



Basis Data Lanjut

Semester Genap 2018/2019

# **Data Manipulation Language (DML), Transaction Control Language (TCL) pada Oracle**

Salhazan Nasution, S.Kom, MIT



# Data Manipulation Language (DML)



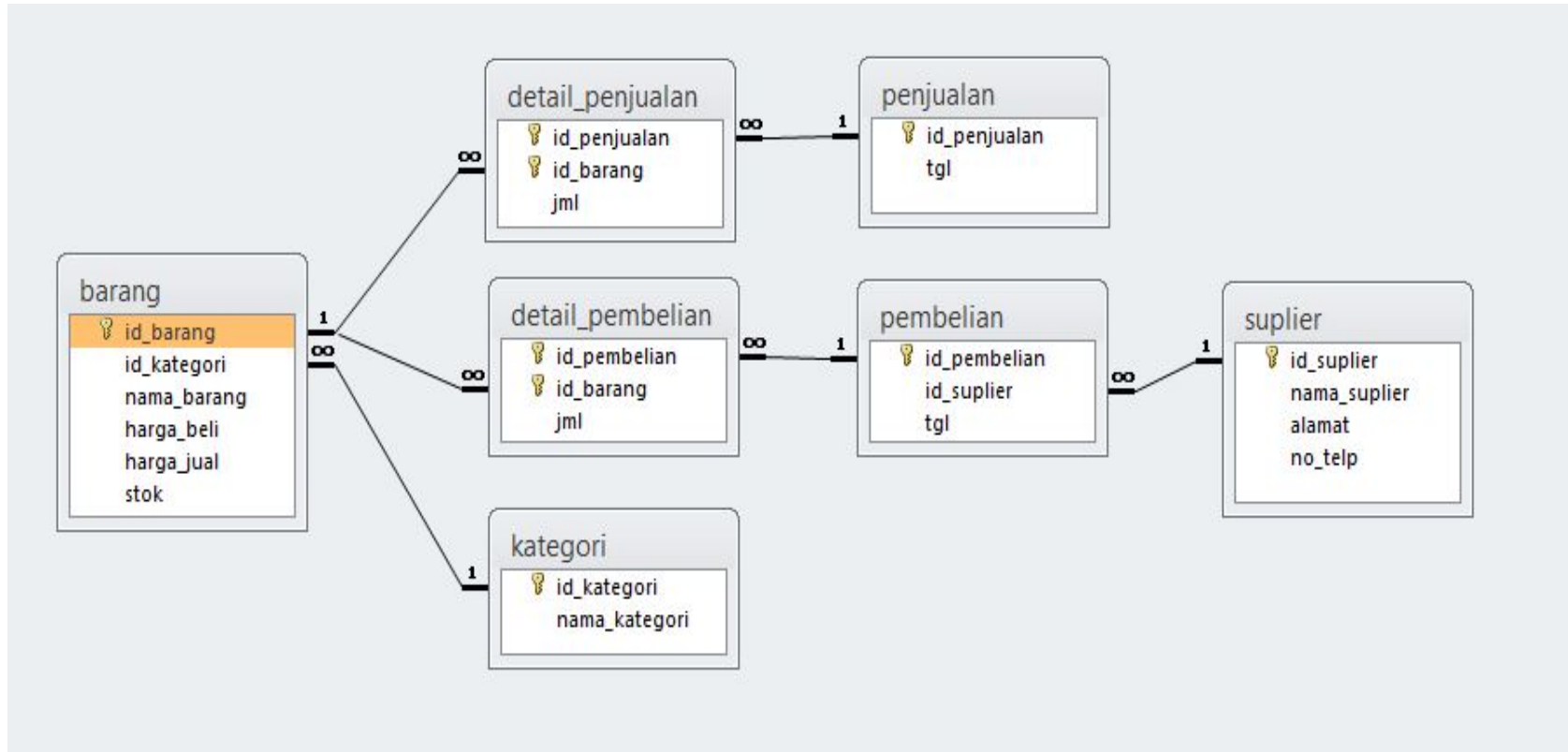
# Data Manipulation Language (DML)

⇒ Digunakan untuk memanipulasi data yang ada dalam suatu tabel.

**Contoh:** Insert, Select, Update, Delete



# Studi Kasus





# Insert

⇒ Untuk menambahkan data baru atau baris ke dalam tabel.

Sintak SQL :

```
INSERT INTO nama_table [(column [, column...])]  
VALUES (value [, value...]);
```



# Insert (2)

Contoh:

```
INSERT INTO kategori VALUES(1, 'mi numan ri ngan');
```

ID_KATEGORI	NAMA_KATEGORI
1	minuman ringan



# Insert (3)

Contoh:

```
INSERT INTO barang (id_barang, nama_barang, harga)  
VALUES(1, 'fanta', 3000);
```

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1		fanta	3000	



# Select

⇒ Untuk memilih atau menyeleksi data

Sintak SQL :

```
SELECT * | { [DISTINCT] column | expression [alias], ... }  
FROM nama_table  
[WHERE condition(s)];
```





# Select (2)

Contoh:

```
SELECT * FROM kategori ;
```

ID_KATEGORI	NAMA_KATEGORI
1	minuman ringan
2	permen
3	mie instan



# Select (3)

Contoh:

```
SELECT nama_barang, harga, stok FROM barang;
```

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	3000	10
2	3	Indomie	1000	20
3	2	relaxa	2000	15



# Update

⇒ Untuk memperbaharui data lama dengan data yang baru.

Sintak SQL :

```
UPDATE nama_table
```

```
SET nama_kolom = value [, column = value, ...]
```

```
[WHERE condition];
```



# Update (2)

Contoh:

```
UPDATE kategori SET nama_kategori = 'makanan ringan'  
WHERE id_kategori = 1;
```

ID_KATEGORI	NAMA_KATEGORI
1	makanan ringan
2	permen
3	mie instan



# Delete

⇒ Untuk menghapus data atau baris

Sintak SQL :

```
DELETE [FROM] nama_table [WHERE condition];
```



# Delete (2)

Contoh:

```
DELETE kategori WHERE id_kategori =3;
```

ID_KATEGORI	NAMA_KATEGORI
1	minuman ringan
2	permen



# Join

- ⇒ *Join* merupakan istilah yang digunakan untuk menggambarkan relasi yang terjadi antara satu tabel dengan tabel lainnya.
- ⇒ Menampilkan data dari dua atau lebih tabel yang memiliki hubungan.

**Join** : Inner Join, Outer join



# Inner Join

⇒ *Join* yang menampilkan data yang sama-sama dimiliki oleh tabel parent dan child

Sintak SQL :

```
SELECT nama_tabel 1. nama_kol om[ , ...]
```

```
FROM nama_tabel 1[ , ...]
```

```
WHERE nama_tabel 1. nama_kol om=nama_tabel 2. nama_kol om  
[AND | OR] . . . ;
```





# Inner Join (2)

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	3000	10
2	3	Indomie	1000	20
3	2	relaxa	2000	15

ID_KATEGORI	NAMA_KATEGORI
1	makanan ringan
2	permen
3	mie instan
4	wafer



## Inner Join (3)

```
SELECT barang. nama_barang, barang. harga,  
kategori . nama_kategori  
FROM barang, kategori  
WHERE  
barang. id_kategori =kategori . id_kategori ;
```



# Inner Join (4)

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	3000	10

NAMA_BARANG	HARGA	NAMA_KATEGORI
fanta	3000	minuman ringan
Indomie	1000	mie instan
relaxa	2000	permen

ID_KATEGORI	NAMA_KATEGORI
1	minuman ringan
2	permen
3	mie instan
4	wafer



# Inner Join (5)

```
SELECT barang.nama_barang, barang.harga,  
kategori.nama_kategori  
FROM barang  
JOIN kategori  
USING (id_kategori);
```



# Outer Join

⇒ *Join* yang menampilkan data parent walaupun pada tabel child tidak terdapat data tersebut.

Sintak SQL :

```
SELECT *  
FROM nama_tabel_parent, nama_tabel_child  
WHERE  
nama_tabel_parent.nama_kolom(+) =  
nama_tabel_child.nama_kolom;
```



# Outer Join (2)

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	3000	10

NAMA_BARANG	HARGA	NAMA_KATEGORI
fanta	3000	minuman ringan
Indomie	1000	mie instan
relaxa	2000	permen
		<b>wafer</b>

1	minuman ringan
2	permen
3	mie instan
4	wafer



# Alias

⇒ karakter/kata yang digunakan untuk menyingkat nama tabel, menyederhanakan query.



## Alias (2)

### Join:

```
SELECT barang.nama_barang, barang.harga,  
kategori.nama_kategori  
FROM barang, kategori  
WHERE barang.id_kategori=kategori.id_kategori;
```

### Menggunakan alias:

```
SELECT b.nama_barang, b.harga, k.nama_kategori  
FROM barang b, kategori k  
WHERE b.id_kategori=k.id_kategori;
```





# 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).



## View (2)

Sintak SQL :

```
CREATE [OR REPLACE] VIEW nama_view AS  
perintah_select_yang_dinginkan;
```

Contoh :

```
CREATE OR REPLACE VIEW vw_lihat_barang AS  
SELECT b.nama_barang, b.harga, k.nama_kategori  
FROM barang b, kategori k  
WHERE b.id_kategori=k.id_kategori;
```



# View (3)

Memanggil View :

```
SELECT * FROM vw_lihat_barang;
```

NAMA_BARANG	HARGA	NAMA_KATEGORI
fanta	3000	minuman ringan
Indomie	1000	mie instan
relaxa	2000	permen



# View (4)

Memanggil View :

```
SELECT nama_barang, nama_kategori FROM  
vw_lihat_barang;
```

NAMA	NAMA_BARANG	NAMA_KATEGORI	KATEGORI
fanta	fanta	minuman ringan	gan
Indomie	Indomie	mie instan	
relaxa	relaxa	permen	



# Transaction Control Language (TCL)



# Transaction Control Language (TCL)

- ⇒ Pada Oracle, hasil manipulasi data tidak langsung ditulis kedalam tabel, tapi ditampung dalam suatu Buffer.
- ⇒ Database Buffer Cache (menyimpan nilai baru) dan Redo log buffer (menyimpan nilai lama atau undo data). Nilai lama ini disimpan dalam segmen-segmen Undo Segment (dulu disebut Rollback Segment, RBS).



# TCL (2)

Perintah TCL :

- **Commit**

⇒ Menyimpan hasil transaksi ke dalam hardisk untuk dijadikan permanen

- **Rollback**

⇒ mengembalikan data ke nilai semula (undo)

- **Savepoint**

⇒ menandai point tertentu pada transaksi agar dapat di-rollback kembali



# TCL (3)

COMMIT otomatis bekerja, jika :

- DDL dieksekusi
- Keluar dari oracle / iSQL\*Plus secara normal

ROLLBACK otomatis bekerja , jika:

- Sesi dalam oracle / iSQL\*Plus berakhir secara tidak normal (electricity failure).
- Kerusakan sistem (system failure)





*Any Question?  
See you next time..*