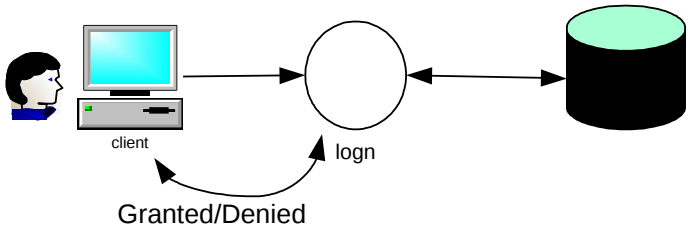
Konsep Security

Sebelum melakukan implementasi security di MySQL, perencanaan harus dilakukan lebih dahulu:

- ❑ Daftar aktifitas yang harus dikendalikan
- ❑ Pemakai secara Individual atau Kelompok yang akan bekerja
- ❑ Daftar pemakai, siapa dan mengakses apa (aktifitas apa yang diijinkan)

Setiap pemakai mengidentifikasikan dirinya dengan menggunakan **nama, password** dan **nama host.** Nama user **tidak mempunyai relasi** dengan nama user di operating system**.**



MySQL menyimpan user-account pada tabel mysql.user , setiap kali proses login, atau akses database, maka mysql memeriksa tabel tersebut. Bila user mempunyai 'credential' , maka user diijinkan untuk mengoperasikan instruksi tersebut.

Instruksi berikut akan menampilkan struktur user dalam MySQL:

```
> SHOW CREATE TABLE mysql.user;
```

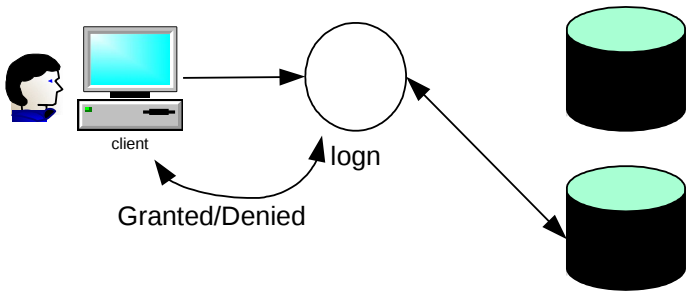
Kolom pada tabel yang penting adalah spesifik ke nama account dan informasi otentikasi.

Nama User dan Host mengidentifikasikan akun (account).
Contoh : 'badu'@'localhost'

Mengidentifikasikan akun 'badu' pada localhost.
'hasan@'inixindo.id' mengidentifikasikan user hasan yang berasal dari host inixindo.id.

Langkah berikut adalah memeriksa apakah pemakai tersebut diijinkan untuk akses dari komputer (host) dimana pemakai tersebut berada.

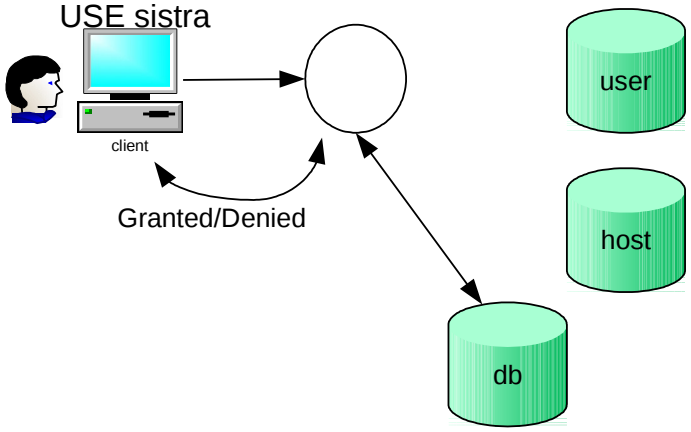
Untuk itu mysql akan memeriksa tabel host.



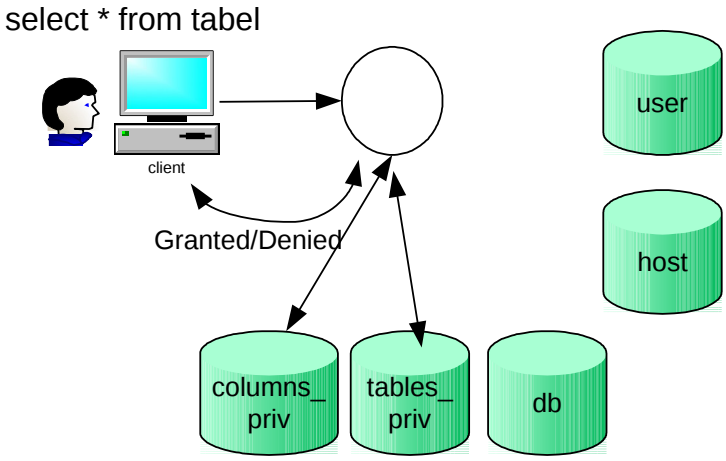
Bila nama host dan mana user terdaftar, maka pemakai tersebut akan mendapatkan mysql prompt.

Berikutnya pemakai tersebut akan memilih salah satu database dengan instruksi USE database.

MySQL akan memeriksa ijin akses pada tabel db, apakah akses diijinkan atau tidak.



Setelah diijinkan untuk menggunakan database, langkah lanjut adalah mengakses TABEL dan KOLOM pada Tabel.



GRANT / REVOKE

MySQL memberikan ijin akses melalui instruksi GRANT dan mencabutnya dengan REVOKE.

user
Berisi daftar pemakai yang dapat berhubungan dengan MySQL Server (connect), beserta global privileges yang dimilikinya.

```
CREATE TABLE user (
`Host` char(60) COLLATE utf8_bin NOT NULL DEFAULT '',
  `User` char(80) COLLATE utf8_bin NOT NULL DEFAULT '',
  `Password` char(41) CHARACTER SET latin1 COLLATE latin1_bin NOT NULL
DEFAULT '',
  `Select_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Insert_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Update_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Delete_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Create_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Drop_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Reload_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Shutdown_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Process_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `File_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Grant_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `References_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Index_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Alter_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
`Show_db_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Super_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Create_tmp_table_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL
DEFAULT 'N',
  `Lock_tables_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT
'N',
  `Execute_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Repl_slave_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Repl_client_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT
'N',
  `Create_view_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT
'N',
  `Show_view_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT 'N',
  `Create_routine_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT
'N',
  `Alter_routine_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT
'N',
  `Create_user_priv` enum('N','Y') CHARACTER SET utf8 NOT NULL DEFAULT
'N',
+-----------------------------------------+
  . . . .
  PRIMARY KEY  (Host,User)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='Users and
global privileges';
```

Pasca instalasi tabel user mempunyai nama account kosong (any-user) sehingga orang dapat login tanpa nama user dan tanpa password.
Perhatikan bahwa MySQL mempunyai **notasi ' '** yang berarti adalah '**any-user'** (**anonymous user**).

```
> select user,host,password from mysql.user where user='' ;
+------+-----------------------+----------+
| user | host                  | password |
+------+-----------------------+----------+
|      | localhost             |          |
|      | localhost.localdomain |          |
+------+-----------------------+----------+
```

Pada sistem ini masih terlihat anonymous user yang dapat login pada host tanpa menggunakan password. Jika pada host terlihat tanda ″%″, notasi **%** berarti **'any-host'** .

Selanjutnya dicoba untuk login tanpa mensuplai password.

```
$ mysql -h localhost
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 5.5.43-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and
others.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

MariaDB [(none)]>
```

Berhasil!!!

Walauun akun root sudah diberikan password, tapi anonymous user tetap ada, dan berpotensi untuk melakukan malfunction. Oleh karena itu, hapus User ini dari sistem.

```
> use test
Database changed
>
```

Berhasil! Anonymous user dapat membuka database test.

```
> use apicta;
ERROR 1044 (42000): Access denied for user ''@'localhost' to
database 'apicta'
```

Gagal!. Karena user '' anonymous tidak mempunyai privilege atas database apicta (Administrator tidak memberikan credential GRANT ON).

Bila dilihat dari tabel db, maka nyata bahwa ijin akses ke database dari **% (any-host)** hanya diijinkan ke database **test.**

```
$ mysql -u root -p
Enter password:

> select host, db, user from mysql.db;
+-----------+-----------------+--------+
| host      | db              | user   |
+-----------+-----------------+--------+
| %         | dataindo_inix23 | inix23 |
| %         | test            |        |←
| %         | test\_%         |        |
| localhost | dataindo_inix23 | inix23 |
+-----------+-----------------+--------


> desc mysql.db;
+-----------------------+---------------+------+-----+---------+-------+
| Field                 | Type          | Null | Key | Default | Extra |
+-----------------------+---------------+------+-----+---------+-------+
| Host                  | char(60)      | NO   | PRI |         |       |
| Db                    | char(64)      | NO   | PRI |         |       |
| User                  | char(16)      | NO   | PRI |         |       |
| Select_priv           | enum('N','Y') | NO   |     | N       |       |
| Insert_priv           | enum('N','Y') | NO   |     | N       |       |
| Update_priv           | enum('N','Y') | NO   |     | N       |       |
| Delete_priv           | enum('N','Y') | NO   |     | N       |       |
| Create_priv           | enum('N','Y') | NO   |     | N       |       |
| Drop_priv             | enum('N','Y') | NO   |     | N       |       |
| Grant_priv            | enum('N','Y') | NO   |     | N       |       |
| References_priv       | enum('N','Y') | NO   |     | N       |       |
| Index_priv            | enum('N','Y') | NO   |     | N       |       |
| Alter_priv            | enum('N','Y') | NO   |     | N       |       |
| Create_tmp_table_priv | enum('N','Y') | NO   |     | N       |       |
| Lock_tables_priv      | enum('N','Y') | NO   |     | N       |       |
| Create_view_priv      | enum('N','Y') | NO   |     | N       |       |
| Show_view_priv        | enum('N','Y') | NO   |     | N       |       |
| Create_routine_priv   | enum('N','Y') | NO   |     | N       |       |
| Alter_routine_priv    | enum('N','Y') | NO   |     | N       |       |
| Execute_priv          | enum('N','Y') | NO   |     | N       |       |
| Event_priv            | enum('N','Y') | NO   |     | N       |       |
| Trigger_priv          | enum('N','Y') | NO   |     | N       |       |
+-----------------------+---------------+------+-----+---------+-------+
22 rows in set (0.02 sec)
```

Dari tabel db dijelaskan ijin akses untuk melakukan **select, insert, update, create, drop, grant, index** dan **alter** . Ijin akses secara default adalah semuanya **NO**. Jika nama database tidak terdaftar disitu, maka privilege lain akan dilihat (dari tabel dan kolom).

Siapa saja nama account /login yang terdaftar?

```
> select host, user,password from mysql.user;
+-----------------------+--------+-------------------------------------+
| host                  | user   | password
|
+-----------------------+--------+-------------------------------------+
| localhost             | root   | *91EE7494F9DE8AF85CD7E68599F44A8230F |
| localhost.localdomain | root   |                                     |
| 127.0.0.1             | root   |                                     |
| ::1                   | root   |                                     |
| localhost             |        |                                     |
| localhost.localdomain |        |                                     |
| localhost             | inix23 | *24595DDBCB2A0EFE6EF7D9C1131BCED4661 |
| %                     | inix23 | *24595DDBCB2A0EFE6EF7D9C1131BCED4661 |
+-----------------------+--------+-------------------------------------+
8 rows in set (0.00 sec)
```

Perhatikan bahwa MySQL memeriksa tabel dari atas kebawah satu persatu.

Bila seseorang login tanpa menggunakan nama user maka identik dengan berikut:

```
Nama Host = localhost
Nama User = ' '  (anonymous)
Password  =      (tanpa password)
```

MySQL akan membandingkan nama host lebih dahulu, bila match baru nama user.

Pada contoh, maka proses login berhasil, karena match dengan baris ke tiga.

Dibaris ketiga, MySQL menemukan any-user dari localhost, dan password tidak ada, oleh karena itu pencarian selesai, akses diberikan kepada pemakai tersebut.

Ijin Akses (Privileges)

Ijin akses diberikan kepada user melalui instruksi GRANT untuk mendapatkan otorisasi menggunakan hal-hal berikut:

ALTER

Mengubah tabel dan index

CREATE

Menciptakan database dan tabel

DELETE

Menghapus record dari tabel

DROP

Menghapus tabel dan database

INDEX

Membuat dan menghapus index

INSERT

Memasukan baris baru di tabel

SELECT

Membaca tabel

UPDATE

meremajakan tabel

REFERENCES

(tidak digunakan)

Beberapa ijin akses yang berhubungan dengan Administrasi:

FILE

Membaca dan menulis File di Server

PROCESS

Melihat informasi tentang threads

RELOAD

Membaca kembali tabel privileges bila telah diadakan perubahan (identik dengan FLUSH PRIVILEGES).

SHUTDOWN

Melakukan shutdown via `mysqladmin shutdown`

ALL

Sebagai singkatan atau sinonim yang berarti adalah semua privileges.

MySQL melakukan proses matching pertama dengan tabel users, kemudian db (database), dan host .

```
> desc mysql.host;
+-----------------------+---------------+------+-----+---------+-------+
| Field                 | Type          | Null | Key | Default | Extra |
+-----------------------+---------------+------+-----+---------+-------+
| Host                  | char(60)      | NO   | PRI |         |       |
| Db                    | char(64)      | NO   | PRI |         |       |
| Select_priv           | enum('N','Y') | NO   |     | N       |       |
| Insert_priv           | enum('N','Y') | NO   |     | N       |       |
| Update_priv           | enum('N','Y') | NO   |     | N       |       |
| Delete_priv           | enum('N','Y') | NO   |     | N       |       |
| Create_priv           | enum('N','Y') | NO   |     | N       |       |
| Drop_priv             | enum('N','Y') | NO   |     | N       |       |
| Grant_priv            | enum('N','Y') | NO   |     | N       |       |
| References_priv       | enum('N','Y') | NO   |     | N       |       |
| Index_priv            | enum('N','Y') | NO   |     | N       |       |
| Alter_priv            | enum('N','Y') | NO   |     | N       |       |
| Create_tmp_table_priv | enum('N','Y') | NO   |     | N       |       |
| Lock_tables_priv      | enum('N','Y') | NO   |     | N       |       |
| Create_view_priv      | enum('N','Y') | NO   |     | N       |       |
| Show_view_priv        | enum('N','Y') | NO   |     | N       |       |
| Create_routine_priv   | enum('N','Y') | NO   |     | N       |       |
| Alter_routine_priv    | enum('N','Y') | NO   |     | N       |       |
| Execute_priv          | enum('N','Y') | NO   |     | N       |       |
| Trigger_priv          | enum('N','Y') | NO   |     | N       |       |
+-----------------------+---------------+------+-----+---------+-------+
20 rows in set (0.00 sec)
```

Kondisi awal dari tabel host adalah sebagai berikut:

```
> select * from mysql.host;
Empty set (0.03 sec)
```

Tabel ini kosong, menandakan bahwa ijin akses tidak diberikan berdasarkan nama host.

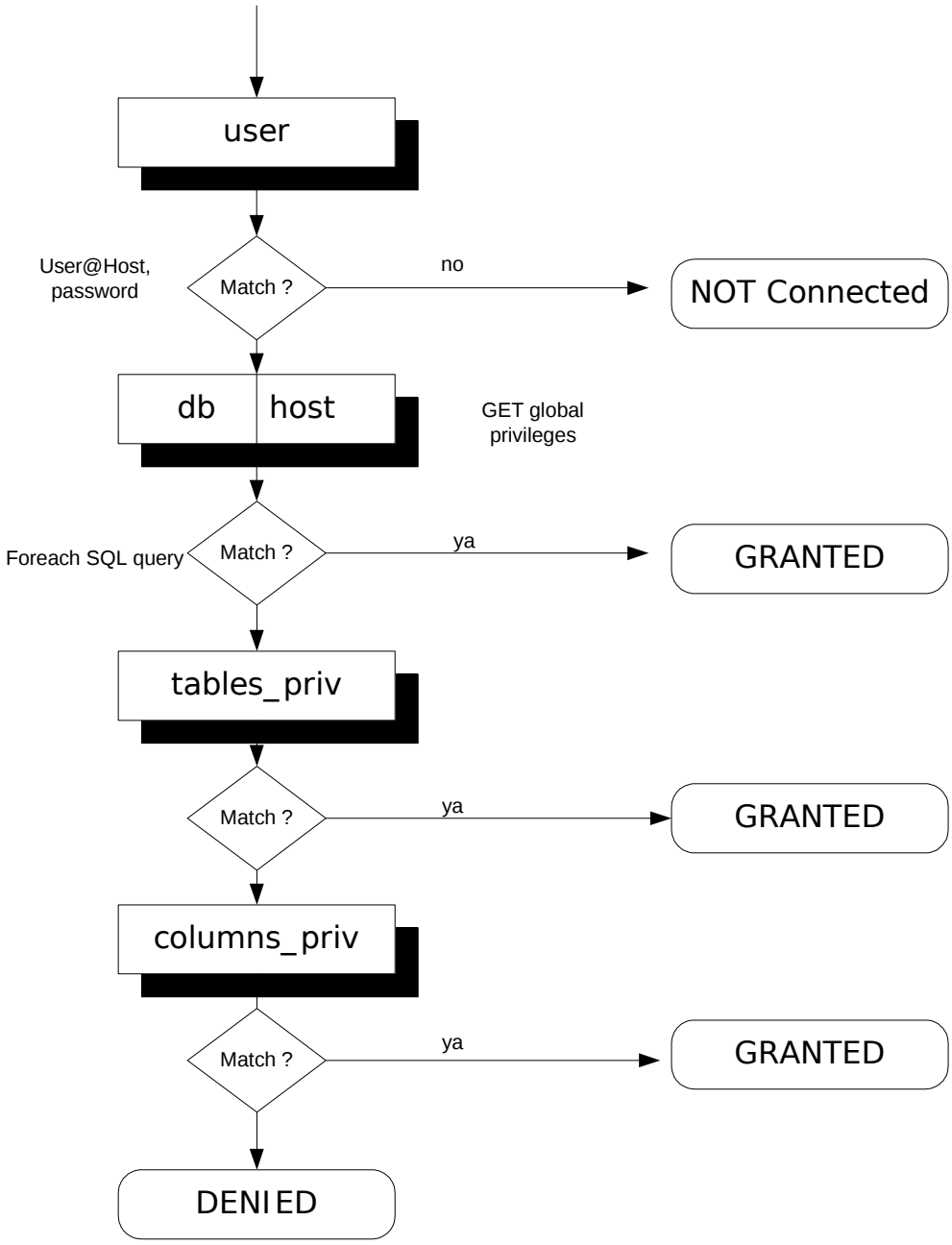
Catatan:
Tabel host digunakan secara kombinasi dengan tabel db untuk mengendalikan akses ke database untuk spesifik host. Tabel

host tidak terpengaruh dengan instruksi GRANT dan REVOKE, karena itu umumnya **tidak mempunyai peran** yang berarti.

Tabel host hanya akan diperiksa, bila kolom host pada tabel **db** kosong (blank).

Proses Pemberian Ijin Akses

Ijin Akses diberikan berdasarkan match nama host, nama user dan password, memandingkan ketiga parameter tersebut dengan isi tabel beikur:



Verifikasi Akses

Setiap kali seorang pemakai memberikan instruksi (query), maka MySQL akan memeriksa, apakah query tersebut dapat dipenuhi sesuai dengan ijin akses yang telah diberikan.

Secara tersusun, pemeriksaan dilakukan mulai dari tabel **`user, db, tables_priv`** dan terakhir **`columns_priv`**.

MySQL memeriksa tabel user untuk mendapatkan global privileges. Jika memenuhi syarat, maka query dilaksanakan.

Jika global privileges dari tabel user tidak memadai, maka MySQL memeriksa table db (dan host), dan menambahkannya sebagai global privileges. Bila cukup, maka query tersebut dilaksanakan.

Jika masih tidak mencukupi, maka MySQL memeriksa tables_priv , dan bila belum memadai, columns_priv.

Setelah yang terakhir belum juga cukup, maka query tersebut ditolak.

Secara ALGORITMA dapat diikuti sebagai berikut:

```
IF (table user) THEN OK
ELSE IF (table db AND host) THEN OK
      ELSE IF(tables_priv) THEN OK
         ELSE IF (columns_priv) THE OK
                ELSE DENY.
```

Membuat Akun Pemakai

CREATE USER digunakan untuk menciptakan user dan akan meremajakan tabel **mysql.user** table. Sebelumnya harus ditentukan dulu 3 bentuk akun:

• Nama akun tertulis dalam format *'namapemakai'@'host_name'* yaitu menentukan user dan host, dari mana user tersebut aakan melakukan koneksi.

MySQL mempunyai opsi berupa beberapa authentication plug-in yang akan dieksekusi jika client berusaha untuk login.

• mysql_native_password menggunakan metoda hasing
• mysql_old_password serupa dengan hashing tapi tidak sebaik native (deprecated).
• sha256_password hash cryptographically lebih aman ketimbang mysql_native_pass tapi lebih kompleks,

```
CREATE USER 'user_name'@'host_name' IDENTIFIED BY 'password';

CREATE USER 'user_name'@'host_name' IDENTIFIED WITH
'auth_plugin';
```

Secara default authentication-plugin setting adalah mysql_native_password .

Contoh:
1. Buat akun CREATE USER dengan otentikasi plugin secara eksplisit. Ubah variable sistem old_passwords untuk memilih password hashing yang akan digunakan oleh fungsi PASSWORD().

```
CREATE USER 'user_name'@'host_name' IDENTIFIED WITH
'mysql_native_password';
SET old_passwords = 0;

CREATE USER 'user_name'@'host_name' IDENTIFIED WITH
'mysql_old_password';
SET old_passwords = 1;

CREATE USER 'user_name'@'host_name' IDENTIFIED WITH
'sha256_password';
SET old_passwords = 2;
```

Plugin sha256_password harus diinstalasi lebih dahulu, Client harus melakukan koneksi SSL atau menggunakan RSA certificate.

2. Mengubah password

```
SET PASSWORD FOR 'user_name'@'host_name' =
PASSWORD('password');
```

Fungsi PASSWORD() melakukan hash sesuai dengan nilai old_passwords yang diberikan.

Mengubah password dapat dilakukan dengan SET PASSWORD FOR atau dengan langsung mengupdate tabel mysql.user (tidak direkomendasikan!).

```
> select host,user,password from mysql.user where
user='dastan';
+-----------+--------+-------------------------------------+
| host      | user   | password                            |
+-----------+--------+-------------------------------------+
| localhost | dastan |                                     |
+-----------+--------+-------------------------------------+

> update mysql.user set password=Password("inixindo") where
user="dastan";
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

> select host,user,password from mysql.user where
user='dastan';
+-----------+--------+-------------------------------------+
| host      | user   | password                            |
+-----------+--------+-------------------------------------+
| localhost | dastan | *91EE7494F9DE8AF85CD7E685E9F44A8230F |
+-----------+--------+-------------------------------------+
```

Hati-hati ! Kunci kata **WHERE** harus diberikan, bila tidak maka password untuk semua user akan diganti, dan ini merepotkan Administrator untuk menata ulang kembali.

Karena itu gunakan instruksi SET PASSWORD FOR !

Fungsi **Password( )** mutlak harus digunakan, karena fungsi ini melakukan **enkripsi** yang dimengerti oleh semua platform yang menjalankan MySQL .

Memberikan ijin akses:

Ijin akses diberikan dengan instruksi seperti berikut:

```
GRANT privileges (kolom)
   ON namaTabel
   TO pemakai IDENTIFIED BY 'password'
```

```
   WITH GRANT OPTION ;
```

Contoh:

```
GRANT ALL ON hrd.pegawai
   TO dastan@localhost ;
```

kolom
Nama kolom secara sepesifik pada tabel (opsi).

namaTabel
Rincian dari nama tabel yang akan diijinkan. Bila mengijinkan satu database, maka gunakan tanda *, seperti : hrd.*

```
GRANT ALL ON hrd.*
   TO dastan@localhost ;
```

pemakai
Adalah nama account yang diberikan ijin akses tersebut. Perhatikan bahwa bila nama mesin tidak diberikan, maka MySQL akan menambahkan kata string **'%'** yang berarti 'any-host'.

```
GRANT ALL ON hrd.*
   TO dastan;

GRANT ALL ON hrd.*
   TO dastan@localhost ;

GRANT ALL ON hrd.*
   TO dastan@sadewa.inix.com;

GRANT ALL ON hrd.*
   TO dastan@%.inix.com ;

GRANT ALL ON hrd.*
   TO dastan@192.168.1.55 ;

GRANT ALL ON hrd.*
   TO dastan@192.168.1.0/24 ;

GRANT ALL ON hrd.*
   TO dastan@% ;
```

Contoh Grant privilege:

Memberikan global privilege untuk akun administratif pada semua database:

```
GRANT FILE ON *.* TO 'dastan'@'localhost';
GRANT CREATE TEMPORARY TABLES, LOCK TABLES ON *.* TO
'dastan'@'localhost';
```

Memberikan privileges pada level database yang memungkinkan akun melakukan akses pada object di database hrd:

```
GRANT ALL ON hrd.* TO 'dastan'@'localhost';
```

Memberikan privilege pada level table agar dapat diakses dengan SELECT:

```
GRANT SELECT ON apicta.judges TO 'dastan'@'localhost';
```

Memberikan privileges pada level kolom:

```
GRANT SELECT(nama,score), UPDATE(score)
ON apicta.products TO 'dastan'@'localhost';
```

Memberikan privilege pada level procedure calc( ) :

```
GRANT EXECUTE ON PROCEDURE hrd.calc TO 'dastan'@'localhost';
```

Privileges dapat dilihat sebagai berikut:

```
> SHOW GRANTS FOR 'dastan'@'localhost';
+-----------------------------------------------------------+
| Grants for dastan@localhost                               |
+-----------------------------------------------------------+
| GRANT FILE, CREATE TEMPORARY TABLES, LOCK TABLES ON *.* TO
'dastan'@'localhost'
| GRANT ALL PRIVILEGES ON `apicta`.* TO 'dastan'@'localhost'
| GRANT SELECT ON `apicta`.`judges` TO 'dastan'@'localhost'
| GRANT SELECT (score, nama), UPDATE (score) ON
`apicta`.`products` TO 'dastan'@'localhost'
+-----------------------------------------------------------+
4 rows in set (0.01 sec)
```

Mencabut Ijin Akses

REVOKE digunakan untuk mencabut ijin akses yang telah diberikan melalui GRANT.

```
REVOKE privileges (kolom)
ON namaTabel
FROM pemakai ;
```

Nama pemakai harus sama, nama privilege boleh berbeda pada saat GRANT. REVOKE dapat mencabut semua atau sebagian *privilege* yang telah diberikan oleh GRANT.

```
REVOKE ALL ON hrd.*
   FROM dastan@sadewa.inix.com;
```

Bila tidak sesuai, maka akan muncul pesan error seperti berikut:

```
ERROR 1141: There is no such grant defined for user 'dastan'
on host 'sadewa.inix.com'
```

```
> REVOKE SELECT,SHUTDOWN ON hrd.* FROM alicia@localhost;
```

REVOKE tidak menghapus nama pemakai dari tabel user. Untuk menghilangkan nama user gunakan DELETE FROM user, kemudian FLUSH PRIVILEGES.

Contoh Revoke:

```
> REVOKE FILE ON *.* FROM 'dastan'@'localhost';
> REVOKE CREATE TEMPORARY TABLES, LOCK TABLES
-> ON *.* FROM 'dastan'@'localhost';
> REVOKE ALL ON cookbook.* FROM 'dastan'@'localhost';
> REVOKE SELECT ON mysql.user FROM 'dastan'@'localhost';
> REVOKE SELECT(User,Host), UPDATE(password_expired)
-> ON mysql.user FROM 'dastan'@'localhost';
> REVOKE EXECUTE ON PROCEDURE cookbook.exec_stmt
-> FROM 'dastan'@'localhost';
> SHOW GRANTS FOR 'dastan'@'localhost';
+-----------------------------------------+
| Grants for dastan@localhost
|
+-----------------------------------------+
| GRANT USAGE ON *.* TO 'dastan'@'localhost' |
+-----------------------------------------+
```

Menghapus Akun

Gunakan DROP USER untuk menghapus akun:

```
DROP USER 'user1'@'localhost';
```

Instruksi ini menghapus semua obyek yang berikaitan dengan user tersebut, semua privileges, dan lainnya.

Nama Host perlu dicantumkan, kecuali bila ingin dihapus seluruh nama host yang ada.

FLUSH PRIVILEGES diperlukan untuk membaca ulang tabel privileges oleh MySQL, karena MySQL menggunakan CACHE, maka proses tersebut perlu dilakukan, jika tidak, isi CACHE tidak sama dengan kondisi data di file.

Mengubah nama akun, lakukan dengan RENAME USER , dan spesifikasikan nama baru.

```
RENAME USER 'user1'@'localhost' TO 'user2'@'localhost';
```

Pesan error muncul jika akun tersebut tidak ada.