



Basis Data Lanjut

Semester Genap 2018/2019

# Data Definition Language pada Oracle

Salhazan Nasution, S.Kom, MIT



# Data Definition Language (DDL)



# Data Definition Language (DDL)

⇒ DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur basis data

**Contoh:** Tabel, Index, View, Sequence, dll



# Tipe Data

Tipe Data	Keterangan
VARCHAR2	<ul style="list-style-type: none"><li>▪ Untuk karakter dengan panjang yang dinamis/fleksibel selama tidak melebihi panjang maksimal yang telah ditentukan.</li><li>▪ Panjang karakter maksimum 4000 karakter.</li></ul>
CHAR	<ul style="list-style-type: none"><li>▪ Untuk string atau karakter dengan panjang yang sudah pasti</li><li>▪ Maksimal 2000 karakter</li></ul>
NUMBER	Untuk isi data berupa angka numerik atau bilangan
DATE	Untuk isi data berupa tanggal dan waktu. Format default untuk kolom ini adalah DD-MON-YY.  Contoh : 11-NOV-86



# Tipe Data (2)

Tipe Data	Keterangan
NVARCHAR2	Digunakan untuk tipe data Unicode. Maksimal 4000 karakter, dengan panjang karakter yang dinamis sesuai dengan panjangmaksimal yang ditentukan.
NCHAR	Digunakan untuk tipe data Unicode juga. Maksimal 2000 karakter, tetapi panjang karakter nya sudah pasti sesuai dengan yang ditentukan.
LONG	Variable-Length karakter. Sampai 2 GB. Sebuah tabel hanya dapat terdiri dari satu kolom tipe data LONG. Constraint tidak dapat digunakan dalam kolom yang diberikan tipe data LONG. Jika anda akan menyimpan data seperti Video (AVI) sebaiknya anda menggunakan tipe data CLOB.



# Tipe Data (3)

Tipe Data	Keterangan
CLOB	Menyimpan karakter sampai 4 GB. Terdapat berbagai tipe data LOB (Large Object) yang digunakan untuk menyimpan file, documents, gambar, dan video.
BLOB	Untuk data binary, Max 4GB
ROWID	Alamat unik (base 64 string) dari sebuah row dalam tabel. Digunakan untuk menampung nilai dari ROWID pseudocolumns.



# Objek Database

- Tabel
- View
- Index
- Sequence
- Constraint



# Tabel

- ⇒ Tabel adalah sebuah objek database dengan struktur dua dimensi yang terdiri dari kolom dan baris
- ⇒ Fungsinya sebagai tempat penyimpanan data di dalam database





# Membuat Tabel

```
CREATE TABLE nama_tabel (nama_kolom tipe_kolom[size]);
```

Cth :

```
create table kategori  
(  
id_kategori number,  
nama_kategori varchar2(25)  
);
```

Melihat struktur tabel :

```
DESC nama_tabel;
```



# Modifikasi Tabel

Menambah/memodifikasi kolom/tipe data kolom:

```
ALTER TABLE nama_tabel ADD|MODIFY  
(nama_kolom tipe_kolom[size]);
```

Mengubah nama kolom:

```
ALTER TABLE nama_tabel RENAME COLUMN nama_kolom TO  
nama_kolom_baru;
```

Menghapus kolom:

```
ALTER TABLE nama_tabel DROP COLUMN nama_kolom;
```

Menghapus table:

```
DROP TABLE nama_tabel;
```



# View

View?

- ⇒ View adalah sebuah tabel *logic* yang sumber datanya berasal dari satu atau lebih tabel atau view lainnya
- ⇒ View merupakan query yang disimpan dalam suatu database (stored query)

Akan dipelajari pada Bab selanjutnya....



# Index

- ⇒ Catatan dari nilai yang muncul pada satu kolom
- ⇒ Mempercepat pencarian data

Index :

- Otomatis ⇒ Primary Key/Unique
- Manual



# Membuat Index

Membuat Index :

```
CREATE INDEX nama_index ON nama_tabel(nama_kolom);
```

Cth :

```
CREATE INDEX index_barang ON barang(id_barang);
```



# Sequence

⇒ Objek database yang berfungsi untuk membuat nilai/penomoran secara terurut kolom secara otomatis



# Membuat Sequence

```
CREATE SEQUENCE nama_sequence  
[ INCREMENT BY nilai_increment ]  
[ START WITH nilai_start ]  
[ MAXVALUE nilai_maximum | NOMAXVALUE ]  
[ MINVALUE nilai_minimum | NOMINVALUE ]  
[ CYCLE | NOCYCLE ]  
[ CACHE nilai_cache | NOCACHE ];
```

Cth :

```
create sequence sq_barang  
start with 1  
increment by 1;
```



# Constraint

- ⇒ Pembatas atau kekangan
- ⇒ Batasan yang diberikan kepada data – data yang akan diberikan kepada kolom tertentu pada sebuah tabel
- ⇒ Constraint bisa diberi nama, bisa juga tidak (disarankan untuk diberi nama)





# Macam-macam Constraint

- Not Null
- Unique
- Primary Key
- Foreign Key
- Check
- Default



# Not Null

⇒ Data tidak boleh kosong

Deklarasi saat CREATE TABLE :

```
CREATE TABLE nama_tabel (nama_kolom tipe_kolom NOT NULL);
```

Deklarasi saat ALTER TABLE adalah:

```
ALTER TABLE nama_tabel MODIFY  
(nama_kolom tipe_kolom NOT NULL);
```



# Unique

⇒ Data tidak boleh ada yang sama

Deklarasi saat CREATE TABLE :

```
CREATE TABLE nama_tabel (nama_kolom tipe_kolom UNIQUE);
```

atau

```
CREATE TABLE nama_tabel (nama_kolom tipe_kolom CONSTRAINT  
nama_constraint UNIQUE);
```

Deklarasi saat ALTER TABLE adalah:

```
ALTER TABLE nama_tabel MODIFY  
(nama_kolom tipe_kolom UNIQUE);
```



# Primary Key

⇒ Gabungan Not Null dan Unique sekaligus menjadi index

Deklarasi saat CREATE TABLE :

```
CREATE TABLE nama_tabel (nama_kolom tipe_kolom  
CONSTRAINT nama_constraint PRIMARY KEY);
```

Deklarasi saat ALTER TABLE adalah:

```
ALTER TABLE nama_tabel ADD CONSTRAINT  
nama_constraint PRIMARY KEY (nama_kolom);
```



# Foreign Key

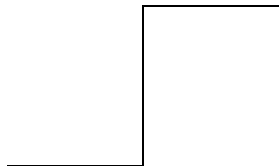
⇒ Menjaga integritas data antara tabel *parent* dengan table *child*

Tabel buku

id_buku	judul	id_kategori
1	Oracle XE	1
2	Belajar Web	1
3	Harry Potter	2

Tabel kategori

id_kategori	nama_kategori
1	Komputer
2	Novel
3	Komik





# Foreign Key (2)

Deklarasi saat CREATE TABLE :

```
CREATE TABLE nama_tabel (nama_kolom tipe_data,...,  
nama_kolom_n tipe_data, CONSTRAINT nama_constraint  
FOREIGN KEY (nama_kolom) REFERENCES  
nama_tabel_parent [ON DELETE CASCADE]);
```

*\*Disarankan untuk memberikan constraint Foreign Key saat ALTER TABLE*



# Foreign Key (3)

Deklarasi saat ALTER TABLE :

```
ALTER TABLE nama_tabel ADD CONSTRAINT  
nama_constraint FOREIGN KEY (nama_kolom_n)  
REFERENCES nama_tabel_parent [ON_DELETE_CASCADE];
```

Contoh saat ALTER TABLE :

```
ALTER TABLE buku ADD CONSTRAINT  
fk_buku_kategori FOREIGN KEY (id_kategori)  
REFERENCES kategori [ON_DELETE_CASCADE];
```



# Check

⇒ Menguji apakah data sesuai dengan syarat yang harus dipenuhi

Contoh :

- Kolom harga minimum 5000
- Kolom jenis kelamin hanya boleh 'L' atau 'P'





# Membuat Check

Deklarasi saat CREATE TABLE :

```
CREATE TABLE nama_tabel (nama_kolom tipe_kolom  
CONSTRAINT nama_constraint CHECK (syarat));
```

Deklarasi saat ALTER TABLE :

```
ALTER TABLE nama_tabel ADD CONSTRAINT  
nama_constraint CHECK (syarat);
```

Deklarasi saat ALTER TABLE :

```
ALTER TABLE buku ADD CONSTRAINT ck_buku CHECK  
(harga>5000);
```



# Default

⇒ Memberikan nilai default jika kolom tersebut tidak diisi

Deklarasi saat CREATE TABLE :

```
CREATE TABLE nama_tabel  
(nama_kolom tipe_data DEFAULT nilai_default);
```

Deklarasi saat ALTER TABLE adalah:

```
ALTER TABLE nama_tabel MODIFY nama_kolom tipe_data  
DEFAULT nilai_default;
```



# Manipulasi Constraint

Menghapus CONSTRAINT :

```
ALTER TABLE nama_tabel DROP CONSTRAINT  
nama_constraint;
```

Aktif/Non-aktif Constraint :

```
ALTER TABLE nama_table ENABLE | DISABLE CONSTRAINT  
nama_constraint;
```



# Kamus Data Constraint

Melihat CONSTRAINT yang sudah dibuat :

```
SELECT * FROM USER_CONSTRAINTS;
```

Melihat nama tabel dan nama kolom tempat CONSTRAINT berlaku :

```
SELECT * FROM USER_CONS_COLUMNS;
```



*Any Question?  
See you next time..*