



**FIKOM**



# MODUL PRAKTIKUM BASIS DATA I

20  
23  
20  
24



## Tim Penyusun

- Dedy Atmajaya, S.Kom., M.Eng., MTA
- Amaliah Faradibah, S.Kom., M.Kom., MTA., MCF
- Andi Ulfa Tenripada, S.Kom., M.Kom
- Ir. Herman, S.Kom., M.Cs., MTA
- Ir. Abdul Rachman Manga, S.Kom., M.T., MTA., MCF
- Ir. Huzain Azis, S.Kom., M.Cs., MTA
- Lutfi Budi Ilmawan, S.Kom., M.Cs., MTA
- Tim Asisten Laboratorium

## KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Modul Praktikum **Basis Data 1** untuk mahasiswa/i Program Studi Teknik Informatika dan Sistem Informasi Fakultas Ilmu Komputer Universitas Muslim Indonesia ini dapat diselesaikan dengan sebaik-baiknya.

Modul praktikum ini dibuat sebagai pedoman dalam melakukan kegiatan praktikum **Basis Data 1** yang merupakan kegiatan penunjang mata kuliah pada Program Studi Teknik Informatika dan Sistem Informasi. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan capaian pembelajaran mata kuliah pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum **Basis Data 1** ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Makassar, Maret 2024

Tim Penyusun

## TATA TERTIB PELAKSANAAN PRAKTIKUM

Tata Tertib Pelaksanaan Praktikum pada Laboratorium Terpadu Fakultas Ilmu Komputer UMI adalah sebagai berikut:

1. Seluruh Pengguna laboratorium harus dalam keadaan sehat tidak menunjukkan gejala sakit (batuk, hidung tersumbat, dan suhu badan diatas 37°C).
2. Praktikan hanya diizinkan melaksanakan praktikum apabila :
  - a. Pria
    - Berpakaian rapi memakai kemeja putih polos;
    - Menggunakan celana kain berwarna hitam bukan dari bahan jeans/semi jeans;
    - Rambut rapi dan tidak panjang;
  - b. Wanita
    - Berpakaian rapi memakai kemeja tunik putih polos (tidak transparan)
    - Memakai Jilbab Segitiga Hitam (bukan pasmina) dan menutupi dada.
    - Menggunakan Rok Panjang berwarna hitam yang tidak terbelah dan tidak span serta bukan dari bahan jeans/semi jeans;
    - Memakai kaos kaki dengan tinggi minimal 10 cm di atas mata kaki;
3. Ketika memasuki dan selama berada dalam ruangan, praktikan diwajibkan :
  - Tenang, tertib, dan sopan;
  - Tidak mengganggu praktikan lain yang sedang melaksanakan praktikum;
  - Tidak diperbolehkan merokok, membawa makanan / minuman senjata tajam dan senjata api ke dalam ruangan praktikum;
  - Tidak diperbolehkan membawa *handphone* ke meja praktikum dan *handphone* dalam mode senyap;
  - Tidak diperbolehkan membawa media penyimpanan eksternal atau *flashdisk* ke meja praktikum tanpa seizin Dosen Pengampu atau Asisten;
4. Dilarang membawa, mengambil, serta memindahkan perangkat yang digunakan pada saat praktikum tanpa instruksi dari Dosen Pengampu atau Asisten.
5. Toleransi keterlambatan praktikan maksimal 5 menit.
6. Praktikan berada diarea laboratorium dengan mengikuti jadwal yang telah ditentukan oleh Kepala Laboratorium.
7. Penggunaan fasilitas Laboratorium menyesuaikan dengan kapasitas ruang Laboratorium.

- 8 Segala pelanggaran yang dilakukan oleh praktikan akan berakibat pada penutupan dan penghentian penggunaan seluruh fasilitas laboratorium dan ditindak sesuai dengan aturan yang berlaku.

## SANKSI-SANKSI

Sanksi terhadap pelanggaran **TATA TERTIB**:

Dosen Pengampu dan Asisten laboratorium berhak menjatuhkan sanksi, sesuai dengan aturan yang berlaku di Laboratorium Terpadu Fakultas Ilmu Komputer UMI apabila :

1. Praktikan merusak peralatan praktikum (*Personal Computer*) secara sengaja, maka praktikan bertanggung jawab untuk mengganti kerusakan tersebut.
2. Praktikan tidak mematuhi dan mentaati aturan praktikum maka tidak diperkenankan mengikuti praktikum.

Pelanggaran point lainnya dikenakan sanksi teguran, dikeluarkan/dicoret namanya dalam kegiatan praktikum (mengulang mata kuliah sesuai dengan semester berjalan) sampai sanksi akademik.



Kepala Laboratorium Terpadu,

**Ir. Abdul Rachman Manga', S.Kom., M.T.,  
MTA., MCF**

## DAFTAR ISI

KATA PENGANTAR .....	2
TATA TERTIB PELAKSANAAN PRAKTIKUM.....	3
DAFTAR ISI.....	3
CAPAIAN PEMBELAJARAN MATA KULIAH (CPMK) .....	6
MODUL 1 – DDL ( <i>Data Definition Language</i> ).....	7
MODUL 2 – DML ( <i>Data Manipulation Language</i> ).....	13
MODUL 3 – Fungsi Agregat, Tanggal, dan String .....	20
MODUL 4 – Query Antar Table & Operator .....	25
MODUL 5 – Operator Join .....	30
MODUL 6 – Sub Query.....	36
MODUL 7 – Normalisasi.....	40
MODUL 8 – DCL ( <i>Data Control Language</i> ).....	44

### **CAPAIAN PEMBELAJARAN MATA KULIAH (CPMK)**

1. CPMK1 : Mahasiswa mampu memahami konsep dasar Basis Data
2. CPMK2 : Mahasiswa mampu melakukan analisis keperluan untuk menyusun deskripsi basis data
3. CPMK3 : Mahasiswa mampu melakukan perancangan basis data
4. CPMK4 : Mahasiswa mampu mengembangkan basis data satu sistem informasi
5. CPMK5 : Mahasiswa memiliki kemampuan softskill dalam pembelajaran berupa:
  - Mampu berkomunikasi lisan dengan baik
  - Mampu bekerja sama dalam kelompok
  - Mampu mengelola / leadership dalam kelompok.



## MODUL 1 – DDL (*Data Definition Language*)

### A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mampu menjelaskan perintah-perintah DDL (*Data Definition Language*)
2. Mampu menggunakan perintah-perintah DDL (*Data Definition Language*) pada Aplikasi DBMS (MySQL)

### B. Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan DML (*Data Manipulation Language*) ?
2. Apa perbedaan Perintah Drop pada DML dengan Perintah Drop pada Alter DDL ?
3. Berikan masing-masing sebuah contoh implementasi perintah-perintah DML !

### C. Instrumen dan Prosedur

#### 1. Instrument

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau di atasnya (sebagai *engine basis data*)

#### 2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

### D. Teori Dasar

#### 1. Basis Data

Basis data adalah kumpulan informasi yang disimpan di dalam Komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program Komputer untuk memperoleh informasi dari basis data tersebut. Pada tabel terbagi 2 bagian yaitu field dan record.

	Field	Field	Field
Record	Stambuk	Nama	Alamat
Record			

## 2. DDL (*Data Definition Language*)

DDL merupakan perintah SQL yang berkaitan dengan pendefinisian suatu struktur database dalam hal ini *database* dan *table*. Perintah DDL adalah: CREATE, ALTER, RENAME, DROP.

### 1) Create

Berfungsi untuk membuat database atau tabel baru.

- a. Membuat database

Query: CREATE DATABASE nama\_database;

- b. Membuat table

Query:

CREATE TABLE nama\_tabel(Field1 TipeData1, Field2 TipeData2,  
..... FieldN TipeDataN);

### 2) Use

Perintah USE digunakan untuk memilih database yang ingin digunakan.

Query : USE nama\_database;

### 3) Show

Berfungsi untuk melihat list/daftar dari database atau tabel yang sudah dibuat.

- a. Melihat daftar database

Untuk melihat daftar database yang sudah dibuat.

Query : SHOW databases;

- b. Melihat daftar table

Untuk melihat daftar tabel yang sudah dibuat.

Query : SHOW tables;

### 4) Desc

Perintah DESC digunakan untuk melihat struktur (metadata) sebuah tabel.

Query : DESC nama\_tabel;

### 5) Alter

Alter adalah perintah untuk mengubah struktur table. Perintah Alter terbagi mejadi 5 bagian, yaitu:

- Rename, digunakan untuk mengubah/mengganti nama tabel
- Change, digunakan untuk mengganti nama field beserta jenisnya pada table
- Modify, digunakan untuk mengganti jenis field pada table
- Add, digunakan untuk menambahkan field pada table
- Drop, digunakan untuk menghapus field pada table

### 6) Rename

Perintah RENAME digunakan untuk mengubah nama tabel.

Query : RENAME nama\_tabel\_lama nama\_tabel\_baru;



**7) Drop**

Perintah DROP digunakan untuk menghapus struktur dalam database, termasuk menghapus database dan table.

Query :

DROP DATABASE nama\_database;

DROP TABLE nama\_tabel;

**3. Tipe Data****a. Tipe Data Numeric**

No	Tipe Data	Jangkauan
1.	INT	-2.147.483.648 s/d 2.147.483.648
2.	BIGINT	$\pm 9.22 \times 10^{18}$
3.	FLOAT	-3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38.
4.	DECIMAL	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.

**b. Tipe Data Date & Time**

No	Tipe Data	Fungsi	Jangkauan
1.	DATE	Menyimpan data tanggal	1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)
2.	TIME	Menyimpan data waktu	-838:59:59 s/d +838:59:59 (HH:MM:SS)
3.	DATETIME	Menyimpan data tanggal dan waktu	'1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'
4.	YEAR	Menyimpan data tahun dari tanggal	1900 2155

**c. Tipe Data String**

No	Tipe Data	Fungsi	Jangkauan
1.	CHAR	Menyimpan data string ukuran tetap	0 s/d 255 karakter
2.	VARCHAR	Menyimpan data string ukuran dinamis	0 s/d 255 karakter (versi 4.1), s/d 65.535
3.	TEXT	Menyimpan data text	0 s/d 65.535
4.	MEDIUMTEXT	Menyimpan data text	0 s/d 224 – 1 karakter
5.	LONGTEXT	Menyimpan data text	0 s/d 232 – 1 karakter

**d. Tipe Data Lain**

No	Tipe Data	Fungsi	Jangkauan
1.	ENUM	Enumerisasi (kumpulan data)	Sampai dengan 65535 string

2.	SET	Combination (himpunan data)	Sampai dengan 255 string anggota
----	-----	-----------------------------	----------------------------------

## E. Kegiatan Praktikum

### 1. Studi Kasus

Sebelum menjalankan perintah DDL, terlebih dahulu kita mengaktifkan MySQL Server. Pada praktikum kali ini, praktikan akan belajar membuat database dan menjalankan perintah DDL.

#### 1. Memasuki Direktori Mysql pada system anda dengan terminal

```
mysql > cd xampp/bin/
mysql > mysql -u root -p
Enter password :
```

#### 2. Menjalankan service MySQL pada system anda dengan terminal

```
mysql> mysql -u root -p
```

#### 3. Menampilkan daftar database yang telah ada

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| tescbtpendaftarcca |
| test |
+-----+
6 rows in set (0.07 sec)
```

#### 4. Membuat database dan tabel

```

MariaDB [(none)]> create database akademik;
Query OK, 1 row affected (0.02 sec)

MariaDB [(none)]> use akademik;
Database changed
MariaDB [akademik]> create table mhs (stambuk int(11),
    -> nama varchar(50),
    -> kelas char(3));
Query OK, 0 rows affected (0.07 sec)

MariaDB [akademik]> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| stambuk | int(11)       | YES  |     | NULL    |      |
| nama    | varchar(50)   | YES  |     | NULL    |      |
| kelas   | char(3)       | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)

```

#### 5. Perintah alter

```

MariaDB [akademik]> alter table mhs RENAME mahasiswa;
Query OK, 0 rows affected (0.11 sec)

MariaDB [akademik]> alter table mahasiswa ADD alamat varchar(50);

Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [akademik]> alter table mahasiswa DROP kelas;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [akademik]> alter table mahasiswa CHANGE nama nama_mhs
    -> varchar(50);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [akademik]> alter table mahasiswa MODIFY alamat text;
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

**LEMBAR EVALUASI PRAKTIKUM**

- Buatlah sebuah database dengan nama “Teknik Informatika” dan sebuah tabel dengan nama tabel Praktikan yang memiliki struktur table : stambuk int, nama varchar(30), jenisKelamin enum(‘L’,’P’), alamat varchar(100).
  - Lihat struktur table anda.
  - Ubah tipe data stambuk menjadi tipe data varchar.
  - Tambahkan field noHp dengan tipe data varchar
  - Ubah nama table anda menjadi nama anda.
- Sebutkan dan jelaskan tipe data yang digunakan pada tabel anda!
- Jelaskan pengertian field atau record! (Berikan contoh).

Evaluasi Praktikum 1:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Pemahaman Perintah Create database dan tabel					
2.	Pemahaman Perintah Alter					
3.	Pemahaman Perintah Drop database dan tabel					
4.	Ketepatan dalam menjabarkan tipe-tipe data					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_

**MODUL 2 – DML (*Data Manipulation Language*)****A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mampu menjelaskan perintah-perintah DML (*Data Manipulation Language*)
2. Mampu menggunakan perintah-perintah DML (*Data Manipulation Language*) pada Aplikasi DBMS (MySQL).

**B. Tugas Pendahuluan**

1. Jelaskan apa yang dimaksud fungsi Agregat, fungsi Tanggal, dan fungsi String?
2. Sebutkan dan jelaskan bagian-bagian fungsi Agregat?
3. Sebutkan dan jelaskan bagian-bagian dari fungsi Tanggal?
4. Sebutkan dan jelaskan bagian-bagian dari fungsi String?
5. Buatlah sebuah database yang mengimplementasikan fungsi agregat, fungsi Tanggal, dan fungsi String.

**C. Instrumen dan Prosedur****1. Instrument**

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau di atasnya (sebagai *engine basis data*)

**2. Prosedur**

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

**D. Teori Dasar**

*Data Manipulation Language* adalah bahasa yang memungkinkan pengguna mengakses atau memanipulasi data seperti yang diatur oleh model data. Manipulasi data adalah:

- Penempatan informasi baru dalam basis data
- Pengambilan informasi yang disimpan dalam basis data
- Modifikasi informasi yang disimpan dalam basis data
- Penghapusan informasi dari basis data

**1. Insert Tabel**

Insert merupakan perintah yang dapat digunakan untuk melakukan input data ke dalam tabel yang sudah ada.

Perintahnya :

Insert into Nama\_Table values (  
 Isi\_Field\_1, Isi\_Field\_2, ... , Isi\_Field\_N) ;

Atau dengan perintah :

Insert into Nama\_Table (Nama\_Field\_1, Nama\_Field\_2)  
 values (Isi\_Field\_1, Isi\_Field\_2);

## 2. Update Tabel

Update merupakan perintah yang dapat digunakan untuk melakukan perubahan terhadap data yang sudah ada/dibuat.

Perintahnya :

Update Nama\_Table set Nama\_Field = 'Data\_Baru'  
 where Nama\_Field\_Key = 'Data\_Key';

Contoh :

Update customers set cust\_address = 'BTP' where cust\_id='10002';

## 3. Delete Tabel

Delete merupakan perintah yang dapat digunakan untuk menghapus data yang terdapat di dalam tabel.

Perintahnya :

Delete From Nama\_Table where Nama\_Field\_Key;

Contoh :

Delete From Mhs Where stambuk='13020210002';

## 4. Select Tabel

Select merupakan perintah yang dapat digunakan untuk :

- Menampilkan data secara keseluruhan yang terdapat di dalam table.
- Menampilkan data tertentu yang terdapat di dalam table.
- Menampilkan dan mengurutkan data secara ascending dan descending.

Perintahnya :

Select \* From Nama\_Table;

Contoh :

Select \*from Mahasiswa;

### Menampilkan Kolom Data Tertentu

Jika hanya ingin menampilkan beberapa field tertentu dalam suatu table. Misalkan dari data yang terdapat pada tabel Mahasiswa yang mempunyai Field (NIM, Nama\_Mhs, Alamat) dan hanya akan menampilkan NIM dan Nama\_Mhs.

Perintahnya :

Select Nama\_Field\_1, ... , Nama\_Field\_N From Nama\_Table;

Contoh :

Select NIM, Nama\_Mhs From Mahasiswa;



**Menampilkan Record Tertentu**

Jika hanya ingin menampilkan beberapa baris tertentu dalam suatu table. Misalkan dari data yang terdapat pada tabel Mahasiswa ingin menampilkan baris tertentu maka akan ditambahkan kondisi pada clause setelah where.

Perintahnya :

Select \* from Nama\_Table where Kondisi;

Contoh :

Select NIM, Nama\_Mhs From Mahasiswa Where Nim=13020210001;

**E. Kegiatan Praktikum****1. Prosedur**

- Baca dan pahami semua tahapan praktikum dengan cermat.
- Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- Pastikan diri anda tidak menyentuh sumber listrik.

**2. Studi Kasus**

Sebelum memanipulasi data, Anda harus memiliki sebuah database dan minimal sebuah table. Pada praktikum kali ini, buatlah sebuah database dengan nama 'perpustakaan' dan berisi tabel buku dan anggota.

- Buat tabel sesuai deskripsi tabel berikut:

- Tabel Buku

No	Kolom	Tipe	Keterangan
1	no_buku	char(5)	Primary Key
2	judul	varchar(50)	-
3	pengarang	varchar(50)	-
4	jenis_buku	enum('pengetahuan','olahraga','novel')	-

- Tabel Anggota

No	Kolom	Tipe	Keterangan
1	id_anggota	char(9)	Primary Key
2	nama	varchar(100)	-
3	alamat	varchar(50)	-
5	no_telp_	char(15)	-
6	tgl_lahir	date	-

```
MariaDB [(none)]> create database perpustakaan;
Query OK, 1 row affected (0.01 sec)
```

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

```
MariaDB [perpustakaan]> create table buku (no_buku char(5)
-> primary key, judul varchar(50), pengarang varchar(50),
-> jenis_buku enum('Pengetahuan','Olahraga','Novel'));
Query OK, 0 rows affected (0.04 sec)
```

```
MariaDB [perpustakaan]> desc buku;
```

Field	Type	Null	Key	Default	Extra
no_buku	char(5)	NO	PRI	NULL	
judul	varchar(50)	YES		NULL	
pengarang	varchar(50)	YES		NULL	
jenis_buku	enum('Pengetahuan','Olahraga','Novel')	YES		NULL	

```
4 rows in set (0.03 sec)
```

```
MariaDB [perpustakaan]> create table anggota (id_anggota char(9)
-> primary key, nama varchar(100), alamat varchar(50),
-> no_telp char(15), tgl_lahir date);
Query OK, 0 rows affected (0.04 sec)
```

```
MariaDB [perpustakaan]> desc anggota;
```

Field	Type	Null	Key	Default	Extra
id_anggota	char(9)	NO	PRI	NULL	
nama	varchar(100)	YES		NULL	
alamat	varchar(50)	YES		NULL	
no_telp	char(15)	YES		NULL	
tgl_lahir	date	YES		NULL	

```
5 rows in set (0.03 sec)
```

## 2. Isi record pada tabel:

```
MariaDB [perpustakaan]> insert into anggota values
-> ('BB001','Anugerah','Jl Genuksari','085677889000','2000/02/01'),
-> ('BB002','Mahrita','Jl. Bates Utara','081228004432','1999/10/20'),
-> ('BB003','Retno','Jl. Mekarjaya','089955432122','1999/07/18'),
-> ('BB004','Debora','Jl. Plamonganray','089954376666','2001/03/19'),
-> ('BB005','Lydia','Jl. Genuk Indah','087987678899','1992/10/10');
```

```
Query OK, 5 rows affected (0.01 sec)
```

```
Records: 5 Duplicates: 0 Warnings: 0
```

```
MariaDB [perpustakaan]> select *from anggota;
```

id_anggota	nama	alamat	no_telp	tgl_lahir
BB001	Anugerah	Jl Genuksari	085677889000	2000-02-01
BB002	Mahrita	Jl. Bates Utara	081228004432	1999-10-20
BB003	Retno	Jl. Mekarjaya	089955432122	1999-07-18
BB004	Debora	Jl. Plamonganray	089954376666	2001-03-19
BB005	Lydia	Jl. Genuk Indah	087987678899	1992-10-10

```
5 rows in set (0.01 sec)
```

```

MariaDB [perpustakaan]> insert into buku values
-> ('CC001','Kesehatan','Erlangga','Pengetahuan'),
-> ('CC002','Matematika','Budi','Pengetahuan'),
-> ('CC003','Geografi','Galih','Pengetahuan'),
-> ('CC004','Sepakbola','Andi','Olahraga'),
-> ('CC005','Cinta Aisyah','Mira','Novel');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

```

```

MariaDB [perpustakaan]> select *from buku;
+-----+-----+-----+-----+
| no_buku | judul      | pengarang | jenis_buku |
+-----+-----+-----+-----+
| CC001   | Kesehatan  | Erlangga  | Pengetahuan |
| CC002   | Matematika | Budi      | Pengetahuan |
| CC003   | Geografi   | Galih     | Pengetahuan |
| CC004   | Sepakbola  | Andi      | Olahraga    |
| CC005   | Cinta Aisyah | Mira     | Novel       |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

```

### 3. Perintah Update dan Delete:

```

MariaDB [perpustakaan]> update buku set pengarang='Erlangga' where no_buku='CC004';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [perpustakaan]> select *from buku;
+-----+-----+-----+-----+
| no_buku | judul      | pengarang | jenis_buku |
+-----+-----+-----+-----+
| CC001   | Kesehatan  | Erlangga  | Pengetahuan |
| CC002   | Matematika | Budi      | Pengetahuan |
| CC003   | Geografi   | Galih     | Pengetahuan |
| CC004   | Sepakbola  | Erlangga  | Olahraga    |
| CC005   | Cinta Aisyah | Mira     | Novel       |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```
MariaDB [perpustakaan]> delete from anggota where id_anggota='BB001';
Query OK, 1 row affected (0.01 sec)
```

```
MariaDB [perpustakaan]> select *from anggota;
```

id_anggota	nama	alamat	no_telp	tgl_lahir
BB002	Mahrita	Jl. Bates Utara	081228004432	1999-10-20
BB003	Retno	Jl. Mekarjaya	089955432122	1999-07-18
BB004	Debora	Jl. Plamonganray	089954376666	2001-03-19
BB005	Lydia	Jl. Genuk Indah	087987678899	1992-10-10

4 rows in set (0.00 sec)

## LEMBAR EVALUASI PRAKTIKUM

1. Buatlah database dengan nama 'Rumah Sakit'.
2. Buatlah table dengan nama pasien dengan field idPasien, namaPasien, alamat, jenisKelamin dan umur. Lalu tentukan masing-masing tipe data dari setiap field tersebut !
3. Tambahkan field golonganDarah pada tabel pasien.
4. Isi semua field yang terdapat pada table Pasien. (min. 5 Record)
5. Tampilkan field idPasien dan namaPasien jika jenis kelamin Wanita.
6. Ubahlah record ke-2 pada field nama menjadi Fia.
7. Hilangkan isi(record) jika golonganDarah B.

Evaluasi Praktikum 2:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Dapat menambahkan record pada tabel					
2.	Dapat mengubah record pada tabel					
3.	Dapat menghapus record pada tabel					
4.	Dapat menampilkan data pada tabel					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_

## MODUL 3 – Fungsi Agregat, Tanggal, dan String

### A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mampu memahami penggunaan fungsi agregat, tanggal, dan karakter/string
2. Mampu menggunakan dan menjalankan fungsi agregat, tanggal, dan karakter/string

### B. Tugas Pendahuluan

1. Jelaskan manfaat query antar tabel.
2. Tuliskan dan jelaskan Operator pada mysql
3. Berikan contoh penggunaan query antar tabel pada Mysql.
4. Berikan contoh implementasi operator pada Mysql.

### C. Instrumen dan Prosedur

#### 1. Instrument

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau di atasnya (sebagai *engine basis data*)

#### 2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

### D. Teori Dasar

#### 1. Fungsi Agregat

Fungsi Agregat merupakan fungsi yang berhubungan dengan sekumpulan data pada database sehingga sering disebut pula sebagai fungsi grup atau ringkasan. Fungsi ini menerima sekumpulan data dan mengembalikan nilai tunggal sebagai hasilnya. Adapun fungsi Agregat yaitu:

- 1) SUM digunakan untuk menghitung total nilai dari kolom tertentu.

Bentuk umum:

```
mysql> SELECT SUM(field) as field_alias From nama_table;
```

- 2) COUNT digunakan untuk menghitung jumlah *record*.

Bentuk umum:

```
mysql> SELECT COUNT(*) as field_alias From nama_table;
```

- 3) AVG digunakan untuk menampilkan nilai rata-rata dari suatu kolom.



Bentuk umum:

```
mysql> SELECT AVG(field) as field_alias From nama_table;
```

- 4) MAX digunakan untuk menampilkan nilai tertinggi dari suatu kolom.

Bentuk umum:

```
mysql> SELECT MAX(field) as field_alias From nama_table;
```

- 5) MIN digunakan untuk menampilkan nilai terendah dari suatu kolom.

Bentuk umum:

```
mysql> SELECT MIN(field) as field_alias From nama_table;
```

## 2. Fungsi Tanggal

Fungsi Tanggal merupakan fungsi yang berhubungan dengan tipe data waktu baik itu date, time, dan datetime. Syarat utama dalam fungsi tanggal ialah field yang bertipe waktu. Adapun fungsi Tanggal, antara lain:

- 1) ADDDATE digunakan untuk menghasilkan tanggal beberapa hari atau bulan kemudian.

Bentuk umum:

```
mysql> SELECT ADDDATE(field, interval tambahan) as field_alias From nama_table;
```

- 2) NOW digunakan untuk mengambil waktu baik hari atau tanggal berdasarkan PC masing-masing.

Bentuk umum:

```
mysql> SELECT NOW() as field_alias From nama_table;
```

- 3) DATEDIFF digunakan untuk menghasilkan selisih tanggal.

Bentuk umum:

```
mysql> SELECT DATEDIFF(field, field) as field_alias From nama_table;
```

- 4) DATE\_FORMAT untuk menampilkan format tanggal dan nama bulan.

Bentuk umum:

```
mysql > SELECT DATE_FORMAT(field, '%d %M %Y') as field_alias From nama_table;
```

- 5) DAYNAME digunakan untuk menampilkan hari berdasarkan tanggal.

Bentuk umum:

```
mysql > SELECT DAYNAME(field) as field_alias From nama_table;
```

## 3. Fungsi Karakter / String

- 1) LENGTH digunakan untuk mengetahui panjang suatu string.

Bentuk umum:

```
mysql > SELECT LENGTH(field) as field_alias From nama_table;
```

- 2) LEFT & RIGHT digunakan untuk memotong string pada bagian kiri atau kanan sebuah field.

Bentuk umum:

```
mysql > SELECT LEFT(field, jumlah_potong) as field_alias From nama_table;
mysql> SELECT RIGHT(field, jumlah_potong) as field_alias From
nama_table;
```

- 3) CONCAT digunakan untuk menggabungkan beberapa field menjadi satu.

Bentuk umum:

```
mysql> SELECT CONCAT(field1,field2,...,fieldn) as field_alias From
nama_table;
```

- 4) LOWER digunakan untuk mengkonversi huruf dalam format huruf kecil.

Bentuk umum:

```
mysql> SELECT LOWER(field) as field_alias From nama_table;
```

- 5) UPPER digunakan untuk mengkonversi huruf kecil menjadi huruf kapital.

Bentuk umum:

```
mysql> SELECT UPPER(field) as field_alias From nama_table;
```

## E. Kegiatan Praktikum

### 1. Studi Kasus

- 1) Buatlah sebuah database “perpus”, lalu buatlah table “Peminjaman”. Rancang table anda seperti ini:

Field	Type
idBuku (PK)	Varchar(5)
judulBuku	Varchar(30)
tanggalTerbit	Date
tanggalPeminjaman	Date
hargaBuku	Bigint

- 2) Tambahkan record seperti dibawah ini:

<i>idBuku</i>	<i>judulBuku</i>	<i>tanggalTerbit</i>	<i>tanggalPeminjaman</i>	<i>hargaBuku</i>
<b>B001</b>	BASIS DATA 1	2015-08-25	2019-01-15	50000
<b>B002</b>	STRUKTUR DATA	2014-05-15	2019-01-16	55000
<b>B003</b>	JAVA FUNDAMENTAL	2017-12-04	2019-01-17	250000
<b>B004</b>	WEB	2014-05-16	2019-01-20	75000
<b>B005</b>	AQIDAH	2013-12-04	2019-01-28	30000

- 3) Lakukan perintah berikut ini:

- a) Gunakan fungsi COUNT untuk menghitung jumlah record table peminjaman

```
mysql > SELECT COUNT(*) FROM peminjaman;
```

- b) Gunakanlah fungsi MAX dan MIN untuk menampilkan nilai tertinggi dan terendah pada field hargaBuku.

```
mysql> SELECT MAX(hargaBuku) FROM peminjaman;
mysql > SELECT MIN(hargaBuku) FROM peminjaman;
```

- c) Gunakan fungsi SUM untuk menghitung total harga dari field hargaBuku.

```
mysql > SELECT SUM(hargaBuku) FROM peminjaman;
```

- d) Gunakan fungsi AVG untuk menampilkan nilai rata-rata dari field hargaBuku.

```
mysql > SELECT AVG(hargaBuku) FROM peminjaman;
```

- e) Gunakan fungsi ADDDATE untuk menampilkan batas tanggal pemngembalian dengan menambahkan 7 hari pada field tanggalPeminjaman.

```
mysql> SELECT ADDDATE (tanggalPeminjaman, INTERVAL 7 DAY)
FROM peminjaman;
```

- f) Gunakan fungsi DATEDIFF untuk menampilkan selisih tanggal pengambilan dan pengembalian pada field tanggalPeminjaman dan ADDDATA.

```
mysql> SELECT DATEDIFF(ADDDATE(tanggalPeminjaman, INTERVAL
7 DAY), tanggalPeminjaman) as tglKembali FROM peminjaman;
```

- g) Gunakan fungsi LENGTH untuk menampilkan panjang string pada field namaBuku

```
mysql > SELECT LENGTH(judulBuku) FROM peminjaman;
```

## LEMBAR EVALUASI PRAKTIKUM

1. Buatlah Database dengan nama **rentalMobil**.
2. Buatlah table dengan nama **daftarMobil** dengan field **kdMobil (PK)**, **MerekMobil**, **tglKeluar**, **hargaSewa**.
3. Buatlah table **sewaMobil** dengan field **idSewa (PK)**, **kdMobil (FK)**, **tglSewa**, **tglKembali**.
4. Isilah table **daftarMobil** dan **sewaMobil** minimal 5 record.
5. Khusus field tglSewa di isi berdasarkan waktu yang ada pada prangkat masing-masing praktikan.
6. Gunakan seluruh fungsi agregat pada field hargaSewa di tabel daftarMobil
7. Gunakan fungsi DAYNAME untuk menampilkan hari pada field tglSewa pada table sewaMobil.

## Evaluasi Praktikum 3:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) $\leq 40$	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) $\geq 86$
1.	Ketepatan dalam memahami fungsi agregat					
2.	Ketepatan dalam memahami fungsi tanggal					
3.	Ketepatan dalam memahami fungsi karakter/string					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_

## MODUL 4 – Query Antar Table & Operator

### A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mampu memahami Query Antar Table dan Operator
2. Mampu dapat menggunakan dan menjalankan Query Antar Table dan Operator

### B. Tugas Pendahuluan

1. Apa yang dimaksud dengan join?
2. Sebutkan dan Jelaskan jenis-jenis Join ?
3. Implementasi jenis-jenis join yang anda sebutkan pada no.2.
4. Apa perbedaan Query Antar Table dengan JOIN?

### C. Instrumen dan Prosedur

#### 1. Instrument

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau di atasnya (sebagai *engine basis data*)

#### 2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

### D. Teori Dasar

#### 1. Key

Terdapat beberapa key pada mysql, diantaranya sebagai berikut:

##### 1. Primary key

Primary Key adalah suatu aturan yang berguna untuk memastikan bahwa setiap baris data pada tabel bersifat unik (berbeda antara baris satu dengan baris lainnya).

##### 2. Foreign key

Foreign Key adalah kolom atau field pada suatu tabel yang berfungsi sebagai kunci tamu dari tabel lain.

#### 2. Query Antar Table

Query antar tabel ialah pernyataan SELECT sederhana untuk menggabungkan beberapa tabel. Anda dapat memanggil lebih dari satu tabel dalam klausa FROM untuk menggabungkan hasil dari beberapa tabel.

**Bentuk Umum:**

```
SELECT table1.column1, table2.column2 FROM table1, table2 WHERE table1.column(PK) = table2.column(FK);
```

Dalam contoh ini notasi dot (table.column) digunakan untuk menentukan kolom tabel itu berasal. Jika kolom tersebut hanya muncul di salah satu tabel yang direferensikan, Anda tidak perlu menyertakan nama field yang memenuhi syarat.

Tabel dipisahkan dalam klausa FROM dengan tanda koma. Query antar tabel dapat menggabungkan tabel sebanyak yang diperlukan.

### 3. Operator

#### 1) Operator Aritmatika

Operator aritmatika ialah operator yang digunakan untuk melakukan proses perhitungan \*, /, %, +, dan -.

#### 2) Operator AND dan OR

##### 1. Logika AND

Logika AND adalah sebuah operator yang berfungsi untuk menampilkan 2 kondisi atau lebih.

##### Bentuk Umum:

```
mysql > SELECT * FROM namatabel WHERE kondisi1 AND kondisi2 AND kondisi3 . . . ;
```

##### 2. Logika OR

Logika OR digunakan untuk memberikan pilihan kondisi untuk di penuhi. Jadi ketika kita menambahkan logika OR maka data yang akan di tampilkan adalah data yang memenuhi semua syarat atau satu persatu syarat.

##### Bentuk Umum:

```
mysql > SELECT * FROM namatabel WHERE kondisi1 OR kondisi2 OR kondisi3 . . . ;
```

#### 3) Operator LIKE

LIKE digunakan ketika kita ingin mencari sebuah data yang hanya diwakili oleh salah satu atau lebih hurufnya saja. Berikut macam-macam operator LIKE yang bisa digunakan:

1. 'S%': Cocok dengan kata yang diawali dengan S, dan diikuti dengan karakter apa saja, contoh: 'S', 'Sa', 'Si', 'Saaaaa', 'Susi', dan 'Sabrina Sari'.
2. 'A\_\_i': Cocok dengan kata yang diawali dengan 'A', diikuti oleh 2 karakter bebas, namun diakhiri dengan i, contoh: 'Andi', 'ardi', 'aaai'.
3. '%e': Cocok dengan seluruh kata dengan panjang berapapun yang diakhiri dengan huruf 'e', contoh: 'Kece', 'Kue', dan 'mie'.
4. '%dia%': Cocok dengan seluruh kata yang mengandung kata 'dia', contoh: 'media', 'kemudian', 'dia', dan 'diaaaa'.

Berikut struktur query dasar untuk pencarian data menggunakan LIKE:

##### Bentuk Umum:

```
mysql > SELECT * FROM namatabel WHERE namakolom LIKE 'operator';
```

#### 4) Operator BETWEEN



Penggunaan Operator BETWEEN Sama dengan AND dan OR, BETWEEN juga salah satu operasi perbandingan pada perintah SELECT. BETWEEN ini digunakan untuk menentukan lebar dari data yang akan di tampilkan. Dimulai dari data terendah yang akan di tampilkan sampai data yang paling tinggi untuk di tampilkan. Pada operator between terdapat beberapa operasi pembandingan yang digunakan, berikut beberapa operasi pembandingan yang bisa di gunakan.

Operator	Penjelasan
=	Sama dengan
< > Atau !=	Tidak sama dengan
<=>	Sama dengan null
<	Kurang dari
<=	Kurang dari atau sama dengan
>	Lebih besar dari
>=	Lebih besar atau sama dengan

**Bentuk Umum:**

```
mysql > SELECT * FROM nama_tabel WHERE field BETWEEN nilai_awal AND nilai_akhir;
```

5) Operator IN dan NOI IN

Operator IN juga merupakan salah satu operator yang bisa di gunakan pada perintah SELECT. Operator IN ini berguna melakukan pencocokan dengan salah satu yang ada pada suatu daftar nilai.

**Bentuk Umum:**

```
mysql > SELECT * FROM namatabel WHERE namakolom IN (katakunci1, katakunci2, katakunci3, ..., katakunci-n);
```

6) Operator LIMIT

LIMIT digunakan untuk membatasi data yang ingin di tampilkan. Berikut Query dasar yang bisa digunakan

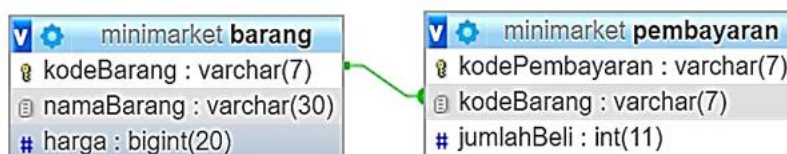
**Bentuk Umum:**

```
mysql > SELECT * FROM namatabel LIMIT jumlahbaris;
```

## E. Kegiatan Praktikum

### 1. Studi Kasus

- Buatlah database dengan nama “minimarket”
- Buatlah table Barang dan Pembayaran



c) Isilah minimal 5 record tiap table

5. barang

kodeBarang	namaBarang	harga
B01	Ultra Milk	7500
B02	Sari Roti	4500
B03	Qtela	6500
B04	Aqua	6500
B05	Milk Tea	6500

6. pembayaran

kodePembayaran	kodeBarang	jumlahBeli
P01	B01	3
P02	B02	5
P03	B03	2
P04	B04	5
P05	B05	2

d) Tampilkan field **kodeBarang**, **namaBarang** dari tabel barang dan **kodePembayaran** dari tabel pembayaran.

```
mysql> SELECT barang.kodeBarang, barang.namaBarang,
pembayaran.kodePembayaran FROM barang, pembayaran WHERE
pembayaran.kodeBarang = barang.kodeBarang;
```

e) Tampilkan totalbayar berdasarkan perkalian **harga** dan **jumlahBeli**.

```
mysql> SELECT pembayaran.kodePembayaran, barang.kodeBarang,
barang.namaBarang, (barang.harga * pembayaran.jumlahBeli) FROM
barang, pembayaran where
```

f) Tampilkan record kedua dan record ketiga dalam 1 perintah

```
mysql> SELECT * from barang where kodeBarang='B01' and
kodeBarang='B02';
```

g) Tampilkan namaBarang yang akhiran hurufnya 'a'

```
mysql> SELECT namaBarang FROM barang WHERE namaBarang LIKE
'%a';
```

**LEMBAR EVALUASI PRAKTIKUM**

1. Buatlah database dengan nama stambuk anda.
2. Buatlah minimal 3 tabel.
3. Isilah tiap record minimal 4 record.
4. Implementasikan query antar table pada ke tiga tabel anda. Dengan ketentuan menggunakan minimal satu operator aritmatika.
5. Terapkan operator BETWEEN, IN, LIKE, dan LIMIT pada tabel anda.

Evaluasi Praktikum 4:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) $\leq 40$	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) $\geq 86$
1.	Ketepatan dalam memahami Query Antar Table					
2.	Ketepatan dalam memahami operator					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_

## MODUL 5 – Operator Join

### A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mampu memahami operator join
2. Mampu dapat menggunakan dan menjalankan jenis-jenis operator join

### B. Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan Sub Query !
2. Tuliskan fungsi dari Sub Query ?
3. Tuliskan dan jelaskan tentang operator IN, ANY, EXISTS, dan ALL !
4. Tuliskan contoh penggunaan operator IN, ANY, EXISTS, dan ALL !

### C. Instrumen dan Prosedur

#### 1. Instrument

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau di atasnya (sebagai *engine basis data*)

#### 2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

### D. Teori Dasar

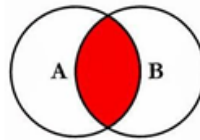
#### 1. Definisi Join

JOIN adalah kombinasi record dari dua tabel atau lebih di dalam basis data relasional dan menghasilkan sebuah tabel (Temporary) baru. JOIN menggunakan perintah SELECT.

#### 2. Macam-macam Join

##### 1) INNER JOIN

Inner join digunakan untuk menampilkan data lebih dari atau sama dengan 2 tabel yang saling berelasi satu sama lain. Tipe join ini akan mengambil semua row dari table sumber dan table tujuan dengan kondisi nilai key yang terkait saja. Inner join terdapat tabel A dan B, maka hasil inner join dapat diperhatikan sebagai bidang pada gambar dibawah.

**Bentuk Umum:**

```
mysql > SELECT field1, field2, ... FROM tabel1 INNER JOIN tabel2 ON
tabel1.kunci_utama = tabel2.kunci_tamu;
```

## 2) NATURAL JOIN

Menggabungkan output isi kedua table berdasarkan nama kolom yang sama tanpa suatu kondisi.

**Bentuk Umum:**

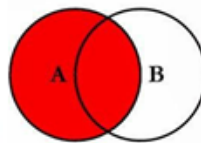
```
mysql > SELECT * FROM tabel1 NATURAL JOIN tabel2;
```

## 3) OUTER JOIN

Outer join terbagi ke dalam dua jenis, yaitu left outer join dan right outer join.

**1. Left Outer Join**

Left Outer join digunakan untuk menampilkan semua data dari tabel sebelah kiri ditambah data tabel sebelah kanan yang sesuai. (atau NULL jika tidak ada data yang sesuai) Leftouter join antara tabel A dan B dapat diilustrasikan pada gambar dibawah ini

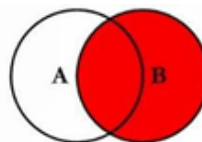
**Bentuk Umum:**

```
mysql > SELECT field1, field2, ... FROM tabel1 LEFT JOIN tabel2 ON
tabel1.kunci_utama = tabel2.kunci_tamu;
```

**2. Right Outer Join**

Right outer join merupakan kebalikan dari left join, digunakan untuk menampilkan semua data dari tabel sebelah kanan ditambah data dari tabel sebelah kiri yang sesuai. (atau NULL jika tidak ada data yang sesuai)

Leftouter join antara tabel A dan B dapat diilustrasikan pada gambar dibawah ini

**Bentuk Umum:**

```
mysql > SELECT tabel.field1, tabel.field2, ... FROM tabel1 RIGHT JOIN
tabel2 ON tabel1.kunci_utama = tabel2.kunci_tamu;
```

## 4) UNION JOIN

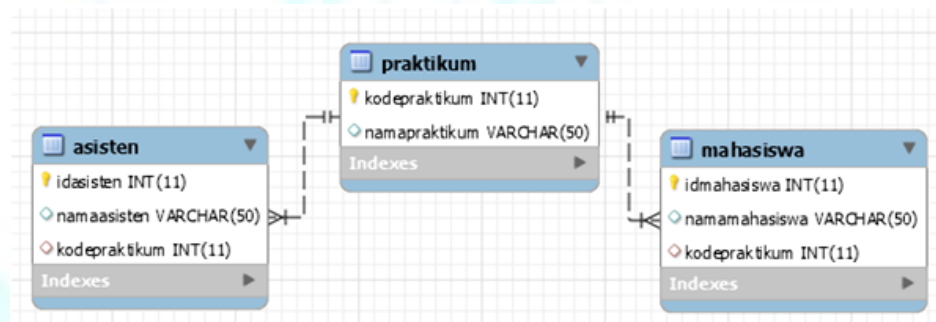
Union Join digunakan untuk menggabungkan dua atau lebih perintah select dari beberapa tabel sehingga menghasilkan suatu gabungan. Union akan menampilkan semua isi dari tabel namun akan menghapus record yang duplikat dari hasil gabungan.

**Bentuk Umum:**

```
mysql > SELECT field1, field2, .. FROM tabel1 UNION SELECT field1, field2,
... FROM tabel2;
```

**E. Kegiatan Praktikum****1. Studi Kasus**

- 1) Buatlah database dengan nama “labfik”
- 2) Buatlah table praktikum, asisten, dan praktikan



- 3) Istilah minimal 3 record tiap table

## 1. praktikum

kodepraktikum	namapraktikum
1	Prak Jaringan
2	Prak Java Fundamental
3	Prak Basis Data 1
4	Prak Pemrograman Web

## 2. praktikan

idmahasiswa	namamahasiswa	kodepraktikum
1	Kiki Sabina	3
2	Abdul Rozak	1
3	Rezki	4

## 3. asisten

idasisten	namaasisten	kodepraktikum
1	Ririn	1
2	Wahyuni	4
3	Mutmainnah	3
4	Bowo	3
5	Fadlan	2



4) Lakukan perintah berikut:

1. Inner Join

```
mysql > SELECT Praktikan.namaPraktikan, praktikum.namapraktikum
FROM Praktikan INNER JOIN praktikum on praktikum.kodepraktikum
= Praktikan.kodepraktikum;
```

namamahasiswa	namapraktikum
Abdul Rozak	Prak Jaringan
Kiki Sabina	Prak Basis Data 1
Rezki	Prak Pemrograman Web

2. Natural Join

```
mysql > SELECT * FROM praktikum NATURAL JOIN asisten;
```

kodepraktikum	namapraktikum	idasisten	namaasisten
1	Prak Jaringan	1	Ririn
4	Prak Pemrograman Web	2	Wahyuni
3	Prak Basis Data 1	3	Mutmainnah
3	Prak Basis Data 1	4	Bowo
2	Prak Java Fundamental	5	Fadlan

3. Left Outer Join

```
mysql > SELECT * FROM praktikum LEFT JOIN Praktikan ON
Praktikan.kodepraktikum = praktikum.kodepraktikum;
```

kodepraktikum	namapraktikum	idmahasiswa	namamahasiswa	kodepraktikum
3	Prak Basis Data 1	1	Kiki Sabina	3
1	Prak Jaringan	2	Abdul Rozak	1
4	Prak Pemrograman Web	3	Rezki	4
2	Prak Java Fundamental	NULL	NULL	NULL

4. Right Outer Join

```
mysql > SELECT * FROM Praktikan RIGHT JOIN praktikum ON
Praktikan.kodepraktikum = praktikum.kodepraktikum;
```

idmahasiswa	namamahasiswa	kodepraktikum	kodepraktikum	namapraktikum
1	Kiki Sabina	3	3	Prak Basis Data 1
2	Abdul Rozak	1	1	Prak Jaringan
3	Rezki	4	4	Prak Pemrograman Web
NULL	NULL	NULL	2	Prak Java Fundamental

5. Union Join

```
mysql > SELECT idPraktikan, namaPraktikan FROM Praktikan
union SELECT idasisten, namaasisten FROM asisten;
```

idmahasiswa	namamahasiswa
1	Kiki Sabina
2	Abdul Rozak
3	Rezki
1	Ririn
2	Wahyuni
3	Mutmainnah
4	Bowo
5	Fadlan

### LEMBAR EVALUASI PRAKTIKUM

1. Buatlah database berbeda dengan praktikan lain.
2. Minimal 3 tabel yang dapat mengimplementasikan perintah JOIN.
3. Disetiap table minimal 5 record.
4. Minimal menggunakan 4 join.
5. Jelaskan setiap JOIN yang digunakan pada kasus masing-masing.

## Evaluasi Praktikum 5:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) $\leq 40$	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) $\geq 86$
1.	Ketepatan dalam memahami inner join					
2.	Ketepatan dalam memahami natural join					
3.	Ketepatan dalam memahami left join					
4.	Ketepatan dalam memahami right join					
5.	Ketepatan dalam memahami union join					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_

## MODUL 6 – Sub Query

### A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mampu memahami sub query
2. Mampu dapat menggunakan dan menjalankan sub query

### B. Tugas Pendahuluan

1. Apa yang di maksud dengan normalisasi ?
2. Mengapa harus melakakun normalisasi pada ?
3. Tuliskan dan jelaskan tahap-tahap normalisasi dari 1NF sampai 3NF ?

### C. Instrumen dan Prosedur

#### 1. Instrument

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau diatasnya (sebagai *engine basis data*)

#### 2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

### D. Teori Dasar

#### 1. Sub Query

Sub query atau query nested adalah query dalam query. Sub query digunakan ketika kita ingin mencari nilai dari suatu tabel dengan perintah SELECT tapi kita tidak mengetahui nilainya dengan pasti atau nilai yang dicari memiliki banyak kondisi.

Adapaun kegunaan dari sub Query adalah:

1. Sub Query digunakan untuk menyelesaikan persoalan dimana terdapat suatu nilai yang tidak diketahui.
2. Meng-Copy data dari satu tabel ke tabel lain
3. Menerima data dari inline view
4. Mengambil data dari tabel lain untuk di update ke tabel yang dituju
5. Menghapus baris dari satu tabel berdasarkan baris dari tabel lain.

**Bentuk Umum:**

```
SELECT SELECT_LIST FROM table WHERE EXPRESI OPERATOR (Select
Select_List From table Where EXPRESI OPERATOR 'value');
```

## 2. Operator IN

Operator In digunakan untuk menentukan beberapa nilai dalam clausa WHERE

**Bentuk Umum:**

```
mysql> SELECT nama_field FROM table_name WHERE nama_field IN (subquery);
```

## 3. Operator Exists

Untuk menyaring tabel yang didasarkan pada keberadaan nilai kolom suatu tabel dalam subquery. EXIST bernilai True jika subquery telah menghasilkan setidaknya satu record data, dan bernilai false jika tidak ada record yang didapatkan.

**Bentuk Umum:**

```
mysql > SELECT nama_field FROM nama_tabel EXISTS (subquery);
```

## 4. Operator ANY (SOME)

Operator Any hampir sama penggunaannya seperti Exists. Operator ANY diikuti dengan operator perbandingan memiliki arti menampilkan nilai yang sesuai dengan apapun yang dihasilkan oleh sub query. ANY berbeda dengan IN, jika IN itu semua data, sedangkan ANY hanya beberapa data.

**Bentuk Umum:**

```
mysql> SELECT nama_field FROM namatabel WHERE kondisi opeatorperbandingan ANY (subquery);
```

## 5. Operator ALL

Operator all digunakan untuk melakukan perbandingan dengan sub query. Kondisi dengan all menghasilkan nilai benar jika perbandingan menghasilkan benar untuk setiap nilai dalam sub query.

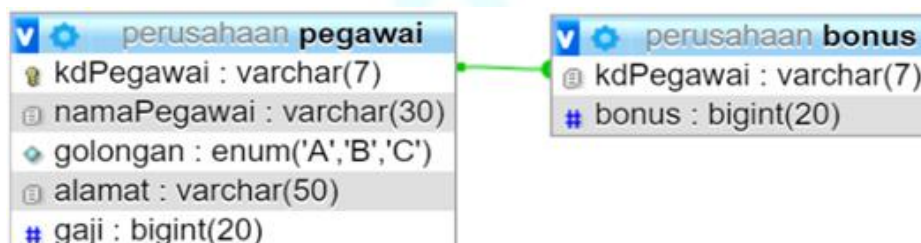
**Bentuk Umum:**

```
mysql > SELECT namakolom FROM namatabel WHERE kondisi opeatorperbandingan ALL (subquery);
```

## E. Kegiatan Praktikum

### 1. Studi Kasus

- 1) Buatlah database dengan nama **perusahaan**
- 2) Buatlah table dengan nama **pegawai** dan **bonus**



3) Isilah masing-masing table tersebut

1. pegawai

kdPegawai	namaPegawai	golongan	alamat	gaji
P01	akhlaqul	A	BTP	2000000
P02	Jamil	B	Antang	1500000
P03	Putri	A	Panakukang	1500000
P04	Ansari	C	BTP	1000000
P05	Abdurauf	B	BTP	1000000

2. bonus

kdPegawai	bonus
P01	300000
P03	400000
P05	600000

4) Lakukan perintah di bawah ini

1. Operator IN

```
mysql > SELECT barang.kodeBarang, barang.namaBarang,
pembayaran.kodePembayaran FROM barang, pembayaran
WHERE pembayaran.kodeBarang = barang.kodeBarang;
```

2. Operator EXISTS

```
mysql > select * from pegawai where EXISTS (select kdPegawai
from bonus where kdPegawai=pegawai.kdPegawai);
```

3. Operator ANY (SOME)

```
mysql > select namaPegawai,gaji from pegawai where gaji > any
(select gaji from pegawai);
```

4. Operator ALL

```
mysql > select namaPegawai,gaji from pegawai where gaji < all
(select gaji from pegawai where gaji > 1300000);
```

## LEMBAR EVALUASI PRAKTIKUM

1. Tetap gunakan database **perusahaan** dikegiatan praktikum mandiri
2. Isi table berdasarkan table pada praktikum mandiri.
3. Kemudian silahkan praktikan mencoba perintah Sub Query yang berbeda dengan Operator dibawah ini:
  - a. Operator IN
  - b. Operator EXISTS



- c. Operator ANY(SOME)
- d. Operator ALL
4. Buatlah table dengan nama **totalGaji** dan field kdPegawai varchar(7) PK, nama varchar(30), totalGaji int
5. Copy record field kdPegawai dan namaPegawai pada table pegawai ke table data. Dimana field totalGaji didapatkan dari field gaji pada table pegawai ditambah dengan field bonus pada table bonus.

Evaluasi Praktikum 6:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan dalam memahami sub query					
2.	Ketepatan dalam memahami operator IN pada sub query					
3.	Ketepatan dalam memahami operator exists pada sub query					
4.	Ketepatan dalam memahami operator any pada sub query					
5.	Ketepatan dalam memahami operator all pada sub query					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_

## MODUL 7 – Normalisasi

### A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mampu mendeskripsikan teknik Normalisasi
2. Mampu menggunakan dan menerapkan normalisasi pada database

### B. Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan DCL (*Data Control Language*)
2. Berikan masing-masing sebuah contoh implementasi perintah-perintah DCL !

### C. Instrumen dan Prosedur

#### 1. Instrument

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau di atasnya (sebagai *engine basis data*)

#### 2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

### D. Teori Dasar

#### 1) Defenisi Normalisasi

Normalisasi merupakan sebuah teknik dalam mendesain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redudansi) sehingga basis data tersebut dapat bekerja dengan optimal.

#### 2) Tujuan Normalisasi:

5. Untuk menghilangkan dan mengurangi redudansi data.
6. Untuk memastikan data berada pada table yang tepat.
7. Mengurangi kompleksitas.
8. Mempermudah modifikasi data.

#### 3) Tahapan Normalisasi:

Tahapan normalisasi dimulai dari tahap 1NF sampai tahap 5NF. Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel – tabel yang berkualitas baik. Urutan normalisasi 1NF, 2NF, 3NF, BCNF, 4NF, 5NF.

#### 4) Bentuk Normalisasi

1. Normal Pertama (1NF)
  - Aturan:

1. Mendefinisikan atribut kunci
  2. Tidak adanya grup berulang (maksudnya)
  3. Semua atribut bukan kunci tergantung pada atribut kunci
2. Normal Kedua (2NF)
- Aturan:
    1. Sudah memenuhi bentuk normal pertama
    2. Tidak ada ketergantungan parsial (dimana seluruh field hanya tergantung pada sebagian field kunci)
3. Normal Ketiga (3NF)
- Aturan:
    1. Sudah berada dalam bentuk normal kedua
    2. Tidak ada ketergantungan transitif (dimana field bukan kunci tergantung pada field bukan kunci lainnya).

## E. Kegiatan Praktikum

### 1. Studi Kasus

Bentuk ABNORMAL

IdBuku	JudulBuku	TglTerbit
B01	Basis Data	2011-10-02
B02	Struktur Data	2011-05-03
B03	Java Lanjut	2011-04-06

IdBuku	JudulBuku	TglTerbit	IdPenyakit	NamaPenyakit	AlamatPenyakit	idPeminjaman	namaPeminjaman	AlamatPeminjaman
B01	Basis Data	2011-10-02	P01	Gramedia	Makassar	PE01	Alifia	Makassar
B02	Struktur Data	2011-05-03	P02	Grahamedia	Makassar	PE02	Ahmad	Makassar
B03	Java Lanjut	2011-04-06	P03	Grahamedia	Makassar	PE03	Jamil	Makassar
B04	Akhlaq	2011-07-24	P04	Gramedia	Makassar	PE04	Ayu	Makassar

- 1) Normalisasi Pertama (1NF)
- Menghilangkan duplikasi kolom dari table yang sama
  - Buat table terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik (primary key)

<b>IdPenerbit</b>	<b>NamaPenerbit</b>	<b>AlamatPenerbit</b>
P01	Gramedia	Makassar
P02	Grahamedia	Makassar
P03	Grahamedia	Makassar
P04	Gramedia	Makassar

<b>IdPeminjaman</b>	<b>namaPeminjaman</b>	<b>AlamatPeminjaman</b>
PE01	Alifia	Makassar
PE02	Ahmad	Makassar
PE03	Jamil	Makassar
PE04	Ayu	Makassar

## 2) Normalisasi Kedua (2NF)

- Menghapus beberapa subset data yang ada pada table dan menempatkan mereka pada table terpisah.
- Menciptakan hubungan antara table baru dan table lama dengan menciptakan foreign key.
- Tidak ada atribut dalam table yang secara fungsional bergantung pada candidate key table tersebut. Setelah dalam bentuk 1NF:

<b>IdTrx</b>	<b>JudulBuku</b>	<b>IdBuku</b>	<b>IdPeminjam</b>	<b>IdPenerbit</b>	<b>NamaPenerbit</b>
T01	Basis Data	B01	PE01	P01	Gramedia
T02	Akhlaq	B02	PE02	P02	Grahamedia

Dalam bentuk 2NF:

<b>IdTrx</b>	<b>IdBuku</b>	<b>IdPeminjam</b>	<b>IdPenerbit</b>
T01	B01	PE01	P01
T02	B02	PE02	P02

## 3) Normalisasi Ketiga 3NF:

- Memenuhi semua persyaratan dari bentuk normal kedua.
- Menghapus kolom yang tidak tergantung pada primary key.
- Tidak semua kasus atau table dapat kita sesuaikan dengan berbagai bentuk normalisasi ini.

Contoh:

<b>idOrder</b>	<b>idCust</b>	<b>Harga</b>	<b>Jumlah</b>	<b>Total</b>
D01	C01	5000	2	10000
D02	C02	7000	3	21000

Dalam bentuk 3NF:

<b>idOrder</b>	<b>idCust</b>	<b>Harga</b>	<b>Jumlah</b>
D01	C01	5000	2
D02	C02	7000	3

**LEMBAR EVALUASI PRAKTIKUM**

1. Carilah nota pembelian resmi.
2. Buatlah satu table abnormal dari nota tersebut.
3. Normalisasikan table tersebut dan buat table tersebut ke dalam database.

Evaluasi Praktikum 7:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) $\leq 40$	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) $\geq 86$
1.	Ketepatan dalam memahami normalisasi					
2.	Ketepatan dalam mengetahui bentuk normal dan tidak normal					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_

**MODUL 8 – DCL (*Data Control Language*)****A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mampu menjelaskan perintah-perintah DCL (*Data Control Language*)
2. Mampu menggunakan perintah-perintah DCL (*Data Control Language*) pada Aplikasi DBMS (MySQL)

**B. Instrumen dan Prosedur****1. Instrument**

- a) Perangkat keras / PC / Laptop
- b) Sistem operasi Windows / Linux
- c) MySql versi 8.0 atau di atasnya (sebagai *engine basis data*)

**2. Prosedur**

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

**C. Teori Dasar**

*Data Control Language* (DCL) adalah salah satu dari kelompok perintah SQL yang digunakan untuk melakukan kontrol terhadap *privilege* atau hak akses khusus untuk berinteraksi dengan *database*. Hak akses khusus ini diperlukan sebagai prasyarat bagi setiap *user database* untuk melakukan berbagai aksi di *database*, seperti: membuat *object*, menghapus *object*, mengubah *object*, menampilkan hasil *query*, dan seterusnya.

Sederhananya, setiap pengguna *database* hanya dapat melakukan aksi-aksi yang sudah diberikan oleh *user* dengan kontrol tertinggi didalam *database* tersebut. Misalnya, pengguna A hanya diberikan akses untuk membuat dan menampilkan sesuatu di *database*. Maka, pengguna tersebut hanya dapat melakukan perintah tersebut saja, tidak bisa melakukan ubah data, hapus data, dan seterusnya. Hal ini menjadi penting untuk diketahui karena di dalam dunia kerja dengan banyak pengguna *database*, manajemen pembagian hak akses seperti ini sangat krusial dan berpengaruh terhadap integritas dan keamanan data.

Berikut ini adalah beberapa perintah DCL:

**a) GRANT**

Perintah *Grant* digunakan untuk memberikan hak akses khusus kepada *user* untuk melakukan aksi tertentu pada *object* tertentu di *database*.

Sintaks:



```
GRANT tipe_privilege
ON nama_objek
TO nama_pengguna;
```

Keterangan:

- *tipe\_privilege* — diisi dengan privilege atau hak akses yang dimaksud (lihat tabel di bawah).
- *nama\_objek* — diisi dengan objek-objek database, seperti: database, table, view, trigger, dan seterusnya.
- *nama\_pengguna* — diisi dengan nama pengguna database yang ingin diberikan privilege.

<b>Tipe Privilege</b>	<b>Keterangan</b>
<i>ALL PRIVILEGE</i>	Memberikan seluruh akses secara penuh ( <i>full access</i> )
<i>CREATE</i>	Memberikan hak akses untuk membuat <i>database</i> atau objek <i>database</i>
<i>DROP</i>	Memberikan hak akses untuk menghapus <i>database</i> atau objek <i>database</i>
<i>SELECT</i>	Memberikan hak akses untuk melakukan <i>query</i> ke <i>database</i>
<i>INSERT</i>	Memberikan hak akses untuk menambahkan <i>record</i> / data baru ke tabel
<i>UPDATE</i>	Memberikan hak akses untuk mengubah <i>record</i> / data di tabel
<i>DELETE</i>	Memberikan hak akses untuk menghapus <i>record</i> / data di tabel

#### b) *REVOKE*

Perintah *Revoke* digunakan untuk menarik kembali / mencabut / membatalkan hak akses khusus yang sebelumnya telah diberikan melalui perintah *Grant*.

Sintaks:

```
REVOKE tipe_privilege
ON nama_objek
FROM nama_pengguna;
```

Keterangan:

- *tipe\_privilege* — diisi dengan privilege atau hak akses yang dimaksud (lihat tabel di bagian atas).
- *nama\_objek* — diisi dengan objek-objek database, seperti: database, table, view, trigger, dan seterusnya.
- *nama\_pengguna* — diisi dengan nama pengguna database yang ingin dicabut kembali hak akses atau privilege-nya.

## D. Kegiatan Praktikum

### 1. Studi Kasus

Sebelum menggunakan perintah *Grant*, masuklah terlebih dahulu sebagai *user root* dengan menggunakan perintah:

```
mysql > cd xampp/bin/
mysql > mysql -u root -p
Enter password :
```

Setelah itu kita akan membuat user baru dengan nama ‘DosenIT’ dengan password ‘dosenit.com’. Gunakan perintah di bawah ini:

```
MariaDB [db_kampus]> CREATE USER 'DosenIT'@'localhost' IDENTIFIED BY 'dosenit.com';
Query OK, 0 rows affected (0.04 sec)
```

Asumsikan bahwa kita akan memberikan akses database kepada user tersebut. Misalnya kita akan membuat database bernama “db\_kampus”.

```
MariaDB [(none)]> create database db_kampus;
Query OK, 1 row affected (0.02 sec)

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

Setelah database-nya berhasil dibuat, langkah selanjutnya adalah memberikan akses tersebut kepada user tersebut. Untuk keperluan tersebut, kita dapat menggunakan perintah *GRANT* mengikuti sintaks yang telah dijabarkan di atas. Misalnya saat ini kita hanya akan memberikan hak akses untuk membuat, menambahkan, dan melakukan query terhadap objek-objek di database “db\_kampus”. Maka gunakanlah perintah di bawah ini:

```
MariaDB [db_kampus]> GRANT CREATE, INSERT, SELECT
-> ON db_kampus.*
-> TO 'DosenIT'@'localhost';
Query OK, 0 rows affected (0.01 sec)

MariaDB [db_kampus]>
MariaDB [db_kampus]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)
```

Sebagai catatan, perintah *FLUSH PRIVILEGES* digunakan setiap kali selesai memberikan atau menarik kembali hak akses user.

Setelah itu, mari kita login kembali ke *mysql* menggunakan akun baru tersebut. Lakukan dengan menggunakan perintah di bawah ini:

```

C:\xampp\mysql\bin>mysql -u DosenIT -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.1.38-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)]> use db_kampus;
Database changed

```

Setelah berhasil masuk dan menggunakan database db\_kampus, cobalah perintah-perintah yang berhubungan dengan CREATE, INSERT, dan SELECT. Silakan kreasikan sebebas mungkin yang Anda inginkan. Anda juga boleh menggunakan perintah-perintah di bawah ini untuk bahan uji coba.

```

MariaDB [db_kampus]> create table mahasiswa(
    -> id int auto_increment primary key,
    -> nim varchar(12),
    -> nama varchar(30));
Query OK, 0 rows affected (0.06 sec)

```

```

MariaDB [db_kampus]> desc mahasiswa;

```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nim	varchar(12)	YES		NULL	
nama	varchar(30)	YES		NULL	

3 rows in set (0.03 sec)

```

MariaDB [db_kampus]> insert into mahasiswa values
    -> (NULL, '2000001', 'Satoshi'),
    -> (NULL, '2000002', 'Serena');
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

```

```

MariaDB [db_kampus]> select *from mahasiswa;

```

id	nim	nama
1	2000001	Satoshi
2	2000002	Serena

2 rows in set (0.00 sec)

Perintah-perintah yang berhubungan dengan CREATE, INSERT, dan SELECT akan berhasil dieksekusi di user ini dikarenakan telah diberikan akses sebelumnya oleh user root menggunakan perintah GRANT. Tetapi bagaimana dengan UPDATE, DELETE,

dan lainnya yang tidak didefinisikan sebelumnya? Silakan dicoba menggunakan kreasi perintah sendiri atau menggunakan contoh di bawah ini.

```
MariaDB [db_kampus]> update mahasiswa set nama ='Serena Kalos Queen'
-> where nim ='2000002';
ERROR 1142 (42000): UPDATE command denied to user 'DosenIT'@'localhost:
ahasiswa'
MariaDB [db_kampus]> delete from mahasiswa where nim = '2000001';
ERROR 1142 (42000): DELETE command denied to user 'DosenIT'@'localhost:
ahasiswa'
```

Hasilnya adalah error.

### Penggunaan REVOKE

Untuk contoh penggunaannya, mari kita tarik kembali hak akses dari user DosenIT.

```
MariaDB [db_kampus]> exit
Bye
```

```
MariaDB [(none)]> REVOKE ALL PRIVILEGES ON db_kampus.* FROM 'DosenIT'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

## LEMBAR EVALUASI PRAKTIKUM

1. Buatlah user baru dengan nama 'fakultasIT' dengan password 'fakultasit.com'.
2. Buatlah database dengan nama 'teknikinformatika' dengan nama table 'mahasiswa' yang memiliki struktur table: stambuk int, nama varchar(30), jenisKelamin enum('L','P'), alamat varchar(100).
3. Disetiap table minimal 5 record
4. Lakukan perintah GRANT dan REVOKE pada database yang telah dibuat.

## Evaluasi Praktikum 8:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) $\leq 40$	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) $\geq 86$
1.	Pemahaman perintah <i>GRANT</i>					
2.	Pemahaman perintah <i>REVOKE</i>					

Catatan Asisten:

Asisten 1 : \_\_\_\_\_

Asisten 2 : \_\_\_\_\_