



PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
Sistem Basis Data



Data Manipulation Language – Menggunakan Operator

TEKNIK INFORMATIKA S1
UNIVERSITAS DIAN NUSWANTORO

Capaian Pembelajaran

Mahasiswa mampu:

- Menggunakan operator BETWEEN, LIKE, dan IN
- Menerapkan perintah ORDER BY untuk mengurutkan data
- Menerapkan perintah GROUP BY untuk mengelompokkan data

Pokok Bahasan

BETWEEN

LIKE

IN

ORDER BY

GROUP BY



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

BETWEEN

Operator BETWEEN

- SQL menyertakan operator perbandingan BETWEEN untuk menyederhanakan klausa WHERE yang menentukan jangkauan nilai kurang dari atau sama dengan dan lebih dari.
- Bentuk umum Penulisan SQL:

```
SELECT <attribute-list>  
FROM <table>  
WHERE <condition> BETWEEN <range1> AND <range2>;
```

Contoh - Menampilkan Gaji Staf antara 3000000 -8000000

- Untuk menampilkan semua informasi pada tabel menggunakan perintah SELECT diikuti dengan klausa WHERE dan operator BETWEEN
- Nama table: staf

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|------------|-----|------------|----------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| a02 | devi | asisten | L | 2022-09-24 | 3500000 | B002 |
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |

- Penulisan SQL:

```
SELECT * FROM staf WHERE gaji  
BETWEEN 3000000 AND 8000000;
```

Hasil - Menampilkan Gaji Staf antara 3000000 -8000000

```
MariaDB [db_usaha]> SELECT * FROM staf WHERE gaji BETWEEN 3000000 AND 8000000;
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|------------|-----|------------|---------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| a02 | devi | asisten | L | 2022-09-24 | 3500000 | B002 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |

Operator NOT BETWEEN

- Operator NOT BETWEEN merupakan negasi dari operator BETWEEN
- Jika operator NOT BETWEEN digunakan, maka akan menghasilkan diluar range yang ditentukan

```
MariaDB [db_usaha]> SELECT * FROM staf WHERE gaji NOT BETWEEN 3000000 AND 8000000;
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|------|---------|-----|------------|----------|-----------|
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |

- Pada contoh diatas, menampilkan hasil diluar range 3.000.000 dan 8.000.000



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

LIKE

Operator LIKE

- Pencocokan pola pada string dapat dilakukan dengan operator LIKE
- Penggunaan operator LIKE diikuti dengan 2(dua) special karakter:
 - Percent (%): karakter persen mewakili beberapa karakter
 - Underscore (_): karakter underscore mewakili sebuah karakter

- Bentuk umum penulisan SQL:

```
SELECT <attribute-list>  
FROM <table>  
WHERE <condition> LIKE 'string';
```

- String dapat berupa: 'A%', '%A%', '%semarang%', 'A__'

Contoh – Penggunaan Operator LIKE

- Menampilkan data staf yang nipnya diawali dengan huruf 'a'
- Menampilkan data staf yang nipnya diakhiri dengan huruf '1'
- Menampilkan data staf yang nipnya terdiri dari 3 karakter dan diawali dengan huruf 'a'

```
MariaDB [db_usaha]> SELECT * FROM staf WHERE nip LIKE 'a%';
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|---------|-----|------------|---------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| a02 | devi | asisten | L | 2022-09-24 | 3500000 | B002 |

```
MariaDB [db_usaha]> SELECT * FROM staf WHERE nip LIKE '%1';
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|------------|-----|------------|----------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |

```
MariaDB [db_usaha]> SELECT * FROM staf WHERE nip LIKE 'a__';
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|---------|-----|------------|---------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| a02 | devi | asisten | L | 2022-09-24 | 3500000 | B002 |

Contoh – Penggunaan Operator LIKE/NOT LIKE

- Menampilkan data staf posisi terdapat kata 'asist'
- Operator **NOT LIKE** digunakan untuk menampilkan string selain string yang dimaksud
- Pada contoh berikut, menampilkan data staf dengan nip yang diawali selain huruf 'a'

```
MariaDB [db_usaha]> SELECT * FROM staf WHERE posisi LIKE '%asist%';
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|---------|-----|------------|---------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| a02 | devi | asisten | L | 2022-09-24 | 3500000 | B002 |

```
MariaDB [db_usaha]> SELECT * FROM staf WHERE nip NOT LIKE 'a%';
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|------|------------|-----|------------|----------|-----------|
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

IN

Operator IN

- SQL memungkinkan pengujian terhadap tuple untuk keanggotaan dalam suatu relasi menggunakan operator IN
- Uji keanggotaan suatu relasi menguji apakah nilai data cocok dengan salah satu daftar nilai yang diberikan
- Negasi dari operator IN adalah NOT IN, yaitu untuk memeriksa nilai data tidak termasuk dalam daftar nilai yang diberikan
- Operator IN/NOT IN digunakan dalam klausa SELECT

```
SELECT <attribute-list>  
FROM <table>  
WHERE <condition> IN (set-membership);
```

Contoh – Penggunaan Operator IN

- Menguji keanggotaan data table staf dengan posisi **asisten** atau **manajer**.

```
SELECT * FROM staf  
WHERE posisi IN ('asisten', 'manajer');
```

```
MariaDB [db_usaha]> SELECT * FROM staf  
-> WHERE posisi IN ('asisten', 'manajer');
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|---------|-----|------------|----------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| a02 | devi | asisten | L | 2022-09-24 | 3500000 | B002 |
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |

- Berdasarkan kasus diatas, maka hasil yang ditampilkan adalah data yang posisinya cocok dengan posisi asisten atau manajer

Contoh – Penggunaan Operator NOT IN

```
MariaDB [db_usaha]> SELECT * FROM staf  
-> WHERE posisi NOT IN ('asisten', 'manajer');  
+-----+-----+-----+-----+-----+-----+-----+  
| nip | nama | posisi      | sex | tgl_masuk | gaji   | id_cabang |  
+-----+-----+-----+-----+-----+-----+-----+  
| s01 | sari | supervisor  | L   | 2000-02-05 | 7500000 | B001      |  
+-----+-----+-----+-----+-----+-----+-----+
```

- Pada contoh kasus diatas, dapat disimpulkan bahwa penggunaan operator NOT IN menampilkan hasil query **selain** posisi asisten dan manajer.
- Penulisan SQL:

```
SELECT * FROM staf  
WHERE posisi NOT IN ('asisten', 'manajer');
```



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

ORDER BY

Pengurutan Data – Klausa ORDER BY

- Secara umum baris (row) pada tabel SQL (hasil Query) tidak menggunakan urutan tertentu (dalam beberapa DBMS, pengurutan hasil query, secara default didasarkan pada primary key)
- Kita dapat menentukan urutan hasil Query menggunakan klausa ORDER BY pada SELECT.
- Klausa ORDER BY terdiri dari pengidentifikasi kolom yang hasilnya akan diurutkan baik secara ascending (ASC) maupun descending (DESC).
- Klausa ORDER BY harus diletakkan di posisi akhir pada penggunaan SELECT (Connolly,2015)

Penulisan SQL – ORDER BY

Tanpa ASC/DESC order

```
SELECT <attribute-list>  
FROM <table>  
WHERE <condition>  
ORDER BY <column-identifier> ;
```

Dengan ASC/DESC order

```
SELECT <attribute-list>  
FROM <table>  
WHERE <condition>  
ORDER BY <column-identifier> DESC;
```

- **Column-identifier** dapat berupa kolom nama (kata) atau kolom angka
- **Column-identifier** dapat terdiri lebih dari 1 atribut (*multiple attributes*)
- Secara default, ORDER BY diurutkan dalam Ascending. Untuk menentukan model pengurutan dapat menambahkan DESC untuk descending setelah penulisan **column identifier**

Contoh – ORDER BY (Pengurutan Satu Kolom)

- Menampilkan semua data staf diurutkan berdasarkan gaji
- Maka hasil tabel akan menampilkan urutan secara ascending berdasarkan gaji yaitu dimulai dengan gaji terkecil sampai terbesar

```
SELECT * FROM staf ORDER BY gaji ;
```

```
MariaDB [db_usaha]> SELECT * FROM staf ORDER BY gaji;
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|------------|-----|------------|----------|-----------|
| a02 | devi | asisten | P | 2022-09-24 | 3500000 | B002 |
| s02 | leni | supervisor | P | 2022-10-04 | 4000000 | B002 |
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |

Contoh – ORDER BY (Pengurutan Satu Kolom) DESC

- Menampilkan semua data staf diurutkan berdasarkan gaji
- Maka hasil tabel akan menampilkan urutan secara descending berdasarkan gaji yaitu dimulai dengan gaji terbesar ke terkecil

```
SELECT * FROM staf ORDER BY gaji ;
```

```
MariaDB [db_usaha]> SELECT * FROM staf ORDER BY gaji DESC;
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|------------|-----|------------|----------|-----------|
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| s02 | leni | supervisor | P | 2022-10-04 | 4000000 | B002 |
| a02 | devi | asisten | P | 2022-09-24 | 3500000 | B002 |

Contoh – ORDER BY (Pengurutan Dua Kolom)

- Menampilkan semua data staf dimana diurutkan berdasarkan posisi dan gaji
- Hasil tabel menampilkan menampilkan urutan pertama dari posisi diurutkan secara ascending kemudian diikuti oleh gaji yang disusun terurut dari gaji terkecil ke terbesar disesuaikan dengan posisi

```
SELECT * FROM staf ORDER BY posisi,gaji;
```

```
MariaDB [db_usaha]> SELECT * FROM staf ORDER BY posisi,gaji;
```

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|------------|-----|------------|----------|-----------|
| a02 | devi | asisten | P | 2022-09-24 | 3500000 | B002 |
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |
| s02 | leni | supervisor | P | 2022-10-04 | 4000000 | B002 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO**

GROUP BY

Pengelompokan Data – Klausa GROUP BY

- Pengelompokan data dapat menggunakan klausa GROUP BY
- Ringkasan suatu tabel dapat dilakukan dengan klausa GROUP BY pada perintah SELECT.
- Kita dapat meringkas banyaknya baris tabel menjadi satu ringkasan seperti:
 - Posisi apa saja dalam tabel staf
 - Jumlah gaji dari masing – masing kantor cabang

- Penulisan SQL:

```
SELECT <attribute-list>  
FROM <table>  
WHERE <condition>  
GROUP BY <column-identifier> ;
```


Contoh – Pengelompokan Data

- Mengelompokkan posisi staf dan menghitung jumlah posisi staf.
- Pada contoh disamping, tabel staf terdapat 3 posisi yang diringkas dari 6 baris tabel
- Klausa GROUP BY lebih tepat diimplementasikan dengan fungsi agregasi seperti: **SUM, AVG, COUNT, MIN, MAX** sebagai detail nilai dari baris yang dikelompokkan. (dibahas pada pertemuan 7)

| nip | nama | posisi | sex | tgl_masuk | gaji | id_cabang |
|-----|-------|------------|-----|------------|----------|-----------|
| a01 | wahyu | asisten | L | 2014-08-10 | 5000000 | B001 |
| a02 | devi | asisten | P | 2022-09-24 | 3500000 | B002 |
| m01 | budi | manajer | L | 1996-01-01 | 10000000 | B001 |
| s01 | sari | supervisor | L | 2000-02-05 | 7500000 | B001 |
| s02 | leni | supervisor | P | 2022-10-04 | 4000000 | B002 |

```
MariaDB [db_usaha]> SELECT
-> posisi,
-> COUNT(posisi) as jumlah_posisi
-> FROM staf
-> GROUP BY posisi;
```

| posisi | jumlah_posisi |
|------------|---------------|
| asisten | 2 |
| manajer | 1 |
| supervisor | 2 |

Referensi

UTAMA

1. Silberschatz, A., Korth, H. F. & Sudarshan, S., 2022. Database System Concepts. 7th ed. New York: McGraw-Hill Education
2. Connolly, T. & Begg, C., 2015. Database Systems A practical Approach to Design, Implementation, and Management. Sixth Edition ed. s.l.:Pearson.
3. Elmasri, R. & Navathe, S. B., 2016. Fundamentals of Database Systems. 7th ed. s.l.:Pearson

PENDUKUNG

Aripin., 2005. *Praktikum Basis Data Dengan Database Server MySQL*. Semarang: Fakultas Ilmu Komputer



TERIMA KASIH

ANY QUESTIONS?