

## **Materi: Penggunaan SQL dalam Pengelolaan Basis Data dengan XAMPP (phpMyAdmin)**

### **1. Menerapkan DDL (Data Definition Language), DML (Data Manipulation Language), dan DCL (Data Control Language) pada Pengelolaan Basis Data**

#### **Pengertian DDL, DML, dan DCL:**

- **DDL (Data Definition Language):** Digunakan untuk mendefinisikan dan mengelola struktur basis data.
- **DML (Data Manipulation Language):** Digunakan untuk mengelola data dalam basis data.
- **DCL (Data Control Language):** Digunakan untuk mengatur hak akses dan kontrol data dalam basis data.

### **2. Menerapkan DDL untuk Membuat dan Mengubah Struktur Basis Data**

#### **Membuat Tabel Baru:**

1. Buka phpMyAdmin melalui XAMPP Control Panel.
2. Pilih basis data yang ingin digunakan atau buat basis data baru.
3. Klik tab "Structure" dan kemudian klik "Create table".
4. Isi nama tabel `siswa` dan tentukan jumlah kolom, lalu klik "Go".
5. Isi kolom dengan nama `id`, `nama`, `kelas`, dan `alamat`, dengan tipe data yang sesuai, dan klik "Save".

#### **Mengubah Struktur Tabel:**

1. Di phpMyAdmin, pilih tabel `siswa`.
2. Klik tab "Structure" dan kemudian klik "Add 1 column".
3. Tambahkan kolom `email` dengan tipe data `VARCHAR(100)` dan klik "Save".

#### **Menghapus Tabel:**

1. Di phpMyAdmin, pilih basis data dan tabel `siswa`.
2. Klik tab "Operations" dan kemudian klik "Drop the table (DROP)".
3. Konfirmasi penghapusan tabel.

### **3. Menggunakan DML untuk Mengelola Data**

#### **Menambah Data:**

1. Di phpMyAdmin, pilih tabel `siswa`.
2. Klik tab "Insert".
3. Isi kolom dengan data berikut:
  - o `id`: 1
  - o `nama`: Budi
  - o `kelas`: 12A
  - o `alamat`: Jl. Merdeka No. 10

4. Klik "Go".

#### **Memperbarui Data:**

1. Di phpMyAdmin, pilih tabel `siswa`.
2. Klik tab "Browse" dan kemudian klik ikon "Edit" pada baris data yang ingin diubah.
3. Ubah kolom `alamat` menjadi `Jl. Kebangsaan No. 20` dan klik "Go".

#### **Menghapus Data:**

1. Di phpMyAdmin, pilih tabel `siswa`.
2. Klik tab "Browse" dan kemudian klik ikon "Delete" pada baris data yang ingin dihapus.
3. Konfirmasi penghapusan data.

#### **Mengambil Data:**

1. Di phpMyAdmin, pilih tabel `siswa`.
2. Klik tab "Browse" untuk melihat semua data dalam tabel.

### **4. Mengaplikasikan DCL untuk Mengatur Hak Akses dan Kontrol Data**

#### **Memberikan Hak Akses:**

1. Di phpMyAdmin, buka basis data dan klik tab "Privileges".
2. Klik "Add user account".
3. Isi nama pengguna `user` dan host `localhost`, kemudian beri hak akses `SELECT` dan `INSERT` pada tabel `siswa`.
4. Klik "Go".

#### **Mencabut Hak Akses:**

1. Di phpMyAdmin, buka basis data dan klik tab "Privileges".
2. Klik "Edit Privileges" pada pengguna yang ingin dicabut hak aksesnya.
3. Hapus hak akses `INSERT` dari tabel `siswa` dan klik "Go".

### **5. Menerapkan Perintah SQL Bertingkat**

#### **Menggunakan Subquery:**

1. Di phpMyAdmin, pilih basis data dan klik tab "SQL".
2. Masukkan query berikut dan klik "Go":

```
SELECT nama FROM siswa WHERE id IN (SELECT id FROM nilai WHERE nilai > 80);
```

#### **Menggunakan JOIN:**

1. Di phpMyAdmin, pilih basis data dan klik tab "SQL".
2. Masukkan query berikut dan klik "Go":

```
SELECT siswa.nama, kelas.nama_kelas  
FROM siswa  
JOIN kelas ON siswa.kelas_id = kelas.id;
```

## 6. Menerapkan Penggunaan Function dan Stored Procedure pada Pengelolaan Basis Data

### Membuat Function:

1. Di phpMyAdmin, pilih basis data dan klik tab "SQL".
2. Masukkan query berikut dan klik "Go":

```
CREATE FUNCTION getStudentName(student_id INT) RETURNS VARCHAR(50)  
BEGIN  
    DECLARE student_name VARCHAR(50);  
    SELECT nama INTO student_name FROM siswa WHERE id = student_id;  
    RETURN student_name;  
END;
```

### Menggunakan Function:

1. Di phpMyAdmin, pilih basis data dan klik tab "SQL".
2. Masukkan query berikut dan klik "Go":

```
SELECT getStudentName(1);
```

### Membuat Stored Procedure:

1. Di phpMyAdmin, pilih basis data dan klik tab "SQL".
2. Masukkan query berikut dan klik "Go":

```
CREATE PROCEDURE addStudent(IN student_name VARCHAR(50), IN  
student_class VARCHAR(10), IN student_address TEXT)  
BEGIN  
    INSERT INTO siswa (nama, kelas, alamat) VALUES (student_name,  
student_class, student_address);  
END;
```

### Menggunakan Stored Procedure:

1. Di phpMyAdmin, pilih basis data dan klik tab "SQL".
2. Masukkan query berikut dan klik "Go":

```
CALL addStudent('Ani', '12B', 'Jl. Pancasila No. 5');
```

## Jobsheet: Penggunaan SQL dalam Pengelolaan Basis Data dengan XAMPP (phpMyAdmin)

### Tujuan:

- Memahami dan mengaplikasikan DDL, DML, dan DCL.
- Menggunakan SQL bertingkat untuk query kompleks.
- Membuat dan menggunakan function dan stored procedure.

## Alat dan Bahan:

- Komputer dengan koneksi internet.
- XAMPP dengan MySQL Server.
- phpMyAdmin.

## Langkah Kerja:

1. **Menggunakan DDL untuk Membuat dan Mengubah Struktur Basis Data:**
  - Buat tabel `siswa` dengan kolom `id`, `nama`, `kelas`, dan `alamat`.
  - Tambahkan kolom `email` pada tabel `siswa`.
  - Hapus tabel `siswa`.
2. **Menggunakan DML untuk Mengelola Data:**
  - Tambahkan data ke dalam tabel `siswa`.
  - Perbarui data pada tabel `siswa`.
  - Hapus data dari tabel `siswa`.
  - Ambil data dari tabel `siswa`.
3. **Mengaplikasikan DCL untuk Mengatur Hak Akses dan Kontrol Data:**
  - Berikan hak akses kepada pengguna.
  - Cabut hak akses dari pengguna.
4. **Menerapkan Perintah SQL Bertingkat:**
  - Gunakan subquery untuk mengambil nama siswa dengan nilai lebih dari 80.
  - Gabungkan data dari tabel `siswa` dan `kelas` dengan perintah JOIN.
5. **Menerapkan Penggunaan Function dan Stored Procedure:**
  - Buat function untuk mengambil nama siswa berdasarkan ID.
  - Panggil function `getStudentName` dengan parameter ID siswa.
  - Buat stored procedure untuk menambah data siswa.
  - Panggil stored procedure `addStudent` dengan parameter nama, kelas, dan alamat siswa.

## Asesmen

### Soal Teori:

1. Apa yang dimaksud dengan DDL, DML, dan DCL?
2. Sebutkan dan jelaskan fungsi perintah CREATE, ALTER, dan DROP.
3. Apa perbedaan antara subquery dan join?
4. Jelaskan fungsi dari function dan stored procedure dalam basis data.

### Praktik:

1. Buat tabel `guru` dengan kolom `id`, `nama`, `mata_pelajaran`, dan `alamat`.
2. Tambahkan minimal 3 data guru ke dalam tabel `guru`.
3. Perbarui alamat salah satu guru.
4. Buat function untuk mengambil nama guru berdasarkan ID.
5. Buat stored procedure untuk menambah data guru.

**Penilaian:**

- Keberhasilan dalam membuat dan mengubah struktur basis data.
- Ketepatan dalam mengelola data menggunakan DML.
- Ketepatan dalam mengatur hak akses dan kontrol data menggunakan DCL.
- Kelengkapan dan ketepatan dalam menerapkan perintah SQL bertingkat.
- Keberhasilan dalam membuat dan menggunakan function serta stored procedure.