



Basis Data Lanjut

Semester Genap 2018/2019

# Fungsi Bawaan, PL/SQL dan Exception Handling

Salhazan Nasution, S.Kom, MIT



# Fungsi Bawaan, PL/SQL dan Exception Handling



# Fungsi Bawaan Oracle

- Group Function

⇒ Mengoperasikan beberapa row untuk memberikan satu hasil, mirip dengan fungsi statistik

- Fungsi Karakter & String

⇒ Manipulasi karakter

- Fungsi Tanggal & Waktu

⇒ Manipulasi tanggal



# Group Function

- Count   ⇒ Menghitung jumlah baris data
- Sum     ⇒ Menghitung jumlah bilangan
- Avg     ⇒ Mencari nilai rata-rata
- Max     ⇒ Mencari nilai maksimum
- Min     ⇒ Mencari nilai minimum



## Group Function (2)

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	5000	20
2	3	indomie	1000	50
3	2	relaxa	3000	30

SQL> select COUNT (stok) from barang;

⇒ 3

SQL> select SUM (stok) from barang;

⇒ 100



# Group Function (3)

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	5000	20
2	3	indomie	1000	50
3	2	relaxa	3000	30

SQL> select **MIN** (stok) from barang;  
⇒ 20

SQL> select **AVG** (stok) from barang;  
⇒ 33,333



# Fungsi Karakter & String

- Concat / || ⇒ Menggabungkan string/karakter
- Length ⇒ Mendapatkan panjang string/karakter
- LPAD ⇒ Menambahkan karakter ke sebelah kiri karakter yang ada
- RPAD ⇒ Menambahkan karakter ke sebelah kanan karakter yang ada



# Fungsi Karakter & String (2)

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	5000	20
2	3	indomie	1000	50
3	2	relaxa	3000	30

SQL> select **CONCAT** (nama\_barang, harga) from barang;  
⇒ fanta5000

SQL> select nama\_barang || ' harganya : ' || harga  
from barang;  
⇒ fanta harganya : 5000





# Fungsi Karakter & String (3)

ID_BARANG	ID_KATEGORI	NAMA_BARANG	HARGA	STOK
1	1	fanta	5000	20
2	3	indomie	1000	50
3	2	relaxa	3000	30

```
SQL> select LENGTH (nama_barang) from barang;  
⇒ 5
```

```
SQL> select LPAD (id_barang, 3, '0') from barang;  
⇒ 001
```



# Fungsi Tanggal & Waktu

- SYSDATE           ⇒ Menampilkan tanggal hari ini
- LAST\_DAY           ⇒ Mendapatkan hari terakhir dari bulan tersebut
- SYSDATE+1       ⇒ Menampilkan tanggal 1 hari dari sekarang



## Fungsi Tanggal & Waktu (2)

SQL> select SYSDATE from DUAL;

⇒ 29-04-2008

SQL> select SYSDATE+7 from DUAL;

⇒ 06-05-2008

SQL> select LAST\_DAY(SYSDATE) from DUAL;

⇒ 30-04-2008



# PL/SQL

- ⇒ Teknologi dari Oracle yang memperbolehkan kita untuk membuat sebuah blok program dalam Oracle sebagaimana layaknya dalam bahasa pemrograman prosedural
- ⇒ dasar dari *stored procedure*, *function*, *package* dan *trigger*.



# Bagian PL/SQL

## ***Bagian Deklarasi***

**DECLARE**

== bagian deklarasi variable, jenis data dan sub program

## ***Bagian Eksekusi***

**BEGIN**

== bagian yang dapat dieksekusi perintah prosedural dan SQL

## ***Bagian Eksepsi***

**EXCEPTION**

== bagian yang menangani kesalahan (*exception handler*)



# DBMS\_OUTPUT

- ⇒ Paket PL/SQL yang menangani interaksi antara basisdata dengan user
- ⇒ Untuk menampilkan keluaran program pada SQL\*Plus

Ketika bekerja dengan SQL\*Plus, untuk dapat menampilkan hasil DBMS\_OUTPUT ke layar, maka sebelumnya harus diberikan perintah:

```
SQL> SET SERVEROUTPUT ON
```



# PL/SQL - Sederhana

```
begin
```

```
dbms_output.put_line (' Selamat Pagi ');
```

```
end;
```

```
/
```

⇒ Selamat Pagi



# PL/SQL - Tipe Data Dasar

**DECLARE**

```
nama_barang varchar2(20);
```

```
harga number;
```

**BEGIN**

```
nama_barang = 'Coffemix';
```

```
harga = 45000;
```

```
dbms_output.enable;
```

```
dbms_output.put_line('Nama barang: ' || nama_barang);
```

```
dbms_output.put_line('Harga: ' || harga);
```

**END;**

/

⇒ Nama barang: Coffemix

Harga : 45000





# %TYPE dan %ROWTYPE

⇒ Mendefinisikan tipe data agar tipe data yang diinginkan sama dengan jenis yang terdapat di tabel sehingga kompatibilitas dan konsistensinya terjamin

%TYPE : tipe data sesuai dengan tipe data pada kolom

%ROWTYPE : tipe data majemuk sesuai tipe data pada 1 baris



# PL/SQL - %TYPE

**DECLARE**

```
    stok_terbanyak barang.stok%TYPE;
```

**BEGIN**

```
    select max(stok) INTO stok_terbanyak from barang;
```

```
    dbms_output.enable;
```

```
    dbms_output.put_line(' stok barang terbanyak adalah  
: ' || stok_terbanyak);
```

**END;**

/

⇒ Stok barang terbanyak adalah : 50



# PL/SQL - %ROWTYPE

**DECLARE**

```
cari_barang barang%ROWTYPE;
```

**BEGIN**

```
select * INTO cari_barang from barang
```

```
where nama_barang='fanta';
```

```
dbms_output.enable;
```

```
dbms_output.put_line (' Harga : ' || cari_barang.harga);
```

```
dbms_output.put_line (' Stok : ' || cari_barang.stok);
```

**END;**

/

⇒ Harga : 5000

Stok : 20



# PL/SQL – input data

**DECLARE**

```
cari_barang barang%ROWTYPE;
```

**BEGIN**

```
select * INTO cari_barang from barang
```

```
where nama_barang=' &brg' ;
```

```
dbms_output.enable;
```

```
dbms_output.put_line (' Harga : ' || cari_barang.harga);
```

```
dbms_output.put_line (' Stok : ' || cari_barang.stok);
```

**END;**

/

⇒ Harga : 5000

Stok : 20



# Exception Handling

Exception Handling?

⇒ Mekanisme yang menggunakan suatu blok (*Exception Handler*) tertentu untuk menjebak error yang bisa mengakibatkan terhentinya program

- Predefined Exception
- User Defined Exception



# Predefined Exception

⇒ Exception yang telah disediakan oleh Oracle dan berhubungan dengan Oracle error yang umum



# Predefined Exception (2)

<i>Exception</i>	<i>Deskripsi</i>	<i>Kode Error/ SQLERRM</i>
<b>DUP_VAL_ON_INDEX</b>	Terdapat nilai yang sama pada constraint index atau primary key	ORA-00001
<b>INVALID_NUMBER</b>	Input karakter pada column yang bertipe number.	ORA-01722
<b>NO_DATA_FOUND</b>	Tidak ada data record yang dapat diambil setelah perintah query select	ORA-01403
<b>PROGRAM_ERROR</b>	Terjadi Error internal PL/SQL	ORA-06501
<b>TOO_MANY_ROWS</b>	Subquery yang menghasilkan banyak baris atau operasi SQL dimana oracle hanya mengharapkan hasil satu baris	ORA-01422
<b>ZERO_DIVIDE</b>	Membagi dengan NOL	ORA-01476



# Predefined Exception (3)

DECLARE

```
cari_barang barang%ROWTYPE;
```

BEGIN

```
select * into cari_barang from barang
```

```
where nama_barang='coca-cola';
```

```
dbms_output.enable;
```

```
dbms_output.put_line (' Harga : ' || cari_barang.harga);
```

```
dbms_output.put_line (' Stok : ' || cari_barang.stok);
```

Exception

```
when NO_DATA_FOUND then
```

```
dbms_output.put_line('Data tidak ditemukan');
```

```
END;
```

```
/
```





# User Defined Exception

- ⇒ Penanganan error yang didefinisikan sendiri secara manual oleh user
- ⇒ User defined exception dibangkitkan pada suatu keadaan tertentu yang telah didefinisikan oleh user



# User Defined Exception (2)

DECLARE

```
stk barang.stok%TYPE;
```

```
eks exception;
```

BEGIN

```
select stok into stk from barang
```

```
where id_barang='1';
```

```
if stk<=25 then RAISE eks;
```

```
end if;
```

EXCEPTION

```
WHEN eks THEN
```

```
dbms_output.put_line('Stok kurang dari 25 buah');
```

```
end;
```

```
/
```



# User Defined Exception - input

DECLARE

stk barang.stok%TYPE;

**eks** exception;

BEGIN

select stok into stk from barang

where id\_barang='&id';

if stk<=25 then RAISE eks;

end if;

EXCEPTION

WHEN **eks** THEN

dbms\_output.put\_line('Stok kurang dari 25 buah');

end;

/



*Any Question?  
See you next time..*