

Kali ini saya membahas tentang cara membackup sebuah website ataupun blog secara otomatis yang mencakup MySQL database serta file-filenya lalu menguploadnya ke FTP menggunakan script bash.

Hal yang paling sering dilupakan para pemilik situs ataupun blog adalah membuat backup padahal backup merupakan hal yang sangat penting dan wajib kita lakukan, entah itu perjam, harian, ataupun mingguan.

Idealnya kita mempunyai dua buah server, satu untuk web servernya dan yang satu lagi hanya untuk FTP servernya jadi misal ada masalah di web server utama kita maka file backupnya tidak akan terpengaruh.

Membackup MySQL database

Langsung saja ke bagian membackup MySQL database lalu menguploadnya ke FTP.

Pertama buatlah file bash misal db-backup.sh:

```
vi db-backup.sh
```

Lalu isikan seperti berikut:

```
#!/bin/sh # Database yang ingin kita backup # Pisahkan dengan spasi untuk tiap database
databases="mydb1 mydb2" # Waktu saat ini date=$(date +"%Y-%m-%d") # User dan
password dari database # Gunakan root supaya lebih enak user=username pass=password
# User, password, dan alamat dari FTP servernya # Untuk folder sesuai selera saja
ftpUser=username ftpPass=password ftpHost=ftp.example.com ftpFolder="backup/" #
Tempat menyimpan database # Ini di webserver dan bukan di FTP server
bPath="/var/backups/databases" # Buat folder bPath diatas jika belum ada if [ ! -d $bPath ];
then mkdir -p $bPath fi # Hapus file backup di bPath jika umurnya melebihi 3 hari find
$bPath/*.sql.gz -mtime +3 -exec rm {} ; # Mulai membackup database for db in $databases;
do # Nama dari file backupnya file=$db-$date.sql.gz # Membackup database dengan
```

Auto Backup MySQL Database dan File ke FTP Server

Written by BiRU

Tuesday, 31 May 2016 21:00 -

```
mysqldump echo "Starting to dump the $db database as $file" mysqldump --user=$user
--password=$pass $db | gzip -9 > $bPath/$file # Upload file tadi ke FTP menggunakan
CURL echo "Starting to upload the $file to FTP server" curl --ftp-create-dirs -T
$bPath/$file -u $ftpUser:$ftpPass ftp://$ftpHost/$ftpFolder done # Clear cache. Hanya untuk
KVM, Xen # ataupun dedicated server free && sync && echo 3 > /proc/sys/vm/drop_caches
&& echo "" && free
```

Simpan dan pastikan script tadi bisa kita eksekusi:

```
chmod +x db-backup.sh
```

Catatan: ganti username dan password database dengan data kalian. Juga sesuaikan data untuk FTP servernya.

Mambackup file dan folder

Nah sekarang giliran membuat file bash untuk membackup file dan folder dari website kita.

Buat file bash misal fd-backup.sh

```
vi fd-backup.sh
```

Tambahkan script berikut:

```
#!/bin/sh # Folder yang ingin kita backup fName=("example1_com example2_com") #
Parent folder dari folder yang ingin kita backup # Biasanya public_html
pFolder="/var/www/html" # Local backup. Tempat dimana kita menaruh backup # di
webserver kita bPath="/var/backups/files" # Waktu sekarang date=$(date +"%Y-%m-%d")
# User, password, dan alamat dari FTP servernya # Untuk folder sesuai selera saja
ftpUser=userfName ftpPass=password ftpHost=ftp.example.com ftpFolder="backup/" # Buat
folder bPath diatas jika belum ada if [ ! -d $bPath ]; then mkdir -p $bPath fi # Hapus file
backup di bPath jika umurnya melebihi 3 hari find $bPath/*.zip -mtime +3 -exec rm {} ; #
Mulai mambackup foldernya for fd in $fName; do # Nama dari file backupnya
file=$fd-$date.zip # Kita zip foldernya echo "Starting to zip the folder and files" cd
$pFolder zip -r $bPath/$file $fd # Upload file tadi ke FTP menggunakan CURL echo
"Starting to upload the $file to FTP server" curl --ftp-create-dirs -T $bPath/$file -u
$ftpUser:$ftpPass ftp://$ftpHost/$ftpFolder done # Clear cache. Hanya untuk KVM, Xen #
ataupun dedicated server free && sync && echo 3 > /proc/sys/vm/drop_caches && echo "" &&
free
```

Simpan dan lakukan chmod supaya bisa dieksekusi:

```
chmod +x fd-backup.sh
```

Catatan: seperti sebelumnya, ganti semua data dengan data kalian.

Membuat cron jobs

Setelah backup script sudah dibuat maka kita perlu mengeksekusinya menggunakan cron supaya berjalan otomatis pada waktu tertentu. Baca [membuat cron jobs di Linux](#) untuk lebih jelasnya.

Buka crontab:

```
crontab -e
```

Isikan perintah berikut:

```
0 2 * * * /root/db-backup.sh 0 3 * * * /root/fd-backup.sh
```

Setelah kita simpan maka restart cron:

```
service crond restart
```

Untuk waktu saya memilih jam 2 dini hari untuk membackup MySQL databasenya dan jam 3 dini hari untuk file dan foldernya.

Kita juga bisa mencoba dahulu apakah script backup tadi bekerja sesuai dengan yang kita harapkan. Eksekusi saja:

```
/root/db-backup.sh /root/fd-backup.sh
```

Setelah selesai kalian cek di FTP servernya ada tidak file-file backup kita.