbmahasiswa change Hobbi hobbi_mahasiswa enum('membaca','menulis','olahraga');
Penjelasan :	Mengubah kolom hobi menjadi Hobbi_mahasiswa
Hasil	<pre>MariaDB [CicilanKuliah]> alter table tbmahasiswa change hobbi hobbi_mahasiswa enum('membaca','menulis','olahraga'); Query OK, 0 rows affected (0.012 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

11. Dengan Perintah Alter Hapus Kolom Jenis_kelamin

Syntax :	Alter table tbmahasiswa Drop jenis_kelamin;
Penjelasan :	Menghapus kolom jenis_kelamin
Hasil	<pre>MariaDB [CicilanKuliah]> alter table tbmahasiswa DROP jenis_kelamin; Query OK, 0 rows affected (0.045 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

12. Dengan Perintah Alter Hapus Kolom Hobbi

Syntax :	Alter table tbmahasiswa drop Hobbi;
Penjelasan :	Menghapus kolom Hobbi
Hasil	<pre>MariaDB [CicilanKuliah]> alter table tbmahasiswa DROP Hobbi; Query OK, 0 rows affected (0.011 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

➤ Sub CMK (3) . DDL Database : **Create :: Alter :: Drop**

13. Dengan perintah Create table, Buat Table Login

Syntax :	Create table login(username Varchar(5) NOT NULL, Password (10) NOT NULL);
Penjelasan :	Membuat Tabel Login yang berisi Username dan Password
Hasil	<pre>MariaDB [CicilanKuliah]> create table login (username varchar(5) NOT NULL, password varchar(10) NOT NULL); Query OK, 0 rows affected (0.018 sec)</pre>

14. Dengan Perintah Alter Tambahkan Kolom id dengan type int(11) Auto_Increment dengan Key Primary Key

Syntax :	Alter table login add id int(11) AUTO_INCREMENT PRIMARY KEY;
Penjelasan :	Menambahkan kolom id dengan type data int(11) auto_increment dengan Key Primary Key
Hasil	<pre>MariaDB [CicilanKuliah]> alter table login add id int(11) Auto_Increment PRIMARY KEY; Query OK, 0 rows affected (0.057 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

15. Dengan Perintah Alter Tambahkan Kolom firstname dan lastname dengan type Varchar(25)

Syntax :	Alter table login add firstname varchar(25);
Penjelasan :	Menambahkan kolom Firstname dan Lastname dengan tipe data varchar(25)
Hasil	<pre>MariaDB [CicilanKuliah]> alter table login add Firstname Varchar(25); Query OK, 0 rows affected (0.015 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [CicilanKuliah]> alter table login add Lastname Varchar(25); Query OK, 0 rows affected (0.012 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

16. Dengan Perintah Alter Tambahkan Kolom Email dengan type Varchar(100) dibawah kolom Password

Syntax :	Alter table login add email varchar(100) AFTER Password;
Penjelasan :	Menambahkan kolom email dengan type Varchar(100) dibawah kolom password
Hasil	<pre>MariaDB [CicilanKuliah]> alter table login add email Varchar(100) AFTER password; Query OK, 0 rows affected (0.016 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

17. Dengan Perintah Alter Rubah Size username menjadi varchar(8)

Syntax :	Alter table login modify username varchar(8);
Penjelasan :	Mengubah Size Username menjadi varchar(8)
Hasil	<pre>MariaDB [CicilanKuliah]> alter table login MODIFY username varchar(8); Query OK, 0 rows affected (0.055 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

18. Dengan Perintah Alter Ganti kolom Firstname menjadi nama_pertama

Syntax :	Alter table login change Firstname nama_pertama varchar(25);
Penjelasan :	Mengubah kolom firstname menjadi kolom nama_pertama
Hasil	<pre>MariaDB [CicilanKuliah]> alter table login change Firstname nama_pertama varchar(25); Query OK, 0 rows affected (0.017 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>

19. Dengan Perintah Alter Ganti nama Tabel Login menjadi ManagemenUser

Syntax :	Alter table login RENAME to ManagemenUser;
Penjelasan :	Mengubah nama Tabel login menjadi ManagemenUser
Hasil	<pre>MariaDB [CicilanKuliah]> alter table login RENAME to ManagemenUser; Query OK, 0 rows affected (0.075 sec)</pre>

20. Dengan Perintah Drop Hapus Tabel ManagemenUser

Syntax :	Drop Table ManagemenUser;
Penjelasan :	Menghapus Tabel ManagemenUser
Hasil	<pre>MariaDB [CicilanKuliah]> drop table ManagemenUser; Query OK, 0 rows affected (0.025 sec)</pre>

BAB IV

DATA MANIPULATION LANGUAGE (DML)

Data Manipulation Language (DML) adalah bahasa yang digunakan untuk memanipulasi data dalam sebuah sistem manajemen basis data (DBMS). DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian database yang telah dibuat. DML memungkinkan pengguna untuk melakukan operasi-operasi seperti menyisipkan (inserting), mengubah (updating), menghapus (deleting), dan mengambil (retrieving) data dari basis data. (Jamaluddin et al., 2022)

DML berperan sebagai penghubung antara pengguna dan basis data untuk memastikan data dapat dikelola secara efisien. Operasi DML bersifat non-procedural, artinya pengguna hanya menentukan apa yang harus dilakukan tanpa memikirkan cara teknis pelaksanaannya. Tujuan utama dari Data Manipulation Language (DML) adalah memfasilitasi interaksi pengguna dengan data di dalam database, memungkinkan mereka untuk melakukan berbagai operasi pada data secara efektif. (Arianti & Jamaluddin, 2022)

Ciri utama DML adalah interaksi langsung dengan data dan kemampuannya untuk bekerja dengan transaksi. Dalam basis data yang mendukung transaction management, perubahan data melalui DML dapat dikendalikan dengan perintah seperti COMMIT (untuk menyimpan perubahan) dan ROLLBACK (untuk membatalkan perubahan). Contoh penerapan DML melibatkan aplikasi seperti sistem kepegawaian, pengelolaan inventaris, atau sistem penjualan yang memerlukan manipulasi data secara real-time untuk mendukung operasional sehari-hari. (Adflin, 2020)

Berikut adalah beberapa perintah dalam DML yaitu:

1. INSERT

INSERT digunakan untuk menambahkan data kedalam table. Ada 5 jenis penggunaan INSERT yaitu:

1. INSERT INTO WITH NO COLUMN → Memasukkan data ke tabel tanpa menyebut nama kolom.

2. INSERT INTO WITH COLUMN → Memasukkan data dengan menyebut nama kolom yang akan diisi.
3. INSERT INTO SET → Memasukkan data dengan Set Kolom = Nilai.
4. INSERT INTO VALUES → Cara memasukkan data dengan VALUES, bisa untuk satu atau banyak baris.
5. INSERT INTO SELECT → Memasukkan data ke tabel dengan cara mengambil data dari tabel lain. (Anggoro Dimas Aryo et al., 2021)

2. UPDATE

UPDATE digunakan untuk mengubah record dalam table.

Syntax:

UPDATE (nama_table) SET (nama_kolom) = (nilai_baru) WHERE (nama_kolom_unik) = (Nilai_unik); (Amin, 2022)

3. DELETE

DELETE digunakan untuk menghapus record dari table.

Syntax:

DELETE FROM (nama_table) WHERE (nama_kolom_unik)=(nilai_unik);

4. SELECT

SELECT digunakan untuk menampilkan data atau record pada table.

Ada beberapa jenis select yaitu:

1. SELECT ... AS → Menampilkan data sambil mengganti nama kolom (alias).
2. SELECT ...ORDER BY → Menampilkan data dengan urutan asc atau desc.
3. SELECT ...LIMIT → Menampilkan data dengan batas jumlah tertentu.
4. SELECT ...WHERE → Menampilkan data dengan syarat tertentu.
5. SELECT ...WHERE...IN → Menampilkan data yang kolomnya cocok dengan beberapa nilai dalam daftar.
6. SELECT...WHERE...BETWEEN → Menampilkan data dengan rentang nilai.
7. SELECT...WHERE...LIKE → Menampilkan data dengan pola teks tertentu.
8. SELECT DISTINCT → Menampilkan data dengan menghilangkan duplikasi.

9. SELECT...GROUP BY → Mengelompokkan data berdasarkan kolom tertentu.
10. SELECT AND,OR,NOT → Menampilkan data dengan kondisi logika.
11. SELECT AGREGATE → Menampilkan data dengan fungsi agregat seperti SUM, AVG, MAX, MIN, COUNT. (Garcia et al., n.d.)

HASIL DAN PEMBAHASAN

➤ Sub CMK (4) . DML Database : INSERT

21. Dengan Perintah Insert into dengan Kolom Tambahkan 4 Record di tbprodi

Syntax :	Insert Into tbprodi ([kolom1], [kolom2], [kolom n]...) VALUES ('[Record1]', '[Record2]', ...);
Penjelasan :	Menambahkan 4 Record kolom di tabel tbprodi
Hasil	<pre> MariaDB [cicilankuliah]> 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 (0.006 sec) Records: 4 Duplicates: 0 Warnings: 0 </pre>

22. Tampilkan tbprodi

Syntax :	select * from tbprodi;
Penjelasan :	Menampilkan record dari tabel tbprodi
Hasil	<pre> 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 (0.001 sec) </pre>

23. Dengan Perintah insert into tanpa Kolom Tambahkan 3 Record di tbsesi diatas

Syntax :	Insert Into tbsesi VALUES ('[Record1]', '[Record2]', ...);
Penjelasan :	Menambahkan 3 record tanpa kolom pada tabel tbsesi
Hasil	<pre> MariaDB [CicilanKuliah]> insert INTO tbsesi VALUES ('M','Malam'); Query OK, 1 row affected (0.013 sec) MariaDB [CicilanKuliah]> insert INTO tbsesi VALUES ('P','Pagi'); Query OK, 1 row affected (0.010 sec) MariaDB [CicilanKuliah]> insert INTO tbsesi VALUES ('S','Siang'); Query OK, 1 row affected (0.010 sec) </pre>

24. Tampilkan tbsesi

Syntax :	select * from tbsesi;
Penjelasan :	Menampilkan isi record dari Tabel tbsesi
Hasil	<pre> MariaDB [CicilanKuliah]> select * from tbsesi; +-----+-----+ kdsesi Namasesi +-----+-----+ M Malam P Pagi S Siang +-----+-----+ 3 rows in set (0.001 sec) </pre>

25. Dengan Perintah Insert into Multi Values Tambahkan 10 Record di tbkelas

Syntax :	Insert Into tbsesi VALUES('kolom1','kolom2',...));
Penjelasan :	Menambahkan 10 record tanpa kolom pada tabel tbkelas
Hasil	<pre> MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('201','20S01'); Query OK, 1 row affected (0.011 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('202','20S02'); Query OK, 1 row affected (0.006 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('203','20S03'); Query OK, 1 row affected (0.010 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('204','20S04'); Query OK, 1 row affected (0.006 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('211','21S01'); Query OK, 1 row affected (0.005 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('212','21S02'); Query OK, 1 row affected (0.009 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('213','21S03'); Query OK, 1 row affected (0.005 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('214','21S04'); Query OK, 1 row affected (0.005 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('215','21S05'); Query OK, 1 row affected (0.011 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('216','21S06'); Query OK, 1 row affected (0.010 sec) MariaDB [CicilanKuliah]> insert INTO tbkelas VALUES ('221','22S01'); Query OK, 1 row affected (0.010 sec) </pre>

26. Tampilkan tbkelas

Syntax :	select * from tbkelas;
Penjelasan :	Menampilkan isi record dari tabel tbkelas
Hasil	<pre> MariaDB [CicilanKuliah]> select * from tbkelas; +-----+-----+ kdkelas Namakelas +-----+-----+ 201 20S01 202 20S02 203 20S03 204 20S04 211 21S01 212 21S02 213 21S03 214 21S04 215 21S05 216 21S06 221 22S01 +-----+-----+ 11 rows in set (0.001 sec) </pre>

27. Dengan Perintah Insert into Set Tambahkan 4 Record di tbkasir

Syntax :	Insert into tbkasirr → Set kdkasir = '[record]', → Namakasir = '[record]';
Penjelasan :	Menambahkan 4 Record dengan Metode Insert Into Set pada tabel tbkasir
Hasil	<pre> MariaDB [cicilankuliah]> insert into tbkasirr -> SET kdkasir = 'KS1', -> Namakasir = 'Fahmi Firzada, A.MD'; Query OK, 1 row affected, 1 warning (0.063 sec) MariaDB [cicilankuliah]> insert into tbkasirr -> SET kdkasir = 'KS2', -> Namakasir = 'Legina, A.MD'; Query OK, 1 row affected (0.060 sec) MariaDB [cicilankuliah]> insert into tbkasirr -> SET kdkasir = 'KS3', -> Namakasir = 'Sri Rahayu Purba'; Query OK, 1 row affected (0.060 sec) MariaDB [cicilankuliah]> insert into tbkasirr -> SET kdkasir = 'KS4', -> Namakasir = 'Elisa, A.MD'; Query OK, 1 row affected (0.060 sec) </pre>

28. Tampilkan tbkasir

Syntax :	select * from tbkasirr;
Penjelasan :	Menambahkan 4 Record dengan Metode Insert Into Set pada tabel tbjenisbayar
Hasil	<pre> MariaDB [cicilankuliah]> select * from tbkasirr; +-----+-----+ kdkasir Namakasir +-----+-----+ KS1 Fahmi Firzada, A.MD KS2 Legina, A.MD KS3 Sri Rahayu Purba KS4 Elisa, A.MD +-----+-----+ </pre>

29. Dengan perintah insert into Set Tambahkan 4 Record di tbjenisbayar

Syntax :	Insert into tbkasirr → Set kdjenis = '[record]', → jenisbayar = '[record]', → TahunAjaran = '[record]';
Penjelasan :	Menambahkan 4 record dengan perintah Insert into Set pada tabel tbjenisbayarr
Hasil	<pre> MariaDB [cicilankuliah]> insert into tbjenisbayarr -> SET kdjenis = 'JN01', -> jenisbayar = 'Pendaftaran', -> TahunAjaran = '2023/2024'; Query OK, 1 row affected (0.008 sec) MariaDB [cicilankuliah]> insert into tbjenisbayarr -> SET kdjenis = 'JN02', -> jenisbayar = 'Perlengkapan', -> TahunAjaran = '2023/2024'; Query OK, 1 row affected (0.003 sec) MariaDB [cicilankuliah]> insert into tbjenisbayarr -> SET kdjenis = 'JN03', -> jenisbayar = 'Uang Kuliah', -> TahunAjaran = '2023/2024'; Query OK, 1 row affected (0.007 sec) MariaDB [cicilankuliah]> insert into tbjenisbayarr -> SET kdjenis = 'JN04', -> jenisbayar = 'ICT', -> TahunAjaran = '2023/2024'; Query OK, 1 row affected (0.006 sec) </pre>

30. Tampilkan tbjenisbayar

Syntax :	select * from tbjenisbayarr;
Penjelasan :	Menampilkan isi record dari tabel tbjenisbayarr
Hasil	<pre> MariaDB [cicilankuliah]> select * from tbjenisbayarr; +-----+-----+-----+ 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) </pre>

31. Dengan perintah insert into dengan kolom Tambahkan 5 Record di tbmahasiswa diatas

Syntax :	Insert into tbmahasiswa (Kolom1,kolom2,kolom n,...) VALUES ('Record1','Record2','Record3',...);
Penjelasan :	Menambahkan 5 Record di kolom tbmahasiswa
Hasil	<pre> MariaDB [cicilankuliah]> insert into tbmahasiswa (Nim,NamaMahasiswa,TempatLahir,TglLahir) VALUES ('2102161','Indah F Sari','P.Siantar','2003-07-15'); Query OK, 1 row affected (0.148 sec) MariaDB [cicilankuliah]> insert into tbmahasiswa (Nim,NamaMahasiswa,TempatLahir,TglLahir) VALUES ('2303006','Najwa Mutia Armayani Nasution','Laras','2005-11-13'); Query OK, 1 row affected (0.060 sec) MariaDB [cicilankuliah]> insert into tbmahasiswa (Nim,NamaMahasiswa,TempatLahir,TglLahir) VALUES ('2303007','Novralina Anggraini Purba','P.Siantar','2002-11-07'); Query OK, 1 row affected (0.060 sec) MariaDB [cicilankuliah]> insert into tbmahasiswa (Nim,NamaMahasiswa,TempatLahir,TglLahir) VALUES ('2303016','Hafizah Rahmi Lubis','P.Siantar','2004-04-2'); Query OK, 1 row affected (0.060 sec) MariaDB [cicilankuliah]> insert into tbmahasiswa (Nim,NamaMahasiswa,TempatLahir,TglLahir) VALUES ('2303017','Zaskia Aulia Zahra','P.Siantar','2006-05-23'); Query OK, 1 row affected (0.062 sec) </pre>

32. Tampilkan tbmahasiswa

Syntax :	select * from tbmahasiswa;
Penjelasan :	Menampilkan isi Record dari tabel tbmahasiswa
Hasil	<pre> MariaDB [cicilankuliah]> select * from tbmahasiswa; +-----+-----+-----+-----+ Nim NamaMahasiswa TempatLahir TglLahir +-----+-----+-----+-----+ 2102161 Indah F Sari P.Siantar 2003-07-15 2303006 Najwa Mutia Armayani Nasution Laras 2005-11-13 2303007 Novralina Anggraini Purba P.Siantar 2002-11-07 2303016 Hafizah Rahmi Lubis P.Siantar 2004-04-02 2303017 Zaskia Aulia Zahra P.Siantar 2006-05-23 +-----+-----+-----+-----+ 5 rows in set (0.001 sec) </pre>

33. Dengan perintah Insert into Multi Values Tambahkan 15 Record di transaksi_cicilan

Syntax :	Insert into transaksi_cicilan → (kolom1, kolom2, kolom3, kolom n...) → VALUES → ('Record1',,Record2',,Record3',,Record n...');
Penjelasan :	Menambahkan 15 Record pada tabel transaksi_cicilan dengan INSERT INTO MULTIVALUES
Hasil	<pre> MariaDB [cicilankuliah]> INSERT INTO transaksi_cicilan -> (Nokwitansi, Nim, KdProdi, KdKelas, TglBayar, KdKasir, KdSesi, UangSejumlah, Denda, KdJenisBayar) -> VALUES -> ('10001', '2303017', '1', '201', '2023-12-20', 'KS1', 'P', '0', '0', 'JN01'), -> ('10002', '2303017', '1', '201', '2024-01-15', 'KS2', 'P', '0', '0', 'JN02'), -> ('10003', '2303017', '1', '201', '2024-02-10', 'KS3', 'P', '0', '0', 'JN03'), -> ('10004', '2303006', '1', '202', '2023-11-25', 'KS1', 'P', '0', '0', 'JN04'), -> ('10005', '2303006', '1', '202', '2023-12-05', 'KS2', 'P', '0', '0', 'JN01'), -> ('10006', '2303006', '1', '202', '2024-01-15', 'KS3', 'P', '0', '0', 'JN02'), -> ('10007', '2303007', '1', '203', '2023-10-10', 'KS4', 'P', '0', '0', 'JN03'), -> ('10008', '2303007', '1', '203', '2023-11-11', 'KS1', 'P', '0', '0', 'JN04'), -> ('10009', '2303007', '1', '203', '2023-12-12', 'KS2', 'P', '0', '0', 'JN01'), -> ('10010', '2303016', '1', '211', '2024-01-05', 'KS3', 'P', '0', '0', 'JN02'), -> ('10011', '2303016', '1', '211', '2024-02-15', 'KS4', 'P', '0', '0', 'JN03'), -> ('10012', '2303016', '1', '211', '2024-03-01', 'KS1', 'P', '0', '0', 'JN04'), -> ('10013', '2102161', '1', '212', '2023-12-01', 'KS2', 'P', '0', '0', 'JN01'), -> ('10014', '2102161', '1', '212', '2024-01-20', 'KS3', 'P', '0', '0', 'JN02'), -> ('10015', '2102161', '1', '212', '2024-02-25', 'KS4', 'P', '0', '0', 'JN03'); Query OK, 15 rows affected (0.045 sec) Records: 15 Duplicates: 0 Warnings: 0 </pre>

34. Tampilkan tbmahasiswa

Syntax :	select * from transaksi_cicilan;
Penjelasan :	Menampilkan Isi record dari transaksi_cicilan
Hasil	<pre> MariaDB [cicilankuliah]> select * from transaksi_cicilan; +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ Nokwitansi Nim KdProdi KdKelas TglBayar KdKasir KdSesi UangSejumlah Denda KdJenisBayar +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ 10001 2303017 1 201 2023-12-20 KS1 P 0 0 JN01 10002 2303017 1 201 2024-01-15 KS2 P 0 0 JN02 10003 2303017 1 201 2024-02-10 KS3 P 0 0 JN03 10004 2303006 1 202 2023-11-25 KS1 P 0 0 JN04 10005 2303006 1 202 2023-12-05 KS2 P 0 0 JN01 10006 2303006 1 202 2024-01-15 KS3 P 0 0 JN02 10007 2303007 1 203 2023-10-10 KS4 P 0 0 JN03 10008 2303007 1 203 2023-11-11 KS1 P 0 0 JN04 10009 2303007 1 203 2023-12-12 KS2 P 0 0 JN01 10010 2303016 1 211 2024-01-05 KS3 P 0 0 JN02 10011 2303016 1 211 2024-02-15 KS4 P 0 0 JN03 10012 2303016 1 211 2024-03-01 KS1 P 0 0 JN04 10013 2102161 1 212 2023-12-01 KS2 P 0 0 JN01 10014 2102161 1 212 2024-01-20 KS3 P 0 0 JN02 10015 2102161 1 212 2024-02-25 KS4 P 0 0 JN03 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ 15 rows in set (0.001 sec) </pre>

➤ Sub CMK (5) . DML Database : **UPDATE**

35. Dengan Perintah Update perbaiki data Transaksi_cicilan untuk

JN01 UangSejumlah = 200000

JN02 UangSejumlah = 1000000

JN03 UangSejumlah = 4000000

JN04 UangSejumlah = 2000000

Syntax :	Update transaksi_cicilan SET [kolom] = 'Nilai' WHERE [kolom_unik] = 'Nilai Unik';
Penjelasan :	Mengubah data Transaksi Jualan sesuai Ketentuan JN yang dibutuhkan

Hasil	<pre> MariaDB [cicilankuliah]> UPDATE transaksi_cicilan -> SET UangSejumlah = 200000 -> WHERE KdJenisBayar = 'JN01'; Query OK, 4 rows affected (0.016 sec) Rows matched: 4 Changed: 4 Warnings: 0 MariaDB [cicilankuliah]> UPDATE transaksi_cicilan -> SET UangSejumlah = 1000000 -> WHERE KdJenisBayar = 'JN02'; Query OK, 4 rows affected (0.061 sec) Rows matched: 4 Changed: 4 Warnings: 0 MariaDB [cicilankuliah]> UPDATE transaksi_cicilan -> SET UangSejumlah = 4000000 -> WHERE KdJenisBayar = 'JN03'; Query OK, 4 rows affected (0.061 sec) Rows matched: 4 Changed: 4 Warnings: 0 MariaDB [cicilankuliah]> UPDATE transaksi_cicilan -> SET UangSejumlah = 2000000 -> WHERE KdJenisBayar = 'JN04'; Query OK, 3 rows affected (0.057 sec) Rows matched: 3 Changed: 3 Warnings: 0 </pre>
--------------	--

36. Dengan Perintah Update perbaiki data transaksi_cicilan untuk denda seluruhnya adalah 10000

Syntax :	Update transaksi_cicilan SET Denda = 10000;
Penjelasan :	Mengubah seluruh denda bernilai 10000 dengan Update
Hasil	<pre> MariaDB [cicilankuliah]> UPDATE transaksi_cicilan -> SET Denda = 10000; Query OK, 15 rows affected (0.060 sec) Rows matched: 15 Changed: 15 Warnings: 0 </pre>

37. Dengan Perintah Update perbaiki data transaksi_cicilan Untuk

NIM 2303007 KDSESI = S

NIM 2303016 KDSESI = M

Syntax :	<pre> Update transaksi_cicilan SET KdSesi = S WHERE Nim = '2303007'; Update transaksi_cicilan SET KdSesi = M WHERE Nim = '2303016'; </pre>
Penjelasan :	Mengubah Kolom KdSesi menjadi S untuk Nim 2303007, dan M untuk Nim 2303016
Hasil	<pre> MariaDB [cicilankuliah]> UPDATE transaksi_cicilan -> SET KdSesi = 'S' -> WHERE Nim = '2303007'; Query OK, 3 rows affected (0.006 sec) Rows matched: 3 Changed: 3 Warnings: 0 </pre>

```
MariaDB [cicilankuliah]> UPDATE transaksi_cicilan
-> SET KdSesi = 'M'
-> WHERE Nim = '2303016';
Query OK, 3 rows affected (0.006 sec)
Rows matched: 3 Changed: 3 Warnings: 0
```

38. Tampilkan transaksi_cicilan

Syntax :	select * from transaksi_cicilan;
Penjelasan :	Menampilkan semua isi record dari tabel transaksi_cicilan
Hasil	<pre>MariaDB [cicilankuliah]> UPDATE transaksi_cicilan -> SET Denda = 10000; Query OK, 15 rows affected (0.060 sec) Rows matched: 15 Changed: 15 Warnings: 0</pre>

➤ Sub CMK (6) . DML Database : **DELETE**

39. Dengan Perintah DELETE Hapus Record dengan NoKwitansi = 10001

Syntax :	DELETE FROM transaksi_cicilan ➔ WHERE NoKwitansi = '10001';
Penjelasan :	Menghapus Record dengan NoKwitansi = 10001
Hasil	<pre>MariaDB [cicilankuliah]> delete from transaksi_cicilan -> WHERE NoKwitansi = '10001'; Query OK, 1 row affected (0.071 sec)</pre>

40. Dengan Printah DELETE Hapus Record dengan NoKwitansi = 10002 dan 10003 dengan Perintah IN

Syntax :	DELETE FROM transaksi_cicilan ➔ WHERE NoKwitansi IN ('10002','10003');
Penjelasan :	Menghapus Record dengan NoKwitansi = 00002 dan 00003 dengan Perintah IN
Hasil	<pre>MariaDB [cicilankuliah]> delete from transaksi_cicilan -> WHERE NoKwitansi IN ('10002','10003'); Query OK, 2 rows affected (0.060 sec)</pre>

Menampilkan Tabel transaksi_cicilan

```
MariaDB [cicilankuliah]> select * from transaksi_cicilan;
```

NoKwitansi	Nim	KdProdi	KdKelas	TglBayar	KdKasir	Kdsesi	UangSejumlah	Denda	KdJenisBayar
10004	2303006	1	202	2023-11-25	KS1	P	2000000	10000	JN04
10005	2303006	1	202	2023-12-05	KS2	P	2000000	10000	JN01
10006	2303006	1	202	2024-01-15	KS3	P	1000000	10000	JN02
10007	2303007	1	203	2023-10-19	KS4	S	4000000	10000	JN03
10008	2303007	1	203	2023-11-11	KS1	S	2000000	10000	JN04
10009	2303007	1	203	2023-12-12	KS2	S	2000000	10000	JN01
10010	2303016	1	211	2024-01-05	KS3	M	1000000	10000	JN02
10011	2303016	1	211	2024-02-15	KS4	M	4000000	10000	JN03
10012	2303016	1	211	2024-03-01	KS1	M	2000000	10000	JN04
10013	2102161	1	212	2023-12-01	KS2	P	2000000	10000	JN01
10014	2102161	1	212	2024-01-20	KS3	P	1000000	10000	JN02
10015	2102161	1	212	2024-02-25	KS4	P	4000000	10000	JN03

12 rows in set (0.001 sec)

PERTEMUAN KE 3 (UJI PROBLEM SOLVING)

Dosen	:	Dr. Dedy Hartama, M.Kom
Topik	:	Select Statement, DML
Pertemuan	:	Ke 3
CPMK Capaian Pembelajaran Mata Kuliah	:	• 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
Nama Mahasiswa	:	Nama Ketua : 1. Arif Hasudungan Silaban (CMK 4) Anggota: 2. Rafael Sinaga (CMK 2 + 5) 3. Diaz Al Latif (CMK 2) 4. Ondo Rajagukguk (CMK 2) 5. Febi Sofia Azzura (CMK 3) 6. Gadis Fitria (CMK 3) 7. Dea Cindiasyahwa (CMK 1) 8. Ayuhana Vanessya purba (CMK 3) 9. Yosi Anggraini (CMK 1) 10. Azura Intan (CMK 1)
Kelas	:	24S04
Semester	:	3 (GANJIL)
Tgl Pertemuan	:	18 September 2025
Nilai		

PRAKTIKUM 3 (SELECT): HASIL DAN PEMBAHASAN

➤ Sub CMK(1) . Review DDL dan DML : Create, ALTER, INSERT

1. Membuat Database dengan Nama: UjiKompetensi2

Syntax :	create database Ujikompetensi2;
Penjelasan :	Membuat Database dengan nama Ujikompetensi2
Hasil	<pre>MariaDB [(none)]> create database Ujikompetensi2; Query OK, 1 row affected (0.002 sec)</pre>

2. Tampilkan database

Syntax :	Show databases;
Penjelasan :	Menampilkan database
Hasil	<pre>MariaDB [(none)]> create database Ujikompetensi2; Query OK, 1 row affected (0.002 sec)</pre>

3. Masuk ke dalam database

Syntax :	use Ujikompetensi2;
Penjelasan :	Masuk ke dalam database Ujikompetensi2
Hasil	<pre>MariaDB [(none)]> use Ujikompetensi2; Database changed</pre>

4. Buat table tbmahasiswa tahap pertama

Syntax :	create table tbmahasiswa(Nim Varchar(10) PRIMARY KEY, NamaMahasiswa varchar(30) NULL, TempatLahir Varchar(30) NOT NULL, Tgl Lahir Date NOT NULL;
Penjelasan :	Membuat Table tbmahasiswa tahap 1 yang berisi kolom Nim, NamaMahasiswa, TempatLahir, dan TglLahir
Hasil	<pre>MariaDB [Ujikompetensi2]> create table tbmahasiswa(Nim Varchar(10) PRIMARY KEY, NamaMahasiswa Varchar(30) NULL, TempatLahir Varchar(30) NOT NULL, TglLahir DATE NOT NULL); Query OK, 0 rows affected (0.023 sec)</pre>

5. 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

Syntax :	Alter table [nama_table] add [kolom] [tipe_data(size)];
Penjelasan :	Menambah kolom sebagai berikut: Agama dan Jenis Kelamin di bawah NamaMahasiswa, Tempat Tinggal dibawah TglLahir, UTS dan Gaji Ortu dibawah TglLahir di kolom terakhir
Hasil	<pre> MariaDB [UjiKompetensi2]> alter table tbmahasiswa add AGAMA Varchar(20) NULL After NamaMahasiswa; Query OK, 0 rows affected (0.034 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [UjiKompetensi2]> alter table tbmahasiswa add JenisKelamin Varchar(10) NULL After Agama; Query OK, 0 rows affected (0.016 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [UjiKompetensi2]> alter table tbmahasiswa add TempatTinggal Varchar(25) NOT NULL after TempatLahir; Query OK, 0 rows affected (0.015 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [UjiKompetensi2]> alter table tbmahasiswa add UTS int(11) NULL After TglLahir; Query OK, 0 rows affected (0.016 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [UjiKompetensi2]> alter table tbmahasiswa add GajiOrtu bigint(16) NULL After UTS; Query OK, 0 rows affected (0.019 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [UjiKompetensi2]> desc tbmahasiswa; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ Nim varchar(10) NO PRI NULL NamaMahasiswa varchar(30) YES NULL AGAMA varchar(20) YES NULL JenisKelamin varchar(10) YES NULL TempatLahir varchar(30) NO NULL TempatTinggal varchar(25) YES NULL TglLahir date NO NULL UTS int(11) YES NULL GajiOrtu bigint(16) YES NULL +-----+-----+-----+-----+-----+-----+ 9 rows in set (0.033 sec) </pre>

6. Isikan Record Tabel tbmahasiswa yang sudah ditambah dengan Alter (Bebas dengan insert apapun)

Syntax :	Insert Into tbprodi ([kolom1], [kolom2], [kolom n]...) VALUES ('[Record1]', '[Record2]',);
Penjelasan :	Menambahkan record pada tabel tbmahasiswa dengan insert bebas
Hasil	<pre> MariaDB [UjiKompetensi2]> Insert Into tbmahasiswa(Nim, NamaMahasiswa, AGAMA, JenisKelamin, TempatLahir, TempatTinggal, TglLahir, UTS, GajiOrtu) VALUES -> ('2302001','Arif Silaban','Kristen','laki-laki','Penatangsiantar','Penatangsiantar','2005-11-15','100','5000000'), -> ('2302002','Diaz Al Latif','Islam','laki-laki','Tanah Jawa','Bah Jambi','2006-07-17','95','6000000'), -> ('2302003','Rafael Sinaga','Kristen','laki-laki','Jakarta','Simbolon','2006-04-19','90','6500000'), -> ('2302004','Dea Cindia','Islam','Perempuan','Papua Barat','P.Siantar','2005-12-16','85','6800000'), -> ('2302005','Azura Intan','Islam','Perempuan','Serapuh','Serapuh','2006-07-16','80','7000000'), -> ('2302006','Febi Sofii','Islam','Perempuan','Rambung Merah','Rambung Merah','2006-02-06','87','5500000'), -> ('2302007','Ondo Saut','Kristen','laki-laki','Balimbingan','Balimbingan','2006-05-12','91','5700000'); Query OK, 7 rows affected (0.012 sec) Records: 7 Duplicates: 0 Warnings: 0 </pre>

➤ Sub CMK (2) . DQL Database : Select Statement Fungsi String

Diperoleh Data sebagai Berikut:

Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu
2302001	Arif Silaban	Kristen	laki-laki	Pematangsiantar	Pematangsiantar	2005-11-15	100	5000000
2302002	Diaz Al Latif	Islam	laki-laki	Tanah Jawa	Bah Jambi	2006-07-17	95	6000000
2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000
2302004	Dea Cindia	Islam	Perempuan	Papua Barat	P.Siantar	2005-12-16	85	6800000
2302005	Azura Intan	Islam	Perempuan	Serapuh	Serapuh	2006-07-16	80	7000000
2302006	Febi Sofii	Islam	Perempuan	Rambung Merah	Rambung Merah	2006-02-06	87	5500000
2302007	Ondo Saut	Kristen	laki-laki	Balimbingan	Balimbingan	2006-05-12	91	5700000

7. Tampilkan data dengan Fungsi As String

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

Syntax :	<code>select [kolom1] AS '[Nama Alias]', [kolom2] AS '[Nama Alias]', [kolom n] AS '[Nama Alias]' From [nama_table];</code>
Penjelasan :	Menampilkan data Nim, NamaMahasiswa, dan Tempat tinggal dengan fungsi AS STRING
Hasil	<pre> MariaDB [UjiKompetensi2]> select nim AS 'Nim Mahasiswa', -> NamaMahasiswa AS 'Nama Mahasiswa', -> TempatTinggal AS 'Tempat Tinggal' From tbmahasiswa; +-----+-----+-----+ Nim Mahasiswa Nama Mahasiswa Tempat Tinggal +-----+-----+-----+ 2302001 Arif Silaban Pematangsiantar 2302002 Diaz Al Latif Bah Jambi 2302003 Rafael Sinaga Simbolon 2302004 Dea Cindia P.Siantar 2302005 Azura Intan Serapuh 2302006 Febi Sofii Rambung Merah 2302007 Ondo Saut Balimbingan +-----+-----+-----+ 7 rows in set (0.001 sec) </pre>

8. Tampilkan data dibawah ini dengan Fungsi TRUNCATE(DATEDIFF())

NIM	NAMAMAHASISWA	TGLLAHIR	UMUR
2302001	ABDULLAH MUJADID AL QAWI	2003-07-15	21
2302002	ADINDA NABILA	2005-11-13	19
2302005	ANGGUN BUNGA LESTARI LUBIS	2002-11-07	22
2302006	ANNISA TRINABILA	2004-04-22	20
2302008	BAGAS TRY ATMAJA	2004-04-22	20
2302009	BUNGA SABILA	2006-05-23	18
2302364	ALMAH WIJAYA ALHAKIM	2002-11-07	22

Syntax :	Select [kolom1] AS '[Nama Alias]', [kolom2] AS '[Nama Alias]', [kolom n] AS '[Nama Alias]' TRUNCATE(DATEDIFF(CURDATE(), [Kolom])/365,0) AS 'Umur' From [nama_table]
Penjelasan :	Menampilkan Data Nim, NamaMahasiswa, TglLahir dengan nama Alias dengan tambahan Umur berdasarkan Curdate(tanggal saat ini) per 365 sebagai hitungan 1 tahun curdate pada tabel tbmahasiswa;
Hasil	<pre> MariaDB [UjiKompetensi2]> select Nim AS 'NIM', -> NamaMahasiswa AS 'NAMA MAHASISWA', -> TglLahir AS 'TANGGAL LAHIR', -> TRUNCATE(DATEDIFF(CURDATE(), TglLahir)/365,0) AS 'UMUR' from tbmahasiswa; +-----+-----+-----+-----+ NIM NAMA MAHASISWA TANGGAL LAHIR UMUR +-----+-----+-----+-----+ 2302001 Arif Silaban 2005-11-15 19 2302002 Diaz Al Latif 2006-07-17 19 2302003 Rafael Sinaga 2006-04-19 19 2302004 Dea Cindia 2005-12-16 19 2302005 Azura Intan 2006-07-16 19 2302006 Febi Sofii 2006-02-06 19 2302007 Ondo Saut 2006-05-12 19 +-----+-----+-----+-----+ 7 rows in set (0.014 sec) </pre>

9. Tampilkan data dibawah ini dengan Fungsi Perkalian Kolom dan Fungsi String. Dimana Gaji Orang Tua Satu tahun = GAJI_ORTU * 12

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

Syntax :	Select [kolom1] AS '[Nama Alias]', [kolom2] AS '[Nama Alias]', [kolom n] AS '[Nama Alias]' TRUNCATE(DATEDIFF(CURDATE(), [Kolom])/365,0) AS 'Umur' From [nama_table]
Penjelasan :	Menampilkan Data Nim, NamaMahasiswa, TglLahir dengan nama Alias dengan tambahan Umur berdasarkan Curdate(tanggal saat ini) per 365 sebagai hitungan 1 tahun curdate pada tabel tbmahasiswa;
Hasil	

	<pre> MariaDB [UjiKompetensi2]> select Nim AS 'NIM', -> NamaMahasiswa AS 'NAMA MAHASISWA', -> TglLahir AS 'TANGGAL LAHIR', -> TRUNCATE(DATEDIFF(CURDATE(), TglLahir)/365,0) AS 'UMUR' from tbmahasiswa; </pre> <pre> +-----+-----+-----+-----+ NIM NAMA MAHASISWA TANGGAL LAHIR UMUR +-----+-----+-----+-----+ 2302001 Arif Silaban 2005-11-15 19 2302002 Diaz Al Latif 2006-07-17 19 2302003 Rafael Sinaga 2006-04-19 19 2302004 Dea Cindia 2005-12-16 19 2302005 Azura Intan 2006-07-16 19 2302006 Febi Sofii 2006-02-06 19 2302007 Ondo Saut 2006-05-12 19 +-----+-----+-----+-----+ 7 rows in set (0.014 sec) </pre>
--	---

10. Tampilkan data dibawah ini dengan Fungsi String

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

Syntax :	<pre> Select Nim AS 'NIM', NamaMahasiswa AS 'NAMA MAHASISWA', 'Memiliki Gaji Orang tua perbulan' AS 'Memiliki gaji Orang Tua Perbulan', GajiOrtu AS 'Gaji Bulanan Orang Tua' from tbmahasiswa; </pre>
Penjelasan :	Menampilkan Data Nim, NamaMahasiswa dengan nama Alias dan 'Memiliki Gaji Orang Tua Perbulan' untuk pengisian Default dengan nama alias Memiliki gaji Orang tua perbulan pada tabel tbmahasiswa;
Hasil	<pre> MariaDB [UjiKompetensi2]> Select nim AS 'NIM', -> NamaMahasiswa AS 'NAMA MAHASISWA', -> 'MEMILIKI GAJI ORANG TUA PERBULAN' AS 'MEMILIKI GAJI ORANG TUA PERBULAN', -> GajiOrtu AS 'GAJI BULANAN ORANG TUA' From tbmahasiswa; </pre> <pre> +-----+-----+-----+-----+ NIM NAMA MAHASISWA MEMILIKI GAJI ORANG TUA PERBULAN GAJI BULANAN ORANG TUA +-----+-----+-----+-----+ 2302001 Arif Silaban MEMILIKI GAJI ORANG TUA PERBULAN 5000000 2302002 Diaz Al Latif MEMILIKI GAJI ORANG TUA PERBULAN 6000000 2302003 Rafael Sinaga MEMILIKI GAJI ORANG TUA PERBULAN 6500000 2302004 Dea Cindia MEMILIKI GAJI ORANG TUA PERBULAN 6800000 2302005 Azura Intan MEMILIKI GAJI ORANG TUA PERBULAN 7000000 2302006 Febi Sofii MEMILIKI GAJI ORANG TUA PERBULAN 5500000 2302007 Ondo Saut MEMILIKI GAJI ORANG TUA PERBULAN 5700000 +-----+-----+-----+-----+ 7 rows in set (0.001 sec) </pre>

11. Tampilkan data dibawah ini dengan Fungsi Case When

Nim	NamaMahasiswa	UTS	NilaiHuruf
2302001	ABDULLAH MUJADID AL QAWI	72	B
2302002	ADINDA NABILA	90	A
2302005	ANGGUN BUNGA LESTARI LUBIS	80	A
2302006	ANNISA TRINABILA	72	B
2302008	BAGAS TRY ATMAJA	91	A
2302009	BUNGA SABILA	75	B
2302364	ALMAH WIJAYA ALHAKIM	90	A

Syntax :	Select Nim AS 'NIM', NamaMahasiswa AS 'NAMA MAHASISWA', UTS,CASE WHEN UTS >= 85 THEN 'A' WHEN UTS >= 70 THEN 'B' WHEN UTS >= 60 THEN 'C' ELSE 'D' END AS 'Nilai Huruf' from tbmahasiswa;																																
Penjelasan :	Menampilkan Data Nim, NamaMahasiswa, UTS dengan nama Alias dan UTS yang menggunakan case untuk beberapa kasus sebagai syarat Nilai memenuhi dan memberikan logika predikat untuk nilai yang memenuhi dan yang tidak																																
Hasil	<pre>MariaDB [UjiKompetensi2]> select Nim As 'NIM', -> NamaMahasiswa AS 'NAMA MAHASISWA', -> UTS, -> CASE -> WHEN UTS >= 85 THEN 'A' -> WHEN UTS >= 70 Then 'B' -> WHEN UTS >= 60 THEN 'C' -> ELSE 'D' -> END AS 'Nilai Huruf' from tbmahasiswa;</pre> <table><tr><th>NIM</th><th>NAMA MAHASISWA</th><th>UTS</th><th>Nilai Huruf</th></tr><tr><td>2302001</td><td>Arif Silaban</td><td>100</td><td>A</td></tr><tr><td>2302002</td><td>Diaz Al Latif</td><td>95</td><td>A</td></tr><tr><td>2302003</td><td>Rafael Sinaga</td><td>90</td><td>A</td></tr><tr><td>2302004</td><td>Dea Cindia</td><td>85</td><td>A</td></tr><tr><td>2302005</td><td>Azura Intan</td><td>80</td><td>B</td></tr><tr><td>2302006</td><td>Febi Sofii</td><td>87</td><td>A</td></tr><tr><td>2302007</td><td>Ondo Saut</td><td>91</td><td>A</td></tr></table> <pre>7 rows in set (0.011 sec)</pre>	NIM	NAMA MAHASISWA	UTS	Nilai Huruf	2302001	Arif Silaban	100	A	2302002	Diaz Al Latif	95	A	2302003	Rafael Sinaga	90	A	2302004	Dea Cindia	85	A	2302005	Azura Intan	80	B	2302006	Febi Sofii	87	A	2302007	Ondo Saut	91	A
NIM	NAMA MAHASISWA	UTS	Nilai Huruf																														
2302001	Arif Silaban	100	A																														
2302002	Diaz Al Latif	95	A																														
2302003	Rafael Sinaga	90	A																														
2302004	Dea Cindia	85	A																														
2302005	Azura Intan	80	B																														
2302006	Febi Sofii	87	A																														
2302007	Ondo Saut	91	A																														

➤ **Sub CMK(3) . Selelet With Filter Where, Between, Like, In, AND, OR,ORDER BY**

12. Tampilkan Semua data TbMahasiswa Yang memiliki NamaMahasiswa Huruf 'A' tetapi Tidak menampilkan NamaMahasiswa 'AN'

Syntax :	Select * from tbmahasiswa where NamaMahasiswa LIKE 'A%' AND NamaMahasiswa NOT LIKE 'AN%';																											
Penjelasan :	Menampilkan data NamaMahasiswa yang dimana NamaMahasiswa memiliki Awalan huruf A dan bukan AN																											
Hasil	<pre>MariaDB [UjiKompetensi2]> Select * from tbmahasiswa Where NamaMahasiswa LIKE 'A%' AND NamaMahasiswa NOT LIKE 'AN%';</pre><table><tr><th>Nim</th><th>NamaMahasiswa</th><th>AGAMA</th><th>JenisKelamin</th><th>TempatLahir</th><th>TempatTinggal</th><th>TglLahir</th><th>UTS</th><th>GajiOrtu</th></tr><tr><td>2302001</td><td>Arif Silaban</td><td>Kristen</td><td>laki-laki</td><td>Pematangsiantar</td><td>Pematangsiantar</td><td>2005-11-15</td><td>100</td><td>5000000</td></tr><tr><td>2302005</td><td>Azura Intan</td><td>Islam</td><td>Perempuan</td><td>Serapuh</td><td>Serapuh</td><td>2006-07-16</td><td>80</td><td>7000000</td></tr></table><pre>2 rows in set (0.006 sec)</pre>	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu	2302001	Arif Silaban	Kristen	laki-laki	Pematangsiantar	Pematangsiantar	2005-11-15	100	5000000	2302005	Azura Intan	Islam	Perempuan	Serapuh	Serapuh	2006-07-16	80	7000000
Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu																				
2302001	Arif Silaban	Kristen	laki-laki	Pematangsiantar	Pematangsiantar	2005-11-15	100	5000000																				
2302005	Azura Intan	Islam	Perempuan	Serapuh	Serapuh	2006-07-16	80	7000000																				

13. Tampilkan Semua data TbMahasiswa Yang memiliki NamaMahasiswa Huruf ke 4 nya adalah 'A' huruf ke 1,2,3 bebas dan huruf terakhir Bebas

Syntax :	select * from tbmahasiswa where NamaMahasiswa LIKE '___A%'																		
Penjelasan :	Menampilkan data NamaMahasiswa yang dimana huruf ke 4 nya adalah huruf A																		
Hasil	<pre>MariaDB [UjiKompetensi2]> select * from tbmahasiswa where NamaMahasiswa LIKE '___A%';</pre> <table><tr><th>Nim</th><th>NamaMahasiswa</th><th>AGAMA</th><th>JenisKelamin</th><th>TempatLahir</th><th>TempatTinggal</th><th>TglLahir</th><th>UTS</th><th>GajiOrtu</th></tr><tr><td>2302003</td><td>Rafael Sinaga</td><td>Kristen</td><td>laki-laki</td><td>Jakarta</td><td>Simbolon</td><td>2006-04-19</td><td>90</td><td>6500000</td></tr></table> <pre>1 row in set (0.001 sec)</pre>	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu	2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000
Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu											
2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000											

14. Tampilkan Semua data TbMahasiswa Yang memiliki NamaMahasiswa Huruf Terakhirnya adalah 'A' yang lain bebas

Syntax :	Select * from tbmahasiswa Where NamaMahasiswa LIKE '%A'																											
Penjelasan :	Menampilkan data NamaMahasiswa yang dimana huruf terakhirnya nya adalah huruf A																											
Hasil	<pre>MariaDB [UjiKompetensi2]> select * from tbmahasiswa where NamaMahasiswa LIKE '%A';</pre> <table><tr><th>Nim</th><th>NamaMahasiswa</th><th>AGAMA</th><th>JenisKelamin</th><th>TempatLahir</th><th>TempatTinggal</th><th>TglLahir</th><th>UTS</th><th>GajiOrtu</th></tr><tr><td>2302003</td><td>Rafael Sinaga</td><td>Kristen</td><td>laki-laki</td><td>Jakarta</td><td>Simbolon</td><td>2006-04-19</td><td>90</td><td>6500000</td></tr><tr><td>2302004</td><td>Dea Cindia</td><td>Islam</td><td>Perempuan</td><td>Papua Barat</td><td>P.Siantar</td><td>2005-12-16</td><td>85</td><td>6800000</td></tr></table> <pre>2 rows in set (0.060 sec)</pre>	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu	2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000	2302004	Dea Cindia	Islam	Perempuan	Papua Barat	P.Siantar	2005-12-16	85	6800000
Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu																				
2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000																				
2302004	Dea Cindia	Islam	Perempuan	Papua Barat	P.Siantar	2005-12-16	85	6800000																				

15. Tampilkan Semua data TbMahasiswa Yang memiliki NamaMahasiswa Huruf Awalnya adalah 'B' yang lain bebas

Syntax :	<code>select * from tbmahasiswa where NamaMahasiswa LIKE 'B%'</code>
Penjelasan :	Menampilkan data NamaMahasiswa yang dimana huruf awal nya adalah huruf B
Hasil	<pre>MariaDB [UjiKompetensi2]> select * from tbmahasiswa where NamaMahasiswa LIKE 'B%';</pre> <pre>Empty set (0.001 sec)</pre> (Perintah ini berhasil, kebetulan data yang diperoleh itu tidak ada yang nama pada NamaMahasiswa nya berawalan huruf B)

16. Tampilkan Semua data TbMahasiswa Yang memiliki Nilai UTS diantara 85 sd 95 untuk jenis_kelamin adalah Perempuan. (dengan Perintah Between).

Syntax :	<code>select * from tbmahasiswa where UTS BETWEEN 85 AND 95 AND JenisKelamin = 'Perempuan';</code>
-----------------	--

Penjelasan :	Menampilkan tabel data tbmahasiswa yang dimana Nilai UTS nya diantara 85 dan 95 dan berjenis Kelamin Perempuan								
Hasil	MariaDB [UjiKompetensi2]> Select * from tbmahasiswa where UTS BETWEEN 85 AND 95 AND JenisKelamin = 'Perempuan';								
	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TgLLahir	UTS	GajiOrtu
	2302004	Dea Cindia	Islam	Perempuan	Papua Barat	P. Siantar	2005-12-16	85	6800000
	2302006	Febi Sofii	Islam	Perempuan	Rambung Merah	Rambung Merah	2006-02-06	87	5500000
	2 rows in set (0.011 sec)								

17. Tampilkan Semua data TbMahasiswa Yang memiliki Nilai UTS diantara 70 sd 80 untuk jenis_kelamin adalah Perempuan. (dengan Perintah AND).

Syntax :	Select * from tbmahasiswa where UTS >= 70 AND <= 80 AND Jeniskelamin = 'Perempuan';																		
Penjelasan :	Menampilkan data pada tabel Mahasiswa yang Nilai UTS nya lebih dari sama dengan 70 atau kurang dari sama dengan 80 dan berjenis kelamin perempuan																		
Hasil	<pre>MariaDB [UjiKompetensi2]> select * from tbmahasiswa Where UTS >=70 AND UTS <=80 AND JenisKelamin = 'Perempuan';</pre> <table><tr><th>Nim</th><th>NamaMahasiswa</th><th>AGAMA</th><th>JenisKelamin</th><th>TempatLahir</th><th>TempatTinggal</th><th>TglLahir</th><th>UTS</th><th>GajiOrtu</th></tr><tr><td>2302005</td><td>Azura Intan</td><td>Islam</td><td>Perempuan</td><td>Serapuh</td><td>Serapuh</td><td>2006-07-16</td><td>80</td><td>7000000</td></tr></table> <pre>1 row in set (0.005 sec)</pre>	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu	2302005	Azura Intan	Islam	Perempuan	Serapuh	Serapuh	2006-07-16	80	7000000
Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu											
2302005	Azura Intan	Islam	Perempuan	Serapuh	Serapuh	2006-07-16	80	7000000											

18. Tampilkan Semua data TbMahasiswa Yang memiliki Tempat_Tinggal di Rambung Merah dan Bah Jambi menggunakan IN

Syntax :	Select * from tbmahasiswa where TempatTinggal IN ('Rambung Merah','Bah Jambi');																											
Penjelasan :	Menampilkan data pada tabel Mahasiswa yang dimana Tempat tinggalnya di Rambung Merah dan di Bah Jambi																											
Hasil	<pre>MariaDB [UjiKompetensi2]> Select * from tbmahasiswa Where TempatTinggal IN ('Rambung Merah','Bah Jambi');</pre><table><tr><th>Nim</th><th>NamaMahasiswa</th><th>AGAMA</th><th>JenisKelamin</th><th>TempatLahir</th><th>TempatTinggal</th><th>TglLahir</th><th>UTS</th><th>GajiOrtu</th></tr><tr><td>2302002</td><td>Diaz Al Latif</td><td>Islam</td><td>laki-laki</td><td>Tanah Jawa</td><td>Bah Jambi</td><td>2006-07-17</td><td>95</td><td>6000000</td></tr><tr><td>2302006</td><td>Febi Sofii</td><td>Islam</td><td>Perempuan</td><td>Rambung Merah</td><td>Rambung Merah</td><td>2006-02-06</td><td>87</td><td>5500000</td></tr></table><pre>2 rows in set (0.001 sec)</pre>	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu	2302002	Diaz Al Latif	Islam	laki-laki	Tanah Jawa	Bah Jambi	2006-07-17	95	6000000	2302006	Febi Sofii	Islam	Perempuan	Rambung Merah	Rambung Merah	2006-02-06	87	5500000
Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu																				
2302002	Diaz Al Latif	Islam	laki-laki	Tanah Jawa	Bah Jambi	2006-07-17	95	6000000																				
2302006	Febi Sofii	Islam	Perempuan	Rambung Merah	Rambung Merah	2006-02-06	87	5500000																				

19. Tampilkan Semua data TbMahasiswa Yang memiliki Tempat_Tinggal di P.Siantar dan Simbolon menggunakan OR

Syntax :	Select * from tbmahasiswa where TempatTinggal = 'P.Siantar' OR TempatTinggal = 'Simbolon';																											
Penjelasan :	Menampilkan data pada tabel Mahasiswa yang dimana Tempat tinggalnya di P.Siantar atau di Simbolon																											
Hasil	MariaDB [UjiKompetensi2]> Select * from tbmahasiswa Where TempatTinggal ='P.Siantar' OR TempatTinggal = 'Simbolon'; <table><tr><th>Nim</th><th>NamaMahasiswa</th><th>AGAMA</th><th>JenisKelamin</th><th>TempatLahir</th><th>TempatTinggal</th><th>TglLahir</th><th>UTS</th><th>GajiOrtu</th></tr><tr><td>2302003</td><td>Rafael Sinaga</td><td>Kristen</td><td>laki-laki</td><td>Jakarta</td><td>Simbolon</td><td>2006-04-19</td><td>90</td><td>6500000</td></tr><tr><td>2302004</td><td>Dea Cindia</td><td>Islam</td><td>Perempuan</td><td>Papua Barat</td><td>P. Siantar</td><td>2005-12-16</td><td>85</td><td>6800000</td></tr></table> 2 rows in set (0.001 sec)	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu	2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000	2302004	Dea Cindia	Islam	Perempuan	Papua Barat	P. Siantar	2005-12-16	85	6800000
Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu																				
2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000																				
2302004	Dea Cindia	Islam	Perempuan	Papua Barat	P. Siantar	2005-12-16	85	6800000																				

20. Tampilkan Semua data TbMahasiswa Yang memiliki Namamahasiswa dimana Huruf ke 2 adalah U dan Huruf ke 4 adalag G yang lain Bebas.

Syntax :	Select * from tbmahasiswa where TempatTinggal = 'P.Siantar' OR TempatTinggal = 'Simbolon';
Penjelasan :	Menampilkan data tabel tbmahasiswa yang kolom NamaMahasiswa nya memiliki Huruf ke 2 adalah U dan huruf ke 4 adalah G
ilHasil	<pre>MariaDB [UjiKompetensi2]> select * from tbmahasiswa Where NamaMahasiswa LIKE '_U_G%';</pre> Empty set (0.001 sec) (Perintah ini berhasil, kebetulan data yang diperoleh itu tidak ada yang nama pada NamaMahasiswa nya tidak ada yang berhuruf ke 2 U dan ke 4 G)

21. Tampilkan Semua data TbMahasiswa Yang memiliki Jenis_kelamin adalah Tidak Perempuan.

Syntax :	select * from tbmahasiswa where JenisKelamin <> Perempuan;								
Penjelasan :	Menampilkan data pada tabel Mahasiswa yang dimana Jenis Kelaminnya tidak perempuan								
Hasil	MariaDB [UjiKompetensi2]> select * from tbmahasiswa where JenisKelamin <> 'Perempuan';								
	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu
	2302001	Arif Silaban	Kristen	laki-laki	Pematangsiantar	Pematangsiantar	2005-11-15	100	5000000
	2302002	Diaz Al Latif	Islam	laki-laki	Tanah Jawa	Bah Jambi	2006-07-17	95	6000000
	2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000
	2302007	Ondo Saut	Kristen	laki-laki	Balimbingan	Balimbingan	2006-05-12	91	5700000
	4 rows in set (0.001 sec)								

22. Tampilkan Semua data TbMahasiswa Yang DIURUTKAN BERDASARKAN NAMA MAHASISWA Berdasarkan Descending.

Syntax :	Select * from tbmahasiswa ORDER BY NamaMahasiswa DESC;								
Penjelasan :	Menampilkan tabel data tbmahasiswa pada kolom berdasarkan DESC dan DESC yang dimaksud pada NamaMahasiswa berarti Huruf Abjad tertinggi								
Hasil	MariaDB [UjiKompetensi2]> select * from tbmahasiswa ORDER BY NamaMahasiswa DESC;								
	Nim	NamaMahasiswa	AGAMA	JenisKelamin	TempatLahir	TempatTinggal	TglLahir	UTS	GajiOrtu
	2302003	Rafael Sinaga	Kristen	laki-laki	Jakarta	Simbolon	2006-04-19	90	6500000
	2302007	Ondo Saut	Kristen	laki-laki	Balimbingan	Balimbingan	2006-05-12	91	5700000
	2302006	Febi Sofii	Islam	Perempuan	Rambung Merah	Rambung Merah	2006-02-06	87	5500000
	2302002	Diaz Al Latif	Islam	laki-laki	Tanah Jawa	Bah Jambi	2006-07-17	95	6000000
	2302004	Dea Cindia	Islam	Perempuan	Papua Barat	P. Siantar	2005-12-16	85	6800000
	2302005	Azura Intan	Islam	Perempuan	Serapuh	Serapuh	2006-07-16	80	7000000
	2302001	Arif Silaban	Kristen	laki-laki	Pematangsiantar	Pematangsiantar	2005-11-15	100	5000000
	7 rows in set (0.004 sec)								

➤ Sub CMK(4). DML Database : AGGREGATE, Group By

23. Tampilkan Rata-rata Gaji_Ortu dari Table TbMahasiswa

Syntax :	Select AVG(GajiOrtu) AS 'Rata-Rata Gaji Ortu' from tbmahasiswa;
Penjelasan :	Menampilkan rata-rata gaji ortu dari tabel tbmahasiswa

Hasil	<pre> MariaDB [UjiKompetensi2]> select AVG(GajiOrtu) AS 'Rata-Rata Gaji Ortu' from tbmahasiswa; +-----+ Rata-Rata Gaji Ortu +-----+ 6071428.5714 +-----+ 1 row in set (0.010 sec) </pre>
--------------	--

24. Tampilkan Semua data TbMahasiswa Yang memiliki GAJI_ORTU Diatas Rata-Rata

Syntax :	<code>select * from tbmahasiswa where GajiOrtu > (select AVG(GajiOrtu) from tbmahasiswa);</code>
Penjelasan :	Menampilkan semua data tabel tbmahasiswa yang memiliki Gaji Ortu diatas rata-rata
Hasil	<pre> MariaDB [UjiKompetensi2]> Select * from tbmahasiswa Where GajiOrtu > (Select AVG(GajiOrtu) From tbmahasiswa); +-----+-----+-----+-----+-----+-----+-----+-----+-----+ Nim NamaMahasiswa AGAMA JenisKelamin TempatLahir TempatTinggal TglLahir UTS GajiOrtu +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 2302003 Rafael Sinaga Kristen laki-laki Jakarta Simbolon 2006-04-19 90 6500000 2302004 Dea Cindia Islam Perempuan Papua Barat P.Siantar 2005-12-16 85 6800000 2302005 Azura Intan Islam Perempuan Serapuh Serapuh 2006-07-16 80 7000000 +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 3 rows in set (0.005 sec) </pre>

25. Tampilkan Semua data TbMahasiswa Yang memiliki GAJI_ORTU Dibawah Rata-Rata

Syntax :	<code>select * from tbmahasiswa where GajiOrtu < (select AVG(GajiOrtu) from tbmahasiswa);</code>
Penjelasan :	Menampilkan semua data tabel tbmahasiswa yang memiliki Gaji Ortu dibawah rata-rata
Hasil	<pre> MariaDB [UjiKompetensi2]> Select * from tbmahasiswa Where GajiOrtu < (Select AVG(GajiOrtu) From tbmahasiswa); +-----+-----+-----+-----+-----+-----+-----+-----+-----+ Nim NamaMahasiswa AGAMA JenisKelamin TempatLahir TempatTinggal TglLahir UTS GajiOrtu +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 2302001 Arif Silaban Kristen laki-laki Pematangsiantar Pematangsiantar 2005-11-15 100 5000000 2302002 Diaz Al Latif Islam laki-laki Tanah Jawa Bah Jambi 2006-07-17 95 6000000 2302006 Febi Sofii Islam Perempuan Rambun Merah Rambun Merah 2006-02-06 87 5500000 2302007 Ondo Saut Kristen laki-laki Balimbingan Balimbingan 2006-05-12 91 5700000 +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 4 rows in set (0.001 sec) </pre>

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

Syntax :	<code>select * from tbmahasiswa where UTS = (select MAX(UTS) from tbmahasiswa);</code>
Penjelasan :	Menampilkan semua data tabel tbmahasiswa yang memiliki nilai UTS Tertinggi
Hasil	<pre> MariaDB [UjiKompetensi2]> Select * from tbmahasiswa Where UTS = (select MAX(UTS) from tbmahasiswa); +-----+-----+-----+-----+-----+-----+-----+-----+-----+ Nim NamaMahasiswa AGAMA JenisKelamin TempatLahir TempatTinggal TglLahir UTS GajiOrtu +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 2302001 Arif Silaban Kristen laki-laki Pematangsiantar Pematangsiantar 2005-11-15 100 5000000 +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 1 row in set (0.007 sec) </pre>

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

Syntax :	<code>select * from tbmahasiswa where UTS = (select MIN(UTS) from tbmahasiswa);</code>
Penjelasan :	Menampilkan semua data tabel tbmahasiswa yang memiliki nilai UTS Terendah
Hasil	<pre> MariaDB [UjiKompetensi2]> Select * from tbmahasiswa Where UTS = (select MIN(UTS) from tbmahasiswa); +-----+-----+-----+-----+-----+-----+-----+-----+-----+ Nim NamaMahasiswa AGAMA JenisKelamin TempatLahir TempatTinggal TglLahir UTS GajiOrtu +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 2302005 Azura Intan Islam Perempuan Serapuh Serapuh 2006-07-16 80 7000000 +-----+-----+-----+-----+-----+-----+-----+-----+-----+ 1 row in set (0.001 sec) </pre>

28.

```
Tampilkan data Sebagai Berikut :
+-----+-----+
| Total Gaji Orang Tua Kelas 23S01 adalah Rp. | SUM(GAJI_ORTU) |
+-----+-----+
| Total Gaji Orang Tua Kelas 23S01 adalah Rp. |      37700000 |
+-----+-----+
```

Syntax :	Select 'Total Gaji Orang Tua Kelas 23S04 Adalah' AS Keterangan, → SUM(GajiOrtu) AS 'Jumlah Semua Gaji Ortu' from tbmahasiswa Where Nim LIKE '23%';
Penjelasan :	Menampilkan data Total Gaji Orang Tua Kelas 23S04 menggunakan Select SUM
Hasil	<pre>MariaDB [UjiKompetensi2]> Select 'Total Gaji Orang Tua Kelas 23S04 Adalah' AS Keterangan -> SUM(GajiOrtu) AS 'Jumlah Semua Gaji Ortu' from tbmahasiswa where Nim LIKE '23%' +-----+-----+ Keterangan Jumlah Semua Gaji Ortu +-----+-----+ Total Gaji Orang Tua Kelas 23S04 Adalah 42500000 +-----+-----+ 1 row in set (0.012 sec)</pre>

29. Tampilkan Jumlah data dari Tabel TbMahasiswa Yang memiliki JENIS_KELAMIN adalah PEREMPUAN

Syntax :	Select Count(*) AS 'Jumlah Data Jenis kelamin Perempuan' from tbmahasiswa where Jeniskelamin = 'Perempuan';
Penjelasan :	Menampilkan Jumlah data tabel tbmahasiswa yang berjenis kelamin Perempuan
Hasil	<pre>MariaDB [UjiKompetensi2]> select Count(*) AS 'Jumlah Data Jenis Kelamin Perempuan' from tbmahasiswa Where Jeniskelamin = 'Perempuan'; +-----+ Jumlah Data Jenis Kelamin Perempuan +-----+ 3 +-----+ 1 row in set (0.001 sec)</pre>

30. Tampilkan Semuan Data dari Tabel TbMahasiswa Yang memiliki JENIS_KELAMIN adalah LAKI-LAKI

Syntax :	Select Count(*) AS 'Jumlah Data Jenis kelamin Laki-Laki' from tbmahasiswa where Jeniskelamin = 'laki-laki';
Penjelasan :	Menampilkan Jumlah data tabel tbmahasiswa yang berjenis kelamin Laki-laki
Hasil	<pre>MariaDB [UjiKompetensi2]> Select Count(*) AS 'Jumlah Data Jenis Kelamin Laki-Laki' from tbmahasiswa where Jeniskelamin = 'laki-laki'; +-----+ Jumlah Data Jenis Kelamin Laki-Laki +-----+ 4 +-----+ 1 row in set (0.001 sec)</pre>

31. Tampilkan Berdasarkan Group Data Nilai rata-Rata UTS berdasarkan Jenis_kelamin

Syntax :	Select Jeniskelamin, AVG(UTS) AS 'Rata-Rata Nilai UTS' from tbmahasiswa GROUP BY Jeniskelamin;
Penjelasan :	Menampilkan Berdasarkan Group Data Nilai rata-rata UTS berdasarkan Jenis_Kelamin

Hasil	<pre>MariaDB [UjiKompetensi2]> Select JenisKelamin, AVG(UTS) AS 'Rata-Rata Nilai UTS' from tbmahasiswa GROUP BY JenisKelamin;</pre>
	<pre> +-----+-----+ JenisKelamin Rata-Rata Nilai UTS +-----+-----+ laki-laki 94.0000 Perempuan 84.0000 +-----+-----+ 2 rows in set (0.007 sec)</pre>

32. Tampilkan Berdasarkan Group Data Rata-rata Gaji Orang Tua berdasarkan Tempat_Tinggal

Syntax :	<code>Select TempatTinggal, AVG(GajiOrtu) AS 'Total Gaji Orang Tua' from tbmahasiswa GROUP BY GajiOrtu;</code>
Penjelasan :	Menampilkan Berdasarkan Group Data rata-rata Gaji Ortu berdasarkan TempatTinggal
Hasil	<pre>MariaDB [UjiKompetensi2]> Select TempatTinggal, AVG(GajiOrtu) AS 'Total Gaji Orang Tua' from tbmahasiswa GROUP BY GajiOrtu;</pre> <pre> +-----+-----+ TempatTinggal Total Gaji Orang Tua +-----+-----+ Pematangsiantar 5000000.0000 Rambung Merah 5500000.0000 Balimbingan 5700000.0000 Bah Jambi 6000000.0000 Simbolon 6500000.0000 P.Siantar 6800000.0000 Serapuh 7000000.0000 +-----+-----+ 7 rows in set (0.001 sec)</pre>

33. Tampilkan Berdasarkan Group Data Rata-Rata Gaji Orang Tua berdasarkan Tempat_Tinggal, dimana Gaji Orang Tua >=500000

Syntax :	<code>Select TempatTinggal, AVG(GajiOrtu) AS 'Total Gaji Orang Tua' from tbmahasiswa GROUP BY TempatTinggal HAVING AVG(gajiOrtu) >= 500000;</code>
Penjelasan :	Menampilkan data table tbmahasiswa berdasarkan Group Data Rata-rata Gaji Orang Tua Berdasarkan tempat tinggal dimana Gaji Orang Tua >= 500000
Hasil	<pre>MariaDB [UjiKompetensi2]> Select TempatTinggal, AVG(GajiOrtu) AS 'Rata-Rata Gaji Ortu' from tbmahasiswa GROUP BY TempatTinggal HAVING AVG(GajiOrtu) >= 5000000;</pre> <pre> +-----+-----+ TempatTinggal Rata-Rata Gaji Ortu +-----+-----+ Bah Jambi 6000000.0000 Balimbingan 5700000.0000 P.Siantar 6800000.0000 Pematangsiantar 5000000.0000 Rambung Merah 5500000.0000 Serapuh 7000000.0000 Simbolon 6500000.0000 +-----+-----+ 7 rows in set (0.001 sec)</pre>

34. Tampilkan Berdasarkan Group Data Jumlah Total Gaji Orang Tua berdasarkan Jenis_Kelamin

Syntax :	<code>Select JenisKelamin, SUM(GajiOrtu) AS 'Total Gaji Orang Tua' from tbmahasiswa GROUP BY JenisKelamin;</code>
Penjelasan :	Menampilkan data tabel tbmahasiswa Berdasarkan Group Data Jumlah Total Gaji Orang Tua berdasarkan JenisKelamin;
Hasil	<pre>MariaDB [UjiKompetensi2]> Select JenisKelamin, SUM(GajiOrtu) AS 'Total Gaji Orang Tua' from tbmahasiswa GROUP BY JenisKelamin;</pre> <pre> +-----+-----+ JenisKelamin Total Gaji Orang Tua +-----+-----+ laki-laki 23200000 Perempuan 19300000 +-----+-----+</pre>

➤ Sub CMK(5) . VIEW

35. Buat View dengan nama ManusiaSiantar dengan menampilkan Semua Data TbMahasiswa dengan Tempat Tinggal di P.Siantar

Syntax :	Create view ManusiaSiantar As select * from tbmahasiswa where TempatTinggal = 'P.Siantar';
Penjelasan :	Membuat View dengan nama ManusiaSiantar dengan menampilkan semua data TbMahasiswa dengan TempatTinggal di P.Siantar
Hasil	MariaDB [UjiKompetensi2]> create view ManusiaSiantar As Select * from tbmahasiswa Where TempatTinggal = 'P.Siantar'; Query OK, 0 rows affected (0.011 sec)

36. Buat Perintah Menampilkan View di Database

Syntax :	Show Full Tables in UjiKompetensi2 Where Table_type LIKE 'VIEW';
Penjelasan :	Menampilkan View di Database
Hasil	MariaDB [UjiKompetensi2]> Show Full Tables in UjiKompetensi2 Where Table_type Like 'VIEW'; +-----+-----+ Tables_in_ujikompetensi2 Table_type +-----+-----+ manusiasiantar VIEW penghunisiantar VIEW +-----+-----+ 2 rows in set (0.006 sec)

37. Buat Perintah Menampilkan View ManusiaSiantar

Syntax :	select * from ManusiaSiantar
Penjelasan :	Menampilkan View dari ManusiaSiantar
Hasil	MariaDB [UjiKompetensi2]> select * from manusiasiantar; +-----+-----+-----+-----+-----+-----+-----+-----+ Nim NamaMahasiswa AGAMA JenisKelamin TempatLahir TempatTinggal TglLahir UTS GajiOrtu +-----+-----+-----+-----+-----+-----+-----+-----+ 2302004 Dea Cindia Islam Perempuan Papua Barat P.Siantar 2005-12-16 85 6000000 +-----+-----+-----+-----+-----+-----+-----+-----+ 1 row in set (0.002 sec)

BAB V

STORED PROCEDURE

Stored Procedure adalah sekumpulan perintah SQL yang disimpan di dalam database dan dapat dijalankan secara berulang-ulang. Stored Procedure dapat menerima parameter, memproses logika tertentu, dan menghasilkan output. Dalam konteks Database Terdistribusi, stored procedure sangat berguna karena:

- Mengurangi lalu lintas jaringan (karena logika dijalankan di sisi server).
- Meningkatkan performa dan konsistensi data.
- Menyediakan kontrol akses yang lebih baik terhadap data.
- Mempermudah integrasi antar node atau server database yang tersebar.

A. Membuat dan Menjalankan Stored Procedure (Praktikum 5)

1. Membuat Database dbprocedure;

Syntax :	CREATE DATABASE dbprocedure; USE dbprocedure;
Penjelasan :	Membuat Database dan masuk ke dalam database dbprocedure
Hasil	<pre>MariaDB [(none)]> create database dbprocedure; Query OK, 1 row affected (0.003 sec) MariaDB [(none)]> use dbprocedure; Database changed</pre>

2. Membuat Tabel Barang

Syntax :	Create table barang(-> KdBarang Varchar(5) PRIMARY KEY, -> NamaBarang Varchar(50));
Penjelasan :	Membuat Tabel Barang
Hasil	<pre>MariaDB [dbprocedure]> create table barang(-> KdBarang Varchar(5) PRIMARY KEY, -> NamaBarang Varchar(50)); Query OK, 0 rows affected (0.028 sec)</pre>

	<pre> MariaDB [dbprocedure]> alter table barang add QtyBarang int; Query OK, 0 rows affected (0.012 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [dbprocedure]> alter table barang add UnitBarang Varchar(10); Query OK, 0 rows affected (0.013 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [dbprocedure]> alter table barang add HargaBarang DECIMAL(10,3); Query OK, 0 rows affected (0.017 sec) Records: 0 Duplicates: 0 Warnings: 0 MariaDB [dbprocedure]> desc barang; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ KdBarang varchar(5) NO PRI NULL NamaBarang varchar(50) YES NULL QtyBarang int(11) YES NULL UnitBarang varchar(10) YES NULL HargaBarang decimal(10,3) YES NULL +-----+-----+-----+-----+-----+-----+ 5 rows in set (0.034 sec) </pre>
--	--

3. Menambahkan record ke dalam tabel

Syntax :	<pre> Insert Into Barang -> VALUES ('CK001','Antarestar Backpack 65L','12','DUS','499'), -> ('CK002','SALOMON Hiking Shoes','20','BOX','899'); </pre>
Penjelasan:	Menambahkan Record ke dalam tabel
Hasil	<pre> MariaDB [dbprocedure]> insert into barang -> VALUES ('CK001','Antarestar Backpack 65L','12','DUS','499'), -> ('CK002','SALOMON Hiking Shoes','20','BOX','899'); Query OK, 2 rows affected (0.010 sec) Records: 2 Duplicates: 0 Warnings: 0 </pre> Isi tabel setelah diisi: <pre> MariaDB [dbprocedure]> select * from barang; +-----+-----+-----+-----+-----+ KdBarang NamaBarang QtyBarang UnitBarang HargaBarang +-----+-----+-----+-----+-----+ CK001 Antarestar Backpack 65L 12 DUS 499.000 CK002 SALOMON Hiking Shoes 20 BOX 899.000 +-----+-----+-----+-----+-----+ 2 rows in set (0.001 sec) </pre>

4. Membuat Procedure

Syntax :	<pre> DELIMITER \$\$ Create PROCEDURE NamaProsedur() BEGIN SELECT NamaField from>NamaTabel; END \$\$ </pre>
Penjelasan:	DELIMITER digunakan supaya SQL tahu bahwa tanda ; di dalam blok procedure bukan akhir perintah. Setelah selesai, DELIMITER ; dikembalikan lagi ke default.
Hasil	<pre> MariaDB [dbprocedure]> DELIMITER \$\$ MariaDB [dbprocedure]> Create Procedure SelectEquip() -> BEGIN -> Select KdBarang,>NamaBarang,>NamaBarang from barang; -> END\$\$ Query OK, 0 rows affected (0.014 sec) </pre>

	MariaDB [dbprocedure]> DELIMITER ;
--	------------------------------------

5. Memanggil Procedure

Syntax :	Call NamaProsedur();
Penjelasan:	Memanggil Procedure yang sudah dibuat sebelumnya, Perintah prosedur ini tidak terlepas dari perintah SQL seperti select, delete, update, insert, dan lain-lain.
Hasil	<pre>MariaDB [dbprocedure]> Call SelectEquip(); +-----+-----+-----+ KdBarang NamaBarang HargaBarang +-----+-----+-----+ CK001 Antarestar Backpack 65L 499.000 CK002 SALOMON Hiking Shoes 899.000 +-----+-----+-----+ 2 rows in set (0.001 sec)</pre>

6. Membuat Procedure Menambahkan Barang

Syntax :	<pre>DELIMITER \$\$ Create Procedure InsertBarang(-> IN variabel1 [TipeData(Size)], -> IN variabel2 [TipeData(Size)], -> IN variabeln [TipeData(Size)]) -> BEGIN -> Insert Into (kolom1, kolom2, kolomn...) -> VALUES (Variabel1, variabel2, variabeln...); -> END \$\$ Setelah Query OK DELIMITER ; Call NamaProcedure('Record1', 'Record2', 'Recordn...');</pre>
Penjelasan:	Pada perintah tersebut terlihat pada p_Kdbarang, p_NamaBarang, dll merupakan field baru sebagai variabel, kemudian pada Insert Into barang merupakan kolom yang akan menjadi target insert data saat kita melakukan call procedure sehingga saat melakukan call procedure data akan bertambah dengan pengisian sesuai dengan variable dan kolomnya.

Hasil	<pre> MariaDB [dbprocedure]> DELIMITER \$\$ MariaDB [dbprocedure]> Create Procedure InsertBarang(-> IN p_KdBarang Varchar(10), -> IN p_NamaBarang Varchar(50), -> IN p_QtyBarang int(11), -> IN p_UnitBarang Varchar(10), -> IN p_HargaBarang DECIMAL(10,3) ->) -> BEGIN -> INSERT INTO barang (KdBarang, NamaBarang, QtyBarang, UnitBarang, HargaBarang) -> VALUES (p_KdBarang, p_NamaBarang, p_QtyBarang, p_UnitBarang, p_HargaBarang); -> END \$\$ Query OK, 0 rows affected (0.015 sec) MariaDB [dbprocedure]> DELIMITER ; MariaDB [dbprocedure]> Call InsertBarang('CK003','Consina Sleeping Bag','6','DUS','149'); Query OK, 1 row affected (0.011 sec) MariaDB [dbprocedure]> Select * from barang; +-----+-----+-----+-----+-----+ KdBarang NamaBarang QtyBarang UnitBarang HargaBarang +-----+-----+-----+-----+-----+ CK001 Antarestar Backpack 65L 12 DUS 499.000 CK002 SALOMON Hiking Shoes 20 BOX 899.000 CK003 Consina Sleeping Bag 6 DUS 149.000 +-----+-----+-----+-----+-----+ 3 rows in set (0.001 sec) </pre> Data Sebelum Call Procedure: <pre> MariaDB [dbprocedure]> select * from barang; +-----+-----+-----+-----+-----+ KdBarang NamaBarang QtyBarang UnitBarang HargaBarang +-----+-----+-----+-----+-----+ CK001 Antarestar Backpack 65L 12 DUS 499.000 CK002 SALOMON Hiking Shoes 20 BOX 899.000 +-----+-----+-----+-----+-----+ 2 rows in set (0.001 sec) </pre>
---------------------	--

7. Membuat Procedure Pencarian Khusus

Syntax :	<pre> DELIMITER \$\$ Create Procedure InsertBarang(IN Variabel [TipeData(Size)]) -> BEGIN -> Select * from barang where kolom = Variabel; -> END \$\$ </pre> Setelah Query OK DELIMITER ; Call NamaProcedure('RecordPencarian');
Penjelasan:	Pada Perintah tersebut, terlihat bahwa Unit Varchar(20) sebagai variabel. Kemudian perintah select tersebut untuk menghubungkan Variabel unit ke kolom UnitBarang, sehingga hal tersebut membuat perintah ini seperti filter pencarian. Saat melakukan Call Procedure kita bisa menyisipkan Jenis unit barang apa yang mau kita cari ke dalam kurung yang ada dalam nama procedure tersebut dengan apostrof, apakah itu DUS apakah itu BOX, dll.

Hasil	<pre> MariaDB [dbprocedure]> DELIMITER \$\$ MariaDB [dbprocedure]> Create Procedure CariBarangku(IN Unit varchar(20)) -> BEGIN -> Select * from barang WHERE UnitBarang = Unit; -> END \$\$ Query OK, 0 rows affected (0.014 sec) MariaDB [dbprocedure]> DELIMITER ; MariaDB [dbprocedure]> Call CariBarangku('DUS'); +-----+-----+-----+-----+-----+ KdBarang NamaBarang QtyBarang UnitBarang HargaBarang +-----+-----+-----+-----+-----+ CK001 Antarestar Backpack 65L 12 DUS 499.000 +-----+-----+-----+-----+-----+ 1 row in set (0.001 sec) Query OK, 0 rows affected (0.006 sec) </pre>
-------	---

8. Membuat Procedure Hapus Data

Syntax :	<pre> DELIMITER \$\$ Create Procedure InsertBarang(-> IN variabel [TipeData(Size)]) -> BEGIN -> Delete from barang -> where kolom = variabel; -> END \$\$ Setelah Query OK DELIMITER ; Call NamaProcedure('RecordPencarian'); </pre>
Penjelasan:	Pada perintah tersebut, terlihat bahwa p_KdBarang Varchar(10) sebagai variabel. Kemudian perintah delete tersebut untuk menghubungkan Variabel unit ke kolom KdBarang, sehingga hal tersebut membuat perintah ini seperti filter penghapusan data. Saat melakukan Call Procedure kita bisa menyisipkan Kode barang apa yang mau kita hapus ke dalam kurung yang ada dalam nama procedure, sehingga perintah tersebut akan menghapus data berdasarkan kode barang yang kita masukkan.
Hasil	<pre> MariaDB [dbprocedure]> DELIMITER \$\$ MariaDB [dbprocedure]> Create Procedure HapusBarang(-> IN p_KdBarang Varchar(10)) -> BEGIN -> Delete from barang -> Where KdBarang = p_KdBarang; -> END \$\$ Query OK, 0 rows affected (0.011 sec) MariaDB [dbprocedure]> DELIMITER ; MariaDB [dbprocedure]> Call HapusBarang('CK002'); Query OK, 1 row affected (0.108 sec) MariaDB [dbprocedure]> select * from barang; +-----+-----+-----+-----+-----+ KdBarang NamaBarang QtyBarang UnitBarang HargaBarang +-----+-----+-----+-----+-----+ CK001 Antarestar Backpack 65L 12 DUS 499.000 CK003 Consina Sleeping Bag 6 DUS 149.000 +-----+-----+-----+-----+-----+ 2 rows in set (0.001 sec) </pre> Data Sebelum Dihapus:

	<pre> MariaDB [dbprocedure]> Select * from barang; +-----+-----+-----+-----+-----+ KdBarang NamaBarang QtyBarang UnitBarang HargaBarang +-----+-----+-----+-----+-----+ CK001 Antarestar Backpack 65L 12 DUS 499.000 CK002 SALOMON Hiking Shoes 20 BOX 899.000 CK003 Consina Sleeping Bag 6 DUS 149.000 +-----+-----+-----+-----+-----+ 3 rows in set (0.001 sec) </pre>
--	--

9. Membuat Procedure Mengganti Data(Harga)

Syntax :	<pre> DELIMITER \$\$ Create Procedure InsertBarang(-> IN Variabel [TipeData(Size)], -> IN Variabeln [TipeData(Size)]) -> BEGIN -> Update barang -> SET KolomPengganti = VariabelPengganti -> where KolomPatokan = VariabelPatokan; -> END \$\$ Setelah Query OK DELIMITER ; Call NamaProcedure('RecordPatokan', 'RecordPengganti'); </pre>
Penjelasan:	Pada Perintah tersebut, terlihat bahwa p_KdBarang Varchar(10), p_HargaBarang sebagai variabel. Kemudian perintah update tersebut untuk menghubungkan Variabel unit ke kolom KdBarang, dan HargaBarang. Sehingga hal tersebut membuat perintah ini seperti filter pengubahan data. Saat melakukan Call Procedure kita bisa menyisipkan Kode barang apa yang mau kita hapus ke dalam kurung yang ada dalam nama procedure, sehingga perintah tersebut akan mengganti data berdasarkan kode barang yang kita masukkan. Kode Barang tersebut menjadi patokan perintah yang akan dijalankan karena WHERE menggunakan variable dari kolom Kode Barang.
Hasil	<pre> MariaDB [dbprocedure]> DELIMITER \$\$ MariaDB [dbprocedure]> Create Procedure GantiHarga(-> IN p_KdBarang Varchar(10), -> IN p_HargaBarang DECIMAL(10,3)) -> BEGIN -> Update barang -> SET HargaBarang = p_HargaBarang -> Where KdBarang = p_KdBarang; -> END \$\$ Query OK, 0 rows affected (0.009 sec) MariaDB [dbprocedure]> DELIMITER ; MariaDB [dbprocedure]> Call GantiHarga('CK003', '219'); Query OK, 1 row affected (0.005 sec) MariaDB [dbprocedure]> Select * from barang; +-----+-----+-----+-----+-----+ KdBarang NamaBarang QtyBarang UnitBarang HargaBarang +-----+-----+-----+-----+-----+ CK001 Antarestar Backpack 65L 12 DUS 499.000 CK003 Consina Sleeping Bag 6 DUS 219.000 +-----+-----+-----+-----+-----+ 2 rows in set (0.001 sec) </pre> Data Sebelum Diganti:

```
MariaDB [dbprocedure]> select * from barang;
```

KdBarang	NamaBarang	QtyBarang	UnitBarang	HargaBarang
CK001	Antarestar Backpack 65L	12	DUS	499.000
CK003	Consina Sleeping Bag	6	DUS	149.000

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

BAB VI

TRIGGER

Trigger adalah serangkaian tindakan yang dijalankan secara otomatis saat megoperasikan perubahan tertentu yang dilakukan pada tabel tertentu. Kejadian (event) yang dapat membangkitkan trigger umumnya berupa pernyataan INSERT, UPDATE, atau DELETE. Berdasarkan ruang lingkupnya, trigger diklasifikasikan menjadi dua jenis : row trigger dan statement trigger. Trigger baris(row) mendefinisikan aksi untuk setiap baris tabel, trigger pernyataan hanya berlaku untuk setiap pernyataan INSERT, UPDATE, atau DELETE.(Server & Hartama, 2000)

Dari sisi perilaku eksekusi, trigger dapat dibedakan menjadi beberapa jenis, namun umumnya ada dua jenis trigger : trigger BEFORE dan AFTER.

Kita dapat mendefinisikan maksimum enam jenis tindakan atau peristiwa dalam bentuk trigger :

- BEFORE INSERT, ini diaktifkan sebelum penyisipan data ke dalam tabel.
- AFTER INSERT, ini diaktifkan setelah penyisipan data ke dalam tabel.
- BEFORE UPDATE, ini diaktifkan sebelum pembaruan data dalam tabel.
- AFTER UPDATE, ini diaktifkan setelah pembaruan data dalam tabel.
- BEFORE DELETE, ini diaktifkan sebelum data dihapus dari tabel.
- AFTER UPDATE, ini diaktifkan setelah penghapusan data sari tabel.

Perintah / Sintaks umum Trigger adalah :

```
DELIMITER $$  
CREATE TRIGGER nama_trigger  
{BEFORE | AFTER}{ INSERT | UPDATE | DELETE}  
ON nama_table  
FOR EACH ROW  
BEGIN  
Kode SQL  
END $$  
DELIMITER ;
```

Keuntungan Trigger, antara lain :

1. Membantu mengotomatiskan perubahan data.
2. Memungkinkan kami menggunakan kembali query yang pernah ditulis.
3. Menyediakan metode untuk memeriksa integritas data database.
4. Membantu mendeteksi kesalahan pada level database.
5. Memungkinkan pengauditan data dengan mudah.

Ada pula kekurangan Trigger, antara lain :

1. Meningkatkan biaya overhead server.
2. Hanya menyediakan validasi yang diperluas yang tidak semua validasi dapat diakses di pemicu SQL.
3. Memecahkan masalah kesalahan karena Trigger adalah pekerjaan yang membosankan.
4. Dapat menyebabkan kesalahan logis dalam aplikasi meskipun ada sedikit kesalahan dalam query.
5. Kita bisa kehilangan data asli jika kita salah menyetel Trigger.

A. Membuat dan Menjalankan Trigger (Praktikum 6)

1. Membuat Database dbtrigger24S04 dan masuk ke dalam database tersebut

Syntax :	Create database DbTrigger24S04; Setelah OK, Use DbTrigger24S04
Penjelasan :	Membuat Database DbTrigger24S04 dan masuk ke database tersebut
Hasil	<pre>MariaDB [(none)]> create database DbTrigger24S04 -> ; Query OK, 1 row affected (0.002 sec) MariaDB [(none)]> use DbTrigger24S04 Database changed</pre>

2. Membuat Tabel Petugas

Syntax :	Create Table petugas(→ Kd_petugas int Primary Key, → Nama_petugas Varchar(50), → Username varchar(50), → Password varchar(10), → Alamat varchar(50));
Penjelasan :	Membuat tabel petugas yang isi nya kd _petugas tipe data int constraint Primary Key, nama _petugas tipe data varchar(50), username tipe data varchar(50), password tipe data varchar(10), alamat dengan tipe data varchar(50)

Hasil	<pre> MariaDB [DbTrigger24S04]> create table petugas(-> kd_petugas int Primary Key, -> nama_petugas varchar(50), -> username varchar(50), -> password varchar(10), -> alamat varchar(50)); Query OK, 0 rows affected (0.019 sec) </pre> Desc nya: <pre> MariaDB [DbTrigger24S04]> desc petugas; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ kd_petugas int(11) NO PRI NULL nama_petugas varchar(50) YES NULL username varchar(50) YES NULL password varchar(10) YES NULL alamat varchar(50) YES NULL +-----+-----+-----+-----+-----+-----+ 5 rows in set (0.036 sec) </pre>
---------------------	--

3. Membuat Tabel Petugas_log

Syntax :	<pre> Create Table petugas_log(-> No int Auto_increment Primary Key, -> Nama Varchar(50), -> Waktu timestamp, -> Keterangan Text); </pre>
Penjelasan :	Membuat tabel petugas log yang akan menjadi Sistem Respon pada Trigger yang akan dibuat
Hasil	<pre> MariaDB [DbTrigger24S04]> create table petugas_log(-> No INT auto_increment PRIMARY KEY, -> Nama Varchar(50), -> Waktu timestamp, -> keterangan Text); Query OK, 0 rows affected (0.027 sec) </pre> Desc nya: <pre> MariaDB [DbTrigger24S04]> desc petugas_log; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ No int(11) NO PRI NULL auto_increment Nama varchar(50) YES NULL Waktu timestamp NO current_timestamp() on update current_timestamp() keterangan text YES NULL +-----+-----+-----+-----+-----+-----+ 4 rows in set (0.039 sec) </pre>

4. Membuat Trigger BeforeTugasInsert

Syntax :	<pre> DELIMITER \$\$ Create Trigger BeforeTugasInsert BEFORE Insert ON Petugas FOR EACH ROW BEGIN Inser Into Petugas_Log VALUES (0,New.username, Now(), 'Petugas Baru'); END \$\$ Delimiter ; </pre>
Penjelasan :	Membuat Trigger Responsif terhadap Tabel Petugas saat memasukkan data pada Petugas akan memicu responsif dari si Trigger ke Petugas_log.

Hasil	<pre> MariaDB [DbTrigger24S04]> DELIMITER \$\$ MariaDB [DbTrigger24S04]> CREATE Trigger BeforeTugasInsert -> 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 (0.033 sec) </pre>
-------	--

5. Memasukkan Data ke Tabel Petugas

Syntax :	<pre> Insert Into Petugas Values(→ Kolom1, Kolom2, Kolom n...); </pre>
Penjelasan :	Memasukkan data ke dalam Tabel Petugas dan memicu Responsif Trigger terhadap Table Petugas log
Hasil	<pre> MariaDB [DbTrigger24S04]> Insert Into Petugas Values(-> 102, 'Rafael Sinaga', '@rafael_sinagaa', 'paell123', 'Sibatu-batu'), -> (103, 'Diaz Al Latif', '@diazallatif20', 'diaz14', 'Bah Jambi'); Query OK, 2 rows affected (0.005 sec) Records: 2 Duplicates: 0 Warnings: 0 </pre> Respon Trigger terhadap Petugas Log: <pre> MariaDB [DbTrigger24S04]> select * from petugas_log; +-----+-----+-----+-----+ No Nama Waktu keterangan +-----+-----+-----+-----+ 1 @arifhrvn150 2025-10-09 08:27:05 PETUGAS BARU 2 @rafael_sinagaa 2025-10-09 08:29:42 PETUGAS BARU 3 @diazallatif20 2025-10-09 08:29:42 PETUGAS BARU +-----+-----+-----+-----+ </pre>

6. Membuat Trigger AfterTugasDelete

Syntax :	<pre> DELIMITER \$\$ Create Trigger AfterTugasDelete AFTER DELETE ON Petugas FOR EACH ROW BEGIN Insert into Petugas_log VALUES (0,'Old.username', Now(), 'Penghapusan Petugas'); END \$\$ DELIMITER ; </pre>
Penjelasan :	Memasukkan data ke dalam Tabel Petugas dan memicu Responsif Trigger terhadap Table Petugas log
Hasil	<pre> MariaDB [DbTrigger24S04]> DELIMITER \$\$ MariaDB [DbTrigger24S04]> Create Trigger AfterTugasDelete -> 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 (0.017 sec) MariaDB [DbTrigger24S04]> DELIMITER ; MariaDB [DbTrigger24S04]> Delete From petugas WHERE username = '@diazallatif20'; Query OK, 1 row affected (0.022 sec) </pre> Respon Trigger terhadap Petugas Log (Nomor 4):

```
MariaDB [DbTrigger24S04]> select * from petugas_log;
```

No	Nama	Waktu	keterangan
1	@arifhrvn150	2025-10-09 08:27:05	PETUGAS BARU
2	@rafael_sinagaa	2025-10-09 08:29:42	PETUGAS BARU
3	@diazallatif20	2025-10-09 08:29:42	PETUGAS BARU
4	@diazallatif20	2025-10-09 08:33:21	PENGHAPUSAN PETUGAS

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

7. Membuat Trigger AfterTugasUpdate

Syntax :	<pre>DELIMITER \$\$ Create Trigger AfterTugasUpdate AFTER UPDATE On Petugas FOR EACH ROW BEGIN DECLARE keterangan TEXT; SET keterangan = CONCAT(Old.Username, ' -> ', New.username); END \$\$ DELIMITER ; Update petugas Set username = '@ondosaut23' WHERE kd_petugas = '104';</pre>																												
Penjelasan :	Mengganti data (@ondosaut23 Sebagai nama baru dan 104 merupakan kd_petugas patokan penggantian kolom) yang ada pada Tabel Petugas dan memicu Responsif Trigger terhadap Table Petugas_log																												
Hasil	<pre>MariaDB [DbTrigger24S04]> Create Trigger AfterTugasUpdate -> 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 (0.021 sec) MariaDB [DbTrigger24S04]> DELIMITER ;</pre> Respon Trigger terhadap Petugas Log (Nomor 6): <pre>MariaDB [DbTrigger24S04]> Update Petugas Set username = '@ondosaut23' WHERE kd_Petugas = '104'; Query OK, 1 row affected (0.013 sec) Rows matched: 1 Changed: 1 Warnings: 0 MariaDB [DbTrigger24S04]> select * from petugas_log;</pre> <table><tr><th>No</th><th>Nama</th><th>Waktu</th><th>keterangan</th></tr><tr><td>1</td><td>@arifhrvn150</td><td>2025-10-09 08:27:05</td><td>PETUGAS BARU</td></tr><tr><td>2</td><td>@rafael_sinagaa</td><td>2025-10-09 08:29:42</td><td>PETUGAS BARU</td></tr><tr><td>3</td><td>@diazallatif20</td><td>2025-10-09 08:29:42</td><td>PETUGAS BARU</td></tr><tr><td>4</td><td>@diazallatif20</td><td>2025-10-09 08:33:21</td><td>PENGHAPUSAN PETUGAS</td></tr><tr><td>5</td><td>@ondorajagukguk</td><td>2025-10-09 08:38:45</td><td>PETUGAS BARU</td></tr><tr><td>6</td><td>@ondosaut23</td><td>2025-10-09 08:39:26</td><td>@ondorajagukguk --> @ondosaut23</td></tr></table> <pre>6 rows in set (0.001 sec)</pre>	No	Nama	Waktu	keterangan	1	@arifhrvn150	2025-10-09 08:27:05	PETUGAS BARU	2	@rafael_sinagaa	2025-10-09 08:29:42	PETUGAS BARU	3	@diazallatif20	2025-10-09 08:29:42	PETUGAS BARU	4	@diazallatif20	2025-10-09 08:33:21	PENGHAPUSAN PETUGAS	5	@ondorajagukguk	2025-10-09 08:38:45	PETUGAS BARU	6	@ondosaut23	2025-10-09 08:39:26	@ondorajagukguk --> @ondosaut23
No	Nama	Waktu	keterangan																										
1	@arifhrvn150	2025-10-09 08:27:05	PETUGAS BARU																										
2	@rafael_sinagaa	2025-10-09 08:29:42	PETUGAS BARU																										
3	@diazallatif20	2025-10-09 08:29:42	PETUGAS BARU																										
4	@diazallatif20	2025-10-09 08:33:21	PENGHAPUSAN PETUGAS																										
5	@ondorajagukguk	2025-10-09 08:38:45	PETUGAS BARU																										
6	@ondosaut23	2025-10-09 08:39:26	@ondorajagukguk --> @ondosaut23																										

DAFTAR PUSTAKA

- Adflin, J. (2020). Bahasa Query. *Universitas Palangka Raya*, 21(1), 1–9. <http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203%0Ahttp://mpoc.org.my/malaysian-palm-oil-industry/>
- Amin, M. (2022). Bahasa Query Menggunakan MySQL. In *Ahatek*.
- Anggoro Dimas Aryo, Supriyanti Wiwit, & Putri Devri Afriyantari Puspa. (2021). Konsep Dasar Sistem Basis Data Dengan Mysql. In *Muhammadiyah University Press* (p. 246).
- Arianti, B. D. D., & Jamaluddin. (2022). *Basis Data Terdistribusi (Client Server, SQL, Basis Data Dalam Web)* (p. 110).
- Budayawan, K., Asmara, D., & Darni, R. (2023). *Basis Data.pdf* (p. 97).
- Canggih Ajika Pamungkas. (2017). Pengantar dan Implementasi Basis Data - Google Books. In *Deepublish* (p. 68).
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=
- Garcia, A. R., Filipe, S. B., Fernandes, C., Estevão, C., & Ramos, G. (n.d.). *No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析* Title.
- Gunawan, A., Ningsih, S., & Lantana, D. A. (2023). Pengantar Basis Data. In *Gastronomia ecuatoriana y turismo local*. (Vol. 15, Issue 2).
- Ismail. (2021). *Dasar-Dasar SQL MariaDB*.
https://www.google.co.id/books/edition/Dasar_Dasar_SQL_MariaDB/OdQpEAAAQBAJ?hl=en&gbpv=1&dq=mariadb+adalah&pg=PA62&printsec=frontcover
- Jamaluddin, Samosir, K., S, W., Devia, E., Santoso, L. W., Yuniansyah, Juanaidi, Nursari, S. R. C., Azizah, N., & Saputra, M. H. (2022). BUKU (Book Chapter)-Sistem Basis Data (Elmi Devia)_oke. In *PT. Global Eksekutif Teknologi*.
[https://repository.unkris.ac.id/id/eprint/518/1/BUKU \(Book Chapter\)-Sistem Basis Data \(Elmi Devia\)_oke.pdf](https://repository.unkris.ac.id/id/eprint/518/1/BUKU%20(Book%20Chapter)-Sistem%20Basis%20Data%20(Elmi%20Devia)_oke.pdf)
- Kurniati, I. D., Setiawan, R., Rohmani, A., Lahdji, A., Tajally, A., Ratnaningrum, K., Basuki, R., Reviewer, S., & Wahab, Z. (2015). *Buku Ajar*.
- Samsoni, Ardiansyah, M., Rofiq Nur, & Haerudin Heri. (2021). *BASIS DATA I Penyusun* (Issue 1). www.unpam.ac.id
- Server, M. S. Q. L., & Hartama, D. (2000). *on sep Dasar SQL Server 1*.
- Silalahi, F. D. (2022). Manajemen Databse MySQL. *Yayasan Prima Agus Teknik Dan Universitas STEKOM*, 1–158.

DOKUMENTASI

