

Pertemuan 14 Basis Data

Anisa Intania Putri

2024071010

1. Pertama buat dulu usernya

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0022 detik.)

```
GRANT USAGE ON *.* TO 'anisa'@'localhost' IDENTIFIED BY '1234';
```

[\[Edit dikotak \]](#) [\[Ubah \]](#) [\[Buat kode PHP \]](#)

Membuat user anisa dengan password '1234'. GRANT USAGE tidak memberikan akses ke database mana pun. USAGE hanya membuat user dapat login tanpa memiliki izin apa pun.

2. Memberi hak akses database

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0025 detik.)

```
GRANT SELECT, INSERT, UPDATE ON little_lemon.* TO 'anisa'@'localhost';
```

[\[Edit dikotak \]](#) [\[Ubah \]](#) [\[Buat kode PHP \]](#)

Memberikan hak akses SELECT, INSERT, UPDATE kepada user anisa di database little_lemon.

3. Menggunakan pernyataan SHOW GRANTS

Kueri SQL Anda berhasil dieksekusi.

```
SHOW GRANTS FOR 'anisa'@'localhost';
```

☐ Profil [\[Edit dikotak \]](#) [\[Ubah \]](#) [\[Buat kode PHP \]](#) [\[Segarkan \]](#)

Extra options

Grants for anisa@localhost

```
GRANT USAGE ON *.* TO `anisa`@`localhost` IDENTIFI...  
GRANT SELECT, INSERT, UPDATE ON `little_lemon`. * T...
```

Kita lihat disini user anisa sudah mendapatkan hak akses SELECT, INSERT, UPDATE.

4. Mengatur Password untuk akun user anisa

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0146 detik.)

```
SET PASSWORD FOR 'anisa'@'localhost' = PASSWORD('pw1');
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Perintah ini digunakan untuk mengganti password user.

5. Menggunakan pernyataan FLUSH PRIVILEGES

```
FLUSH PRIVILEGES;
```

Perintah ini dipakai untuk memuat ulang hak akses (privileges) dari tabel internal MySQL.

6. Menggunakan pernyataan REVOKE

Tampilkan kotak kueri

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0148 detik.)

```
REVOKE SELECT ON `little_lemon`. * FROM 'anisa'@'localhost';
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Tampilkan kotak kueri

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0139 detik.)

```
REVOKE INSERT, UPDATE ON `little_lemon`. * FROM 'anisa'@'localhost';
```

[Edit dikotak] [Ubah] [Buat kode PHP]

Perintah ini dipakai untuk menghapus hak akses SELECT, INSERT, UPDATE dari user anisa.

Hak Akses SELECT, INSERT, UPDATE berhasil dihapus sudah tidak ada

Kueri SQL Anda berhasil dieksekusi.

```
SHOW GRANTS FOR 'anisa'@'localhost';
```

☐ Profil [Edit dikotak] [Ubah] [Buat kode PHP] [Segarkan]

Extra options

Grants for anisa@localhost

GRANT USAGE ON *.* TO `anisa`@`localhost` IDENTIFI...

7. Menggunakan pernyataan DROP USER

Sebelum DROP USER

User	Host
root	127.0.0.1
root	::1
anisa	localhost
pma	localhost
root	localhost

Sesudah DROP USER

✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0148 detik.)

```
DROP USER 'anisa'@'localhost';
```

[\[Edit kotak \]](#) [\[Ubah \]](#) [\[Buat kode PHP \]](#)

Perintah ini digunakan untuk menghapus akun user secara permanen.

User	Host
root	127.0.0.1
root	::1
pma	localhost
root	localhost

Setelah di hapus user “anisa” hilang

Kesimpulan:

Pada praktikum ini saya belajar bagaimana cara mengelola user dan hak akses sebagai bagian penting dari keamanan database. Saya membuat user baru, memberikan hak akses sesuai kebutuhan, mengecek hak aksesnya, mengganti password, memuat kembali konfigurasi privilege, mencabut hak akses, dan akhirnya menghapus user tersebut. Dari proses ini saya memahami bahwa pengaturan user sangat penting untuk membatasi siapa saja yang boleh mengakses data dan apa yang boleh mereka lakukan. Praktikum ini juga membuat saya lebih paham bahwa keamanan database bukan hanya tentang enkripsi, tetapi juga mengatur hak akses agar sistem tetap aman dan terkontrol.