

Sinkronisasi File dengan rsync

Dalam bekerja dengan lebih dari satu *host*, pertukaran dan sinkronisasi file adalah hal yang sangat umum dihadapi. Walaupun terdengar sederhana, sinkronisasi file bisa memicu proses *maintenance* yang berat apabila tidak dikelola dengan baik. Untuk melakukan sinkronisasi file, kita bisa mempergunakan beberapa cara. Salah satunya adalah dengan mempergunakan rsync. Kita akan membahas bagaimana mengatur rsync server dan mempergunakan rsync.

Kalau mau enak, beginilah bayangannya. Kita tinggal di lingkungan *wireless* di mana tidak perlu ada kabel-kabel yang menghambat. Lalu, semua juga mempergunakan notebook yang baterainya bisa tahan sampai 20 jam sehingga cukup di-charge sehari sekali (atau bahkan kurang). Selanjutnya, ada server yang selalu hidup dan memiliki *hardware* yang sangat bisa diandalkan. Lalu, sebagai perekatnya, ada koneksi broadband dengan harga yang masuk akal dan senantiasa bisa menyambungkan kita dengan server setiap saat.

Dengan demikian, kita bahkan bisa menjalankan aplikasi dari server, sehingga Application Service Provider benar-benar menjadi nyata dan kita bisa menyimpan file-file kita di server tanpa harus selalu melakukan *download/upload* berulang-ulang untuk sekadar melakukan sinkronisasi file lokal dan *remote*.

Namun, ini semua terlalu ideal. Jadi, untuk saat ini, belum bisa terwujud. Oleh karena itu, kita kembali ke permasalahan utama: sinkronisasi file. Masalah ini mungkin terdengar sederhana dan mengada-ada. Tapi, apabila tidak dikelola dengan baik, sinkronisasi dapat memicu usaha *maintenance* yang besar.

Bayangkan saja skenario berikut, yang cukup sederhana. Anda adalah seorang *web developer* yang sayangnya tidak selalu terkoneksi ke Internet. Anda membuat website di lokal dan apabila sudah mencapai titik tertentu, Anda meng-upload ke Internet. Melalui FTP atau SCP misalnya. Anda selalu menghapus website yang lama di Internet, dan menimpa dengan yang Anda kirimkan dari lokal. Namun terkadang,

untuk kesalahan kecil pada web, Anda akan kerepotan apabila harus menghapus keseluruhan, meng-update lokal dan mengirimkan kembali ke server. Anda pun masuk ke shell server dan mengedit satu dua hal yang sederhana. Setelah beberapa saat, karena perubahannya kecil, Anda lalu lupa dan kembali bekerja seperti biasa.

Minimal terdapat dua hal yang tidak efisien atau bahkan bisa memicu masalah. Pertama, sebenarnya tidak perlu melakukan siklus hapus-upload seperti disebutkan. Yang kedua, Anda mungkin harus mengedit berulang kali file di server karena: 1) Anda lupa mengaplikasikan ke lokal dan 2) ketika melakukan update lagi, Anda juga menghapus perubahan di server karena menghapus file di server tersebut. Untuk kondisi pertama, di mana terjadi siklus hapus-upload, kita sebenarnya memboroskan *bandwidth*. Karena tidak yakin file mana saja yang berubah, maka yang termudah adalah meng-upload ulang. Untuk yang kedua, kita sebenarnya sedang melakukan hal konyol dengan mengedit berulang-ulang perubahan kecil.

Ini hanyalah masalah sederhana dari sinkronisasi file. Yang bekerja juga Anda sendiri. Bagaimana kalau yang bekerja adalah banyak orang, dan lain sebagainya. Yang ingin ditekankan adalah, masalah sinkronisasi file tidak bisa dianggap remeh dan sejak awal, harus dikelola dengan baik.

Kita mengenal berbagai cara untuk melakukan sinkronisasi lokal ke *remote*. Cara pertama telah disebutkan dalam contoh. Menghapus satu bagian dan menimpa dengan bagian lain. Tentu saja, cara ini tidak efisien. Cara kedua yang lebih baik adalah

dengan menggunakan program tertentu, seperti rsync.

Rsync adalah utiliti yang sangat hebat dari Andrew Tridgell (pembuat SAMBA; website rsync sendiri di host di [samba.org](http://rsync.samba.org); <http://rsync.samba.org>). Rsync dapat meng-copy hanya perubahan file, dapat pula dikompres dan dapat mempergunakan ssh untuk menjaga keamanan. Ini jauh lebih baik dengan script FTP yang meng-copy-kan keseluruhan file dan tidak mempergunakan protokol yang aman. Berikut ini adalah penjelasan lebih lanjut:

- Meng-copy-kan perubahan. Hanya bagian file yang berubah yang ditransfer. Ini akan membuat proses update berjalan lebih cepat, terutama untuk kita yang menghadapi masalah koneksi Internet lambat. Sebagai perbandingan, FTP akan meng-copy-kan keseluruhan file, walaupun hanya 1 byte yang berubah.
- Kompresi. Bagian yang berubah tadi, yang akan dikirimkan, masih dapat pula dikompres sehingga akan memperkecil ukuran file yang benar-benar dikirimkan. Rsync mempergunakan *zlib*, yang sudah teruji.
- Kerjasama dengan secure shell. Sejak awal, masalah ini harus kita perhatikan. Masalah keamanan adalah masalah fundamental. Rsync dapat mempergunakan ssh untuk mengenkripsi sesi rsync.

Di luar semua keunggulan tersebut, berikut ini adalah beberapa fitur utama rsync:

- Mendukung peng-copy-an link, device, dan dapat mempertahankan atribut seperti pemilik, group, dan hak akses secara umum.

- Kita dapat memberikan catatan kepada rsync untuk tidak memasukkan file-file tertentu dalam proses sinkronisasi. Ini juga berlaku untuk CVS, untuk mengabaikan file yang CVS abaikan.
- Tidak membutuhkan hak root.
- Melakukan pipelining transfer untuk meminimasi latensi.
- Mendukung akses anonim dan user.

Anda dapat mempergunakan rsync untuk berbagai kebutuhan. Berikut ini adalah kegunaan rsync:

- Sinkronisasi lokal. Dengan demikian, rsync akan berfungsi seperti halnya program cp biasa, namun dengan kemampuan mendeteksi perubahan yang hebat. Tidak ada koneksi ke host lain dalam hal ini.
- Untuk meng-copy-kan file dari lokal ke remote mempergunakan remote shell seperti ssh.
- Untuk meng-copy-kan file dari remote ke lokal mempergunakan remote shell seperti ssh.
- Untuk meng-copy-kan file dari server rsync ke lokal.
- Untuk meng-copy-kan file dari lokal ke server rsync.
- Untuk meng-copy-kan file dari remote ke lokal dengan mempergunakan remote shell seperti ssh, mempergunakan server rsync di remote.
- Untuk meng-copy-kan file dari lokal ke remote mempergunakan remote shell seperti ssh, mempergunakan server rsync di remote.
- Untuk melihat daftar file di remote.

Untuk bekerja dengan rsync, kita dapat mengatur sebuah mesin sebagai rsync server dan satu mesin sebagai rsync client. Sebenarnya, terdapat banyak metode yang bisa kita lakukan, namun kita akan mengambil pendekatan sederhana untuk sinkronisasi file lokal ke remote dan sebaliknya.

Pertama-tama, Anda harus memiliki rsync terinstal di sistem Anda. Cobalah memberikan perintah rsync dan seharusnya, hampir semua distro telah menginstal program ini pada saat instalasi distro (paket rsync). Apabila belum terinstal, cobalah cari ke CDROM atau DVD distro Anda. Hampir pasti akan ditemukan. Kalaupun tidak, cobalah cari rsync untuk distro Anda di in-

ternet, sebelum Anda melakukan kompilasi dari source code.

Rsync lokal ke lokal

Untuk contoh sederhana, sebelum kita melanjutkan ke pembahasan rsync antar-host, kita akan berlatih untuk mempergunakan rsync dari lokal ke lokal. Untuk itu, penulis akan memiliki direktori A dan B, di mana isi B akan diselalu disinkronkan dengan isi A. Harap diperhatikan bahwa kita tidak membutuhkan paket server dan client, cukup paket rsync saja. Paket ini akan berisikan di antaranya program rsyncd dan rsync, yang berfungsi sebagai server dan client. Sebenarnya, keduanya adalah file yang sama (tipe multi call).

Isi direktori A:

```
drwxr-xr-x  2 nop users 192
2004-12-19 12:02 ./
drwxr-xr-x  4 nop users  96
2004-12-19 12:00 ../
-rw-r--r--  1 nop users  14
2004-12-19 12:01 file1
-rw-r--r--  1 nop users   6
2004-12-19 12:01 file2
-rw-r--r--  1 nop users  17
2004-12-19 12:01 file3
-rw-r--r--  1 nop users  18
2004-12-19 12:01 file4
-rw-r--r--  1 nop users   2
2004-12-19 12:02 file5
-rw-r--r--  1 nop users   8
2004-12-19 12:02 file6
```

Pada awalnya, direktori B belum berisi apapun.

Pertama-tama, kita akan menyinkronisasi terlebih dahulu A dan B, agar B memiliki isi yang sama dengan A. Berikut ini adalah perintahnya.

```
$ rsync A/* B/
```

Untuk tampilan yang lebih baik, barangkali Anda akan memberikan perintah ini:

```
$ rsync --verbose --progress
A/* B/
file1
 14 100%  0.00kB/s  0:00:00
file2
  6 100%  0.00kB/s  0:00:00
file3
 17 100%  0.00kB/s  0:00:00
```

```
file4
 18 100%  0.00kB/s  0:00:00
file5
  2 100%  0.00kB/s  0:00:00
file6
  8 100%  0.00kB/s  0:00:00

wrote 425 bytes  read 140 bytes
1130.00 bytes/sec
total size is 65  speedup is
0.12
```

Umumnya, Anda juga akan memberikan opsi `-recursive` untuk sinkronisasi rekursif, sehingga, perintah akan menjadi:

```
$ rsync --recursive --verbose --
progress A/* B/
```

Apabila Anda ingin mempertahankan atribut file (hak akses, waktu, group, pemilik), sekaligus meng-copy-kan secara rekursif, termasuk mengopikan symlink dan device, Anda dapat mempergunakan opsi `-a`. Tambahkan opsi `-z`, maka file Anda akan sekaligus dikompres. Opsi `-verbose` bisa dipersingkat menjadi `-v`. Sayang sekali, tidak terdapat versi pendek dari opsi `-progress`. Berikut ini adalah contohnya:

```
rsync --progress -avz A/* B/
```

Satu hal yang harus diperhatikan adalah opsi kompres. Harus diperhatikan bahwa kompresi akan memakan CPU dan memory. Secara default, rsync telah menghindari untuk kompresi ulang file yang susah dikompres seperti file hasil kompresi. Namun, perhatikan juga ukuran file Anda. Apabila contoh file seperti pada contoh, yang memiliki ukuran sangat kecil, maka kompresi sebenarnya tidak diperlukan.

Sekarang, kita akan memasuki tahapan selanjutnya, walaupun masih meng-copy-kan file dari lokal ke lokal. Yang pertama adalah masalah keamanan. Ini tidak sungguh-sungguh diperlukan dalam sinkronisasi lokal. Namun, apabila Anda ingin menerapkannya (dalam sinkronisasi lokal remote), Anda bisa mempergunakan opsi `-rsh` seperti pada contoh berikut:

```
$ rsync --progress -avz --
rsh=/usr/bin/ssh A/* B/
```

Berikut ini adalah beberapa catatan penggunaan:

- Kita sebenarnya bisa saja menambahkan opsi `-W` untuk peng-copy-an file secara keseluruhan, tanpa pemeriksaan *incremental*. Ini tidak disarankan.
- Anda bisa mempergunakan opsi `-n` untuk dry run rsync. Dengan opsi ini, file-file yang akan ditransfer akan diperlihatkan. File tidak sungguh-sungguh ditransfer. Sebagai contoh:

```
$ rsync --progress -avz -n
A/* B/
building file list ...
6 files to consider
file1
file2
file3
file4
file5
file6

wrote 154 bytes read 44 bytes
396.00 bytes/sec
total size is 65 speedup is
0.33
```

- Untuk operasi lebih lanjut, Anda mungkin akan memberikan opsi `-delete` untuk menghapus dari server remote yang juga telah dihapus secara lokal (menghapus apa yang tidak ada di pengirim).
- Masih berhubungan dengan penghapusan, untuk amannya, kita mungkin akan memberikan opsi `-delete-after` untuk menghapus file setelah transfer, bukan sebelum. Kita juga akan memberikan opsi `-ignore-errors` untuk tetap menghapus walaupun terjadi IO error. Berikan juga opsi `-max-delete=NUM` untuk membatasi jumlah file yang dihapus.
- Satu opsi yang berguna ketika diaplikasikan antar host adalah opsi `-timeout`. Dengan penambahan opsi ini, kita bisa mengatur berapa lama timeout dalam satuan detik. Timeout 10 menit cukup masuk akal.
- Pemberian opsi `-c` akan membuat checksum selalu dilakukan. Cukup disarankan.
- Pemberian opsi `-R` akan membuat rsync mempergunakan relative path name.
- Bagi Anda yang paranoid, Anda mungkin akan membuat backup dengan opsi `-backup`.
- Berhubungan dengan symlink, disa-

rankan Anda memberikan opsi `-l` atau `-a` (yang mana di dalamnya telah mengandung opsi `-l`), sehingga symlink akan dikopikan sebagai symlink. Pemberian opsi `-L` akan menyebabkan peng-copy-an referent symlink.

- Opsi `-a` adalah sama dengan opsi `-rlptgoD` (rekursif, link, mempertahankan hak akses, mempertahankan waktu, mempertahankan pemilik dan group serta Device).
- Untuk mempertahankan ACL, Anda akan memberikan opsi `-A` (telah termasuk `-p`).
- Beberapa filesystem memiliki fitur yang tidak ada di filesystem lain. Oleh karena itu, peng-copy-an lintas filesystem mungkin akan menyebabkan masalah tersendiri. Berikan opsi `-x` untuk pengopian untuk satu filesystem.
- Pemberian opsi `-existing` akan menyebabkan rsync hanya akan meng-update file yang telah ada sebelumnya.
- Pemberian opsi `-force` akan menyebabkan penghapusan direktori walaupun tidak kosong.
- Opsi `-partial` akan menyebabkan rsync tetap mempertahankan file yang baru saja di-copy sebagian. Beberapa pengguna tidak menyarankan fitur ini untuk integrasi data.
- Pemberian opsi `-stat` akan memperlihatkan beberapa statistik. Sebagai contoh:

```
$ rsync --progress -avz --stats
A/* B/
building file list ...
6 files to consider

Number of files: 6

Number of files transferred: 0
Total file size: 65 bytes
Total transferred file size:
0 bytes
Literal data: 0 bytes
Matched data: 0 bytes
File list size: 118
Total bytes written: 130
Total bytes read: 20

wrote 130 bytes read 20 bytes
300.00 bytes/sec
total size is 65 speedup is
0.43
```

- Dalam memanfaatkan bandwidth secara bijak, apalagi untuk sinkronisasi yang tidak memiliki urgensi tinggi, Anda mungkin akan mempergunakan opsi `-bwlimit=KBPS`. Bandwidth diukur dalam satuan Kilo Byte (bukan bit).
- Pada pembahasan selanjutnya, kita akan akan membangun rsync server, di mana rsync dijalankan dengan memberikan perintah `-daemon`.

Mengatur rsync server

Untuk menjalankan rsync sebagai server, kita perlu melakukan dua langkah:

- mengedit file `/etc/rsyncd.conf` dan mungkin `/etc/rsyncd.secret`.
- Menjalankan rsync dengan opsi `-daemon`.

Kita akan melakukan langkah pertama terlebih dahulu. Isi dari `rsyncd.conf` dalam beberapa hal akan mirip dengan `smb.conf` (dibuat oleh orang yang sama) sehingga memiliki konfigurasi global dan konfigurasi lokal per “share”. Berikut ini adalah contoh `rsyncd.conf` yang sederhana:

```
read only = true

[MP3]
    path = /srv/mp3

    comment = Beberapa MP3
```

Sebenarnya, kita tidak harus memiliki bagian global seperti opsi `read only = true`. Namun, ini disajikan sebagai contoh file konfigurasi sederhana yang terdiri dari bagian global dan lokal. Sementara, kita membuat semacam “share” atau “nick name” dengan nama MP3, yang akan merujuk ke path `/srv/mp3` (yang berisikan beberapa lagu MP3). Kita juga memberikan komentar dengan opsi `comment`.

Selanjutnya, kita akan melanjutkan ke langkah kedua, yaitu menjalankan rsync dengan memberikan opsi `-daemon`. Berikut ini adalah contohnya:

```
# rsync -daemon
```

Selanjutnya, kita dapat melakukan sinkronisasi dengan contoh berikut ini. Dalam contoh ini, password akan diminta sesuai user sistem yang aktif.


```
$ rsync --progress -avz
192.168.0.50:/srv/mp3 M
Password:
receiving file list ...
12 files to consider
mp3/
mp3/01 - Let's Hang On.mp3
    4503092 100%    1.93MB/s
0:00:02
mp3/01 Pejantan Tangguh.mp3
    3321480 100%    1.92MB/s
0:00:01
mp3/02 Itu Aku...mp3
    4467106 100%    1.73MB/s
0:00:02
mp3/03 Pemuda Rahasia.mp3
    3706420 100%    1.80MB/s
0:00:01
mp3/04 Pilihlah Aku.mp3
    4218420 100%    1.65MB/s
0:00:02
mp3/Backstreet Boys - It's True.
mp3
    5070848 100%    1.87MB/s
0:00:02
```

```
mp3/Cat Stevens - Morning Has
Broken.mp3
    3142763 100%    1.81MB/s
0:00:01
mp3/Charlene - I've Never Been
To Me.mp3
    9509744 100%    1.85MB/s
0:00:04
mp3/Chicago - You're
Inspiration.mp3
    3645568 100%    1.92MB/s
0:00:01
mp3/Clief Richard - The Best Of
Me.mp3
    10087572 100%    1.74MB/s
0:00:05
mp3/David Tao - Ai Hen Jian Dan.
mp3
    3342336 100%    1.59MB/s
0:00:02

wrote 236 bytes
read 53924526 bytes
1711897.21 bytes/sec
```

```
total size is 55015349
speedup is 1.02
```

Beberapa distro seperti SUSE telah menyediakan rsync untuk dijalankan sebagai proses xinetd. Beberapa *initialization script* juga disediakan sehingga Anda dapat memberikan perintah seperti berikut ini:

```
/etc/init.d/rsyncd start
```

Merujuk kepada delapan penggunaan rsync yang dibahas pada awal tulisan, Anda sebenarnya tidak harus selalu menjalankan rsync server sebagai daemon untuk sekadar melakukan sinkronisasi file.

Berikutnya, kita akan membuat file rsyncd.conf menjadi lebih kompleks dengan penambahan beberapa opsi, sekaligus menggunakan rsyncd.secret. Berikut ini adalah contohnya:

```
uid = nobody
gid = nobody
read only = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
```

Colocation dengan 3 Internet Backbone (USA, EUROPA & HONGKONG)

harga mulai dari Rp. 1.250.000,-



Anda mendapatkan :

- 128 Kbps Bandwidth Internasional
- Unlimited 100 Mbps IIX Bandwidth
- 8 Public IP Address
- Technical Support 24 x 7 x 365
- 99.9% Uptime Guarantee
- UPS Backup

Email : sales@saranadata.com
Telp : +62 (21) 526 9257

Business Services :

- Colocation Server
- Webhosting Colocation
- Dedicated Server
- Rack Solution
- Internet Service Solution
- Corp. Application Hosting
- System Mirroring & DRC
- Data Storage Services
- Custom Made Solution

Personal Services :

- Download Services
- E- File Storage
- E- Photo Album

```
motd file = /etc/rsyncd.motd
```

```
[MP3]
```

```
path = /srv/mp3
```

```
comment = Beberapa MP3
```

```
auth users = userku
```

```
secrets file = /etc/rsyncd.  
secrets
```

Berikut ini adalah isi dari rsyncd.secret:

```
# user:passwd
```

```
userku:passwordku
```

User yang tersimpan tidak harus selalu user yang ada di dalam sistem. Perhatikan bahwa file ini seharusnya hanya bisa diakses oleh root.

Kita juga mempergunakan opsi motd file, yang akan menampilkan message of the day. Berikut ini adalah isi dari rsyncd.motd:

```
SELAMAT DATANG !!!
```

```
=====
```

```
ini adalah ucapan selamat datang
```

```
ini adalah ucapan selamat datang
```

```
ini adalah ucapan selamat datang
```

```
Selamat mempergunakan rsync
```

```
server saya :)
```

```
=====
```

Selain cara koneksi seperti sebelumnya, rsync --progress -avz 192.168.0.50:/srv/mp3 M, Anda juga bisa memberikan perintah berikut ini (khusus rsync, menggunakan rsync://). Dengan demikian, Anda bisa sekaligus menentukan user mana saja yang mencoba melakukan koneksi. Dengan cara ini, Anda juga bisa sekaligus mempergunakan “share” atau “nick” yang ditentukan dalam rsyncd.conf. Harap perhatikan tanda :: yang dipergunakan untuk memisahkan host dan “share”.

```
$ rsync --progress -avz rsync://  
userku@192.168.0.50:873::MP3 M  
SELAMAT DATANG !!!
```

```
=====
```

```
ini adalah ucapan selamat datang
```

```
ini adalah ucapan selamat datang
```

```
ini adalah ucapan selamat datang
```

```
Selamat mempergunakan rsync
```

```
server saya :)
```

```
=====
```

```
Password:
```

```
receiving file list ...
```

```
12 files to consider
```

```
./
```

```
03 Pemuda Rahasia.mp3
```

```
3706420 100% 2.41MB/s
```

```
0:00:01
```

```
04 Pilihlah Aku.mp3
```

```
4218420 100% 2.44MB/s
```

```
0:00:01
```

```
Backstreet Boys - It's True.mp3
```

```
5070848 100% 1.92MB/s
```

```
0:00:02
```

```
Cat Stevens - Morning Has
```

```
Broken.mp3
```

```
3142763 100% 2.50MB/s
```

```
0:00:01
```

```
Charlene - I've Never Been To
```

```
Me.mp3
```

```
9509744 100% 2.53MB/s
```

```
0:00:03
```

```
Chicago - You're Inspiration.mp3
```

```
3645568 100% 1.85MB/s
```

```
0:00:01
```

```
Clief Richard - The Best Of
```

```
Me.mp3
```

```
10087572 100% 2.65MB/s
```

```
0:00:03
```

```
David Tao - Ai Hen Jian Dan.mp3
```

```
3342336 100% 1.50MB/s
```

```
0:00:02
```

```
wrote 254 bytes
```

```
read 41824820 bytes
```

```
1858892.18 bytes/sec
```

```
total size is 55015349
```

```
speedup is 1.32
```

Berikut ini adalah beberapa opsi global yang mungkin berguna:

- **syslog facility.** Mengizinkan Anda untuk menentukan syslog facility yang dipakai. Anda bisa mempergunakan standar nama yang dipakai di sistem. Nilai default-nya adalah daemon. Umumnya, opsi-opsi berikut tersedia: auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, security, syslog, user, uucp, local0, local1, local2, local3, local4, local5, local6, dan local7.
- **log format.** Format log yang dituliskan dalam log file. Ini sangat penting ketika Anda ingin mengimplementasikan parser sendiri untuk log Anda. Umumnya, pengguna akan mempergunakan format standar dan mempergunakan log visualization tool yang umum.

- **log file.** Nama log file yang digunakan.
- **motd file.** Menentukan message of the day.

Selanjutnya, berikut ini adalah opsi yang dipergunakan dalam “share” atau “nick”:

- **comment.** Menspesifikasikan komentar.
- **path.** Path yang digunakan.
- **use chroot.** Apabila diset, maka chroot akan dilakukan pada path. Ini akan memberikan proteksi ekstra, namun sekaligus memiliki kekurangan seperti membutuhkan hak akses root, tidak mampu mengambil symlink yang mengacu diluar root baru. Default dari opsi ini adalah true. Gunakan ini untuk file-sistem yang sensitif terhadap masalah keamanan.
- **read only.** Menentukan hak akses *read only*. Berikan hak yang bersesuaian agar update menjadi lebih mudah.
- **uid.** Menentukan user yang digunakan pada saat rsync dijalankan sebagai root.
- **gid.** Menentukan group yang digunakan pada saat rsync dijalankan sebagai root.
- **auth users.** Menentukan user yang valid (tidak harus merupakan user yang aktif di sistem).
- **secrets file.** Digunakan bersama *auth users* untuk menyimpan *username* dan *password*.
- **hosts allow.** Menentukan host yang diizinkan.
- **hosts deny.** Menentukan host yang tidak diizinkan.
- **dont compress.** Menentukan file mana yang tidak perlu dikompres (karena tidak akan memberikan pengurangan ukuran yang berarti dan malah memberatkan CPU). Default-nya adalah *.gz *.tgz *.zip *.z *.rpm *.deb *.iso *.bz2 *.tbz. Anda mungkin ingin menambahkan MP3 dan OGG misalnya.

Setelah rsync server selesai dibuat, Anda bisa mempergunakan juga opsi-opsi yang telah dibahas pada bagian-bagian awal tulisan ini.

Dalam pembahasan ini, kita telah melihat betapa mudah dan serunya membangun server rsync. Rsync adalah teknologi yang sangat berguna dalam melakukan mirroring. Selamat mencoba! 🐱

Noprianto (noprianto@infolinux.co.id)