

Basis Data Terapan

Yoannita, S.Kom



Membuat Tabel

```
CREATE TABLE MAHASISWA (  
    NPM CHAR(10),  
    NAMA VARCHAR (25),  
    ALAMAT VARCHAR (55),  
    NILAI INTEGER,  
    TGLLAHIR DATETIME  
)
```


Menghapus Tabel

- *DROP TABLE nama_tabel*
- Contoh :

DROP TABLE MAHASISWA

- Tidak dapat digunakan untuk menghapus tabel yang terhubung(referenced) dengan tabel yang lain melalui FOREIGN KEY. Konstrain FOREIGN KEY tersebut harus dihapus terlebih dahulu.
- DROP TABLE tidak dapat digunakan untuk menghapus *system tables*.
- Jika Anda menghapus semua baris dalam tabel (*DELETE tablename*) atau menggunakan syntax TRUNCATE TABLE, tabel tersebut akan tetap ada sampai tabel tersebut di-DROP.

Mendefinisikan Null/Not Null

- Secara default, suatu atribut boleh bernilai kosong(null)/tidak diisi ketika pengguna menambah data.
- Namun dibutuhkan pula suatu atribut yang harus diisi (not null).

Mendefinisikan Null/Not Null

```
CREATE TABLE MAHASISWA (  
    NPM CHAR(10) not null,  
    NAMA VARCHAR (25) not null,  
    ALAMAT VARCHAR (55),  
    NILAI INTEGER ,  
    TGLLAHIR DATETIME  
)
```

Mendefinisikan Nilai Default

- Nilai default : nilai yang otomatis diberikan oleh sistem untuk suatu field ketika ada penambahan baris/record baru, sementara nilai pada field tersebut tidak diisi oleh user.
- Apabila tidak didefinisikan, nilai default dari suatu atribut adalah NULL.

Mendefinisikan Nilai Default

```
CREATE TABLE MAHASISWA (  
    NPM CHAR(10),  
    NAMA VARCHAR (25),  
    ALAMAT VARCHAR (55)  
        default 'palembang',  
    NILAI INTEGER,  
    TGLLAHIR DATETIME  
)
```

Mendefinisikan Primary Key

Ada 3 cara untuk mendefinisikan primary key :

A. Didefinisikan di belakang atribut yang terpilih sebagai primary key

- Cara ini dilakukan bersama-sama pada saat pembuatan tabel.
- Apabila atribut yang terpilih sebagai primary key lebih dari satu (komposit key) cara ini tidak dapat mengimplementasikannya.

Mendefinisikan Primary Key(A)

```
CREATE TABLE MAHASISWA (  
    NPM CHAR(10) not null primary key,  
    NAMA VARCHAR (25) not null,  
    ALAMAT VARCHAR (55),  
    NILAI INTEGER ,  
    TGLLAHIR DATETIME  
)
```

Mendefinisikan Primary Key(B)

- B. Mendefinisikan konstrain primary key pada bagian akhir pendefinisian tabel
 - Dilakukan pada saat pembuatan tabel
 - Pada cara ini, primary key boleh merupakan atribut tunggal atau kumpulan atribut.

Mendefinisikan Primary Key(B)

Contoh 1 :

```
CREATE TABLE MAHASISWA (  
    NPM CHAR(10) not null,  
    NAMA VARCHAR (25) not null,  
    ALAMAT VARCHAR (55),  
    NILAI INTEGER ,  
    TGLLAHIR DATETIME ,  
    PRIMARY KEY (NPM)  
)
```

Mendefinisikan Primary Key(B)

Contoh 2 :

```
CREATE TABLE KHS (  
    NPM CHAR(10) not null,  
    MATKUL VARCHAR(25),  
    NILAIHURUF CHAR(1) default 'B' ,  
    PRIMARY KEY (NPM, MATKUL)  
)
```


Mendefinisikan Primary Key(C)

C. Menambah Query khusus

pendefinisian konstrain primary key.

- Dilakukan pada saat tabel sudah terbuat, tetapi primary key belum didefinisikan.
- Primary key yang berupa atribut tunggal atau komposit key dapat didefinisikan dengan cara ini.

Mendefinisikan Primary Key(C)

```
CREATE TABLE MAHASISWA (  
    NPM CHAR(10) not null,  
    NAMA VARCHAR (25) not null,  
    ALAMAT VARCHAR (55),  
    NILAI INTEGER ,  
    TGLLAHIR DATETIME  
)
```

- **ALTER TABLE MAHASISWA ADD CONSTRAINT PK_MHS PRIMARY KEY (NPM)**
- **ALTER TABLE MAHASISWA DROP CONSTRAINT PK_MHS**

Mendefinisikan Primary Key(C)

KETERANGAN :

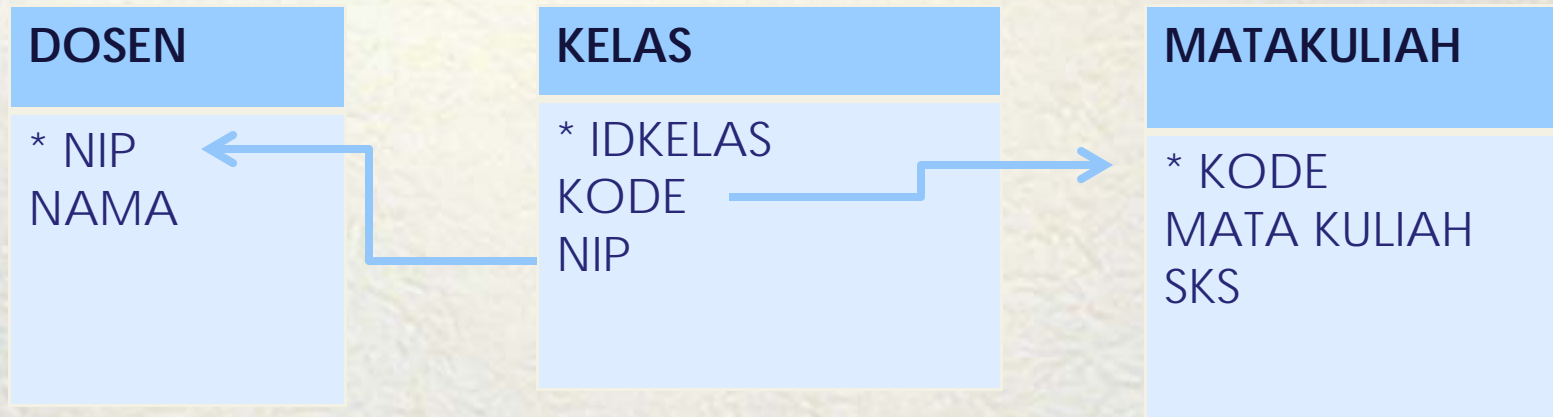
- **PK_MHS** adalah nama konstrainnya sedangkan **NPM** merupakan atribut yang terpilih sebagai primary key.
- Atribut yang dipilih sebagai primary key harus terdefinisi terlebih dahulu dalam pembuatan tabel dan memiliki konstrain not null pada tabel tersebut.

Mendefinisikan Foreign Key

- Untuk mendefinisikan foreign key harus dipastikan terlebih dahulu bahwa tabel dan atribut yang dirujuk (tabel induk dari foreign key) sudah didefinisikan.

Mendefinisikan Foreign Key

- CONTOH :



* = PRIMARY KEY

Mendefinisikan Foreign Key:

CARA 1: mendefinisikan pada saat pembuatan tabel

```
CREATE TABLE KELAS (  
    IDKELAS INTEGER NOT NULL PRIMARY KEY  
    KODE CHAR(2) NOT NULL  
        REFERENCES MATAKULIAH(KODE)  
            ON UPDATE CASCADE  
            ON DELETE NO ACTION  
    NIP INTEGER NOT NULL  
        REFERENCES DOSEN (NIP)  
            ON UPDATE CASCADE  
            ON DELETE NO ACTION  
)
```


Mendefinisikan Foreign Key:

CARA 2: mendefinisikan FK setelah tabel dibuat

```
ALTER TABLE KELAS ADD CONSTRAINT  
FK_KELAS1 FOREIGN KEY (KODE)  
REFERENCES MATAKULIAH (KODE)  
ON UPDATE CASCADE  
ON DELETE NO ACTION
```

```
ALTER TABLE KELAS ADD CONSTRAINT  
FK_KELAS2 FOREIGN KEY (NIP)  
REFERENCES DOSEN(NIP)  
ON UPDATE CASCADE  
ON DELETE NO ACTION
```

Mendefinisikan Foreign Key:

CARA 2: mendefinisikan FK setelah tabel dibuat

- Pendefinisian FK dengan cara kedua ini mempunyai keuntungan dalam memodifikasi/menghapus FK karena nama konstrainnya jelas

Contoh :

```
ALTER TABLE KELAS DROP CONSTRAINT FK_KELAS2
```


CASCADE dan NO ACTION

- Variabel ON UPDATE dan ON DELETE dapat berisi CASCADE atau NO ACTION.
- **CASCADE** : segala perubahan data pada tabel induk (dimana data tersebut sudah dipakai pada tabel yang ada FK-nya tersebut), maka data pada tabel yang ada FK akan menyesuaikan/mengikuti perubahan dari tabel induk

CASCADE dan NO ACTION

- **NO ACTION** : jika sebuah data pada tabel induk sudah dirujuk pada tabel yang ada FK-nya, maka tabel induknya tidak diijinkan untuk melakukan pengubahan/penghapusan.

CASCADE dan NO ACTION

- Jika didefinisikan **ON UPDATE CASCADE** maka ketika NIP di tabel Dosen diubah menjadi 5, maka semua NIP pada tabel Kelas yang bernilai 2 akan diubah menjadi 5.
- Jika didefinisikan **ON DELETE CASCADE** maka yang terjadi ketika NIP 1 di tabel Dosen dihapus, maka semua NIP yang bernilai 1 pada tabel Kelas juga akan terhapus.

CASCADE dan NO ACTION

- Jika didefinisikan **ON UPDATE NO ACTION** maka pengeditan tidak diijinkan
- Jika didefinisikan **ON DELETE NO ACTION** maka penghapusan tidak diijinkan.
- Jika terpaksa diinginkan pengubahan pada tabel induk, maka sebelum pengubahan, data terkait pada tabel yang ada FK-nya tersebut harus dihapus terlebih dahulu.

Menambah Kolom pada Tabel

ALTER TABLE <nama tabel>

ADD <namaFieldBaru> <tipeData>

- Contoh

- ALTER TABLE MHSW ADD JDWL CHAR(1)
- ALTER TABLE

Menghapus Kolom pada Tabel

ALTER TABLE <namaTabel>

DROP COLUMN <namaField>

- Contoh :
 - ALTER TABLE MAHASISWA DROP COLUMN JDWL

Mengubah Tipe Kolom

```
ALTER TABLE <namaTabel> ALTER COLUMN  
<namaField> <tipeDatabaru>
```

- Contoh :
 - ALTER TABLE MHSW ALTER COLUMN ALAMAT VARCHAR(255)
- Terkadang pengubahan tipe kolom tidak dapat dilakukan pada kolom yang telah berisi data yang tidak kompatibel. Contohnya kolom alamat tidak bisa diubah menjadi tipe integer

Mengubah Nama Kolom

```
EXEC sp_rename '<namaTabel>.[<namaKolom>]',  
'<namaKolomYangBaru>', 'COLUMN'
```

- Stored Procedure digunakan untuk mengubah objek2, seperti me-rename nama kolom pada tabel
- Contoh:
mengubah nama kolom 'no_tlp' pada tabel pegawai menjadi 'telepon'

```
EXEC sp_rename 'pegawai.[no_tlp]', 'telepon', 'COLUMN'
```


Cara Penulisan Komentar

- SQL mempunyai 2 cara dalam menulis comments/komentar :

-- Line Comment

/* Block Comment */

Melihat Informasi Tabel

- **sp_help <nama tabel>**

Melihat Informasi Seluruh Database

- **sp_helpdb**

Melihat Informasi Detail Database

- **sp_helpdb <namaDatabase>**