

# *Sistem Basis Data 1*

DR. HJ. SILVIA RATNA, S.KOM., M.KOM  
NADIYA HIJRIANA, S.T., M.KOM  
ERFAN KARYADIPUTRA, S.KOM., M.KOM  
TRI ANA QUR'ANA, S.KOM., M.KOM

A large, irregular pink brushstroke shape serves as a background for the text. It has a soft, painterly texture with visible brush marks and a slightly darker pink shadow on the right side.

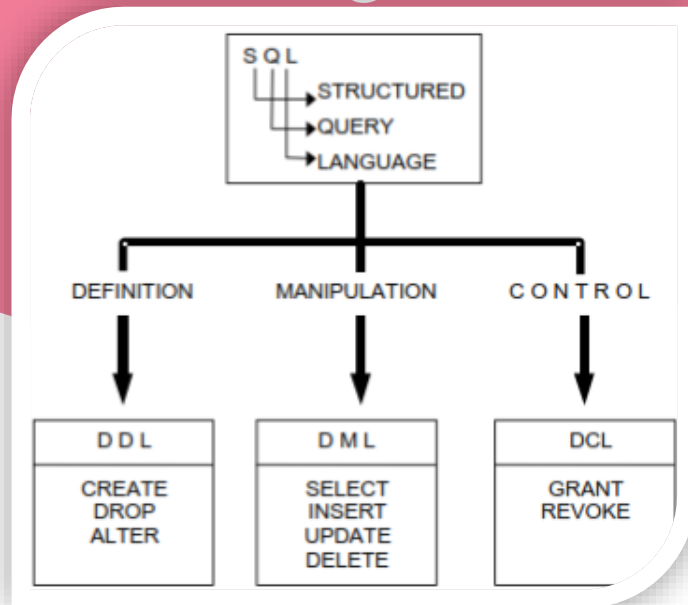
# *Pengenalan SQL* *(Structured Query language)*

A large, irregular pink brushstroke shape on the left side of the slide, containing the text 'SQL'.

*SQL*

SQL merupakan bahasa pemrograman database yang biasanya digunakan untuk memanipulasi, mengakses, dan memodifikasi database relasional.

# *Komponen SQL*



- **Data Definition Language (DDL) :**

Digunakan untuk mendefinisikan struktur database dengan menggunakan perintah : **create, drop, alter.**

- **Data Manipulation Language (DML) :**

Digunakan untuk memanipulasi data dengan menggunakan perintah : **select, insert, update, delete.**

- **Data Control Language (DCL) :**

Digunakan untuk mengontrol hak akses pengguna data dengan perintah : **grant, revoke**



## *Type Data MySQL*

Tipe data merupakan jenis nilai yang bisa ditampung pada suatu variable, bisa berupa angka (numerik), teks, ataupun berupa gambar.

Berikut ini adalah beberapa tipe data pada MySQL :

Tipe Data Angka (Numerik)

Tipe Data Teks (String)

Tipe Data Date

Tipe Data BLOB

# Type Data Numerik

	Nama	Fungsi	Jangkauan	Ukuran
1	TINYINT	Menyimpan data bilangan bulat positif dan negatif.	-128 s/d 127	1 byte (8 bit).
2	SMALLINT	Menyimpan data bilangan bulat positif dan negatif.	-32.768 s/d 32.767	2 byte (16 bit).
3	MEDIUMINT	Menyimpan data bilangan bulat positif dan negatif.	-8.388.608 s/d 8.388.607	Ukuran : 3 byte (24 bit).
4	INT	Menyimpan data bilangan bulat positif dan negatif.	-2.147.483.648 s/d 2.147.483.647	4 byte (32 bit).
5	BIGINT	Menyimpan data bilangan bulat positif dan negatif.	$\pm 9,22 \times 10^{18}$	8 byte (64 bit).
6	FLOAT	menyimpan data bilangan pecahan positif dan negatif presisi tunggal	-3.402823466E+38 s/d -1.175494351E-38, 0, dan 1.175494351E-38 s/d 3.402823466E+38.	4 byte (32 bit)
7	DOUBLE	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	5 byte (64 bit).
8	REAL	menyimpan data bilangan pecahan positif dan negatif presisi ganda.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	6 byte (64 bit).
9	DECIMAL	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	7 byte (64 bit).
	NUMERIC	menyimpan data bilangan pecahan positif dan negatif.	-1.79...E+308 s/d -2.22...E-308, 0, dan 2.22...E-308 s/d 1.79...E+308.	8 byte (64 bit).

# *Type Data Teks*

No	Nama	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), 0 s/d 65.535
3	TINYTEXT	menyimpan data text.	1 s/d 255 karakter (versi 4.1), 0 s/d 65.535
4	TEXT	menyimpan data text.	0 s/d 65.535
5	MEDIUMTEXT	menyimpan data text.	0 s/d 224 - 1 karakter
6	LONGTEXT	menyimpan data text.	1 s/d 224 - 1 karakter

# *Type Data Date & Blob*

No	Nama	Fungsi	Jangkauan	Ukuran
1	DATE	menyimpan data tanggal	1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)	3 byte.
2	TIME	menyimpan data waktu	-838:59:59 s/d +838:59:59 (HH:MM:SS)	3 byte.
3	DATETIME	menyimpan data tanggal dan waktu.	1000-01-01 00:00:00 s/d '9999-12-31 23:59:59'	8 byte
4	YEAR	menyimpan data tahun dari tanggal	1900 s/d 2155	1 byte

No	Nama	Fungsi	Jangkauan
1	BIT	Menyimpan data biner.	64 digit biner
2	TINYBLOB	menyimpan data biner/ Gambar ukuran kecil	255 byte
3	BLOB	Menyimpan data biner/ Gambar	4
4	MEDIUMBLOB	Menyimpan data biner/ Gambar kuran sedang	224-1 byte
5	LOB	Menyimpan data biner/ Gambar ukuran besar	232- 1 byte





# *DDL*

- **CREATE :**

CREATE DATABASE, CREATE FUNCTION,  
CREATE INDEX, CREATE PROCEDURE,  
CREATE TRIGGER, CREATE VIEW, CREATE  
TABLE

- **RENAME :**

RENAME TABLE

- **ALTER :**

ALTER DATABASE, ALTER FUNCTION,  
ALTER PROCEDURE, ALTER TABLE,  
ALTER VIEW

- **DROP :**

DROP DATABASE, DROP FUNCTION,  
DROP INDEX, DROP PROCEDURE, DROP  
TABLE, DROP TRIGGER, DROP VIEW



## *Syntax Terkait Database*

- **Membuat database**

Create database nmDB;

Ex : create database kampus;

Jika database yang akan dibuat sudah ada, maka akan muncul pesan error. Namun jika ingin otomatis menghapus database yang lama, maka tambahkan syntax if not exists.

Ex : create database [if not exist] kampus;

- **Menghapus database**

Drop database nmDB;

Ex : drop database kampus;

- **Memanggil database**

Use nmDB;

Ex : Use kampus;

- **Menampilkan semua database**

Ex : Show databases;



## *Syntax Terkait Tabel*

- **catatan**

Not null artinya data tidak boleh kosong

- **Membuat tabel**

Create tabel nmTBL

(col1 tipe data (lebar) not null primary key, col2 tipe data (lebar));

Ex : create table mahasiswa

(npm int(8) not null primary key, nama varchar(20));

- **Utk penomoran otomatis**

Create tabel nmTBL

Ex : create table mahasiswa

(npm int(8) not null primary key **auto\_increment**, nama varchar(20));

- **Menghapus tabel**

Drop Table nmTBL;

Ex : drop table nmTBL;

- **Menampilkan struktur tabel**

Desc nmTBL;

Ex : desc mahasiswa;

- **Merubah nama tabel**

Ex : alter nmTBL lama rename table nmTBL baru;

- **Menampilkan semua tabel**

Ex : show tables;



## *Syntax Terkait Tabel*

- **Menambah kolom/field baru**

Alter tabel nmTBL **add** col3 tipe data (lebar);

Ex : **alter table** mahasiswa **add** alamat varchar(25);

- **Menambah kolom/field baru dgn cara menyisipkannya diantara tabel yg ada**

Alter tabel nmTBL add col4 tipe data (lebar) after col2;

Ex : **alter table** mahasiswa **add** j\_kelamin char(1) **after** nama;

- **Merubah tipe data kolom/field**

Alter tabel nmTBL modify col4 tipe data baru (lebar);

Ex : **alter table** mahasiswa **modify** j\_kelamin enum('l','p');

- **Merubah keseluruhan kolom/field tertentu (merubah nama kolom, merubah posisi dll)**

Alter tabel nmTBL change col4 lama col4 baru tipe data (lebar) after col3;

Ex : **alter table** mahasiswa **change** j\_kelamin jenis\_kelamin enum('l','p') **after** alamat;

- **Menghapus kolom/field**

Alter tabel nmTBL drop col4;

Ex : **alter table** mahasiswa **drop** jenis\_kelamin;

- **Menambah foreign key**

Ex : **alter table** nilai **add foreign key** (npm) **references** mahasiswa (npm);

A large, irregular pink brushstroke shape that serves as a background for the text.

***D.MIL***

- **SELECT**
- **INSERT**
- **UPDATE**
- **DELETE**



## *Syntax Terkait Tabel*

- **Menambah data**

Insert into nmTBL values (data1,'data2','data3');

Ex : **insert into** mahasiswa **values** (1,'Erfan','jln hawai');

Jika tipe data yg digunakan integer maka saat input data tanpa tanda petik ('')

Jika sebelumnya menggunakan auto\_increment pada kolom pertama, maka syntax:

Ex : **insert into** mahasiswa **values** (null,'Erfan','jln hawai');

- **Merubah Data**

Update nmTBL set col2 = 'data baru' where col1 = 'data1';

Ex : **update mahasiswa set** nama = 'Erfan Karyadiputra' **where** npm = '1';

- **Menghapus data**

Delete from nmTBL where col1 = 'data1';

Ex : **delete from** mahasiswa **where** npm = '1';

Jika menghapus semua data, maka syntax :

Ex : **delete from** mahasiswa;

**atau**

Ex : **truncate** mahasiswa;



## *Syntax Terkait Tabel*

- **Menampilkan semua data**

Select \* from nmTBL;

Ex : **select \* from** mahasiswa;

- **Menampilkan data berdasarkan kriteria khusus menggunakan klausa where**

Ex : **select \* from** mahasiswa **where** jenis\_kelamin ='p';

- **Menampilkan data berdasarkan kriteria khusus menggunakan klausa where dan and**

Ex : **select \* from** mahasiswa **where** jenis\_kelamin ='p' **and** alamat ='jln hawai';

- **Menampilkan data dari kecil ke besar a-z menggunakan klausa order by dan asc**

Ex : **select \* from** mahasiswa **order by** npm **asc**;

- **Menampilkan data dari besar ke kecil z-a menggunakan klausa order by dan desc**

Ex : **select \* from** mahasiswa **order by** npm **desc**;

- **Menampilkan data sesuai kebutuhan menggunakan klausa limit**

Ex : **select \* from** mahasiswa **limit** 5;



# *DCL*

- SQL standar menggunakan klausa **select**, **insert**, **update**, dan **delete** untuk otoritas Pembacaan, Penambahan, Pengubahan dan Penghapusan
- **GRANT (memberikan otoritas)**
  - GRANT SELECT
  - GRANT UPDATE
  - GRANT INSERT
  - GRANT DELETE
- **REVOKE (membatalkan otoritas)**
  - REVOKE SELECT
  - REVOKE DELETE
  - REVOKE INSERT
  - REVOKE UPDATE





## *Pemberian Otoritas*

- **Grant** <daftar otoritas> **on** <nama objek basis data> **to** <daftar user>

Contoh:

**Grant** update (nim, nama\_kul) **on** nilai **to** aisyah\_ayana\_kaori, erfan;

keterangan :

DBA memberikan otoritas pengubahan data hanya atribut npm dan nama di tabel mahasiswa kepada aisyah ayana kaori dan erfan

**Grant select on** kampus **to** erfan;

DBA hanya memberikan otoritas untuk membaca data dari database kampus kepada user erfan



## *Pembatalan Otoritas*

- Revoke <daftar otoritas> **on** <nama objek basis data> **from** <daftar user>

contoh:

**Revoke** select, insert **on** all **from** anna;

keterangan :

DBA membatalkan otoritas pembacaan dan penambahan data ke semua tabel dari user anna

**Revoke insert on** mahasiswa **from** anna;

DBA membatalkan akses menambahkan data ke tabel mahasiswa dari anna

# *Evaluasi & Latihan*

- Andi ingin membuat sebuah database **rumah sakit**, dengan struktur tabel **dokter** sebagai berikut

ID	nama	alamat	telp	spesialis
1	Dr. Aisyah Ayana Kaori, Sp.OG	Jln hawai	085300000025	kandungan
2	Dr. Anna Noviyana, Sp.A	Jln Griya	085300000023	anak

- Tuliskanlah masing-masing syntax sqlnya !