

Materi III

Nama : Roy Agus Martin Marbun

Kelas : XII – RPL

Mata Pelajaran : Basis Data

BAB I

Implementasi SQL

A. Penggunaan Sintaks dan Subquery (Query Nested)

1. *Subquery* dengan Operator **EXIST** dan **NOT EXIST**

Operator **EXISTS** merupakan jenis operator Boolean yang menghasilkan nilai benar (**TRUE**) atau salah (**FALSE**). Operator **EXIST** akan memberikan nilai benar (**TRUE**) kalau *subquery* menghasilkan paling tidak sebuah baris/record. Jika hasilnya kosong maka *query* utama tidak akan dijalankan. Lawan dari statement **EXIST** adalah **NOT EXISTS**.

Berikut adalah contoh penggunaan **EXISTS** dan **NOT EXISTS**.

```
SELECT * FROM pegawai WHERE EXISTS (SELECT * FROM pegawai WHERE  
alamat_peg = 'Yogyakarta');
```

Query utama tersebut akan menampilkan info pegawai yang dimana alamat pegawai adalah Yogyakarta.

Dan hasilnya :

nip	nama_peg	alamat_peg	jabatan	gaji
PEG-1001	Soeharto Mangundirejo	Yogyakarta	C.E.O	9000000
PEG-1002	Felix Nababan	Medan	Manager	8000000
PEG-1003	Olga Syahputra	Jakarta	Kepala Unit	6000000
PEG-1004	Chelsea Olivia	Bandung	Kepala Unit	6000000
PEG-1005	Tuti Wardani	Jawa Tengah	Supervisor	4500000
PEG-1006	Budi Drajat	Malang	Supervisor	4500000
PEG-1007	Bambang Pamungkas	Kudus	Staff Senior	3000000
PEG-1008	Ely Oktafiani	Yogyakarta	Staff Senior	3000000
PEG-1009	Rani Wijaya	Magelang	Staff Senior	3000000
PEG-1010	Rano Karno	Solo	Staff Junior	2000000
PEG-1011	Rahmadi Sholeh	Yogyakarta	Staff Junior	2000000
PEG-1012	Ilham Ungara	Jakarta	Staff Junior	2000000
PEG-1013	Endang Melati	Madiun	Staff Junior	2000000
PEG-1014	Donny Damara	Makasar	Tenaga Kontrak	1000000
PEG-1015	Paijem	Yogyakarta	Tenaga Kontrak	500000

Dari perintah di atas menunjukkan bahwa *query* utama dieksekusi karena *subquery* bernilai **TRUE** atau ada hasilnya yang dimana terdapat pegawai yang alamatnya adalah Yogyakarta.

Berikutnya jika kita mengganti kata **EXIST** menjadi **NOT EXISTS** dan alamatnya kita ganti menjadi Kebumen maka akan bernilai **TRUE**. Karena di dalam data pegawai tidak ada alamat pegawai yang bernilai Kebumen.

SELECT * FROM pegawai WHERE NOT EXISTS (SELECT * FROM pegawai WHERE alamat_peg = 'Kebumen');

Dan hasilnya :

nip	nama_peg	alamat_peg	jabatan	gaji
PEG-1001	Soeharto Mangundirejo	Yogyakarta	C.E.O	9000000
PEG-1002	Felix Nababan	Medan	Manager	8000000
PEG-1003	Olga Syahputra	Jakarta	Kepala Unit	6000000
PEG-1004	Chelsea olivia	Bandung	kepala Unit	6000000
PEG-1005	Tuti wardani	Jawa Tengah	Supervisor	4500000
PEG-1006	Budi Drajat	Malang	Supervisor	4500000
PEG-1007	Bambang Pamungkas	Kudus	Staff Senior	3000000
PEG-1008	Ely oktafiani	Yogyakarta	Staff Senior	3000000
PEG-1009	Rani wijaya	Magelang	Staff Senior	3000000
PEG-1010	Rano Karno	Solo	Staff Junior	2000000
PEG-1011	Rahmadi Sholeh	Yogyakarta	Staff Junior	2000000
PEG-1012	Ilham Ungara	Jakarta	Staff Junior	2000000
PEG-1013	Endang Melati	Madiun	Staff Junior	2000000
PEG-1014	Donny Damara	Makasar	Tenaga Kontrak	1000000
PEG-1015	Paijem	Yogyakarta	Tenaga Kontrak	500000

2. Subquery dengan Operator ANY (SOME)

Operator **ANY** hampir sama penggunaannya seperti **EXISTS**. Tetapi operator relasi yang digunakan biasanya selain = (sama dengan). Hal tersebut disebabkan apabila operator relasi = yang digunakan, maka sebetulnya fungsi operator **ANY (SOME)** sama seperti operator **IN**, sehingga kondisi seperti itu tidak dianjurkan karena lebih mudah pemahamannya apabila menggunakan operator **IN**.

Berikut adalah contoh penggunaan **ANY**.

```
SELECT nama_peg, gaji FROM pegawai WHERE gaji < ANY (SELECT gaji FROM pegawai WHERE gaji > '2000000');
```

nama_peg	gaji
Felix Nababan	8000000
Olga Syahputra	6000000
Chelsea Olivia	6000000
Tuti Wardani	4500000
Budi Drajat	4500000
Bambang Pamungkas	3000000
Ely Oktafiani	3000000
Rani Wijaya	3000000
Rano Karno	2000000
Rahmadi Sholeh	2000000
Ilham Ungara	2000000
Endang Melati	2000000
Donny Damara	1000000
Paijem	500000

Pada *query* diatas dapat diperhatikan bahwa *query* **SELECT** yang berada di dalam kurung (*sub query*), pada *query* itu menjelaskan bahwa penampilan daftar nama pegawai yang gajinya lebih kecil daripada hasil *sub query*-nya.

3. *Subquery* dengan Operator **ALL**

Operator **ALL** digunakan untuk melakukan perbandingan dengan *subquery*. Kondisi dengan **ALL** menghasilkan nilai benar (**TRUE**) jika perbandingan menghasilkan nilai **TRUE** untuk setiap nilai dalam *subquery*.

Berikut adalah contoh penggunaan **ALL**.

SELECT nama_peg, gaji **FROM** pegawai **WHERE** gaji < **ALL** (**SELECT** gaji **FROM** pegawai **WHERE** gaji > '2500000');

nama_peg	gaji
Rano Karno	2000000
Rahmadi Sholeh	2000000
Ilham Ungara	2000000
Endang Melati	2000000
Donny Damara	1000000
Paijem	500000

Contoh di samping menampilkan semua nama pegawai dan gaji yang dimana jumlah gajinya kurang dari 2,5 juta.

4. *Subquery* dengan Operator **IN**

Operator **IN** digunakan untuk menampilkan data yang hasilnya lebih dari satu. Berbeda dengan Operator '=' yang hanya menampilkan hasil yang cuma satu.

Berikut adalah format dari *subquery* **IN**.

```
SELECT namakolom FROM namatabel WHERE kondisi operasi perbandingan  
IN (Subquery);
```

5. *Subquery* dengan *Single Row*

Single Row subquery memberikan hasil hanya satu baris pada bagian *subquery*. Untuk *single row subquery* ini yang digunakan adalah operator perbandingan seperti >, >=, <, <=, atau <>.

Contoh :

Menampilkan data guru yang memiliki jabatan sama dengan Roy.

```
SELECT nama_guru, gelar FROM daftarguru WHERE gelar = (SELECT gelar FROM  
daftarguru WHERE nama_guru = 'Roy');
```

6. *Subquery* dengan Multiple Rows

Multiple Row Subquery adalah *subquery* yang menghasilkan lebih dari satu baris data. Untuk *multiple row subquery*, yang digunakan adalah operator perbandingan **IN**, **ANY** atau **ALL**.

Contoh :

Menampilkan data karyawan yang bekerja pada bagian kasir atau pada ruang 5.

```
SELECT last_name, first_name, title FROM kary WHERE dept_id IN (SELECT id  
FROM departemen WHERE name = 'KASIR' OR ruang_id = 5);
```