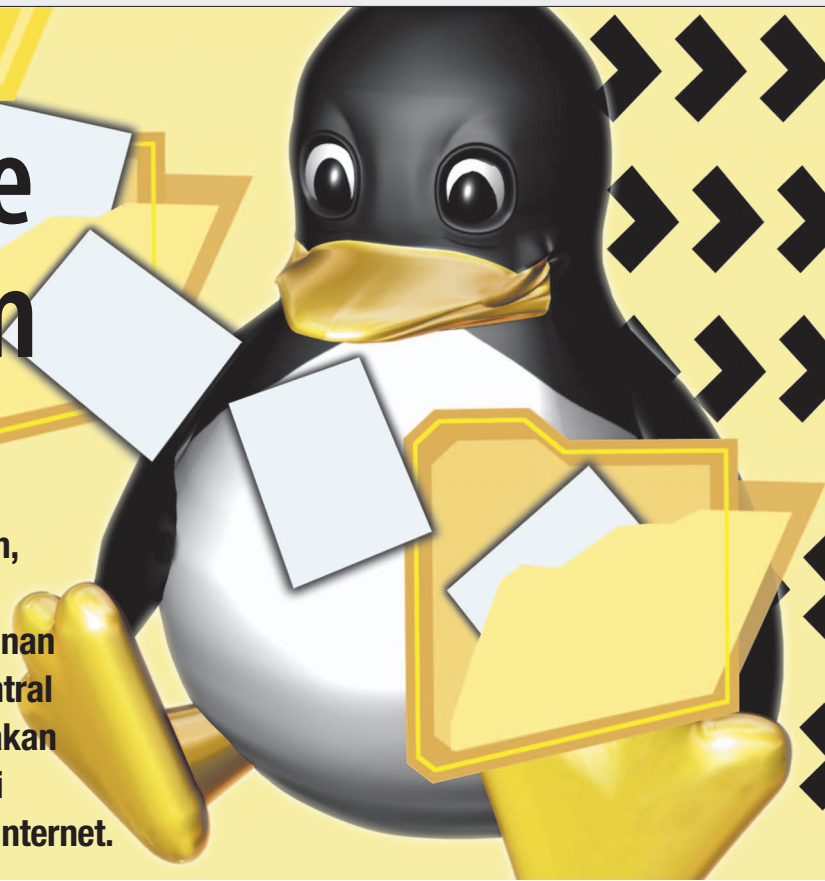


# Mengakses File Lewat Jaringan

Noprianto

**Dengan terhubungnya komputer ke jaringan, pengguna kemudian dapat mengakses file yang tersimpan di komputer lain. Penyimpanan dokumen juga bisa dilakukan secara terpusat berkat file server. Di “Utama” kali ini, kita akan membahas berbagai cara mengakses file di jaringan, a la Windows, UNIX, maupun a la Internet.**



**K**etika nilai fungsi komputer bertambah, bertambah pula kompleksitas yang harus kita hadapi sebagai pemilik atau pengelola komputer. Dulu, karena komputer hanya bisa *stand-alone* dan kemampuan pemrosesan masih rendah, kita tidak akan peduli dengan masalah-masalah yang melibatkan jaringan, seperti virus dan lain sebagainya. Sama juga ketika ukuran data dan penyimpanan masih terbatas. Dengan sebuah harddisk, rasanya semua data sudah dapat disimpan di sebuah floppy berukuran beberapa ratus kilo.

Namun saat ini, semua telah berubah. Komputer menjadi sedemikian canggih sehingga mampu melakukan apa saja. Penyimpanan dan data menjadi begitu mudah bertambah sehingga harddisk 100 GB juga rasa-rasanya masih kurang. Teknologi jaringan menjadi begitu canggih sehingga banyak kabel bisa difungsikan menjadi tali jemuran.

Dengan semua perkembangan tersebut beserta makin tingginya kompleksitas, akan muncul permasalahan baru. Contoh yang paling mudah adalah inefisiensi. Kita bisa saja menyimpan segalanya dan ketika menemukan harddisk telah penuh, kita akan

membeli yang baru. *Toh*, harganya juga cukup terjangkau. Atau, begitu menemukan komputer sudah terasa lambat, kita bisa saja membeli yang baru. Tanpa kita sadari, banyak sekali hal yang terbuang sia-sia.

Ada kata-kata bijak yang pernah penulis ingat. Manfaatkanlah sesuatu benar-benar. Salah satu nilai yang terkandung dalam kata-kata tersebut mungkin juga bermaksud untuk mengajak kita menjaga apa yang kita miliki baik-baik.

Salah satu contoh yang dekat mungkin masalah penyimpanan file. Ini bukanlah masalah sepele. Penulis setidaknya menghabiskan sekitar setengah tahun untuk mempelajari hal yang satu ini. Banyak sekali ditemukan file-file yang berganda, yang pada akhirnya, pengguna komputer tersebut akan mengajukan anggaran untuk pembelian harddisk baru. Ada juga karena penyimpanan file yang berantakan, begitu ingin digunakan, file tersebut sangat susah untuk ditemukan. Lebih parah lagi, ada file yang digunakan bersama, disimpan oleh masing-masing pengguna dengan cara masing-masing, revisi dilakukan oleh masing-masing, *merger* juga dilakukan sekenanya, dan setelah itu, kalang kabut kalau ada informasi yang hilang.

Dengan prinsip bahwa kita perlu mengelola apa yang kita miliki sebaik mungkin, masalah penyimpanan file sebenarnya bisa kita siasati dengan apa yang selama ini telah tersedia. Di tulisan ini, kita akan membahas beberapa ide penyimpanan file di jaringan, kemudian bagaimana pengaturan di sisi user, dan, tak lupa, kita juga membahas bagaimana cara teknis untuk berbagi file. Menurut cara Windows-kah, atau menurut cara Linux? Atau, malah tidak dua-duanya, namun, menurut standar Internet? Bagaimana pula kita harus mengatur file sharing di lingkungan sistem operasi yang tidak homogen?

Kita akan mencoba membahas dari berbagai sisi, dan akan sedikit membahas hal teknis. Semoga tulisan ini bisa sedikit membantu. Kita kelola aset kita baik-baik dan manfaatkan sebaik mungkin.

## TEKNOLOGI

Pertama-tama, kita akan melihat beberapa cara berbagi file yang dikenal umum dalam berbagai *platform*. Pemilihan teknologi yang cocok dengan kebutuhan user terkadang tidak mudah. Ada kalanya, kita menemukan bahwa cara berbagi file a la Windows akan

lebih baik jika diterapkan pada jaringan tertentu, walaupun sebagian komputer di jaringan tersebut terinstal Linux. Atau, terkadang, pendekatan ala sistem operasi rupanya tidak menunjang karena biaya yang dibutuhkan untuk penyelesaian kebutuhan tidak sebanding dengan fungsinya. Oleh karena itu, solusi yang lebih murah harus dicari.

Kita akan membahas secara umum apa saja yang umumnya digunakan untuk berbagi file dalam berbagai platform dan kebutuhan apa saja yang perlu kita penuhi agar dapat berbagi file sesuai protokol tertentu. Kita juga akan membahas beberapa kekurangan dan kelebihan suatu cara berbagai file—lebih pada sisi manajemen. Dan, tak lupa, beberapa sisi teknis juga kita bahas untuk bahas pertimbangan.

## Berbagi file ala Windows

Pada saat tulisan ini dibuat, Microsoft masih menguasai sekitar 60% pangsa pasar server dan sekitar 95% pangsa pasar *desktop*. Walaupun Linux makin canggih dan makin banyak dilirik, Microsoft tidak akan begitu saja melepaskan apa yang telah dimilikinya. Dengan pasar yang dikuasai oleh Microsoft, tak heran apabila di mana-mana kita akan menemukan komputer yang terinstal Windows.

Harus diakui, terutama pada Windows 2000 ke atas, file sharing merupakan hal yang trivial. Pengguna cukup melakukan *browsing* ke *entire network* melalui Windows explorer, dan apabila ada komputer yang ditemukan, pengguna dapat langsung melakukan klik ganda, mengisi password dan memasuki *share* pada komputer tersebut.

Apabila ingin lebih kompleks, bisa mengakses map network drive dari menu Tools dan setelah mengisi share dan informasi user, Anda bisa mengakses share jaringan sebagai sebuah drive baru.

Mau lebih terotomatisasi? Gunakan saja batch file atau kombinasi program net di Windows dan mapping pun dapat dilakukan dari *command line*. Kombinasikan dengan *login script*, dan semua pun beres. Urusan sharing file ala Windows pun selesai. Dengan pengaturan tertentu, user dapat saling berbagi file dengan cara yang mudah. Tak heran apabila Microsoft menguasai

pangsa pasar desktop. Semuanya begitu sederhana.

## Protokol SMB

Mari kita masuk ke masa puluhan tahun lalu, ketika PC masih bayi. Pada masa-masa awal ini, IBM dan perusahaan Sytec bersama-sama mengembangkan protokol untuk membangun LAN kecil-kecilan. Pengembangan tersebut salah satunya kemudian menghasilkan NetBIOS, yang merupakan singkatan dari *Network Basic Input Output System*. NetBIOS adalah program yang di load ke dalam memory dan menjadi penengah antara program dan hardware network. NetBIOS menggunakan *addressing scheme* berupa nama sepanjang 16 byte untuk mengidentifikasi *workstation* dan aplikasi jaringan.

Microsoft di satu sisi, kemudian menambahkan feature ke DOS yang memungkinkan disk I/O diarahkan ke interface NetBIOS sehingga memungkinkan disk space di-share dalam LAN. Penemuan ini luar biasa dan menjadikan Microsoft salah satu penemu kunci dalam teknologi file sharing. Beberapa saat setelah itu, protokol tersebut kemudian dikenal sebagai protokol SMB.

Kerjasama antara TCP/IP (untuk jaringan besar) dan NetBIOS (untuk LAN) dengan memetakan pengenal 16 byte yang digunakan oleh NetBIOS menjadi alamat IP memungkinkan suatu pesan, melalui

routing tertentu, dapat sampai ke host lain dalam jaringan besar. Protokol SMB kemudian terus dikembangkan oleh Microsoft, sehingga pada akhirnya, SMB memiliki fasilitas *browsing* dan *authentication* (Windows NT Domain control).

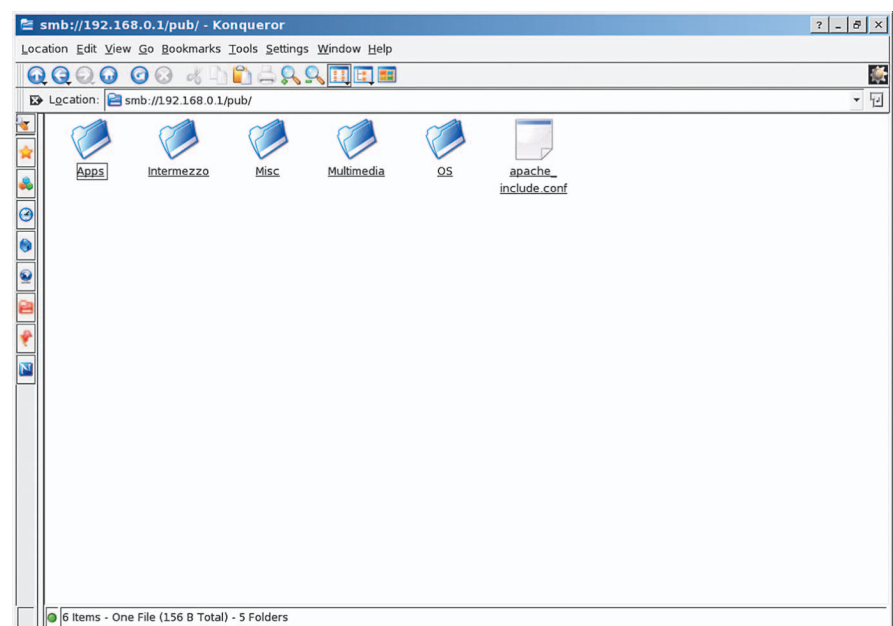
Saat ini, suite protokol tersebut dikenal sebagai CIFS (*Common Internet File System*), yang merupakan inkarnasi terakhir (untuk saat ini) dari protokol SMB. Nama yang bagus ini sekaligus mencerminkan visi Microsoft akan file sistem di masa depan.

## Kebutuhan

Karena kita bekerja dengan mesin Linux, maka diperlukan seperangkat tool agar dapat bekerja dengan atau bekerja sebagai jaringan Windows. Semua hal tersebut dimungkinkan dengan adanya paket SAMBA.

Hacker dari Australia, Andrew Tridgell, pada suatu waktu mengalami masalah untuk melakukan *mounting* server UNIX ke dalam PC DOS miliknya. Hal ini kemudian dapat diselesaikan dengan client NFS (akan dibahas kemudian pada cara berbagi file ala UNIX) untuk DOS. Sayangnya, Andrew memiliki aplikasi yang membutuhkan NetBIOS dan berbagai hal menjadi tidak sederhana lagi. Ada dua protokol yang harus digunakan, dan di DOS, hal ini menjadi masalah besar.

Andrew kemudian menulis sebuah *packet sniffer* dan melakukan *reverse engineering*



Konqueror dan SMB.

protokol SMB (luar biasa!) dan kemudian mengimplementasikan protokol ini di server berbasis UNIX. Andrew mempublikasikan *source code* produknya pada tahun 1992 dan segera mendapatkan banyak sambutan. Setelah hampir 13 tahun dari kode tersebut ditulis, SAMBA menjadi produk yang sangat matang. Beberapa waktu yang lalu, SAMBA bahkan menjadi topik pembicaraan dalam Halloween Document Microsoft, yang merupakan perhatian Microsoft untuk ancaman dari dunia open source.

Dengan memanfaatkan SAMBA saja, server Linux Anda dapat berfungsi sebagai server SMB. Client Windows dapat mengakses server Anda layaknya mengakses server Windows. Anda dapat berbagi file ataupun printer, berfungsi sebagai PDC dan lain sebagainya.

SAMBA kemudian dapat digabungkan dengan berbagai fasilitas lain seperti LDAP untuk *backend* otentikasi.

Beberapa waktu yang lalu, satu proyek dengan nama Samba TNG lahir dengan tujuan integrasi LDAP yang sangat tinggi. Walau, sampai tulisan ini dibuat, rilis Samba TNG baru mencapai versi 0.4 dan sudah lebih dari satu tahun tidak dirilis lagi. Dukungan LDAP untuk samba saat ini pun sudah sangat mencukupi.

Saat ini, hampir setiap distro telah datang dengan paket SAMBA. Anda hampir tidak perlu melakukan kompilasi dari *source code* lagi. Cukup instal saja paket SAMBA (apabila belum terinstal). Berbagai proyek desktop seperti KDE dan GNOME pun telah memiliki fasilitas untuk membrowsing server Samba yang ada di jaringan. KDE bahkan datang dengan I/O Slave SMB untuk dukungan SAMBA.

### Keuntungan

Keuntungan dari menggunakan SAMBA sudah jelas adanya. Protokol SMB sendiri adalah protokol yang sangat matang. Kita tidak perlu lagi meragukan kemampuannya. Dan, SAMBA sendiri juga sudah matang.

Dengan menggunakan SAMBA, terutama di lingkungan kerja nonhomogen (Windows dan Linux digunakan secara merata pada workstation, plus beberapa server Windows), maka sharing ala Windows memiliki keuntungan tersendiri. Kita tidak perlu lagi menambah stack protokol

dengan menggunakan NFS misalnya.

Apabila kita memaksa untuk harus menggunakan NFS, client Linux mungkin akan bergembira ria, sementara, client Windows mungkin harus bermuram durja dan menginstall client NFS untuk Windows. Hal ini akan memacu instalasi dan maintenance tambahan.

Sementara, apabila kita menggunakan protokol SMB, client Linux juga tidak terlalu banyak berkorban, sementara client Windows akan bersorak sorai bergembira. Semua bahagia.

### Kekurangan

Sebenarnya, menggunakan SMB tidaklah memiliki banyak kekurangan. Beberapa gangguan non teknis mungkin akan muncul. Seperti, Linux kok berbicara ala Windows, malu dong (hal ini lucu, namun sangat realistis). Atau, karena NFS lebih Unix-ish, maka SMB tidaklah cocok untuk Linux.

Di luar semua alasan subjektif tersebut, kehadiran SMB di Linux memang tidak sematang NFS. Menggunakan SMB, proyek SAMBA harus selalu mengikuti perkembangan, sementara, dukungan NFS telah mencapai level matang di Unix dan Linux.

### Integrasi lokal

Workstation di Windows dapat melakukan mounting share menjadi sebuah drive khusus. Oleh karena itu, tidak masalah dari sisi client Windows untuk memiliki integrasi lokal suatu service SMB.

Di Linux, integrasi lokal juga dimungkinkan dengan masuknya SMBFS ke dalam kernel. Saat ini, kita juga bisa dapat melakukan mounting dengan tipe file sistem adalah *smbfs*. Dengan demikian, kita tidak perlu lagi harus menggunakan program *smbclient* atau sejenis itu untuk dapat mendownload atau mengupload file. Dengan kemampuan mounting SMB sebagai suatu direktori, user dapat bekerja pada direktori mount seolah-olah di direktori lokal.

Artinya, saat ini, sangat menunggingkan bagi client Linux untuk bekerja dengan nyaman ala Windows, dari sisi integrasi lokal.

### Keamanan

Dari sisi keamanan, rasa-rasanya tidak ada masalah berarti pada protokol SMB atau

paket SAMBA itu sendiri. Protokol SMB dicermati oleh banyak pihak termasuk Microsoft dan IBM, sementara SAMBA memiliki kontributor dari berbagai kalangan. Semuanya mengamati dan memperbaiki kelemahan apabila ditemukan.

Beberapa waktu yang lalu, memang terdapat kelemahan pada implementasi SAMBA, namun, seperti halnya proyek open source yang terus berkembang, kini SAMBA sudah cukup matang untuk diterapkan.

### Lebih detail

SAMBA datang dengan dua program kunci dengan berbagai tool tambahan. Kedua program kunci tersebut adalah *smbd* dan *nmbd*. Kedua program tersebut mengemban amanat untuk mengimplementasikan keempat service dasar CIFS, yaitu:

- Service file dan printer.
- Authentication dan authentication.
- Name resolution.
- Service announcement (browsing).

Servis yang pertama adalah kunci dari layanan CIFS. Fasilitas ini dimungkinkan dengan kerja keras dari *smbd*. Dalam mengakses layanan file dan printer ini, user dapat dibatasi dengan adanya otentikasi. Dengan demikian, hanya user yang terdaftar dan mampu melewati proses otentikasi yang bisa mengakses layanan tersebut. Namun, ada juga layanan yang bisa diakses oleh siapa saja, tanpa proses otentikasi dan otorisasi. Service ini juga ditangani oleh program *smbd*.

Bicara soal otentikasi dan otorisasi, Windows NT Domain system menyediakan cara yang hebat untuk menangani layanan yang bisa diakses oleh user. User hanya perlu login sekali untuk mengakses berbagai service yang tersedia untuk user tersebut. Hal ini ditangani oleh Domain Controller. Satu domain controller akan menangani satu NT Domain (beberapa komputer). SAMBA kini sudah dapat berfungsi sebagai Primary Domain Controller ataupun Backup. Backup otentikasi juga bisa beragam, mulai dari file *smbpasswd* sampai LDAP.

Kedua service CIFS terakhir, name resolution dan browsing, ditangani oleh *nmbd*. Inti kerja dari *nmbd* adalah manajemen dan distribusi daftar name NetBIOS.

Name resolution dapat bekerja dalam

dua cara: broadcast dan *point-to-point*. Umumnya, suatu mesin akan bekerja dengan dua cara tersebut. Cara pertama akan menyebabkan *traffic* sedikit ramai namun lebih sederhana. Pada LAN, cara ini tidak mengakibatkan masalah.

Cara kedua adalah dengan memanfaatkan NBNS (NetBIOS Name Service) server. Suatu mesin dapat meninggalkan pesan yang berisi nama dan IP untuk dikenali oleh mesin lain. Microsoft menyebut implementasi NBNS milik mereka sebagai WINS (Windows Internet Name Service). NBNS dapat bekerja lintas subnet dan tidak terbatas oleh LAN. Dalam banyak hal, NBNS bekerja seperti halnya DNS (Domain Name System), hanya bedanya, entri-entri dalam NBNS server hampir semua dinamis.

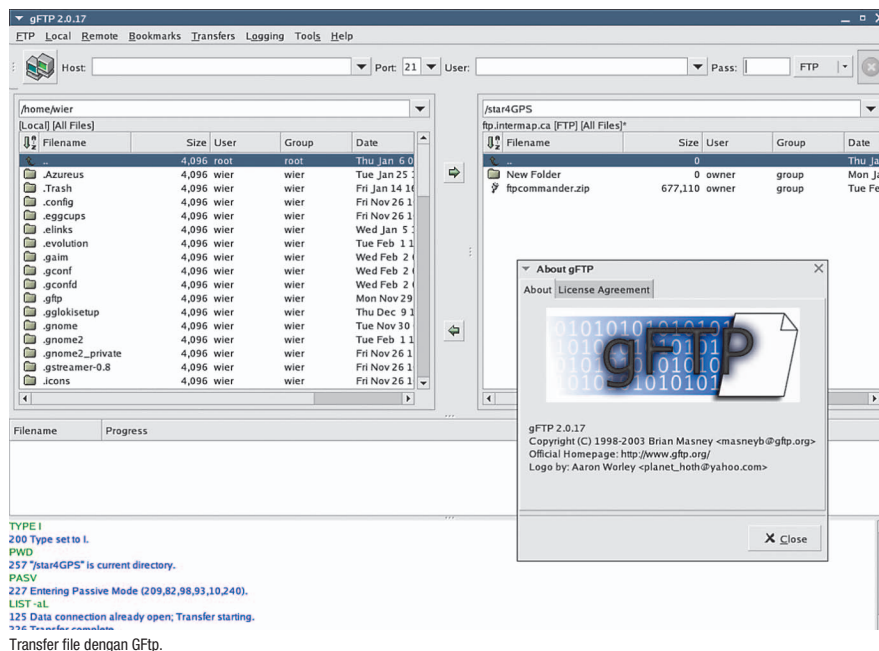
Tugas terakhir CIFS, yaitu *browsing* melibatkan kemampuan untuk menampilkan daftar layanan suatu komputer dalam jaringan. Dalam suatu LAN, komputer-komputer akan bersaing untuk menjadi Local Master Browser. LMB kemudian akan *maintain* daftar service yang tersedia, yang akan tampil ketika Anda membuka Network Neighbourhood di Windows.

Lebih jauh lagi, kita mengenal DMB (*Domain Master Browser*) untuk LMB skala besar, lintas domain. Namun, fasilitas ini hampir tidak dapat bekerja dengan baik karena masalah waktu. Dengan lintas jaringan besar, bisa-bisa butuh waktu dalam hitungan jam untuk memunculkan, mengupdate dan memberikan service yang bisa diakses pada Network Neighbourhood.

## Berbagi file ala UNIX

Bagi Anda yang hidup di dunia UNIX, tentunya akan wajar apabila fasilitas file sharing dilakukan dengan cara UNIX. Saat ini, UNIX dan Linux memegang pasar sekitar 30% persen di dunia server, dan di desktop Linux kini memiliki pasar sebesar 2% sampai 3%. Belakangan ini, banyak pembuat distro telah menargetkan distro mereka untuk melayani kebutuhan desktop. Tak tanggung-tanggung, yang bermain di sini adalah perusahaan sekelas Sun Microsystems dan Novell.

Dengan mulai digunakannya Linux di sisi desktop, terutama untuk kebutuhan kantor, file sharing ala UNIX atau Linux bisa diterapkan dengan lebih mudah. Client



dengan sistem operasi Windows juga sebenarnya dapat berbicara dengan file sharing NFS yang akan kita bahas ini, namun membutuhkan sedikit tambahan usaha *maintenance*.

Berikut ini kita akan membahas sistem file sharing dengan protokol NFS, suatu protokol yang sudah sangat matang, digunakan secara meluas dan masih terus dikembangkan untuk fitur lebih lanjut.

## Protokol NFS

NFS adalah singkatan dari *Network File System*, yang kali pertama dikembangkan oleh Sun Microsystems pada tahun 80-an, sebagai sarana untuk berbagi file pada lingkungan kerja *diskless*. NFS menyediakan sarana untuk berbagi file antarjaringan, sehingga suatu mesin dapat mengakses file-file di mesin lain seolah-olah mengakses file sistem lokal.

Protokol ini bekerja dengan sistem client/server. Server akan meng-*export* suatu file-sistem sehingga dapat di-share di jaringan, selanjutnya, user akan melakukan mounting ke server tersebut. Dengan dukungan NFS yang sudah sangat matang di kernel, berbagai distro kini umumnya telah memasukkan NFS sebagai salah satu filesistem yang akan di-mount pada saat booting.

Dengan demikian, user tidak perlu sama sekali tahu apa yang terjadi. File yang diakses akan terlihat lokal, walaupun tersimpan

di mesin berbeda.

Tercatat ada tiga versi NFS yang pernah, sedang dan akan digunakan. Yang pertama adalah NFS versi 2, NFS versi 3 dan NFS versi 4. Di Linux, ketiga versi NFS ini didukung dengan baik oleh kernel 2.6. Beberapa sistem operasi berbasis UNIX atau beberapa distro Linux dengan kernel 2.4 (atau bahkan 2.6) masih mengandalkan NFS versi 3. Harus diakui, saat ini, NFS versi 3 adalah NFS yang paling matang dan umum. Untuk client dan server yang hanya mendukung salah satu versi, interoperabilitas antara suatu versi dengan versi lainnya tidak akan terjadi.

Pembahasan mengenai versi 2,3 dan 4, termasuk fitur akan dibahas di bagian lebih detail dari NFS.

## Kebutuhan

Apabila Anda menggunakan distro Linux yang cukup baru seperti Red Hat 9/Fedora Core atau SUSE 9, maka dukungan NFS sudah bisa Anda dapatkan dengan sangat mudah. Teorinya, sudah sejak lama sekali, dukungan NFS dimasukkan ke dalam distro Linux.

Apa yang perlu Anda pastikan adalah telah diaktifkannya dukungan NFS di dalam kernel, baik sebagai *built-in* ataupun *module*. Untuk memeriksanya, bukalah file `/proc/filesystems`.

NFS sendiri kini memiliki implemen-

tasi server kernel-based dan user-based. Artinya, proses NFS berjalan sebagai proses kernel (lebih cepat, lebih tidak aman) ataupun berjalan sebagai proses user (lebih lambat, lebih aman).

Setelah itu, Anda akan membutuhkan paket `nfs utilities`. Mounting sendiri bisa dilakukan oleh program `mount`.

Untuk menggunakan NFS di Linux, apa yang Anda harus lakukan boleh dikatakan sangat sedikit. Aktifkan NFS di kernel, install `nfs utilities`, ekspor filesistem, jalankan server dan client tinggal menggunakan program `mount` untuk mengakses share. Sederhana, singkat, dan mudah.

### Keuntungan

NFS sendiri adalah teknologi yang sudah sangat tua dan matang, sekaligus memiliki cukup banyak fitur. Dalam beberapa hal, NFS sedikit berbeda dengan SMB, dan oleh karena itu, kita tidak bisa membandingkan begitu saja antara SMB dan NFS.

Dengan menggunakan NFS, kita dapat mengatur penyimpanan semua file di server dengan cara yang mudah. Kita bisa saja meng-export semua home direktori dengan hak masing-masing, menggunakan LDAP sebagai backend otentikasi dan semua file user dapat di-maintain dengan mudah di satu tempat. Pendekatan ini juga bisa dilakukan dengan protokol SMB, namun jauh lebih sederhana menggunakan NFS untuk client Linux.

### Kekurangan

Implementasi NFS juga memiliki kekurangan. Salah satunya adalah ketika banyak client yang menggunakan Windows (apalagi jika banyak versi, lalu, kita perlu menginstall beraneka ragam paket). Selain itu, kita masih belum bicara soal penyediaan printer sharing yang umumnya diasumsikan oleh pengguna adalah satu paket dengan file sharing.

NFS tidak ideal apabila Anda ingin mencapai segala sesuatu yang mampu dicapai oleh SMB. Namun, apabila Anda ingin mengimplementasikan *centralized* file sharing ala Linux, dengan integrasi LDAP sebagai *backend authentication*, mungkin NFS masih bisa diterima.

### Integrasi lokal

Integrasi lokal NFS bisa dicapai dengan mudah dengan dukungan NFS FS di dalam kernel Linux. User tinggal memanfaatkan program `mount` untuk dapat bekerja dengan NFS.

Dengan memanfaatkan `fstab`, NFS volume dapat dimount pada saat booting, dan selesai melakukan otentikasi, user sudah dapat bekerja dengan file-file yang tersimpan di server NFS tersebut sesuai dengan haknya.

### Keamanan

Di luar semua masalah keamanan sistem, NFS memiliki satu peluang keamanan

sistem ketika berurusan dengan User ID mapping.

Pada saat meng-export suatu filesistem, berbagai atribut juga ikut di-export. Salah satunya adalah mode file, yang mengandung hak akses. Katakanlah Komputer A memiliki sekian banyak user dan meng-export home direktori. User di komputer B mungkin memiliki UID yang sama dengan UID user tertentu di server, padahal nama user-nya tidak sama. Dengan demikian, akan ada user yang tidak berhak mengakses file atau direktori tertentu, namun bisa mengaksesnya. Ini adalah masalah pertama.

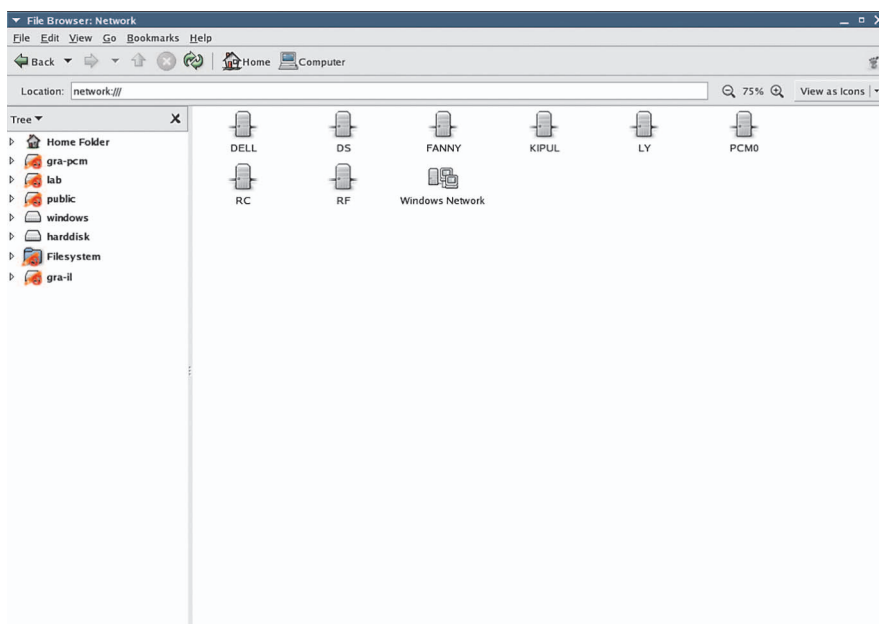
Yang kedua, jangan sampai Anda mengaktifkan pilihan agar root di Komputer B yang tadi dapat memiliki *full access* terhadap filesistem yang di-export dari komputer A. Secara default, opsi ini tidak diaktifkan sehingga user root di komputer B akan dianggap sebagai user yang tidak memiliki hak akses penuh untuk mengakses share.

Solusi untuk masalah pertama sebenarnya sederhana apabila Anda memanfaatkan *centralized authentication* seperti halnya memanfaatkan LDAP sebagai backend autentikasi.

Selain masalah keamanan hak akses, ada pula masalah keamanan yang berhubungan dengan hilangnya data akibat penulisan tidak sempurna. NFS sangat mengandalkan cache dalam bekerja. Penggunaan caching ini mungkin saja membuat penulisan menjadi tidak terlalu responsif. NFS menyediakan fasilitas untuk menonaktifkan caching dan menulis sesegera mungkin. Namun, hal ini bisa beresiko besar sekali.

Selain itu, masih berhubungan dengan caching, NFS menyediakan opsi untuk bekerja secara sinkronous ataupun asinkronous. Dengan mengatur agar NFS bekerja secara sinkronous (`sync`), maka reply akan dilakukan setelah penulisan. Sebaliknya, asinkron, reply dilakukan sebelum penulisan. Bekerja secara asinkron akan meningkatkan performa, namun akan berakibat kehilangan data, apabila server dimatikan. Sebaliknya, sinkronous lebih aman. Modus sinkronous adalah default pada NFS utilities di atas versi 1.0.0.

Satu hal yang harus diperhatikan berkaitan dengan masalah sinkron atau asinkron adalah terjadinya file corrupt yang tidak



Melihat file sharing dengan Nautilus.

terdeteksi. Hal ini bahaya sekali. Jangan sampai Anda menganggap operasi beres padahal isi file sudah berantakan.

### Lebih detail

Seperti telah disebutkan sebelumnya, kita mengenal tiga versi NFS, yaitu 2, 3 dan 4. NFS 4 adalah NFS yang termasuk baru dan implementasinya tidak semasal NFS 3.

Antara NFS 2 dan 3, kita mengenal beberapa perbedaan sebagai berikut:

- Client versi 2 hanya bisa mengakses file 2 GB (32 bit), sementara, client versi 3 mampu mengakses sampai 64 bit. Tentu saja, masalah dukungan file besar akan bergantung pula kepada sistem operasi di mana NFS server berada.
- NFS 2 membatasi maksimal operasi NFS sebesar 8 KB sementara NFS 3 UDP secara teori mampu sampai 56 KB. NFS 3 TCP akan bergantung kepada implementasi (umumnya tak lebih dari 32 KB).
- NFS 3 memperkenalkan konsep Weak Cache Consistency, yang akan membantu client NFS 3 untuk lebih cepat mendeteksi perubahan file.
- NFS 2 mengharuskan *reply* setelah penulisan.
- NFS 3 juga memperkenalkan adanya *safe asynchronous write*.
- NFS 2 memiliki skema terbatas untuk masalah hak akses, versi 3 lebih fleksibel.

NFS 4 datang dengan fitur-fitur sebagai berikut:

- NFS 2 dan 3 adalah protokol stateless. NFS 4 memperkenalkan state. Client NFS 4 mempergunakan state untuk bekerja dengan *locking, reading, writing*, dan lain sebagainya.
- Locking pada NFS 4 adalah lease-based locking.
- NFS 4 memperkenalkan delegasi file. Server kini mengizinkan client untuk melakukan perubahan pada file dalam cache sendiri tanpa mengirimkan permintaan ke server.
- NFS 4 mengizinkan adanya *single remote procedure call* untuk menggabungkan beberapa operasi NFS. Hal ini bertujuan untuk mengurangi kompleksitas operasi jaringan.
- Keamanan yang lebih baik.

- NFS 4 menstandarisasi penggunaan dan interpretasi ACL Posix dan Windows.
- NFS 4 menggabungkan protokol yang selama ini terpisah (stat, NLM, mount, ACL dan NFS) menjadi satu spesifikasi protokol untuk mengizinkan kompatibilitas yang lebih baik dengan firewall.
- NFS 4 memperkenalkan protokol untuk migrasi dan replikasi file.
- NFS 4 membutuhkan dukungan RPC melewati transport protocol seperti TCP yang lebih reliabel.

Beberapa keluhan pada penggunaan NFS adalah lambat. Sebenarnya, ada beberapa hal yang bisa dilakukan untuk meningkatkan kecepatan:

- Mempercepat disk I/O pada server (harddisk yang lebih cepat dan sistem yang lebih powerful) akan berpengaruh besar pada NFS 2 dan 3.
- Terkadang, akses terasa lambat karena aplikasi membuka file dengan modus *O\_SYNC*, dan hal ini akan membuat NFS 3 bekerja sinkron seperti NFS 2 walaupun file tidak berukuran besar dan NFS server sangat reliabel.
- Besarkan ukuran socket buffer apabila Anda menemukan fragmentation error pada keluaran perintah netstat.
- Perbanyak *thread* (pastikan server Anda mampu).
- Gunakan perangkat jaringan yang cukup cepat.
- Apabila diinginkan, Anda bisa memisahkan jurnal pada disk yang berbeda ketika Anda menggunakan *reiserfs* ataupun *ext3*.

### Berbagi file a la Internet

Baiklah. Anda mungkin sudah sangat bosan dengan protokol yang tergantung pada salah satu sistem operasi tertentu. Atau, lingkungan kerja Anda sangat beragam dan akan sangat merepotkan apabila Anda harus mengatur agar beberapa komputer client harus mengalah sehingga keseluruhan sistem file sharing Anda condong kepada *platform* tertentu, dan lain sebagainya. Selain itu, rupanya ada beberapa client Anda yang hidup di tepian teknologi sehingga bekerja mobile dengan PDA. Tentu saja, protokol yang senetral mungkin harus diterapkan pada lingkungan kerja Anda ka-

lau begitu.

Di Internet, kita mengenal protokol HTTP yang telah lama bercokol dan sepertinya sangat bisa diandalkan. Dengan HTTP, Anda bisa berbagi file dengan mudah. Untuk *upload*, cara lama yang cukup favorit adalah mempergunakan FTP, walaupun, saat ini, kita bisa banyak memanfaatkan WebDAV.

Mempergunakan HTTP dan WebDAV memang menarik, walaupun, masih memiliki beberapa kelemahan dibandingkan dengan cara biasa. Kita harus tahu benar apa kebutuhan kita sebelum yakin memilih suatu teknologi. Namun, secara umum, cara berbagi file a la Internet saat ini sudah mulai dilirik dan diterapkan.

### Protokol HTTP (+WebDAV) dan FTP

Dengan HTTP server seperti Apache, kita dapat membangun suatu website. HTTP telah dikembangkan sangat lama dan merupakan protokol yang matang. Saat ini, fungsi dari sebuah HTTP server bukanlah sekedar menghadirkan konten web, melainkan juga mampu menyediakan fasilitas file sharing.

Dengan beberapa pengaturan, Apache juga mampu menghadirkan file sharing yang terproteksi dengan baik. User melakukan login dan apabila sukses, resource yang menjadi haknya akan ditampilkan.

Untuk meng-upload file ke server, walaupun protokol HTTP juga mampu melakukannya, umumnya FTP-lah yang lebih umum dilakukan. Sekelompok file langsung dapat di-upload ke server dan dapat diambil oleh user lain dalam waktu yang sama.

Namun, meng-upload file seperti ini bukanlah cara yang paling efisien. Protokol HTTP kemudian dikembangkan sehingga mampu menghadirkan apa yang saat ini kita kenal dengan nama WebDAV. WebDAV adalah singkatan *Web based Distributed Authoring and Versioning*. Tujuan dasar dari WebDAV adalah menyediakan fasilitas I/O melalui protokol HTTP. Dengan demikian, selain dapat men-download file, user juga bisa meng-upload dan mengatur file.

### Kebutuhan

Untuk dapat bekerja dengan HTTP, Anda hanya membutuhkan Apache dan modul

## Mengakses File Lewat Jaringan

Apache yang sesuai dengan kebutuhan Anda. Umumnya, hampir semua distro general menyediakan Apache. Konfigurasi Apache dapat dilakukan dengan sangat mudah dan dokumentasi Apache sangatlah bagus.

Sementara, untuk bekerja dengan WebDAV, semua sudah mampu ditangani oleh Apache itu sendiri melalui modul `dav` dan `dav_fs`. Anda hanya butuh mengonfigurasi Apache dan dalam waktu relatif singkat, Anda sudah mendapatkan fasilitas Web server yang WebDAV enabled.

Berbagai application server, seperti Zope, juga telah mendukung WebDAV dengan sangat baik. Beberapa desktop populer seperti Windows dan KDE telah menyediakan cara yang sangat mudah untuk bekerja dengan WebDAV.

Untuk FTP, apabila Anda tidak ingin menggunakan WebDAV, Anda bisa dengan mudah menginstal FTP server dan mengaktifkannya! Client FTP juga di mana-mana. Di Windows, Windows Explorer akan menjadi client FTP yang baik. Begitupun juga dengan Konqueror di KDE.

Pendekatan file sharing a la Internet boleh dikatakan sangat mudah untuk diterapkan.

### Keuntungan

Keuntungan file sharing a la Internet terutama karena OS independent. Anda bisa mengakses file dari berbagai sistem operasi, mesin dan perangkat *mobile*. HTTP adalah protokol standar. Sementara, di desktop, apapun sistem operasi Anda, FTP juga sudah sangat umum ditemukan dan sangat mudah untuk dilakukan.

WebDAV, walaupun belum semudah dan seumum FTP, sudah didukung oleh berbagai OS. Paling tidak, Windows dan Linux serta berbagai UNIX sudah mendukung dengan baik.

Anda dapat mengatur dan menyediakan file sharing yang sangat independen dan dapat diakses langsung dari Internet.

### Kekurangan

Walaupun memiliki banyak kelebihan, masalah file sharing a la Internet juga memiliki cukup banyak kelemahan. Yang paling dirasa adalah masalah kedekatan dengan sistem operasi dan filesistem (wajar saja,

karena di bagian keuntungan, cara ini paling OS independent).

Artinya, akan sedikit lebih sukar untuk mengatur hak akses file dan skema keamanan yang paling baik. Apabila menggunakan SMB atau NFS, dengan mudah masalah hak akses bisa kita selesaikan.

Selain itu, spesifikasi ACL (*Access Control List*) untuk WebDAV terhitung baru dan implementasinya masih membutuhkan waktu. Beberapa web server plus WebDAV memang datang dengan dukungan ACL sendiri.

Selain itu, untuk kebutuhan integrasi lokal, masih tidak banyak platform yang mendukung. Oleh karenanya, aplikasi tambahan mungkin masih harus digunakan. Dan, hal ini tidaklah trivial.

### Integrasi lokal

Untuk integrasi lokal, kita bisa menggunakan `davfs`. `Davfs` bisa di-download di <http://dav.sf.net>. Dengan demikian, kita bisa melakukan mounting volume WebDAV kita dan kemudian mengaksesnya seperti halnya direktori lokal.

Untuk command line tool, Anda barangkali akan senang mempergunakan program *Cadaver*. *Cadaver* akan berfungsi seperti client ftp. Anda mengelola file-file di volume WebDAV Anda dengan perintah seperti perintah ftp.

### Keamanan

Untuk masalah keamanan di server, terutama berhubungan dengan hak akses, beberapa hal harus diatur sehingga HTTP server tidak dijadikan target *Denial of Service*. Sementara, untuk masalah keamanan data lintas jaringan, kita bisa memanfaatkan protokol HTTPS.

Sementara, untuk FTP, kita bisa menggantikan dengan penggunaan OpenSSH (yang juga datang dengan Secure FTP).

WebDAV sendiri juga bisa bekerja dalam protokol HTTP Secure. Program `davfs` untuk melakukan mounting volume WebDAV juga mendukung HTTPS.

Semua hal tersebut telah disediakan dan telah berfungsi. Walau, kita harus terus cermat dalam melakukan update.

### Lebih detail

Pembahasan ini akan difokuskan pada

WebDAV. Umumnya, untuk mengelola suatu web, orang akan mempergunakan ftp atau e-mail. Melakukan perbaikan lokal, dan setelah itu mengirimkan kembali. Akan lebih sederhana kalau dikerjakan sendiri. Namun, kalau melibatkan banyak orang, hal akan menjadi jauh lebih kompleks.

Dengan hadirnya WebDAV (RFC 2518 dan WebDAV+ACL RFC 3744), kita dapat dengan mudah melakukan kolaborasi dalam pengaturan konten, tanpa tergantung pada platform yang dipakai oleh user dan tanpa peduli sistem operasi yang digunakan.

Beberapa fitur utama WebDAV adalah:

- **Locking:** WebDAV memiliki kemungkinan diakses melalui Internet di mana koneksi pengguna bisa terputus setiap saat. Bagaimana kalau client terputus sementara sedang memegang hak eksklusif untuk menulis ke dalam sebuah file? Masalah *locking* di sini tidaklah sederhana. Walau demikian, *locking* pada WebDAV sudah didesain dengan sangat baik dan implementasinya juga sudah sangat bisa diandalkan.
- **Properties.** Properti XML dapat diatur menggunakan protokol WebDAV. Masing-masing objek di volume WebDAV dapat memiliki properti yang cukup lengkap.
- **Mamespace manipulation.** Dengan adanya fasilitas seperti ini, kita dapat bekerja seperti halnya pada filesistem. File dapat dibuat, di-copy, dihapus, dan lain sebagainya. Begitupun dengan direktori.

Beberapa tambahan juga sedang dikerjakan dan beberapa sudah dalam tahap akhir persetujuan atau tahapan awal implementasi (beberapa pihak telah melakukan implementasi tanpa mengacu kepada persetujuan):

- **Advanced Collection.** Membuat volume WebDAV lebih mirip dengan filesistem yang kita kenal.
- **Versioning and Configuration Management,** mendukung versioning secara penuh.
- **Access Control,** saat ini masih tergolong baru dan membutuhkan waktu sampai diimplementasikan penuh.

## USER DAN LINGKUNGAN KERJA

Hampir semua teknologi menawarkan solusi yang cukup baik dalam file sharing. Namun, salah pilih teknologi atau terlalu fanatik pada suatu teknologi bisa-bisa meyesatkan, dan lebih parah lagi, bisa-bisa menyusahkan user.

Di sini, kita memikirkan bagaimana agar file sharing bisa cukup transparan bagi user, sekaligus transparan bagi perubahan minor. Akan lebih baik lagi, jika proses transisi juga tidak terlalu memakan biaya.

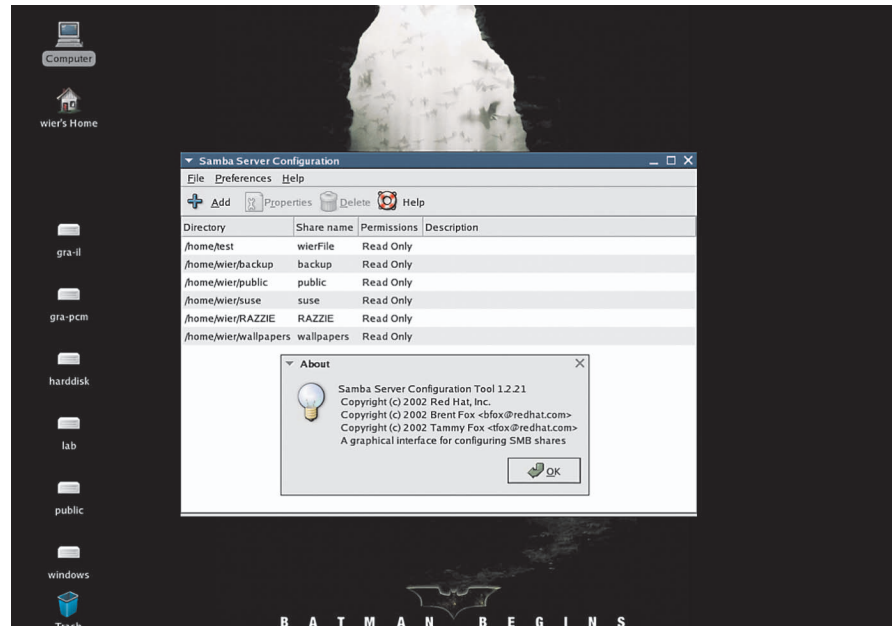
Beberapa pengaturan tambahan bisa diterapkan sehingga file sharing dapat dimanfaatkan dengan optimal. Beberapa penyesuaian mungkin juga harus dilakukan. Dan, dalam mengelola jaringan besar, penyeragaman cara kerja juga perlu dipikirkan. Bagaimana mengakses resource di jaringan, bagaimana melakukan revisi dan lain sebagainya.

Kita tentunya juga harus melihat lingkungan kerja yang nonhomogen. Suatu teknologi mungkin supercanggih, namun, kalau membutuhkan perlakuan khusus atau kasarnya menyusahkan banyak pihak, kita perlu berpikir sekali lagi untuk menerapkan teknologi tersebut.

Lingkungan kerja nonhomogen memang susah dan mungkin susah dihindari. Jangan heran apabila Anda ternyata menonaktifkan NFS dan menggunakan hanya protokol SMB padahal lebih dari 50% workstation Anda adalah Linux. Server-nya juga Linux.

Intinya, kita perlu memahami bahwa untuk printing dan file sharing, SMB masih cukup menawan, walaupun kita menggunakan Linux. Implementasi Printing dan file sharing ala Linux atau UNIX harus dikonfigurasi lagi dan tidak terintegrasi sebaik SMB yang satu protokol. Sementara, cara sharing ala Internet mungkin bukan cara yang ideal karena tidak sematang kedua sebelumnya dan relatif memiliki banyak isu (untuk WebDAV).

Jangan lupa untuk memperhatikan transparansi untuk user. User sebenarnya tidak harus terlalu tahu banyak hal teknis. Cara untuk bekerja dengan file sharing juga harus dibuat setransparan mungkin. Kalau perlu, misalnya di komputer berbasis



Konfigurasi Samba.

Windows, begitu habis login, sebuah drive sudah diasosiasikan untuk file sharing (apapun cara Anda, baik ala Windows, Linux, maupun internet). Kalau terjadi perubahan cara sharing misalnya, user juga harus diperhatikan.

Ketahuilah apa yang paling dibutuhkan dan solusi mana saja yang membutuhkan biaya lebih sedikit. Perhatikan juga di masa depan.

## ASPEK KEAMANAN?

Di manapun, keamanan adalah hal yang harus diperhatikan. Menyimpan file secara lokal adalah hal yang sebenarnya paling aman. Lebih aman lagi jika tidak terhubung ke jaringan dan melakukan semuanya sendiri. Namun, tentu saja hal ini tidak bisa dilakukan.

Dalam berbagi file yang tidak bersifat rahasia, seperti MP3, MPEG movie, gambar, dan lain sebagainya, tentu saja keamanan bukan masalah besar. Namun, bagaimana kalau perusahaan menerapkan penyimpanan dokumen tersentralisasi? Mulai dari Anda membuat dokumen, mentransmisikan dokumen, sampai dokumen tersimpan di server, ada salah celah bagi pihak yang tidak bertanggung jawab untuk mengintip dokumen Anda.

Masalah keamanan adalah hal yang harus ikut naik dipikirkan ketika kita me-

nyusun infrastruktur file sharing, baik untuk penyimpanan data ataupun sekadar berbagi *resource*. Tidak bisa tidak. Ketika kita bicara soal suatu teknologi, otomatis masalah keamanan akan membuntuti.

Mulai dari filesistem, kini kita mengenal enkripsi pada filesistem. Apabila mungkin, lakukanlah proteksi *se-low level* mungkin. Selain itu, perhatikan juga keamanan jaringan dan penggunaan ACL. Saat ini, file-file sistem populer sudah mendukung ACL dengan baik. Untuk file sharing ala Internet, jangan lupa untuk menggunakan protokol HTTP secure sebisa mungkin.

Dari sisi user, seharusnya kita tidak membiarkan user untuk mengisikan *password* yang berbeda untuk mengakses dokumennya. User mungkin akan memberikan password yang mudah atau menempelkannya di layar monitor. Gunakan pendekatan single sign on dan pastikan user bisa bekerja dengan semudah mungkin.

Demikianlah pembahasan kita mengenai file sharing dan berbagai peran perniknya. File sharing bukan masalah sederhana yang bisa dikerjakan sambil lalu. Kita perlu memikirkan dari ujung ke ujung mulai dari manajemen, keamanan dan pengaturan di lingkungan kerja. Bagi Anda yang berminat, Anda bisa mengakses <http://docs.sys.keant.com> untuk pengaturan sistem dokumen, yang salah satunya memanfaatkan file sharing dan ACL. 🐱