# PRAKTIKUM 11

# EXCEPTION (PL/SQL)

## A. Dasar Teori

Ekspresi merupakan identifier di dalam PL/SQL yang dipanggil ketika eksekusi yang akan menghentikan action blok PL/SQL. Sebuah blok akan selalu berhenti ketika PL/SQl memanggil ekspresi. Setiap kesalahan (error) atau peringatan (warning) yang muncul karena suatu perintah disebut dengan exception.

Exception Handling adalah Mekanisme yang menggunakan suatu blok (Exception Handler) tertentu untuk menjebak error yang bisa mengakibatkan terhentinya program. Bagian exception ini ditujukan untuk menangani exception tersebut. Bagian ini disebut juga exception handler, sedangkan mekanisme penanganan exception itu disebut exception handling.Jika tidak terdapat EXCEPTION dalam sebuah blok, maka semua proses akan terhenti dengan tanpa adanya penanganan kesalahan. Sebaliknya jika terdapat exception section, maka semua perintah berikutnya dalam bagian executable section tidak dieksekusi dan proses akan berpindah ke bagian exception handler.

Terdapat dua jenis exception:

1. Predefined exception, Exception yang telah disediakan oleh Oracle dan berhubungan dengan Oracle error yang umum. diantaranya adalah : NO_DATA_FOUND, DUP_VAL_ON_INDEX, TOO_MANY_ROWS, INVALID_NUMBER, ZERO_DEVIDE dan PROGRAM_ERROR.

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

Berikut ini contoh predefined exception.

```
set serveroutput on size 20000
DECLARE
vnim varchar2(5) := 12345;
vnama mahasiswa.nama%type;
vkomen char(25);
BEGIN
select nama into vnama
from mahasiswa
where nim=vnim;
exception
when no_data_found then
vkomen := 'Data tidak ada';
dbms_output.put_line(vkomen);
END; /
```

Contoh 2 :

```
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;
/
```

2. User-Defined Exception merupakan penanganan error yang didefinisikan oleh programmer dalam suatu script. Variabel exception dideklarasikan dan kemudian dalam program yang dapat dieksekusi, kondisi exception ditangkap dengan mempergunakan perintah if dan menyerahkannya ke variabel exception dengan perintah RAISE.

   Berikut ini contoh user-defined exception

```
DECLARE
Exep_lebih EXCEPTION;
v1 NUMBER := ‘&v1’;
v2 NUMBER := 99;
BEGIN
If v1 > v2 then
RAISE Exep_lebih;
```

```
Else
DBMS_OUTPUT.PUT_LINE('volume masih bisa menampung');
End if;
EXCEPTION
WHEN Exep_lebih THEN
DBMS_OUTPUT.PUT_LINE('volume '|| v1 ||'lebih dari '|| v2);
WHEN OTHERS then
DBMS_OUTPUT.PUT_LINE('volume tidak bisa menampung');
END;
```

**RAISE_APPLICATION_ERROR** merupakan sebuah prosedur yang disediakan oleh Oracle yang dapat digunakan untuk membuat dan sekaligus membangkitkan sebuah exception dengan cara membuat pesan dan nomor kesalahan untuk sebuah aplikasi. Nomor kesalahan harus berada dalam rentang nilai danSintaks:RAISE_APPLICATION_ERROR(angka_kesalahan, pesan[TRUE|FALSE]);

```
Begin
Insert into mahasiswa(nim, nama, alamat)
Values(' ','jammie', 'pga');
Exception
When DUP_VAL_ON_INDEX then
Raise_application_error (-20000, 'nim mahasiswa harus unik');
End;
```

**EXCEPTION_INIT PRAGMA** memungkinkan dibuatnya suatu exception (yang sifatnya sama dengan predefine exception) yang dikaitkan dengan nomor kesalahan. Keuntungan dari penggunaan exception_init ini adalah tidak diperlukan lagi pengecekan kondisi secara eksplisit.

```
declare
v_nama varchar2(20) := '&vnama';
v_nim char(9) := '&vnim';
v_alamat varchar2(25) := '&valamat';
jangan_null exception;
kepanjangan exception;
pragma exception_init(jangan_null,-1400);
pragma exception_init(kepanjangan,-6502);
begin
insert into Mahasiswa values(v_nim,v_nama,v_alamat);
exception
```

```
when jangan_null then
dbms_output.put_line('kolom harus diisi');
when kepanjangan then
dbms_output.put_line('isinya kepanjangan');
when others then dbms_output.put_line('apa coba yang salah?');
end;
```

Index adalah objek schema yang berisi catatan dari nilai-nilai yang muncul pada satu kolom atau kombinasi kolom di index dari sebuah tabel.Developer membuat indeks agar unjuk kerja aplikasi lebih baik. Perintah create index menghasilkan indeks dengan entry berupa nilai data yang diperoleh dari suatu kolom tunggal, gabungan beberapa kolom, ekspresi, dan fungsi.

Index bisa dibuat secara otomatis untuk constraint primary key atau unique key dan secara manual melalui CREATE INDEX statement.Ketika bekerja dengan indeks disarankan untuk mengacu pada kolom-kolom yang diindeks agar meningkatkan performansi join tabel, buatlah indeks dengan urutan kolom-kolom tabel yang tepat atau yang sering digunakan pada klausa where agar indeks digunakan untuk pencarian row.

Kolom yang dipilih sebagai bagian dari indeks sebaiknya mengandung nilai data yang unik atau kolom yang sering digunakan dalam klausa where. Jika perbedaan nilai data dari suatu atau beberapa kolom sangat bervariasi, gunakan indeks B-Tree. Sedangkan untuk nilai data yang kurang bervariasi gunakan indeks bitmap.Untuk performansi query Oracle mendukung penerapan indeks B-Tree yang merupakan indeks default, indeks bitmap untuk kumpulan key yang cardinality-nya rendah, indeks pada cluster B-Tree

**Membuat Index (CREATE INDEX)**

Query :

CREATE INDEX nama_index ON nama_tabel(nama_field1, nama_field2,….);

Contohnya :

SQL>CREATE INDEX mahasiswa_idx ON mahasiswa(nim, nama, alamat );

**Memodifikasi Index (CREATE INDEX)**

Sintaks:

ALTER INDEX nama_index

[INITRANS integer]

[MAXTRANS integer]

[STORAGE storage_clause]

Contoh :

SQL > ALTER INDEX mahasiswa_idx INITRANS 10;

SQL > ALTER INDEX mahasiswa_idx RENAME TO mhs_idx;

**Menghapus Index (CREATE INDEX)**

Sintaks:

DROP INDEX nama_index;

Contoh :

SQL > DROP INDEX mahasiswa_idx;

## B. Tugas Praktikum

1. Perhatikan blok anonim PL/SQL berikut:

```
1. DECLARE
2.     Var1 NUMBER;
3.     Var2 VARCHAR2(30);
4. BEGIN
5.     SELECT 10/0, kodekelas INTO Var1, Var2 from Kelas WHERE
       rownum=1;
6.     DBMS_OUTPUT.PUT_LINE (Var1||' '||Var2);
7. EXCEPTION
8.     WHEN ZERO_DIVIDE THEN
9.          DBMS_OUTPUT.PUT_LINE ('Bilangan tak hingga');
10. END;
11. /
```

a. Tuliskan yang anda pahami tentang blok PL/SQL tersebut!
b. Blok PL/SQL tersebut akan memiliki error, pada baris ke berapa errornya dan apa penyebabnya?

2. Perhatikan blok anonim PL/SQL berikut:

```
1. DECLARE
2.     Nilai NUMBER;
3.     Hasil NUMBER:=12;
4.   BEGIN
5.     Nilai := &input_nilai;
6.     Hasil := Nilai / 0;
7.   EXCEPTION
8.     WHEN ZERO_DIVIDE THEN
9.       DBMS_OUTPUT.PUT_LINE('Bilangan   tidak   boleh   dibagi
    nol!');
10.  END;
11.  /
```

a. Tuliskan dan jelaskan blok anonim PL/SQL yang memiliki Exception!
b. Tuliskan Output blok PL/SQL Diatas
c. Jika variabel Hasil nilainya ditampilkan pada baris ke-7 apa yang ditampilkan?
d. Tuliskan pengertian dari WHEN ZERO_DIVIDE pada bagis ke-8!
e. Apakah blok Exception harus selalu ditempatkan pada akhir sebuah blok PL/SQL? Jelaskan alasannya!

3. Perhatikan blok anonim PL/SQL berikut:

```
1. DECLARE
2.       Var1 DATE;
3.       Var2 VARCHAR2(32);
4. BEGIN
5.       Var1 := TO_DATE('&tanggal','DD-MON-YYYY');
6.       Var2 := TO_CHAR(Var1,'DAY');
7.       DBMS_OUTPUT.PUT_LINE ('Today is '||Var2);
8. EXCEPTION
9.       WHEN OTHERS THEN
10.           DBMS_OUTPUT.PUT_LINE (SQLCODE||'-'||SQLERRM);
11. END;
12. /
```

a. Perhatikan baris ke-5, apa maksud TO_DATE('&tanggal','DD-MON-YYYY'); ?
b. Perhatikan baris ke-6, apa maksud TO_CHAR(Var1,'DAY'); ?
c. Kapankah exception aktif? Berikan contoh input yang menyebabkan exception aktif!
d. Apakah SQLCODE dan SQLERRM itu?

4. Diketahui tabel kelas sebagai berikut:

| Kelas | | |
|---|---|---|
| **kdKelas** | **NamaKelas** | **Kapasitas** |
| 1 | PIS-10-01 | 40 |
| 2 | PIS-10-02 | 37 |
| 3 | PIS-10-03 | 40 |
| 4 | PIS-10-04 | 39 |
| 5 | PIS-10-05 | 40 |

Tabel kelas tersebut memiliki constraint Primary Key pada atribut kdkelas. Jika dieksekusi perintah INSERT INTO kelas (kdkelas, namakelas, kapasitas) VALUES (5, 'PIS-10-06',40) akan mengakibatkanterjadinya error. Gunakan exception untuk menangkap error tersebut dan menampilkan kode error serta pesan Kode Kelas tidak boleh Ganda!.

5. Tetap menggunakan table yang sama seperti pada soal nomer 4. Buatlah blok anonim PL/SQL untuk menampilkan informasi kdkelas. Gunakan SELECT INTO setelah BEGIN untuk menampilkan informasi tersebut, jika baris data yang didapatkan lebih dari satu, gunakan exception untuk menampilkan pesan Hanya dapat menampilkan satu kode kelas saja.