



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

**MODUL 1
BAHASA QUERY**

Mengambil Data Menggunakan Perintah SQL SELECT

A. TUJUAN

1. Mahasiswa dapat menyebutkan kemampuan-kemampuan perintah SELECT.
2. Mahasiswa mampu menggunakan perintah dasar SELECT untuk mengambil data dari database.

B. LANDASAN TEORI & LANGKAH PRAKTIKUM

B.1. Kemampuan perintah SELECT

1. **Projection** : Memilih kolom dalam suatu tabel menggunakan *query*. Anda dapat memilih satu, beberapa, atau banyak kolom sesuai kebutuhan.
2. **Selection** : Memilih baris dalam suatu tabel menggunakan *query*. Berbagai kriteria dapat digunakan untuk membatasi baris-baris yang akan ditampilkan.
3. **Joining** : Mengambil data dari beberapa tabel yang berbeda dimana tabel-tabel tersebut saling berelasi.

Projection

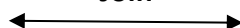
Tabel 1

Selection

Tabel 2

Tabel 1

Join



Tabel 2

Gambar 1. Kemampuan-kemampuan Perintah SQL SELECT

B.2. Sintaks umum dari perintah SELECT

```
SELECT *|{[DISTINCT] column|expression [alias],...}
FROM    table;
```

- SELECT mengidentifikasi kolom-kolom yang akan ditampilkan.
- FROM mengidentifikasi tabel yang mengandung kolom-kolom tsb.

Keterangan:

Sebuah **perintah SELECT** harus mengandung :

- Sebuah **klausula SELECT** yang berfungsi untuk menspesifikasikan kolom-kolom yang akan ditampilkan.
- Sebuah **klausula FROM** yang berfungsi menunjukkan tabel dimana kolom-kolom tersebut berada.

Parameter	Keterangan
SELECT	menampilkan satu atau beberapa kolom.
*	menampilkan semua kolom.
<i>DISTINCT</i>	menghilangkan duplikasi.
<i>column/expression</i>	memilih nama kolom atau expression.
<i>alias</i>	memberi nama lain pada kolom yang dipilih.
<i>FROM table</i>	Menspesifikasikan tabel yang mengandung kolom-kolom tsb.

B.3. Aturan Penulisan Perintah-Perintah SQL :

- Perintah SQL tidak *case-sensitive* (huruf besar dan huruf kecil tidak dibedakan)
- Perintah SQL dapat berbentuk 1 baris atau beberapa baris,
- Keyword tidak boleh dipisahkan.
- Klausula (*clause*) biasanya diletakkan pada baris terpisah.
- Inden dapat digunakan untuk mempermudah pembacaan.

Cobalah contoh-contoh statement SELECT seperti tersebut di bawah ini :

a. Memilih semua kolom

```
SELECT *
FROM departments;
```

Results:				
	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400
5	50	Shipping	121	1500
6	60	IT	103	1400
7	70	Public Relations	204	2700
8	80	Sales	145	2500

Keterangan :

- Anda dapat menampilkan semua kolom dari semua baris yang ada pada sebuah tabel dengan menggunakan perintah SELECT diikuti asterisk (*).
- Perhatikan gambar 2 di atas, tabel Departments terdiri dari 4 kolom (field) yaitu : department_id, department_name, manager_id, dan location_id. Tabel tersebut seluruhnya terdiri dari 8 baris.
- Anda dapat pula menampilkan semua kolom dengan cara menyebutkan semua kolom setelah perintah SELECT, sehingga perintah tersebut di atas dapat pula ditulis sbb:

```
SELECT department_id, department_name, manager_id, location_id
FROM departments;
```

c. Memilih kolom tertentu

```
SELECT department_id, location_id
FROM departments;
```

Results:			
	DEPARTMENT_ID	LOCATION_ID	
4	40	2400	
5	50	1500	
6	60	1400	
7	70	2700	
8	80	2500	
9	90	1700	
10	100	1700	
11	110	1700	
12	120	1700	
13	130	1700	
14	140	1700	

Keterangan :

- Untuk menampilkan kolom tertentu gunakan perintah SELECT diikuti nama kolom yang mana antara satu nama kolom dengan nama kolom yang lain dipisahkan dengan tanda koma (,).
- Perhatikan bahwa urutan nama kolom yang anda tulis dalam perintah SELECT akan sama urutannya dengan output yang ditampilkan. Misal : bila anda ingin membalik urutan menampilkan location_id terlebih dulu sebelum department_id maka perintahnya adalah sbb :

```
SELECT location_id, department_id
FROM departments;
```

d. Menghilangkan duplikasi baris

Default perintah query akan menampilkan semua baris termasuk baris (record) dengan data yang sama.

Perhatikan dan cobalah 2 contoh query di bawah ini serta amati perbedaannya !

```
SELECT department_id
FROM employees;
```

Results:	
	DEPARTMENT_ID
1	90
2	90
3	90
4	60
5	60
6	60
7	60
8	60
9	100
10	100

```
SELECT DISTINCT department_id
FROM employees;
```

Results:	
	DEPARTMENT_ID
1	100
2	30
3	(null)
4	90
5	20
6	70
7	110
8	50
9	80
10	40

Catatan :

- Pada query no 1 ditampilkan semua department_id yang ada pada tabel employee, tanpa mengeliminasi baris dengan data yang sama.
- Dengan memberikan perintah DISTINCT setelah SELECT maka hanya ada 7 department yang benar-benar unik.
- Perintah DISTINCT dapat juga digunakan untuk beberapa kolom sekaligus. Cobalah query di bawah ini dan amati hasilnya :

```
SELECT DISTINCT department_id, job_id
FROM employees;
```

B.4. Ekspresi Aritmatika

Anda dapat membuat ekspresi aritmatika dengan data bertipe tanggal (*date*) dan bertipe angka (*number*) dengan menggunakan operator aritmatika sbb:

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide

Hirarki urutan :

- Perkalian dan pembagian : prioritas ke-1
- Penjumlahan dan pengurangan : prioritas ke-2
- Jika operator memiliki prioritas yang sama, maka akan dieksekusi dari kiri ke kanan.
- Tanda kurung dapat digunakan untuk mengubah hirarki prioritas.

Cobalah contoh query di bawah ini :

a. Menggunakan operator aritmatika :

```
SELECT last_name, salary, salary+300
FROM employees;
```

Results:			
	LAST_NAME	SALARY	SALARY+300
1	King	24000	24300
2	Kochhar	17000	17300
3	De Haan	17000	17300
4	Hunold	9000	9300
5	Ernst	6000	6300
6	Austin	4800	5100
7	Pataballa	4800	5100
8	Lorentz	4200	4500
9	Greenberg	12000	12300

Keterangan :

- Query di atas bertujuan untuk menampilkan nama pegawai, gaji lama, dan gaji setelah mengalami kenaikan \$300 bagi seluruh pegawai.
- Perhatikan bahwa hasil perhitungan salary+300 tidak membentuk sebuah kolom baru pada tabel employee.

b. Mengubah hirarki operator

Cobalah kedua query di bawah ini dan amatilah hasilnya :

```
SELECT last_name, salary, 12*salary+100
FROM employees;
```

Results:			
	LAST_NAME	SALARY	12*SALARY+100
1	King	24000	288100
2	Kochhar	17000	204100
3	De Haan	17000	204100
4	Hunold	9000	108100
5	Ernst	6000	72100
6	Austin	4800	57700
7	Pataballa	4800	57700
8	Lorentz	4200	50500
9	Greenberg	12000	144100
10	Faviet	9000	108100

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

Results:			
	LAST_NAME	SALARY	12*(SALARY+100)
1	King	24000	289200
2	Kochhar	17000	205200
3	De Haan	17000	205200
4	Hunold	9000	109200
5	Ernst	6000	73200
6	Austin	4800	58800
7	Pataballa	4800	58800
8	Lorentz	4200	51600
9	Greenberg	12000	145200
10	Faviet	9000	109200

Catatan :

- Pada query no 1 : menampilkan last_name, salary, serta salary per tahun ditambah bonus 1 kali sebesar \$100. Ingat bahwa perkalian memiliki hirarki lebih tinggi dibandingkan penjumlahan.
- Sedangkan pada query no 2 : menampilkan last_name, salary, serta salary per tahun dimana salary pertahun dihitung dari salary tiap bulan ditambah bonus \$100 kemudian dikalikan 12 bulan. Perhatikan bahwa tanda kurung akan mengubah urutan hirarki operator, sehingga penjumlahan salary+100 akan dikerjakan terlebih dulu.

B.5. Nilai Null (Null Value)

Nilai Null (Null Value) adalah :

- Null adalah sebuah nilai yang tidak tersedia atau tidak diketahui.
- Null tidak sama dengan 0 (nol) atau spasi kosong (*blank space*) karena 0 (nol) adalah suatu angka (number) dan spasi adalah suatu character.

Contoh :

```
SELECT last_name, job_id, salary, commission_pct
FROM employees;
```

Results:				
	LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
1	King	AD_PRES	24000	(null)
2	Kochhar	AD_VP	17000	(null)
3	De Haan	AD_VP	17000	(null)
4	Hunold	IT_PROG	9000	(null)
5	Ernst	IT_PROG	6000	(null)
6	Austin	IT_PROG	4800	(null)
7	Pataballa	IT_PROG	4800	(null)

Keterangan :

- Pada kolom commission_pct pada tabel employee tampak bahwa hanya karyawan yang pekerjaannya sebagai sales manager atau sales representative yang memiliki commission_pct sedangkan yang lainnya bernilai null.

Contoh Nilai Null dalam Ekspresi Aritmatika

```
SELECT last_name, 12*salary*commission_pct
FROM employees;
```

Results:		
	LAST_NAME	12*SALARY*COMMISSION_PCT
1	King	(null)
2	Kochhar	(null)
3	De Haan	(null)
4	Hunold	(null)
5	Ernst	(null)
6	Austin	(null)
7	Pataballa	(null)

Perhatikan bahwa ekspresi aritmatika yang mengandung nilai null akan menghasilkan nilai null juga.

B.6. Mendefinisikan suatu kolom alias

Kolom Alias :

- Pemberian nama lain suatu header kolom.
- Berguna untuk memberi nama kolom yang merupakan hasil perhitungan
- Gunakan tanda petik (" ") bila nama alias mengandung spasi, mengandung karakter khusus (\$, #), atau bersifat case sensitive.

Contoh penggunaan kolom alias :

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

Results:

	NAME	COMM
1	King	(null)
2	Kochhar	(null)
3	De Haan	(null)
4	Hunold	(null)
5	Ernst	(null)

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

Results:

	Name	Annual Salary
1	King	288000
2	Kochhar	204000
3	De Haan	204000
4	Hunold	108000
5	Ernst	72000

Cobalah kedua query tersebut di atas dan amatilah hasilnya !

Catatan :

- Penggunaan keyword AS untuk memberi nama alias bersifat optional. Hasil query akan tetap sama baik menggunakan keyword AS atau tidak.
- Kolom alias akan ditampilkan secara default dalam bentuk huruf besar (*uppercase*). Bila anda akan mengubah default uppercase gunakan tanda *double quotation* (" "), misal : "Name".
- Bila nama alias mengandung spasi, gunakan tanda *double quotation* (" ") , misal : "Annual Salary"

Perhatikan bahwa output hasil query akan menggunakan nama kolom alias.

B.7. Operator Concatenation

Anda dapat mengaitkan/ menggabungkan beberapa kolom membentuk satu kolom output menggunakan operator *concatenation* (`||`). Perhatikan contoh berikut ini dimana **last_name** dan **job_id** dikaitkan/digabungkan membentuk satu kolom output dengan nama alias **employees**.

Contoh penggunaan operator concatenation :

```
SELECT last_name || job_id AS "Employees"
FROM employees;
```

Results:

	Employees
1	AbelSA_REP
2	AndeSA_REP
3	AtkinsonST_CLERK
4	AustinIT_PROG
5	BaerPR_REP
6	BaidaPU_CLERK

Catatan :

- Keyword AS sebelum nama alias bertujuan agar perintah SELECT tsb menjadi lebih mudah dan jelas dibaca.

B.8. Literal Character Strings

Literal :

- dapat berbentuk character, number, atau date yang ada dalam clausa SELECT namun bukan merupakan nama kolom atau kolom alias.
- Literal akan ditampilkan pada setiap baris output hasil query.
- Literal berbentuk date dan character harus ditulis dalam tanda `' '`.

Contoh penggunaan *Literal Character String* :

```
SELECT last_name || ' is a ' || job_id AS "Employee Details"
FROM employees;
```

Results:

	Employee Details
1	Abel is a SA_REP
2	Ande is a SA_REP
3	Atkinson is a ST_CLERK
4	Austin is a IT_PROG
5	Baer is a PR_REP
6	Baida is a PU_CLERK

Catatan :

- Pada query di atas ditampilkan last_name, dan job_id yang mana ditengahnya diberi keterangan 'is a'.
- 'is a' disebut *literal character string* yang penulisannya harus di dalam tanpa petik (' ') mengingat bertipe character.
- Perhatikan bahwa setelah tanda petik awal ada spasi demikian pula sebelum tanda petik akhir juga diberi spasi dengan tujuan untuk mempermudah pembacaan.

Editlah query tersebut dengan menghilangkan spasi pada ' is a ' ! Perhatikan perbedaannya.

B.9. Menampilkan Struktur Tabel

Struktur tabel dapat ditampilkan menggunakan perintah DESCRIBE atau dapat pula ditulis 4 character di depan saja yaitu DESC. Catatan : semua perintah SQL yang panjangnya lebih dari 4 character dapat ditulis hanya 4 character paling depan saja.

Contoh :

Untuk melihat struktur tabel employees digunakan perintah :

```
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)
11 rows selected		

atau dapat pula disingkat menjadi :

```
DESC employees;
```

D. TUGAS

Untuk mengerjakan tugas di bawah ini digunakan tabel-tabel sebagai berikut :

EMPLOYEES

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT
1	100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	(null)
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	(null)
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	(null)
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	(null)
5	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	(null)
6	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT_PROG	4800	(null)
7	106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT_PROG	4800	(null)
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	(null)
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FI_MGR	12000	(null)
10	109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-94	FI_ACCOUNT	9000	(null)

DEPARTMENTS

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400
5	50	Shipping	121	1500
6	60	IT	103	1400
7	70	Public Relations	204	2700
8	80	Sales	145	2500
9	90	Executive	100	1700
10	100	Finance	108	1700

1. Tampilkan employee_id, first_name, job_id, salary dari tabel employees.
2. Tampilkan tanpa duplikasi job_id apa saja yang ada di tabel employees.
3. Tampilkan employee_id, first_name, last_name, hire_date dengan tampilan first_name dan Last_name digabungkan menjadi satu kolom dengan header kolom menjadi 'Full Name'.
4. Terdapat 4 kesalahan pada query di bawah ini :

```
SELECT employee_id, last_name
      Sal x 12      ANNUAL SALARY
FROM employees;
```

 Betulkan query tersebut!
5. Tampilkan last_name dan salary dari tabel employees dengan output tampilan sebagai berikut :

Gaji Perbulan
King : 1 bulan gaji = 24000
Kochhar : 1 bulan gaji = 17000
De Haan : 1 bulan gaji = 17000
....

6. Tampilkan struktur tabel Departments.
7. Buatlah query dengan output sebagai berikut :

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8. Buatlah query untuk menampilkan employee_id, last_name, job_id, hire_date dengan output header kolom sebagai berikut :

Emp #	Employee	Job	Hire Date
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorentz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98

9. Buatlah query untuk menampilkan output sebagai berikut :

THE_OUTPUT													
100	Steven	King	SKING	515.123.4567	AD_PRES	17-JUN-87	24000	90					
101	Neena	Kochhar	NKOCHHAR	515.123.4568	AD_VP	21-SEP-89	17000	90					
102	Lex	De Haan	LDEHAAN	515.123.4569	AD_VP	13-JAN-93	17000	90					
103	Alexander	Hunold	AHUNOLD	590.423.4567	IT_PROG	03-JAN-90	9000	60					
104	Bruce	Ernst	BERNST	590.423.4568	IT_PROG	21-MAY-91	6000	60					
107	Diana	Lorentz	DLORENTZ	590.423.5567	IT_PROG	07-FEB-99	4200	60					
124	Kevin	Mourgos	KMOURGOS	650.123.5234	ST_MAN	16-NOV-99	5800	50					
141	Trenna	Rajs	TRAJS	650.121.8009	ST_CLERK	17-OCT-95	3500	50					
142	Curtis	Davies	CDAVIES	650.121.2994	ST_CLERK	29-JAN-97	3100	50					
143	Randall	Matos	RMATOS	650.121.2874	ST_CLERK	15-MAR-98	2600	50					
144	Peter	Vargas	PVARGAS	650.121.2004	ST_CLERK	09-JUL-98	2500	50					

E. DAFTAR PUSTAKA

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

😊 Selamat Berlatih 😊