

# **BASIS DATA**

**SQL**  
**(Structured Query Language)**

Disusun oleh:  
**AMELIA YUSNITA**

# MATERI

1. Pengertian SQL
2. Tugas Utama SQL
3. Konsep Dasar SQL
4. Perintah-Perintah SQL



# PENGERTIAN SQL

**SQL (Structured Query Language)** adalah bahasa yang digunakan untuk mengelola dan memanipulasi data di dalam basis data relasional.

SQL pertama kali dikembangkan oleh IBM pada 1970-an dan sejak itu menjadi Bahasa standar dalam pengelolaan basis data.

**DBMS (Database Management System)** : adalah perangkat lunak yang menyediakan kerangka kerja untuk menyimpan, mengelola, dan mengamankan data. seperti MySQL, PostgreSQL, Oracle, SQL Server

Dengan kata lain, SQL adalah bahasa yang digunakan untuk berkomunikasi dengan DBMS, sedangkan DBMS adalah sistem yang menjalankan instruksi SQL untuk mengelola data

# TUGAS UTAMA SQL

Beberapa tugas utama yang dapat dilakukan dengan SQL adalah:

1. Menyimpan data di dalam basis data
2. Mengambil data dari basis data
3. Memperbarui data dalam basis data
4. Menghapus data dari basis data
5. Mengelola hak akses ke basis data

# KONSEP DASAR SQL

Dalam SQL, data disimpan dalam bentuk tabel (table). Tabel terdiri dari baris (row) dan kolom (column), di mana setiap kolom menyimpan atribut tertentu, dan setiap baris menyimpan satu entri data.

## Elemen Penting dalam SQL:

1

### **Tabel:**

Struktur data yang menyimpan data dalam format baris dan kolom

2

### **Baris (Row):**

Setiap entri dalam tabel, mewakili satu rekaman

3

### **Kolom (Column):**

Atribut atau tipe data dalam tabel.

4

### **Primary Key:**

Identifikasi unik untuk setiap baris dalam tabel.

5

### **Foreign Key:**

Kolom yang menghubungkan tabel satu dengan tabel lain untuk menjaga integritas referensial.

# PERINTAH-PERINTAH SQL

SQL dibagi menjadi beberapa kategori perintah:

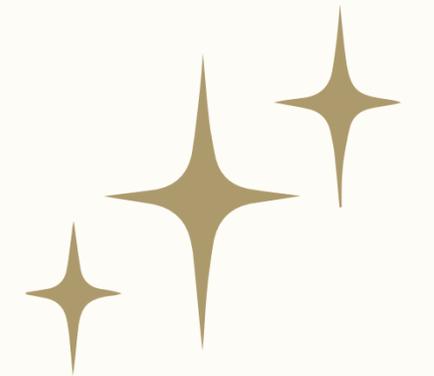
1. **Data Definition Language (DDL)** : DDL digunakan untuk mendefinisikan struktur basis data.
2. **Data Manipulation Language (DML)**: DML digunakan untuk memanipulasi data di dalam tabel.
3. **Data Query Language (DQL)** : DQL digunakan untuk mengambil data dari basis data
4. **Data Control Language (DCL)** : DCL digunakan untuk mengontrol akses pengguna terhadap database.
5. **Transaction Control Language (TCL)** : TCL digunakan untuk mengelola transaksi dalam database.

# A. DATA DEFINITION LANGUAGE (DDL)



**DDL** adalah Kumpulan perintah SQL yang berkaitan dengan pembuatan, perubahan, dan penghapusan database

**Perintah-perintah DDL meliputi:**

1. **CREATE** : Perintah CREATE digunakan untuk membuat database dan objek-objek di dalam database
  2. **ALTER** : Perintah ALTER digunakan untuk mengubah database dan objek-objek di dalam database
  3. **DROP** : Perintah DROP digunakan untuk menghapus database dan objek-objek di dalam database
- 
- 

# 1.CREATE

**CREATE** : Perintah CREATE digunakan untuk membuat database dan objek-objek di dalam database

## A. MEMBUAT DATABASE

```
CREATE DATABASE Pendidikan;
```

Pada contoh di jelaskan, perintah CREATE DATABASE digunakan untuk membuat Database bernama "Pendidikan"

## B. MEMBUAT TABEL

```
CREATE TABLE Mahasiswa (  
  ID_Mahasiswa INT PRIMARY KEY,  
  Nama VARCHAR(100),  
  Jurusan VARCHAR(50),  
  Tahun_Masuk INT  
);
```

Pada contoh di jelaskan, perintah CREATE TABLE digunakan untuk membuat tabel bernama "Mahasiswa" dengan kolom:

- ID\_Mahasiswa: sebagai primary key dengan tipe data integer
- Nama: tipe data Varchar dengan panjang maksimal 100 karakter.
- Jurusan: tipe data Varchar dengan panjang maksimal 50 karakter.
- Tahun\_Masuk: tipe data integer yang menyimpan tahun.

## 2. ALTER

**ALTER** : Perintah ALTER digunakan untuk mengubah database dan objek-objek di dalam database



### A. MENAMBAH KOLOM

```
ALTER TABLE Mahasiswa  
ADD Alamat VARCHAR(255);
```

Contoh ini merupakan perintah untuk menambahkan kolom baru "Alamat" pada tabel "Mahasiswa".

### B. MENGUBAH KOLOM

```
ALTER TABLE mahasiswa  
CHANGE id_mahasiswa NIM Int(5);
```

Contoh ini merupakan perintah untuk mengubah nama kolom/field "Id\_Mahasiswa" menjadi NIM dengan tipe data INTEGER Panjang karakter maksimal = 5



### C. MENGUBAH TIPE DATA KOLOM

```
ALTER TABLE Mahasiswa  
MODIFY Nama VARCHAR(150);
```

Pada contoh ini, tipe data dari kolom Nama diubah dari panjang 100 karakter menjadi 150 karakter.

### E. MENGHAPUS KOLOM

```
ALTER TABLE mahasiswa  
DROP alamat;
```

Pada contoh ini, merupakan perintah untuk menghapus kolom/filed “Alamat” pada tabel “Mahasiswa”

### E. MENGUBAH NAMA TABEL

```
ALTER TABLE mahasiswa RENAME TO mahasiswi;
```

Contoh ini merupakan perintah untuk mengubah tabel “Mahasiswa” menjadi “Mahasiswi”

# 3. DROP

**DROP** : Perintah DROP digunakan untuk menghapus database dan objek-objek di dalam database

## A. MENGHAPUS TABEL

```
DROP TABLE Mahasiswa;
```

Contoh di akan menghapus tabel "Mahasiswa" beserta semua data dan strukturnya dari basis data.

## B. MENGHAPUS DATABASE

```
DROP DATABASE Akademik;
```

 Contoh ini merupakan perintah untuk menghapus databases "Akademik"

## B. DATA MANIPULATION LANGUAGE (DML)

**DML** memungkinkan pengguna untuk melakukan manipulasi data di dalam tabel seperti memasukkan, mengubah, dan menghapus data

**Perintah-perintah DML meliputi:**

1. **INSERT:** Perintah INSERT digunakan untuk menambahkan baris baru ke dalam tabel
2. **UPDATE :** Perintah UPDATE digunakan untuk mengubah data yang sudah ada di dalam tabel.
3. **DELETE :** Perintah DELETE digunakan untuk menghapus data dari tabel.

# 1. INSERT

**INSERT:** Perintah INSERT digunakan untuk menambahkan baris baru ke dalam tabel. Anda dapat menambahkan satu atau lebih baris sekaligus.

## A. MENAMBAH DATA KEDALAM TABEL

```
INSERT INTO mahasiswa (ID_Mahasiswa, nama, jurusan, tahun_masuk)
VALUES (101, 'Rina Andayani', 'Teknik Informatika', 2023);
```

Perintah ini untuk menambahkan data mahasiswa dengan ID 101, nama "Rina Andayani", jurusan "Teknik Informatika", dan tahun masuk 2023.

## 2. UPDATE

**UPDATE** : Perintah UPDATE digunakan untuk mengubah data yang sudah ada di dalam tabel.

### A. MENGUBAH DATA DALAM TABEL

**Mengubah data jurusan mahasiswa dengan ID\_Mahasiswa = 101.**

```
UPDATE mahasiswa  
SET jurusan = 'Sistem Informasi'  
WHERE ID_Mahasiswa = 101;
```

Penjelasan ini adalah Perintah untuk mengubah jurusan mahasiswa dengan ID 101 dari "Teknik Informatika" menjadi "Sistem Informasi".

# 3. DELETE

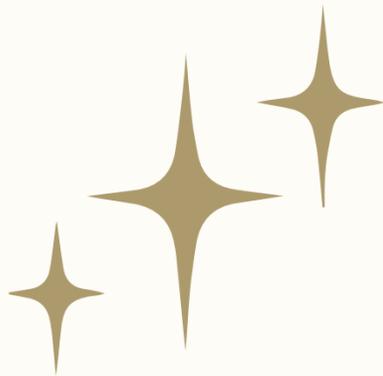


**DELETE** : Perintah DELETE digunakan untuk menghapus data dari tabel.

## MENGHAPUS DATA DARI DALAM TABEL

```
DELETE FROM mahasiswa  
WHERE ID_Mahasiswa = 101;
```

Penjelasan: Perintah di atas menghapus data mahasiswa dengan ID 101 dari tabel mahasiswa.



# PERBEDAAN DROP, dan DELETE

## DROP

**Menghapus tabel beserta seluruh data dan struktur tabelnya.**

## DELETE

**Menghapus data tertentu dalam tabel sesuai dengan kondisi yang diberikan (tanpa menghapus struktur tabel), tetapi transaksi ini dapat di-rollback (dikembalikan) jika sedang dalam transaksi**

# C. DATA QUERY LANGUAGE (DQL)

**Data Query Language (DQL)** adalah bagian dari SQL yang digunakan untuk mengambil data dari database.

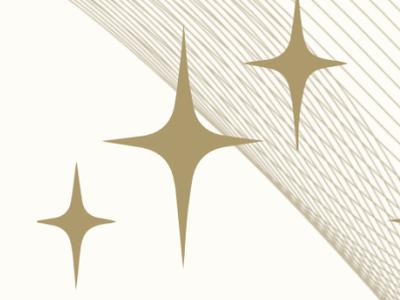
Perintah utama dalam DQL adalah **SELECT**, yang memungkinkan kita untuk menampilkan data dari satu atau lebih tabel berdasarkan kriteria tertentu.

## Contoh Kasus: Data Mahasiswa

### 1. Menampilkan Seluruh Data Mahasiswa

```
SELECT *  
FROM mahasiswa;
```

Hasil yang diperoleh adalah semua kolom dari tabel mahasiswa.



## 2. Menampilkan Nama dan Jurusan Mahasiswa

Jika kita hanya ingin menampilkan kolom nama dan jurusan, perintahnya adalah:

```
SELECT nama, jurusan  
FROM mahasiswa;
```

Ini akan menampilkan daftar nama dan jurusan mahasiswa tanpa menampilkan kolom lainnya.

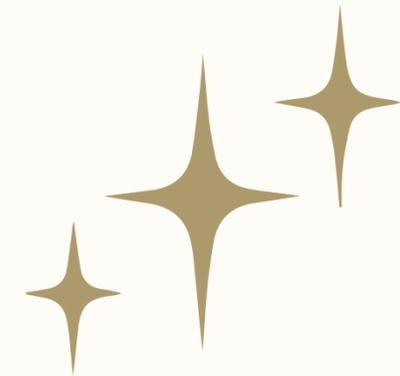
## 3. Menampilkan Data Mahasiswa yang Masuk pada Tahun Tertentu

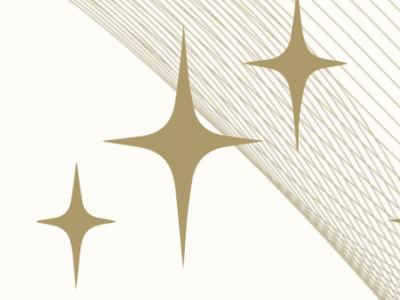
Untuk menampilkan mahasiswa yang masuk pada tahun 2020, kita bisa menggunakan klausa

**WHERE:**

```
SELECT *  
FROM mahasiswa  
WHERE tahun_masuk = 2020;
```

Perintah ini akan menampilkan seluruh informasi mahasiswa yang masuk pada tahun 2020.





## 4. Menampilkan Mahasiswa dari Jurusan Tertentu

Jika kita ingin menampilkan mahasiswa yang berasal dari jurusan "Teknik Informatika",

```
SELECT *  
FROM mahasiswa  
WHERE jurusan = 'Teknik Informatika';
```

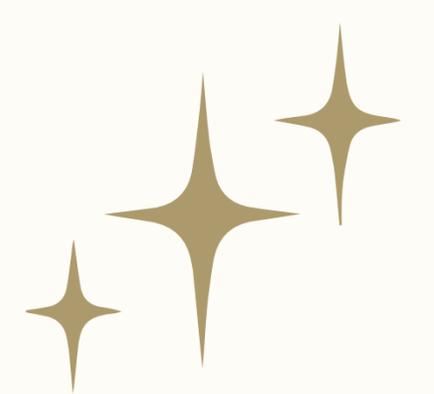
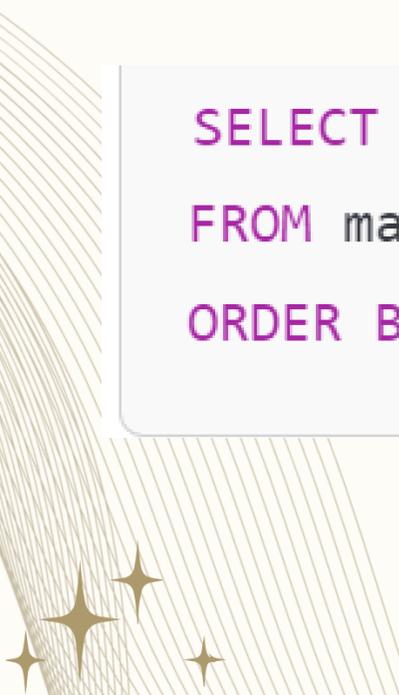
Hasilnya adalah daftar mahasiswa yang mengambil jurusan "Teknik Informatika".

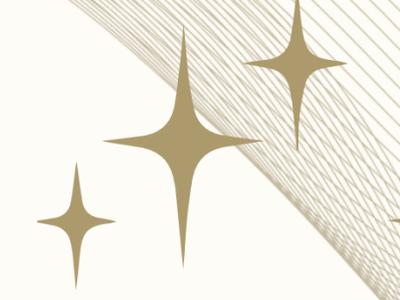
## 5. Mengurutkan Data Mahasiswa Berdasarkan Nama

Untuk menampilkan data mahasiswa dan mengurutkannya berdasarkan nama, kita dapat menambahkan klausa ORDER BY:

```
SELECT *  
FROM mahasiswa  
ORDER BY nama;
```

Ini akan menampilkan semua mahasiswa dengan urutan alfabetis berdasarkan nama mereka.





## 6. Menampilkan Mahasiswa yang Masuk Setelah Tahun 2019

Jika kita ingin menampilkan mahasiswa yang masuk setelah tahun 2019, kita dapat menggunakan operator >:

```
SELECT *  
FROM mahasiswa  
WHERE tahun_masuk > 2019;
```

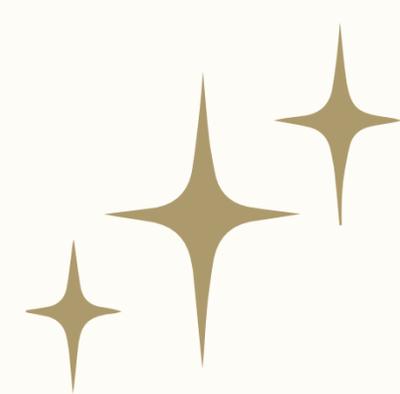
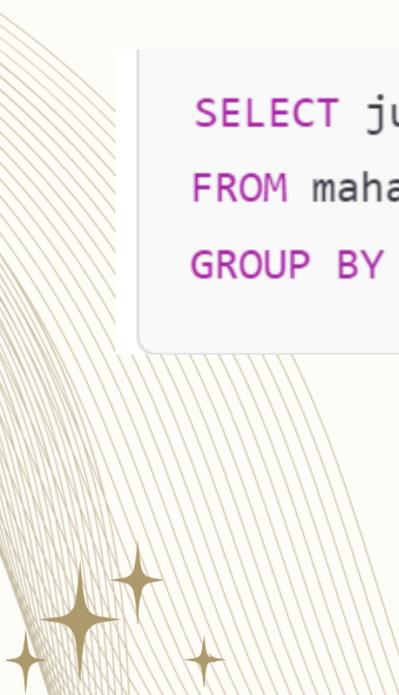
Perintah ini akan menampilkan mahasiswa yang masuk setelah tahun 2019.

## 7. Menghitung Jumlah Mahasiswa per Jurusan

Untuk mengetahui jumlah mahasiswa di setiap jurusan, kita dapat menggunakan klausa GROUP BY dengan fungsi agregat COUNT:

```
SELECT jurusan, COUNT(*) AS jumlah_mahasiswa  
FROM mahasiswa  
GROUP BY jurusan;
```

Perintah ini akan menampilkan jumlah mahasiswa di setiap jurusan.



The background features a series of thin, wavy lines in a light brown color that sweep across the top and bottom of the frame. Scattered throughout are several dark brown, four-pointed starburst shapes of varying sizes. The central focus is a dark brown, rounded rectangular banner with a decorative notch on each side, containing the text "TERIMA KASIH" in white, uppercase, sans-serif font.

TERIMA KASIH