PERTEMUAN 14
PENGELOLAAN KEAMANAN

MATERI

a. MySQL *grant tables*, meliputi tabel user, db, host, tables_priv, dan columns_priv
b. Proses yang digunakan untuk autentikasi koneksi ke MySQL server dan verifikasi kebutuhan hak (*privileges*) untuk melakukan berbagai macam operasi
c. Kebutuhan pernyataan-pernyataan untuk mengelola akun user MySQL, meliputi pernyataan GRANT, SHOW GRANTS, SET PASSWORD, FLUSH PRIVILEGES, REVOKE, dan DROP USER

A. SISTEM HAK AKSES

Database mysql merupakan database administrative yang berisi table-tabel yang berkaitan dengan pengamanan instalasi MySQL, penyimpanan fungsi-fungsi pendefinisian user, dan penyediaan data yang berkaitan dengan sistem *help* MySQL dan fungsionalitas *time-zone*. Dan tentunya fokus pada pencegahan akses-akses yang tidak berhak pada MySQL server merupakan tabel-tabel yang berkaitan dengan keamanan, dimana disebut sebagai *grant tables*.

A.1 MySQL Grant Tables

Ketika Anda menginstal MySQL, lima tabel grant ditambahkan ke database mysql. Tabel-tabel ini adalah user, db, host, tables_priv, dan columns_priv.
Masing-masing tabel berisi dua tipe kolom:

- Scope columns: Kolom dalam tabel grant yang menentukan siapa yang berhak mengakses MySQL server dan jangkauan dari akses tersebut.
- Privilege columns: Kolom dalam tabel grant yang menentukan operasi-operasi apa saja yang dapat dilakukan oleh user yang diidentifikasi dalam *scope columns*.

Melaui penggunaan hak akses, kelima tabel tersebut berpartisipasi dalam proses penentuan apakah seorang user dapat melakukan operasi tertentu.

a. Tabel User
   Tabel user adalah tabel grant utama dalam database mysql. Tabel ini mengontrol siapa yang dapat terhubung ke MySQL, dari host mana mereka dapat terhubung, dan hak akses global (*global privileges*) apa yang mereka punyai.
   Tipe-tipe kolom:
   - *Scope columns*: Meliputi kolom Host, User, dan Password. Ketika koneksi diinisialisasi, koneksi harus dibuat dari host yang telah dispesifikasikan di kolom Host. Dengan tambahan, nama user yang membuat koneksi harus sesuai dengan nilai yang terdapat pada kolom User, bergitu juga dengan password. Koneksi akan diijinkan jika ketiga nilai tersebut sesuai.
   - *Data-related privilege columns*: Meliputi kolom-kolom hak akses yang mengijinkan operasi-operasi yang berkaitan dengan data secara global. Terdapat 11 kolom yang berkaitan dengan hak akses data.
   - *Administrative privilege columns*: Meliputi kolom-kolom hak akses yang mengijinkan operasi-operasi administratif ke MySQL server. Terdapat 8 kolom hak akses administratif.

- *Encryption-related privilege columns*: Meliputi kolom-kolom ssl_type, ssl_cipher, x509_issuer, dan x509_subject, yang mendefinisikan apakah akun user memerlukan koneksi yang aman dan mendefinisikan sifat dasar dari koneksi tersebut.
- *Connection-related privilege columns*: Meliputi kolom-kolom max_questions, max_updates, dan max_connections, yang mendefinisikan apakah sebuah batasan/limit harus diterapkan pada jumlah query, jumlah update data, dan jumlah koneksi yang dapat dibuat dalam satu jam.

```
SELECT Host, User, Select_priv, Process_priv, ssl_type, max_updates
FROM user
WHERE User='user1';
```

b. Tabel db

Tujuan dari tabel db adalah memberikan hak akses database secara spesifik pada user. Hak-hak akses yang diterapkan pada tabel db juga secara spesifik pada database tertentu.

Kolom-kolom yang terdapat dalam tabel db adalah:

- *Scope columns*: Meliputi kolom-kolom Host, DB, dan User. Untuk menerapkan hak akses dalam tabel ini, koneksi harus dibuat dari host yang dispesifikasi pada kolom Host, dan akun nama user yang membuat koneksi harus sesuai dengan nilai pada kolom User. Jika kolom host kosong, maka hak akses juga didefinisikan di tabel host yang diterapkan. Hak-hak akses yang terdapat pada tabel db hanya diterapkan pada database yang dispesifikasikan pada kolom Db.
- *Privilege columns*: Meliputi hak-hak akses yang dapat diterapkan pada tingkat database. Terdapat 11 hak-hak akses yang berkaitan dengan data yang digunakan untuk mengijinkan operasi-operasi yang berkaitan dengan data.

```
SELECT Host, Db, User, Select_priv, Update_priv
FROM db
WHERE User='user1';
```

c. Tabel Host

Tabel host berkaitan dengan tabel db dan diperiksa hanya ketika seorang user terdaftar dalam tabel db namun kolom Host kosong. Kombinasi dari dua tabel ini mengijinkan Anda untuk menerapkan hak akses ke user yang terkoneksi dari banyak host. Misal, jika seorang user bernama Sarah terkoneksi ke MySQL dari domain big.domain1.com dan little.domain1.com, Anda dapat menambahkan user tersebut ke tabel db, dengan nilai host kosong, dan kemudian tambahkan dua host ke tabel host, dengan menspesifikasikan nama database yang sama pada baris Sarah dari tabel db dan baris big.domain1.com dan little.domain1.com dari tabel host.

Kolom-kolom yang terdapat dalam tabel host adalah:

- *Scope columns*: Meliputi kolom Host dan Db. Untuk menerapkan hak akses pada tabel ini, kolom Host pada tabel db harus kosong, dan koneksi harus dibuat dari host yang dispesifikasi dalam kolom Host dari tabel host. Hak-hak akses yang terdapat pada tabel host diterapkan hanya pada database yang dispesifikasikan pada kolom Db dari tabel host.
- *Privilege columns*: Meliputi hak-hak akses tersebut yang dapat diterapkan pada tingkat database untuk akun user yang mengakses database dari host tertentu.

```
SELECT Host, Db, Select_priv, Update_priv
FROM host
WHERE Host='host1.domain1.com' OR Host='host2.domain1.com';
```

d. Tabel tables_priv
Tabel ini lebih spesifik ke hak akses tingkat tabel. Hak-hak akses yang terdaat di tabel ini diterapkan hanya pada tabel yang dispesifikasikan pada tabel tables_priv.
- o *Scope columns*: Meliputi kolom Host, Db, User, dan Table_name.
- o *Privilege columns*: Meliputi kolom Table_priv dan Column_priv. Kolom Table_priv mendefinisikan hak akses yang diterapkan pada tingkat tabel. Kolom Column_priv mendefinisikan hak akses yang diterapkan pada tingkat kolom.

```
SELECT Host, Db, User, Table_name, Table_priv, Column_priv
FROM tables_priv
WHERE User='user1';
```

e. Tabel columns_priv
Tabel ini menunjukkan hak-hak akses yang berhubungan dengan kolom-kolom secara individu.
- o *Scope columns*: Meliputi kolom Host, Db, User, Table_name, dan Column_name.
- o *Privilege columns*: Meliputi kolom Column_priv, yang mendefinisikan hak-hak akses yang diterapkan pada tingkatan kolom.

```
SELECT Host, Db, User, Table_name, Column_name, Column_priv
FROM columns_priv
WHERE User='user1';
```

A.2 Hak-Hak Akses MySQL

Tabel-tabel user, db, dan host berisi kolom-kolom dimana masing-masing merepresentasikan hak-hak akses individu. Ketiga tabel meliputi hak-hak akses yang berkaitan dengan data, dimana secara spesifik berhubungan dengan pengelolaan data. Tabel berikut menjelaskan hak-hak akses yang dapat diberikan pada tabel user, db, dan host, dan ditunjukkan pulan tabel mana yang berisi hak akses mana.

Column	Type	Allows user to	user table	db table	host table
Select_priv	Data-related	Query data in a database.	X	X	X
Insert_priv	Data-related	Insert data in a database.	X	X	X
Update_priv	Data-related	Update data in a database.	X	X	X
Delete_priv	Data-related	Delete data from a database.	X	X	X
Create_priv	Data-related	Create a table in a database.	X	X	X
Drop_priv	Data-related	Remove a table from a database.	X	X	X
Reload_priv	Administrative	Reload the data in the grant tables in MySQL.	X		

Lanjutan.

Column	Type	Allows user to	user table	db table	host table
Shutdown_priv	Administrative	Shut down the MySQL server.	X		
Process_priv	Administrative	View a list of MySQL processes.	X		
File_priv	Administrative	Export data from a database into a file.	X		
Grant_priv	Data-related	Grant privileges on database objects.	X	X	X
Index_priv	Data-related	Create and delete indexes in a database.	X	X	X
Alter_priv	Data-related	Alter database objects.	X	X	X
Show_db_priv	Administrative	View all databases.	X		
Super_priv	Administrative	Perform advanced administrative tasks.	X		
Create_tmp_table_priv	Data-related	Create temporary tables.	X	X	X
Lock_tables_priv	Data-related	Place locks on tables.	X	X	X
Repl_slave_priv	Administrative	Read binary logs for a replication master.	X		

Repl_client_priv	Administrative	Request information about slave and master servers used for replication.	X		
ssl_type	Encryption-related	Specifies whether a secure connection is required. If required, the column specifies the type of secure connection.	X		
ssl_cipher	Encryption-related	Specifies the cipher method that should be used for a connection. If the column is blank, no special cipher method is required.	X		
x509_issuer	Encryption-related	Specifies the name of the certificate authority that issues the x509 certificate. The name should be used for an x509 connection. If the column is blank, the issuer name is not required.	X		
x509_subject	Encryption-related	Specifies the subject that should be included on the x509 certificate when establishing a secure connection. If the column is blank, the subject is not required.	X		
max_questions	Connection-related	Specifies the number of queries that an account can issue in an hour. If set to 0, the user account can issue an unlimited number of queries.	X		
max_updates	Connection-related	Specifies the number of data updates that an account can perform in an hour. If set to 0, the user account can perform an unlimited number of updates.	X		
max_connections	Connection-related	Specifies the number of connections that an account can establish in an hour. If set to 0, the user account can connect an unlimited number of times.	X		

Hak-hak akses yang terdaftar dalam tabel tersebut diurutkan berdasarkan kemunculan mereka dalam tabel user. Dikarenakan permisi/ijin dalam tabel user diterapkan secara global, merupakan satu-satunya tabel yang berisi semua hak-hak akses.

B. KONTROL AKSES MYSQL

Ketika MySQL mengijinkan user untuk melakukan berbagai macam operasi dalam lingkungan MySQL, dia pertama-tama mengautentikasi koneksi tersebut untuk mengijinkan akses ke MySQL server, dan kemudian memverifikasi hak-hak akses yang diberikan pada akun user untuk menentukan apakah *request* operasinya diijinkan.

B.1 Autentikasi Koneksi

Langkah pertama untuk mengijinkan user mengakses MySQL server adalah memastikan bahwa user memiliki akses. Jika akses diijinkan, MySQL mengautentikasi koneksi. Koneksi diautentikasi hanya jika nilai Host dalam tabel user sesuai dengan nama dari host yang melakukan koneksi. Dengan tambahan, nama user yang membuat koneksi harus sesuai dengan nilai yang tedapat dalam tabel User, dan jika password diperlukan, password yang diberikan untuk koneksi harus sesuai dengan nilai yang terdapat dalam kolom Password. Jika parameter-parameter yang diberikan koneksi sesuai dengan nilai-nilai dalam tabel user, maka koneksi diijinkan.

Jika Anda ingin mengijinkan user *anonymous* (tak dikenal/bebas) atau berencana mengijinkan user terhubung dari host manapun, Anda harus merencanakan tabel user Anda dengan hati-hati untuk memastikan bahwa hak user sesuai dengan hak aksesnya. Untuk merencanakan akun user Anda, Anda harus mengingat bagaimana MySQL mengakses tabel user:

a. Ketika MySQL server mulai, data dari tabel user disalin ke memori.
b. Ketika klien berusaha log in ke server, akun user diperiksa/disesuaikan dengan data user di memori.
c. Server menggunakan inputan user untuk mengautentikasi user, berdasarkan nilai Host dan User.

B.2 Verifikasi Hak Akses

Setelah MySQL mengautentikasi seorang user, dia memeriksa hak-hak akses yang berhubungan dengan akun user untuk menentukan operasi-operasi apa yang dapat dilakukan oleh user. Ketika memverifikasi hak-hak akses, pertama MySQL memeriksa tabel user. Jika hak akses tidak diberikan untuk aku yang terdapat dalam tabel user, MySQL memeriksa tabel db, dan jika sesuai, tabel host. Jika hak akses tidak diberikan pada tingkatan database, MySQL memeriksa tabel tables_priv, dan jika sesuai, tabel columns_priv. Jika, setelah pemeriksaan semua tabel, ternyata operasi tidak diijinkan, operasi gagal.

C. MENGELOLA AKUN USER MYSQL

MySQL menyediakan sejumlah pernyataan-pernyataan SQL yang mengijinkan Anda mengelola akun-akun Anda. Dengan menggunakan pernyataan-pernyataan ini, Anda dapat menambahkan akun user ke sistem Anda dan pemberian hak akses ke akun tersebut. Anda juga dapat menampilkan hak-hak akses yang telah diberikan pada akun seorang user, atau Anda dapat mengganti password akun tersebut. Dengan

tambahan, pernyataan-pernyataan tersebut juga mengijinkan Anda untuk me-*revoke* (mencabut) hak-hak akses dan menghapus user dari sistem Anda.

C1. Menambahkan User dan Perijinan Hak Akses

Anda dapat menambahkan sebuah akun user ke sistem Anda dan memberikan hak akses ke akun tersebut dengan menggunakan pernyataan GRANT, yang mengijinkan Anda untuk melakukan kedua operasi tersebut dalam satu pernyataan. Meskipun Anda dapat memasukkan informasi akun secara langsung ke tabel grant, pernyataan GRANT lebih mudah dan lebih sedikit dalam menyebabkan error yang dapat terjadi saat penambahan informasi dalam tabel grant yang berbeda yang konflik dengan lainnya.
Sekali Anda menambahkan akun user dan memberikan hak-hak akses ke akun tersebut, Anda dapat menggunakan pernyataan SHOW GRANTS untuk menampilkan detil dari akun tersbeut.

a. Menggunakan pernyataan GRANT
Sintaks:

```
GRANT <privilege> [(<column> [{, <column>}...])]
[{, <privilege> [(<column> [{, <column>}...])]}...]
ON {<table> | * | *.* | <database>.*}
TO '<user>'@'<host>' [IDENTIFIED BY [PASSWORD] '<new password>']
[{, '<user>'@'<host>' [IDENTIFIED BY [PASSWORD] '<new password>']}...]
[REQUIRE {NONE | SSL | X509 | {<require definition>}]
[WITH <with option> [<with option>...]]

<require definition>::=
<require option> [[AND] <require option>] [[AND] <require option>]

<require option>::=
{CIPHER '<string>'}
| {ISSUER '<string>'}
| {SUBJECT '<string>'}

<with option>::=
{GRANT OPTION}
| {MAX_QUERIES_PER_HOUR <count>}
| {MAX_UPDATES_PER_HOUR <count>}
| {MAX_CONNECTIONS_PER_HOUR <count>}
```

Sebagaimana pernyataan-pernyataan SQL lainnya, pernyatan GRANT tersusun atas banyak klausa-klausa dan opsi-opsi, beberapa di antaranya harus diisikan dan lainnya sifatnya opsional.
Tabel berikut ini menunjukkan *case sensitivity* dari masing-masing kolom:

Column	Case sensitive?
Host	No
User	Yes
Password	Yes
Db	Yes
Table_name	Yes
Column_name	No

Mendefinisikan klausa GRANT

Klausa GRANT menentukan tipe dari hak akses yang harus diberikan pada akun user.

```
GRANT <privilege> [(<column> [{, <column>}...])]
[{, <privilege> [(<column> [{, <column>}...])]}...]
```

Setipa hak akses berhubungan dengan kolom hak akses dalam tabel user, db, atau host atau dengan salah satu hak akses yang terdaftar dalam kolom Table_priv dan Column_priv dari tabel tables_priv dan columns_priv.
Tabel berikut menjelaskan semua hak-hak akses yang dapat Anda gunakan dalam pernyataan GRANT (dalam <privilege>).

Privilege syntax	Columns set to Y	Actions permitted
ALL [PRIVILEGES]	All columns (at the level that the GRANT statement applies to), except the Grant_priv column	Execute all statements except GRANT, REVOKE, and DROP USER statements.
ALTER	Alter_priv	Execute ALTER TABLE statements.
CREATE	Create_priv	Execute CREATE TABLE statements.
CREATE TEMPORARY TABLES	Create_tmp_table_priv	Execute CREATE TEMPORARY TABLE statements.
DELETE	Delete_priv	Execute DELETE statements.
DROP	Drop_priv	Execute DROP TABLE statements.
FILE	File_priv	Execute SELECT...INTO OUTFILE and LOAD DATA INFILE statements.
INDEX	Index_priv	Execute CREATE INDEX and DROP INDEX statements.
INSERT	Insert_priv	Execute INSERT statements.
GRANT OPTION	Grant_priv	Execute GRANT, REVOKE, and DROP USER statements.
LOCK TABLES	Lock_tables_priv	Execute LOCK TABLES statements. (User must also have SELECT privilege.)
PROCESS	Process_priv	Execute SHOW FULL PROCESSLIST statements.
RELOAD	Reload_priv	Execute FLUSH statements.
REPLICATION CLIENT	Repl_client_priv	Locate slave and master replication servers.
REPLICATION SLAVE	Repl_slave_priv	Read binary log events from master replication servers.
SELECT	Select_priv	Execute SELECT statements.
SHOW DATABASES	Show_db_priv	Execute SHOW DATABASES statements.
SHUTDOWN	Shutdown_priv	Use shutdown option of the mysqladmin client utility

SUPER	Super_priv	Execute CHANGE MASTER, KILL, PURGE MASTER LOGS, and SET GLOBAL statements; use the debug command of the mysqladmin client utility; and connect to MySQL server even if max_connections system variable limit has been reached.
UPDATE	Update_priv	Execute UPDATE statements.
USAGE	No columns affected	No actions permitted. Option used to add user with no privileges or to update user options not related to privileges.

Mendefinisikan klausa ON

Setelah Anda mendefinisikan klausa GRANT, Anda dapat mendefinisikan klausa ON, sintaksnya:

```
ON {<table> | * | *.* | <database>.*}
```

Klausa ON menspesifikasikan ke tabel atau database mana pernyataan GRANT diterapkan. Sebagaimana Anda lihat, klausa itu mencakup empat opsi. Opsi yang Anda pilih menentukan tingkatan dimana hak akses diterapkan. Daftar berikut menggambarkan bagaimana opsi-opsi dapat digunakan untuk tiap tingkatan:

- *Global*: Menggunakan opsi *double wildcard* (*.*) untuk menspesifikasikan bahwa hak-hak akses harus diaplikasikan ke MySQL server dan ke semua database dan tabel-tabelnya. Anda juga dapat menggunakan opsi single wildcard (*) jika tidak ada database yang aktif ketika mengeksekusi pernyataan GRANT; sebaliknya, opsi sing*le wildcard* diterapkan hanya ke database aktif.
- *Database*: Gunakan opsi <database>.* untuk menspesifikasikan bahwa hak-hak akses harus diterapkan pada database tertentu secara keseluruhan dan ke semua tabel dalam database. Anda juga dapat menggunakan opsi single wildcard (*) jika database akif.
- *Table*: Gunakan opsi <table> untuk menspesifikasikan bahwa hak-hak akses harus diterapkan pada tabel tersebut. Opsi ini biasanya didahului oleh nama database dan sebuah titik (*period*). Contoh, tabel buku dalam database toko menunjuk ke toko.buku.
- *Column*: Gunakan opsi <table> untuk menspesifikasikan bahwa hak-hak akses harus diterapkan hanya ke tabel tersebut, sebagaimana saat memberikan hak akses pada tingkatan tabel. Untuk membuat tabel GRANT spesifik ke kolom, opsi <tabel> seharusnya digunakan bersamaan dengan kolom yang dispesifikasikan dalam klausa GRANT.

Mendefinisikan klausa TO

Klausa terakhir yang diperlukan yang harus Anda sertakandalam pernyataan GRANT adalah klausa TO, dimana sintaksnya adalah:

```
TO '<user>'@'<host>' [IDENTIFIED BY [PASSWORD] '<new password>']
[{, '<user>'@'<host>' [IDENTIFIED BY [PASSWORD] '<new password>']}...]
```

Di sintaks tersebut, Anda dapat memberikan hak akses ke satu atau lebih user, selama Anda memisahkan pasangan definisi user dengan koma. Untuk setiap akun user, Anda harus menspesifikasikan nama user, nama host, dan opsional password.

- *Host*: Host dimana user akan terkoneksi. Anda dapat menspesifikasikan nama host, IP address, atau localhost. Anda dapat menggunakan tanda persen (%) atau *underscore* (_) dalam nama host. Tanda persen menandakan bahwa semua host bisa terkoneksi.
- *User*: Nama user berkaitan dengan akun user yang pernah dibuat. Anda tidak dapat menggunakan *wildcards*, namun Anda dapat mengosongkannya, yang menandakan bahwa user tak dikenal/bebas diijinkan mengakses.
- *Password*: Password berkaitan dengan akun user yang pernah dibuat. Anda tidak dapat menggunakan *wildcards*, namun Anda dapat mengosongkannya. Jika kososng, user harus menyediakan password kosong untuk log in ke server. Ketika Anda menggunakan pernyataan GRANT untuk membuat password, password secara otomatis dienkripsi ketika disimpan ke tabel user.

b. Membuat pernyataan GRANT dasar
Contoh pernyataan GRANT pertama adalah membuat akun user dan perijinan hak akses tingkat global ke akun tersebut:

```
GRANT ALL
ON *.*
TO 'user1'@'domain1.com' IDENTIFIED BY 'pw1';
```

Dalam pernyataan tersebut, klausa GRANT menyertakan satu hak akses (ALL). Hak akses ALL mengijinkan semua hak akses ke user kecuali mengijinkan user kecuali hak akses untuk memberikan ijin (*grant*) dan pencabutan ijin (*revoke*). Klausa ON menyertakan opsi *double wildcards* (*.*), dimana menandakan bahwa hak akses diberikan ke user tersebut adalah tingkat global. Klausa TO menspesifikasikan nama user (user1) dan nama host (domain1.com). Dalam klausa juga meliputi sub klausa IDENTIFIED BY, yang mendefinisikan password (pw1) pada akun user.

Dari pernyataan tersebut dapat disimpulkan, bahwa user1 dapat terkoneksi dari domain1.com, dan dapat melakukan semua aktivitas atau tugas-tugas pada semua database dan tabel kecuali dalam hal perijinan dan pencabutan hak akses. Dengan tambahan, user1 harus menyediakan password pw1 ketika log in ke MySQL server. Jika Anda ingin menampilkan bagaimana akun ini akan ditambahkan ke tabel user, Anda dapat menggunakan pernyataan SELECT berikut:

```
SELECT host, user, select_priv, update_priv FROM user WHERE user='user1';
```

Hasilnya:

```
+-------------+-------+-------------+-------------+
| host        | user  | select_priv | update_priv |
+-------------+-------+-------------+-------------+
| domain1.com | user1 | Y           | Y           |
+-------------+-------+-------------+-------------+
1 row in set (0.00 sec)
```

Contoh berikut lebih membatasi jangkauan akses ketimbang sebelumnya. Pernyataan GRANT memberikan hak akses SELECT dan UPDATE pada semua tabel dalam database test.

```
GRANT SELECT, UPDATE
ON test.*
TO 'user1'@'domain1.com' IDENTIFIED BY 'pw1';
```

Dikarenakan pernyataan GRANT spesifik ke database, Anda dapat menggunakan pernyataan SELECT berikut melihat bagaimana akun user akan ditambahkan ke tabel user dan db:

```
SELECT host, user, select_priv, update_priv FROM user WHERE user='user1';
SELECT host, db, user, select_priv, update_priv FROM db WHERE user='user1';
```

Hasilnya adalah:

```
+-------------+-------+-------------+-------------+
| host        | user  | select_priv | update_priv |
+-------------+-------+-------------+-------------+
| domain1.com | user1 | N           | N           |
+-------------+-------+-------------+-------------+
1 row in set (0.00 sec)

+-------------+------+-------+-------------+-------------+
| host        | db   | user  | select_priv | update_priv |
+-------------+------+-------+-------------+-------------+
| domain1.com | test | user1 | Y           | Y           |
+-------------+------+-------+-------------+-------------+
1 row in set (0.00 sec)
```

Berikut ini diasumsikan Anda telah membuat database TOKO dengan tabel BUKU (KodeBuku, JudulBuku, Copyright).

Contoh-contoh pernyataan:

```
GRANT SELECT, UPDATE
ON TOKO.BUKU
TO 'user1'@'domain1.com' IDENTIFIED BY 'pw1';
```

```
GRANT SELECT, UPDATE (JudulBuku, Copyright)
ON TOKO.BUKU
TO 'user1'@'domain1.com' IDENTIFIED BY 'pw1';
```

C2. Menggunakan pernyataan SHOW GRANTS

Pernyataan SHOW GRANTS menampilkan informasi spesifik dari akun user. Sintaksnya:

```
SHOW GRANTS FOR '<user>'@'<host>'
```

Contoh:

```
SHOW GRANTS FOR 'user1'@'domain1.com';
```

Hasilnya:

```
+-----------------------------------------------------------------------------+
| Grants for user1@domain1.com                                                |
+-----------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'user1'@'domain1.com' IDENTIFIED BY PASSWORD
'*2B602296A79E0A8784ACC5C88D92E46588CCA3C3' WITH MAX_QUERIES_PER_HOUR 50
MAX_UPDATES_PER_HOUR 50                                                     |
| GRANT SELECT, UPDATE ON 'test'.* TO 'user1'@'domain1.com' WITH GRANT OPTION |
+-----------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

C3. Mengatur password untuk akun user MySQL

Sintaks:

```
SET PASSWORD [FOR '<user>'@'<host>'] = PASSWORD('<new password>');
```

Contoh:

```
SET PASSWORD FOR 'user1'@'domain1.com' = PASSWORD('pw3');
```

C4. Menggunakan pernyataan FLUSH PRIVILEGES

Sintaks:

```
FLUSH PRIVILEGES;
```

Setelah Anda mengeksekusi pernyataan ini, tabel grant dipanggil dan informasi akun user baru segera dimuatkan atai di-*refresh*.

C5. Menggunakan pernyataan REVOKE

Pernyataan REVOKE mengijinkan Anda untuk menghapus hak-hak akses dari akun user.

Sintaks:

```
REVOKE ALL PRIVILEGES, GRANT OPTION
FROM '<user>'@'<host>' [{, '<user>'@'<host>'}...]
```

Contoh:

```
GRANT SELECT, UPDATE
ON test.*
TO 'user1'@'domain1.com' IDENTIFIED BY 'pw1'
WITH GRANT OPTION MAX_QUERIES_PER_HOUR 50 MAX_UPDATES_PER_HOUR 50;
```

```
REVOKE ALL PRIVILEGES, GRANT OPTION
FROM 'user1'@'domain1.com';
```

```
REVOKE SELECT, UPDATE, GRANT OPTION
ON test.*
FROM ‘user1’@’domain1.com’;
```

C6. Menggunakan pernyataan DROP USER

Sintaks:

```
DROP USER ‘<user>’@’<host>’ [{, ‘<user>’@’<host>’}...]
```

Contoh:

```
DROP USER ‘user1’@’domain1.com’;
```