

# MODUL PRAKTIKUM BASIS DATA

## Pertemuan 6 : Query Standar (SELECT) pada PostgreSQL

### 1. Capaian Pembelajaran Mata Kuliah (CPMK)

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Memahami konsep dan fungsi perintah SELECT dalam basis data
2. Menggunakan SELECT untuk menampilkan data dari tabel
3. Menerapkan alias kolom (AS) untuk memperjelas hasil query
4. Menggunakan klausa WHERE untuk melakukan filtering data
5. Menggunakan pencarian NULL dan string (LIKE, ILIKE)
6. Mengurutkan data menggunakan ORDER BY
7. Membatasi hasil query dengan LIMIT dan OFFSET
8. Mengkombinasikan berbagai klausa SELECT untuk kebutuhan pengolahan data

### 2. Tujuan Praktikum

Praktikum ini bertujuan agar mahasiswa:

- Mampu mengambil data dari tabel menggunakan SELECT
- Mampu memodifikasi tampilan hasil query
- Mampu menyaring data sesuai kondisi tertentu
- Mampu melakukan pencarian data teks dan NULL
- Mampu mengurutkan dan membatasi jumlah data yang ditampilkan

### 3. Dasar Teori

#### 3.1 Pengertian SELECT

SELECT adalah perintah dalam SQL yang digunakan untuk mengambil atau menampilkan data dari satu atau lebih tabel dalam basis data.

### 3.2 Struktur Umum SELECT

```
SELECT [ALL | DISTINCT]
kolom1, kolom2, ...

FROM nama_tabel
WHERE kondisi
ORDER BY kolom ASC | DESC
LIMIT jumlah
OFFSET jumlah;
```

#### Penjelasan:

- **SELECT** → menentukan kolom yang ingin ditampilkan
- **FROM** → menentukan tabel sumber data
- **WHERE** → menyaring data berdasarkan kondisi
- **ORDER BY** → mengurutkan data
- **LIMIT** → membatasi jumlah data
- **OFFSET** → melewati sejumlah data

### 3.3 SELECT Dasar

#### Menampilkan semua kolom:

```
SELECT * FROM nama_tabel;
```

#### Menampilkan kolom tertentu:

```
SELECT kolom FROM nama_tabel;

SELECT kolom1, kolom2 FROM nama_tabel;
```

### 3.4 Column Alias (AS)

Digunakan untuk mengganti nama kolom pada hasil query.

```
SELECT kolom AS "nama lain kolom" FROM nama_tabel;

SELECT kolom1 AS " nama lain kolom1 ",
       kolom2 AS " nama lain kolom2"
FROM nama_tabel;
```

#### Fungsi:

- Mempermudah pembacaan hasil
- Cocok untuk laporan

### 3.5 Klausula WHERE (Filtering Data)

Digunakan untuk menampilkan data sesuai kondisi tertentu.

```
SELECT * FROM nama_tabel  
WHERE kolom = 'data_kolom';
```

#### Operator perbandingan:

- = sama dengan
- <> atau != tidak sama dengan
- > lebih besar
- < lebih kecil
- >= lebih besar sama dengan
- <= lebih kecil sama dengan

#### Kombinasi kondisi:

```
SELECT * FROM nama_tabel  
WHERE kolom1 = 'Rina Marsudi'  
AND kolom2 = 'Yogyakarta';
```

```
SELECT * FROM nama_tabel  
WHERE kolom1 = 'Ali Topan'  
OR kolom2 = 2;
```

### 3.6 Pencarian NULL

NULL adalah nilai kosong (tidak ada data).

#### Mencari NULL:

```
SELECT * FROM nama_tabel  
WHERE kolom IS NULL;
```

#### Mencari yang tidak NULL:

```
SELECT * FROM nama_tabel  
WHERE kolom IS NOT NULL;
```

#### Catatan:

NULL tidak bisa dibandingkan dengan operator =

### 3.7 Pencarian String (LIKE dan ILIKE)

**LIKE (case sensitive): untuk mencari string tertentu**

```
SELECT * FROM nasabah  
WHERE nama_nasabah LIKE 'Ali Topan';
```

**Menggunakan wildcard %:**

```
SELECT * FROM nasabah  
WHERE alamat_nasabah LIKE '%negara%';
```

**Menggunakan \_:**

```
SELECT * FROM nasabah  
WHERE nama_nasabah LIKE 'Ali T_p_n';
```

**ILIKE (tidak sensitif huruf besar/kecil): untuk pencarian yang case insensitive (tidak mempedulikan huruf besar atau kecil)**

```
SELECT * FROM nasabah  
WHERE nama_nasabah ILIKE '%marsudi';
```

**Penjelasan:**

- % = banyak karakter
- \_ = satu karakter

### 3.8 ORDER BY (Pengurutan Data)

**Ascending (default): Untuk mengurutkan dari kecil ke besar:**

```
SELECT * FROM nasabah  
ORDER BY nama_nasabah;
```

**Descending: Untuk mengurutkan dari besar ke kecil**

```
SELECT * FROM nasabah  
ORDER BY nama_nasabah DESC;
```

**Lebih dari satu kolom: pisahkan dengan tanda koma:**

```
SELECT * FROM nasabah_has_rekening  
ORDER BY no_rekening, id_nasabah DESC;
```

### 3.9 LIMIT dan OFFSET

**LIMIT:**

Digunakan untuk membatasi jumlah baris yang ditampilkan dalam SELECT.

```
SELECT * FROM nasabah
ORDER BY id_nasabah
LIMIT 3; // Hanya menampilkan 3 baris pertama
```

**OFFSET:**

Melewati sejumlah data

```
SELECT * FROM nasabah
ORDER BY id_nasabah
LIMIT 2 OFFSET 2; // Menampilkan 2 baris setelah melewati 2 baris pertama
```

**Tips:**

Gunakan LIMIT bersama ORDER BY agar urutan yang ditampilkan akan selalu konsisten

## 4. Percobaan Praktikum

### 4.1 Membuat Tabel

Tabel: nasabah

id_nasabah	nama_nasabah	alamat_nasabah
1	Ali Topan	Jakarta
2	Rina Marsudi	Yogyakarta
3	Budi Santoso	Surabaya
4	Siti Aminah	Bandung
5	Dewi Lestari	NULL

```
CREATE TABLE nasabah (
  id_nasabah SERIAL PRIMARY KEY,
  nama_nasabah VARCHAR(100),
  alamat_nasabah TEXT
);
```

### 4.2 Menambahkan Data

```
INSERT INTO nasabah (nama_nasabah, alamat_nasabah) VALUES
('Ali Topan', 'Jakarta'),
```

```
('Rina Marsudi', 'Yogyakarta'),  
( 'Budi Santoso', 'Surabaya'),  
( 'Siti Aminah', 'Bandung'),  
( 'Dewi Lestari', NULL);
```

### 4.3 Percobaan Query

Jalankan dan amati hasilnya:

1. Menampilkan semua data

```
SELECT * FROM nasabah;
```

2. Menggunakan alias

```
SELECT nama_nasabah AS "Nama" FROM nasabah;
```

3. Menggunakan WHERE

```
SELECT * FROM nasabah WHERE alamat_nasabah = 'Surabaya';
```

4. Mencari NULL

```
SELECT * FROM nasabah WHERE alamat_nasabah IS NULL;
```

5. Menggunakan LIKE

```
SELECT * FROM nasabah WHERE nama_nasabah LIKE '%Siti%';
```

6. Mengurutkan data

```
SELECT * FROM nasabah ORDER BY nama_nasabah DESC;
```

7. Menggunakan LIMIT & OFFSET

```
SELECT * FROM nasabah ORDER BY id_nasabah LIMIT 2 OFFSET  
1;
```

## 5. Latihan Praktikum

### A. Latihan 1

1. Tampilkan semua data dari tabel nasabah
2. Tampilkan hanya kolom nama\_nasabah
3. Tampilkan data nasabah yang berasal dari Bandung
4. Tampilkan data nasabah dengan id\_nasabah lebih dari 2
5. Tampilkan data dengan nama yang mengandung "Dewi"

6. Urutkan data berdasarkan nama secara ascending
7. Gunakan alias untuk kolom nama dan alamat
8. Tampilkan data dengan kondisi alamat bukan NULL urutkan berdasarkan nama
9. Tampilkan data dengan nama berawalan "A"
10. Tampilkan nasabah yang namanya mengandung kata "Siti"
11. Tampilkan nasabah yang alamatnya mengandung kata "arta"
12. Urutkan berdasarkan nama secara descending
13. Tampilkan data diurutkan berdasarkan nama dan id\_nasabah
14. Tampilkan 3 data pertama
15. Tampilkan data dari baris ke-2 sebanyak 2 data
16. Tampilkan 1 data mulai dari data ke-3
17. Tampilkan data dengan nama mengandung huruf "i" dan alamat tidak NULL
18. Tampilkan data nasabah selain dari Jakarta dan Bandung