

MODUL 2

DATA MANIPULATION LANGUAGE DAN RETRIEVE DATA

MODUL 2 : DATA MANIPULATION LANGUAGE (DML) DAN RETRIEVE DATA

Tujuan :

Setelah menyelesaikan modul ini, Mahasiswa diharapkan dapat :

1. Mencari dan menampilkan data dengan perintah select
2. Mengkombinasikan perintah select dengan perintah lainnya

Pembahasan

- a. Memberikan nama lain pada kolom :

SELECT namakolomlama AS namakolombaru FROM namatabel;

Contoh: Buat tabel pegawai dan desc tabel pegawai seperti berikut.

```
mysql> create table pegawai
-> (NIP char(6) primary key,
-> Nama Varchar(20),
-> Alamat Varchar(30),
-> id_div char(3),
-> foreign key fk_id_div(id_div) references divisi(id_div));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| NIP    | char(6)       | NO   | PRI | NULL    |       |
| Nama   | varchar(20)   | YES  |     | NULL    |       |
| Alamat | varchar(30)   | YES  |     | NULL    |       |
| id_div | char(3)       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

Berikut ini perintah untuk memberikan nama lain pada kolom **nama** menjadi **nama_pegawai** pada tabel pegawai:

SELECT Nama AS Nama_Pegawai FROM pegawai;

```
mysql> select * from pegawai;
```

NIP	Nama	Alamat	id_div
P01	Arika	Jl. Tanjung Sari No.5	KEU
P02	Andri	Griya Bumi Asri A.20	LIT
P03	Saskia	Jl. Karanganyar No.3	LIT
P04	Budi	Jl. Cempaka No.7	KEU
P05	Wanda	Jl. Merdeka No.16	LOG

```
5 rows in set (0.00 sec)
```



```
mysql> select Nama as Nama_Pegawai from pegawai;
```

Nama_Pegawai
Arika
Andri
Saskia
Budi
Wanda

```
5 rows in set (0.00 sec)
```

- b. Menggunakan alias untuk nama tabel:

Berikut ini perintah untuk memberikan alias pada tabel pegawai :

SELECT p.Nama, p.Alamat, p.id_div FROM pegawai p;

```
mysql> select p.Nama, p.Alamat, p.id_div from pegawai p;
```

Nama	Alamat	id_div
Arika	Jl. Tanjung Sari No.5	KEU
Andri	Griya Bumi Asri A.20	LIT
Saskia	Jl. Karanganyar No.3	LIT
Budi	Jl. Cempaka No.7	KEU
Wanda	Jl. Merdeka No.16	LOG

```
5 rows in set (0.00 sec)
```

- c. Menampilkan data lebih dari dua tabel :

SELECT * FROM namatabel1, namatabel2, namatabel-n;

d. *Nested Queries / Subquery* (**IN, NOT IN, EXISTS, NOT EXISTS**)

Subquery berarti *query* di dalam *query*. Dengan menggunakan *subquery*, hasil dari *query* akan menjadi bagian dari *query* di atasnya. *Subquery* terletak di dalam klausa **WHERE** atau **HAVING**.

Pada klausa **WHERE**, *subquery* digunakan untuk memilih baris-baris tertentu yang kemudian digunakan oleh *query*. Sedangkan pada klausa **HAVING**, *subquery* digunakan untuk memilih kelompok baris yang kemudian digunakan oleh *query*.

Contoh 1:

Perintah untuk menampilkan data pada tabel pegawai yang mana data pada kolom id_div-nya tercantum pada tabel divisi menggunakan **IN** :

SELECT * FROM pegawai WHERE id_div IN (SELECT id_div FROM divisi); atau menggunakan **EXISTS**.

```
mysql> select * from pegawai where id_div in(select id_div from divisi);
```

NIP	Nama	Alamat	id_div
P01	Arika	Jl. Tanjung Sari No.5	KEU
P02	Andri	Griya Bumi Asri A.20	LIT
P03	Saskia	Jl. Karangan No.3	LIT
P04	Budi	Jl. Cempaka No.7	KEU
P05	Wanda	Jl. Merdeka No.16	LOG

```
5 rows in set (0.00 sec)
```

Pada contoh di atas:

SELECT id_div FROM divisi disebut *subquery*, sedangkan

SELECT * FROM Pegawai berkedudukan sebagai *query*.

Contoh 2: Perintah untuk menampilkan data pada tabel pegawai yang mana data pada kolom id_div-nya tidak tercantum pada tabel divisi menggunakan **NOT IN**:

SELECT * FROM pegawai WHERE id_div NOT IN (SELECT id_div FROM divisi);
atau menggunakan **NOT EXISTS**.

e. Sintak **ORDER BY**

Klausa ORDER BY digunakan untuk mengurutkan data berdasarkan kolom tertentu sesuai dengan tipe data yang dimiliki.

Contoh : Perintah untuk mengurutkan data pegawai berdasarkan nama.

SELECT * FROM pegawai ORDER BY nama;

```
mysql> select * from pegawai order by nama;
+----+-----+-----+-----+
| NIP | Nama   | Alamat                | id_div |
+----+-----+-----+-----+
| P02 | Andri  | Griya Bumi Asri A.20  | LIT    |
| P01 | Arika  | Jl. Tanjung Sari No.5 | KEU    |
| P04 | Budi   | Jl. Cempaka No.7      | KEU    |
| P03 | Saskia | Jl. Karangan No.3     | LIT    |
| P05 | Wanda  | Jl. Merdeka No.16     | LOG    |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```

atau tambahkan ASC untuk pengurutan secara **ascending (menaik)** :

SELECT * FROM pegawai ORDER BY Nama ASC;

```
mysql> select * from pegawai order by nama ASC;
+----+-----+-----+-----+
| NIP | Nama   | Alamat                | id_div |
+----+-----+-----+-----+
| P02 | Andri  | Griya Bumi Asri A.20  | LIT    |
| P01 | Arika  | Jl. Tanjung Sari No.5 | KEU    |
| P04 | Budi   | Jl. Cempaka No.7      | KEU    |
| P03 | Saskia | Jl. Karangan No.3     | LIT    |
| P05 | Wanda  | Jl. Merdeka No.16     | LOG    |
+----+-----+-----+-----+
5 rows in set (0.02 sec)
```

untuk pengurutan secara **DESC** atau **descending (menurun)**:

SELECT * FROM pegawai ORDER BY Nama DESC;

```
mysql> select * from pegawai order by nama DESC;
+----+-----+-----+-----+
| NIP | Nama   | Alamat                | id_div |
+----+-----+-----+-----+
| P05 | Wanda  | Jl. Merdeka No.16     | LOG    |
| P03 | Saskia | Jl. Karangan No.3     | LIT    |
| P04 | Budi   | Jl. Cempaka No.7      | KEU    |
| P01 | Arika  | Jl. Tanjung Sari No.5 | KEU    |
| P02 | Andri  | Griya Bumi Asri A.20  | LIT    |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```

f. Sintak **DISTINCT**

Distinct adalah kata kunci ini untuk menghilangkan duplikasi. Sebagai Contoh, pada tabel pegawai yang terdapat id_div dengan beberapa *record* yang sama. Kemudian ketikkan perintah berikut:

SELECT DISTINCT id_div FROM Pegawai;

```
mysql> select distinct id_div from pegawai;
+-----+
| id_div |
+-----+
| KEU    |
| LIT    |
| LOG    |
+-----+
3 rows in set (0.00 sec)
```

Dengan perintah di atas maka id_div yang sama hanya akan ditampilkan satu saja.

g. **UNION, INTERSECT dan EXCEPT**

UNION merupakan operator yang digunakan untuk **menggabungkan hasil query**, dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama:

SELECT id_div FROM pegawai WHERE id_div = 'KEU' UNION

SELECT id_div FROM divisi WHERE id_div = 'LOG';

```
mysql> select id_div from pegawai where id_div='KEU' OR id_div='LOG';
+-----+
| id_div |
+-----+
| KEU    |
| KEU    |
| LOG    |
+-----+
3 rows in set (0.00 sec)
```

Perintah di atas identik dengan:

SELECT id_div FROM divisi WHERE

id_div = 'KEU' OR id_div = 'LOG';

```
mysql> select id_div from divisi where id_div='KEU' OR id_div='LOG';
+-----+
| id_div |
+-----+
| KEU    |
| LOG    |
+-----+
2 rows in set (0.00 sec)
```

Namun tidak semua penggabungan dapat dilakukan dengan OR, yaitu jika bekerja pada dua tabel atau lebih.

INTERSECT merupakan operator yang digunakan untuk memperoleh data dari dua buah *query* dimana data yang ditampilkan adalah yang memenuhi kedua *query* tersebut dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

SELECT * FROM namatabel1 INTERSECT SELECT * FROM namatabel2;

Pada **MySQL** tidak terdapat operator **INTERSECT** namun sebagai gantinya dapat menggunakan **operator IN** seperti contoh 1 pada bagian *Nested Queries*.

EXCEPT / Set Difference merupakan operator yang digunakan untuk memperoleh data dari dua buah *query* dimana data yang ditampilkan adalah data yang ada pada hasil *query* 1 dan tidak terdapat pada data dari hasil *query* 2 dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

SELECT * FROM namatabel1 EXCEPT SELECT * FROM namatabel2;

Pada **MySQL** tidak terdapat operator **EXCEPT** namun sebagai gantinya dapat menggunakan **operator NOT IN** seperti contoh 2 pada bagian *Nested Queries*.

CONTOH :

1. Buatlah tabel matakuliah1 dan matakuliah2 seperti berikut :

```
mysql> create table matakuliah1
-> (kd_mk char(4) primary key,
-> Nama_Matakul varchar(20));
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> create table matakuliah2
-> (kd_mk char(4) primary key,
-> Nama_Matakul varchar(20));
Query OK, 0 rows affected (0.03 sec)
```

2. Isi tabel dengan data berikut :

```
mysql> select * from matakuliah1;
+-----+-----+
| kd_mk | Nama_Matakul |
+-----+-----+
| K101  | Fisika 1      |
| K102  | Bahasa Inggris 1 |
| K103  | Kalkulus      |
+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from matakuliah2;
+-----+-----+
| kd_mk | Nama_Matakul |
+-----+-----+
| K201  | Fisika 2      |
| K202  | Bahasa Inggris 2 |
| K203  | Kalkulus Lanjut |
+-----+-----+
3 rows in set (0.00 sec)
```

3. Gunakan perintah **UNION** antara tabel matakuliah1 dan matakuliah2.

```
mysql> select Nama_Matakul from matakuliah1 UNION
-> select Nama_Matakul from matakuliah2;
+-----+
| Nama_Matakul |
+-----+
| Fisika 1      |
| Bahasa Inggris 1 |
| Kalkulus      |
| Fisika 2      |
| Bahasa Inggris 2 |
| Kalkulus Lanjut |
+-----+
6 rows in set (0.00 sec)
```

4. Gunakan perintah **EXCEPT (NOT IN)** antara tabel matakuliah1 dan matakuliah2.

```
mysql> select * from matakuliah1 where Nama_Matakul NOT IN  
-> (select Nama_Matakul from matakuliah2);  
+-----+-----+  
| kd_mk | Nama_Matakul |  
+-----+-----+  
| K101  | Fisika 1     |  
| K102  | Bahasa Inggris 1 |  
| K103  | Kalkulus     |  
+-----+-----+  
3 rows in set (0.00 sec)
```

5. Gunakan perintah **INTERSECT (IN)** antara tabel matakuliah1 dan matakuliah2.

```
mysql> select * from matakuliah1 where Nama_Matakul IN  
-> (select Nama_Matakul from matakuliah2);  
Empty set (0.01 sec)
```