



Hari 2 (Selasa) – Dasar SQL: CREATE, INSERT, SELECT



Tujuan Pembelajaran

- Memahami konsep dasar query SQL untuk membuat dan mengelola data
 - Mengetahui tipe data umum di PostgreSQL
 - Mampu membuat dan membaca data user melalui query SQL
-

1. Pengantar SQL & Relasional

SQL (Structured Query Language) adalah bahasa standar untuk mengelola database relasional. Dengan SQL, kita bisa membuat tabel, menambah data, membaca data, mengubah, dan menghapus data.

Analogi:

Bayangkan database seperti perpustakaan:

- **Tabel** = Rak buku
 - **Row** = Satu buku di rak
 - **Column** = Informasi tentang buku (judul, penulis, tahun)
 - **Query** = Cara kita mencari, menambah, atau mengubah buku di perpustakaan
-

2. Perintah SQL Dasar

a. CREATE TABLE

Digunakan untuk membuat tabel baru di database.

```
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  nama VARCHAR(100),  
  email VARCHAR(100)  
);
```

b. INSERT INTO

Digunakan untuk menambah data ke tabel.

```
INSERT INTO users (nama, email) VALUES ('Budi', 'budi@email.com');  
INSERT INTO users (nama, email) VALUES ('Siti', 'siti@email.com');
```

c. SELECT

Digunakan untuk mengambil/membaca data dari tabel.

```
SELECT * FROM users;  
SELECT nama, email FROM users WHERE nama = 'Budi';
```

3. Tipe Data Umum di PostgreSQL

- **VARCHAR(n)**: Teks dengan panjang maksimal n karakter
- **INT**: Angka bulat
- **BOOLEAN**: Nilai true/false
- **TIMESTAMP**: Tanggal dan waktu

Contoh Penggunaan

```
CREATE TABLE products (  
  id SERIAL PRIMARY KEY,  
  nama VARCHAR(100),  
  harga INT,  
  tersedia BOOLEAN,  
  dibuat_pada TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

4. Studi Kasus: Membuat & Membaca Data User

1. Buat tabel **users** seperti contoh di atas
2. Tambahkan minimal 3 data user
3. Tampilkan seluruh data user
4. Tampilkan user dengan nama tertentu

5. Praktik Mandiri (8 Jam)

- Buat tabel **products** dan **categories** dengan struktur dan tipe data yang sesuai
- Isi data dummy menggunakan **INSERT**
- Menampilkan data dengan **SELECT**
- Filter data dengan **WHERE**, urutkan dengan **ORDER BY**
- Simulasi membuat laporan data sederhana (misal: daftar produk terlaris)



Tips untuk Pemula

- Selalu cek hasil query dengan **SELECT** setelah melakukan perubahan
- Gunakan pgAdmin untuk eksplorasi data secara visual
- Dokumentasikan query penting untuk referensi ke depan