

[Backup MySQL Database Secara Periodik dalam Server Solaris 10](#)

Setelah dua tahun terakhir ini bergelut dengan Solaris Server, saya bersyukur banyak ilmu baru yang dapat digali selama masa tersebut, dan hal tersebut membuat saya ingin berbagi dengan rekan – rekan semua. Dalam kesempatan ini saya ingin berbagi bagaimana cara backup database MySQL secara periodik, menjadi berupa sebuah file *.tar.gz yang telah dikompres, hal ini dapat dilakukan dengan menggunakan 'time scheduler' jika menggunakan os windows, tetapi apa yang akan kita gunakan jika bekerja dalam lingkuan Solaris sebagai server, berikut pembahasannya.

Yang pertama harus diketahui adalah bagaimana proses 'dump and restore' dalam database MySQL, sebagaimana yang telah tercantum dalam manual documentation proses tersebut dapat dilakukan dengan cara eksekusi perintah sebagai berikut:

```
mysqldump -u username --password=pass_user --databases schema_db > suatu_file.sql
```

Perintah diatas merupakan contoh perintah proses backup database ke dalam file *.sql, untuk sekilas kita dapat menguraikan perintah diatas sebagai berikut; 'mysqldump' merupakan salah satu dari beberapa perintah dasar dalam perintah database MySQL untuk proses save database ke file, '-u' adalah properti mysql untuk mengakses username, yang merupakan komponen penting untuk mengakses database MySQL, 'username' adalah otorisasi pengguna yang dipakai untuk dapat mengakses dan melakukan manipulasi perintah dalam database, untuk hal ini akan menggunakan user 'root', '--password=pass_user', sedangkan password adalah properti MySQL yang harus dicantumkan apabila pengguna hendak mengakses database sebagai key otentikasi, sedangkan 'schema_db' adalah nama database schema yang akan diambil datanya menjadi file sql dump.

Setelah proses 'dump' dari database schema menjadi file 'suatu_file.sql' selesai, maka proses selanjutnya adalah menyimpan file tersebut menjadi file yang lebih ringkas dengan menggunakan ekstension '*.tar.gz', dengan cara sebagai berikut:

```
tar -cf suatu_file.sql | gzip -c > /folder_tujuan/suatu_file_tar.tar.gz
```

Maka didapatkan file sql dump yang sudah di zip menjadi file tar.gz, setelah itu apa lagi yang harus dilakukan? Berikutnya adalah menghapus file sql dari folder sehingga yang tersisa hanya file tar.gz saja dengan menggunakan perintah 'rm'.

```
rm *.sql
```

Setelah kita mengetahui beberapa perintah utama dalam melakukan proses autobackup dalam sebuah server, sekarang kita akan menggabungkan kumpulan perintah – perintah tersebut kedalam sebuah file 'sh' yang dapat dieksekusi. Jika kita memisalkan sebuah server database yang menggunakan solaris platform dengan database schema bernama 'gienet' yang akan disave ke sebuah folder '/export/home0/gienet_backup', dengan format penamaan file backup 'gienet-yyyy-MM-dd-HH-mm-ss.sql', yang kemudian akan di zip dengan format penamaan 'gienet-yyyy-MM-dd-HH-mm-ss.tar.gz', proses backup ini akan berlangsung ketika jam sistem menunjukkan pukul 00:30. Maka prosesnya akan seperti berikut:

Buatlah sebuah file dengan nama backup-script.sh (nama tidak wajib menggunakan nama ini), dengan perintah 'touch backup-script.sh' didalam shell console, kemudian lakukan editing dengan perintah 'vi backup-script.sh', dimulai dengan baris pertama berisi '#!/bin/sh' dengan diikuti beberapa baris comment dengan contoh seperti berikut:

```
#!/bin/sh
#
# Ini adalah contoh membuat executable file backup database mysql
#
```

Dilanjutkan dengan deklarasi variable tanggal untuk menambahkan format tanggal pada nama file, diikuti dengan deklarasi nama folder yang akan menjadi tempat menyimpan file backup tersebut.

```
NEW_DATE='date +%Y-%m-%d-%T'
NEW_DIR='/export/home0/gienet_backup'
```

Kemudian dilanjutkan dengan script backup dari database, dengan asumsi kita menggunakan user root untuk melakukan proses backup dan 'admin' sebagai passwordnya, maka command script selanjutnya yang harus ditulis adalah:

```
mysqldump -u root -password=admin --databases gienet > $NEW_DIR/gienet-$NEW_DATE.sql
```

Setelah proses mysqldump selesai dan file 'gienet-yyyy-MM-dd-HH-mm-ss.sql' dilanjutkan dengan kompres file menjadi ukuran lebih kecil ke dalam file tar.gz dengan perintah sebagai berikut:

```
cd $NEW_DIR
tar -cf gienet-$NEW_DATE.sql | gzip -c > $NEW_DIR/gienet-$NEW_DATE.tar.gz
```

Lakukan proses hapus file sql didalam folder yang sudah di setting dalam variable '\$NEW_DIR' dengan tujuan file yang tersisa didalam folder itu hanya tar.gz dan memelihara space harddisk supaya tidak penuh oleh file yang isinya sama, dengan perintah berikut:

```
rm $NEW_DIR/*.sql
```

Selesai sudah membuat automatic MySQL database backup script dalam file 'backup-script.sh', lakukan proses saving pada aktif editor 'vi' dengan perintah ':wq!'. Proses dilanjutkan dengan menyimpan file tersebut ke folder '/usr/bin' dan memberikan hak akses system terhadap file tersebut dengan perintah 'chmod +x backup-script.sh', kemudian daftarkan file tersebut ke solaris scheduler yaitu 'crontab', dengan terlebih dahulu setting default editor dalam unix shell. Dengan urutan perintah sebagai berikut:

```
#export DISPLAY=vi
#crontab -e
```

Maka unix shell akan menghasilkan output seperti berikut:

```
#ident "@(#)root 1.21 04/03/23 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

Arahkan cursor pada kolom terakhir baris terakhir lalu tekan enter, kemudian masukan script untuk menjalankan file backup-script.sh untuk setiap jam 00:30 setiap hari diserver dengan script sebagai berikut:

```
#
# Script for autobackup database
#
30 0 * * * sh /usr/bin/backup-script.sh
```

Untuk mengetahui lebih lanjut tentang crontab script dapat dilihat di <http://adminschoice.com/crontab-quick-reference>, untuk mengetahui script ini berjalan seperti yang diinginkan, keesokan paginya dapat dilihat di folder '/export/home0/gienet_backup' apakah ada file 'gienet-yyyy-MM-dd-HH-mm-ss.tar.gz' jika ada berarti script sukses tanpa ada error, jika tidak ada berarti masih ada error dalam script.

Mungkin saat ini hanya sekian ilmu yang bisa dibagi dengan pembaca semoga bermanfaat, jika ada kekurangan atau masukan yang dapat meningkatkan kemampuan jangan sungkan untuk memberikan komentar.