

Pertemuan 10

Bahasa Query Terapan

Structured Query Language (SQL)

SQL merupakan bahasa query terapan yang banyak digunakan oleh berbagai DBMS, diterapkan dalam berbagai *development tools* dan program aplikasi untuk berinteraksi dengan basis data.

Subdivisi SQL:

1. *Data Definition Language (DDL)*

Query-query ini digunakan untuk mendefinisikan struktur atau skema basis data.

2. *Data Manipulation Language (DML)*

Query-query ini digunakan untuk manajemen data dalam basis data.

SQL lanjutan

PENGELOMPOKAN STATEMEN SQL

1. *Data Definition Language* (DDL)

CREATE DATABASE DROP DATABASE

CREATE TABEL DROP TABEL

CREATE INDEX DROP INDEX

CREATE VIEW DROP VIEW

ALTER TABLE

2. *Data Manipulation Language* (DML)

INSERT, SELECT, UPDATE, DELETE

SQL lanjutan

3. Data Access

GRANT , REVOKE

4. Data Integrity

RECOVER TABLE

5. Auxiliary

SELECT INTO OUTFILE,
LOAD, RENAME TABLE

Data Definition Language (DDL)

A. CREATE

1. Pembuatan Database

Nama Database adalah yang dapat mewakili suatu kejadian dapat berupa nama organisasi atau perusahaan.

Sintaks : `CREATE DATABASE nama_database`

Contoh : Buat database dengan nama KAMPUS

```
CREATE DATABASE KAMPUS
```

2. Pembuatan Tabel

Sintaks : `CREATE TABLE nama_table`

```
( nama_kolom1 tipe_data_kolom1,  
  nama_kolom2,tipe_data_kolom2,...)
```

Contoh :

Buat struktur tabel dengan nama tabel Mahasiswa dengan data NIM char(8), NAMA char(25), ALAMAT char(30)

```
CREATE TABLE Mahasiswa (NIM char(8) not null,  
  NAMA char(25) notnull, ALAMAT char(30) notnull)
```

DDL lanjutan

3. Pembuatan Index

Sintaks : `CREATE [UNIQUE] INDEX nama_index
ON nama_table (nama_kolom) ;`

Contoh :

Buat index data Mahasiswa berdasarkan NIM dengan nama MHSIDX
Dimana NIM tidak boleh sama

```
CREATE UNIQUE INDEX MHSIDX ON Mahasiswa(NIM)
```

4. Pembuatan View

Sintaks :

```
CREATE VIEW nama_view [ (nama_kolom1,...) ]  
AS SELECT statement  
[WITH CHECK OPTION] ;
```

Contoh :

Buat view dengan nama MHSVIEW yang berisi semua data mahasiswa

```
CREATE VIEW MHSVIEW  
AS SELECT * FROM Mahasiswa
```

DDL lanjutan

B. DROP (MENGHAPUS)

1. Menghapus Database

Sintaks : DROP DATABASE nama_db ;

2. Menghapus Tabel

Sintaks : DROP TABLE nama_table ;

3. Menghapus Index

Sintaks : DROP INDEX nama_index ;

4. Menghapus View

Sintaks : DROP VIEW nama_view ;

Contoh :

DROP DATABASE **KAMPUS**;

DROP TABLE MHS;

DROP INDEX MHSIDX;

DROP VIEW MHSVIEW;

DDL lanjutan

C. ALTER TABLE (MERUBAH STRUKTUR TABEL)

Sintaks: ALTER TABLE nama_tabel

ADD nama_kolom jenis_kolom

[FIRST | AFTER nama_kolom]

CHANGE [COLUMN] oldnama newnama

MODIFY nama_kolom jenis kolom, ...

DROP nama_kolom

RENAME newnama_tabel

Contoh :

1. Tambahkan kolom JKEL dengan panjang 1 char pada tabel Mahasiswa

```
ALTER TABLE Mahasiswa ADD JKEL char(1);
```

2. Ubah panjang kolom JKEL menjadi 15 char

```
ALTER TABLE Mahasiswa MODIFY COLUMN JKEL  
char(15);
```

3. Hapus kolom JKEL dari data table MHS

```
ALTER TABLE Mahasiswa DROP JKEL;
```


Data Manipulation Language (DML)

A. INSERT

Sintaks SQL yang digunakan untuk penambahan record baru kedalam sebuah tabel.

Sintaks: `INSERT INTO Nama_tabel [(nama_kolom1,...)]
values (nilai atribut1, ...)`

Contoh:Masukan data Mahasiswa dengan Nim 10296832, Nama Nurhayati beralamat di Jakarta

```
INSERT INTO Mahasiswa (Nim, Nama, Alamat) values  
("10296832","Nurhayati","Jakarta");
```

DML lanjutan

B. UPDATE

Sintaks SQL yang digunakan untuk mengubah nilai atribut pada suatu record dari sebuah tabel.

Sintaks : UPDATE nama_tabel
 SET nama_kolom = value_1
 WHERE kondisi ;

Contoh:

Ubah alamat menjadi “Depok” untuk mahasiswa yang memiliki NIM “10296832”

```
UPDATE Mahasiswa  
SET ALAMAT="Depok"  
WHERE NIM=" 10296832";
```

DML lanjutan

C. DELETE

Sintaks SQL yang digunakan untuk menghapus record dari sebuah tabel.

Sintaks: DELETE FROM nama_table
WHERE kondisi

Contoh:

Hapus data Mahasiswa yang mempunyai NIM
"21198002"

```
DELETE FROM Mahasiswa  
WHERE NIM=" 21198002"
```

DML lanjutan

Tabel dibawah ini untuk mengerjakan perintah **SELECT**

Tabel Nilai

NIM	KD_MK	MID	FINAL
10296832	KK021	60	75
10296126	KD132	70	90
31296500	KK021	55	40
41296525	KU122	90	80
21196353	KU122	75	75
50095487	KD132	80	0

Tabel Mahasiswa

NIM	NAMA	ALAMAT
10296832	Nurhayati	Jakarta
10296126	Astuti	Jakarta
31296500	Budi	Depok
41296525	Prananigrum	Bogor
50096487	Pipit	Bekasi
21196353	Quraish	Bogor
10296001	Fintri	Depok
21198002	Julizar	Jakarta

Tabel MataKuliah

KD_MK	NAMA_MK	SKS
KK021	Sistem Basis Data	2
KD132	Sistem Informasi Manajemen	3
KU122	Pancasila	2

DML lanjutan

D. SELECT

Sintaks : `SELECT [DISTINCT | ALL] nama_kolom`
`FROM nama_tabel`
`[WHERE condition]`
`[GROUP BY column_list]`
`[HAVING condition]`
`[ORDER BY column_list [ASC | DESC]]`

DML lanjutan

Contoh :

- a. Tampilkan semua data Mahasiswa

```
SELECT NIM,NAMA,ALAMAT FROM Mahasiswa;
```

Atau

```
SELECT * FROM Mahasiswa;
```

Maka hasilnya adalah :

NIM	NAMA	ALAMAT
10296832	Nurhayati	Jakarta
10296126	Astuti	Jakarta
31296500	Budi	Depok
41296525	Prananingrum	Bogor

DML lanjutan

b. Tampilkan Mata Kuliah yang SKS nya 2

```
SELECT NAMA_MK FROM MataKuliah WHERE SKS=2
```

Maka Hasilnya:

NAMA_MK
Sistem Basis Data Pancasila

DML lanjutan

c. Tampilkan semua data nilai dimana nilai MID lebih besar sama dengan 60 atau nilai finalnya lebih besar 75.

maka penulisannya :

```
SELECT * FROM Nilai WHERE MID >= 60 OR FINAL > 75
```

Hasilnya:

NIM	KD_MK	MID	FINAL
10296832	KK021	60	75
10296126	KD132	70	90
41296525	KU122	90	80
21196353	KU122	75	75

Aplikasi yang digunakan sebagai contoh adalah Mysql

Dari Address ketik : <http://localhost/phpmyadmin>

Tampilan user ketik **root** dan **password** dikosongkan

Tugas

- Gunakan perintah-perintah Bahasa Query terapan yang dipelajari pada pertemuan ke-10 pada project yang dipilih pada pertemuan sebelumnya