



MODUL

Praktikum Konsep Basis Data TAHUN AJARAN 2025 -2026

S1 Informatika
Fakultas Ilmu Komputer



PRAKTIKUM IX
Pattern Matching (Like), Order By, Group By, Having

1. Tujuan Pembelajaran

Setelah praktikum ini, mahasiswa diharapkan mampu:

- a. Menggunakan klausa LIKE untuk melakukan pencarian pola pada teks.
- b. Mengurutkan data dengan ORDER BY menggunakan satu atau lebih kolom.
- c. Mengelompokkan data dengan GROUP BY dan menggunakan fungsi agregat.
- d. Menyaring hasil pengelompokan dengan HAVING.

2. Pattern Matching (Like)

Fungsi string digunakan untuk menampilkan data yang di dasarkan pada pencarian dengan karakter. Pada pencarian data digunakan syntax LIKE, pada dasarnya syntax LIKE hampir sama dengan syntax = .

Bedanya kalau syntax = , maka pencarian karakter harus sesuai dengan kata yang kita buat tetapi dengan menggunakan LIKE karakter yang akan kita tampilkan tidak harus lengkap hanya dengan menuliskan salah satu huruf atau kata saja, maka semua data yang akan kita cari akan ditampilkan.

Klausa LIKE digunakan untuk mencari data dengan pola tertentu di dalam kolom teks. Operator ini biasanya digunakan bersama wildcard:

Simbol	Arti	Contoh
%	cocok dengan 0 atau lebih karakter	'A%' = semua teks yang dimulai huruf A
_	cocok dengan 1 karakter	'_n%' = huruf kedua adalah n
\	digunakan untuk escape karakter khusus seperti _ atau %	'%_%' mencari teks yang mengandung karakter underscore



Bentuk umumnya :

```
SELECT * FROM nama_tabel WHERE nama_kolom LIKE 'char%';
```

```
SELECT * FROM nama_tabel WHERE nama_kolom LIKE '%char';
```

```
SELECT * FROM nama_tabel WHERE nama_kolom LIKE '%char%';
```

```
SELECT * FROM nama_tabel WHERE nama_kolom NOT LIKE '%char%';
```

```
SELECT * FROM nama_tabel WHERE nama_kolom LIKE '_';
```

Untuk membandingkan suatu nilai yang mirip dengan nilai tertentu menggunakan operator wildcards berupa persen (%) atau underscore (_).

Persen (%) merepresentasikan 0,1, atau banyak karakter/angka sedangkan underscore(_) merepresentasikan satu karakter/angka.

Statement	Deskripsi
Like '200%'	Cari semua nilai yang diawali dengan 200
Like '%200%'	Cari semua nilai yang dimana saja ada 200
Like '_00%'	Cari semua nilai yang memiliki 00 di posisi 2 dan 3 (posisi 1 apa saja → hanya satu)
Like '2_%_%'	Cari semua nilai yang diawali dengan 2, dengan Panjang minimal 3 karakter
Like '%2'	Cari semua nilai yang diakhiri dengan 2
LIKE '_2%3'	Cari semua nilai yang memiliki angka 2 di posisi kedua dan diakhiri dengan 3
Like '2___3'	Cari semua nilai yang diawali dengan 2 dan diakhiri dengan 3 dengan total 5 karakter

Contoh 1 — Nama diawali huruf "A"

```
SELECT * FROM pelanggan
WHERE nama LIKE 'A%';
```



Contoh 2 — Kota mengandung huruf “o”

```
SELECT id, nama, kota FROM pelanggan
WHERE kota LIKE '%o%';
```

Contoh 3 — Email mengandung tanda underscore _

```
SELECT id, nama, email FROM pelanggan
WHERE email LIKE '%\_%';
```

3. ORDER BY

Digunakan untuk **mengurutkan hasil query** berdasarkan satu atau lebih kolom. Secara default, MySQL menggunakan urutan **ASC (Ascending)**. Untuk urutan menurun gunakan **DESC (Descending)**.

Sintaks:

```
SELECT kolom1, kolom2 FROM tabel
ORDER BY kolom1 [ASC|DESC], kolom2 [ASC|DESC];
```

Contoh 1 — Urutkan pelanggan berdasarkan kota (A–Z), lalu nama (Z–A)

```
SELECT nama, kota FROM pelanggan ORDER BY kota ASC, nama DESC;
```

Contoh 3 — Urutkan transaksi berdasarkan total pembelian terbesar

```
SELECT id, pelanggan_id, kategori, total FROM transaksi
ORDER BY total DESC;
```



4. GROUP BY

GROUP BY digunakan untuk **mengelompokkan baris data** berdasarkan kolom tertentu agar dapat dihitung menggunakan fungsi agregat seperti:

Fungsi	Kegunaan
COUNT()	Menghitung jumlah baris
SUM()	Menjumlahkan nilai numerik
AVG()	Menghitung rata-rata
MIN()	Nilai minimum
MAX()	Nilai maksimum

Contoh:

Data transaksi dikelompokkan berdasarkan kategori, lalu dihitung total penjualan (SUM(total)) tiap kategori.

```
SELECT kategori, SUM(total) AS omzet
FROM transaksi
GROUP BY kategori;
```

5. HAVING

HAVING digunakan untuk menyaring hasil dari GROUP BY.

Bedanya dengan WHERE:

- WHERE → menyaring **data mentah sebelum agregasi**
- HAVING → menyaring **hasil agregasi setelah GROUP BY**

Contoh:

Menampilkan kategori yang memiliki total penjualan lebih dari 2 juta.

```
SELECT kategori, SUM(total) AS total_penjualan
FROM transaksi
GROUP BY kategori
HAVING SUM(total) > 2000000;
```



Contoh menggunakan WHERE dan HAVING:

```
SELECT kategori, SUM(total) AS total_kategori
FROM transaksi
WHERE tanggal >= '2025-09-01'
GROUP BY kategori
HAVING SUM(total) > 3000000;
```

Contoh gabungan WHERE + GROUP BY + HAVING + ORDER BY:

```
SELECT p.kota, COUNT(t.id) AS jumlah_transaksi
FROM pelanggan p
JOIN transaksi t ON t.pelanggan_id = p.id
GROUP BY p.kota
HAVING COUNT(t.id) >= 2
ORDER BY jumlah_transaksi DESC;
```



LATIHAN PRAKTIKUM 9

1. Buatlah database baru bernama db_penjualan!
2. Buatlah tabel berikut!

Tabel Produk

```
CREATE TABLE produk (  
  id_produk INT AUTO_INCREMENT PRIMARY KEY,  
  nama_produk VARCHAR(100),  
  kategori VARCHAR(50),  
  harga DECIMAL(10,2)  
);
```

Tabel Transaksi

```
CREATE TABLE transaksi (  
  id_transaksi INT AUTO_INCREMENT PRIMARY KEY,  
  id_produk INT,  
  tanggal DATE,  
  jumlah INT,  
  total DECIMAL(12,2),  
  pelanggan VARCHAR(100),  
  kota VARCHAR(50),  
  FOREIGN KEY (id_produk) REFERENCES produk(id_produk)  
);
```

3. Isilah tabel dengan data berikut!

Tabel Produk

id_produk	nama_produk	kategori	harga (Rp)
1	Laptop Asus X441	Elektronik	7.000.000
2	Headset JBL T110	Aksesoris	250.000
3	Kemeja Batik Pria	Fashion	180.000
4	Sepatu Sneakers	Fashion	350.000
5	Powerbank Xiaomi 10000mAh	Elektronik	300.000
6	Flashdisk Sandisk 32GB	Aksesoris	120.000
7	Smartphone Vivo Y20	Elektronik	2.500.000
8	Tas Kulit Wanita	Fashion	400.000



Tabel Transaksi

id_transaksi	id_produk	tanggal	jumlah	total (Rp)	pelanggan	kota
1	1	2025-09-01	1	7.000.000	Andi Pratama	Depok
2	2	2025-09-02	2	500.000	Budi Santoso	Bandung
3	3	2025-09-03	3	540.000	Citra Anindya	Jakarta
4	4	2025-09-04	1	350.000	Dewi Kurnia	Bogor
5	5	2025-09-05	1	300.000	Andi Pratama	Depok
6	6	2025-09-06	2	240.000	Eka Putra	Bandung
7	7	2025-09-07	1	2.500.000	Fani Lestari	Depok
8	8	2025-09-08	1	400.000	Gilang Saputra	Jakarta
9	1	2025-09-09	1	7.000.000	Budi Santoso	Bandung
10	3	2025-09-10	2	360.000	Hani Oktaviani	Bogor

4. Tampilkan pelanggan yang kota-nya mengandung huruf 'a'!
5. Tampilkan pelanggan yang namanya dimulai dengan 'Budi'!
6. Tampilkan transaksi berdasarkan **tanggal terbaru!**
7. Tampilkan **rata-rata total transaksi per pelanggan!**
8. Tampilkan kategori produk dengan **total penjualan lebih dari 2 juta!**
9. Tampilkan kota yang memiliki **lebih dari 2 transaksi!**
10. Tampilkan pelanggan dengan **rata-rata pembelian di atas 500.000!**
11. Siapa pelanggan dengan **total transaksi terbesar?**
12. Kota mana yang **paling banyak** melakukan transaksi?
13. Kategori apa yang memiliki **rata-rata harga tertinggi?**
14. Tampilkan semua pelanggan dari kota "Depok" yang **total belanjanya di atas 3 juta.**
15. Tampilkan produk yang paling sering dibeli berdasarkan jumlah transaksi.

