



# Lebih Detail dengan Filesystem

Linux mendukung sangat banyak *filesystem*, mulai dari filesystem *native* sampai filesystem untuk sistem operasi lain. Kita akan melihat lebih detail tentang beberapa filesystem populer yang digunakan di Linux.

Ketika kita menggunakan DOS atau Windows, umumnya, apa yang perlu kita ketahui soal bekerja dengan harddisk adalah membuat partisi (misal dengan fdisk) dan memformat partisi yang kita buat tersebut. Bahkan sejujurnya, terkadang yang perlu kita ketahui hanyalah memformat drive C agar dapat menginstal DOS atau Windows ke dalamnya.

Ketika kita berpindah ke Linux, ada satu tambahan pertanyaan ketika kita bekerja dengan harddisk. Pertanyaan tersebut adalah filesystem yang akan kita gunakan. Sekitar lima atau enam yang tahun lalu, filesystem yang tersedia untuk digunakan sebagai partisi Linux tidaklah banyak. Umumnya orang akan menggunakan ext2 (dibaca sebagai second extended filesystem) dan beberapa *pengoprek* yang akan mencoba-coba patch reiserfs di Linux. Namun saat ini, banyak filesystem baru telah lahir. Sebut saja ext3, jfs, xfs, dan lain sebagainya. Ini belum lagi ditambah dengan berbagai distributed filesystem lain yang didukung oleh Linux.

Di satu sisi, kondisi ini membuat pengguna Linux harus berhadapan dengan pilihan filesystem, yang memiliki konsekuensi cukup besar. Di sisi lain, banyaknya dukungan akan filesystem menjadikan Linux sebagai sistem operasi yang boleh dikatakan paling banyak mendukung filesystem yang ada. Kita bisa dengan mudah membaca partisi milik sistem operasi lain.

Di artikel ini, kita akan membahas filesystem secara umum beserta isu *journaling*, sejarah filesystem di Linux, serta berbagai filesystem yang populer di Linux.

## Filesystem secara umum

Secara umum dan sederhana, filesystem adalah suatu format internal bagaimana data diorganisasikan di dalam partisi harddisk. Di dunia DOS dan Windows, kita mengenal

filesystem FAT (8 dan 16bit), FAT32 dan NTFS. Di Linux, saat ini, filesystem populer diantaranya ext2, ext3, reiserfs, xfs, dan jfs. Pengguna Linux yang ingin mudah berbagi data dalam satu komputer juga umumnya akan bekerja dengan filesystem vfat.

Ketika kita membeli harddisk baru, maka kita perlu menyiapkan partisi-partisi untuk menyimpan sistem dan data. Dalam membuat partisi, kita harus menyiapkan tipe partisi. Untuk Linux, kita harus memilih partisi khusus Linux (kode heksa fdisk: 83). Setelah itu, kita perlu menyiapkan filesystem di partisi tersebut. Kegiatan inilah yang disebut sebagai memformat di DOS atau Windows.

Karena kita bebas memilih filesystem, maka tidak ada perintah 'format' yang umum. Walau, hampir semua pembuat filesystem dan distro telah memaketkan perintah 'format' tersebut sebagai mkfs.<filesystem>, dimana <filesystem> digantikan dengan nama filesystem. Sebagai contoh: untuk menggunakan filesystem reiserfs, maka kita harus membuat filesystem reiserfs dengan perintah mkfs.reiserfs sebagai root. Setelah filesystem dibuat (atau, setelah diformat), kita pun dapat segera dapat menggunakan partisi tersebut.

Sekarang, bagaimana kalau ada masalah pada filesystem? Apabila di DOS/Windows, kita mengenal program scandisk (dan variasinya, misal: Norton Disk Doctor yang sangat sering penulis gunakan ketika masih menggunakan Windows). Di Linux, sekali lagi, karena filesystem-nya berbeda-beda, maka tidak ada 'scandisk' yang umum. Sebagai gantinya, sama seperti 'format' yang kita bahas sebelumnya, hampir semua pembuat filesystem dan distro telah memaketkan perintah 'scandisk' tersebut sebagai fsck.<filesystem>, di mana <filesystem> digantikan dengan nama filesystem. Sebagai

contoh, untuk memeriksa filesystem reiserfs, maka gunakanlah perintah fsck.reiserfs.

Sebagai tambahan, istilah mkfs merupakan pemendekan dari make filesystem dan fsck merupakan pemendekan dari filesystem check.

Lebih lanjut lagi, apabila di DOS/Windows kita mengenal Partition Magic yang dapat me-*resize* partisi, bagaimana dengan di Linux? Tool semacam itu ada, dan GNU parted adalah contoh yang paling baik. Sayangnya, tidak (atau belum) semua filesystem di Linux dapat di-*resize*, baik secara *online* ataupun *offline*. Pembuat filesystem juga umumnya datang dengan tool resizer sendiri.

Yang terakhir, apabila di Windows kita mengenal istilah *undelete* (berserta seribu satu tool 3rd party lainnya), tidak semua filesystem di Linux mendukung sistem *undelete*. Ada yang bisa, ada yang tidak, walau, lebih banyak yang tidak. Sifat keras ini umumnya ditengahi dengan *recycle bin* di desktop KDE atau GNOME.

## Sejarah filesystem di Linux

Sebelum kita melanjutkan, mari kita lihat sejenak bagaimana sejarah filesystem di sistem operasi penguin gendut ini. Melongok sejenak ke masa-masa 14 tahun lalu, Linux pertama dikembangkan di bawah sistem operasi Minix (oleh Andrew Tanenbaum, Belanda). Untuk memudahkan berbagi data, maka Linux pun mengembangkan dukungan untuk minixfs. Namun, setelah digunakan dan mengetahui bahwa minixfs sangatlah terbatas (ukuran maksimal filesystem adalah 64 MB karena pengalamatan blok disimpan dalam integer 16 bit, direktori mengandung entri terbatas, panjang nama file maksimal 14 karakter, dan lain-lain), maka Linux pun memilikikan filesystem lain yang lebih hebat.

Sadar bahwa Linux akan memiliki banyak filesystem, maka Linus pun merancang layer *Virtual File System* (VFS), agar filesystem dapat diintegrasikan dengan mudah. Setelah VFS dikembangkan, filesystem baru dengan nama EXT (extended filesystem) pun dikembangkan pada April 1992 dan dimasukkan pada kernel versi 0.96c. Filesystem EXT ini mengizinkan maksimal ukuran filesystem sampai 2 GB dan maksimal panjang nama file adalah 255 karakter. Namun sayangnya, EXT masih memiliki sejumlah kelemahan. Di antaranya adalah masih belum ada akses terpisah, masih belum ada modifikasi inode dan masih belum ada modifikasi timestamp. Filesystem ini menggunakan *linked-list* untuk melacak blok yang masih belum terpakai, di samping untuk melacak inode. Hal ini, menyebabkan menurunnya performa apabila ukuran filesystem membesar.

Untuk mengatasi masalah pada EXT, maka dua filesystem baru pun dirilis untuk Linux pada arsitektur Alpha pada bulan Januari 1993, yaitu EXT2 (akan dibahas tersendiri) dan XIAFS (oleh Frank Xia, saat ini sudah tidak dikembangkan). XIAFS pada saat itu berbasis source code kernel untuk minixfs dan menambahkan beberapa fitur. Utamanya, XIA menambahkan nama file panjang, partisi yang lebih besar dan dukungan timestamp. Sementara, EXT2 berbasis source code EXT dan menambahkan banyak fitur serta memperbaiki cukup banyak hal.

Sebagai perbandingan filesystem Linux di masa awal, mari lihat tabel 1: Perbandingan filesystem Linux di masa awal.

Semenjak itu, berbagai filesystem baru pun lahir. Kini, kita memiliki puluhan filesystem yang didukung, dengan empat journaling filesystem sangat populer, serta secara umum, Linux memiliki filesystem yang sangat stabil.

## Journaling Filesystem

Dari awal, kita beberapa kali menyinggung istilah journaling filesystem. Sebenarnya, apakah journaling filesystem tersebut? Untuk jelasnya, mari kita lihat apa yang terjadi apabila sistem kita *crash* ketika kita sedang bekerja dengan dokumen.

Kita sedang bekerja dengan dokumen A. Tiba-tiba, listrik mati. Pada saat mesin *crash*

Tabel 1. Perbandingan filesystem Linux di masa awal.

	MINIX	EXT	EXT2
Maksimal ukuran FS	64M	2G	4T
Maksimal ukuran file	64M	2G	2G
Maksimal panjang nama file	16/30C	255C	255C
Timestamp	Tidak	Tidak	Ya
Bisa dikembangkan	Tidak	Tidak	Ya
Variabel blocksize	Tidak	Tidak	Ya

sebelum Anda menyimpan, maka Anda memang kehilangan perubahan terbaru, walaupun, data Anda yang asli masih tidak tersentuh. Kalau mesin crash setelah Anda menyimpan perubahan, maka inilah yang kita harapkan. Data dan perubahannya tersimpan. Masalahnya, bagaimana kalau mesin crash pada saat proses penyimpanan sedang berlangsung? Ini adalah mimpi buruk. Data Anda mungkin akan tidak terbaca oleh aplikasi yang Anda gunakan karena data tidak konsisten lagi dengan format yang diharapkan aplikasi. Tapi, ini bukan kasus paling buruk. Yang paling buruk, selain kerusakan hardware, bagaimana kalau kerusakan terjadi pada metadata filesystem? Anda mungkin tidak hanya berhubungan dengan data yang rusak, tapi filesystem yang rusak. Anda mungkin kehilangan semua data, Anda mungkin bisa menyelamatkan sebagian di antaranya.

Filesystem umumnya membagi penyimpanan dalam metadata dan data. Untuk mencegah korup pada metadata, ada beberapa filesystem yang mengimplementasikan multiple metadata. Tapi, ini bukan solusi yang baik. Untuk mencegah semua hal yang tidak diinginkan pada ilustrasi dokumen A tersebut, maka journaling filesystem pun diperkenalkan.

Secara sederhana, journaling filesystem menggunakan log untuk mencatat semua hal yang dilakukan pada filesystem. Jadi, sebelum perubahan dilakukan pada filesystem, akan ada entri journal yang mengatakan bahwa perubahan akan dilakukan. Setelah itu, perubahan pun dilakukan. Setelah itu, perubahan dilakukan, maka kita akan kembali ke journal dan menandai sebagai berhasil, atau kalau gagal, maka semua akan dikembalikan ke status awal.

Dengan demikian, kembali ke kasus Dokumen A tersebut, apabila mesin crash pada tahapan pertama, bahwa ada entri perubahan pada journal, maka perubahan

belum dilakukan. Perubahan terbaru pada file memang hilang, namun file asli tetap tidak tersentuh. Apabila mesin crash pada saat penulisan pada disk, maka perubahan memang telah dilakukan. Tapi, entri pada journal masih belum ditandai sebagai komplit ataupun gagal. Ketika *reboot*, maka journal akan dicek lagi. Pada saat itulah, journal bisa diperiksa, dan akan diputuskan untuk menandai entri perubahan sebagai komplit, atau semua digagalkan dan kembali ke status awal. Kondisi demikian mirip dengan transaction pada SQL.

Dengan memanfaatkan journaling filesystem, kita akan lebih terbebas dari pemeriksaan superlama pada saat reboot, dan data kita akan senantiasa lebih aman.

Setelah ini, kita akan membahas beberapa filesystem populer di Linux, yaitu EXT2, EXT3, REISERFS, JFS, XFS, dan VFAT.

## EXT2

Filesystem yang satu ini merupakan filesystem yang paling populer di Linux. Boleh dikatakan, pada masa-masa awal sampai saat ini, filesystem ini pula yang membantu Linux menjadi semakin populer. Pada awalnya, EXT2 didesain untuk memperbaiki kekurangan pada filesystem EXT.

Hasil karya tiga hacker hebat, yaitu Theodore Ts'o, Remy Card, Stephen Tweedie ini memiliki fitur-fitur standar sebagai berikut:

- Mendukung berbagai tipe file standar UNIX.
- Dengan modifikasi pada layer VFS, filesystem ini bisa berukuran sampai 4T.
- Mendukung nama file panjang (255 karakter sampai 1012 karakter apabila diinginkan).
- Dapat memesan beberapa tempat untuk superuser (umumnya 5%) agar membantu ketika ada masalah.

Di luar fitur-fitur standar tersebut, EXT2 juga memiliki fitur tambahan, di antaranya:

- Bloksize yang bervariasi.
- File dapat dihapus dengan aman.
- Mendukung *flag immutable* pada file, yang tidak dapat diubah atau dihapus oleh siapapun.
- Memiliki *hook* untuk ditambahkan banyak fitur. Seperti *online compression* dan *undelete*. Dengan adanya *hook* tersebut, filesystem ini siap dikembangkan lebih lanjut.

Dari sisi desain, filesystem ini sangat banyak dipengaruhi oleh filesystem BSD. Sistem EXT2 sendiri dibagi dalam sistem group, yang merupakan bagian dari partisi. Pembagian ini dilakukan pada saat format dan tidak dapat diubah sebelum pemformatan ulang. Setiap group mengandung superblock, group descriptor, block bitmap, inode bitmap, inode table, dan data block dalam urutan seperti ini.

Dalam membaca, untuk mempercepat, beberapa blok yang berurutan akan dialokasikan sehingga blok berikut akan dapat dibaca lebih cepat (karena sudah disimpan di dalam *buffer*). Sementara, dalam melakukan penulisan, EXT2 melakukan prealokasi sampai delapan blok yang berdekatan pada saat alokasi blok baru. Dengan demikian, diharapkan penulisan dapat dilakukan jauh lebih cepat daripada selalu harus memesan untuk penulisan setiap blok.

EXT2 dapat di-*resize* dengan aman baik pada saat *offline* ataupun *online*. Untuk *resize offline*, tool *ext2resize* bisa digunakan. Sementara, untuk *resize online*, *ext2online* bisa digunakan.

Di luar semua fiturnya, filesystem ini memiliki satu kelemahan yang cukup mengganggu. Masalahnya adalah penggunaan sistem blok. Setiap file akan ditulis dalam blok-blok sesuai ukuran file yang akan disimpan. Misal, ukuran blok adalah 1024. Maka, penulisan file berukuran 1025 byte akan disimpan pada dua blok, padahal blok kedua hanya berisi satu byte. Memang terlihat sedikit yang terbuang, tapi, apabila kita memiliki banyak file berukuran tergantung tersebut, maka space yang terbuang tidak bisa dibuang sedikit. Kelemahan ini memang bisa diakali dengan mengubah bloksize, tapi, secara keseluruhan, tidak bisa mengatasi masalah inefisiensi ini dengan tuntas.

Bagi developer, EXT2 menyediakan pustaka yang mengizinkan program user mode untuk memanipulasi struktur kontrol filesystem. Banyak aplikasi besar yang memanfaatkan pustaka ini. Sebut saja *mke2fs*, *e2fsck*, *tune2fs*, *dumpe2fs*, dan *debugfs*.

Dan, bagi pengguna sistem operasi lain, filesystem EXT2 bukanlah barang aneh. Banyak sekali tool-tool sistem operasi lain yang bisa mengakses atau bahkan membuat filesystem ini. Beberapa contoh di antaranya:

- DOS: <ftp://sunsite.unc.edu/pub/Linux/system/filesystems/ext2/>.
- DOS/Windows atau UNIX lain, *ltools* bisa didapatkan di <http://www.it.fht-esslingen.de/%7Ezimmerma/software/ltools.html>.
- Akses dari Windows 95, *explore2fs*, bisa didapatkan di <http://uranus.it.swin.edu.au/%7Ejnl/linux/explore2fs.htm>.
- Partition Magic (<http://www.powerquest.com>).

## EXT3

Dari namanya, kita tahu bahwa EXT3 adalah kakak dari EXT2. EXT3 juga dibuat oleh salah seorang pembuat EXT2, yaitu DR. Stephen Tweedie. Secara kasar, EXT3 merupakan EXT2 yang ditambahkan fitur journaling dan beberapa fitur lain.

EXT3 kompatibel ke bawah dengan EXT2, dan pengguna EXT2 bisa melakukan konversi ke EXT3 dengan memanfaatkan program *tune2fs* (dengan opsi *-j*) dan setelah itu dapat di-*remount* sebagai EXT3. Namun, tentu saja, dukungan EXT3 pada kernel harus diaktifkan terlebih dahulu. EXT3 tercatat resmi didukung oleh kernel 2.4.16.

Bagi Anda pengguna Red Hat dan turunannya (misal: Fedora Core), Anda bisa melihat bahwa secara resmi, Red Hat akan menyarankan EXT3 dan EXT3 merupakan filesystem default pada saat pembuatan partisi. Di beberapa benchmark antara Red Hat dan Windows Server misalnya, filesystem yang digunakan adalah EXT3.

Hal ini wajar karena EXT3 di-host oleh Red Hat. DR Stephen Tweedie adalah karyawan Red Hat dan bertanggung jawab penuh untuk Linux. Bahkan, Red Hat memiliki alasan kenapa EXT3 digunakan:

- *Backward* dan *forward* kompatibel de-

ngan EXT2. EXT3 bahkan bisa di-mount sebagai EXT2 tanpa harus menghapus journal.

- EXT2 memiliki sejarah yang cukup baik dan EXT3 melanjutkan.
- Tidak tergantung pada vendor (hal ini setidaknya didasari alasan bahwa JFS tergantung IBM, yang sesungguhnya sudah tidak lagi semenjak JFS dilepas ke publik, dan alasan bahwa XFS tergantung pada SGI).
- Tool-tool pendukung seperti *e2fsck* sudah sangat matang.
- Mengandung beberapa mode journaling: yang pertama adalah semua metadata dan data dan yang kedua adalah hanya metadata.

## JFS

JFS merupakan journaling filesystem yang dibuat oleh IBM, dan saat ini sudah dilepas ke publik. JFS bukanlah barang baru karena teknologi JFS telah diterapkan pada AIX selama lebih dari 10 tahun. JFS 1.0, yang telah siap untuk lingkungan produksi, telah dirilis pada tanggal 28 Juni 2001.

Saat ini, JFS telah matang di Linux. JFS telah dimasukkan secara resmi ke kernel oleh Marcello pada seri 2.4.20, dan oleh Linus pada 2.5.6. Sayangnya, tidak banyak distribusi Linux yang memaketkan JFS sebagai filesystem default.

JFS menawarkan performa, robustness, dan dukungan SMP yang baik. Dan, dengan adanya JFS, pengguna IBM yang ingin berpindah ke Linux setidaknya memiliki jalur migrasi yang lebih jelas pada level filesystem.

Berikut ini adalah fitur JFS:

- Journaling filesystem. JFS merupakan salah satu dari journaling filesystem populer di Linux.
- Extents-based filesystem dengan algoritma B+Tree. JFS memiliki kebijakan dalam menangani fragmentasi dengan memanfaatkan extents, atau bagian-bagian kecil secara logikal dalam filesystem. Untuk mengoptimalkan filesystem, JFS mengandalkan utility untuk melakukan defragmentasi.
- Bloksize yang bervariasi, mulai dari 512 B sampai 4K.
- Dynamic disk inode allocation, yang memungkinkan alokasi inode pada saat

dibutuhkan dan tidak harus pada saat pembuatan filesystem.

JFS sendiri juga memiliki beberapa kekurangan seperti:

- Ukuran minimal filesystem adalah 16M, dan ukuran maksimal adalah 512 T untuk blok 512B dan 4 P (4 Petabyte) untuk blocksize 4K.
- Ukuran file sesuai dukungan VFS Linux.
- Tidak bisa diterapkan pada disket dan beberapa media removable lain.

## REISERFS

Reiserfs adalah filesystem yang dibangun oleh **Hans Reiser** (Naming System Venture, namesys.com) dan merupakan filesystem journaling pertama yang dimasukkan ke kernel Linux (Reiser bisa jalan pada setiap arsitektur yang didukung oleh Linux sejak 2.4.13). Dengan demikian, Reiserfs memiliki *track record* yang paling lama untuk journaling di Linux. Reiserfs juga merupakan filesystem default pada distribusi SUSE. Pada kenyataannya, untuk Reiserfs 3.x, SUSE (Chris Mason) menyumbang sangat banyak untuk pengembangan Reiserfs. Untuk reiserfs 4.x, pengembangan Reiserfs disponsori langsung oleh DARPA dan memiliki standar keamanan militer Amerika Serikat.

Reiserfs 3.x memanfaatkan *balanced tree*, di mana Reiserfs akan melakukan kunjungan tree lengkap untuk setiap 4K blok yang ditulis, dan kemudian memasukkan satu pointer ke tree tersebut. Hal ini berarti setiap penulisan sebanyak 4K menyebabkan balancing pada tree (istilah nonteknisnya, pemindahan data). Hal ini juga berarti bahwa dalam *balanced tree*, selain metadata, data juga disimpan. Kondisi demikian membuat Reiserfs kalah cepat dengan XFS dalam menangani file-file besar. Apalagi, pada saat kernel 2.4.x, reiserfs menggunakan algoritma generic untuk membaca dan menulis data, yang walau mudah diintegrasikan ke dalam kernel, tapi memiliki masalah performa pada file besar. Untuk reiserfs 4.x, *balanced tree* sudah tidak digunakan lagi. Algoritma yang digunakan adalah *DANCING Tree*.

Reiserfs memiliki ciri khas sebagai filesystem yang terus menyempurnakan algoritmanya untuk dapat diadopsi pada filesystem untuk dunia enterprise. Dalam

beberapa hal, fitur reiserfs sedikit terlambat dengan filesystem lain. Dukungan kuota misalnya. Pada seri 2.4.x, patch (dari SUSE) harus diterapkan agar kuota dapat bekerja).

Reiserfs diketahui sangat cocok bekerja dengan LVM (*Logical Volume Manager*) yang dapat digunakan untuk melakukan virtualisasi filesystem. Pembuat LVM menyatakan bahwa lebih dari 90% pengguna LVM menggunakan Reiserfs sebagai filesystem dasar. Dengan memanfaatkan LVM, kita juga dapat membuat *snapshot* untuk filesystem reiser yang kita gunakan.

Untuk lebih detail, kita akan melihat beberapa spesifikasi teknis Reiser 3.x pada tabel 2: Spesifikasi reiserfs 3.x

Reiserfs 4.x sendiri, sebagai generasi terbaru dari ReiserFS, datang dengan fitur-fitur sebagai berikut:

- Filesystem tercepat menurut namesys.
- Atomic: ya atau tidak dalam satu kesatuan. Tidak ada *performance lost* karena menemukan algoritma baru tanpa mengopikan data dua kali.
- Menggunakan *dancing tree*, bukan *balanced tree*. Lebih efisien karena akan

memadatkan file-file kecil bersama-sama daripada membuang space. Butuh jutaan file dalam direktori dan cepat? Anda bisa melakukannya dengan Reiserfs.

- Datang dengan dukungan *plugin*. Anda kini bisa menambahkan plugin ke filesystem sama seperti menambahkan plugin ke GIMP. Tentunya, plugin akan bekerja secara low level dan sangat berpengaruh terhadap data itu sendiri.
- Didesain untuk keamanan level militer (DARPA adalah sponsor utama, bersama linspire dan SUSE).
- Mendukung multiple-stram attribute dan live query (LQ) milik BFS (BeOS). LQ adalah query yang mengirim otomatis perubahan pada saat perubahan dilakukan.
- Mendukung inheritance untuk isi file dan data statistik

Contoh kasus penggunaan Reiserfs yang cukup populer adalah FTP server milik SourceForge.Net. FTP server SourceForge.Net pada saat pemberian testimoni adalah 850 GB, dengan komposisi sebagian Reiserfs dan sebagian EXT2.

MORE SPACE  
 RELIABILITY & TIME  
 LESS...  
 & MONEY

**LINUX and FreeBSD**

Features :

- Unlimited data transfer
- Complete control panels
- POP3 email, FTP access
- SSH, CGI, SQL.
- and much more...
- Start from Rp. 19.500,-/ month
- Free Setup \*)
- 2 Months Free \*\*)

**Server Hosting**

Features :

- Location NOC Jakarta - Indonesia (IIX)
- Size server : 1 U Rackmount
- Bandwidth : 128 kbps
- IP Address : 8 (max)
- Colocation : Rp. 1.000.000,-/ month

**ALSO**

- Colocation & Dedicated Server in USA
- Domain Name Register
- Benefit Reseller Program

*"IT'S NEVER BEEN EASIER  
TO TAKE YOUR BUSINESS ONLINE"*

Note : \*) Transfer (restriction apply)  
\*\*) 1 year payment

**CAKRAWEB**  
*Supporting You to a Web Success*

Cyber Building (d/h Elektrindo) 10 th Floor  
 Jl. Kuningan Barat No. 8 Jakarta Selatan 12710  
 Phone. (021) 526 8000 Fax. (021) 52 66 444  
<http://www.cakraweb.com> - [info@cakraweb.com](mailto:info@cakraweb.com)



Di kantor tempat penulis bekerja, file server lokal menggunakan sepenuhnya Reiserfs dengan 20 G adalah partisi utama dan 300 G adalah LVM dengan dasar filesystem Reiserfs. Dari pengalaman, sangat stabil dan *resizing* dapat dilakukan tanpa masalah sama sekali.

Terakhir, bagi pengguna sistem operasi Windows yang ingin mengakses Reiserfs, Anda bisa mempergunakan tool *rfstool*, yang bisa mengizinkan Anda untuk mengakses partisi Reiserfs (bahkan dari Linux). Sayangnya, akses adalah *readonly* dan journal akan diabaikan.

## VFAT

Walaupun merupakan teknologi yang sudah cukup lama, banyak pengguna Linux yang tidak dapat berpindah dari filesystem ini. Alasan utamanya adalah karena menggunakan filesystem VFAT adalah cara termudah untuk berbagi data dari sistem operasi Linux dan Windows. Sayangnya, cara seperti ini juga memiliki kelemahan disamping desain filesystem itu sendiri. Microsoft mungkin akan menarik biaya lisensi dari penggunaan FAT (apabila teknologi yang satu ini berhasil dipatenkan).

Dukungan VFAT di Linux sudah ada sejak zaman dahulu kala, pada kernel 1.2.x. Di Linux, dukungan VFAT sangat baik dan kompatibel dengan dukungan LFN (*Long FileName*) pada Windows 9x dan NT.

Berikut ini adalah sedikit catatan bagi Anda yang menggunakan file-file besar di Linux dan terpaksa menggunakan VFAT:

- Ukuran maksimal file adalah 2G, sehingga Anda mungkin perlu memecah file Anda (menggunakan program *split*, yang merupakan bagian dari paket GNU *coreutils*).
- Beberapa problem berikut mungkin terjadi:
  - Pada kernel 2.4.x, lost cluster mungkin terjadi ketika ukuran filesystem melebihi 100G. Pada 2.6, menurut kasus-kasus yang dilaporkan, hal ini sudah lebih jarang terjadi.
  - Butuh RAM besar untuk disk besar pada saat *fsck* karena *dosfsck* mengopi FAT ke RAM.
  - Jangan lupa tambahkan `-F 32` pada saat format, karena kalau tidak, akan digunakan FAT 12 atau 16 tergantung ukuran disk.

Tabel 2. Spesifikasi reiserfs 3.x.

	Reiserfs 3.5.x (kernel 2.2.x)	Reiserfs 3.6.x (kernel 2.4.x)
Maksimal jumlah file	4G-3	4G-3
Maksimal jumlah file/dir	518701895, tapi pada kenyataan dibatasi hash function. Hash r5 mengizinkan sekitar 1200000 nama file tanpa collision.	4G-4, tapi pada kenyataan dibatasi hash function. Hash r5 mengizinkan sekitar 1200000 nama file tanpa collision.
Maksimal ukuran file	2G-1	1ExaByte (1000000 T) walau pada kenyataannya dibatasi 8T untuk sistem 32 bit.
Maksimal link ke file	64K	4G
Maksimal ukuran FS	16T (blocksize 4K)	16T (blocksize 4K)

## XFS

XFS adalah filesystem hebat dari SGI. Filesystem ini diport dari Irix, walaupun ketika diterapkan ke Linux, tim pembuat XFS harus mendesain dan menulis ulang dari awal. Walau demikian, ketangguhannya sudah tidak diragukan lagi karena telah terbukti pada sistem komersial seperti Irix. Untuk dapat menggunakan XFS, Anda harus menggunakan kernel 2.4.x atau yang lebih baru.

Berikut ini adalah beberapa fitur XFS:

- Direct IO akan membuat pekerjaan dengan file berukuran besar tetap cepat.
- Fast recovery.
- Extents based space management. Untuk menyiasati fragmentasi, XFS akan melakukan alokasi nomor blok ke blok setiap penulisan.
- Dukungan journaling.
- Mendukung extended attribute file.
- Mendukung Access Control List.
- Filesystem yang sudah 64bit ready. Ukuran filesystem bisa mencapai 9 ExaByte dengan blocksize minimal adalah 512B.
- Banyak *tool userspace* untuk *dump*, *restore*, *repair*, *grow*, *snapshot*, dan lain-lain
- Fast transaction.
- Jalan pada setiap arsitektur yang didukung oleh Linux.
- Mendukung Quota.

Sayangnya, filesystem ini tidak mendukung *undelete*, jadi berhati-hatilah dalam menghapus file. Anda juga tidak dapat menggunakan XFS bersama-sama LILO ketika menginstall LILO ke root filesystem. Hal ini dikarenakan XFS menggunakan blok 0, yang juga digunakan oleh LILO. Apabila Anda menginstall LILO ke MBR, maka hal

ini tidak terjadi sehingga Anda bisa menggunakan XFS bersama-sama LILO.

Diam-diam, banyak distribusi Linux telah mendukung XFS sejak lama, sebagai contoh:


- Mandrake, sejak 8.1
- SUSE, sejak 8.0
- Gentoo, sejak 1.0
- Slackware, sejak 8.1
- Knoppix, sejak 3.1
- TurboLinux, sejak 7.0
- Debian, sejak 3.1 Sarge
- Fedora, sejak core 2

## Memilih Filesystem

Banyak sekali filesystem yang hebat-hebat di Linux. Namun, filesystem manakah yang harus saya andalkan? Pertama, pastikan bahwa Anda menggunakan journaling filesystem untuk partisi yang penting. Setelah itu, sebaiknya Anda menggunakan filesystem yang didukung oleh distribusi yang Anda gunakan.

Apabila membutuhkan fitur-fitur khusus, maka pastikan Anda menggunakan filesystem yang sesuai. Anda butuh dan selalu bekerja dengan file-file besar, maka Anda bisa mempergunakan XFS. Penulis senang bekerja dengan LVM, maka penulis menggunakan Reiserfs. Reiserfs juga didukung resmi oleh distribusi yang penulis gunakan.

Yang jelas, gunakan filesystem yang untuk kapabel untuk menangani kebutuhan data Anda.

Demikianlah pembahasan kita tentang filesystem di Linux. Filesystem adalah bagian yang sangat penting dalam penyimpanan data. Dengan memahami lebih lanjut, kita bisa mengambil keputusan dengan lebih baik, dan memahami bagaimana kita harus menjaga data kita. 

**Noprianto** ([noprianto@infolinux.co.id](mailto:noprianto@infolinux.co.id))