



BASIS DATA

DASAR MYSQL

TIM AJAR BASIS DATA JTI-POLINEMA



OVERVIEW MYSQL

- ❖ MySQL termasuk ke dalam Relational Database Management System (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (General Public License)
- ❖ MySQL merupakan turunan dari konsep database SQL (Structured Query Language) untuk pemilihan /seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

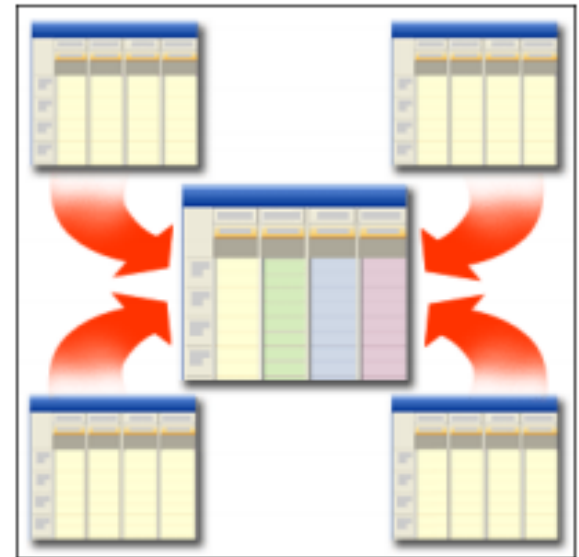
OVERVIEW MYSQL (CONT)

❑ MySQL merupakan Database Server yang bersifat :

- Open Source
- Multiplatform
- Berbasis database relasional

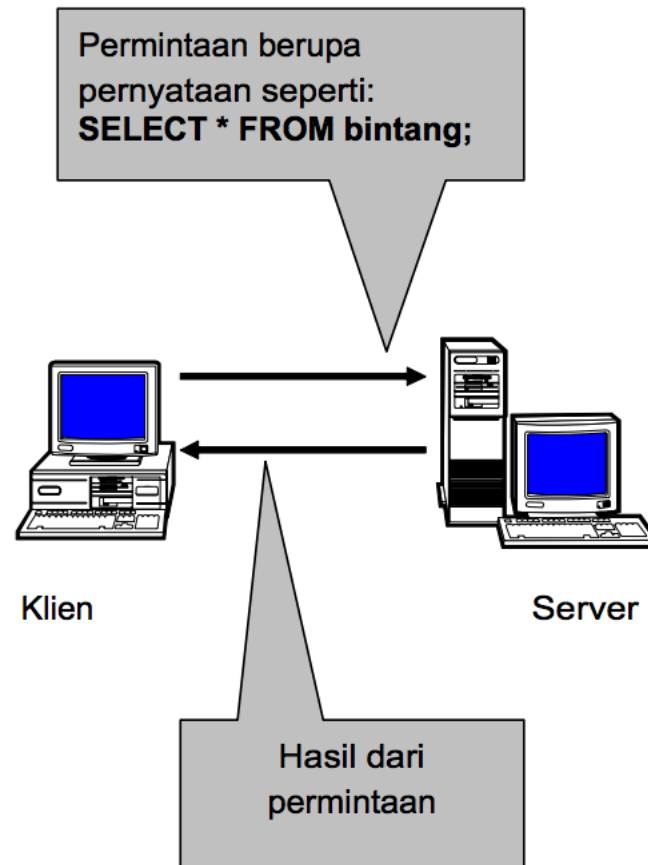
❑ Bisa dipakai untuk database pribadi atau pada level korporat berskala kecil hingga besar

❑ Selain bersifat free, ada juga yang bersifat komersial



OVERVIEW MYSQL (CONT)

- ❑ Menggunakan SQL untuk mendukung pengaksesan data (query)





KEISTIMEWAAN MYSQL

- **Portability** : berjalan stabil pada berbagai sistem operasi (Windows, Linux, Mac OS, Solaris dsb)
- **Open Source** : didistribusikan secara open source (gratis)
- **Multuser** : dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik
- **Performance Tuning** : memiliki kecepatan yang baik dalam menangani query sederhana.
- **Column Types** : memiliki tipe kolom yang kompleks, seperti : signed/unsigned integer, float, double, char, varchar, blob, time, datetime, timestamp, year, set serta enum
- **Command dan Functions** : memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query



KEISTIMEWAAN MYSQL (CONT)

- **Security** : memiliki lapisan sekuritas, seperti level subnetmask, nama host dan izin akses user disertai dengan password enkripsi.
- **Scalability dan Limits** : mampu menangani database dalam skala besar dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris.
- **Connectivity** : dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix socket atau Named Pipes.
- **Localisation** : dapat mendeteksi pesan kesalahan (error code) pada client dengan menggunakan lebih dari dua puluh bahasa.
- **Interface** : memiliki interface terhadap berbagai aplikasi dan bahasa pemrograman dengan fungsi API (Application Programming Interface).
- **Clients dan Tools** : dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database
- **Struktur Tabel** : memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE dibandingkan Oracle atau PostgreSQL



DASAR SQL

- SQL = Structured Query Language
- Digunakan untuk mengakses basis data relasional
- Bersifat standar >> bisa dipakai untuk basis data relasional lainnya
- Perintah SQL yang biasa digunakan dibagi menjadi DDL dan DML

5 BAGIAN UTAMA DARI BAHASA SQL



- DDL
bahasa yang digunakan untuk mendefinisikan data
contoh : create, drop
- DML
bahasa yang digunakan untuk memanipulasi data
contoh : insert, update
- Retrieving Data
perintah untuk menampilkan data dari database
contoh : select
- DCL
bahasa untuk kontrol pengendalian akses data ke database
contoh : grant, revoke
- DTL
bahasa untuk mengelola transaksi di database
contoh : commit transaction, rollback transaction



PERINTAH DDL

- DDL = Definition Data Language
- Digunakan untuk kepentingan penciptaan database, tabel, hingga penghapusan database atau tabel
- Contoh :
 - ☐ CREATE DATABASE
 - ☐ CREATE TABLE
 - ☐ DROP TABLE
 - ☐ ALTER TABLE



PERINTAH DML

- DML = Data Manipulation Language
- Digunakan untuk memanipulasi data
- Contoh :
 - ☐ SELECT mengambil data
 - ☐ DELETE menghapus data
 - ☐ INSERT menyisipkan data
 - ☐ UPDATE mengubah data

PERSIAPAN PEMAKAIAN MYSQL



- Lakukan instalasi mysql terlebih dulu
- Mysql memiliki sejumlah tool.
 - Program **mysql**, yang dipakai untuk mengakses database dari sisi klien
 - Program **mysqladmin**, untuk mengelola mysql dari sisi administrator sistem

MENGATUR PASSWORD ROOT



- Root adalah pemakai dengan wewenang paling tinggi dan digunakan untuk admin sistem
- Perintah untuk mengubah password root :
mysqladmin – uroot password **rahasia**

MASUK KE PROGRAM KLIEN MYSQL



- Connect :

mysql [-h host] [-u user] [-p [password]] [dbname]

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

```
Enter password: *****
```

```
Welcome to the MySQL monitor.  Commands end with ;  
or \g.
```

```
Your MySQL connection id is 14 to server version:  
3.23.34a
```

```
Type 'help;' or '\h' for help. Type '\c' to clear  
the buffer
```

```
mysql>
```

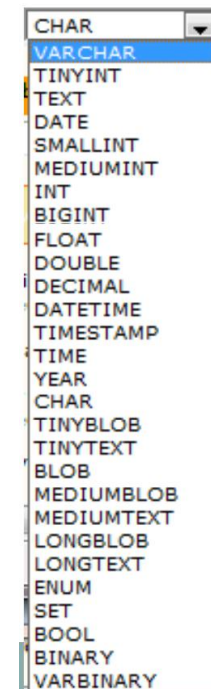
- Keluar dengan perintah quit atau exit
- Setting ada di config.inc.php (untuk merubah user dan password)



TIPE DATA DI MYSQL

- Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabel-tabel yg g p (an g secara lo gik meru pakan struktur dua dimensi terdiri dari baris (row atau record) dan kolom(column atau field). Sedangkan dalam sebuah database dapat terdiri dari beberapa table.
- Beberapa tipe data dalam MySQL yang sering dipakai:

Type data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LONGBLOB	Teks dengan panjang maksimum 4294967295 karakter



PEMBUATAN DATABASE DAN TABEL



Database : Pegawai

Tabel : Pribadi
Field :

- NIP
- Nama
- Tgl_lahir
- Sex
- Alamat
- kota

Tabel : pekerjaan
Field :

- NIP
- Tgl_Masuk
- Kode_bag
- Gaji

Tabel : Bagian
Field :

- Kode_bag
- Nama_bag

DATA DEFINITION LANGUAGE (DDL)



- ☐ CREATE/DROP DATABASE dbname;
- ☐ SHOW DATABASES;
- ☐ USE dbname;
- ☐ CREATE TABLE table_name (field_name type,..., constraints,...);
- ☐ SHOW TABLES;
- ☐ SHOW COLUMNS FROM table name;
- ☐ SHOW COLUMNS FROM table_name;
- ☐ DROP TABLE table_name;

MEMBUAT DATABASE DAN TABEL



- Cara untuk membuat sebuah database baru adalah dengan perintah:

`create database namadatabase;`

Contoh: `create database privatdb;`

- Untuk membuka sebuah database dapat menggunakan perintah berikut ini:

`use namadatabase;`

Contoh: `use privatdb;`

- Perintah untuk membuat tabel baru adalah:

`create table namatabel`

`(`

`struktur`

`);`

MEMBUAT DATABASE DAN TABEL



- Berikan perintah pada prompt mysql :
create database pegawai ;

```
mysql> create database pegawai;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> █
```



MENAMPILKAN DATABASE

- Perintah untuk menampilkan seluruh database yang ada di sistem :
show databases ;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| pegawai ← |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```



MEMILIH DATABASE

- Untuk memilih database mana yang akan digunakan, berikan perintah berikut ini :

USE namadatabase;

Contoh : USE **pegawai**;

Pegawai adalah nama database yang akan digunakan.

MENGHAPUS DATABASE



- Untuk menghapus database dapat menggunakan perintah :

`drop database nama_database;`

Contoh : bila ingin menghapus database pegawai karena sudah tidak digunakan, maka perintahnya sebagai berikut :

`drop database pegawai;`

MENAMPILKAN ISI DATABASE



- Isi dari database adalah berupa kumpulan tabel-tabel
- Untuk melihat tabel apa saja yang ada didalam database, maka berikan perintah berikut :
 show tables;



MEMBUAT TABEL BARU

- Untuk membuat tabel baru didalam sebuah database, maka berikan perintah :
 create tabel nama_tabel (nama_kolom
 tipe_data (panjang data) key)
- Menampilkan struktur table
 describe nama_tabel;
- Menghapus tabel
 drop nama_tabel;

CONTOH : MEMBUAT TABEL PRIBADI



```
CREATE TABLE tabel_pribadi (  
    nip char(5) NOT NULL,  
    nama varchar (35) NOT NULL,  
    tgl_lahir date,  
    sex enum('p','w'),  
    alamat varchar (35),  
    kota varchar (15),  
    primary key(nip)  
)
```




CONSTRAINTS

- Not Null
tidak boleh berisi NULL (kosong)
- UNIQUE
satu data dengan data lainnya tidak boleh sama
- PRIMARY KEY
- FOREIGN KEY
sebagai relasi antara 2 tabel
- AUTO_INCREMENT
nilai naik secara otomatis tanpa diisi



PENJELASAN TIPE DATA

- **Karakter**

- CHAR : Teks dengan maksimal 255 karakter
- VARCHAR : Teks maksimal 255 karakter dan bersifat variabel
- TEXT : Teks dengan panjang maksimal 65535

- **Bilangan**

- TINYINT : Bilangan 1 byte
- SMALLINT : Bilangan 2 byte
- INT atau INTEGER : Bilangan 4 byte
- BIGINT : Bilangan 8 byte
- FLOAT : Bilangan pecahan (4 byte)
- DOUBLE atau REAL : Bilangan pecahan (8 byte)
- DECIMAL(M,D) atau NUMERIC(M,D) : Bilangan pecahan



Tipe Data	Jangkauan SIGNED	Jangkauan UNSIGNED	Ukuran
TINYINT	-128 to 127	0 to 255	1 byte
SMALLINT	-32,768 to 32,767	0 to 65,535	2 bytes
MEDIUMINT	-8,388,608 to 8,388,607	0 to 16,777,215	3 bytes
INT	-2,147,483,648 to 2,147,483,647	0 to 4,294,967,295	4 bytes
BIGINT	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	0 to 18,446,744,073,709,551,615	8 bytes



PENAMBAHAN DATA

- Penambahan data dilakukan dengan menggunakan pernyataan INSERT
- Bentuk dasar :

```
INSERT INTO nama_tabel(nama_field1,  
nama_field2,...) values (nilai1, nilai2,...);
```

Contoh :

```
INSERT INTO pribadi(nip,nama,tgl_lahir,sex,alamat,kota)  
values ('001','yoyon','1965/10/10','P','jl. Kutisari  
67','surabaya');
```



MELIHAT ISI TABEL

- Untuk melihat isi dari tabel yang sudah diinputkan, dapat menggunakan perintah SELECT

`select * from nama_tabel`

contoh :

`select * from pribadi`

```
mysql> select * from pribadi;
+-----+-----+-----+-----+-----+-----+-----+
| nip | nama | tgl_lahir | sex | alamat | kota |
+-----+-----+-----+-----+-----+-----+-----+
| 001 | yoyon | 1965-10-10 | p | jl. kutisari 67 | surabaya |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> █
```



MELIHAT STRUKTUR TABEL

- Gunakan perintah :
DESC nama_tabel;

Contoh :

DESC pribadi;

```
mysql> desc pribadi;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip        | char(5)       | NO   | PRI | NULL    |       |
| nama       | varchar(35)   | NO   |     | NULL    |       |
| tgl_lahir  | date          | YES  |     | NULL    |       |
| sex        | enum('p','w') | YES  |     | NULL    |       |
| alamat     | varchar(35)   | YES  |     | NULL    |       |
| kota       | varchar(15)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```



MENGGANTI NAMA FIELD

- Perintah yang digunakan adalah ALTER TABLE

Contoh :

ALTER TABLE Pribadi

CHANGE sex kelamin ENUM('p','w');

```
mysql> desc pribadi;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip        | char(5)       | NO   | PRI | NULL    |       |
| nama       | varchar(35)   | NO   |     | NULL    |       |
| tgl_lahir  | date         | YES  |     | NULL    |       |
| kelamin    | enum('P','W') | YES  |     | NULL    |       |
| alamat     | varchar(35)   | YES  |     | NULL    |       |
| kota       | varchar(15)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

MENGGANTI UKURAN / TIPE FIELD



- Perintah yang digunakan adalah alter table
- Contoh :

ALTER TABLE pribadi

CHANGE kota kota VARCHAR(20);

```
mysql> desc pribadi;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip        | char(5)       | NO   | PRI | NULL    |       |
| nama       | varchar(35)   | NO   |     | NULL    |       |
| tgl_lahir  | date          | YES  |     | NULL    |       |
| kelamin    | enum('P','W') | YES  |     | NULL    |       |
| alamat     | varchar(35)   | YES  |     | NULL    |       |
| kota       | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```




MENAMBAHKAN DEFAULT

- Default pada struktur tabel digunakan untuk memberikan nilai bawaan pada suatu field kalau nilai bersangkutan tidak dimasukkan
- Contoh :

ALTER TABLE pribadi CHANGE kelamin kelamin
ENUM('P','W')DEFAULT 'P';

```
mysql> desc pribadi;
```

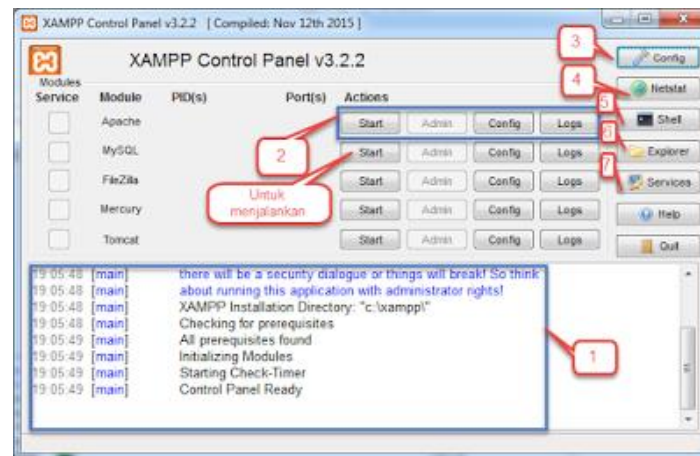
Field	Type	Null	Key	Default	Extra
nip	char(5)	NO	PRI	NULL	
nama	varchar(35)	NO		NULL	
tgl_lahir	date	YES		NULL	
kelamin	enum('P','W')	YES		P	
alamat	varchar(35)	YES		NULL	
kota	varchar(20)	YES		NULL	

6 rows in set (0.00 sec)

XAMPP CONTROL PANEL



- a. No 1 merupakan tempat Log dari semua aktifitas aplikasi yang sedang berjalan.
- b. No 2 merupakan Alat atau tool untuk mengontrol setiap module yang ada pada aplikasi XAMPP.
- c. No 3 yaitu digunakan untuk membuka panel konfigurasi.
- d. No 4 yaitu digunakan untuk menampilkan semua proses yang ada di dalam server.
- e. No 5 bisa Anda gunakan untuk membuka Unix Shell.
- f. No 6 digunakan untuk membuka windows explorer.
- g. No 7 berguna untuk menampilkan service yang sedang berjalan dibalik layar atau tidak tampak.



DATABASE AKADEMIK: DDL

```
CREATE DATABASE akademik;
```

```
USE akademik;
```

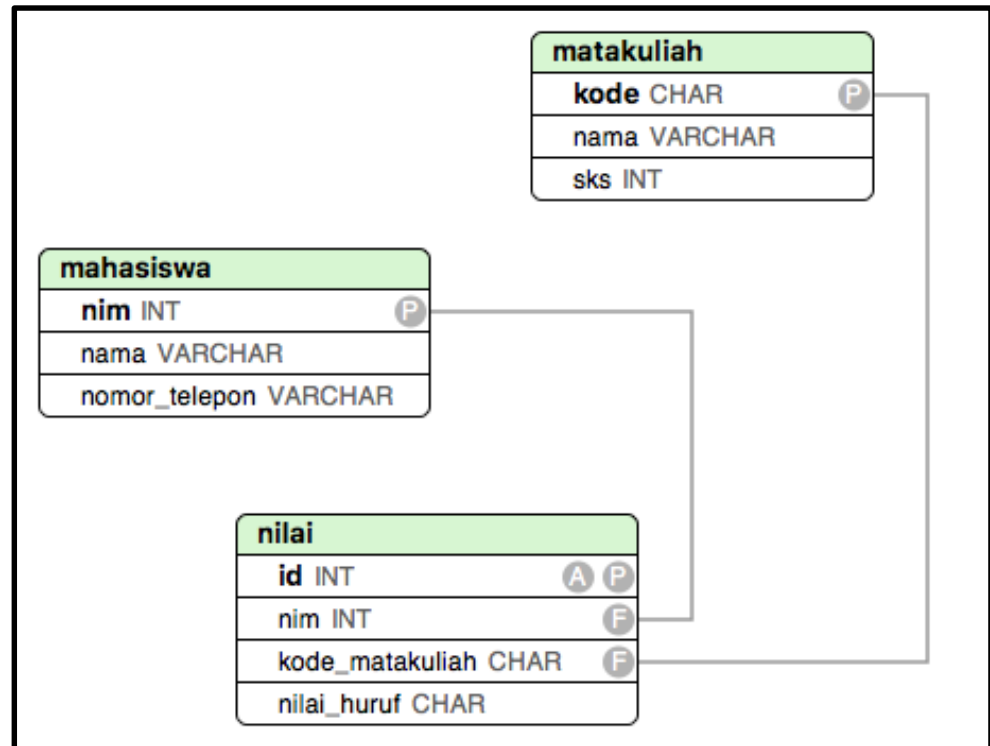
```
CREATE TABLE mahasiswa (
  nim INT(2),
  nama VARCHAR(255),
  nomor_telepon VARCHAR(20),
  PRIMARY KEY (nim)
);
```

```
CREATE TABLE matakuliah (
  kode CHAR(3),
  nama VARCHAR(255),
  sks INT(1),
  PRIMARY KEY (kode)
);
```

```
CREATE TABLE nilai (
  id INT AUTO_INCREMENT,
  nim INT(2),
  kode_matakuliah CHAR(3),
  nilai_huruf CHAR(2),
  PRIMARY KEY (id)
);
```

```
ALTER TABLE nilai
  ADD FOREIGN KEY nim_idxfk (nim) REFERENCES mahasiswa (nim);
```

```
ALTER TABLE nilai
  ADD FOREIGN KEY kode_matakuliah_idxfk (kode_matakuliah) REFERENCES matakuliah (kode);
```





INSERT

- Digunakan untuk **menambahkan** RECORD/Baris baru pada suatu tabel.
- Klausa pembentuk:
 - **INSERT**
 - **INTO**
 - **VALUES**
- Format:
 1. **INSERT INTO** nama_tabel (kolom1, kolom2, ...dst.) **VALUES** (nilai_kolom1, nilai_kolom2, ...dst.);
 2. **INSERT INTO** nama_tabel **VALUES** (nilai_kolom1, nilai_kolom2, ...dst.);
 3. [Salah satu dari kedua format sebelumnya],
(nilai_kolom_kolom_baris1), (nilai_kolom_kolom_baris2), ...dst.



INSERT

- Contoh Format #1:
 - **INSERT INTO** nama_tabel (kolom1, kolom2, ...dst.) **VALUES** (nilai_kolom1, nilai_kolom2, ...dst.);
- Digunakan jika kita ingin menambahkan data pada **sebagian** kolom saja.

SQL:

```
INSERT INTO mahasiswa (nim, nama) VALUES (1, 'Ani Rahmawati');
```

Akan menghasilkan:

+	-----	+	-----	+	-----	+
	nim		nama		nomor_telepon	
+	-----	+	-----	+	-----	+
	1		Ani Rahmawati		NULL	
+	-----	+	-----	+	-----	+



INSERT

- Contoh Format **#2**:
 - **INSERT INTO** nama_tabel **VALUES** (nilai_kolom1, nilai_kolom2, ...dst.);
- Digunakan jika kita ingin menambahkan baris baru dengan data pada **semua** kolom.

SQL:

```
INSERT INTO mahasiswa VALUES (2, 'Budi Raharjo', '0858776453');
```

Akan menghasilkan:

nim	nama	nomor_telepon
1	Ani Rahmawati	NULL
2	Budi Raharjo	0858776453

INSERT

- Contoh Format **#3**:
 - [Salah satu dari kedua format sebelumnya], (nilai_kolom_kolom_baris1), (nilai_kolom_kolom_baris2), ...dst.
- Digunakan jika kita ingin menambahkan **beberapa baris** baru **sekaligus** dalam 1 SQL.

SQL:

```
INSERT INTO mahasiswa VALUES  
  (3, 'Charlie Setiabudi', '0859767553'),  
  (4, 'Diandra Paramita', '0858998745');
```

Akan menghasilkan:

nim	nama	nomor_telepon
1	Ani Rahmawati	NULL
2	Budi Raharjo	0858776453
3	Charlie Setiabudi	0859767553
4	Diandra Paramita	0858998745

INSERT

- [Salah satu dari kedua format sebelumnya], (nilai_semua_kolom_baris1), (nilai_semua_kolom_baris2), ...dst.

SQL:

```
INSERT INTO matakuliah (kode, nama) VALUES
('BDD', 'Basis Data Dasar'),
('PBO', 'Pemrograman Berorientasi Objek'),
('MMT', 'Multimedia Terapan'),
('SPK', 'Sistem Pendukung Keputusan'),
('KCB', 'Kecerdasan Buatan'),
('ASD', 'Algoritma dan Struktur Data');
```

Akan menghasilkan:

kode	nama	sks
ASD	Algoritma dan Struktur Data	NULL
BDD	Basis Data Dasar	NULL
KCB	Kecerdasan Buatan	NULL
MMT	Multimedia Terapan	NULL
PBO	Pemrograman Berorientasi Objek	NULL
SPK	Sistem Pendukung Keputusan	NULL



LATIHAN SOAL

Database : Persewaan

KASET

KODE KASET	PRODUKSI	JENIS MUSIK	HARGA	JUDUL ALBUM	PENYANYI
DMS	DALAM NEGERI	POP	Rp 15.000,-	PERUBAHAN	D'MASIV
KGN	DALAM NEGERI	POP	Rp 15.000,-	BINTANG 14 HARI	KANGEN BAND
NIR	LUAR NEGERI	ROCK	Rp 20.000,-	NEVERMIND	NIRVANA

PEMBELI

KODE_CUSTOMER	NAMA_PEMBELI	STATUS
CS001	DWIKY	MEMBER
CS002	ARI	NON MEMBER
CS003	ZONA	NON MEMBER

TRANSAKSI

NO_FAKTUR	KODE_KASET	KODE_CUSTOMER	TANGGAL TRANSAKSI	JUMLAH BELI
F001	DMS	CS001	19-01-09	3
F002	KGN	CS002	20-01-09	4
F003	NIR	CS003	21-01-09	5

Tulislah perintah apa saja yang digunakan dalam pembuatan basis data persewaan !



TERIMAKASIH



REFERENSI

- Dwi Puspitasari, S.Kom, “**Buku Ajar Dasar Basis Data**”, *Program Studi Manajemen Informatika Politeknik Negeri Malang*, 2012.
- Fathansyah, “**Basisdata Revisi Kedua**”, Bandung: Informatika, 2015.