


# **MODUL PRAKTIKUM BASIS DATA**



**Oleh :**

**Amelia Yusnita, S.Kom.,M.Kom**

**PROGRAM STUDI SISTEM INFORMASI  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
WIDYA CIPTA DHARMA  
SAMARINDA**

	<b>STMIK WIDYA CIPTA DHARMA SAMARINDA</b>	<b>S1- SISTEM INFORMASI</b>
<b>Prak. BASIS DATA</b>	<b>Query Lanjutan</b>	<b>LABSHEET 03</b>
<b>Semester III</b>		<b>Dosen : Amelia Yusnita,S.Kom.,M.Kom Email : amelia@wicida.ac.id</b>

## 1. TUJUAN

Mahasiswa mampu dan memahami dalam pengoperasian perintah-perintah SQL

## 2. TEORI

Untuk mengambil dan menampilkan data dari database, kita perlu menggunakan perintah `SELECT`. Perintah ini merupakan perintah yang sangat penting dalam SQL. Dengan perintah `SELECT`, kita dapat mengambil baris-baris data dari satu tabel atau lebih

## 3. LANGKAH KERJA

### A. Operator BETWEEN dan NOT BETWEEN

Operator `BETWEEN` sangat berguna untuk menangani operasi “jangkauan”. Sebagai contoh praktekkan perintah SQL berikut ini:

```
SELECT Id_RkmMedis,Id_Pasien,Id_Dokter,Tgl_Periksa from rekammedis  
WHERE Tgl_Periksa BETWEEN '2022-01-02' AND '2022-01-10';
```

<b>Id_RkmMedis</b>	<b>Id_Pasien</b>	<b>Id_Dokter</b>	<b>Tgl_Periksa</b>
Rkm001	RS22L001	IDK004	2022-01-02
Rkm002	RS22P002	IDK005	2022-01-02
Rkm003	RS22L003	IDK004	2022-01-02
Rkm005	RS22L009	IDK005	2022-01-10
Rkm012	RS22L010	IDK004	2022-01-10

Gambar 1. Operator BETWEEN

Jika tidak menggunakan `BETWEEN`, perintah dapat ditulis sebagai berikut :

```
SELECT Id_RkmMedis,Id_Pasien,Id_Dokter,Tgl_Periksa from rekammedis  
WHERE Tgl_Periksa >='2022-01-02' AND Tgl_Periksa <='2022-01-10';
```

Jika ditambahkan `NOT` didepan `BETWEEN` , akan diperoleh keadaan yang sebaliknya. Sebagai contoh praktekkan dengan perintah SQL :

```
SELECT Id_RkmMedis,Id_Pasien,Id_Dokter,Tgl_Periksa from rekammedis
WHERE Tgl_Periksa NOT BETWEEN '2022-01-02' AND '2022-01-10';
```

Id_RkmMedis	Id_Pasien	Id_Dokter	Tgl_Periksa
Rkm004	RS22L008	IDK004	2022-03-05
Rkm006	RS22L010	IDK004	2022-01-01
Rkm007	RS22P002	IDK004	2022-02-12
Rkm008	RS22L003	IDK004	2022-03-05
Rkm009	RS22P004	IDK001	2022-02-12
Rkm010	RS22L008	IDK003	2022-03-17
Rkm011	RS22L009	IDK004	2022-01-01

Gambar 2. Operator NOT BETWEEN

## B. Operator IN dan NOT IN

Operator ini berguna untuk melakukan pencocokan dengan salah satu yang ada pada suatu daftar nilai. Sebagai contoh praktekkan dengan perintah SQL :

```
SELECT Id_Dokter,Nama_Dokter FROM dokter WHERE Id_Dokter ='IDK001'
OR Id_Dokter ='IDK003' OR Id_Dokter='IDK005' ;
```

Id_Dokter	Nama_Dokter
IDK001	Ahmad
IDK003	Fahysh Ahmad
IDK005	Salsabila

Gambar 3. Perintah Untuk Memperoleh Nama Yang Sesuai Dengan Nim

Perintah diatas jika ditulis dengan memakai IN akan tampil seperti :

```
SELECT Id_Dokter,Nama_Dokter FROM dokter WHERE Id_Dokter IN
('IDK001','IDK003','IDK005');
```

Id_Dokter	Nama_Dokter
IDK001	Ahmad
IDK003	Fahysh Ahmad
IDK005	Salsabila

Gambar 4. Operator IN

Apabila yang diinginkan adalah SELAIN, maka perlu ditambahkan NOT didepan IN.

```
SELECT Id_Dokter,Nama_Dokter FROM dokter WHERE Id_Dokter NOT IN
('IDK001','IDK003','IDK005');
```

Id_Dokter	Nama_Dokter
IDK002	Syafa Nada
IDK004	Almahyra

Gambar 5. Operator NOT IN

### C. Operator LIKE dan NOT LIKE

Operator LIKE sangat bermanfaat untuk mencari data, misalnya siapa saja yang Mahasiswa yang namanya mengandung huruf “a” atau dan lain-lain. Dalam melakukan pencarian dengan operator ini, perlu disebutkan tanda *wildcard* berupa garis bawah (\_) atau persen (%)

- Tanda garis bawah ( ) berarti karakter apa saja

Contoh

a\_i cocok dengan ani, ali, asi, ataupun abi, tetapi tidak cocok dengan abri ataupun andi

- Tanda persen (%) berarti cocok dengan karakter apa saja dan berapun panjangnya

Contoh

%a% : cocok dengan apa saja yang mengandung karakter a atau A

%a : cocok dengan yang berkaitan a atau A hanya berlaku untuk yang bertipe VARCHAR

a% : cocok dengan yang berawalan a atau A

Sebagai contoh praktekkan dengan perintah SQL :

- Tampilkan semua Pasien yang namanya mengandung “au“

```
SELECT Id_Pasien, Nama_Pasien FROM pasien2 WHERE Nama_Pasien LIKE '%au%';
```

Id_Pasien	Nama_Pasien
RS22L003	Firdaus
RS22L008	Maulana Karim

Gambar 6. Operator LIKE Menggunakan Karakter %a%

- Tampilkan semua Pasien yang namanya berawalan “F“

```
SELECT Id_Pasien, Nama_Pasien, Jenis_Kelamin FROM pasien WHERE Nama_Pasien LIKE 'f%';
```

Id_Pasien	Nama_Pasien	Jenis_Kelamin
RS22L003	Firdaus	L
RS22P002	Fatimah	P

Gambar 7. Operator LIKE Menggunakan Karakter a%

3. Tampilkan semua Pasien yang namanya tidak berawalan “F”

```
SELECT Id_Pasien, Nama_Pasien, Jenis_Kelamin FROM pasien WHERE Nama_Pasien NOT LIKE 'f%';
```

Id_Pasien	Nama_Pasien	Jenis_Kelamin
RS22L001	Nursalam	L
RS22L005	Putra	L
RS22L006	Bayu	L
RS22L007	Ammar Zen	L
RS22L008	Maulana Karim	L
RS22L009	Rhomeo	L
RS22L010	Kevin	L
RS22P004	Nabila	P

Gambar 8. Operator NOT LIKE Menggunakan Karakter a%

#### D. Operator REGEXP

Selain LIKE. MySQL menyediakan operator REGEXP (berasal dari “regular expression”) untuk kepentingan pencocokan string. Simbol-simbol yang digunakan pada REGEXP serupa dengan regular expression yang digunakan pada utilitas `grep` yang terdapat pada system operasi UNIX dan Linux. Sebagai contoh praktekkan dengan perintah SQL :

1. Tampilkan semua Obat yang namanya berawalan “V”

```
SELECT * FROM obat WHERE Nama_Obat REGEXP '^v';
```

Id_Obat	Nama_Obat	Ket_Obat
IDO004	Vicks	Balm
IDO009	Vitamin C	Tablet
IDO010	Vitamin D	Tablet

Gambar 9. Operator REGEXP Menggunakan Simbol ‘^’

2. Tampilkan semua Obat yang namanya berakhiran “in”

```
SELECT * FROM obat WHERE Nama_Obat REGEXP 'in$';
```

Id_Obat	Nama_Obat	Ket_Obat
IDO002	Aspirin	Tablet
IDO007	Claritin	Tablet

Gambar 10. Operator REGEXP Menggunakan Simbol ‘\$’

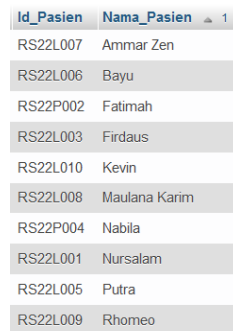
#### E. Pengurutan Data

Hasil query dapat diurutkan menurut kriteria yang dikehendaki, dengan menambahkan klausa `ORDER BY`. Pengurutan dapat dilakukan baik secara *ascending* (urut naik) maupun

*descending* (urut turun). Pengurutan hasil query didasarkan satu kolom ataupun beberapa kolom. Sebagai contoh praktekkan dengan perintah SQL :

### 1. Mengurutkan Satu Kolom

```
SELECT Id_Pasien, Nama_Pasien FROM pasien ORDER BY Nama_Pasien;
```



Id_Pasien	Nama_Pasien
RS22L007	Ammar Zen
RS22L006	Bayu
RS22P002	Fatimah
RS22L003	Firdaus
RS22L010	Kevin
RS22L008	Maulana Karim
RS22P004	Nabila
RS22L001	Nursalam
RS22L005	Putra
RS22L009	Rhomeo

Gambar 11. Perintah Mengurutkan Satu Kolom

### 2. Pengurutan Secara Descending

```
SELECT Id_Pasien, Nama_Pasien FROM pasien ORDER BY Nama_Pasien  
DESC;
```

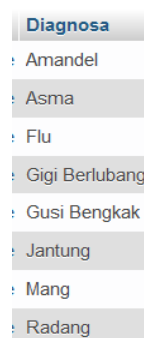


Id_Pasien	Nama_Pasien
RS22L009	Rhomeo
RS22L005	Putra
RS22L001	Nursalam
RS22P004	Nabila
RS22L008	Maulana Karim
RS22L010	Kevin
RS22L003	Firdaus
RS22P002	Fatimah
RS22L006	Bayu
RS22L007	Ammar Zen

Gambar 12. Perintah Mengurutkan Secara DESCENDING

## F. Pengelompokan Data

Hasil dari SELECT dapat dikelompokkan, Sebagai contoh praktekkan dengan perintah SQL : `SELECT Diagnosa FROM rekammedis GROUP BY Diagnosa;`



Diagnosa
Amandel
Asma
Flu
Gigi Bertubang
Gusi Bengkak
Jantung
Mang
Radang

Gambar 13. Pengelompokan Data Menggunakan GROUP BY

## G. Klausula HAVING

Pemakaian klausula HAVING terkait dengan klausula GROUP BY. Kegunaannya adalah untuk menentukan kondisi bagi GROUP BY. Kelompok yang memenuhi HAVING saja yang akan dihasilkan. Sebagai contoh praktekkan dengan perintah SQL :

```
SELECT Diagnosa FROM rekammedis WHERE Diagnosa IS NOT NULL GROUP BY Diagnosa HAVING Diagnosa <> 'Radang';
```



Diagnosa
Amandel
Asma
Flu
Gigi Berlubang
Gusi Bengkak
Jantung
Mang

Gambar 14. Pembatasan Baris Pada GROUP BY Dengan HAVING

Diagnosa “Radang” tidak tampil, disebabkan dalam HAVING disebutkan bahwa kondisinya adalah untuk diagnosa selain “Radang”.

Perintah lain dapat dipraktekkan dengan contoh sebagai berikut :

```
SELECT Diagnosa FROM rekammedis WHERE Diagnosa IS NOT NULL AND Diagnosa <>'radang' GROUP BY Diagnosa;
```

## 4. LATIHAN