



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA YOGYAKARTA**

**MODUL 12  
BAHASA QUERY**

## Menggunakan Operator-operator Himpunan

### A. TUJUAN

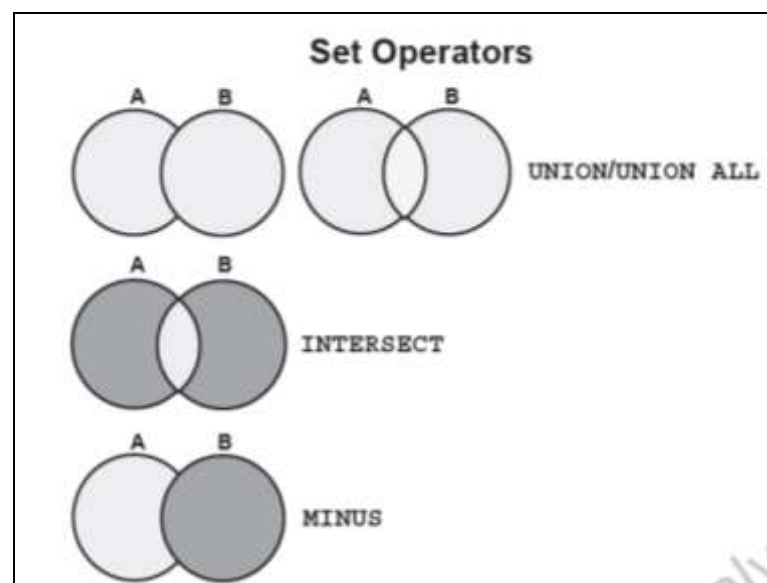
Setelah menyelesaikan praktikum ini, diharapkan mahasiswa dapat :

1. Mendeskripsikan operator-operator himpunan.
2. Menggunakan suatu operator himpunan untuk menggabungkan *multiple query* menjadi *single query*.
3. Menulis query menggunakan operator-operator himpunan.
4. Mengontrol urutan baris-baris hasil query.

### B. LANDASAN TEORI & LANGKAH PRAKTIKUM

#### B.1. Macam Operator-operator Himpunan

Operator-operator himpunan menggabungkan hasil dari 2 atau lebih komponen *query* menjadi satu hasil. *Query* yang mengandung operator himpunan disebut *compound query*. Macam-macam operator himpunan adalah sebagai berikut :



**Gambar 12.1. Macam Operator-operator Himpunan (Oracle, 2004)**

Operator	Hasil query
UNION	Semua baris yang berbeda ( <i>distinct rows</i> ) dari salah query.
UNION ALL	Semua baris hasil query, termasuk semua duplikasi.
INTERSECT	Semua baris yang berbeda ( <i>distinct rows</i> ) dari kedua query.
MINUS	Semua baris yang berbeda ( <i>distinct rows</i> ) yang terpilih dari statement SELECT yang pertama dan tidak terpilih dari statement SELECT yang kedua.

Semua operator himpunan memiliki tingkat hirarki yang sama. Jika sebuah *query* mengandung beberapa operator himpunan, maka Oracle akan mengeksekusi dari kiri (atas) ke kanan (bawah). Jika kita akan mengubah urutan hirarki gunakan tanda kurung ().

Tabel-tabel yang digunakan dalam praktikum ini adalah :

- Tabel EMPLOYEES : berisi detail data yang berkaitan dengan semua pegawai saat ini. Sejumlah pegawai yang telah bekerja cukup lama, ada kalanya berganti job yang berbeda. Hal ini dimonitor menggunakan Tabel JOB\_HISTORY.
- Tabel JOB\_HISTORY : merekam detail data tentang tanggal mulai, tanggal akhir pekerjaan lama, beserta no id pekerjaan dan departemen saat pegawai berganti pekerjaan.

Struktur Tabel EMPLOYEES adalah :

DESCRIBE employees;		
Name	Null	Type
-----		
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

Struktur Tabel JOB\_HISTORY adalah :

```
describe JOB_HISTORY;
```

Name	Null	Type
-----	-----	-----
EMPLOYEE_ID	NOT NULL	NUMBER(6)
START_DATE	NOT NULL	DATE
END_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
DEPARTMENT_ID		NUMBER(4)

Relasi antar kedua buah tabel tersebut dapat dilihat pada lampiran modul praktikum ini.

Sebagai contoh :

```
SELECT employee_id, last_name, job_id, hire_date, department_id
FROM employees;
```

Results	Script Output	Explain	Autotrace	DBMS Output	OWA Output
Results:					
	EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE	DEPARTMENT_ID
1	100	King	AD_PRES	17-JUN-87	90
2	101	Kochhar	AD_VP	21-SEP-89	90
3	102	De Haan	AD_VP	13-JAN-93	90
...					
77	176	Taylor	SA_REP	24-MAR-98	80
78	177	Livingston	SA_REP	23-APR-98	80
79	178	Grant	SA_REP	24-MAY-99	(null)
80	179	Johnson	SA_REP	04-JAN-00	80
...					
96	195	Jones	SH_CLERK	17-MAR-99	50
97	196	Walsh	SH_CLERK	24-APR-98	50
98	197	Feeney	SH_CLERK	23-MAY-98	50
99	198	OConnell	SH_CLERK	21-JUN-99	50
100	199	Grant	SH_CLERK	13-JAN-00	50
101	200	Whalen	AD_ASST	17-SEP-87	10
102	201	Hartstein	MK_MAN	17-FEB-96	20
103	202	Fay	MK_REP	17-AUG-97	20
104	203	Mavris	HR_REP	07-JUN-94	40
105	204	Baer	PR_REP	07-JUN-94	70
106	205	Higgins	AC_MGR	07-JUN-94	110
107	206	Gietz	AC_ACCO...	07-JUN-94	110

Ada sejumlah pegawai yang memiliki posisi job yang sama lebih dari satu kali selama dia bekerja di perusahaan. Misal : Taylor bergabung di perusahaan pada tanggal 24-MAR-98. Dia memiliki job sebagai SA\_REP untuk periode 24-MAR-98 sampai 31-DEC-98 dan job sebagai SA\_MAN untuk periode 01-JAN-99 sampai 31-DEC-99. Taylor kemudian kembali sebagai SA\_REP yang menjadi job nya sampai saat ini.

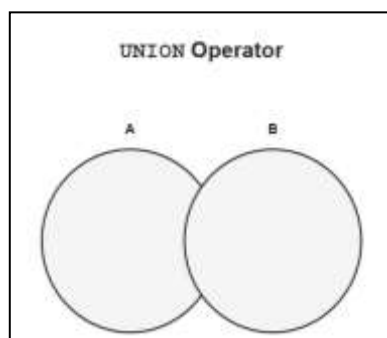
Demikian pula Whalen yang bergabung di perusahaan pada tanggal 17-SEP-1987. Dia bekerja sebagai AD\_ASST untuk periode 17-SEP-87 sampai 17-JUN-93, kemudian berganti sebagai AC-ACCOUNT untuk periode 01-JUL-94 sampai 31-DEC-98. Setelah itu dia kembali sebagai AD\_ASST sampai saat ini.

Semua perubahan tersebut dicatat dalam Tabel JOB\_HISTORY. Lihat isi Tabel JOB\_HISTORY, perhatikan record dengan Employee\_ID 176 (Taylor) dan 200 (Whalen).

select * from JOB_HISTORY						
Results						
Results:						
	EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID	
1	102	13-JAN-93	24-JUL-98	IT_PROG	60	
2	101	21-SEP-89	27-OCT-93	AC_ACCOUNT	110	
3	101	28-OCT-93	15-MAR-97	AC_MGR	110	
4	201	17-FEB-96	19-DEC-99	MK_REP	20	
5	114	24-MAR-98	31-DEC-99	ST_CLERK	50	
6	122	01-JAN-99	31-DEC-99	ST_CLERK	50	
7	200	17-SEP-87	17-JUN-93	AD_ASST	90	
8	176	24-MAR-98	31-DEC-98	SA_REP	80	
9	176	01-JAN-99	31-DEC-99	SA_MAN	80	
10	200	01-JUL-94	31-DEC-98	AC_ACCOUNT	90	

## B.2. Operator UNION

Operator UNION mengembalikan hasil dari kedua *query* setelah menghilangkan duplikasi-duplikasi.



Gambar 12.2. Operator UNION (Oracle, 2004)

Aturan penggunaan operator UNION :

- ✓ Jumlah kolom dan tipe datanya harus identik antara satu statemen SELECT yang satu dengan statemen SELECT yang lain. Sedangkan nama kolom boleh tidak identik.
- ✓ Nilai null tidak diabaikan selama pengecekan duplikasi.
- ✓ Secara *default*, output *query* akan diurutkan secara *ascending* sesuai kolom pertama.

**Contoh 1:**

- Tampilkan job saat ini maupun job terdahulu untuk seluruh karyawan.

```
SELECT employee_id, job_id
FROM employees
UNION
SELECT employee_id, job_id
FROM job_history;
```

Hasil :

	EMPLOYEE_ID	JOB_ID
1	100	AD_PRES
2	101	AC_ACCOUNT
3	101	AC_MGR
4	101	AD_VP
5	102	AD_VP
6	102	IT_PROG

...

109	201	MK_MAN
110	201	MK_REP
111	202	MK_REP
112	203	HR_REP
113	204	PR_REP
114	205	AC_MGR
115	206	AC_ACCOUNT

**Catatan :**

- Perhatikan bahwa operator UNION akan menghilangkan duplikasibaris/record. Jika record hasil *query* tabel EMPLOYEES dan JOB\_HISTORY adalah identik, maka record di tampilkan hanya sekali saja (tanpa duplikasi).

**Contoh 2:**

- Tampilkan job saat ini maupun job terdahulu beserta departmentnya untuk seluruh karyawan.

```
SELECT employee_id, job_id, department_id
FROM employees
UNION
SELECT employee_id, job_id, department_id
FROM job_history;
```

Hasil :

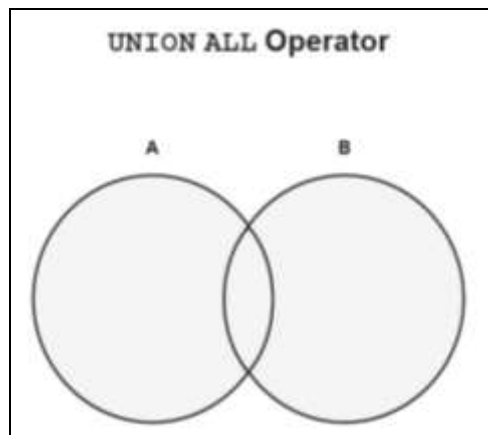
	EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
107	200	AC_ACCOUNT	90
108	200	AD_ASST	10
109	200	AD_ASST	90
110	201	MK_MAN	20
111	201	MK_REP	20
112	202	MK_REP	20
113	203	HR_REP	40
114	204	PR_REP	70
115	205	AC_MGR	110
116	206	AC_ACCOUNT	110

**Catatan :**

- Bandingkan hasil query contoh ke-1 dan ke-2. Mengapa employee\_id 200 muncul 2 kali pada contoh ke-1 sedangkan padaa contoh ke-2 muncul 3 kali?

### B.3. Operator UNION ALL

Operator UNION mengembalikan semua hasil dari kedua *query*, termasuk semua duplikasi.



**Gambar 12.3. Operator UNION ALL (Oracle, 2004)**

Petunjuk penggunaan operator UNION ALL:

- ✓ Secara umum aturan untuk UNION dan UNION ALL adalah sama, kecuali dalam 2 hal yaitu :
  - Keyword DISTINCT tak boleh digunakan.
  - Output tidak otomatis (default) tampil secaraurut.

#### **Contoh 3:**

- Tampilkan job saat ini maupun job terdahulu beserta departemennya untuk seluruh karyawan.

```
SELECT employee_id, job_id, department_id
FROM employees
UNION ALL
SELECT employee_id, job_id, department_id
FROM job_history
ORDER BY employee_id;
```

**Hasil :**

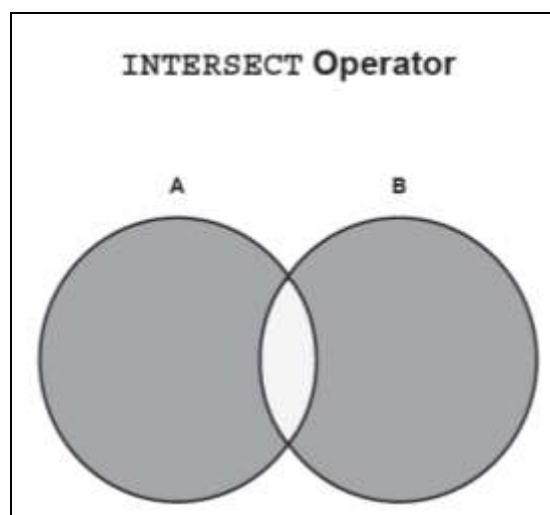
	EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
1	100	AD_PRES	90
2	101	AD_VP	90
3	101	AC_ACCOUNT	110
4	101	AC_MGR	110
5	102	IT_PROG	60
6	102	AD_VP	90
7	103	IT_PROG	60
8	104	IT_PROG	60
....			
108	200	AD_ASST	10
109	200	AD_ASST	90
110	200	AC_ACCOUNT	90
111	201	MK_MAN	20
112	201	MK_REP	20
113	202	MK_REP	20
114	203	HR_REP	40
115	204	PR_REP	70
116	205	AC_MGR	110
117	206	AC_ACCOUNT	110

**Catatan :**

- Tampak terdapat 117 record. Ubahlah dengan menggunakan operator UNION. Berapa record yang muncul? Hanya 116 record saja karena mengeliminasi employee\_id 176 (dianggap duplikasi record).

**B.4. Operator INTERSECT**

Operator INTERSECT mengembalikan hasil yang sama-sama dimiliki (irisan) dari kedua *query*.



**Gambar 12.4. Operator INTERSECT (Oracle, 2004)**



Petunjuk penggunaan operator INTERSECT:

- ✓ Jumlah kolom dan tipe datanya harus identik antara satu statemen SELECT yang satu dengan statemen SELECT yang lain. Sedangkan nama kolom boleh tidak identik.
- ✓ Peletakan tabel yang ditukar (atas/bawah) tidak mempengaruhi hasil query.
- ✓ Operator INTERSECT tidak mengabaikan nilai null.

#### Contoh 4:

- Tampilkan employee\_ID dan job\_ID bagi karyawan yang saat ini memiliki job title yang sama dengan saat mereka pertama kali masuk bekerja (yaitu siapa saja yang telah berubah job namun kembali lagi ke job awal.

```
SELECT employee_id, job_id,
FROM employees
INTERSECT
SELECT employee_id, job_id,
FROM job_history;
```

Hasil :

	EMPLOYEE_ID	JOB_ID
1	176	SA_REP
2	200	AD_ASST

#### Contoh 5:

- Modifikasi contoh 4 dengan menambahkan kolom department\_id. Amati apakah perbedaan contoh 4 dibandingkan contoh 5.

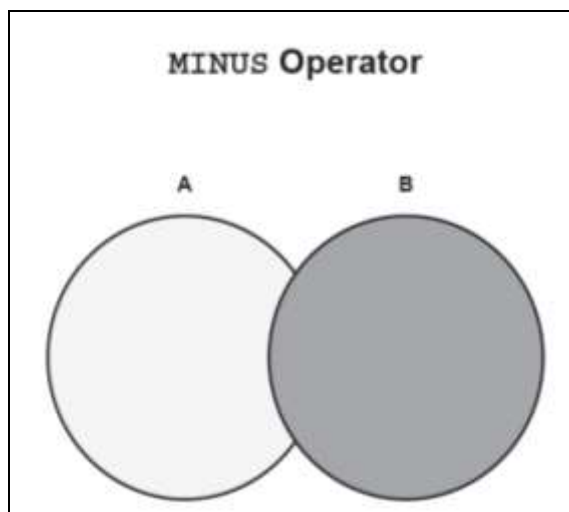
```
SELECT employee_id, job_id, department_id
FROM employees
INTERSECT
SELECT employee_id, job_id, department_id
FROM job_history;
```

Hasil :

	EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID
1	176	SA_REP	80

### B.5. Operator MINUS

Operator MINUS mengembalikan hasil dari *query* pertama yang tidak merupakan hasil *query* kedua. (Statemen SELECT pertama MINUS statemen SELECT kedua).



Gambar 12.5. Operator MINUS (Oracle, 2004)

Petunjuk penggunaan operator MINUS:

- ✓ Jumlah kolom dan tipe datanya harus identik antara satu statemen SELECT yang satu dengan statemen SELECT yang lain. Sedangkan nama kolom boleh tidak identik.
- ✓ Semua kolom yang digunakan pada klausa WHERE harus ada pula pada klausa SELECT agar operator MINUS dapat bekerja.

#### Contoh 6:

- Tampilkan employee\_ID dan job\_ID bagi karyawan yang sama sekali tidak pernah sekalipun berubah posisi (*job*) nya.

```
SELECT employee_id, job_id,
FROM employees
MINUS
SELECT employee_id, job_id,
FROM job_history;
```

Hasil :

	EMPLOYEE_ID	JOB_ID
1	100	AD_PRES
2	101	AD_VP
3	102	AD_VP
4	103	IT_PROG
5	104	IT_PROG
6	105	IT_PROG
7	106	IT_PROG
8	107	IT_PROG
...		
96	196	SH_CLERK
97	197	SH_CLERK
98	198	SH_CLERK
99	199	SH_CLERK
100	201	MK_MAN
101	202	MK_REP
102	203	HR_REP
103	204	PR_REP
104	205	AC_MGR
105	206	AC_ACCO...

**Keterangan :**

- Pada contoh di atas, Employee\_ID dan Job\_ID pada Tabel JOB\_HISTORY dikurangkan pada Table EMPLOYEES. Hasil query menampilkan pegawai-pegawai yang merupakan sisa dari hasil pengurangan, yaitu mereka yang ada di tabel EMPLOYEES namun tidak ada di tabel JOB\_HISTORY. Record-record tersebut adalah pegawai yang tidak pernah berubah posisi (job) nya bahkan sekalipun.

## B.6. Penggunaan dan Peletakan Klausa WHERE

Petunjuk / aturan :

- ✓ Query yang menggunakan operator UNION, UNION ALL, INTERSECT, dan MINUS maka pada klausa WHERE harus memiliki Jumlah kolom dan tipe data yang sama dengan yang ada pada klausa SELECT.
- ✓ Klausa ORDER BY hanya dapat muncul sekali pada akhir statemen. Boleh menggunakan nama alias pada klausa ORDER BY.
- ✓ Nama kolom atau alias, jika digunakan pada klausa ORDER BY, harus diletakkan pada urutan pertama dalam klausa SELECT.

**Contoh 7 (Penulisan Klausa WHERE):**

```

SELECT employee_id, department_id
FROM employees
WHERE (employee_id, department_id)
      IN (SELECT employee_id, department_id
          FROM employees
          UNION
          SELECT employee_id, department_id
          FROM job_history);

```

**Hasil :**

	EMPLOYEE_ID	DEPARTMENT_ID
96	196	50
97	197	50
98	198	50
99	199	50
100	200	10
101	201	20
102	202	20
103	203	40
104	204	70
105	205	110
106	206	110

**Contoh 8:**

Pada contoh di bawah ini memberikan contoh bahwa jumlah dan tipe kolom harus sama.

Dengan demikian harus dituliskan angka 0 sebagai pengganti kolom salary pada Tabel JOB\_HISTORY.

```

SELECT employee_id, job_id, salary
FROM employees
UNION
SELECT employee_id, job_id, 0
FROM job_history;

```

Hasil :

	EMPLOYEE_ID	JOB_ID	SALARY
1	100	AD_PRES	24000
2	101	AC_ACCOUNT	0
3	101	AC_MGR	0
4	101	AD_VP	17000
5	102	AD_VP	17000
6	102	IT_PROG	0
7	103	IT_PROG	9000
8	104	IT_PROG	6000
9	105	IT_PROG	4800
10	106	IT_PROG	4800
11	107	IT_PROG	4200
12	108	FI_MGR	12000
13	109	FI_ACCOUNT	9000
14	110	FI_ACCOUNT	8200
15	111	FI_ACCOUNT	7700
16	112	FI_ACCOUNT	7800
17	113	FI_ACCOUNT	6900
18	114	PU_MAN	11000



### B.7. Kesimpulan

- Gunakan UNION untuk memperoleh hasil semua baris tanpa duplikasi.
- Gunakan UNION ALL untuk memperoleh hasil semua baris termasuk duplikasi.
- Gunakan INTERSECT untuk memperoleh hasil semua baris yang sama-sama dimiliki oleh kedua *query*.
- Gunakan MINUS untuk memperoleh hasil baris yang berbeda (*distinct rows*) dari hasil *query* pertama namun tidak merupakan hasil *query* kedua.
- Gunakan ORDER BY hanya pada akhir statemen.

### C. TUGAS

1. Buatlah query untuk menampilkan Department\_ID yang tidak mengandung job\_ID ST\_CLERK. Gunakan operator himpunan untuk membuat query ini!

DEPARTMENT_ID	
	10
	20
	60
	80
	90
	110
	190

2. Tampilkan daftar negara-negara (Country\_ID dan Country\_Name) yang tidak memiliki department yang terletak di negara tersebut. Gunakan operator himpunan untuk membuat query ini!

CO	COUNTRY_NAME
DE	Germany

3. Tampilkan daftar job pada department 10,50, dan 20. Tampilkan JOB\_ID dan Department\_ID menggunakan operator himpunan.
4. Tampilkan Employee\_ID dan JOB\_ID bagi karyawan yang saat ini memiliki posisi (job) yang sama dengan posisi (job) saat mereka pertama kali masuk bekerja (yaitu mereka yang telah berganti posisi/job namun kembali ke job awal mereka).

EMPLOYEE_ID	JOB_ID
176	SA_REP
200	AD_ASST

### D. DAFTAR PUSTAKA

Oracle Database 10g : SQL Fundamental, Oracle Inc. 2004

☺ *Practices Makes Perfect* ☺

## LAMPIRAN STRUKTUR TABEL DAN RELASI TABEL

