

# MODUL PRAKTIKUM BASIS DATA

## DCL (Data Control Language) – User/Role, GRANT, dan REVOKE

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

Setelah mengikuti praktikum ini, mahasiswa mampu:

1. Memahami konsep keamanan dalam basis data menggunakan DCL.
2. Membuat dan mengelola user serta role pada DBMS.
3. Memberikan hak akses (privilege) menggunakan perintah **GRANT**.
4. Mencabut hak akses menggunakan perintah **REVOKE**.
5. Menerapkan kontrol akses pada sistem basis data secara tepat.

### B. Tujuan Praktikum

Mahasiswa diharapkan dapat:

- Menjelaskan fungsi **DCL (Data Control Language)**.
- Membuat **user** dan **role** dalam DBMS.
- Menggunakan perintah **GRANT** untuk pemberian hak akses.
- Menggunakan perintah **REVOKE** untuk pencabutan hak akses.
- Menganalisis hak akses pada suatu basis data.

### C. Dasar Teori

#### 1. Data Control Language (DCL)

DCL adalah bagian dari SQL yang digunakan untuk mengatur hak akses terhadap database. DCL berfungsi untuk menjaga keamanan data dengan menentukan siapa yang dapat mengakses dan melakukan operasi tertentu.

Perintah utama dalam DCL:

- **GRANT** → Memberikan hak akses
- **REVOKE** → Mencabut hak akses

## 2. User dan Role

- **User:** Akun yang digunakan untuk mengakses database.
- **Role:** Sekumpulan hak akses yang dapat diberikan ke user.

Keuntungan menggunakan role:

- Mempermudah manajemen hak akses
- Mengurangi redundansi pengaturan

### a. Membuat User

```
CREATE USER nama_user [WITH] PASSWORD 'password';
```

Atau

```
CREATE ROLE nama_user LOGIN PASSWORD 'password';
```

### b. Membuat Role

```
CREATE ROLE nama_role;
```

## 3. Privileges (Hak Akses)

Beberapa hak akses yang umum:

- SELECT → membaca data
- INSERT → menambah data
- UPDATE → mengubah data
- DELETE → menghapus data
- ALL PRIVILEGES → semua hak akses

## 4. GRANT

Digunakan untuk memberikan hak akses kepada user atau role.

Sintaks:

```
GRANT privilege ON nama_tabel TO user;
```

### a. Memberikan Hak Akses ke Role

```
GRANT jenis_privilege [, ...]
ON TABLE nama_tabel
TO nama_role;
```

Untuk level schema:

```
GRANT jenis_privilege
ON ALL TABLES IN SCHEMA nama_schema
TO nama_role;
```

b. Memberikan Role ke User

```
GRANT nama_role TO nama_user;
```

c. Memberikan Hak Akses Langsung ke User

```
GRANT jenis_privilege [, ...]
ON TABLE nama_tabel
TO nama_user;
```

## 5. REVOKE

Digunakan untuk mencabut hak akses.

Sintaks:

```
REVOKE privilege ON nama_tabel FROM user;
```

a. Mencabut Hak Akses

```
REVOKE jenis_privilege [, ...]
ON TABLE nama_tabel
FROM nama_user_atau_role;
```

b. Menghapus Role dari User

```
REVOKE nama_role FROM nama_user;
```

## D. Percobaan

Sebuah sistem database kampus memiliki tabel **mahasiswa**.  
Admin ingin:

1. Membuat role **role\_mahasiswa** dengan akses baca (SELECT)

2. Membuat user **user\_mhs**
3. Memberikan role ke user
4. Menguji akses
5. Mengubah dan mencabut hak akses

### Langkah 1: Membuat Database dan Tabel

```
CREATE TABLE mahasiswa (  
  nim VARCHAR(10) PRIMARY KEY,  
  nama VARCHAR(50),  
  jurusan VARCHAR(50)  
);
```

Isi data:

```
INSERT INTO mahasiswa VALUES  
( '101', 'Andi', 'Informatika'),  
( '102', 'Budi', 'Sistem Informasi');
```

### Langkah 2: Membuat Role

```
CREATE ROLE role_mahasiswa;
```

### Langkah 3: Memberikan Hak Akses ke Role

```
GRANT SELECT ON TABLE mahasiswa TO role_mahasiswa;
```

### Langkah 4: Membuat User

```
CREATE USER user_mhs WITH PASSWORD '12345';
```

### Langkah 5: Memberikan Role ke User

```
GRANT role_mahasiswa TO user_mhs;
```

### Langkah 6: Uji Akses (Login sebagai user\_mhs)

```
SELECT * FROM mahasiswa;
```

☑ Output:

```

nim | nama | jurusan
-----+-----+-----
101 | Andi | Informatika
102 | Budi | Sistem Informasi

```

### ✗ Coba INSERT (harus gagal)

```
INSERT INTO mahasiswa VALUES ('103','Citra','Informatika');
```

#### Output:

```
ERROR: permission denied for table mahasiswa
```

### Langkah 7: Tambah Hak Akses (INSERT)

```
GRANT INSERT ON TABLE mahasiswa TO role_mahasiswa;
```

### ✓ Uji kembali INSERT:

```
INSERT INTO mahasiswa VALUES ('103','Citra','Informatika');
```

#### Output:

```
INSERT 0 1
```

### Langkah 8: Melihat Privilege

```
SELECT grantee, privilege_type
FROM information_schema.role_table_grants
WHERE table_name = 'mahasiswa';
```

#### Output:

```

grantee          | privilege_type
-----+-----
role_mahasiswa  | SELECT
role_mahasiswa  | INSERT

```

### Langkah 9: Mencabut Hak Akses

```
REVOKE INSERT ON TABLE mahasiswa FROM role_mahasiswa;
```

**✘ Uji kembali INSERT:**

```
INSERT INTO mahasiswa VALUES ('104', 'Dewi', 'SI');
```

**Output:**

```
ERROR: permission denied for table mahasiswa
```

**Langkah 10: Menghapus Role dari User**

```
REVOKE role_mahasiswa FROM user_mhs;
```

**✘ Uji SELECT:**

```
SELECT * FROM mahasiswa;
```

**Output:**

```
ERROR: permission denied for table mahasiswa
```

**E. LATIHAN PRAKTIKUM**

**Studi Kasus SISTEM PENJUALAN**

Sebuah sistem penjualan memiliki 2 tabel utama:

- produk(id\_produk, nama\_produk, harga, stok)
- penjualan(id\_penjualan, tanggal, id\_produk, jumlah)

Terdapat 2 user:

- user\_kasir → bertugas melakukan transaksi penjualan
- user\_manager → bertugas mengawasi data dan laporan

**Tugas Praktikum**

**SOAL 1 – Pembuatan Role**

Buat 2 role:

1. role\_kasir
2. role\_manager

**Ketentuan:**

role\_kasir:

- SELECT pada tabel produk
- INSERT pada tabel penjualan

role\_manager:

- SELECT pada tabel produk
- SELECT pada tabel penjualan
- UPDATE pada tabel produk

### SOAL 2 – Pembuatan User

Buat 2 user:

- user\_kasir
- user\_manager

**Ketentuan:**

- Berikan password pada masing-masing user

Hubungkan:

- user\_kasir → role\_kasir
- user\_manager → role\_manager

### SOAL 3 – Pengujian Hak Akses

#### A. Login sebagai user\_kasir

Lakukan:

1. Tampilkan data produk
2. Tambah data ke tabel penjualan
3. Coba UPDATE harga produk

**Analisis:**

- Perintah mana yang berhasil dan gagal?
- Jelaskan alasan berdasarkan privilege

#### B. Login sebagai user\_manager

Lakukan:

1. Tampilkan data penjualan
2. Update harga produk
3. Coba INSERT ke tabel penjualan

**Analisis:**

- Identifikasi perintah yang gagal
- Jelaskan penyebabnya

### SOAL 4 – Modifikasi Hak Akses

Manajemen ingin perubahan:

- Kasir boleh **mengurangi stok produk (UPDATE produk)**
- Manager boleh **menambah data penjualan (INSERT)**

**Tugas:**

- Ubah hak akses tanpa menghapus role
- Uji kembali kedua user

### **SOAL 5 – Pencabutan Hak Akses**

Lakukan:

1. Cabut hak INSERT penjualan dari role\_kasir
2. Uji kembali apakah kasir masih bisa melakukan transaksi

### **SOAL 6 – Audit Hak Akses**

Buat query untuk:

1. Menampilkan siapa saja yang memiliki akses pada tabel penjualan
2. Menampilkan semua privilege pada tabel produk
3. Menampilkan semua role dan hak login

### **SOAL 7 – Studi Kasus Masalah (Troubleshooting)**

Terjadi masalah:

- user\_kasir tidak bisa INSERT ke tabel penjualan, padahal sebelumnya bisa

**Tugas:**

- Identifikasi kemungkinan penyebab
- Tunjukkan query untuk mengecek hak akses
- Berikan solusi SQL

### **SOAL 8 – Analisis Keamanan**

Jawab pertanyaan berikut:

1. Apa risiko jika kasir diberi akses UPDATE pada semua tabel?
2. Mengapa manager tidak diberi DELETE?
3. Apa keuntungan menggunakan role dibanding langsung ke user?