

Book Series  
of  
**jago{coding}**  
Learn & Share {Your Code}

[www.jagocoding.com](http://www.jagocoding.com)

# Tutorial Database Oracle Chapter4

## Pembatasan dan Pengurutan

Oleh: Ardie Jocong

**Pada Tutorial Database Oracle Chapter4 ini, saya akan Memberikan sedikit ilmu Bagaimana Melakukan Pembatasan dan Pengurutan di dalam Database Oracle..Pada Sesi ini saya akan membahas bagaimana melakukan pembatasan dengan menggunakan Perintah WHERE dan Pengurutan di dalam Database Oracle... Note : Query-Query yang diberikan ini berlaku juga Pada Aplikasi-Aplikasi Database lainnya seperti database MySQL, Ms. Access, SQLite, dkknya.. karena Secara Umum Semua Aplikasi Database memiliki Aturan yang tidak Jauh Berbeda..**

Pada Tutorial Database Oracle Chapter4 ini, saya akan Memberikan sedikit ilmu Bagaimana Melakukan Pembatasan dan Pengurutan di dalam Database Oracle.. jadi Pada Sesi ini saya akan membahas 2 Materi **bagaimana melakukan pembatasan dengan menggunakan Perintah WHERE dan Pengurutan** di dalam Database Oracle... Note : Query-Query yang diberikan ini berlaku juga Pada Aplikasi-Aplikasi Database lainnya seperti database MySQL, Ms. Access, SQLite, dkknya.. karena Secara Umum Semua Aplikasi Database memiliki Aturan yang tidak Jauh Berbeda.. namun di dalam database oracle sendiri terdapat beberapa query yang tidak dapat diterapkan dalam aplikasi Database Lain.. Perlu diketahui Selama ini database Oracle Dinilai Merupakan System Database Terbaik Yang Sering digunakan Oleh Perusahaan-perusahaan besar.. mungkin dari segi keamanannya (Oracle Memiliki System Security Yang paling Baik diantara yang lainnya)...

dalam Pembahasannya kita Akan Menggunakan Salah satu Tools yang terdapat di dalam Database Oracle yaitu SQL\*Plus yaitu sebuah Aplikasi Command yang digunakan untuk melakukan Perintah-perintah Query Untuk melakukan Aktifitas dengan Database..

Langkah Pertama Buka Dulu SQL\*Plus nya.... Jika Anda Belum menginstal Silahkan Instal Terlebih Dahulu Karena Tutorial Kali ini saya membahas dengan menggunakan Resource yang terdapat di database Bawaan Oracle. dan untuk melakukan Penginstalan Oracle silahkan untuk mengikuti Tutorial Part 1 klik di [sini](#).. disediakan link untuk download oracle nya juga..dan lebih baik lagi jika pembaca mengikuti tutorial2 sebelumnya....biar pembahasannya terstruktur...

Okeh Langsung saja pada Pembahasan Mengenai Pembatasan dan Pengurutannya. **Check These Out !!**

### PERINTAH WHERE

Perintah WHERE digunakan untuk membatasi baris yang dihasilkan oleh query. Perintah WHERE dapat membandingkan nilai Literal String serta ekspresi Aritmatika.

berikut ini adalah sebuah contoh penggunaan perintah WHERE untuk menampilkan kolom **employee\_id, last\_name, job\_id, department\_id** dari tabel **employees** dimana yang tampil adalah **department\_id** yang sama dengan 90

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE department_id=90;
```

Jika perintah **WHERE** digunakan untuk membatasi tampilan string maka harus diapit oleh apostrop yaitu simbol tanda petik tunggal ('...'), nilai untuk string adalah **case-sensitive (Berbeda Antara huruf kecil dan huruf besar, contoh: [a berbeda dengan A] )**. Perhatikan contoh berikut ..

```
SELECT last_name, job_id, department_id
FROM employees
WHERE last_name = 'Whalen';
```

Untuk Membatasi tampilan tanggal menggunakan default format tanggal yaitu DD-MON-RR. Contoh **WHERE hire\_date='01-JAN-95'**.

## KONDISI PERBANDINGAN

Operator	Keterangan
=	Sama Dengan
>	Lebih Besar
>=	Lebih Besar atau Sama Dengan
<	Lebih Kecil
<=	Lebih Kecil atau Sama Dengan
<>	Tidak Sama Dengan
<b>BETWEEN...AND...</b>	Diantara Dua Nilai
<b>IN (set)</b>	Menentukan Nilai Tertentu
<b>LIKE</b>	Menyamakan Karakter Tertentu
<b>IS NULL</b>	Nilai Null

**Tabel Kondisi Perbandingan**

Tabel diatas merupakan tabel Kondisi Perbandingan yang terdapat di dalam database Oracle. Kondisi perbandingan tersebut bisa diletakan di dalam perintah WHERE..

Contoh penerapan kondisi perbandingan untuk menampilkan nama dan gaji yang nilai gajinya lebih kecil dari 3000 atau bernilai 3000, maka gunakan perintah berikut :

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
```

perintah tersebut akan menampilkan kolom last\_name dan salary yang terdapat di dalam tabel employees dimana record yang tampil hanya record yang memiliki Nilai Salary dibawah 3000 atau

memiliki nilai 3000

## MENGGUNAKAN KONDISI BETWEEN

Perintah **BETWEEN** digunakan untuk menampilkan diantara dua nilai ... Nilai yang bisa ditampilkan dari semua tipe data. Berikut ini adalah contoh untuk menampilkan nama dan gaji dimana gaji yang tampil harus diantara 2500 dan 3500 maka dapat menggunakan perintah berikut :

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
```

## MENGGUNAKAN KONDISI IN

Kondisi IN Digunakan untuk menampilkan nilai tertentu. Nilai yang ditampilkan bisa berupa numerik ataupun karakter string. Jika nilai yang akan ditampilkan karakter string maka harus diapit oleh apostrop yaitu simbol tanda petik tunggal ('.....'),

Perintah berikut adalah untuk menampilkan kode pegawai, nama, gaji dan kode manajer dimana kode manajernya harus 100, 101, dan 201.

```
SELECT employeeid, last_name, salary, manager_id
FROM employees
WHERE manager_id IN(100,101,201);
```

## MENGGUNAKAN KONDISI LIKE

Operator **LIKE** digunakan untuk menampilkan data yang dicari dimana nilai kondisi yang dicarinya tidak utuh. Simbol yang digunakan menggunakan **wildcard**. yaitu simbol '%' (**persen**) untuk mewakili beberapa karakter sedangkan simbol '\_' (**underscore**) untuk mewakili satu karakter .

berikut adalah contoh penggunaan kondisi **LIKE** , untuk menampilkan nama karyawan dimana huruf keduanya adalah 'o',

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%';
```

Coba dengan Perintah berikut .

```
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
```

Perintah diatas adalah untuk menampilkan **first\_name** karyawan dimana record yang ditampilkan hanya yang berawalan huruf **S**.

Bagaimana bila output yang dicarinya adalah simbol % dan \_ ?, maka caranya dengan menggunakan pilihan **ESCAPE**. Untuk mencobanya buatlah sembarang tabel misal diberi nama TESLIKE isinya

satu kolom namanya TES tipenya CHAR 10 digit. Gunakan perintah perintah berikut , lakukan eksekusi Perbaris.

```
CREATE TABLE teslike(testing CHAR(10));
INSERT INTO teslike VALUES('COBA%1');
INSERT INTO teslike VALUES('COBA2');
INSERT INTO teslike VALUES('CO_BA3');
```

Perintah Pada Baris pertama adalah untuk membuat Sebuah tabel, sedangkan perintah pada baris-baris berikutnya adalah untuk mengisi Record Pada tabel yang telah dibuat sebelumnya .

Kemudian Coba jalankan perintah di bawah ini , amati bagaimana hasilnya :

```
SELECT * FROM TESLIKE WHERE TES LIKE '%\%%'ESCAPE'\';
```

### MENGGUNAKAN KONDISI NULL

Kondisi NULL akan menampilkan semua data yang tidak mempunyai nilai atau kosong. berikut ini adalah perintah untuk menampilkan nama belakang dan kode manager yang kode manajernya tidak ada .

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL;
```

Kondisi diatas akan menampilkan **last\_name, manager\_id** dari tabel **employees** dimana record yang tampil adalah yang tidak memiliki manager\_id atau **manager\_id** nya bernilai **kosong(null)** ;

### KONDISI LOGIKA

Operator	Keterangan
<b>AND</b>	Bernilai TRUE bila kedua komponennya bernilai benar
<b>OR</b>	Bernilai TRUE bila salah satu komponennya bernilai benar
<b>NOT</b>	Bernilai TRUE bila kondisi yang dihadapinya bernilai salah

### MENGGUNAKAN OPERATOR AND

Perintah ini akan menampilkan gaji yang lebih besar dari 10000 dan kode jabatannya yang mengandung string **MAN**.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >=10000
AND job_id LIKE '%MAN%';
```

Kode diatas akan menampilkan employee\_id, last\_name, job\_id, dan salary dari tabel employees dimana record yang akan tampil adalah yang memiliki nilai salary 10000 dan diatas 10000 dan yang memiliki job\_id nya Terdapat STRING **MAN**.

### MENGGUNAKAN OPERATOR OR

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >=10000
OR job_id LIKE '%MAN%';
```

### MENGGUNAKAN OPERATOR NOT

Operator **NOT** digunakan sebagai lawan dari nilai yang dihadapi.. Perhatikan... Berikut ini adalah perintah untuk menampilkan kode pekerjaan yang kodenya bukan **'IT\_PROG'**, **'ST\_CLERK'** dan **'SA\_REP'**

```
SELECT last_name, job_id
FROM employees
WHERE job_id
NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

### URUTAN Pengerjaan Operator (Precedensi)

Operator	Keterangan
1	Operator Aritmatika
2	Operator Konkatenation
3	Kondisi Komparasi
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Tidak Sama Dengan
7	Kondisi logika NOT
8	Kondisi logika AND
9	Kondisi logika OR

Contoh penerapannya adalah pada kode berikut :

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id='SA_REP'
OR job_id='AD_PRES'
AND salary>15000;
```

Kondisi diatas akan melakukan Pengerjaan pada bagian AND terlebih dahulu kemudian bandingkan dengan code berikut :

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id='SA_REP'
OR job_id='AD_PRES')
AND salary>15000;
```

kode yang kedua tersebut akan mengerjakan kondisi yang terdapat di dalam tanda kurung tersebut kemudian selanjutnya mengerjakan kondisi pada Bagian AND

## MENGGUNAKAN ORDER BY

**ORDER BY** digunakan untuk mengurutkan data hasil query. Cara mengurutkan data ada dua jenis yaitu secara **ASCENDING** (mengurutkan menaik) dan **DESCENDING** (mengurutkan menurun). Defaultnya adalah **ASCENDING**. **ORDER BY** diletakkan di akhir **SELECT**. Perhatikan Pada Perintah Berikut.

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date;
```

kode diatas adalah untuk menampilkan last\_name, job\_id, department\_id, hire\_date dari tabel employees yang di susun atau di urutkan berdasarkan **hire\_date**.. secara Ascending.. karena tidak didefinisikan. Query berikut ini hampir sama dengan query diatas hanya saja dalam pengurutannya adalah secara **Descending** atau **Menurun**..

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC;
```

selain itu juga perintah **ORDER** bisa mengurutkan data menggunakan Nama Alias dari kolom secara utuh maupun kolom yang sudah menggunakan operator aritmatika. berikut contoh untuk mengurutkan dengan nama kolom alias:

```
SELECT employee_id, last_name, salary*12 totalgaji
FROM employees
ORDER BY totalgaji;
```

Perintah diatas adalah untuk menampilkan employee\_id ,salary per tahun dengan kolom alias totalgaji yang diurutkan berdasarkan totalgaji..

pemakaian ORDER BY juga bisa menggunakan parameter dua kolom atau lebih ... berikut ini contohnya :

```
SELECT last_name, department_id, salary
FROM employees
```

```
ORDER BY department_id, salary DESC;
```

Perintah diatas adalah akan menampilkan kolom last\_name, department\_id, salary dari tabel employees dengan mengurutkan berdasarkan kolom **department\_id** dan kolom **salary** secara **Descending** atau **Menurun**.

untuk materi Pembatasan sdan Pengurutan saya cukupkan segitu dulu, Okeh.. Sekian dulu!! Happy Coding .. dan mudah2an bermanfaat !! Walaupun Sedikit Pembahasannya !.. Silahkan Berkomentar Jika Ada Pertanyaan !!

---

## Tentang Penulis



### Ardie Jocong

let's we learn coding together ,.. Learning by doing.. cause the best way of learning about anything is by doing. explore your brain ..