

Pemrograman Basis Data Berbasis Web

Pertemuan Ke 9
(Basis Data MySQL)



MySQL?



- MySQL termasuk di dalam jenis RDBMS (*Relational Database Management System*).
- Suatu basis data (*database*) memiliki satu atau beberapa tabel. Tabel terdiri dari sejumlah baris dan kolom.
- Pada umumnya, di dalam konteks bahasa SQL, informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi yang terdiri atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column* atau *field*).



- Tipe data Numerik

TINYINT, SMALLINT, MEDIUMINT, INT, INTEGER, BIGINT, FLOAT(x), FLOAT, DOUBLE, DOUBLE PRECISION, REAL, DECIMAL(M,D), NUMERIC(M,D)

- Tipe data *String*

CHAR, VARCHAR, TINYBLOB, TINYTEXT, BLOB, TEXT, MEDIUMBLOB, MEDIUMTEXT, LONGBLOB, LONGTEXT, ENUM('elemen1', 'elemen2', ...), SET('elemen1', 'elemen2', ...)

- Tipe data char() dan varchar()

- Tipe data Tanggal

DATETIME, DATE, TIMESTAMP, TIME, YEAR



- Operator matematika
+ - * / %
- Operator range
< > <= >=
- Operator kesamaan
= <>
- Operator logika
&& ||

Pernyataan (statement) SQL dapat digolongkan atas Dua golongan yaitu:



- ***Data Definition Language (DDL)*** yang mendefinisikan struktur suatu data. Perintah-perintah SQL yang termasuk DDL antara lain:
 - CREATE – untuk membuat tabel.
 - DROP – untuk menghapus suatu tabel.
- ***Data Manipulation Language (DML)*** yang dapat mencari (query) dan mengubah (modify) suatu tabel. Perintah-perintah SQL yang termasuk DML antara lain:
 - SELECT – untuk membaca (query) isi tabel.
Syntaks: **SELECT [ALL | DISTINCT [*] FROM [Nama Table] WHERE [Kondisi];**
 - INSERT – untuk memasukkan data ke table.
Syntaks: **INSERT INTO [Nama Tabel]([Nama Kolom]) VALUE [. Ekspresi];**
 - UPDATE – untuk mengubah isi tabel.
Syntaks:
UPDATE [Nama Tabel]SET [Nama Kolom= Ekspresi] WHERE [Kondisi];
 - DELETE – untuk menghapus isi tabel.
Syntaks:
DELETE FROM [Pemakai.]Nama Table[@Nama Database][WHERE Kondisi];

Perintah *CREATE*



Sintaks:

```
create DATABASE nama;
```

atau

```
create TABLE nama;
```

Contoh:

- *create database*:
create DATABASE db_akademik;
- *create tabel*:
create TABLE mahasiswa
(
nim SMALLINT(6) NOT NULL **PRIMARY KEY**,
Nama CHAR(20),
Tanggal_lahir DATE,
alamat VARCHAR(3),
nomortlp VARCHAR(12)
);

Perintah *INSERT*



Sintaks:

```
insert INTO NAMA_TABEL (kolom1, kolom2, kolom3,...)  
values ('nilai1',' nilai2',' nilai3',...);
```

Contoh :

```
insert INTO mahasiswa (nim,nama,tanggal_lahir, alamat,nomortlp)  
values (1111,"Setia Budi","1985-06-25","Jl.Kutai 33, Sby","031-3459097");
```

```
insert INTO mahasiswa values  
(1112,"Dewi Sekartaji","1985-10-01","Jl. Mangga 2, Sby","031-6749331"),  
(1113,"Adinda Cendrakasih","1986-01-10","Jl. Ketintang 45C, Sby","031-8291190");
```

Perintah *SELECT*



Sintaks:

```
select kolom1, kolom2, kolom3, [...] FROM  
NAMA_TABEL;
```

Contoh:

```
select * FROM mahasiswa;
```

```
select nim, Nama FROM mahasiswa;
```


Perintah *SELECT* ... [lanjutan]



- **Select banyak tabel**

→ mengambil data dari banyak tabel sekaligus

Sintaks:

```
select TABEL1.tabel_kolom1, TABEL1.tabel_kolom2,  
TABEL2.tabel_kolom1, TABEL2.tabel_kolom2  
FROM TABEL1, TABEL2;
```

(tanda titik diperlukan sebagai penghubung antara nama tabel dan nama kolom)

- **Order by**

→ mengurutkan (*ascending* atau *descending*) *record* hasil *query* berdasarkan suatu kolom.

Sintaks:

```
select kolom1, kolom2, [...] FROM NAMA_TABEL  
ORDER BY nama_kolom;
```

- **Where**

→ menentukan kondisi atau persyaratan dari *query*

Sintaks:

```
select kolom1, kolom2, [...] FROM NAMA_TABEL WHERE persyaratan;
```

Suatu pernyataan kondisional mempunyai bagian sisi kiri yang merupakan kolom tabel dan bagian sisi kanan kondisi yang dijadikan syarat pencarian.

Contoh persyaratan:

```
WHERE nama_kolom = kondisi;
```

Mengubah Struktur Tabel



Sintaks:

```
alter TABLE NAMA_TABEL ADD/DROP NAMA_KOLOM;  
alter TABLE NAMA_TABEL CHANGE KOLOM_PERINTAH;  
alter TABLE NAMA_TABEL RENAME NAMA_TABEL_BARU;
```

Contoh:

- Menambahkan sebuah kolom baru (Hobi) ke dalam tabel mahasiswa:
alter TABLE mahasiswa
ADD Hobi;
- Menghapus kolom:
alter TABLE mahasiswa
DROP Hobi;
- Mengganti nama tabel:
alter TABLE mahasiswa RENAME mahasiswaatrun;



Sintaks:

```
update NAMA_TABEL  
SET kolom1="nilai", kolom2="nilai",  
kolom3="nilai"  
WHERE kolom_kunci = "nilai";
```

Contoh:

- Mengubah nilai kolom Nama untuk mahasiswa dengan nim = 1111.
UPDATE mahasiswa
SET Nama = "Setiawan Budi"
WHERE nim = '1111';

Menghapus (*DELETE*) Data



Sintaks:

```
delete FROM NAMA_TABEL  
WHERE persyaratan;
```

Contoh:

- Menghapus *record* pada tabel mahasiswa yang field nim-nya bernilai 1114 :
delete FROM mahasiswa
WHERE nim=1114;



TRUNCATE

→ untuk membersihkan tabel secara keseluruhan (sama seperti *DELETE* tanpa kondisi), yaitu dengan cepat menghapus semua *record* tanpa menghapus struktur tabel.

Sintaks:

```
truncate TABEL NAMA_TABEL;
```

Contoh:

- Menghapus semua *record* dalam tabel mahasiswa:
truncate TABLE mahasiswa;