

29 Fitur Dahsyat

Database **MySQL**

Yang biasa ternyata luar biasa



Dendie Sanjaya

Blog Penulis
<http://dendieisme.blogspot.com>

Kata Pengantar

Dalam perkembangan teknologi informasi penggunaan database adalah sesuatu yang begitu penting untuk memenuhi tuntutan kebutuhan zaman. Diantara sekian banyak database yang telah diciptakan dari yang berbayar hingga free & open source terdapat sebuah database yang begitu populer yaitu MySQL.

Database MySQL adalah sebuah database yang bersifat open source dan didistribusikan secara free. Walaupun MySQL bersifat free tetapi sanggup untuk menjawab tuntutan kebutuhan teknologi informasi, hal ini terbukti dengan semakin banyaknya pengguna database MySQL dan berbagai fitur yang tersedia pada MySQL.

Bagi Anda yang baru mengenal database MySQL buku ini dapat memberikan penjelasan yang menarik mengenai kemampuan-kemampuan yang dimiliki MySQL dalam menyelesaikan berbagai kasus pengelolaan data. Sedangkan bagi yang telah berpengalaman dengan database MySQL buku ini akan membantu Anda untuk meng-explore lebih jauh MySQL dalam memanfaatkan fitur-fitur MySQL untuk menyelesaikan berbagai persoalan pengolahan data.

Meteri buku ini memang ditujukan untuk tingkatan yang telah mengenal database MySQL. Materi pada buku ini disampaikan dalam berbagai contoh kasus yang sering dijumpai dengan penyampaian gaya bahasa yang singkat & jelas sehingga Anda mudah

memahaminya. Anda akan mendapatkan cara-cara yang praktis dalam menyelesaikan berbagai persoalan pengelolaan data, yang semuanya itu dikumpulkan dari pengalaman penulis, dokumentasi online, berbagai tips di internet dan mailing-list serta sumber lainnya.

Tak lupa pada kesempatan yang berbahagia ini, penulis ingin menyampaikan rasa terima kasih kepada beberapa pihak, antara lain.

1. PT. Cinox Media Insani tempat penulis berkerja yang telah memberikan kesempatan pada penulis untuk berkarya dan belajar dibidang teknologi informasi dan komputer.
2. Ayah dan Ibu penulis yang telah memberikan dukungan moral dan materi hingga akhirnya penulis dapat berkarya di bidang teknologi informasi dan komputer.
3. Persembahan spesial untuk kekasih penulis, ulie yang selalu menyemangati penulis untuk terus berkarya.
4. Para pembaca blog penulis di <http://dendieisme.blogspot.com> terima kasih telah memberikan komentar dan responnya,
5. Teman-teman luar biasa penulis di PT. Cinox Media Insani yang telah membuat penulis berkembang.
6. Dosen kampus penulis di Unibi (Universitas Informatika dan Bisnis Indonesia) Bandung yang telah berbagi pengetahuan dan wawasannya.

Akhirnya penulis menyadari masih banyak kekurangan dan kelemahan dalam buku ini, oleh karena itu saran, pendapat dan koreksi dari pembaca sangat diharapkan penulis. Semoga buku ini dapat bermamfaat bagi pembaca sekalian.

Bandung, 17 April 2011

Penulis

Blog : <http://dendieisme.blogspot.com>
E-mail : dendie_sanjaya@yahoo.com

Daftar Isi

Kata Pengantar.....	2
Daftar Isi.....	5
1. Pengenalan MySQL.....	7
1.1 Tentang MySQL.....	8
1.2 Keistimewaan MySQL.....	13
1.3 Instalasi MySQL.....	16
1.4 GUI Front End for MySQL.....	26
2. 29 Fitur Dahsyat.....	32
2.1 Trigger.....	33
2.2 Replikasi Database.....	37
2.3 View.....	43
2.4 Store Procedure.....	46
2.5 Function.....	48
2.6 Backup dan Restore Database	50
2.7 Transaction.....	52
2.8 Database Firewall.....	57
2.9 Partition Table.....	61
2.10 MySQL Event Scheduler.....	68
2.11 Cursor Store Procedure.....	73
2.12 Lock Tables.....	75
2.13 Auto Backup Database Menggunakan Crontab.....	78
2.14 Referential Integrity.....	83
2.15 MySQL Smart Search.....	89
2.16. MySQL Soundex.....	94
2.17 Manajemen User.....	95
2.18. MySQL Remove Definer.....	99

2.19 MySQL - Show Procedure, Function, Triggers.....	101
2.20 Load Data Infile.....	103
2.21 MySQL Copy Table.....	107
2.22 MySQL Export to CSV.....	109
2.23 MySQL Concat.....	111
2.24 MySQL Date Time.....	113
2.25 MySQL Unicode UTF8.....	118
2.26 Prepare Statement.....	122
2.27 Repair Table.....	124
2.28 Into Select	128
2.29 Show Processlist.....	130

Pengenalan MySQL



1.1 Tentang MySQL

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan developer software dan konsultan database bernama MYSQL AB yang berada di Swedia. Saat itu perusahaan tersebut masih bernama TcX DataKonsult AB dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis web pada client.

Seseorang yang bernama Michael "Monty" Widenius adalah developer satu-satunya di TcX yang memiliki sebuah aplikasi UNIREG dan rutin ISAM buatannya sendiri dan sedang mencari interface SQL yang cocok untuk diimplementasikan ke dalamnya.

Mula-mula Monty memakai miniSQL (mSQL) pada eksperimennya itu, namun SQL dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan query. Akhirnya Monty menghubungi David Hughes, pembuat mSQL yang sedang merilis versi kedua dari mSQL.

Kemudian Monty mencoba membuat sendiri mesin SQL yang memiliki interface mirip dengan SQL, tetapi dengan kemampuan yang lebih sesuai sehingga lahirlah MySQL. Tentang pengambilan nama MySQL sampai saat ini masih belum jelas asal usulnya. Ada yang berpendapat nama My diambil dari huruf depan dan belakang Monty, tetapi versi lain mengatakan nama itu diambil dari putri Monty yang kebetulan juga bernama My.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial

Pengembangan Database MySQL telah memberikan pengaruh yang positif dalam proyek software yang bersifat Free & Open Source untuk mendapatkan database free yang berfitur lengkap. Joomla, Wordpress, MyBB, phpBB, Drupal adalah beberapa software yang menggunakan database MySQL. MySQL juga digunakan di banyak skala besar World Wide Web seperti wikipedia.org, nokia.com, youtube.com, google.com (meskipun tidak untuk pencarian).

MySQL ditulis dalam bahasa pemrograman C dan C++, MySQL tersedia diberbagai sistem operasi seperti AIX, BSDI, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS / 2 Warp, QNX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos dan Tru64. Dan menyediakan berbagai library untuk mengakses database MySQL seperti MySQL Connector untuk integrasi dengan Microsoft Visual Studio (bahasa seperti C # dan VB), driver ODBC untuk Java yang disebut MyODBC.

MySQL dapat diinstal secara manual dengan cara melakukan kompilasi sendiri source code mysql yang biasanya memerlukan kustomisasi khusus agar seesuai dengan yang dibutuhkan tetapi pada

umumnya user menyukai paket biner mysql yang telah siap diinstal.

Pengembangan Database MySQL dimulai sebagai alternatif low-end database tetapi secara bertahap berkembang untuk mendukung kebutuhan yang lebih tinggi dan skala yang lebih luas. Untuk keperluan skala menengah MySQL dapat ditingkatkan dengan memperkuat spesifikasi perangkat keras yang lebih besar, seperti menggunakan server multi-prosesor dan memori yang memiliki kapasitas gigabyte.

Untuk skala yang lebih besar diperlukan peningkatan performa dan reliabilitas dengan cara melakukan konfigurasi database level high-end seperti implementasi replikasi database yang terbagi menjadi master database untuk proses write data dan slave database untuk menangani operasi read data.

Untuk peningkatan dalam kinerja yang lebih baik pada skala yang sangat besar dapat dicapai dengan melakukan caching query database dalam memori menggunakan memcached atau dengan merancang topologi database menjadi beberapa bagian yang tersebar atau dikenal dengan database cluster terdistribusi.

MySQL menawarkan Support 24 jam dan 7 hari dalam seminggu dengan waktu respon 30 menit yang diberikan pada customer produk MySQL Enterprise, Tim Support memiliki akses langsung ke para developer MySQL yang diperlukan untuk menyelesaikan masalah. serta akan mendapatkan akses update MySQL dalam perbaikan bug.

Pada bulan Oktober 2005, Oracle Corporation melakukan akuisisi perusahaan Innobase OY sebuah perusahaan asal Finlandia yang mengembangkan store engine InnoDB yang diperlukan MySQL dalam menyediakan fungsionalitas seperti transaksi dan foreign key.

Setelah akuisisi tersebut Oracle mengumumkan bahwa kontrak perusahaan Innobase OY dengan MySQL AB dalam menyediakan storage innodb akan jatuh tempo pada tahun 2006 dan kontrak tersebut perlu diperbaharui. Pada April 2006 MySQL mengumumkan dan menegaskan bahwa MySQL dan Innobase OY sepakat untuk memperpanjang perjanjian mereka.

Pada bulan Februari 2006, Oracle Corporation mengakuisisi perusahaan Sleepycat yang membuat store engine dari Berkeley DB untuk Database MySQL, tetapi hal ini tidak memiliki pengaruh yang besar karena pengguna storage engine Berkeley DB tidak banyak yang menggunakan.

Pada bulan Januari 2008, Perusahaan besar Sun Microsystems membeli MySQL sebesar US\$ 1 miliar, kemudian Pada bulan April 2009, Oracle Corporation menandatangani kesepakatan membeli Sun Microsystems dan menjadi pemilik hak cipta dan merek dagang MySQL yang disetujui dewan direksi dan pemegang saham Sun Microsystem serta pemerintah USA pada 20 Agustus 2009.

Pada tanggal 14 Desember 2009, Oracle berjanji untuk terus memelihara dan meningkatkan MySQL seperti yang dilakukan selama

empat tahun sebelumnya.

Sebuah gerakan melawan akuisisi Oracle pada MySQL dengan nama "Save MySQL" dari Oracle dimulai oleh pendiri MySQL yaitu Monty Widenius. Petisi dari 50.000+ developer dan pengguna MySQL meminta Komisi Eropa untuk memblokir persetujuan akuisisi.

Sebagai bagian dari negosiasi antara Komisi Eropa dan Oracle berkomitmen bahwa Database MySQL Server akan terus menggunakan dual-lisensi yaitu seperti yang digunakan MySQL AB yaitu versi komersial dan GPL setidaknya sampai tahun 2015, Akuisisi Oracle akhirnya disetujui tanpa syarat oleh Komisi Eropa pada tanggal 21 Januari 2010.

Sementara itu, Monty Widenius telah merilis database bernama MariaDB berlisensi GPL, MariaDB dibuat berdasarkan pada source kode yang sama dengan Database MySQL Server dan berusaha untuk menjaga kompatibilitas dengan Database Oracle.

1.2 Keistimewaan MySQL

MySQL adalah sebuah RDMS (Relation Database Manajemen System) yang berfungsi untuk menyimpan dan mengelolah data. MySQL meng-klaim telah di install lebih dari enam juta di seluruh dunia. Database MySQL tersedia gratis menggunakan lisensi GPL (General Public License). Berikut ini adalah beberapa keistimewaan Databse MySQL

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Open Source Software. MySQL didistribusikan sebagai open source software, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Performance tuning. MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
11. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
12. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis

data lainnya semacam PostgreSQL ataupun Oracle.

13. Memiliki berbagai macam Store Engine, seperti MyISAM yang sangat cepat untuk menanggapi data-data non transaksional. InnoDB yang menunjukkan performance yang baik dalam menangani data-data transaksional.

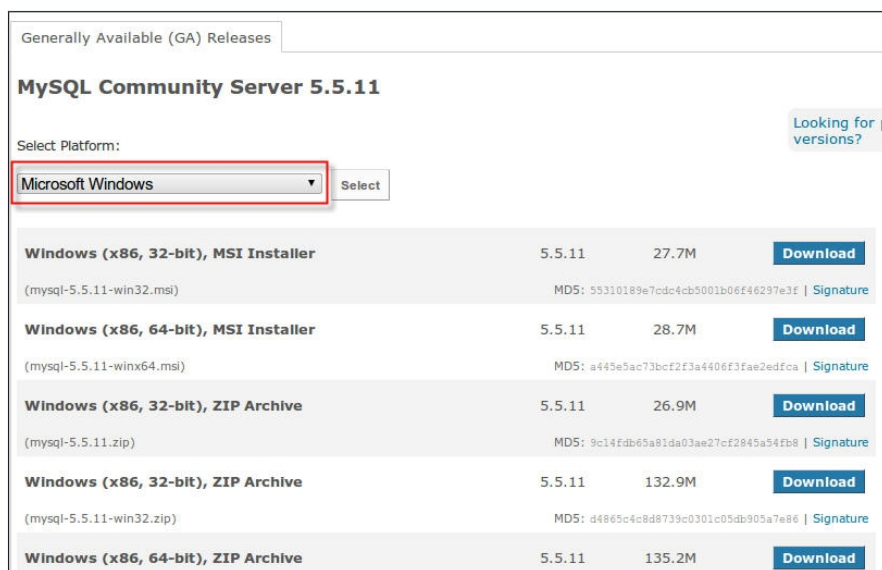
1.3 Instalasi MySQL

MySQL adalah sebuah RDMS (Relation Database Management System) yang berfungsi untuk menyimpan dan mengelolah data yang bisa didapatkan secara gratis dan tersedia diberbagi sistem operasi seperti AIX, FreeBSD, Linux, Mac OS, Microsoft Windows, OpenSolaris dan sebagainya.

Berikut ini penulis akan menjelaskan cara instalasi pada sebuah sistem operasi yang populer yaitu Microsoft Windows. Sebelum memulai instalasi Anda harus memiliki Instaler MySQL yang bisa didapat di alamat <http://www.mysql.com/downloads/mysql/>

a. Berikut ini adalah cara untuk download database MySQL

1. Kunjungi alamat <http://www.mysql.com/downloads/mysql/> kemudian pilih platform Microsoft Windows seperti yang ditunjukkan pada gambar dibawah ini.



Generally Available (GA) Releases

MySQL Community Server 5.5.11

Select Platform: Looking for previous versions?

Microsoft Windows

Windows (x86, 32-bit), MSI Installer (mysql-5.5.11-win32.msi)	5.5.11	27.7M	Download
Windows (x86, 64-bit), MSI Installer (mysql-5.5.11-winx64.msi)	5.5.11	28.7M	Download
Windows (x86, 32-bit), ZIP Archive (mysql-5.5.11.zip)	5.5.11	26.9M	Download
Windows (x86, 32-bit), ZIP Archive (mysql-5.5.11-win32.zip)	5.5.11	132.9M	Download
Windows (x86, 64-bit), ZIP Archive	5.5.11	135.2M	Download

2. Apabila platform Microsoft Windows akan menampilkan daftar tipe pake instalasi yang dapat dipilih, sebagai contoh pilih Windows (x86, 32 Bit), MSI Installer kemudian klik tombol download.



3. Setelah tombol download ditekan akan menampilkan form seperti gambar dibawah ini. Klik link **No tanks, just take me to the downloads!** Untuk melanjutkan proses download.

Please take the time to let us know about you.

If this is the first time you have downloaded from us, you will be sent a password to enable you to log into web sites, including forums and bugs.

If you already have a MySQL.com account, save time by logging in now.

Returning Users	New Users
Save time by logging in	Proceed with registration
Email: <input type="text"/>	
Password: <input type="password"/>	
Forgot your password?	
<input type="button" value="Login"/>	<input type="button" value="Proceed"/>

[» No tanks, just take me to the downloads!](#)

4. Kemudian akan menampilkan daftar lokasi yang dapat dipilih untuk memulai download MySQL. Pilih negara yang paling dekat dengan Indonesia karena semakin dekat maka proses download semakin cepat. Sebagai contoh saya memilih negara Singapura.

Asia		
	sPD Hosting, Israel	HTTP
	Internet Initiative Japan Inc., Japan	HTTP FTP
	JAIST, Japan	HTTP FTP
	Kyung Hee University Linux User Group, Korea, Republic of	HTTP FTP
	CYBER Internet Services / Lahore University of Management Sciences, Pakistan	HTTP FTP
	Advanced Science and Technology Institute, Philippines	HTTP FTP
	STC Riyadh, Saudi Arabia	HTTP FTP
	ezNetworking Solutions Pte. Ltd., Singapore	HTTP FTP
	National Taiwan University, Taiwan	HTTP FTP
	Computer Center, Shu-Te University / Kachsiung, Taiwan	HTTP FTP
	Providence University, Taiwan	HTTP FTP

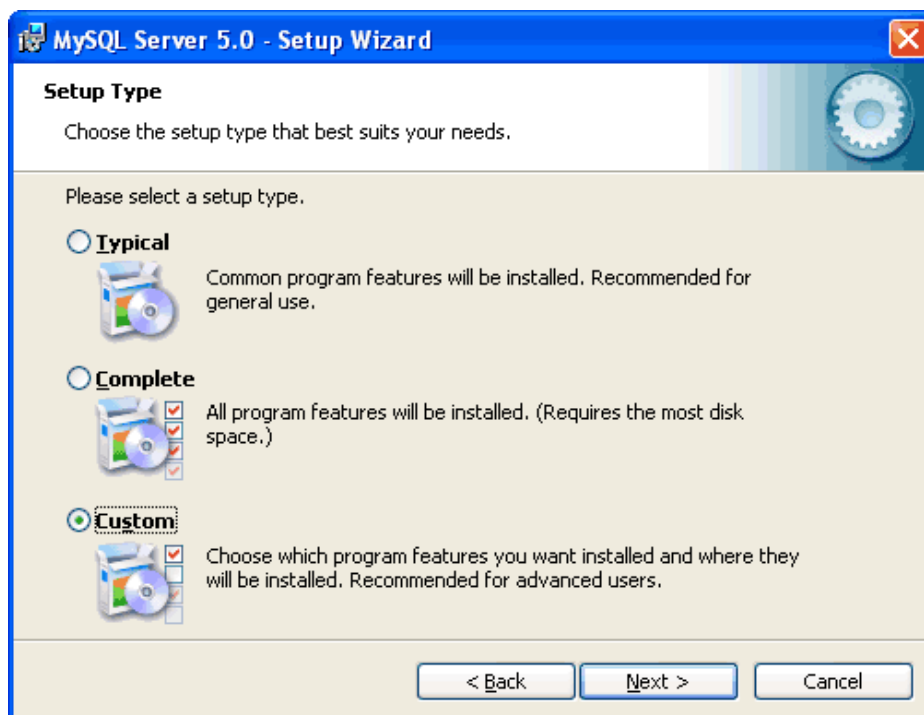
b. Berikut ini adalah cara untuk instalasi MySQL di Microsoft windows

Setelah Anda mendapatkan installer Database MySQL yang sesuai dengan Sistem Operasi yang digunakan, selanjutnya adalah melakukan instalasi MySQL. Berikut ini adalah tahapan dalam instalasi MySQL.

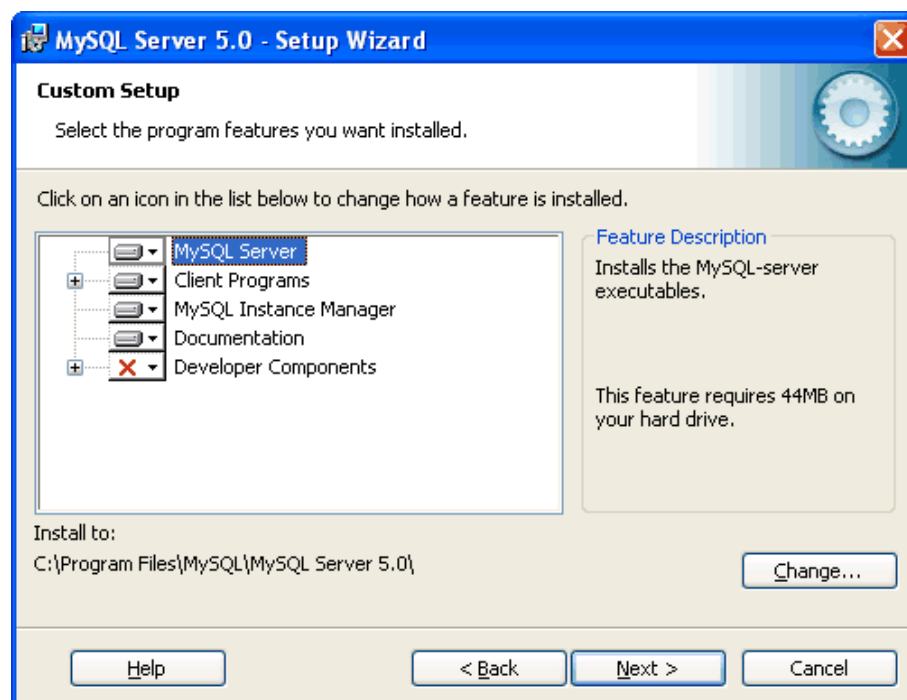
1. Klik file setup MySQL, setelah diklik maka akan menghasilkan tampilan seperti dibawah ini. Kemudian klik tombol Next untuk melanjutkan proses instalasi.



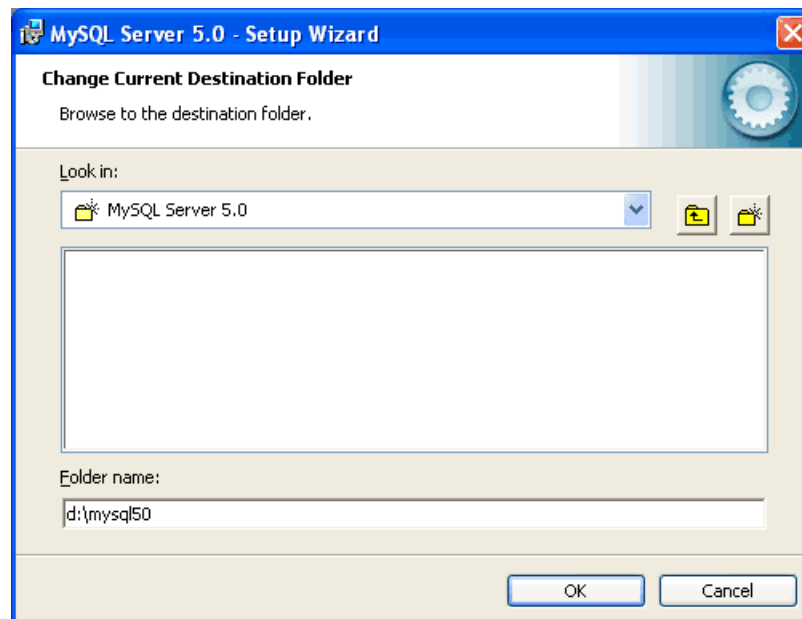
2. Pilih tipe setup instalasi Custom karena kita akan melakukan sedikit kustomisasi instalasi



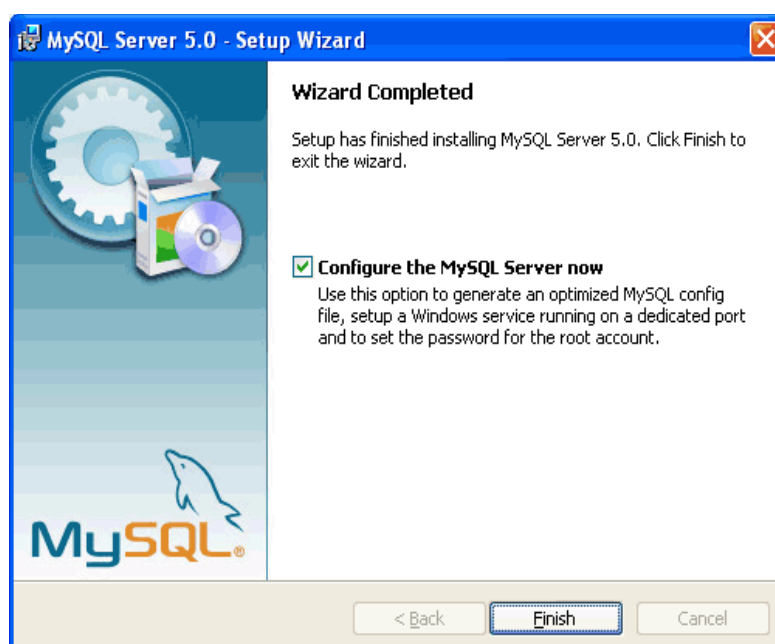
3. Pada pilihan fitur cukup memilih MySQL Server. Kemudian dilanjutkan dengan menekan tombol Next.



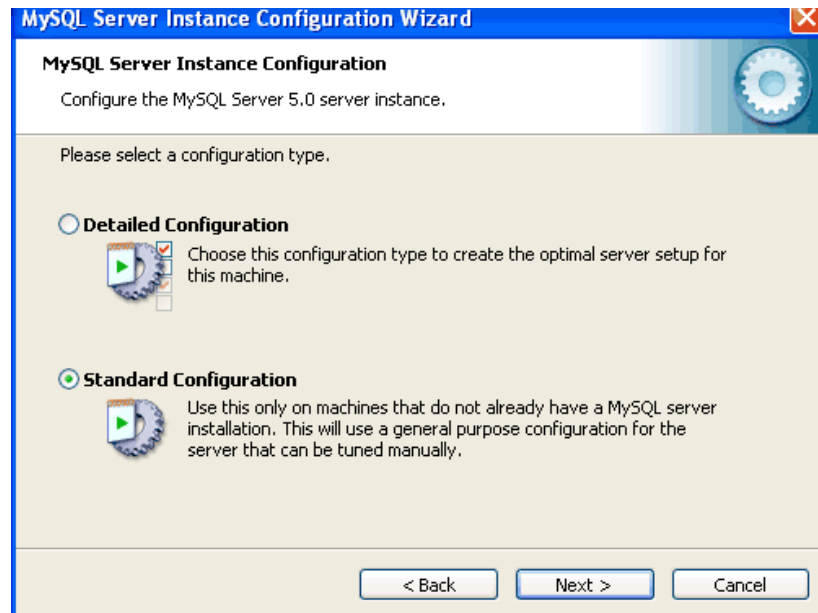
4. Kemudian akan muncul tujuan folder instalasi MySQL. Pada contoh ini penulis menentukan alamat [d:\mysql50](#) sebagai tujuan instalasi.



5. Setelah proses instalasi selesai dan berjalan dengan sempurna maka akan menampilkan window seperti dibawah ini. Chckbox tulisan **Configure the MySQL Server now** untuk memulai konfigurasi pertama.



6. Setelah tombol Finish ditekan maka akan menampilkan pilihan tipe konfigurasi. Sebagai contoh penulis memilih **Standard Configuration**.



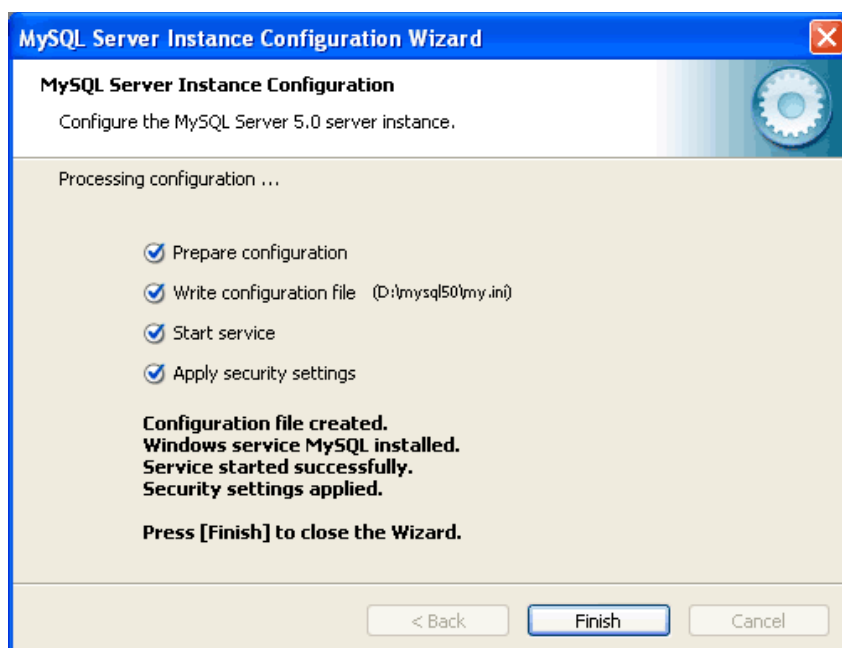
7. Kemudian akan menampilkan pilih cara inisialiasi MySQL database pada sistem operasi windows. Sebagai contoh penulis memilih **Install As windows Service**.



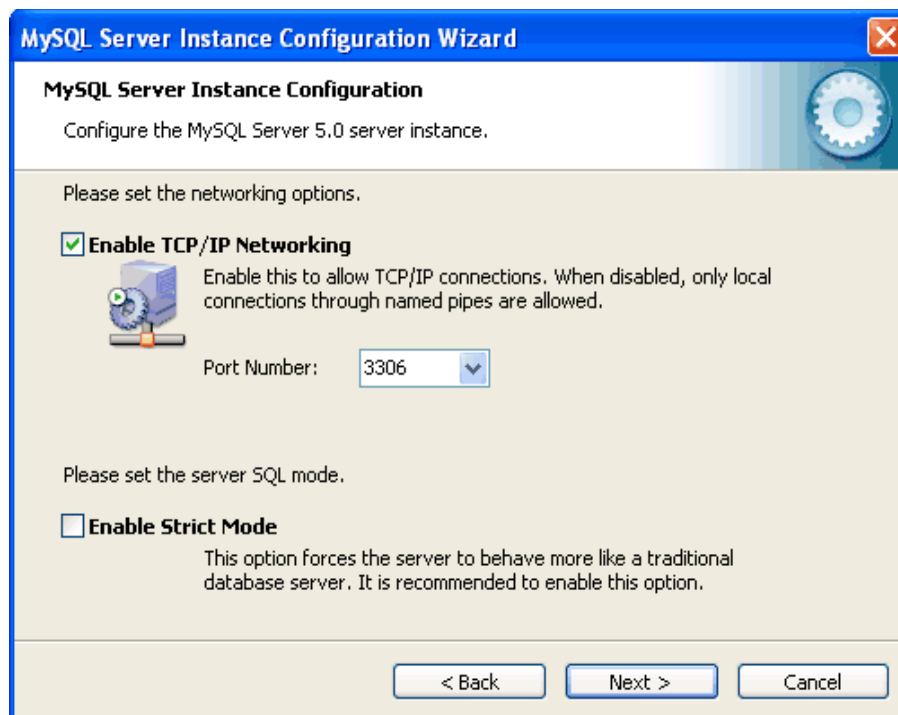
8. Kemudian untuk menjaga keamanan database, maka MySQL meminta kita untuk meng- inputkan password untuk root atau super admin.



9. Selanjutnya adalah persiapan proses konfigurasi. Pada contoh dibawah ini penulis memilih semua option yang diberikan oleh MySQL.



10. Secara default service database MySQL Server beroperasi di port number 3306, seperti yang dapat dilihat pada gambar dibawah ini.



11. Apabila proses instalasi berhasil maka pada menu program Windows akan terdapat MySQL seperti yang ditunjukkan pada gambar dibawah ini.



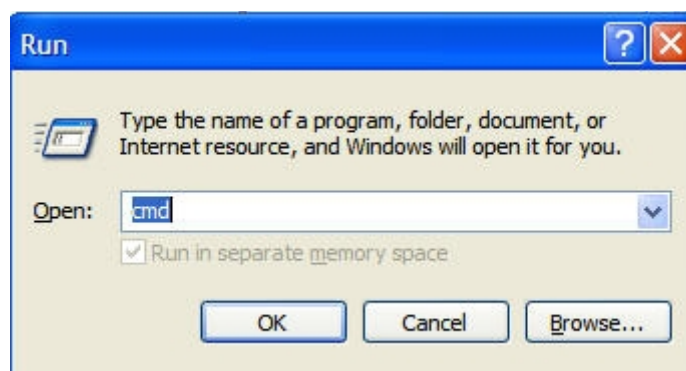
Terkadang dalam pengembangan sebuah software diperlukan untuk dapat mengakses database MySQL melalui Command Line. Berikut ini adalah contoh untuk mengakses MySQL Command Line Client.

1. MySQL Command Line Client dapat diakses pada menu program MySQL, seperti yang ditunjukkan pada gambar dibawah ini.

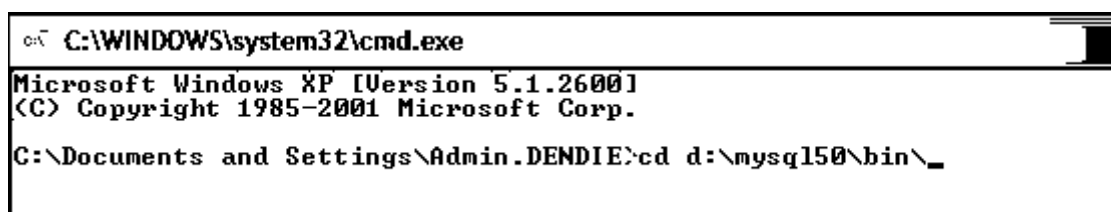


2. Selain melauli menu program MySQL untuk medapatkan Command Line Client, Anda dapat menggunakan cara dibawah ini.

a. Jalan Command Line Windows melalui program Run, seperti yang ditunjukkan pada gambar dibawah ini.



b. Setelah program Command Line muncul, masuk kedalam folder Bin dari MySQL. Pada contoh dibawah ini folder tersebut berada di **d:\mysql50\bin** dan kata **cd** sebelum **d:\mysql50\bin** adalah perintah untuk membuka folder.



c. Kemudian ketikkan **mysql** dan masukan **username** beserta **password** untuk masuk ke dalam database MySQL Server, seperti pada gambar dibawah ini.

```
mysql -uroot -ppass
```

d. Apabila proses login ke database MySQL Server berhasil maka layar akan menampilkan gambar seperti dibawah ini.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.8 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, 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> _
```

c. Berikut ini adalah cara untuk instalasi *MySQL* di Linux Ubuntu

Untuk instalasi MySQL pada linux ubuntu Anda dapat dengan cara download source code kemudian melakukan kompilasi sendiri atau melalui repositori ubuntu. Berikut ini penulis akan mencontohkan cara instalasi MySQL melalui repositori ubuntu.

- Pada Console ketikkan cukup ketikkan perintah seperti dibawah ini

```
sudo apt-get install mysql-server
```

Apabila Anda ingin masuk ke MySQL Command Line Client, pada console cukup ketikkan perintah seperti ini.

```
mysql -uroot -ppass
```

- Secara default username MySQL adalah **root** dengan password **pass**

1.4 GUI Front End for MySQL

SQL (Struktur Query Language) adalah sebuah bahasa standar untuk melakukan manipulasi (menambah, membuat, merubah, menghapus, menampilkan) data & objek pada sebuah Database berjenis Relasi.

MySQL adalah sebuah Database berjenis Relasi sehingga untuk manipulasi datanya dapat menggunakan SQL (Structure Query Language) Bentuk dari SQL (Structure Query Language) berupa teks, berikut ini adalah contoh perintah SQL untuk menginputkan data ke sebuah table.

```
Insert Into pegawai ('id','nama','gender','status') value ('1','Dendie Sanjaya','L','TETAP');
```

Dengan menggunakan GUI Front End, Anda tidak perlu menulis langsung SQL seperti diatas, tetapi disediakan tampilan atau GUI (Graphical User Interface) untuk melakukan operasi seperti membuat/tambah/hapus/delete/edit data atau objek (Database, Table, Function, dsb).

GUI Front End untuk MySQL dari yang free hingga berbayar diantaranya adalah Adminer, DBEdit, dbForge GUI Tools, HeidiSQL, Navicat, phpMyAdmin, MySQL Query Browser.

Pada buku ini penulis akan memilih salah satu dari sekian banyak

Front End MySQL yaitu MySQL Query Browser. Berikut ini adalah cara download dan instalasi MySQL Query Browser.

a. Download MySQL Query Browser

Untuk mendapatkan MySQL Query browser dapat di download di situs resminya yaitu <http://dev.mysql.com/downloads/gui-tools/5.0.html>. MySQL Query browser tersedia untuk beragam Sistem Operasi seperti Linux, Windows, Macintos, dll.

MySQL GUI Tools: Archived Online Documentation

- [MySQL Administrator Documentation](#)
- [MySQL Query Browser Documentation](#)
- [MySQL Migration Toolkit Documentation](#)

MySQL GUI Tools: Archived Downloads

Windows downloads

The install package uses the Windows Installer, which is built in to Windows XP and more recent Microsoft Windows versions.

Download an updated [Windows Installer for Windows 2000](#)

Windows (x86)	5.0-r17	16.9M	Download
			MDS: a389ba3831bfb8497eba81ae8a7db
Without installer (unzip in C:\)	5.0-r17	16.1M	Download
			MDS: 54593fd76f2f2fb13a2f6c9ac4a200293e

Mac OSX downloads

MAC OSX 10.4 (Universal binaries)	5.0-r12	12.4M	Download
			MDS: 87989a094c5ed6e6fb28dad50dd1a44

Linux downloads

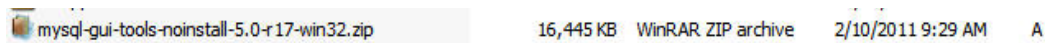
Individual RPM packages are supplied for the GUI Tools components and combined into a single TAR archive.

RedHat Enterprise Linux 3 (x86) RPM (bundled dependencies)	5.0r12	19.4M	Download
--	--------	-------	--------------------------

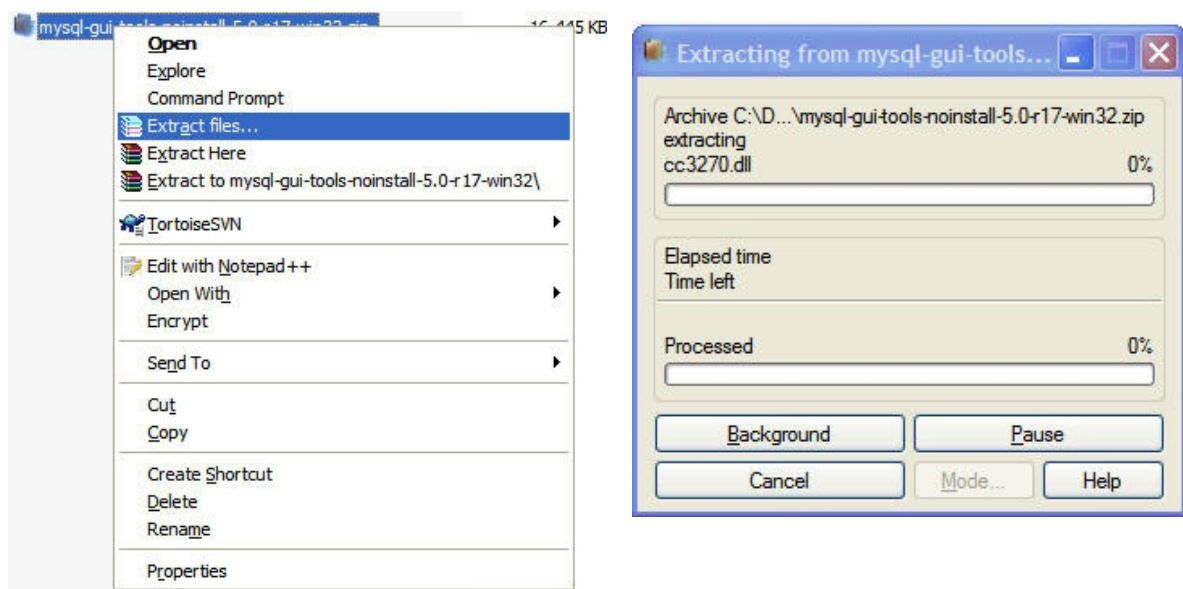
Pada buku ini MySQL Query browser yang digunakan adalah **Windows** dengan varian **without Installer (Unzip in C:)**. Varian tersebut dipilih karena tidak perlu instalasi melainkan cukup dengan **copy-paste** folder MySQL Query browser ke harddisk atau dapat dikatakan varian portable-nya.

b. Instalasi MySQL Query Browser

Paket MySQL Query browser yang digunakan adalah yang **portable** (without Installer) maka proses instalasinya lebih mudah. Berikut ini adalah cara melakukan instalasi MySQL Query browser.



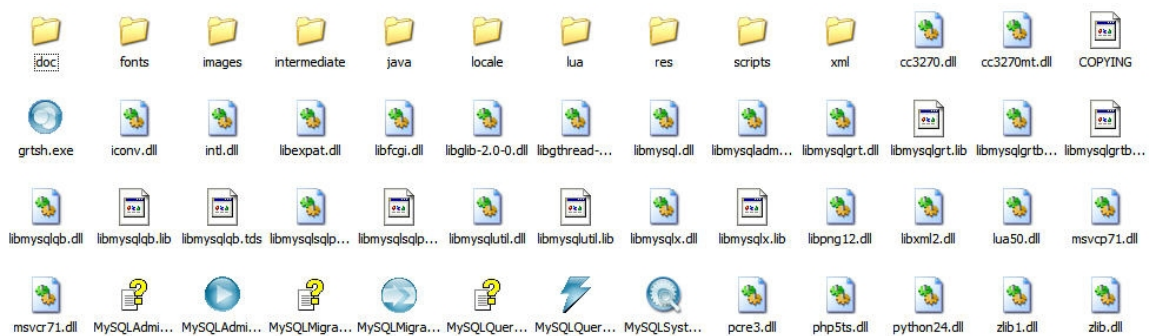
1. Setelah *mendownload* MySQL Query browser maka Anda akan mendapatkan sebuah file terkompresi yaitu **mysql-gui-tools-noinstall-5-r17-win32.zip** seperti yang tampak pada gambar dibawah ini.
2. Ekstak file **mysql-gui-tools-noinstall-5-r17-win32.zip** di alamat mana saja. Sebagai rekomendasi dapat di ekstrak di partisi **C:** pada Harddisk anda.



3. Dan jika proses ekstraksi berjalan sempurna maka akan menampilkan sebuah folder bernama **MySQL Gui Tools 5.0**

Name	Size	Type	Date Modified	Attribu...
MySQL GUI Tools 5.0		File Folder	2/11/2011 8:43 AM	

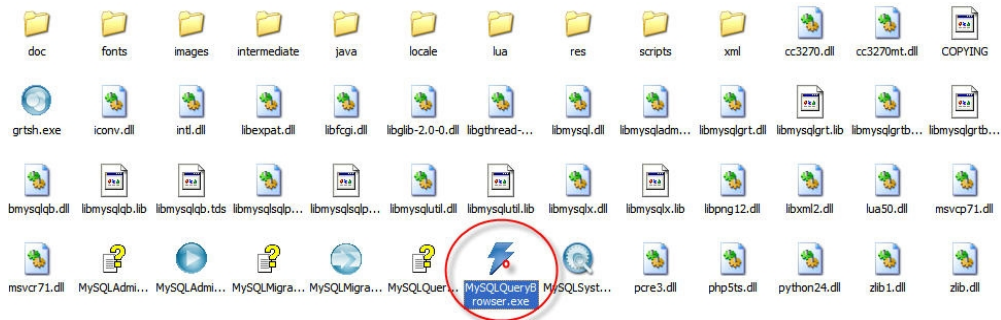
4. Apabila Folder **MySQL Gui Tools 5.0** di buka akan tampak seperti gambar dibawah ini.



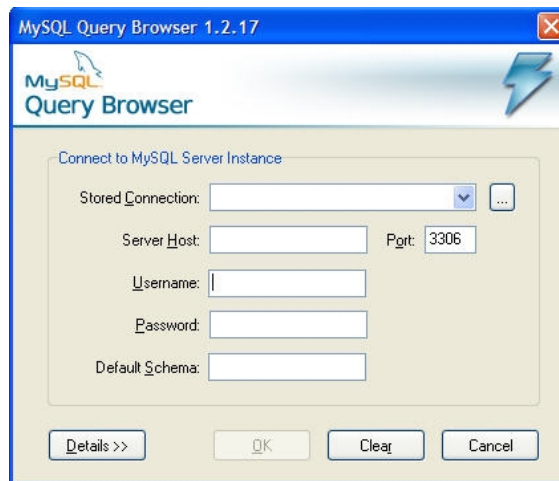
C. Koneksi ke database MySQL melalui MySQL Query Browser

Koneksi ke MySQL Server adalah sebuah cara untuk menghubungkan **MySQL Query Browser** ke Database Server MySQL. Berikut ini adalah cara **MySQL Query Browser** melakukan koneksi ke Database Server MySQL.

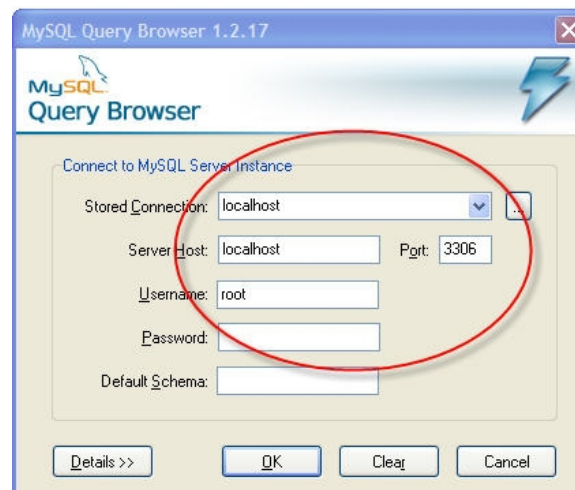
1. Pertama Anda masuk terlebih dahulu ke folder **MySQL Gui Tools 5.0** kemudian klik file bernama **MySQLQueryBrowser.exe**



2. Apabila **MySQLQueryBrowser.exe** telah di klik maka akan muncul tampilan seperti gambar dibawah ini.



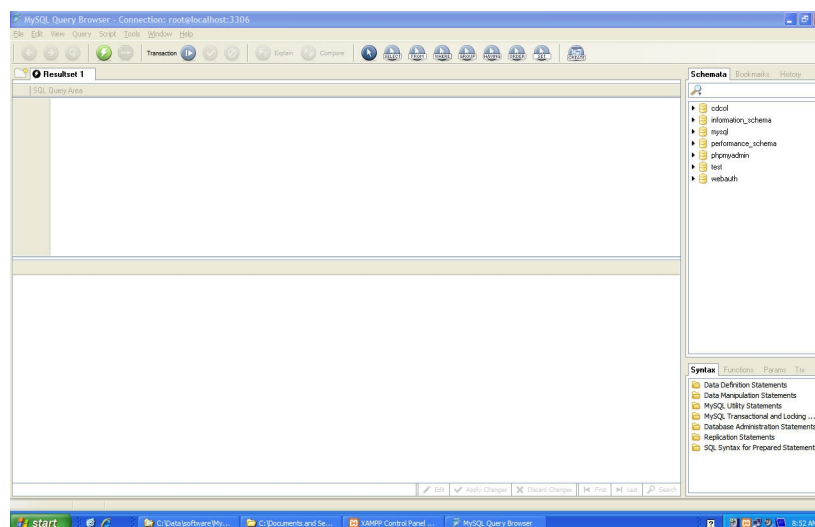
3. Kemudian isi inputan mengenai informasi yang diperlukan untuk melakukan koneksi Database Server MySQL.



Berikut ini adalah penjelasan mengenai Inputan diatas.

- **Store Connection** : Untuk memberi nama koneksi
- **Server Host** : Untuk menuliskan alamat komputer atau IP dari keberadaan Database MySQL Server, Anda dapat menuliskan *localhost* bila Database MySQL Server berada dikomputer yang sama dengan MySQL Query Browser.
- **Username** : Untuk menuliskan username yang dapat digunakan mengakses Database MySQL Server.
- **Password** : Untuk menuliskan password yang dapat digunakan mengakses Database MySQL Server. Password dapat dikosongkan bila Database MySQL Server mengosokan passwordnya.

4. Apabila koneksi ke **Database MySQL Server** berhasil maka akan tampak tampilan seperti gambar dibawah ini.



29 Fitur Dahsyat



2.1 Trigger

Salah satu feature yang terdapat di mysql versi 5.x.x adalah trigger yang secara harfiah dalam bahasa Indonesia adalah pemicu. Apabila Trigger diartikan sesuai dengan nama memang ada kaitannya dengan picu-memicu.

Tampaknya akan lebih mudah dimengerti apabila penjelasan Trigger menggunakan contoh kasus. Misalkan ada sebuah database yang didalamnya terdapat tabel untuk mencatat **transaksi penjualan barang harian** dan **rekapitulasi penjualan barang bulanan**, jadi ketika terjadi perubahan data (insert, update, delete) pada tabel **transaksi penjualan barang harian** secara otomatis akan update ke tabel **transaksi rekapitulasi barang bulanan**.

Berikut ini cara membuat Trigger, tahap pertama adalah membuat dua buah tabel, tabel pertama untuk menyimpan transaksi harian, dan yang table kedua untuk menyimpan rekapitulasi transaksi bulannya, di bawah ini adalah struktur kedua tabel tersebut

1. Membuat tabel **tbl_penjualan** untuk menyimpan transaksi penjualan harian

```
CREATE TABLE `tbl_penjualan` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `kode_barang` char(10) NOT NULL,  
  `tgl_transaksi` datetime NOT NULL,  
  `jml` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM AUTO_INCREMENT=10 DEFAULT CHARSET=latin1 COMMENT='tabel yang mencatat penjualan tiap hari'
```

2. Membuat tabel **tbl_penjualan_rekap_bulanan** untuk menyimpan rekap penjualan bulanan

```
CREATE TABLE `tbl_penjualan_rekap_bulanan` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `kode_barang` char(10) NOT NULL,  
  `tgl_transaksi` char(10) NOT NULL,  
  `jml` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `bulankode` (`tgl_transaksi`,`kode_barang`)  
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1 COMMENT='tabel yang rekap penjualan harian'
```

Selanjutnya adalah membuat Trigger untuk Insert Data, jadi dengan Trigger ini ketika terjadi penambahan data pada tabel **tbl_penjualan** maka secara otomatis pada tabel **tbl_penjualan_rekap_bulanan** akan terjadi update data (data yang di update adalah field jml), tetapi apabila belum terdapat data pada table **tbl_penjualan_rekap_bulanan** maka akan membuat row/baris/record baru pada table tersebut.

```
CREATE TRIGGER `rekap_bulan` AFTER INSERT ON `tbl_penjualan` FOR EACH ROW  
BEGIN  
  INSERT INTO `tbl_penjualan_rekap_bulanan` SET  
    `kode_barang`=NEW.`kode_barang`,  
    `tgl_transaksi`=date_format(NEW.`tgl_transaksi`, '%Y%m'),  
    `jml`=NEW.`jml`  
  ON DUPLICATE KEY UPDATE `jml`=`jml`+NEW.`jml`;  
END
```

Sekarang kita akan lakukan percobaan Insert Data ke tabel **tbl_penjualan** yang secara otomatis akan terjadi insert data ke table **tbl_penjualan_rekap_bulanan** yang berfungsi sebagai rekap bulan.

```

insert into tbl_penjualan values (null,'0000000001',now(),20);
insert into tbl_penjualan values (null,'0000000001',now(),30);
insert into tbl_penjualan values (null,'0000000001',now(),10);
insert into tbl_penjualan values (null,'0000000002',now(),5);
insert into tbl_penjualan values (null,'0000000002',now(),7);

```

Setelah proses percobaan Insert Data berhasil maka pada table **tbl_penjualan** dan **tbl_penjualan_rekap_bulanan** akan tampak seperti dibawah ini.

```

mysql> select * from tbl_penjualan ;
+----+-----+-----+-----+
| id | kode_barang | tgl_transaksi | jml |
+----+-----+-----+-----+
| 5  | 0000000001 | 2009-02-05 15:10:50 | 20 |
| 6  | 0000000001 | 2009-02-05 15:14:01 | 30 |
| 7  | 0000000001 | 2009-02-05 15:14:32 | 10 |
| 8  | 0000000002 | 2009-02-05 15:15:26 | 5  |
| 9  | 0000000002 | 2009-02-05 15:15:35 | 7  |
+----+-----+-----+-----+

```

```

mysql> select * from tbl_penjualan_rekap_bulanan;
+----+-----+-----+-----+
| id | kode_barang | tgl_transaksi | jml |
+----+-----+-----+-----+
| 2  | 0000000001 | 200902       | 60 |
| 3  | 0000000002 | 200902       | 12 |
+----+-----+-----+-----+

```

Selanjutnya membuat Trigger untuk Update Data, jadi dengan Trigger ini ketika ada perubahan data di tabel **tbl_penjualan** maka secara otomatis di tabel **tbl_pejualan_rekap_bulanan** akan terjadi update data (data yang di update adalah field jml)

```

CREATE TRIGGER `rekap_bulan_update` AFTER UPDATE ON `tbl_penjualan` FOR EACH ROW
BEGIN
UPDATE `tbl_penjualan_rekap_bulanan`
SET `jml` = `jml` + NEW.jml - OLD.jml
WHERE
`kode_barang` = NEW.`kode_barang` AND
`tgl_transaksi` = date_format(NEW.`tgl_transaksi`,`%Y%m`);
END;

```

Tahap selanjutnya membuat Trigger untuk Delete Data, jadi ketika ada data yang dihapus pada tabel **tbl_penjualan** maka secara otomatis di tabel **tbl_penjualan_rekap_bulanan** akan terjadi update data (data yang di update adalah field jml)

```

CREATE TRIGGER `rekap_bulan_delete` AFTER DELETE ON `tbl_penjualan` FOR EACH ROW
BEGIN
UPDATE `tbl_penjualan_rekap_bulanan`
SET `jml` = `jml` - OLD.jml
WHERE
`kode_barang` = OLD.`kode_barang` AND
`tgl_transaksi` = date_format(OLD.`tgl_transaksi`,`%Y%m`);
END;

```

Apabila ingin melihat Trigger yang telah dibuat dapat menggunakan perintah dibawah ini.

```
mysql> select trigger_name from information_schema.triggers;
```

```
+-----+  
| trigger_name |  
+-----+  
| rekap_bulan |  
| rekap_bulan_update |  
| rekap_bulan_delete |  
+-----+
```

2.2 Replikasi Database

Replikasi adalah sebuah proses untuk membuat salinan atau backup. Tujuan dari membuat Replikasi Database untuk membuat salinan/backup database dari satu mesin (pc, laptop atau semacamnya) ke mesin yang lain.

Perbedaannya Replikasi Database dan Backup Database terletak pada otomatisasi backup yang dilakukan secara realtime (ketika ada perubahan pada database utama maka secara langsung database salinannya akan mengikuti perubahannya).

Fitur replikasi ini baru ada di database MySQL versi 5.x.x. Eksperimen yang ditulis disini menggunakan MySQL versi 5.x.x. Spesifikasi database dan Operating System yang digunakan penulis adalah sebagai berikut.

a. Spesifikasi komputer/mesin master/sumber yang digunakan.

- Operating system Linux Ubuntu 8.10 (Intrepid)
- MySQL versi 5.1.22

b. Spesifikasi komputer/mesin slave/salinan yang digunakan.

- Operating system Linux Ubuntu 8.04 (HardyHeron)
- MySQL versi 5.0.51

Tahap pertama untuk membuat replikasi database adalah menyiapkan dua buah mesin (pc, laptop, atau semacamnya). Mesin pertama sebagai database utama atau disebut juga master dan mesin yang kedua untuk database salinan atau disebut slave.

Tahap kedua adalah melakukan instalasi database MySQL di kedua mesin tersebut (master dan slave), Spesifikasi versi MySQL dan Operating System di rekomendasikan sama atau versi MySQL di master lebih baru dari yang di slave.

Selanjutnya adalah membuat konfigurasi pada mesin master dan mesin slave, berikut ini adalah cara melakukan konfigurasi tersebut.

a. Konfigurasi di sisi Master

Database yang akan di replikasi kita asumsikan bernama **repli_db** (jadi database tersebut telah dibuat terlebih dahulu berikut dengan tabel-nya pada mesin master). Berikut ini adalah tahapan konfigurasi di sisi master.

1. Buka file **my.cnf** (biasanya terletak di `/etc/mysql/my.cnf`), yang merupakan file berisikan berbagai macam setting mysql.

2. Pada file **my.cnf** tambahkan setting master seperti dibawah ini.

```
log_bin = /var/log/mysql/mysql-bin.log
binlog-do-db = repli_db
server-id = 1
```

Berikut ini penjelasan dari setting diatas.

- **log_bin** adalah alamat untuk menyimpan log binari mysql
- **binlog-do-db** untuk nama database yang akan direplikasi
- **server-id** adalah id dari mesin master

3. Membuka akses agar yang bisa menggunakan database tidak hanya dari localhost saja atau ip 127.0.0.1. Masih pada file **my.cnf** tambahkan simbol kres (#) di baris bind-address seperti yang ada dibawah ini.

- Sebelum

```
bind-address = 127.0.0.1
```

- Sesudah

```
#bind-address = 127.0.0.1
```

4. Restart MySQL agar setting yang telah diberikan segera di eksekusi

```
sudo /etc/init.d/mysql restart;
```

5. Login ke MySQL sebagai user Root

```
sudo mysql -uroot -ppassword_root_mu
```

6. Menyiapkan akses slave untuk mesin slave agar dapat berhubungan dengan mesin master. Username yang diberikan untuk mesin slave adalah **slave_user** dengan password **pass**

```
GRANT REPLICATION SLAVE ON *.* TO 'slave_user'@'%' IDENTIFIED BY 'pass';  
FLUSH PRIVILEGES;
```

7. Memberikan akses agar username slave_user dapat menggunakan databases

```
GRANT ALL ON *.* TO 'slave_user'@'%' IDENTIFIED BY 'pass';  
FLUSH PRIVILEGES;
```

8. Catat status master dari database repli_db. Status master tersebut diperlukan untuk membuat setting pada mesin slave.

```
USE repli_db;  
FLUSH TABLES WITH READ LOCK;  
SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+  
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |  
+-----+-----+-----+-----+  
| mysql-bin.000119 | 268     | repli_db     |                   |  
+-----+-----+-----+-----+
```

```
UNLOCK TABLES;
```


b. Konfigurasi di sisi Slave

1. Buka file **my.cnf** (biasanya terletak di `/etc/mysql/my.cnf`), yang merupakan file berisikan berbagai macam setting MySQL.
2. Pada file **my.cnf** tambahkan setting slave agar dapat berkomunikasi dengan mesin master

```
server-id=2
master-host=192.168.10.91
master-user=slave_user
master-password=pass
master-connect-retry=30
replicate-do-db=repli_db
```

Berikut ini penjelasan dari setting diatas.

- **server-id** adalah id dari mesin slave
 - **master-host** adalah ip address komputer/mesin master
 - **master-user** adalah user name untuk akses komputer/mesin master
 - **master-password** adalah password untuk akses komputer/mesin master
 - **master-connect-retry** adalah waktu jeda (satuan detik) untuk melakukan koneksi ke master apabila connection ke master putus
 - **replicate-do-db** adalah nama database yang di replikasi
3. Restart MySQL agar setting yang telah diberikan segera di eksekusi

```
sudo /etc/init.d/mysql restart;
```

4. Login ke MySQL sebagai user Root

```
sudo mysql -uroot -ppassword_root_mu
```

5. Melakukan inisialisasi data atau mengambil data untuk yang pertama kalinya dari master ke slave.

```
LOAD DATA FROM MASTER;
```

Apabila proses inisialisasi berhasil maka pada mesin slave akan terbuat database **repli_db** beserta isinya (tabel dan datanya).

6. Sekarang memberikan setting replikasi databasenya (masih berada di console mysql)

```
SLAVE STOP;  
CHANGE MASTER TO MASTER_HOST='192.168.10.91',  
MASTER_USER='slave_user',  
MASTER_PASSWORD='pass',  
MASTER_LOG_FILE='mysql-bin.000119',  
MASTER_LOG_POS=268;  
SLAVE START;
```

Isi dari parameter **MASTER_LOG**, **MASTER_LOG_FILE** dan **MASTER_LOG_POS** berasal dari status master yang didapat di mesin/komputer master.

7. Untuk melihat status Slave dapat menggunakan perintah dibawah ini

```
SHOW SLAVE STATUS;
```

2.3 View

View salah satu feature yang terdapat pada mysql 5.0. Terkadang view disebut tabel temporari atau virtual tabel. View berguna untuk menyederhanakan pengambilan data dari sebuah query yang rumit, misalkan pada query tersebut terdapat join, grup by, agregasi (sum, min, max dsb) sub query, union dsbnya.

Kasus yang akan diambil adalah tentang nilai siswa, dimana pada databases tersebut terdapat tiga buah tabel.

- Tabel pertama untuk menyimpan biodata siswa
- Tabel kedua untuk menyimpan mata pelajaran
- Tabel ketiga untuk menyimpan nilai

Berikut ini adalah SQL DDL (Data Definition Language) dari ketiga buah tabel tersebut.

1. tbl_siswa (tabel untuk menyimpan biodata siswa)

```
CREATE TABLE `sisfo_sekolah`.`tbl_mp` (  
  `mp` char(10) NOT NULL,  
  `nama` varchar(255) NOT NULL,  
  PRIMARY KEY (`mp`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

2. tbl_mp (tabel untuk menyimpan mata pelajaran)

```
CREATE TABLE `sisfo_sekolah`.`tbl_mp` (  
  `mp` char(10) NOT NULL,  
  `nama` varchar(255) NOT NULL,  
  PRIMARY KEY (`mp`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

3. tbl_nilai (tabel untuk menyimpan nilai2 siswa)

```
CREATE TABLE `sisfo_sekolah`.`tbl_nilai` (  
  `kode` varchar(255) NOT NULL,  
  `nis` char(10) NOT NULL,  
  `mp` char(10) NOT NULL,  
  `nilai` int(11) NOT NULL,  
  PRIMARY KEY (`kode`),  
  KEY `nis_fk_constraint` (`nis`),  
  KEY `mp_fk_constraint` (`mp`),  
  CONSTRAINT `mp_fk_constraint` FOREIGN KEY (`mp`) REFERENCES `tbl_mp` (`mp`) ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `nis_fk_constraint` FOREIGN KEY (`nis`) REFERENCES `tbl_siswa` (`nis`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Ketiga tabel diatas adalah tabel-tabel yang dibutuhkan untuk menghasilkan nilai-nilai siswa dari setiap mata pelajaran.

Sekarang kita akan membuat sebuah report nilai yang berisikan data nis, nama_siswa, nama_matapelajaran dan nilai. Untuk menghasilkan report tersebut akan dibuat sebuah view dengan nama **view_nilai**. Berikut ini adalah cara membuat view.

```
CREATE VIEW `view_nilai`  
AS select `n`.`nis` AS `nis`,  
  `n`.`nilai` AS `nilai`,  
  `s`.`nama` AS `nama_siswa`,  
  `m`.`nama` AS `mata_pelajaran`  
from ((`tbl_nilai` `n` join `tbl_siswa` `s` on((`n`.`nis` = `s`.`nis`))) join `tbl_mp` `m` on((`n`.`mp` =  
  `m`.`mp`)))
```

Setelah **view_nilai** tercipta dengan query yang cukup kompleks seperti diatas, maka untuk mendapatkan report nilai cukup memanggil **view_nilai** seperti di bawah ini.

```
select * from view_nilai
```

2.4 Store Procedure

Store Procedure adalah salah satu fitur yang terdapat di mysql 5.0. Store procedure sendiri adalah sekumpulan script sql berupa routine yang disimpan dalam database MySQL Server. Manfaat dari menggunakan Store Procedure ini adalah sebagai berikut.

1. Memperkecil trafik request dari aplikasi ke database karena semua logika program dilakukan di database mysql dan aplikasi hanya menerima hasil proses saja.
2. Agar aplikasi menjadi lebih portable. Ketika sebuah aplikasi yang dikembangkan terdiri dari berbagai bahasa pemrograman (misalkan desktop dan web) yang mana kedua tipe aplikasi tersebut membutuhkan hasil dan operasi database yang sama.
3. Meningkatkan keamanan aplikasi, misalkan sebuah aplikasi yang login dengan user xxx tidak dapat mengakses tabel-tabel tertentu secara langsung tetapi harus melalui Store Procedure, dengan cara ini seperti ini dapat meningkatkan data yang valid.

Dalam buku ini kita akan membuat Store Procedure yang di implementasikan pada kasus Sistem Informasi Sekolah. Store Procedure yang akan dibuat memiliki tujuan menghasilkan data siswa yang di filter berdasarkan kelas.

Berikut ini adalah struktur tabel siswa yang digunakan untuk menyimpan seluruh data siswa

```
CREATE TABLE `data_siswa`.`tbl_siswa` (  
  `nis` char(10) NOT NULL,  
  `nama` varchar(255) NOT NULL,  
  `kelas` int(11) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Berikut ini cara membuat Store Procedure untuk menampilkan data siswa

```
DELIMITER $$  
CREATE PROCEDURE sp_tampil_siswa_kelas(p_kelas int)  
BEGIN  
  SELECT * FROM tbl_siswa where KELAS = p_kelas;  
END$$  
DELIMITER ;
```

Penjelasan dari script diatas adalah sebagai berikut

1. **DELIMITER** adalah untuk memberikan deklarasi kepada MySQL bahwa delimiter/pemisah yang digunakan untuk menunjukkan akhir dari sebuah statemen sql adalah simbol \$\$.
2. **CREATE PROCEDURE** adalah header untuk membuat Store Procedure. Nama dari Store Procedure tersebut adalah **sp_tampil_siswa_kelas**.
3. **BEGIN END** adalah penanda awal dan akhir dari isi Store Procedure, jadi semua sql di tulis dalam blok ini.

Berikut ini adalah cara penggunaan dari Store Procedure **sp_tampil_siswa_kelas** yang telah kita buat.

```
call sp_tampil_siswa_kelas(2);
```

Pada script pemanggilan Store Procedure **sp_tampil_siswa_kelas** diatas akan menghasilkan seluruh data siswa kelas dua.

2.5 Function

Function adalah salah satu fitur MySQL. Function adalah sekumpulan sql berupa routine yang di simpan dalam database MySQL Server. Penggunaan function dapat dikombinasikan atau tidak di kombinasikan dengan Store Procedure.

Function yang akan dibuat pada contoh ini akan diimplementasikan pada sebuah kasus sistem informasi sekolah. Function yang dibuat akan berperan untuk memberikan jumlah siswa dari setiap kelas.

Berikut ini adalah sturuktur tabel siswa yang digunakan untuk menyimpan seluruh data siswa

```
CREATE TABLE `data_siswa`.`tbl_siswa` (  
  `nis` char(10) NOT NULL,  
  `nama` varchar(255) NOT NULL,  
  `kelas` int(11) NOT NULL  
  ) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Berikut ini cara membuat Function untuk menghitung jumlah siswa di setiap kelas


```

DELIMITER $$
CREATE FUNCTION sf_tampil_siswa_kelas (p_kelas int) RETURNS INT DETERMINISTIC
BEGIN
  DECLARE jml INT;
  SELECT COUNT(*) AS jml_kelas INTO jml FROM tb_siswa WHERE kelas = p_kelas;
  RETURN jml;
END$$
DELIMITER ;

```

Penjelasan dari script diatas adalah sebagai berikut

1. **DELIMITER** adalah untuk memberikan deklarasi kepada Mysql bahwa delimiter/pemisah yang digunakan untuk menunjukkan akhir dari sebuah statemen sql adalah simbol \$\$.
2. **CREATE FUNCTION** adalah header untuk membuat Store Procedure. Nama dari Function tersebut adalah **sf_tampil_siswa_kelas**.
3. **RETURNS** adalah untuk menentukan tipe data yang di returnkan oleh function
4. **DETERMINISTIC/ NOT DETERMINISTIC** adalah untuk menentukan yang bisa menggunakan function ini adalah user pembuatnya saja (determinisric) atau user apa saja (not determinisric).
5. **BEGIN END** adalah penanda awal dan akhir dari isi Function, jadi semua sql di tulis dalam blok ini.

Berikut ini adalah cara penggunaan dari Function **sf_tampil_siswa_kelas** yang telah kita buat.

```
select sf_tampil_siswa_kelas("2");
```

Pada script pemanggilan Function **sf_tampil_siswa_kelas** diatas akan menghasilkan return jumlah siswa dari kelas dua. Sebuah function hanya bisa memberikan return berupa nilai dan tidak bisa berupa resultset.

Secara default ketika membuat function adalah NOT DETERMINISTIC, untuk memberikan DETERMINISTIC pada function harus dituliskan secara eksplisit.

Apabila Anda ingin ketika membuat sebuah function secara default adalah DETERMINISTIC maka Anda bisa melakukan sedikit setting dengan mengetikkan perintah dibawah ini pada console mysql.

```
SET GLOBAL log_bin_trust_function_creators = 1;
```

2.6 Backup dan Restore Database

Proses backup data pada sebuah database server secara berkala adalah sebuah prosedur umum untuk menjaga ketersediaan data. Apabila terjadi kerusakan data atau kehilangan data yang dapat disebabkan harddisk rusak atau komputer hilang.

Backup data dapat dikatakan sebuah maintance/perawatan database. Pada database MySQL terdapat fitur untuk melakukan Backup dan Restore database dari yang sederhana hingga berbagai

macam option yang tersedia agar lebih customizable.

Berikut ini adalah cara melakukan backup database, sebagai contoh nama database yang akan di backup adalah **db_test**. Pada console sistem operasi Anda ketikkan perintah seperti dibawah ini .

```
mysqldump -uroot -p1234 -R db_test > db_test.sql
```

Berikut ini adalah cara melakukan restore database, sebagai contoh nama database yang akan di restore adalah **db_test**. Pada console sistem operasi Anda ketikkan seperti dibawah ini .

```
mysqldump -uroot -p1234 db_test < db_test.sql
```

Berikut ini adalah penjelasan dari script diatas

- **mysqldump** adalah command untuk backup dan restore
- **-u** adalah username untuk akses ke database
- **-p** adalah password untuk akses ke database
- **-R** adalah proses backup yang menyertakan routine (triger, function, store procedure, dsb) apabila tidak memberikan option **-R** maka yang akan di backup hanya data dan struktur table .
- **db_test** adalah contoh nama database yang akan di backup dan restore
- **simbol >** adalah proses backup database
- **simbol <** adalah proses restore database
- **db_test.sql** adalah hasil backup yang berupa file berisikan SQL

Apabila yang perlu di backup hanya data dapat menggunakan perintah seperti ini

```
mysqldump -uroot -p1234 --skip-triggers --no-create-info db_test > db_test.sql
```

- **--skip-triggers** adalah option agar proses backup tidak menyertakan trigger
- **--no-create-info** adalah optiion agar proses backup tidak menyertakan struktur table

Apabila yang perlu di backup hanya struktur table, dapat menggunakan perintah seperti ini

```
mysqldump -uroot -p1234 --no-data db_test > db_test.sql
```

- **--no-data** adalah option agar proses backup tidak menyertakan data

2.7 Transaction

Mungkin Anda pernah mengalami dalam pemrograman komputer terdapat sebuah kondisi yang didalam satu proses terdiri dari banyak query yang di eksekusi dalam satu waktu, Misalkan terdiri dari tiga buah sql insert data ke table yang berbeda, dua buah sql update data pada table berbeda dan satu buah sql delete data, apabila dari salah satu dari query tersebut error maka proses tersebut menjadi tidak valid.

Sejak MySQL versi 3.23, MySQL memiliki fitur untuk menangani hal seperti di atas, nama dari fitur tersebut adalah MySQL Transaction. Dalam transaction apabila terjadi error/problem pada sebuah query maka semua perubahan yang terjadi sebelum kejadian error akan diabaikan dan query selanjutnya tidak akan di eksekusi, penanganan ini dinamakan dengan Roolback.

Penggunaan MySQL Transaction memiliki pola penulisan yang harus diikuti agar fitur dari transaction dapat berfungsi, berikut ini adalah pola penulisan transaction yang biasa disebut dengan Blok Transaction.

```
START TRANSACTION;  
[BLOK QUERY]  
COMMIT/ROOLBACK;
```

Untuk menggunakan fitur Transaction, harus dimulai dengan **Start Transaction** dan di akhir dengan **Commit** atau **Roolback**. Commit adalah sebuah perintah untuk menyimpan perubahan secara fisik ke database apabila tidak terdapat error pada seluruh query yang terdapat didalam blok transaction, sedangkan Rollback adalah perintah yang akan mengabaikan semua perubahan apabila terdapat error pada blok transaction.

Fitur Transaction hanya dapat digunakan pada table yang bertipe Store Engine transaction seperti innodb.

Berikut ini saya akan contohkan penggunaan transaction dalam kasus pemesanan barang. Setiap proses pemesan barang melibatkan table master yang berfungsi untuk menyimpan data-data pemesan barang dan table detil untuk menyimpan data barang yang dipesan, query yang diperlukan dari proses pemesanan barang adalah untuk menyimpan ke table master dan ke table detil.

Berikut ini adalah struktur table bernama **tbl_pesanan_mst** untuk menyimpan data pemesan

```
CREATE TABLE `db_test`.`tbl_pesanan_dtl` (  
  `kode_dtl` char(10) NOT NULL,  
  `kode_mst` char(10) NOT NULL,  
  `kode_brg` char(10) NOT NULL,  
  `jml` int(11) NOT NULL,  
  PRIMARY KEY (`kode_dtl`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='tabel untuk menyimpan pesanan detail'
```

Berikut ini adalah struktur table bernama **tbl_pesanan_detail** untuk menyimpan data barang yang dipesan.

```
CREATE TABLE `db_test`.`tbl_pesanan_mst` (  
  `kode` char(10) NOT NULL,  
  `nama` varchar(255) NOT NULL,  
  `alamat` text NOT NULL,  
  `status` enum('0','1') NOT NULL COMMENT '0=belum dikirm, 1=sudah dikirim',  
  PRIMARY KEY (`kode`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='tabel untuk menyimpan data-dat pesanan'
```

Dibawah ini adalah contoh penggunaan transaction untuk kasus pemesanan barang

```

START TRANSACTION;
insert into tbl_pesanan_dtl values ('PSNDTL0001','PSN0000001','BRG0000001',10);
insert into tbl_pesanan_dtl values ('PSNDTL0002','PSN0000001','BRG0000002',20);
insert into tbl_pesanan_dtl values ('PSNDTL0003','PSN0000001','BRG0000003',30);
insert into tbl_pesanan_dtl values ('PSNDTL0004','PSN0000001','BRG0000004',40);
insert into tbl_pesanan_dtl values ('PSNDTL0005','PSN0000001','BRG0000005',50);

insert into tbl_pesanan_mst values ('PSN0000001','Dendie','Jalan Asmi No 18 Bandung',1);
COMMIT;

```

Statemen query di atas disimpan ke database secara fisik apabila tidak terjadi satupun query yang error atau terjadi problem. Perintah Start Transaction sebagai awal dari blok transaction dan Commit adalah perintah untuk menyimpan perubahan ke database secara fisik.

Contoh diatas baru bagian dasar dari transaction, selanjutnya kita akan melengkapi script di atas yang apabila terjadi error diantara blok transaction maka secara otomatis akan melakukan Rollback tetapi bila tidak terjadi error maka perintah Commit yang akan dieksekusi.

Penggunaan Commit dan Roolback dari blok transaction akan diimplementasikan didalam Store Procedure.

```

DELIMITER $
CREATE PROCEDURE sp_input_pesanan()
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLWARNING, NOT FOUND ROLLBACK;
START TRANSACTION;
insert into tbl_pesanan_dtl values ('PSNDTL0001','PSN0000001','BRG0000001',10);
insert into tbl_pesanan_dtl values ('PSNDTL0002','PSN0000001','BRG0000002',20);
insert into tbl_pesanan_dtl values ('PSNDTL0003','PSN0000001','BRG0000003',30);
insert into tbl_pesanan_dtl values ('PSNDTL0004','PSN0000001','BRG0000004',40);
insert into tbl_pesanan_dtl values ('PSNDTL0005','PSN0000001','BRG0000005',50);

insert into tbl_pesanan_mst values ('PSN0000001','Dendie','Jalan Asmi No 18 Bandung','0');
COMMIT;
END$
DELIMITER ;

```

Berikut ini adalah penjelasan dari script diatas

- `DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLWARNING, NOT FOUND ROLLBACK;` adalah untuk menangani bila terjadi error, warning dan proses-proses yang abnormal maka akan dilakukan perintah Roolback.
- `START TRANSACTION` adalah awalan dari blok transaction
- `COMMIT` adalah perintah untuk menyimpan perubahan ke database secara fisik apabila tidak terdapat error.

Apabila kita telah membuat blok transaction dalam store procedure maka untuk implementasi transaction dalam level aplikasi cukup dengan cara memanggil store procedure tersebut.

Untuk menggunakan transcation pada level aplikasi selain dengan cara memanggil store procesure dapat menggunakan native bahasa pemrograman, berikut ini saya contohkan dengan menggunakan scripting PHP.


```
<?php
$dbh = 'database_contoh';
mysqli_autocommit($dbh, FALSE);

$result=mysqli_query($dbh, $query1);
if ($result !== TRUE) {
    mysqli_rollback($dbh); //kalau ada error di query 1 akan di rollback
}

$result=mysqli_query($dbh, $query2);
if ($result !== TRUE) {
    mysqli_rollback($dbh); //kalau ada error di query 2 akan di rollback
}

//kalau tidak ada error maka akan di commit
mysqli_commit($dbh);

mysqli_close($dbh);
?>
```

2.8 Database Firewall

Sebuah aplikasi yang dipasang pada jaringan internet yang jumlah pengunjungannya sulit terkontrol memiliki resiko keamanan aplikasi yang cenderung meningkat. Percobaan yang sering dilakukan oleh para user jahil maupun jahat adalah Sql Injection.

Sql Injection adalah sebuah aksi hacking pada aplikasi web base dengan cara memodifikasi atau menyisipkan perintah-perintah sql pada parameter yang dikirim dari client (web-browser) ke server.

Salah satu solusi yang dapat dicoba untuk menangani Sql Injection dengan menggunakan GreenSQL. GreenSQL adalah sebuah open source Firewall Database yang berfungsi sebagai solusi keamanan database MySql dan PosgreSql antara lain untuk menghadapi serangan Sql Injection.

Secara arsitektur GreenSQL berada pada aplikasi berada di antara aplikasi dengan database server, apabila digambarkan akan tampak seperti dibawah ini.



Arisitektur GreenSql

Berikut ini adalah tahapan-tahapan dalam implementasi GreenSql pada sebuah aplikasi berbasis web.

1. Download GreenSql di <http://www.greensql.net/download> sesuai dengan sistem operasi yang digunakan, saat ini yang tersedia adalah untuk windows, ubuntu dan centos. Pada tulisan ini sistem operasi yang digunakan adalah **ubuntu 8.10**.

Pada GreenSql terbagi menjadi dua bagian utama yaitu.

- GreenSQL Firewall adalah program yang berfungsi sebagai firewall
- Management Console adalah software berbasis web untuk untuk monitoring firewall yang dibuat menggunakan php.

2. Instalasi GreenSQL Firewall melalui console, ketikan perintah seperti dibawah ini.

```
dpkg -i greensql-fw_1.0.0_i386.deb
```

3. Setelah proses instalasi selesai selanjutnya lakukan konfigurasi pertama GreenSql, ketikkan perintah dibawah ini pada console. Perintah dibawah ini akan membuat sebuah database untuk keperluan GreenSql.

```
sudo /usr/sbin/greensql-create-db.sh
```

4. Untuk melakukan konfigurasi GreenSql selanjutnya berada pada sebuah file yang berlokasi dibawah ini.

```
/etc/greensql/greensql.conf
```

5. Selanjutnya adalah menuliskan pola-pola query yang dilarang berada di alamat dibawah ini.

```
/etc/greensql/mysql.conf
```

6. Setelah konfigurasi dan menentukan pola query telah selesai saatnya menghidupkan GreenSql.

```
sudo /etc/init.d/greensql start
```

7. Apabila terjadi masalah dari service GreenSql dapat dilihat log GreenSql yang terletak pada alamat dibawah ini.

```
var/log/greensql.log
```

8. Untuk memeriksa service GreenSql telah aktif dapat menggunakan cara dibawah ini.

```
mysql -h 127.0.0.1 -P 3305 -uroot -p pass
```

Service GreenSql secara default beroperasi dengan menggunakan port 3305, hal tersebut dapat dilihat pada parameter **-P 3305**

Untuk monitoring Firewall GreenSql dengan menggunakan software Management Console yang berada di paket download GreenSql. Ekstrak file bernama greensql-console-0.5.0.tar.gz di folder www atau htdocs dari web server yang Anda gunakan.

Setelah proses ekstraksi greensql-console-0.5.0.tar.gz selesai software Management Console dapat dites dengan mengetikkan <http://127.0.0.1/green-0.5.0/login.php> pada web browser (IE, Firefox, Opera, Chrome, dsb). Secara default login software Management Console adalah sebagai berikut.

- **Username** : admin
- **Password** : pwd

Berikut ini adalah contoh dalam menggunakan GreenSql menggunakan php.

- Cara koneksi default php ke database mysql

```
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
```

- Cara koneksi default php ke database mysql melalui greenSql, perbedaannya adalah php melakukan koneksi ke database mysql melalui port 3305 yang merupakan lokasi operasi dari GreenSql Firewall beroperasi.

```
$link = mysql_connect('127.0.0.1:3305', 'mysql_user', 'mysql_password');
```

2.9 Partition Table

Secara definisi partisi table adalah sebuah cara untuk memecah tabel menjadi beberapa bagian/segmen, tujuan dari menggunakan partisi adalah untuk mempercepat proses query dalam mencari dan mengumpulkan data.

Misalkan kita memiliki data sebanyak 10 juta record, apabila pada table konvensional maka data sebanyak itu akan di simpan dalam satu segmen, tetapi bila menggunakan partisi maka data sebanyak itu akan di pecah-pecah ke banyak partisi berdasarkan sebuah kondisi misalkan berdasarkan tanggal, maka ketika melakukan query pencarian atau pengumpulan data hanya melakukan scanning ke segmen dimana data itu berada sehingga menjadikan proses query menjadi lebih cepat.

Fitur partisi tabel ini telah hadir dari MySQL versi 5.1. sedangkan MySQL yang digunakan pada tulisan ini adalah MySQL versi 5.1.31 dengan sistem operasi Ubuntu 9.04 Jaunty Jackalope .

Sebelum melakukan partisi tabel sebaiknya terlebih dahulu memeriksa database untuk mengetahui apakah support partisi atau

tidak, berikut ini adalah untuk memeriksanya.

1. Masuk terlebih dahulu ke console mysql, seperti contoh dibawah ini

```
mysql -uroot -p1234
```

2. Setelah masuk kedalam console mysql ketikkan perintah dibawah ini. Apabila database support partition maka akan menghasilkan value YES.

```
SHOW VARIABLES LIKE '%partition%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_partitioning | YES  +
+-----+-----+
1 row in set (0.06 sec)
```

Sebelum membahas cara membuat partisi, saya akan menjelaskan beberapa tipe dari partisi table.

1. **Partisi Range** adalah membuat partisi berdasarkan sebuah range, misalkan berdasarkan tanggal atau sebuah nilai, seperti di bawah ini.

```
CREATE TABLE tbl_penjualan (
  id INT NOT NULL,
  nama VARCHAR(30),
  tgl_transaksi DATE NOT NULL DEFAULT '1970-01-01'
)
PARTITION BY RANGE ( YEAR(tgl_transaksi) ) (
  PARTITION p0 VALUES LESS THAN (1990),
  PARTITION p1 VALUES LESS THAN (1995),
  PARTITION p2 VALUES LESS THAN (2000),
  PARTITION p3 VALUES LESS THAN MAXVALUE
);
```

Contoh di atas membuat partisi berdasarkan tahun transaksi, jadi transaksi yang dari tahun 1990 akan masuk ke partisi p0, kemudian dari tahun 1995 akan masuk ke partisi p1, kemudian yang dari tahun 2000 akan masuk ke partisi p2 dan selebihnya akan masuk ke partisi p3.

2. **Partisi List** adalah membuat partisi berdasarkan kondisi dari sebuah nilai yang sudah di tentukan, contohnya seperti dibawah ini .

```
CREATE TABLE tbl_pegawai (  
  kode_pegawai VARCHAR(30),  
  kode_golongan INT,  
  nama_lengkap VARCHAR(255)  
)  
PARTITION BY LIST(kode_golongan) (  
  PARTITION p0 VALUES IN (1, 2, 3),  
  PARTITION p1 VALUES IN (4, 5, 6)  
);
```

Contoh diatas membuat partisi berdasarkan golongan pegawai, jadi pegawai dengan golongan 1,2 dan 3 akan masuk ke partisi p0 sedangkan karyawan dengan golongan 4,5 dan 6 akan masuk ke partisi p1 .

3. **Partisi Hash** adalah membuat partisi berdasarkan sebuah kolom yg sudah ditentukan dengan pembagian segmennya diatur oleh MySQL, contohnya seperti dibawah ini.

```
CREATE TABLE tbl_pegawai (  
  kode_pegawai VARCHAR(30),  
  kode_golongan INT,  
  nama_lengkap VARCHAR(255)  
)  
PARTITION BY HASH( kode_pegawai )  
PARTITIONS 4
```

contoh diatas membuat partisi berdasarkan kode pegawai, jadi data pegawai akan di pecah menjadi empat partisi, dan yg mengatur sebuah kode pegawai masuk ke partisi yang mana langsung diatur otomatis oleh MySQL

4. **Partisi Key** adalah membuat partisi berdasarkan sebuah key dari tabel, seperti dibawah ini.

```
CREATE TABLE tbl_nama (  
  id INT NOT NULL PRIMARY KEY,  
  nama_lengkap VARCHAR(255)  
)  
PARTITION BY KEY()  
PARTITIONS 2;
```

Contoh diatas membuat partisi berdasarkan primary key yang bernama id, jadi data dari tbl_nama akan di pecah menjadi dua partisi dan yang mengatur sebuah data masuk ke partisi yang mana dilakukan otomatis oleh MySQL.

Untuk melihat jumlah partisi pada sebuah table dapat menggunakan perintah explain partitions. Dibawah ini adalah table yang akan diberikan perintah explain partitions.


```
CREATE TABLE anggota (
  username VARCHAR(16) NOT NULL,
  namalengkap VARCHAR(25) NOT NULL,
  tanggal DATE NOT NULL,
  PRIMARY KEY(username)
)
PARTITION BY KEY(username)
PARTITIONS 6;
```

Untuk memeriksa jumlah partisi dari table diatas yang bernama anggota adalah sebagai berikut.

```
explain partitions select * from anggota;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	anggota	p0,p1,p2,p3,p4,p5	ALL	NULL	NULL	NULL	NULL	99570	

1 row in set (0.00 sec)

Hasil diatas menunjukan terdapat enam buah partisi yang ditunjukkan pada kolom partitions. Selanjutnya untuk melihat sebuah data berada didalam partisi yang mana dapat dengan cara dibawah ini.

```
explain partitions select * from anggota where username='50000' ;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	anggota	p4	const	PRIMARY	PRIMARY	18	const	1	

1 row in set (0.00 sec)

Perintah explain partitions diatas menunjukan bahwa data username dengan nilai 50000 berada di partisi p4 yang ditampilkan pada kolom partitions.

Selanjutnya saya akan membuat sebuah pengujian kecepatan proses query antara tabel tanpa partisi, dengan tabel yang di partisi menggunakan tipe key dengan jumlah data eksperimen sebanyak 100.000 record.

1. Table tanpa partisi.

```
CREATE TABLE anggota
  username INT NOT NULL,
  namalengkap VARCHAR(25) NOT NULL,
  tanggal DATE NOT NULL,
  PRIMARY KEY(username)
```

2. Table dengan partisi tipe key.

```
CREATE TABLE anggota2 (
  username INT NOT NULL,
  namalengkap VARCHAR(25) NOT NULL,
  tanggal DATE NOT NULL,
  PRIMARY KEY(username)
)
PARTITION BY KEY(username)
PARTITIONS 6;
```

3. Table dengan partisi tipe range.

```
CREATE TABLE anggota3 (
  username INT NOT NULL,
  namalengkap VARCHAR(25) NOT NULL,
  tanggal DATE NOT NULL,
  PRIMARY KEY(username)
)
PARTITION BY RANGE(username)
(
  PARTITION P0 VALUES LESS THAN (20000),
  PARTITION P1 VALUES LESS THAN (40000),
  PARTITION P2 VALUES LESS THAN (60000),
  PARTITION P3 VALUES LESS THAN (80000),
  PARTITION P4 VALUES LESS THAN (100001)
)
```

Cara pengukuran pengujian adalah dengan sebuah query select sederhana dan yang menjadi parameter pengujiannya adalah waktu yang di butuhkan dalam eksekusi query tersebut, dibawah ini adalah hasil dari pengujian tersebut.

1. Hasil pengujian pada table tanpa partisi menghasilkan waktu eksekusi query **0.15 sec.**

```
Select count(*) from anggota where username > 25000 and username < 85000
+-----+
| count(*) |
+-----+
|  59999 |
+-----+
1 row in set (0.15 sec)
```

2. Hasil pengujian pada table dengan menggunakan partisi key menghasilkan waktu eksekusi query **0.20 sec.**

```
Select count(*) from anggota2 where username > 25000 and username < 85000
+-----+
| count(*) |
+-----+
|  59999 |
+-----+
1 row in set (0.20 sec)
```

3. Hasil pengujian pada table dengan menggunakan partisi range menghasilkan waktu eksekusi query **0.07 sec.**

```
Select count(*) from anggota3 where username > 25000 and username < 85000
+-----+
| count(*) |
+-----+
| 59999 |
+-----+
1 row in set (0.07 sec)
```

Pengujian tersebut mencatat waktu eksekusi terlama adalah tabel yang menggunakan partisi key (0.20 sec) dan yang tercepat adalah tabel yang menggunakan partisi range (0.07).

2.10 MySQL Event Scheduler

Pada sebuah aplikasi terkadang membutuhkan sebuah query yang harus dieksekusi secara periodik secara otomatis, misalkan setiap satu hari sekali atau satu bulan sekali yang biasa disebut dengan penjadwalan.

Apabila Anda pengguna sistem operasi linux untuk keperluan tersebut dapat menggunakan service Job Cron kalau pengguna sistem operasi Windows dapat menggunakan fitur Schedule Task.

Pada MySQL versi 5.1 telah memiliki fitur untuk melakukan keperluan penjadwalan yang diberi nama Event Scheduler. Dalam tulisan ini database MySQL yang digunakan oleh penulis adalah versi 5.1.31 dengan sistem operasi Ubuntu Jaunty.

Sebelum memulai membuat Event Scheduler sebaiknya memeriksa fitur Event Schedulennya hidup atau tidak, berikut ini cara memeriksa fitur Event Scheduler. Apabila kolom value bernilai ON

mengartikan fitur Event Scheduler pada database MySQL telah hidup.

```
SHOW VARIABLES LIKE 'event_scheduler'
+-----+-----+
| Variable_name | Value |
+-----+-----+
| event_scheduler | ON |
+-----+-----+
1 row in set (0.02 sec)
```

Berikut cara menghidupkan MySQL Event Schedule secara manual, ketikkan perintah seperti dibawah ini pada console mysql.

```
SET GLOBAL event_scheduler = 1;
```

Dan berikut cara mematikan MySQL Event Schedule secara manual, ketikkan perintah seperti dibawah ini pada console mysql.

```
SET GLOBAL event_scheduler = 0;
```

MySQL Event Schedule terdiri dari beberapa macam tipe. Sebelum menjelaskan tentang tipe MySQL Event saya akan membuat terlebih dahulu tabel dengan nama tbl_sch untuk keperluan demo Event Schedule.

```
CREATE TABLE tbl_sch (
  no int(11) NOT NULL AUTO_INCREMENT,
  des varchar(255) DEFAULT NULL,
  time datetime NOT NULL,
  PRIMARY KEY (no)
)
```

1. Event Schedule berdasarkan tanggal dan waktu yg sudah di tentukan, berikut ini adalah pola untuk membuat Event Schedule tersebut.

```
CREATE EVENT event_name
ON SCHEDULE
AT {DATE AND TIME}
DO
{SQL COMMAND}
```

Dibawah ini adalah contoh membuat insert data ke table_sch yang akan dieksekusi pada tanggal 1 juni 2009 jam 18:06:49 dengan menggunakan Event Schedule.

```
CREATE EVENT e_sch_1
ON SCHEDULE AT '2009-06-01 18:06:49'
DO insert into tbl_sch values (null, 'sch 1', now())
```

Berikut ini adalah hasil dari eksekusi Event Schedule tersebut.

```
select * from tbl_sch;
+----+-----+-----+
| no | des | time |
+----+-----+-----+
| 1 | sch 1 | 2009-06-01 18:06:49 |
+----+-----+-----+
```

2. Event Schedule berdasarkan pengulangan, berikut ini adalah pola untuk membuat Event Schedule tersebut.

```
CREATE EVENT event_name
ON SCHEDULE
EVERY {x}
{SECOND | MINUTE | HOUR | DAY | MONTH | YEAR | WEEK}
DO
{SQL COMMAND};
```

Dibawah ini adalah contoh untuk melakukan insert data ke tabel `tbl_sch` setiap satu menit.

```
CREATE EVENT e_sch_2
ON SCHEDULE EVERY 1 MINUTE
DO insert into tbl_sch values (null, 'sch 2', now());
```

Berikut ini adalah hasil dari eksekusi Event Schedule tersebut.

```
+---+-----+-----+
| no | des  | time                |
+---+-----+-----+
| 2  | sch 2 | 2009-06-01 18:16:27 |
| 3  | sch 2 | 2009-06-01 18:17:27 |
| 4  | sch 2 | 2009-06-01 18:18:27 |
| 5  | sch 2 | 2009-06-01 18:19:27 |
| 6  | sch 2 | 2009-06-01 18:20:27 |
| 7  | sch 2 | 2009-06-01 18:21:27 |
| 8  | sch 2 | 2009-06-01 18:22:27 |
| 9  | sch 2 | 2009-06-01 18:23:27 |
| 10 | sch 2 | 2009-06-01 18:24:27 |
| 11 | sch 2 | 2009-06-01 18:25:27 |
+---+-----+-----+
10 rows in set (0.01 sec)
```

Event Schedule yang telah dibuat dapat diedit, berikut ini adalah pola untuk mengubah Event Schedule yang ditunjukkan pada script di bawah ini.

```
ALTER EVENT event_name
[ ON SCHEDULE schedule ]
[ RENAME TO event_name2 ]
[ DO sql_statement ]
```

Dibawah ini adalah contoh untuk mengubah nama event schedule dari `e_sch_2` ke `e_sch_3`.

```
ALTER EVENT e_sch_2
ON SCHEDULE EVERY 2 MINUTE
RENAME TO e_sch_3
DO insert into tbl_sch values (null, 'sch 3', now());
```

Event Schedule yang telah dibuat dapat dihapus, berikut ini adalah pola untuk menghapus Event Schedule yang ditunjukkan pada script di bawah ini.

```
DROP EVENT event_name ;
```

Dibawah ini adalah contoh untuk menghapus schedule e_sch_3.

```
DROP EVENT e_sch_3;
```

Sedangkan untuk melihat Event Schedule yang terdapat pada database dapat menggunakan cara seperti dibawah ini.

```
SELECT * FROM mysql.event\G
```



```
db: test
name: e_sch_1
body: insert into tbl_sch values (null, 'sch 1', now())
definer: root@localhost
execute_at: 2009-07-01 11:06:49
interval_value: NULL
interval_field: NULL
created: 2009-06-04 18:03:54
modified: 2009-06-04 18:03:54
last_executed: NULL
starts: NULL
ends: NULL
status: ENABLED
on_completion: DROP
sql_mode:
comment:
originator: 1
time_zone: SYSTEM
character_set_client: latin1
collation_connection: latin1_swedish_ci
db_collation: latin1_swedish_ci
body_utf8: insert into tbl_sch values (null, 'sch 1', now())
1 row in set (0.00 sec)
```

2.11 Cursor Store Procedure

Pada Store Procedure hasil sebuah query select atau biasa disebut dengan result set dapat disimpan didalam sebuah variable, kemudian dikeluarkan untuk digunakan sebagai filter pada query yang lain atau kebutuhan yang lain.

Tampaknya akan lebih dimengerti apabila dengan sebuah contoh, dibawah ini terdapat store procedure tentang siswa yang memiliki tugas untuk mengubah data siswa yang berstatus telah dihapus pada table siswa akan diubah menjadi berstatus aktif atau tidak dihapus dengan menggunakan cursor.

```

PROCEDURE sp_update_status_siswa()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE v_nisn varchar(255) default '';
  DECLARE cur1 CURSOR FOR select nisn from tbl_siswa where is_delete=1;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

  OPEN cur1;
  REPEAT
    FETCH cur1 INTO v_nisn;
    IF NOT done THEN
      update tbl_siswa set is_delete=0
      where nisn = v_nisn;
    END IF;
  UNTIL done END REPEAT;
  CLOSE cur1;
END

```

Penjelasan dari script diatas adalah sebagai berikut.

1. **DECLARE done INT DEFAULT 0;** adalah mendefinisikan variabel untuk menyimpan status dari cursor.
2. **DECLARE v_nisn varchar(255) default '';** adalah membuat variabel untuk nisn dari table siswa.
3. **DECLARE cur1 CURSOR FOR select nisn from tbl_siswa where is_delete=1;** adalah membuat CURSOR dengan nama CUR1 untuk menyimpan hasil query .
4. **DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;** adalah untuk handle bila hasil looping CUR1 habis maka variabel done di set 1.
5. **OPEN cur1;** adalah membuka CURSOR cur1.
6. **FETCH cur1 INTO v_nisn;** adalah memasukan hasil field pertama (nisn) dari cursor ke varibel v_nisn .
7. **IF NOT done THEN** adalah bila varibel done masih bernilai 0 maka eksekusi query di bawahnya.
8. **UNTIL done END REPEAT;** adalah pengulangan berakhir bila variabel done sama dengan satu.
9. **CLOSE cur1;** adalah menutup CURSOR cur1.

2.12 Lock Tables

Lock Tables apabila diartikan secara harfiah ke bahasa Indonesia adalah penguncian tabel. Pada suatu kasus sebuah database bisa digunakan oleh lebih dari satu aplikasi (misalkan aplikasi yang pertama berbentuk web-base, dan aplikasi yang lain berbentuk desktop aplikasi). kedua aplikasi tersebut mengakses tabel yang sama secara bersamaan, maka mungkin terjadi kesalahan data.

Misalkan kesalahan yang terdapat pada stok barang, pada kasus ini aplikasi pertama sedang melakukan proses penambahan stok pada tabel_stok tiba-tiba aplikasi kedua melakukan penghitungan stok pada tabel_stok ketika proses penambahan stok belum selesai.

Untuk mencegah kejadian seperti ini dapat menggunakan sebuah fitur pada MySql yang bernama Lock Tables. Fitur Lock Tables akan melakukan penguncian tabel ketika sebuah proses pada table sedang berlangsung, kemudian akan membebaskan table ketika proses tersebut telah berakhir.

Rule dari Locks Tables tersedia dua macam yaitu Read Lock dan Write Lock berikut ini adalah penjelasan dari keduanya. Dalam eksperimen ini saya menggunakan database Mysql versi 5.1.37 dengan store engine tabel MyIsam dan Sistem Operasi Ubuntu Karmic Koala .

1. **Read Lock**, adalah penguncian tabel yang menjadikan tabel tersebut hanya dapat melakukan manipulasi proses read yang diterapkan pada semua session termasuk sessionnya sendiri. Jadi apabila ada

manipulasi tabel seperti insert, update, delete, truncate statusnya akan selalu waiting hingga dilakukan Unlock Table.

Berikut ini adalah contoh skrip penggunaan rule Read Lock yang diimplementasikan pada tabel bernama tabel_telepon.

```
mysql> LOCKTABLES tabel_telepon READ;  
Query OK, 0 rows affected (0.00 sec)
```

Bila manipulasi insert/update/delete/truncate pada session sendiri, maka akan menampilkan pesan error seperti contoh dibawah ini.

```
mysql> insert into tabel_telepon values (null, 'DENDIE','081394345678');  
ERROR 1099 (HY000): Table 'tabel_telepon' was locked with a READ lock and can't be updated
```

Tetapi apabila manipulasi insert/update/delete/truncate dilakukan pada session yang lain, maka akan menampilkan proses selalu waiting seperti yang ditunjukkan pada contoh dibawah ini.

```
mysql> insert into tabel_telepon values (null, 'ULIE','08133453455');
```

Proses insert diatas akan dieksekusi apabila telah dilakukan Unlock Table pada session yang bersangkutan, dibawah adalah ini skrip untuk melakukan Unlock Tables.

```
mysql> UNLOCKTABLES;  
Query OK, 0 rows affected (0.00 sec)
```

Proses insert diatas akan dieksekusi apabila telah dilakukan Unlock Table pada session yang bersangkutan, dibawah adalah ini skrip untuk melakukan Unlock Tables.

2. **Write Lock**, adalah penguncian tabel yang menjadikan tabel tersebut dapat dilakukan manipulasi proses read, insert, update, delete, truncate pada sessionnya sendiri dan yang terjadi pada session lain hanya dapat melakukan proses read tabel, apabila terdapat proses insert, update, delete, truncate statusnya waiting hingga dilakukan Unlock Tables.

Berikut ini adalah contoh skrip penggunaan rule Write Lock yang diimplementasikan pada tabel bernama tabel_telepon.

```
mysql> LOCK TABLES tabel_telepon WRITE;  
Query OK, 0 rows affected (0.00 sec)
```

Apabila melakukan manipulasi select/ insert/ update/ delete/ truncate pada session sendiri maka akan mendapatkan hasil sukses, tetapi bila proses select/insert/update/delete/truncate di lakukan pada session yang lain maka akan menampilkan proses waiting seperti yang ditunjukkan dibawah ini.

```
mysql> insert into tabel_telepon values (null, 'DAZ','081334534345');
```

Proses insert diatas akan dieksekusi bila telah dilakukan Unlock Tables pada session yang bersangkutan, dibawah adalah ini skrip untuk melakukan Unlock Tables.

```
mysql> UNLOCK TABLES;  
Query OK, 0 rows affected (0.00 sec)
```

Berikut ini adalah struktur table yang dari tabel_telepon yang digunakan pada eksperimen diatas.

```
CREATE TABLE `tabel_telepon` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nama` varchar(255) DEFAULT NULL,  
  `telepon` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=utf8
```

2.13 Auto Backup Database Menggunakan Crontab

Linux Shell Scripting adalah serangkaian command yang ditulis dalam file teks biasa, kemudian file teks tersebut dipanggil untuk eksekusi command didalamnya. Command pada linux shell bermacam-macam seperti create/update/delete file dan folder, melakukan mounting, hingga melakukan backup database.

Sedangkan Crontab adalah aplikasi penjadwalan pada linux yang memungkinkan untuk eksekusi sebuah program/aplikasi/script secara otomatis dengan waktu yang telah dijadwalkan. Dengan dua buah kemampuan linux tersebut adalah kombinasi lengkap untuk membuat backup database secara otomatis dan terjadwal.

Berikut ini adalah tahapan-tahapan dalam membuat auto backup database dengan memanfaatkan crontab. Sistem operasi yang digunakan oleh penulis dalam eksperimen adalah Ubuntu Karmic Koala dan Bash sebagai program Linux shell.

1. Buat semua file, sebagai contoh bernama **auto_backup.sh** kemudian pada file tersebut tuliskan linux shell scripting, seperti dibawah ini.

```
v_tgl=`date +%Y%m%d_%H%M%S`;  
v_name=bukutamu_${v_tgl}.sql;  
mysqldump -uroot -ppass bukutamu > /home/dendie/Desktop/${v_name};
```

Script diatas adalah shell scripting untuk melakukan dump/backup dari database bernama buku tamu ke alamat **/home/dendie/Desktop/** sedangkan bagian script diatas yang melakukan backup database adalah.

```
mysqldump -uroot -ppass bukutamu > /home/dendie/Desktop/${v_name};
```

Variabel \$vname diatas berisikan format penamaan file yang berisikan tanggal dan jam ketikan proses backup dilakukan. Contoh hasil backup database tersebut adalah **bukutamu_20100128_203001.sql**

Bagian Script yang membuat format penamaan file adalah sebagai berikut.

```
v_tgl=`date +%Y%m%d_%H%M%S`;  
v_name=bukutamu_${v_tgl}.sql;
```

2. Pada command line linux berikan perintah untuk set file **auto_backup.sh** sebagai execute file dengan cara seperti dibawah ini.

```
sudo chmod +x auto_backup.sh
```

3. Pada command line buka program crontab/scheduling untuk dengan cara seperti dibawah ini.

```
sudo crontab -e
```

Kemudian masukan file **auto_backup.sh** kedalam crontab/scheduling seperti yang ditunjukkan dibawah.

```
*/5 **** */home/dendie/Desktop/my_data/auto_backup.sh > /home/dendie/Desktop/auto_backup.log
```

Perintah diatas membuat crontab/scheduling untuk melakukan eksekusi file auto_backup.sh setiap 5 menit sekali dan membuat log/catatan ke file auto_backup.log apabila terjadi trouble saat eksekusi. Berikut ini adalah penjelasan setiap bagian setting crontab diatas.

1. Bagian untuk memberikan setting schedule eksekusi setiap lima menit sekali adalah sebagai berikut.

```
*/5 ****
```

2. Bagian untuk mengarahkan alamat yang akan dieksekusi adalah sebagai berikut.

```
/home/dendie/Desktop/my_data/auto_back
```

3. Bagian Membuat file log auto_backup.log yang otomatis dibuat saat crontab dieksekusi untuk mencatat bila terjadi trouble adalah sebagai

berikut.

```
/home/dendie/Desktop/my_data/auto_backup.log
```

Berikut ini adalah penjelasan dari format penjadwalan dari `crontab`, apabila barisan simbol bintang `crontab` dijadikan tuliskan akan tampak seperti dibawah ini.

```
[menit] [jam] [tanggal dlm satu bulan bulan] [bulan] [hari dalam satu minggu]
```

- Bintang pertama mengartikan menit dari range 0-59. Apabila menggunakan simbol * mengartikan bahwa sebuah perintah yang akan dieksekusi setiap menit.
- Bintang kedua mengartikan jam dari 0-23. Apabila menggunakan simbol * mengartikan bahwa sebuah perintah yang akan dieksekusi setiap jam.
- Bintang ketiga mengartikan tanggal dalam satu bulan dari range 1-31, Apabila menggunakan simbol * mengartikan bahwa perintah akan dieksekusi setiap hari.
- Bintang keempat mengartikan bulan dari range 1-12, Apabila menggunakan simbol * mengartikan bahwa sebuah perintah yang akan di eksekusi setiap bulan.
- Bintang kelima mengartikan hari dalam satu minggu dari range 0-6. Angka 0 mewakili hari minggu, angka 1 hari mewakili senin

dan seterusnya. Apabila menggunakan simbol * mengartikan bahwa sebuah perintah yang akan di eksekusi setiap hari.

Berikut ini adalah beberapa contoh dari penerapan setting crontab.

- Contoh Pertama, Akan menjalankan perintah /usr/bin/foo setiap 15 menit sekali sepanjang waktu selama komputer hidup.

```
0,15,30,45 * * * * * /usr/bin/foo
```

- Contoh Kedua, Akan menjalankan perintah /usr/bin/foo jam 3.10 subuh setiap hari.

```
10 3 * * * /usr/bin/foo
```

- Contoh Ketiga, Akan menjalankan perintah /usr/bin/foo jam 00.10 pada tanggal pertama di setiap bulan

```
10 * 1 * * /usr/bin/foo
```

- Contoh Keempat Akan menjalankan perintah /usr/bin/foo jam 14.10 pada hari pertama di setiap bulan

```
10 14 * * 1 /usr/bin/foo
```

2.14 Referential Integrity

MySQL Memiliki sebuah fitur dalam Relational Database yaitu Referential Integrity. Referential Integrity adalah sebuah cara untuk menjaga konsistensi data antara tabel yang saling berrelasi.

Untuk menggunakan fitur Referential Integrity tipe dari Store Engine tabel yang digunakan adalah yang support Transaksional, salah satu contoh di MySQL yang support Transaksional adalah Store Engine Innodb.

Referential Integrity terdiri dari beberapa macam antara lain adalah.

1. **Restrict** bila dalam bahasa Indonesia adalah membatasi, maksudnya adalah data pada table induk tidak bisa di delete atau di update bila data tersebut memiliki relasi pada tabel lainnya.
2. **Cascade** bila dalam bahasa Indonesia adalah bertingkat, maksudnya adalah apabila data pada table_ induk di delete atau di update maka secara otomatis data pada tabel lain yang memiliki relasi akan di delete/di update secara otomatis.

Pada tulisan ini saya akan membahas Referential Integrity dengan sebuah Kasus yang tentang data prestasi karyawan.

Berikut ini adalah design tabel pada kasus ini yang terdiri dari dua table yaitu tbl_karyawan untuk menyimpan data karyawan dan tbl_prestasi untuk meyimpan data-data prestasi karyawan.

a. Studi Kasus Membuat Referential Integrity tipe Cascade

Buatlah terlebih dahulu sebuah table dengan nama tbl_karyawan untuk menyimpan data karyawan.

```
CREATE TABLE `tbl_karyawan` (  
  `kode` char(6) NOT NULL,  
  `nama` varchar(255) NOT NULL,  
  `tgl_lahir` date DEFAULT NULL,  
  `alamat` tinytext,  
  PRIMARY KEY (`kode`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Perhatikan tulisan ENGINE=InnoDB, itu adalah cara membuat tabel betipe Store Engine InnoDB. Berikutnya adalah memasukan sample data karyawan sebanyak dua buah.

```
insert into tbl_karyawan values  
( '000001', 'Dendie Sanjaya', '1985-01-26', 'Buah Batu'),  
( '000002', 'Yulianti', '1990-10-30', 'Laswi');
```

Selanjutnya membuat tabel tbl_prestasi untuk menyimpan data prestasi karyawan-karyawan tersebut.

```
CREATE TABLE `tbl_prestasi` (  
  `kode` char(6) NOT NULL,  
  `kode_karyawan` char(6) NOT NULL,  
  `prestasi` varchar(255) NOT NULL,  
  `keterangan` text,  
  PRIMARY KEY (`kode`),  
  FOREIGN KEY (`kode_karyawan`) REFERENCES `tbl_karyawan` (`kode`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Perhatikan script diatas terdapat hal yang penting dalam Referential Integrity, yaitu sebagai berikut.

- FOREIGN KEY (`kode_karyawan`) REFERENCES `tbl_karyawan` (`kode`) adalah untuk membuat field kode_karyawan pada tbl_prestasi berelasi dengan field kode pada table karyawan
- ON DELETE CASCADE ON UPDATE CASCADE adalah untuk membuat relasi tersebut bertipe Cascade untuk event delete dan update data.

Berikutnya adalah memasukkan sample data prestasi karyawan sebanyak tiga buah seperti yang ditunjukkan dibawah ini.

```
insert into tbl_prestasi values
('P-0001','000001','Juara 1 Programming Tingkat Kota','-' ),
('P-0002','000001','Juara 2 Olimpiade Matematika Tingkat Kota','-' ),
('P-0003','000002','Runner Up Debat Tingkat 2','-' );
```

Apabila table_karyawan dan table_prestasi tersebut dilihat akan menampilkan data seperti dibawah ini.

```
select * from tbl_karyawan;
```

```
+-----+-----+-----+-----+
| kode  | nama      | tgl_lahir | alamat  |
+-----+-----+-----+-----+
| 000001 | Dendie Sanjaya | 1985-01-26 | Buah Batu |
| 000002 | Yulianti      | 1990-10-30 | Laswi    |
+-----+-----+-----+-----+
```

```
select * from tbl_prestasi;
```

kode	kode_karyawan	prestasi	keterangan
P-0001	000001	Juara 1 Programming Tingkat Kota	-
P-0002	000001	Juara 2 Olimpiade Matematika Tingkat Kota	-
P-0003	000002	Runner Up Debat Tingkat 2	-

Sekarang kita akan coba buktikan melakukan update data field kode dari tbl_karyawan, yang berkode 000001 menjadi 000003.

```
update tbl_karyawan set kode = '000003' where kode = '000001';
```

Maka yang terjadi data pada field kode di tabel tbl_karyawan akan berubah dari kode 000001 menjadi 000003. Dan secara otomatis field kode_karyawan pada table tbl_prestasi akan berubah juga dari kode 000001 menjadi 000003.

```
select * from tbl_prestasi;
```

kode	kode_karyawan	prestasi	keterangan
P-0001	000003	Juara 1 Programming Tingkat Kota	-
P-0002	000003	Juara 2 Olimpiade Matematika Tingkat Kota	-
P-0003	000002	Runner Up Debat Tingkat 2	-

Sekarang kita akan coba buktikan melakukan delete data field kode dari tbl_karyawan, yang berkode 000003.

```
delete from tbl_karyawan where kode = '000003';
```

Maka yang terjadi data pada pada tabel tbl_karyawan kode 000003 telah di hapus dan secara otomatis field kode_karyawan pada table tbl_prestasi yang berkode 000003 akan ikut terhapus.

```
select * from tbl_prestasi;
```

kode	kode_karyawan	prestasi	keterangan
P-0003	000002	Runner Up Debat Tingkat 2	-

b. Studi Kasus Membuat Referential Integrity tipe Restrict

Buatlah terlebih dahulu sebuah table dengan nama tbl_karyawan untuk menyimpan data karyawan.

```
CREATE TABLE `tbl_karyawan` (
  `kode` char(6) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `tgl_lahir` date DEFAULT NULL,
  `alamat` tinytext,
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Perhatikan tulisan ENGINE=innodb, itu adalah cara membuat tabel betipe Store Engine InnoDB. Berikutnya adalah memasukan sample data karyawan sebanyak dua buah.

```
insert into tbl_karyawan values
('000001','Dendie Sanjaya','1985-01-26','Buah Batu'),
('000002','Yulianti','1990-10-30','Laswi');
```

Selanjutnya membuat tabel tbl_prestasi untuk menyimpan data prestasi karyawan-karyawan tersebut.

```
CREATE TABLE `tbl_prestasi` (
  `kode` char(6) NOT NULL,
  `kode_karyawan` char(6) NOT NULL,
  `prestasi` varchar(255) NOT NULL,
  `keterangan` text,
  PRIMARY KEY (`kode`),
  FOREIGN KEY (`kode_karyawan`) REFERENCES `tbl_karyawan` (`kode`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Perhatikan script diatas terdapat hal yang penting dalam Referential Integrity, yaitu sebagai berikut.

- FOREIGN KEY (`kode_karyawan`) REFERENCES `tbl_karyawan` (`kode`) adalah untuk membuat field kode_karyawan pada tbl_prestasi berelasi dengan field kode pada table karyawan .
- ON DELETE RESTRICT ON UPDATE RESTRICT adalah untuk membuat relasi tersebut bertipe Restrict untuk event delete dan update data.

Berikutnya adalah memasukan sample data prestasi karyawan sebanyak tiga buah seperti yang ditunjukkan dibawah ini.

```
insert into tbl_prestasi values
('P-0001','000001','Juara 1 Programming Tingkat Kota','-'),
('P-0002','000001','Juara 2 Olimpiade Matematika Tingkat Kota','-'),
('P-0003','000002','Runner Up Debat Tingkat 2','-');
```

Sekarang kita akan coba buktikan melakukan update data field kode dari tbl_karyawan, yang berkode 000001 menjadi 00003.


```
update tbl_karyawan set kode = '000003' where kode = '000001';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails
(`latihan`.`tbl_prestasi`, CONSTRAINT `tbl_prestasi_ibfk_1` FOREIGN KEY (`kode_karyawan`) REFERENCES
`tbl_karyawan` (`kode`))
```

Maka yang terjadi adalah memunculkan ERROR 1451 (23000) yang mengartikan bahwa kode 000001 tidak dapat di update karena data 00001 digunakan pada tabel tbl_prestasi

Sekarang kita akan coba buktikan melakukan delete data field kode dari tbl_karyawan. yang kode 000001.

```
delete from tbl_karyawan where kode = '000001';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails
(`latihan`.`tbl_prestasi`, CONSTRAINT `tbl_prestasi_ibfk_1` FOREIGN KEY (`kode_karyawan`) REFERENCES
`tbl_karyawan` (`kode`))
```

Maka yang terjadi adalah menampilkan error ERROR 1451 (23000) yang mengartikan bahwa kode 000001 tidak dapat di delete karena data 00001 digunakan pada tabel tbl_prestasi.

Agar data 000001 pada tbl_karyawan dapat di delete maka data 00001 pada field kode_karyawan di table tbl_prestasi harus dihapus lebih dahulu.

2.15 MySQL Smart Search

MySQL adalah sebuah database yang populer, beragam fitur dimiliki oleh mysql salah satunya adalah fitur pencarian data secara smart/cerdas yaitu Full-Text Searches.

Full-Text Searches adalah sebuah pencarian yang kemampuannya lebih dari kemampuan pencarian menggunakan statemen Like, untuk lebih jelasnya pada tulisan ini akan diberikan contoh sederhana dalam penggunaan fitur full-text. Pada contoh disini kasus yang akan dibuat adalah tentang pencarian data dari sekumpulan artikel .

1. Membuat table untuk menyimpan data-data artikel dan menambahkan feature Full-Text.

```
CREATE TABLE artikel (  
  id int AUTO_INCREMENT NOT NULL PRIMARY KEY,  
  judul VARCHAR(255),  
  isi TEXT,  
  FULLTEXT (judul,isi));
```

Bagian yang penting dari DDL tabel artikel diatas adalah baris FULLTEXT (judul,isi). baris tersebut adalah sebuah definisi yang memberitahu pada mysql bahwa field judul dan isi menggunakan fitur full-text.

2. Memasukan data-data artikel pada table artikel, dibawah ini sample data yang digunakan.

```
INSERT INTO artikel (judul, isi) VALUES  
( 'Apa itu PHP ', 'php adalah server side scripting open source....'),  
( 'Instalasi PHP', 'Tahap pertama instalasi PHP adalah... '),  
( 'Instalasi APC di PHP', 'Tahap pertama instalasi APC adalah...'),  
( 'Optimasi PHP', 'Hal-hal untuk melakukan optimasi PHP adalah sebagai berikut...'),  
( 'PHP, RUBY, JAVA', 'Perbandingan PHP dengan RUBY, JAVA adalah sebagai berikut...'),  
( 'PHP dan MySQL ', 'PHP biasa nya dipasangkan dengan Database MySQL karena...'),  
( 'XAMPP', 'XAMPP adalah sebuah paket instalasi yang didalamnya ada PHP, MYSQL..'),  
( 'Framework PHP', 'Berbagai macam framework php antarlain ZEND, SYMFONY, ...')
```

3. Melakukan pencarian data menggunakan fitur full-text. Fitur full-text memiliki tiga buah mode pencarian, yaitu sebagai berikut.

- IN NATURAL LANGUAGE MODE adalah sebuah mode pencarian yang bersifat natural/umum, secara sekilas mode pencarian ini seperti pencarian menggunakan LIKE.
- WITH QUERY EXPANSION adalah sebuah mode pencarian yang mana kata-kata dari hasil pencariannya dari sebuah row yang paling relevan menjadi kata kunci dalam pencarian juga.
- IN BOOLEAN MODE adalah sebuah mode pencarian dengan mengkombinasikan operator-operator seperti +, -, (), " dsb untuk meningkatkan kemampuan dalam pencarian.

Sepintas mode pencarian IN BOOLEAN MODE seperti keyword yang ada di search engine google.com. Secara pribadi IN BOOLEAN MODE adalah mode pencarian yang menurut saya yang paling menarik.

Mode pencarian full-text yang akan dibahas pada tulisan ini adalah mode IN BOOLEAN MODE, berikut ini adalah contoh-contoh penggunaannya.

a. Melakukan pencarian kata instalasi pada field judul dan isi tetapi yang tidak mengandung kata xampp.

```
SELECT * FROM artikel WHERE MATCH (judul,isi) AGAINST ('INSTALASI -XAMPP' IN BOOLEAN MODE);
```

id	judul	isi
22	Instalasi PHP	Tahap pertama instalasi PHP adalah...
23	Instalasi APC di PHP	Tahap pertama instalasi APC adalah...

Bagian penting dari perintah pencarian data diatas adalah MATCH sebagai perintah untuk menentukan pencarian kedalam field apa saja.

Kemudian AGAINST adalah tempat untuk meletakkan kata atau keyword yang akan cari dan IN BOOLEAN MODE adalah perintah untuk menggunakan mode pencarian IN BOOLEAN MODE . Simbol minus (-) pada tulisan -XAMPP mengartikan pencarian tidak boleh mengandung kata XAMPP.

b. Melakukan pencarian pada row yang ditemukan kata instalasi dan mysql.

```
SELECT * FROM artikel WHERE MATCH (judul,isi) AGAINST ('+instalasi +mysql' IN BOOLEAN MODE);
```

id	judul	isi
27	XAMPP	XAMPP adalah sebuah paket instalasi yang didalamnya ada PHP, MYSQL..

Simbol plus (+) pada kata +instalasi +mysql adalah sebuah operator yang mengartikan pada row harus mengandung kata tersebut.

c. Melakukan pencarian pada row yang ditemukan kata berawalan my.

```
SELECT * FROM artikel WHERE MATCH (judul,isi) AGAINST ('my*' IN BOOLEAN MODE);
```

id	judul	isi
26	PHP dan MySQL	PHP biasa nya dipasangkan dengan Database MySQL karena...
27	XAMPP	XAMMP adalah sebuah paket instalasi yang didalamnya ada PHP, MYSQL..

Simbol bintang (*) pada kata my* adalah sebuah operator yang mengartikan sebuah karakter apa saja pada sebuah kata.

d. Melakukan pencarian pada row mengandung kata persis sama dengan instalasi php.

```
SELECT * FROM artikel WHERE MATCH (judul,isi) AGAINST ("INSTALASI PHP" IN BOOLEAN MODE);
```

id	judul	isi
22	Instalasi PHP	Tahap pertama instalasi PHP adalah...

Simbol kutip ganda (") pada kata "INSTALASI PHP" mengartikan pencarian kata yang sama persis dengan yang kata yang ada didalam kutip .

2.16. MySQL Soundex

MySQL adalah database yang memiliki banyak fitur selain itu MySQL adalah database yang populer. Diantara fitur-fitur yang dimiliki MySQL salah satunya adalah Soundex.

Soundex adalah sebuah function pada MySQL yang menghasilkan kode soundex dari sebuah kata atau string, yang mana kode soundex tersebut bernilai sama untuk kata atau string yang apabila diucapkan akan terdengar mirip. Soundex biasanya digunakan pada pencarian data.

Pada tulisan ini penggunaan soundex akan diimplementasikan pada sebuah kasus pencarian data di table kamus komputer.

1. Membuat tabel kamus untuk menyimpan istilah komputer berikut dengan definisinya.

```
CREATE TABLE `kamus` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `keyword` varchar(255) DEFAULT NULL,  
  `description` text,  
  PRIMARY KEY (`id`)  
)
```

2. Memasukkan tiga buah sample data pada tabel kamus.

```
insert into kamus values
(null, 'Harddisk', 'Hardisk adalah bla bla...'),
(null, 'Keybord', 'Keybord adalah bla bla...'),
(null, 'Mouse', 'Mouse adalah bla bla...');
```

```
select * from kamus;
+----+-----+-----+
| id  | keyword | description      |
+----+-----+-----+
| 1   | Harddisk | Hardisk adalah bla bla... |
| 2   | Keybord  | Keybord adalah bla bla... |
| 3   | Mouse    | Mouse adalah bla bla...   |
+----+-----+-----+
```

3. Melakukan tes pencarian data menggunakan soundex. Pada kasus disini memberikan contoh pencarian kata "Keyboard" tetapi yang diketikan menjadi "Keybborod".

```
select * from kamus where soundex(keyword) = soundex('Keybborod')
+----+-----+-----+
| id  | keyword | description      |
+----+-----+-----+
| 2   | Keybord  | Keybord adalah bla bla... |
+----+-----+-----+
```

Bagian terpenting dari SQL diatas adalah **soundex(keyword) = soundex('Keybborod')** Walaupun kata "Keybborod" tidak sama dengan "Keyboard" tetapi bila diucapkan seperti terdengar sama, maka function soundex akan memberikan sugesti kata "Keyboard".

2.17 Manajemen User

MySQL Database adalah sebuah database yang populer, salah satu fitur yang dimiliki oleh MySQL adalah Manajemen User. Dalam sebuah Database MySQL Server biasanya terdapat banyak database dan pada umumnya satu buah database (dapat lebih dari satu)

digunakan oleh satu buah aplikasi/web.

Contohnya adalah pada jasa share hosting, satu buah Database MySQL server di gunakan bersama-sama oleh banyak penyewa. Karena sebuah database MySQL digunakan secara bersama-sama maka dibutuhkan Manajemen User untuk mengelolah hak akses dari setiap user dalam mengakses database agar tercipta ketertiban setiap user dalam menggunakan database.

Pada kasus lain Manajemen User database dirasakan perlu diterapkan pada prodak-prodak aplikasi/program dalam mengakses database yang dapat berupa CMS, Framework atau apapun yang mana prodak tersebut bersifat Open Source.

Karena Open Source membuat setiap orang bisa mengetahui dimana letak konfigurasi koneksi database. Apabila konfigurasi itu berhasil 'dibongkar' dan koneksi database dari program tersebut menggunakan akses user level Root maka dapat terjadi kemungkinan terburuk yaitu database dapat dirusak oleh pihak pembobol atau hacker.

Untuk memenuhi kebutuhan tersebut dan mencegah kejadian buruk diatas maka menggunakan Manajemen User adalah solusinya. Berikut ini adalah cara mengelola manajemen user di database MySQL yang dimulai dari membuat user/penggunaan database.

1. Masuk ke Console MySQL menggunakan user Root, sebagai contoh password yang saya gunakan adalah pass.


```
mysql -uroot -ppass
```

2. Selanjutnya membuat user baru, sebagai contoh saya membuat sebuah user bernama dendie dengan password pass .

```
CREATE USER 'dendie'@'localhost' IDENTIFIED BY 'pass';
```

3. Kemudian user dendie diberikan akses hanya dapat menggunakan database bernama sms

```
GRANT ALL PRIVILEGES ON sms.* TO 'dendie'@'%' WITH GRANT OPTION;
```

simbol **sms.*** mengartikan database yang dapat diakses adalah database sms beserta seluruh tabelnya. Bila Anda menuliskannya **sms.inbox** artinya yang dapat diakses hanya tabel inbox pada tabel sms

simbol **'dendie'@'%'** mengartikan bahwa database dapat diakses menggunakan user dendie dari IPaddress apa saja. Bila Anda menuliskan **'dendie'@'119.168.10.28'** artinya user dendie hanya dapat digunakan dari IP Address 119.168.10.28

Apabila user dendie mencoba mengakses database selain sms maka akan menampilkan pesan error seperti dibawah ini.

```
ERROR 1044 (42000): Access denied for user 'dendie'@'localhost' to database 'portofolio'
```

Sedangkan untuk menghapus user dapat menggunakan perintah seperti ini , pada contoh dibawah adalah menghapus user dendie yang

menggunakan ip address bebas yang disimbolkan dengan persen (%) .

```
DROP USER 'dendie'@'%';
```

Pada contoh dibawah adalah menghapus user yuli yang menggunakan ip address berasal dari localhost

```
DROP USER 'yuli'@'localhost';
```

Untuk melihat daftar user yang ada pada database dapat menggunakan cara seperti ini, terlebih dulu Anda harus login sebagai root .

```
use mysql  
select host,user from user;
```

Kemudian dari SQL tersebut akan muncul daftar user yang dimiliki database MySQL server, seperti yang ditunjukkan dibawah ini.

host	user
%	root
127.0.0.1	root
192.168.10.%	root
dendie-laptop	root
localhost	debian-sys-maint
localhost	root
localhost	websekolah
localhost	yuli

2.18. MySQL Remove Definer

Pada database MySQL Bila melakukan dump/backup database menyertakan view, store procedure, function dan trigger menggunakan perintah mysqldump secara otomatis selalu menyertakan string definer dari hasil backup nya.

Definer adalah sebuah penanda yang menjelaskan User dan Ip yang dapat me-create view, store procedure, function dan trigger ketika akan di lakukan restore pada sebuah database.

Apabila User dan Ip yang tertera pada definer tidak sama dengan User dan Ip anda dalam mengakses database MySQL maka yang terjadi saat restore view, store procedure, function dan trigger tersebut akan mendapatkan error alias gagal. Agar restore view, store procedure, function dan trigger berhasil makan string definer harus dihilangkan terlebih dahulu.

Berikut ini adalah cara menghilangkan string definer menggunakan bahasan pemrograman perl sedangkan Operation System yang digunakan pada tulisan ini adalah Linux Ubuntu Lucid Lynx yang secara default telah terinstal perl, di Linux Ubuntu untuk install PERL dapat menggunakan perintah sebagai berikut.

```
sudo apt-get install perl
```

Berikut ini adalah sebuah contoh dump database yang menyertakan view, store procedure, function dan trigger.

```
mysqldump -uroot -p123 --routines databasesaya > databasesaya.sql
```

Setelah file backup bernama databasesaya.sql terbuat selanjutnya adalah menghilangkan string definer. Pada console linux tuliskan perintah seperti dibawah ini.

```
perl -p -i.bak -e "s/DEFINER=\`root\`@\`localhost\`//g" databasesaya.sql
```

Pada contoh di atas adalah menghapus string `DEFINER=\`root\`@\`localhost\`` menjadi spasi kosong dari file yang bernama `databasesaya.sql` . Berikut ini penjelasan dari setiap option perl diatas .

- **perl** adalah perintah untuk menjalankan bahasa pemrograman perl
- **-p** akan membaca isi file sampai baris terakhir
- **-i.bak** adalah untuk membuat backup file dari original file `databasesaya.sql` menjadi `databasesaya.sql.bak`
- **-e** adalah untuk mengganti tulisan `DEFINER=\`root\`@\`localhost\`` menjadi spasi kosong

2.19 MySQL - Show Procedure, Function, Triggers

Pada umumnya untuk mempermudah dalam melakukan Administrasi Database Server MySQL, user menggunakan tool MySQL GUI (Graphical User Interface) Client.

MySQL GUI (Graphical User Interface) Client Seperti Navicate, PHPmyAdmin, Nerocode SQLWave, EngInSite Client, dapat meningkatkan efesiensi waktu bila dibandingkan menggunakan command line.

Yang dapat dilakukan melalui MySQL GUI (Graphical User Interface) Client di antaranya adalah dapat dengan mudah melihat sebuah Function atau Store Procedure yang terdapat pada sebuah database.

Tetapi bagaimana bila Tool MySQLGUI Client tidak terdapat pada komputer, dan kita membutuhkan informasi mengenai Function, Store Pecedure dan Trigger yang terdapat pada sebuah database.

MySQL Server meyimpan berbagai informasi sistemnya di database termasuk Function, Store Pecedure dan Trigger disebuah database bernama INFORMATION_SCHEMA. Untuk dapat mengakses database tersebut kita harus login sebagai root terlebih dahulu.

Berikut ini adalah cara untuk menampilkan informasi Function, Store Pecedure dan Trigger yang terdapat pada sebuah database . Login Ke MySQL melalui MySQL Client Command Line Base kemudian

ketikan seperti dibawah ini.

```
mysql -uusername -ppasswordroot
```

- Dibawah ini adalah contoh dari login ke MySQL.

```
mysql -uroot -ppass
```

1. Menampilkan Store Procedure

```
SELECT ROUTINE_NAME FROM INFORMATION_SCHEMA.ROUTINES WHERE ROUTINE_TYPE="PROCEDURE" AND  
ROUTINE_SCHEMA="[nama database]";
```

- Contoh dari menampilkan Store Procedure

```
SELECT ROUTINE_NAME FROM INFORMATION_SCHEMA.ROUTINES WHERE ROUTINE_TYPE="PROCEDURE" AND  
ROUTINE_SCHEMA="dbpegawai";
```

- Hasil dari menampilkan Store Procedure adalah sebagai berikut.

```
+-----+  
| ROUTINE_NAME |  
+-----+  
| sp_combowilayah |  
| sp_combowilayah_semua |  
| sp_const |  
| sp_const_hapus |  
| sp_const_simpan |  
| sp_const_tampil |  
| sp_feed_combosekolah |  
| sp_feed_tampil |  
| sp_jurusansmk_tampil |  
| sp_kelas_tingkat_combo |
```

2. Menampilkan Function

```
SELECT ROUTINE_NAME FROM INFORMATION_SCHEMA.ROUTINES WHERE ROUTINE_TYPE="FUNCTION" AND  
ROUTINE_SCHEMA="[nama database]";
```

- Contoh dari menampilkan Function

```
SELECT ROUTINE_NAME FROM INFORMATION_SCHEMA.ROUTINES WHERE ROUTINE_TYPE="FUNCTION" AND  
ROUTINE_SCHEMA="dbpegawai";
```

3. Menampilkan Triggers

```
SELECT TRIGGER_NAME FROM INFORMATION_SCHEMA.TRIGGERS WHERE TRIGGER_SCHEMA="[nama  
database]";
```

- Contoh dari menampilkan Triggers

```
SELECT TRIGGER_NAME FROM INFORMATION_SCHEMA.TRIGGERS WHERE TRIGGER_SCHEMA="dbpegawai";
```

2.20 Load Data Infile

Pada sebuah kasus terdapat sejumlah data pada file excel yang perlu di inputkan pada table MySQL. Apabila jumlah data yang perlu dimasukan tersebut hanya beberapa buah dapat dilakukan dengan cara manual yaitu memasukan satu persatu data tersebut ke table database melalui aplikasi atau SQL mungkin terasa ringan, tetapi apabila jumlah data tersebut berjumlah puluhan ribu makan akan sangat melelahkan dan memerlukan waktu yang lama sekali untuk menyelesaikannya.

Pada Database MySQL memiliki fitur Load Data Infile. Fitur tersebut adalah untuk memasukan data ke table database dengan sumber data berasal dari file. Database MySQL yang saya gunakan versi 5.1 dengan Operation System Ubuntu 11.04 - the Natty Narwhal

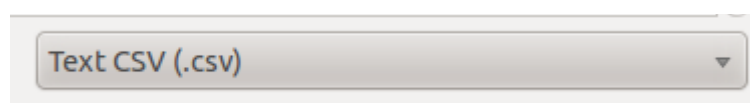
Berikut ini adalah langkah-langkah untuk melakukan input dari data dari file excel ke database MySQL menggunakan fitur Load Data Infile.

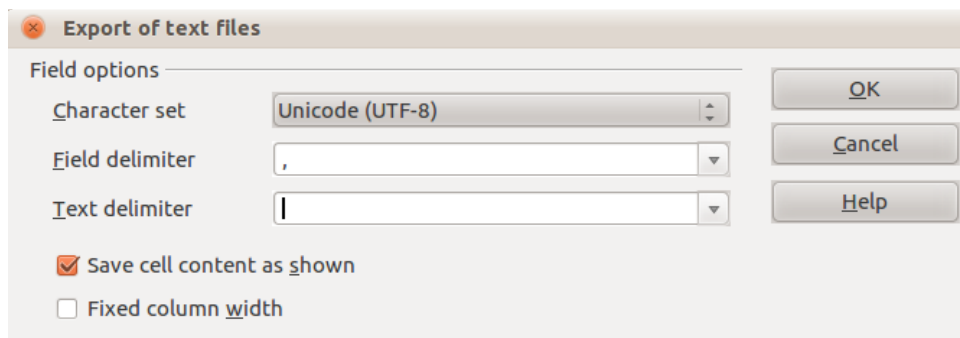
a. Rubah data pada Excel menjadi bertipe CSV .

1. Hilangkan Judul Kolom pada File Excel, menjadi seperti contoh dibawah ini.

	A	B	C
1	1	Afghanistan	AF
2	2	Albania	AL
3	3	Algeria	DZ
4	4	American Samoa	AS
5	5	Andorra	AD
6	6	Angola	AO
7	7	Anguilla	AI
8	8	Antarctica	AQ
9	9	Antigua and Barbuda	AG
10	10	Argentina	AR
11	11	Armenia	AM
12	12	Aruba	AW

2. Lakukan konversi ke csv dan berikan delimiter koma (koma sebagai contoh) untuk membedakan antar field seperti contoh dibawah ini.





3. Apabila konversi Excel ke CSV berhasil maka akan menghasilkan file berekstensi *.csv yang bila file tersebut dibuka akan tampak seperti dibawah ini.

```
1,Afghanistan,AF
2,Albania,AL
3,Algeria,DZ
4,American Samoa,AS
5,Andorra,AD
6,Angola,AO
7,Anguilla,AI
8,Antarctica,AQ
9,Antigua and Barbuda,AG
10,Argentina,AR
11,Armenia,AM
12,Aruba,AW
```

b. Lakukan perintah load file csv seperti ini contoh dibawah ini, sebelumnya Anda masuk terlebih dahulu ke prompt command line mysql melalui console .

- Pola Load Data Infile

```
mysql -u[username] -p[password mysql] -e "LOAD DATA LOCAL INFILE '[alamat file csv]' INTO TABLE [nama table tujuan] FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'" [nama database tujuan]
```

- Contoh Load Data Infile

```
mysql -uroot -ppass -e "LOAD DATA LOCAL INFILE '/home/dendie/Desktop/negara.csv' INTO TABLE tbl_negara  
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'" db_teritorial
```

Berikut ini adalah penjelasan dari script diatas.

- **mysql** mengartikan menggunakan service database mysql
- **-u** adalah untuk memasukan username database mysql
- **-p** adalah untuk memasukan password database mysql
- **-e** adalah argumen untuk memberikan perintah eksekusi kepada mysql
- **LOAD DATA LOCAL INFILE** adalah perintah untuk melakukan load/memuat data dari sebuah file .
- **INTO TABLE** adalah tujuan import table untuk data yang di load
- **FIELDS TERMINATED BY ','** mengartikan bahwa karakter koma sebagai pemisah antar kolom dari data yang di load .
- **TERMINATED BY '\n'** mengartikan bawah karakter \n atau baris baru sebagai pemisah setiap baris dari data yang di load .
- **db_teritorial** adalah nama tujuan database untuk data yang di load .

Berikut ini adalah contoh hasil import data menggunakan Load Data Infile yang ditampilkan menggunakan software MySQL Query Browser Versi 1.2.12 by MySQL AB.

id	nama	kode
1	Afghanistan	AF
2	Albania	AL
3	Algeria	DZ
4	American Samoa	AS
5	Andorra	AD
6	Angola	AO
7	Anguilla	AI
8	Antarctica	AQ
9	Antigua and Barbuda	AG
10	Argentina	AR
11	Armenia	AM
12	Aruba	AW

2.21 MySQL Copy Table

Untuk keperluan backup atau keperluan lain terkadang perlu menyalin atau melakukan duplikasi data. Yang di duplikasi dapat berupa stuktur table atau beserta seluruh data di salin ke table yang baru.

Sebagai contoh ingin melakukan percobaan manipulasi data pada sebuah table yang menggunakan operasi update, insert dan delete. Maka untuk menjaga kesalahan dari percobaan, table tersebut perlu di backup terlebih dahulu.

Dalam eksperimen ini penulis menggunakan database MySQL versi 5.1 dengan Operation System Ubuntu 11.04 - the Natty Narwha.

1. Berikut ini adalah contoh untuk duplikasi sebuah table menjadi table yang baru.

- Pola duplikasi struktur table

```
CREATE TABLE [nama table copy] LIKE [nama table sumber]
```

- Contoh duplikasi struktur table

```
CREATE TABLE siswa_copy LIKE siswa
```

2. Berikut ini adalah contoh untuk duplikasi table beserta data menjadi table yang baru.

- Pola duplikasi struktur table beserta data

```
CREATE TABLE [nama table copy] SELECT * FROM [nama table sumber]
```

- Contoh duplikasi struktur table beserta data

```
CREATE TABLE siswa_copy SELECT * FROM siswa
```

2.22 MySQL Export to CSV

Untuk beberapa keperluan data yang berada di database perlu diolah software spread sheet. Data yang diolah spread sheet (Microsoft Excel & OpenOffice SpreadSheet) biasanya untuk keperluan membuat Grafik dan Statistik yang akan disajikan dalam presentasi atau untuk analisa bisnis.

Sebagai contoh ingin melakukan export data Artikel dari sebuah table mysql agar dapat dibuka di Microsoft Excel. Berikut ini adalah cara membuatnya. Database MySQL yang penulis gunakan adalah versi 5.1 dengan Operation System Ubuntu 11.04 - the Natty Narwhal

- Apabila data Artikel dibuka pada MySQL akan tampak seperti dibawah ini

```
mysql> select * from artikel;
+-----+-----+-----+
| id | judul                | isi                |
+-----+-----+-----+
| 21 | Apa itu PHP          | php adalah server side scripting open source.... |
| 22 | Instalasi PHP        | Tahap pertama instalasi PHP adalah... |
| 23 | Instalasi APC di PHP | Tahap pertama instalasi APC adalah... |
| 24 | Optimasi PHP         | Hal-hal untuk melakukan optimasi PHP adalah sebagai berikut... |
| 25 | PHP, RUBY, JAVA     | Perbandingan PHP dengan RUBY, JAVA adalah sebagai berikut... |
| 26 | PHP dan MySQL        | PHP biasa nya dipasangkan dengan Database MySQL karena... |
| 27 | XAMPP                | XAMPP adalah sebuah paket instalasi yang didalamnya ada PHP, MYSQL.. |
| 28 | Framework PHP        | Berbagai macam framework php antarlain ZEND, PRDDO, SYMFONY, ... |
+-----+-----+-----+
```

Untuk melakukan export data tersebut ke file Microsoft Excel dapat menggunakan perintah mysql outfile seperti yang di tunjukan dibawah ini .

- Pola export data

```
[select query] OUTFILE '[alamat tujuan outfile]' FIELDS TERMINATED BY '[simbol pemisah kolom]' FROM [nama tabel];
```

- Contoh export data

```
SELECT * INTO OUTFILE '/tmp/artikel.csv' FIELDS TERMINATED BY ';' FROM artikel;
```

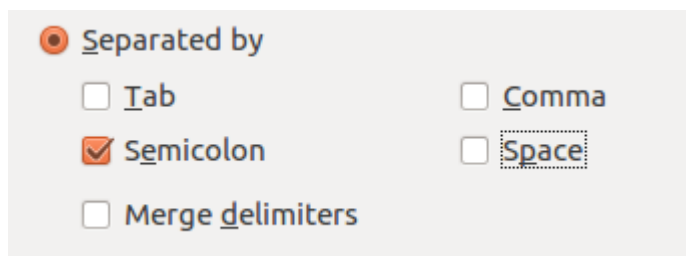
Bila di saat melakukan outfile mengalami error seperti dibawah ini, hal tersebut mengartikan bahwa tujuan folder belum diberi akses untuk creat file.

```
ERROR 1 (HY000): Can't create/write to file '/tmp/artikel.csv' (Errcode: 13)
```

Untuk mengatasi error tersebut berikan akses untuk create file pada folder tujuan dengan permission 777 seperti contoh dibawah ini.

```
sudo chmod 777 tmp/
```

Bila proses export berhasil maka pada direktori tujuan akan terdapat file berekstensi *.csv. Kemudian buka file tersebut menggunakan Microsoft Excel atau OpenOffice SpreadSheet dan pilih delimiter kolom nya adalah semicolon (;) seperti yang di contohkan dibawah ini.



Apabila file *.csv hasil export data tersebut dibuka maka hasilnya akan tampak seperti dibawah ini.

A	B	C
21	Apa itu PHP	php adalah server side scripting open source....
22	Instalasi PHP	Tahap pertama instalasi PHP adalah...
23	Instalasi APC di PHP	Tahap pertama instalasi APC adalah...
24	Optimasi PHP	Hal-hal untuk melakukan optimasi PHP adalah sebagai berikut...
25	PHP, RUBY, JAVA	Perbandingan PHP dengan RUBY, JAVA adalah sebagai berikut...
26	PHP dan MySQL	PHP biasanya dipasangkan dengan Database MySQL karena...
27	XAMPP	XAMMP adalah sebuah paket instalasi yang didalamnya ada PHP, MYSQL...
28	Framework PHP	Berbagai macam framework php antara lain ZEND, PRDDO, SYMFONY, ...

2.23 MySQL Concat

Dalam menampilkan data agar lebih mudah dan cepat dibaca diperlukan query yang berorientasi kepada pengguna data.

Penggabungan string adalah sebuah cara yang dapat di manfaatkan untuk menampilkan data agar lebih mudah dibaca. Untuk melakukan penggabungan string di MySQL dapat menggunakan fungsi concat atau contact_ws

Database MySQL yang penulis gunakan adalah versi 5.1 dengan Operation System Ubuntu 11.04 - the Natty Narwhal.

Sebagai contoh saya memiliki data pegawai yang memiliki nama depan dan nama belakang yang tampak pada tabel seperti dibawah ini.

id	nama_depan	nama_belakang
1	Dendie	Sanjaya
2	Yulianti	Sanjaya
3	Dewi	Lestari
4	Khelda	Ginangjar
5	Khelda	Prasetyo
6	Irwan	Prasetyo
7	Irma	Dewi Lestari

Kemudian Saya memerlukan data nama lengkap pegawai yang merupakan penggabungan dari nama depan dan nama belakang dari table tersebut. Berikut ini adalah cara untuk melakukan penggabungan string nama_depan pegawai dan nama_belakang menjadi nama_lengkap.

- Pola penggabungan field

```
SELECT [function concat field] AS nama_lengkap FROM [nama table];
```

- Contoh penggabungan field

```
SELECT CONCAT(nama_depan,' ',nama_belakang) AS nama_lengkap from pegawai;
```

- Contoh lain penggabungan field

```
SELECT CONCAT_WS(' ',nama_depan,nama_belakang) AS nama_lengkap from pegawai;
```

Berikut ini adalah hasil dari penggabungan string antara field nama_depan dan nama_belakang.


```

+-----+
| nama_lengkap |
+-----+
| Dendie Sanjaya |
| Yulianti Sanjaya |
| Dewi Lestari |
| Khelda Ginanjar |
| Khelda Prasetyo |
| Irwan Prasetyo |
| Irma Dewi Lestari |
+-----+

```

Penggabungan field atau concat dapat dimanfaatkan untuk melakukan pencarian data yang lebih baik seperti contoh dibawah ini .

```

SELECT CONCAT(nama_depan,' ',nama_belakang) AS nama_lengkap FROM pegawai WHERE
CONCAT(nama_depan,nama_belakang) LIKE '%dewi%';

```

Pada query diatas melakukan pencarian nama yang mengandung kata dewi di nama_depan atau nama_belakang. Dibawah ini adalah hasil dari pencarian tersebut.

```

+-----+
| nama_lengkap |
+-----+
| Dewi Lestari |
| Irma Dewi Lestari |
+-----+

```

2.24 MySQL Date Time

Date dan Time adalah sebuah "mahluk" yang tidak dapat dipisahkan dalam kehidupan kita. Katanya bila bisa mengatur waktu dengan baik maka akan lebih dekat dengan kesuksesan.

Dalam pemrograman komputer persoalan Date dan Time dibanyak aplikasi selalu terjadi, dari menampilkan format tanggal hingga operasi-operasi yang melibatkan Date & Time.

Dari beragam fitur yang dimiliki Database MySQL diantaranya adalah fitur untuk melakukan manipulasi Time dan Date yaitu DATEDIFF, TIMEDIFF, TIME_TO_SEC. Fungsi-fungsi tersebut dapat mempermudah dan mempercepat dalam mengelolah hal-hal yang berhubungan dengan Time & Date.

Berikut ini adalah beberapa kegunaan dari memanfaatkan fitur Time & Date yang dimiliki Database MySQL. Database MySQL yang dipergunakan penulis adalah versi 5.1 dengan Operation System Ubuntu 11.04 - the Natty Narwhal.

1. Menghitung Keterlambatan Hari. Menghitung keterlambatan hari Biasanya di implemetasikan di persoalan-persoalan yang berhubungan dengan peminjaman, seperti perpustakaan dan rental film.

Misalkan saya memiliki data peminjam buku seperti dibawah ini. Kemudian saya ingin menghitung keterlambatan pengembalian buku.

id	id_pegawai	id_buku	tgl_pinjam	tgl_batas_pinjam
3	3	3	2011-01-22	2011-01-28
2	2	2	2011-01-20	2011-01-25
1	1	1	2011-01-25	2011-01-31

- Berikut ini adalah cara untuk menghitung keterlambatan hari pengembalian buku.

```
SELECT id_pegawai, id_buku, DATEDIFF(NOW(),tgl_batas_pinjam) AS keterlambatan FROM peminjaman;
```

id_pegawai	id_buku	keterlambatan
3	3	4
2	2	7
1	1	1

Penjelasan dari sql diatas adalah sebagai berikut.

- **DATEDIFF** adalah fungsi untuk menghitung selisih tanggal

2. **Menghitung Usia** . Misalkan saya memiliki data tanggal lahir karyawan sebagai berikut, kemudian saya ingin menghitung usia setiap karyawan tersebut

id	nama_depan	nama_belakang	tgl_lahir
1	Dendie	Sanjaya	1965-05-20 00:00:00
2	Yulianti	Sanjaya	1975-06-10 00:00:00
3	Dewi	Lestari	1985-08-15 00:00:00
4	Khelda	Ginangjar	1991-11-07 00:00:00
5	Khelda	Prasetyo	1982-12-17 00:00:00
6	Irwan	Prasetyo	1982-10-17 00:00:00
7	Irma	Dewi Lestari	1983-09-16 00:00:00

- Berikut ini adalah cara untuk menghitung usia karyawan

```
SELECT nama_depan, tgl_lahir, ((YEAR(NOW()) - YEAR(tgl_lahir)) - (RIGHT(NOW(),5) <
RIGHT(DATE(tgl_lahir),5)) AS usia FROM pegawai
```

nama_depan	tgl_lahir	usia
Dendie	1965-05-20 00:00:00	46
Yulianti	1975-06-10 00:00:00	35
Dewi	1985-08-15 00:00:00	25
Khelda	1991-11-07 00:00:00	19
Khelda	1982-12-17 00:00:00	28
Irwan	1982-10-17 00:00:00	28
Irma	1983-09-16 00:00:00	27

Penjelasan dari SQL diatas adalah sebagai berikut.

- **NOW** adalah fungsi untuk menampilkan tanggal sekarang
- **YEAR** adalah fungsi untuk mengambil tahun dari sebuah tanggal
- **RIGTH** adalah fungsi untuk mengambil karakter dari sebelah kanan

3. **Menghitung Jam Kerja.** Menghitung jam kerja dapat di manfaatkan untuk menghitung jam kerja karyawan di hari yang berjalan atau untuk menghitung akumulasi jam kerja yang telah berlalu.

Misalkan saya memiliki table absensi pegawai seperti dibawah ini. Kemudian saya memerlukan data tentang jumlah jam kerja berjalan pegawai dan jumlah total jam kerja pegawai.

id	id_pegawai	tanggal	jam_masuk	jam_keluar
1	1	2010-12-20	08:00:00	17:00:00
2	1	2010-12-21	08:20:00	17:10:00
3	2	2010-12-20	08:10:00	17:13:00
4	2	2010-12-21	08:05:00	17:01:00

- Berikut ini adalah cara untuk menampilkan jumlah jam kerja karyawan dihari yang berjalan.

```
SELECT id_pegawai, (TIMEDIFF(jam_keluar,jam_masuk)) as jam_kerja FROM absen;
```

id_pegawai	jam_kerja
1	09:00:00
1	08:50:00
2	09:03:00
2	08:56:00

- Berikut ini adalah cara untuk menampilkan total jam kerja karyawan.

```
SELECT id_pegawai, SEC_TO_TIME(SUM((TIME_TO_SEC(TIMEDIFF(jam_keluar,jam_masuk)))) as jam_kerja FROM absen GROUP BY id_pegawai;
```

id_pegawai	jam_kerja
1	17:50:00
2	17:59:00

Penjelasan fungsi MySQL diatas adalah sebagai berikut

- **TIMEDIFF** adalah fungsi untuk membandingkan jam
- **TIME_TO_SEC** adalah fungsi untuk merubah jam menjadi detik
- **SUM** adalah fungsi untuk menjumlahkan
- **SEC_TO_TIME** adalah fungsi untuk merubah detik menjadi jam

2.25 MySQL Unicode UTF8

Di bumi tempat kita tinggal ini terdapat banyak bahasa, evolusi bahasa yang terjadi di setiap daerah telah melalui masa yang sangat lama. Salah satu faktor terjadi evolusi bahasa adalah karena daerah tempat tinggal manusia.

Semakin banyaknya bahasa akan diikuti pula dengan hurup dan angka untuk merepresentasikan bahasa dalam tulisan. seperti bahasa Arab, Thailand, China, Jepang, Jawa, india mempunyai bentuk hurup dan angka yang berbeda. Database yang berfungsi sebagai penyimpan data seharusnya telah dilengkapi fitur untuk dapat menerima karakter-karakter non-latin.

Pada MySQL Database telah di lengkapi dengan fitur utf8 (Unicode Transformation Format with 8-bit units) Character Set yang berfungsi sebagai alternatif untuk menyimpan Unicode Data. Character Set tipe utf8 dapat dimanfaatkan untuk menyimpan data-data karakter non latin seperti Arab, Korea, Japan, China.

Unicode adalah suatu standar industri yang dirancang untuk mengizinkan teks dan simbol dari semua sistem tulisan di dunia untuk ditampilkan dan dimanipulasi secara komputer. Unicode dikoordinasi dan dikembangkan oleh suatu organisasi nirlaba yaitu Unicode Consortium

Database MySQL yang penulis gunakan dalam eksperimen ini adalah versi 5.1 dengan Operation System Ubuntu 11.04 - the Natty Narwhal . Berikut adalah contoh beberapa nama kota yang mengandung karakter selain latin di negara Argentina.

a. Penampakan karakter apabila database tidak dapat menerima karakter selain latin.

loc_id	c_id	state_id	loc_name	loc_status	loc_desc
AR.LujndeCuyo	AR	AR.Mendoza	LujÃ¡n de Cuyo	NULL	NULL
AR.Malarge	AR	AR.Mendoza	MalargÃ¼e	NULL	NULL

Pada gambar diatas tampak nama lokasi yang terdapat pada field loc_name mengandung karakter yang tidak diterjemahkan dengan baik.

b. Berikut ini adalah penampakan karakter yang databasenya dapat menerima karakter selain latin

loc_id	c_id	state_id	loc_name	loc_status	loc_desc
AR.LujndeCuyo	AR	AR.Mendoza	Luján de Cuyo	NULL	NULL
AR.Malarge	AR	AR.Mendoza	Malargüe	NULL	NULL

Pada gambar diatas tampak nama lokasi non latin pada field loc_name berhasil diterjemahkan dengan baik.

Berikut ini adalah cara membuat Database yang support UTF 8, dicontoh ini database bernama "kamus" sebagai sample nama database .

```
CREATE DATABASE kamus DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci
```

Apabila informasi dari database dilihat maka yang menggunakan utf8 akan tampak seperti dibawah ini

```
SHOW CREATE DATABASE kamus;
```

```
+-----+-----+-----+-----+
| Database | Create Database |
+-----+-----+-----+-----+
| kamus | CREATE DATABASE `kamus` /*!40100 DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci */ |
+-----+-----+-----+-----+
```

Dengan memberikan set utf8 saat membuat database maka secara otomatis disaat membuat table pada database tersebut akan diberikan utf8. Dibawah ini adalah contoh tabel bernama "artikata" dari database kamus yang menggunakan utf8.

```
SHOW CREATE TABLE artikata ;

CREATE TABLE `artikata`
  `id` bigint(20) NOT NULL DEFAULT '0',
  `istilah` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `deskripsi` text COLLATE utf8_unicode_ci,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
```

Pada contoh diatas "artikata" adalah nama table yang memiliki utf8, hal tersebut dapat terlihat pada CHARSET=utf8 COLLATE=utf8_unicode_ci.

Pada field-field table "artikata" bertipe data char secara otomatis akan diberikan utf8, hal tersebut dapat dilihat di tulisan COLLATE utf8_unicode_ci pada setiap field .

- `istilah` varchar(255) COLLATE utf8_unicode_ci
- `deskripsi` text COLLATE utf8_unicode_ci

Pada sebuah kasus mungkin charset database yang digunakan terlanjur bukan utf8, tetapi kita memerlukan table yang dapat menyimpan karakter selain latin. Berikut ini cara membuat table agar menyertakan charset utf8 .

```
create table test (id int, nama varchar(255), primary key(id) ) DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

SQL diatas adalah membuat table bernama "test" ditambah Default Character Set utf8 Collate utf8_unicode_ci yang berfungsi memberikan charset utf8 . Andaikan yang perlu diubah ke utf8 hanya field tertentu Anda dapat menggunakan perintah modifikasi field seperti dibawah ini.

```
ALTER TABLE test MODIFY description TEXT CHARSET utf8 COLLATE utf8_unicode_ci;
```

SQL diatas terdapat CHARSET utf8 COLLATE utf8_unicode_ci yang berfungsi memberikan charset utf8 pada field description pada table test.

2.26 Prepare Statement

Pada umumnya sebuah aplikasi terdapat fasilitas pencarian data. SQL dalam pencarian data umumnya berupa query select yang memiliki kriteria where dengan sebuah atau beberapa field. Apabila field dari kriteria where tersebut terdapat banyak pilihan misalkan sebagai berikut.

- Ingin mencari data judul buku menggunakan field Judul Buku
- Ingin mencari data penerbit hanya menggunakan field penerbit
- Ingin mencari data penulis hanya menggunakan field penulis

Untuk keperluan tersebut pada umumnya memerlukan tiga buah SQL yaitu sebagai berikut.

- SQL SELECT dengan WHERE field judul buku untuk mencari judul buku
- SQL SELECT dengan WHERE field penerbit untuk mencari penerbit buku
- SQL SELECT dengan WHERE field penulis untuk mencari penulis buku

Apabila hanya menulis tiga buah SQL mungkin tidak apa-apa, tetapi bila mencapai belasan SQL akan terasa tidak efisien, untuk keperluan tersebut pada MySQL dapat menggunakan fitur Prepare Statement.

Prepare Statement berguna ketika ingin menjalankan query yang memiliki struktur SQL yang mirip sehingga dengan menggunakan Prepare Statement menulis sript SQL menjadi lebih singkat.

Pada kasus ini saya membuat Prepare Statemen di dalam Store Procedure untuk membuat sebuah query menjadi lebih dinamis, sebagai contoh terdapat sebuah query yang memiliki kriteria where dengan field kunci-nya berubah-ubah.

Pada contoh dibawah ini saya menuliskannya dalam bentuk Store Procedure agar terlihat manfaat penggunaan Prepare Statement.

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_barang_view`(v_keyfield varchar(255), v_keyword
varchar(255))
begin
  set @qry = concat('select * from barang where ', v_keyfield, ' = "', v_keyword, '"');
  prepare exec from @qry;
  execute exec;
end $$
DELIMITER;
```

Perintah diatas adalah sebuah query Select untuk menampilkan data barang dengan kriteria field pencarian yang berbubah-ubah tergantung dari nilai paramater v_keyfield yang diberikan dari Store Procedure tersebut.

- **SET** adalah untuk pemberian nilai ke dalam sebuah variable
- **CONCAT** adalah fungsi mysql untuk mengabungkan string
- **PREPARE FROM** untuk melakukan persiapan eksekusi query
- **EXECUTE** perintah untuk melakukan eksekusi query

2.27 Repair Table

Kejadian database yang tiba-tiba rusak atau korup dapat terjadi pada Database MySQL dimanapun hal tersebut biasanya terjadi karena listrik tiba-tiba mati karena Power Supply komputer yang rusak, Konslate atau disebabkan aliran listrik dari PLN terputus.

Untuk meminimalisir dampak buruk dari kejadian tersebut maka pada komputer server perlu dipasang stabilizer dan UPS untuk komputer server yang harus available setiap saat serta dilengkapi dengan Generator Listrik (Genset) untukantisipasi ketika listrik mati maka secara otomatis UPS akan menggantikan supply listrik ke Komputer Server hingga Genset hidup atau Listrik dari PLN kembali normal.

Apabila kerusakan database telah terjadi, Anda dapat menggunakan fitur yang dimiliki MySQL untuk Repair Table. Cara Repair table yang saya jelaskan disini adalah untuk table-table yang menggunakan Store Engine MyISAM.

Store Engine adalah sebuah engine yang berisikan mekanisme tentang bagaimana data disimpan. Pada MySQL terdapat banyak Storage Engine diantaranya adalah MyISAM, InnoDB, Federate, Memory.

Apabila database yang digunakan untuk menyimpan data-data berjenis transaksi seperti keuangan, penjualan direkomendasikan menggunakan Storage Engine berjenis Transaksional seperti InnoDB

tetapi apabila database tersebut digunakan untuk data bukan transaksional seperti blog disarankan menggunakan storage engine yang berjenis seperti MyISAM karena akan memberikan kecepatan yang lebih baik.

Berikut ini adalah sebuah contoh untuk memeriksa table yang korup, rusak atau error pada sebuah database bernama sekolah. Pada console jalankan perintah `mysqlcheck` kemudian masukan parameter username & password agar bisa akses ke Database MYSQL serta nama database yang akan diperiksa.

```
mysqlcheck -uroot -ppass sekolah;
```

```
sekolah.siswa          error
sekolah.pegawai       OK
sekolah.kelas        OK
sekoLAH.mata_pelajaran OK
```

Apabila hasil pemeriksa tidak terdapat masalah pada table yang diperiksa maka keterangan pada table tersebut akan bertuliskan OK. Apabila terdapat masalah akan bertuliskan warning atau error seperti yang terjadi pada table `sekolah.siswa` diatas.

Untuk memperbaiki error dari table `sekolah.siswa` diatas pada console ketikkan perintah seperti dibawah ini.

```
mysqlcheck -uroot -ppass -r sekolah;
```

Pada perintah diatas terdapat parameter `-r` yang mengartikan

melakukan proses repair/perbaiki apabila ditemukan table yang rusak, korup atau error.

Selain menggunakan `mysqlcheck` untuk memeriksa dan memperbaiki tabel anda dapat menggunakan perintah `check and repair table`. Berikut ini cara menggunakan `check table`.

1. Login terlebih dahulu kedalam console mysql, seperti contoh ini

```
mysql -uroot -ppass
```

2. Pilih database yang akan diperiksa, sebagai contoh adalah database sekolah

```
mysql> use sekolah;
```

3. Lakukan perintah cek pada table yang diduga rusak atau korup

```
mysql> CHECK TABLE siswa;
```

```
+-----+-----+-----+-----+
| Table      | Op    | Msg_type | Msg_text |
+-----+-----+-----+-----+
| sekolah.siswa | check | status   | error    |
+-----+-----+-----+-----+
```

Hasil dari pemeriksaan bahwa table siswa terjadi error yang ditunjukkan pesannya pada kolom `Msg_text` diatas.

4. Lakukan perbaikan pada table bermasalah dengan menggunakan

perintah Repair

```
mysql> REPAIR TABLE siswa;
```

```
+-----+-----+-----+-----+
| Table      | Op      | Msg_type | Msg_text |
+-----+-----+-----+-----+
| sekolah.siswa | check   | status   | OK       |
+-----+-----+-----+-----+
```

Untuk memeriksa Storage Engine yang digunakan sebuah table dapat menggunakan perintah Show Create Table seperti contoh dibawah ini .

```
SHOW CREATE TABLE siswa;
```

```
siswa | CREATE TABLE `siswa` (  
  `id` bigint(20) NOT NULL DEFAULT '0',  
  `nama` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `tgl_lahir` date  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci |
```

Pada tampilan diatas tertera ENGINE=MyISAM yang mengartikan bahwa tabel tersebut menggunakan store engine MyISAM

2.28 Into Select

Proses input data pada database adalah sesuatu yang sangat wajar terjadi yang dapat dilakukan melalui aplikasi atau dengan menuliskan SQL secara native. Untuk beberapa keperluan terkadang memerlukan input data yang dapat berasal dari file seperti *.txt, *.csv atau dari sebuah table.

Apabila data yang dimasukan hanya beberapa buah. Cara memasukan data satu persatu ke table database melalui SQL mungkin terasa ringan, tetapi bila jumlah data tersebut berjumlah puluhan ribu, rasanya sangat melelahkan dan memerlukan waktu yang lama untuk menyelesaikannya

Pada Database MySQL memiliki fitur bernama **Into Select**. Fitur tersebut berguna untuk memasukan data ke sebuah table dengan sumber data yang berasal dari table yang lain.

Sebagai contoh saya memiliki data negara asia yang tersimpan pada table bernama negara_asia, kemudian saya perlu memindahkan seluruh isi data tersebut ke table baru bernama negara yang merupakan tempat baru untuk menampung seluruh data negara.

Database MySQL yang saya gunakan oleh penulis adalah versi 5.1 dengan Operation System Ubuntu 11.04 - the Natty Narwhal. Berikut ini adalah cara melakukan Into Select pada kasus diatas.

- Apabila data pada table negara_asia dibuka akan tampak seperti dibawah ini.

```
mysql> select * from negara_asia;
```

```
+----+-----+-----+
| id | nama   | ibu_kota |
+----+-----+-----+
| 1  | Indonesia | Jakarta  |
| 2  | Jepang   | Tokyo    |
| 3  | China    | Beijing  |
+----+-----+-----+
3 rows in set (0.00 sec)
```

- Apabila data table negara dibuka akan tampak seperti dibawah ini.

```
mysql> select * from negara;
```

```
+----+-----+-----+
| id | nama   | ibu_kota |
+----+-----+-----+
| 1  | England | London   |
| 2  | USA     | Washington DC |
| 3  | Australia | Canberra |
+----+-----+-----+
3 rows in set (0.00 sec)
```

Berikut ini adalah cara import data dari table negara_asia ke table negara dengan menggunakan perintah Into Select.

- Pola Into Select

```
INSERT INTO [nama table tujuan] ([nama field tujuan])
SELECT [nama field sumber]
FROM [nama table sumber];
```

- Contoh penerapan Into Select

```
INSERT INTO negara (nama,ibu_kota)
SELECT nama,ibu_kota
FROM negara_asia;
```

- Berikut ini adalah hasil dari proses Into Select dari negara_asia ke table negara

```
mysql> select * from negara;
```

```
+---+-----+-----+
| id | nama   | ibu_kota   |
+---+-----+-----+
| 1  | England | London     |
| 2  | USA     | Washington DC |
| 3  | Australia | Canberra  |
| 4  | Indonesia | Jakarta   |
| 5  | Jepang  | Tokyo      |
| 6  | China   | Beijing    |
+---+-----+-----+
6 rows in set (0.00 sec)
```

2.29 Show Processlist

Show Processlist adalah sebuah fitur pada MySQL untuk menunjukkan thread yang berjalan pada Server MySQL. Fitur ini berguna untuk monitoring MySQL Server yang memiliki banyak user dan koneksi sehingga bisa mendapatkan pesan error apabila terjadi error disebabkan oleh banyaknya koneksi.

Untuk menggunakan fitur processlist Anda harus login terlebih dahulu sebagai root, berikut ini adalah contoh untuk login sebagai root.

```
mysql -uroot -ppass
```

- Berikut ini adalah contoh sebuah tampilan Show Process list

```
mysql> SHOW FULL PROCESSLIST;
+----+-----+-----+-----+-----+-----+-----+-----+
| Id  | User      | Host      | db      | Command | Time | State | Info          |
+----+-----+-----+-----+-----+-----+-----+-----+
| 286 | root      | localhost | mysql   | Query   | 0    | NULL  | SHOW FULL PROCESSLIST |
| 289 | yuli      | localhost | sms     | Sleep   | 726  |       | NULL          |
| 290 | websekolah | localhost | websekolah | Sleep   | 717  |       | NULL          |
+----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Pada gambar table diatas erdapat tiga buah thread yang dimiliki oleh tiga user. berikut ini adalah penjelasan dari setiap thread tersebut

- **Thread 286** pemilik root melakukan akses ke database mysql menjalankan sebuah query yang berisikan SHOW FULL PROCESSLIST
- **Thread 289** pemilik yuli melakukan akses ke database sms tidak melakukan apa-pun atau sleep.
- **Thread 290** pemilik websekolah melakukan akses ke database websekolah tidak melakukan apa-pun atau sleep.

Berikut ini adalah penjelasan lebih lengkap dari dari setiap kolom yang ditampilkan dari Show Process list.

- **Id** adalah sebagai nomor identifikasi sebuah thread
- **User** dan **Host** adalah untuk menunjukkan user pemilik thread,

apabila terdapat user bernama system user itu mengartikan buka client melainkan system yang sedang melakukan tugas-tugas internal.

- **db** adalah mengindikasikan database yang digunakan pada thread.
- **Command & State** mengartikan apa yang sedang dilakukan oleh thread. Berikut ini adalah contoh dari command.
 1. Query mengartikan thread sedang melakukan eksekusi sebuah perintah.
 2. sleep mengartikan thread tersebut tidak melakukan apapun.
- **Time** menunjukkan waktu yang telah dihabis pada sebuah command dan status.
- **Info** berisikan statemen yang di eksekusi oleh thread.

Show Processlist selain berguna untuk monitoring thread dapat memiliki fungsi untuk menghentikan sebuah thread, biasanya thread yang dihentikan adalah thread "zombie" yaitu sebuah thread yang tidak melakukan aktivitas apapun atau thread yang dicurigai bermasalah seperti sebuah query yang terlampau lama dalam menyelesaikan eksekusi sehingga menyebabkan server lumpuh. Untuk menghentikan sebuah thread dapat diberikan dengan perintah kill diikuti id thread.

- Contoh untuk menghentikan sebuah thread 289 milik user yuli

```
mysql> KILL 289;
```

- Hasil Show Processlist setelah dilakukan kill
- Berikut ini adalah contoh sebuah tampilan Show Process list

```
mysql> SHOW FULL PROCESSLIST;
+----+-----+-----+-----+-----+-----+-----+-----+
| Id  | User      | Host      | db      | Command | Time | State | Info          |
+----+-----+-----+-----+-----+-----+-----+-----+
| 286 | root      | localhost | mysql   | Query   | 0    | NULL  | SHOW FULL PROCESSLIST |
| 290 | websekolah | localhost | websekolah | Sleep   | 717  | NULL  |                |
+----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```