

Pocket Book MySQL



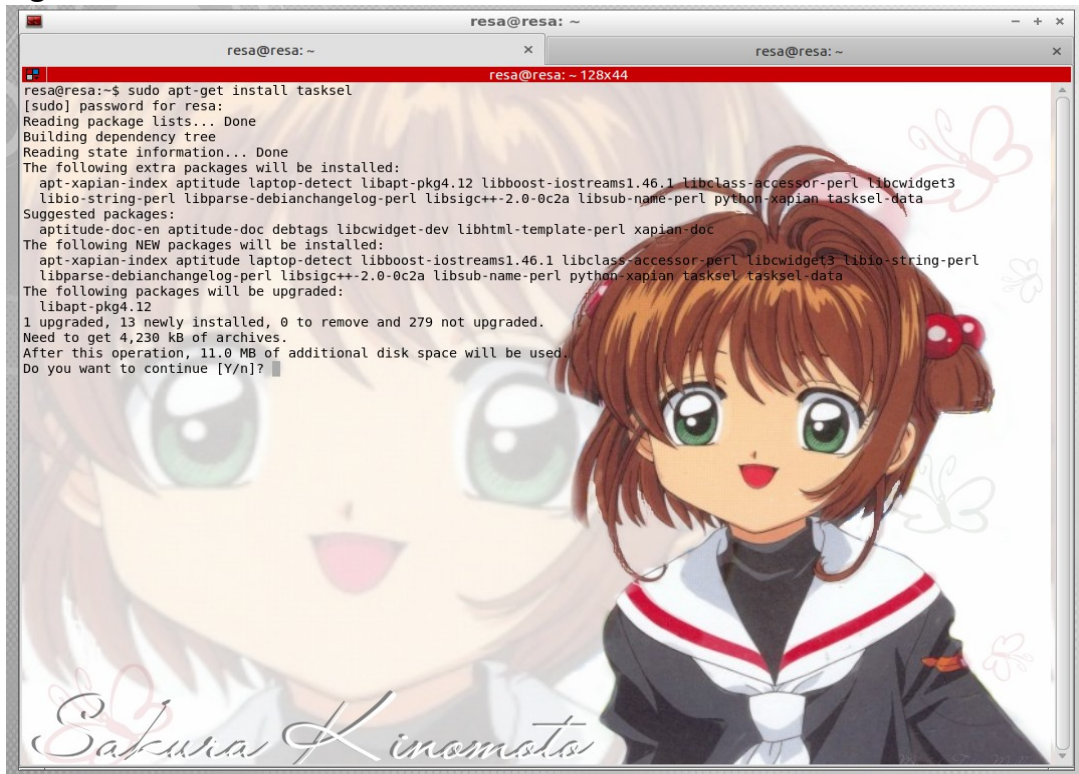
Author : Resa C.R

Founder and Chief www.marisharingilmu.wordpress.com

Lisensi Pocket Book

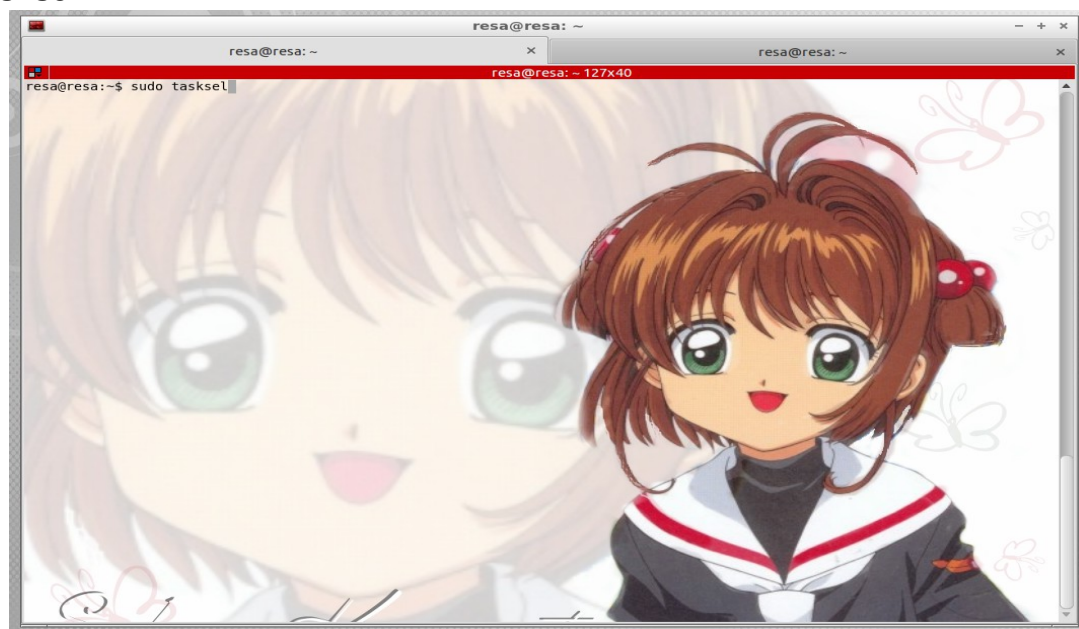
Pocket Book ini dapat anda distribusikan ulang,dicopy,dan dicetak untuk pembelajaran.

Sebelum memulai tutorial MySQL,anda terlebih dahulu install MySQL di Lubuntu.Kali ini saya menggunakan LAMP.**LAMP** adalah kependekan dari **Linux,Apache,MySQL, dan Php**.Di dalam LAMP sudah terdapat software lengkap untuk pemrograman.**Apache** digunakan sebagai web server.**MySQL** adalah sebagai DBMS dan **Php** adalah bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis web.Untuk memudahkan install LAMP saya menggunakan Tasksel.**Tasksel** adalah sebuah paket software yang cukup lengkap.Dengan Tasksel anda dapat melakukan instalasi software dengan mudah seperti LAMP, Mail Server, DNS Server, dsb.Langsung saja saya berikan tutorial install LAMP dengan Tasksel.
Buka terminal dan ketikkan
sudo apt-get install tasksel



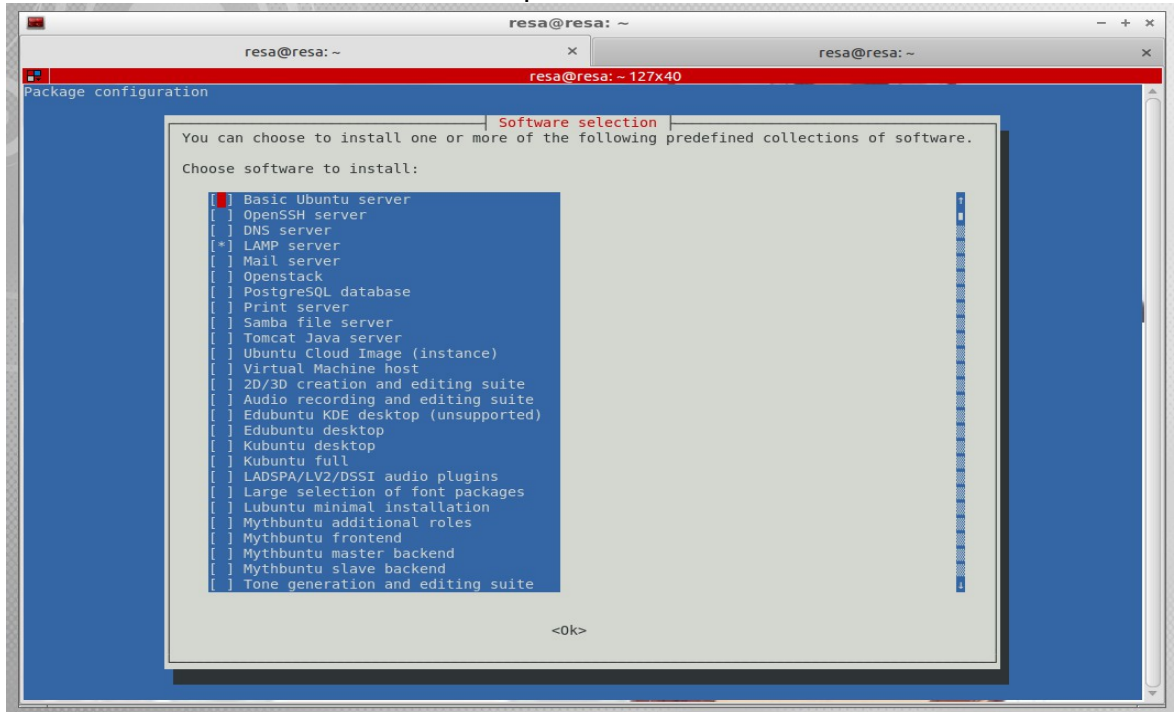
```
resa@resa:~$ sudo apt-get install tasksel
[sudo] password for resa:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apt-xapian-index aptitude laptop-detect libapt-pkg4.12 libboost-iostreams1.46.1 libclass-accessor-perl libcwid3
  libio-string-perl libparse-debianchangelog-perl libsigc++2.0-0c2a libsub-name-perl python-xapian tasksel-data
Suggested packages:
  aptitude-doc-en aptitude-doc debtags libcwidget-dev libhtml-template-perl xapian-doc
The following NEW packages will be installed:
  apt-xapian-index aptitude laptop-detect libboost-iostreams1.46.1 libclass-accessor-perl libcwid3 libio-string-perl
  libparse-debianchangelog-perl libsigc++2.0-0c2a libsub-name-perl python-xapian tasksel tasksel-data
The following packages will be upgraded:
  libapt-pkg4.12
1 upgraded, 13 newly installed, 0 to remove and 279 not upgraded.
Need to get 4,230 kB of archives.
After this operation, 11.0 MB of additional disk space will be used.
Do you want to continue [Y/n]? 
```

Untuk menjalankan Tasksel ketikkan
sudo tasksel



```
resa@resa:~$ sudo tasksel
```

Setelah itu anda akan dibawa ke menu seperti berikut

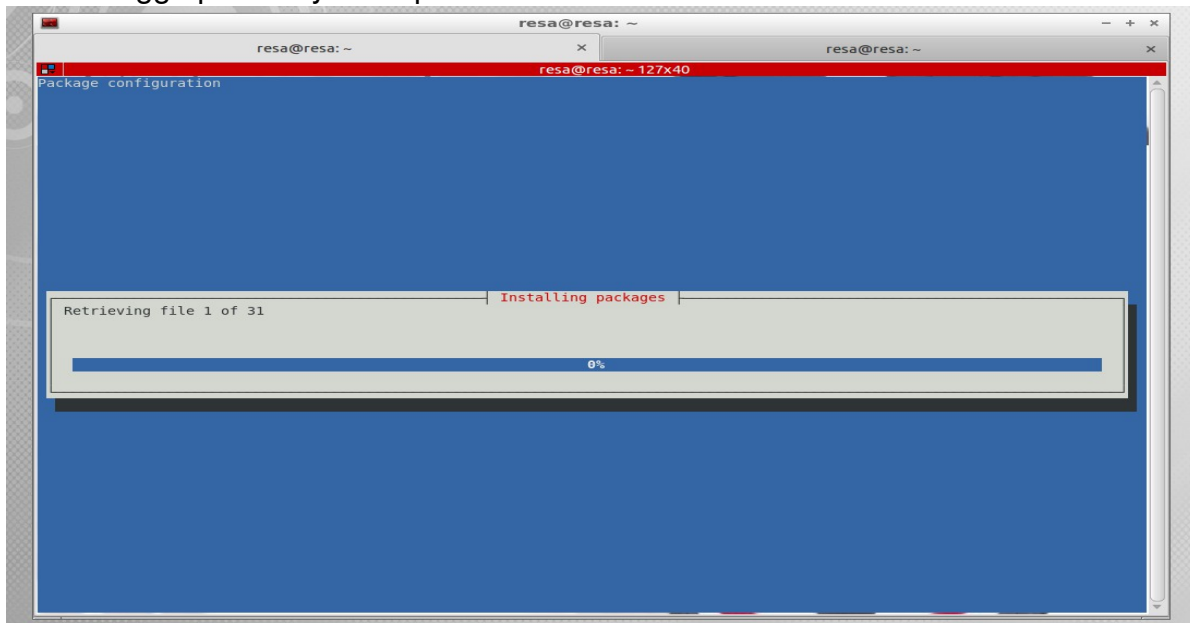


Pilih **LAMP Server** ,dengan cara menekan **tombol spasi** untuk memilih software yang akan diinstall dilanjutkan dengan menekan **tombol Tab** untuk memilih ok.

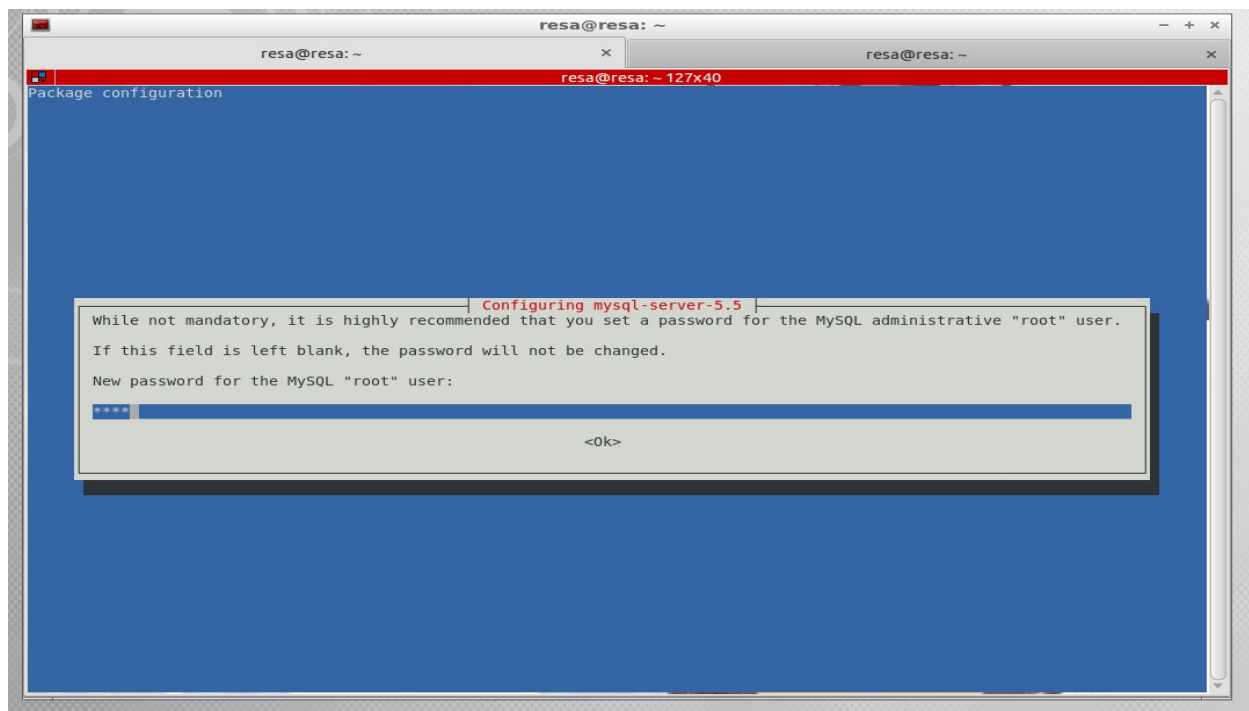
Bisa juga dengan cara mengetikkan perintah berikut ini

sudo taskel install lamp-server

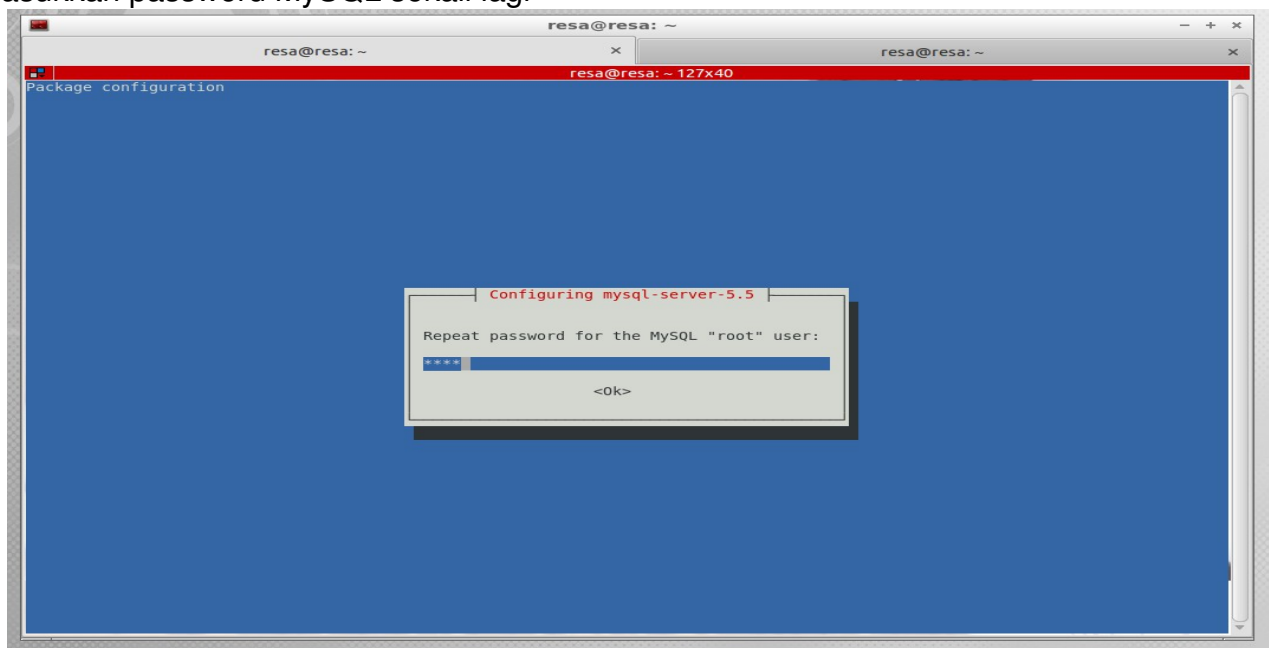
Kemudian tunggu prosesnya sampai selesai



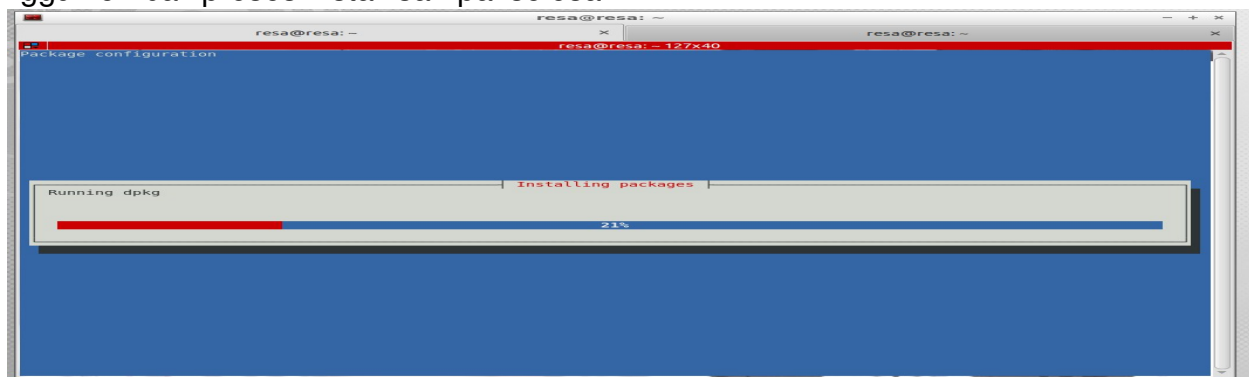
Kalau sudah masukkan password MySQL



Masukkan password MySQL sekali lagi

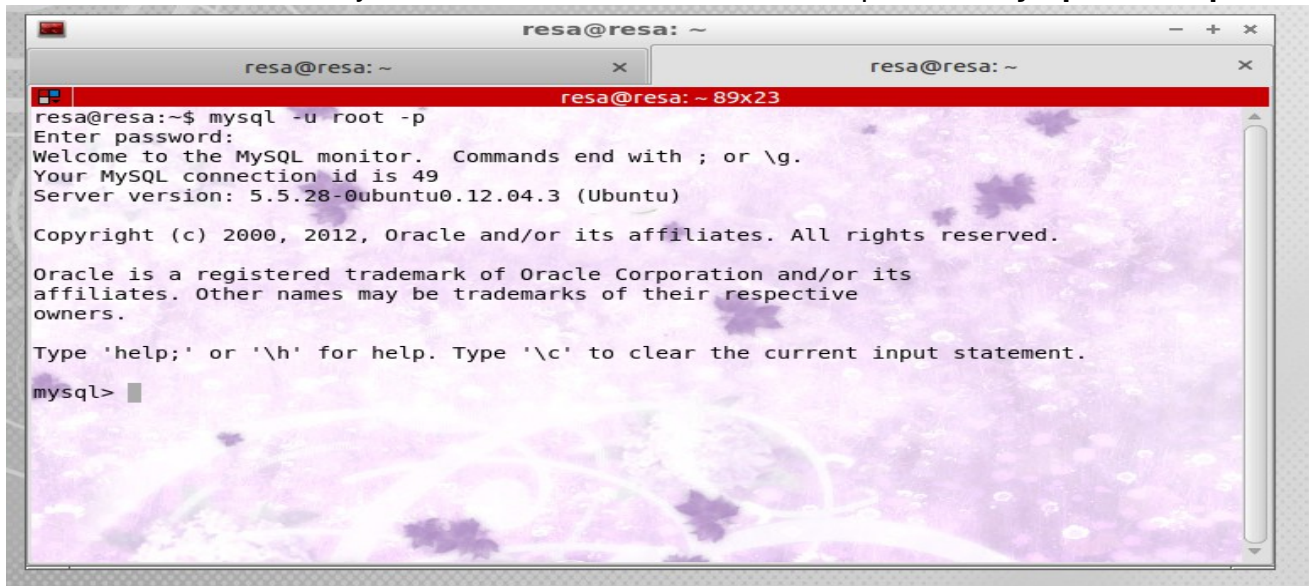


Tunggu kembali proses install sampai selesai.



Proses install selesai.

Untuk masuk ke dalam MySQL, buka terminal dan masukkan perintah **mysql -u root -p**



```
resa@resa: ~  
resa@resa:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 49  
Server version: 5.5.28-0ubuntu0.12.04.3 (Ubuntu)  
  
Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

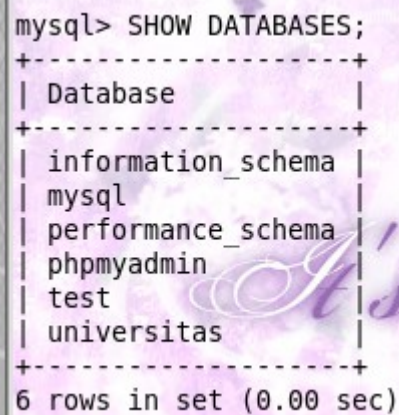
Langkah pertama adalah membuat database di MySQL. Pada contoh kali ini, membuat database dengan nama **universitas**.

```
CREATE DATABASE universitas;
```

```
mysql> CREATE DATABASE universitas;  
Query OK, 1 row affected (0.00 sec)
```

Untuk melihat database yang sudah pernah dibuat ketikkan

```
SHOW DATABASES;
```



```
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| phpmyadmin |  
| test |  
| universitas |  
+-----+  
6 rows in set (0.00 sec)
```

Langkah selanjutnya setelah membuat database adalah membuat table. Sebelum membuat table anda harus masuk ke dalam database dulu. Kalau diibaratkan database adalah sebuah rumah maka untuk masuk ke dalam rumah tersebut membutuhkan kunci. Nah untuk masuk kedalam database menggunakan perintah

```
USE universitas;
```

```
mysql> USE universitas;  
Database changed  
mysql>
```

Pada contoh ini saya membuat table dengan nama **mahasiswa**. Dengan Field **nim,nama,alamat,jurusan**. Nim saya buat menjadi **primary key**. **Primary Key** digunakan agar tidak ada data dengan nim sama yang disimpan, artinya nim tersebut bersifat unique/hanya ada satu.

```
CREATE TABLE mahasiswa  
(  
  nim int(10) primary key,  
  nama varchar(25) not null,  
  alamat varchar(35) not null,  
  jurusan varchar(20) not null  
);
```

```
mysql> CREATE TABLE mahasiswa  
-> (  
-> nim int(10) primary key,  
-> nama varchar(25) not null,  
-> alamat varchar(35) not null,  
-> jurusan varchar(20) not null  
-> );  
Query OK, 0 rows affected (0.31 sec)
```

Untuk melihat struktur table gunakan perintah

```
DESCRIBE mahasiswa;
```

atau

```
DESC mahasiswa;
```

```
mysql> DESC mahasiswa;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| nim   | int(10)       | NO   | PRI | NULL    |       |  
| nama  | varchar(25)   | NO   |     | NULL    |       |  
| alamat | varchar(35)   | NO   |     | NULL    |       |  
| jurusan | varchar(20)   | NO   |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> DESCRIBE mahasiswa;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| nim   | int(10)       | NO   | PRI | NULL    |       |  
| nama  | varchar(25)   | NO   |     | NULL    |       |  
| alamat | varchar(35)   | NO   |     | NULL    |       |  
| jurusan | varchar(20)   | NO   |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

Untuk mengubah struktur table dapat menggunakan perintah **ALTER TABLE**, seperti berikut

- Menambah Field

```
ALTER TABLE namaTable ADD namaField;
```

- Menambahkan Primary Key

```
ALTER TABLE namaTable ADD PRIMARY KEY namaField;
```

- Merubah Field

```
ALTER TABLE namaTable CHANGE namaField_yangDiubah namaField_baru;
```

- Menghapus Field

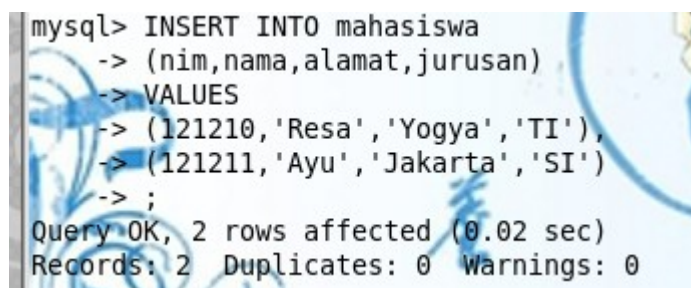
```
ALTER TABLE namaTable DROP namaField;
```

- Mengubah nama table

```
ALTER TABLE namaTable RENAME TO namaTabel_baru;
```

Setelah membuat table, langkah selanjutnya adalah mengisi data/record ke dalam field – field yang telah dibuat tadi. Untuk insert record ke dalam field gunakan perintah

```
INSERT INTO mahasiswa  
(nim,nama,alamat,jurusan)  
VALUES  
(121210,'Resa','Yogya','TI'),  
(121211,'Ayu','Jakarta','SI')  
;
```



```
mysql> INSERT INTO mahasiswa  
-> (nim,nama,alamat,jurusan)  
-> VALUES  
-> (121210,'Resa','Yogya','TI'),  
-> (121211,'Ayu','Jakarta','SI')  
-> ;  
Query OK, 2 rows affected (0.02 sec)  
Records: 2 Duplicates: 0 Warnings: 0
```

untuk melihat record ketikkan perintah

```
SELECT * FROM mahasiswa;
```



```
mysql> SELECT * FROM mahasiswa;
```

nim	nama	alamat	jurusan
121210	Resa	Yogya	TI
121211	Ayu	Jakarta	SI

```
2 rows in set (0.00 sec)
```

Jika ingin melakukan update data/record. Menggunakan perintah **UPDATE**. Contohnya pada field **jurusan** dengan record **TI** akan saya rubah menjadi **SI**.

```
UPDATE mahasiswa SET jurusan='SI' WHERE nim=121210;
```

```
mysql> UPDATE mahasiswa SET jurusan='SI' WHERE nim = 121210
-> ;
Query OK, 1 row affected (0.06 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

Sekarang jurusan telah berganti menjadi **SI**

```
mysql> SELECT * FROM mahasiswa;
```

nim	nama	alamat	jurusan
121210	Resa	Yogya	SI
121211	Ayu	Jakarta	SI

```
2 rows in set (0.00 sec)
```

Untuk menghapus record dapat menggunakan perintah **DELETE**. Contohnya saya akan menghapus record field **nim** dengan data **121210**.

```
DELETE FROM mahasiswa WHERE nim=121210;
```

```
mysql> DELETE FROM mahasiswa WHERE nim=121210;
Query OK, 1 row affected (0.02 sec)
```

Data dengan **nim 121210** berhasil dihapus.

```
mysql> SELECT * FROM mahasiswa;
```

nim	nama	alamat	jurusan
121211	Ayu	Jakarta	SI

```
1 row in set (0.00 sec)
```

Selain dengan menggunakan query **SELECT * FROM namaTabel** .Anda juga dapat menampilkan data sesuai kebutuhan dengan menggunakan kondisi.

Menampilkan Seluruh data/record

```
SELECT * FROM mahasiswa;
```

```
mysql> SELECT * FROM mahasiswa;
```

nim	nama	alamat	jurusan
121210	Resa	Yogya	TI
121211	Ayu	Jakarta	SI

```
2 rows in set (0.00 sec)
```

Menampilkan data **nim dan nama** dari **jurusan**

```
SELECT nim,nama FROM mahasiswa WHERE jurusan='TI';
```

```
mysql> SELECT nim,nama FROM mahasiswa WHERE jurusan='TI';
```

nim	nama
121210	Resa

```
1 row in set (0.00 sec)
```

Menampilkan data **nama** dari **nim**

```
SELECT nama FROM mahasiswa WHERE NIM=121211;
```

```
mysql> SELECT nama FROM mahasiswa WHERE nim=121211;
```

nama
Ayu

```
1 row in set (0.06 sec)
```

Pada tutorial selanjutnya saya bahas relasi tabel di database. Ada 6 jenis relasi tabel di MySQL.

- 1) Inner Join
- 2) Cross Join
- 3) Straight Join
- 4) Right Join
- 5) Left Join
- 6) Natural Join

Pada contoh relasi ini saya membuat 2 tabel. Tabel pertama adalah **dosen** dan tabel kedua adalah **matkul**. Disini saya akan menampilkan kode dosen sesuai dengan nama mata kuliah dosen tersebut. Cara membuat tabel sudah saya jelaskan caranya diatas. Setelah membuat tabel, isi dengan data seperti dibawah ini sebagai contoh.

Tabel dosen

```
mysql> SELECT * FROM dosen;
```

kd_dosen	nama	email
11	Hari	hari@gmail.com
12	Ani	ani@gmail.com
13	Candra	candra@yahoo.com
14	Ayu	ayu@gmail.com

```
4 rows in set (0.00 sec)
```

Tabel matkul

```
mysql> SELECT * FROM matkul;
```

kd_matkul	kd_dosen	nama_matkul	sks
1	11	PBO	3
2	12	Web Programming	3
3	13	RPL	2
4	14	Algoritma 1	2

```
4 rows in set (0.00 sec)
```

1. INNER JOIN

a. INNER JOIN – ON

Relasi INNER JOIN – ON adalah untuk menampilkan data/record dari dua tabel yang berelasi.

```
SELECT * FROM dosen INNER JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen;
```

```
mysql> SELECT * FROM dosen INNER JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen;
```

kd_dosen	nama	email	kd_matkul	kd_dosen	nama_matkul	sks
11	Hari	hari@gmail.com	1	11	PBO	3
12	Ani	ani@gmail.com	2	12	Web Programming	3
13	Candra	candra@yahoo.com	3	13	RPL	2
14	Ayu	ayu@gmail.com	4	14	Algoritma 1	2

```
4 rows in set (0.00 sec)
```

b. INNER JOIN – USING

Berbeda dengan relasi INNER JOIN – ON. Relasi INNER JOIN - USING menampilkan data yang saling berelasi tetapi hanya menampilkan satu field saja yang digunakan sebagai penghubung.

```
SELECT * FROM dosen INNER JOIN matkul USING (kd_dosen);
```



```
mysql> SELECT * FROM dosen INNER JOIN matkul USING (kd_dosen);
```

kd_dosen	nama	email	kd_matkul	nama_matkul	sks
11	Hari	hari@gmail.com	1	PBO	3
12	Ani	ani@gmail.com	2	Web Programming	3
13	Candra	candra@yahoo.com	3	RPL	2
14	Ayu	ayu@gmail.com	4	Algoritma 1	2

4 rows in set (0.00 sec)

2. CROSS JOIN

Berikut ini adalah contoh relasi tabel dengan menggunakan CROSS JOIN

```
SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,dosen.nama,dosen.email FROM
dosen CROSS JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen;
```

```
mysql> SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,dosen.nama,dosen.email FROM
dosen CROSS JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen;
```

kd_matkul	nama_matkul	sks	nama	email
1	PBO	3	Hari	hari@gmail.com
2	Web Programming	3	Ani	ani@gmail.com
3	RPL	2	Candra	candra@yahoo.com
4	Algoritma 1	2	Ayu	ayu@gmail.com

4 rows in set (0.00 sec)

3. STRAIGHT JOIN

Berikut ini adalah contoh relasi tabel dengan menggunakan STRAIGHT JOIN

```
SELECT kd_matkul,nama_matkul,sks,nama,email FROM dosen STRAIGHT JOIN matkul;
```

```
mysql> SELECT kd_matkul,nama_matkul,sks,nama,email FROM dosen STRAIGHT JOIN matkul;
```

kd_matkul	nama_matkul	sks	nama	email
1	PBO	3	Hari	hari@gmail.com
1	PBO	3	Ani	ani@gmail.com
1	PBO	3	Candra	candra@yahoo.com
1	PBO	3	Ayu	ayu@gmail.com
2	Web Programming	3	Hari	hari@gmail.com
2	Web Programming	3	Ani	ani@gmail.com
2	Web Programming	3	Candra	candra@yahoo.com
2	Web Programming	3	Ayu	ayu@gmail.com
3	RPL	2	Hari	hari@gmail.com
3	RPL	2	Ani	ani@gmail.com
3	RPL	2	Candra	candra@yahoo.com
3	RPL	2	Ayu	ayu@gmail.com
4	Algoritma 1	2	Hari	hari@gmail.com
4	Algoritma 1	2	Ani	ani@gmail.com
4	Algoritma 1	2	Candra	candra@yahoo.com
4	Algoritma 1	2	Ayu	ayu@gmail.com

16 rows in set (0.00 sec)

Jadi STRAIGHT JOIN dan CROSS JOIN hampir sama.

4. RIGHT JOIN

Untuk membuat relasi tabel **RIGHT JOIN** dan **LEFT JOIN**, kita rubah dulu struktur datanya dan recordnya menjadi seperti ini

```
mysql> SELECT * FROM dosen;
```


kd_dosen	nama	email
11	Hari	hari@gmail.com
12	Ani	ani@gmail.com
13	Candra	candra@yahoo.com
14	Ayu	ayu@gmail.com

4 rows in set (0.00 sec)

```
mysql> SELECT * FROM matkul;
```

kd_matkul	nama_matkul	sks	jurusan	kd_dosen
1	PBO	3	TI	11
2	Web Programming	3	SI	12
3	RPL	2	TI	11
4	Algoritma 1	2	SI	14

4 rows in set (0.00 sec)



Pada tabel matkul terdapat kode dosen yang sama yaitu 11, biasanya dalam beberapa mata kuliah terdapat dosen yang sama mengajar mata kuliah tersebut. Berikut ini query RIGHT JOIN untuk melihat relasi ke dua tabel.

```
SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,matkul.jurusan,dosen.nama,dosen.email  
FROM dosen RIGHT JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen ORDER BY dosen.email;
```

```
mysql> SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,matkul.jurusan,dosen.nama,dosen.email FROM dosen  
RIGHT JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen ORDER BY dosen.email;
```

kd_matkul	nama_matkul	sks	jurusan	nama	email
2	Web Programming	3	SI	Ani	ani@gmail.com
4	Algoritma 1	2	SI	Ayu	ayu@gmail.com
3	RPL	2	TI	Hari	hari@gmail.com
1	PBO	3	TI	Hari	hari@gmail.com

4 rows in set (0.00 sec)

Dari data yang ditampilkan dapat dilihat bahwa RIGHT JOIN menampilkan record pada tabel sebelah kanan.

5. LEFT JOIN

Selain RIGHT JOIN kita juga dapat menggunakan LEFT JOIN untuk menampilkan record. Berikut ini contoh jika menggunakan query LEFT JOIN.

```
SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,matkul.jurusan,dosen.nama,dosen.email  
FROM dosen LEFT JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen ORDER BY dosen.email;
```

```
mysql> SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,matkul.jurusan,dosen.nama,dosen.email FROM dosen
LEFT JOIN matkul ON dosen.kd_dosen = matkul.kd_dosen ORDER BY dosen.email;
```

kd_matkul	nama_matkul	sks	jurusan	nama	email
2	Web Programming	3	SI	Ani	ani@gmail.com
4	Algoritma 1	2	SI	Ayu	ayu@gmail.com
NULL	NULL	NULL	NULL	Candra	candra@yahoo.com
1	PBO	3	TI	Hari	hari@gmail.com
3	RPL	2	TI	Hari	hari@gmail.com

5 rows in set (0.00 sec)

Dari data yang ditampilkan dapat dilihat bahwa LEFT JOIN menampilkan record pada tabel sebelah kiri. Terdapat record dengan nilai NULL, karena tidak ada pasangan tidak ada pasangan dari tabel dosen dengan tabel matkul. Karena tidak ada pasangan maka akan diisi dengan NULL. Penjelasannya adalah dosen dengan nama Candra ada di dalam data dosen tetapi dia tidak mengajar kuliah di semester tersebut.

6. NATURAL JOIN

Jika anda ingin merelasikan tabel dengan mudah dan cepat, anda dapat menggunakan NATURAL JOIN. NATURAL JOIN akan merelasikan tabel tersebut secara otomatis. Berikut ini query NATURAL JOIN

```
SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,matkul.jurusan,dosen.nama,dosen.email
FROM dosen NATURAL JOIN matkul;
```

```
mysql> SELECT matkul.kd_matkul,matkul.nama_matkul,matkul.sks,matkul.jurusan,dosen.nama,dosen.email FROM dosen
NATURAL JOIN matkul;
```

kd_matkul	nama_matkul	sks	jurusan	nama	email
1	PBO	3	TI	Hari	hari@gmail.com
3	RPL	2	TI	Hari	hari@gmail.com
2	Web Programming	3	SI	Ani	ani@gmail.com
4	Algoritma 1	2	SI	Ayu	ayu@gmail.com

4 rows in set (0.00 sec)

Untuk berjaga – jaga terhadap kerusakan database atau hilangnya database, maka langkah yang paling penting adalah backup database dengan menggunakan **mysqldump**. Berikut ini saya akan berikan contoh dump database dengan mysql.

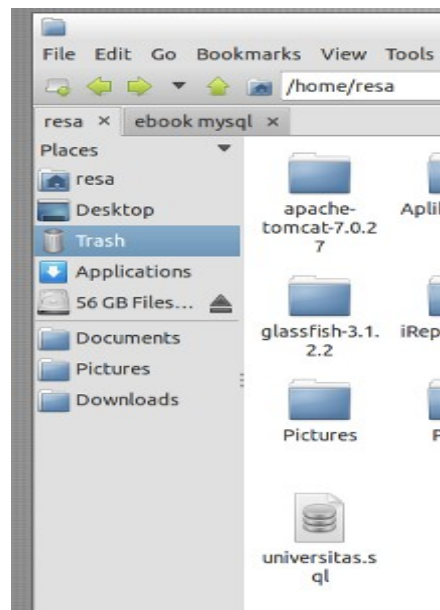
```
resa@resa:~$ mysqldump -u root -p universitas > /home/resa/universitas.sql
Enter password:
```

```
mysqldump -u root -p universitas > /home/resa/universitas.sql
```

Penjelasan :

-u	= username mysql
-p	= password mysql
universitas	= nama database
> /home/resa/	= lokasi penyimpanan file hasil backup mysql
universitas.sql	= nama file sqlnya

Lokasi file sql di directory **/home/resa**



Jika ingin membackupnya sekaligus dicompress dengan zip.Maka perintahnya

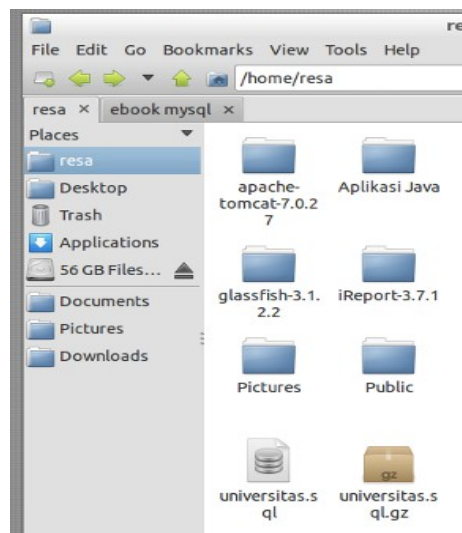
```
resa@resa:~$ mysqldump -u root -p universitas | gzip > /home/resa/universitas.sql.gz  
Enter password:
```

```
mysqldump -u root -p universitas | gzip > /home/resa/universitas.sql.gz
```

Penjelasan :

-u	= username mysql
-p	= password mysql
universitas	= nama database
 gzip > /home/resa/	= jenis kompresi dan lokasi penyimpanan
universitas.sql.gz	= nama file kompresi

Lokasi file sql yang telah dikompres di directory **/home/resa**



Untuk mencoba restore database, drop/hapus database yang pernah dibuat. Menghapus database dengan perintah

```
DROP DATABASE universitas;
```

```
mysql> DROP DATABASE universitas;
Query OK, 3 rows affected (0.22 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.00 sec)
```

Kalau sudah berhasil dihapus, sekarang waktunya untuk merestore database. Pertama buat dulu database di MySQL. Contohnya database universitas. Untuk merestore ketikkan perintah

```
resa@resa:~$ mysql -u root -p universitas < /home/resa/universitas.sql
Enter password:
```

```
mysql -u root -p universitas < /home/resa/universitas.sql
```

Penjelasan :

-u	= username mysql
-p	= password mysql
universitas	= nama database
< /home/resa/	= lokasi file restore sqlnya
universitas.sql	= nama file sqlnya

Database yang berhasil direstore

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
| universitas |
+-----+
6 rows in set (0.00 sec)

mysql> USE universitas;
Reading table information for d
You can turn off this feature t

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_universitas |
+-----+
| dosen |
| mahasiswa |
| matkul |
+-----+
```



```
mysql> SELECT * FROM dosen;
```

kd_dosen	nama	email
11	Hari	hari@gmail.com
12	Ani	ani@gmail.com
13	Candra	candra@yahoo.com
14	Ayu	ayu@gmail.com

```
4 rows in set (0.00 sec)
```



```
mysql> SELECT * FROM matkul;
```

kd_matkul	nama_matkul	sks	jurusan	kd_dosen
1	PBO	3	TI	11
2	Web Programming	3	SI	12
3	RPL	2	TI	11
4	Algoritma 1	2	SI	14

```
4 rows in set (0.00 sec)
```

Selamat mencoba dan semoga bermanfaat :)