

SISTEM FILE

WILLY PERMANA PUTRA



KONSEP FILE

- File adalah unit penyimpan logika yang diabstraksi sistem operasi dari perangkat penyimpan.
- File berisi informasi yang disimpan pada penyimpan sekunder (seperti magnetic disk, magnetic tape dan optical disk).
- Sebuah file mempunyai struktur tertentu tergantung tipenya.
- Tipe file terdiri dari data baik data numeric, karakter maupun binary serta program misalnya source program, object program dan executable program.



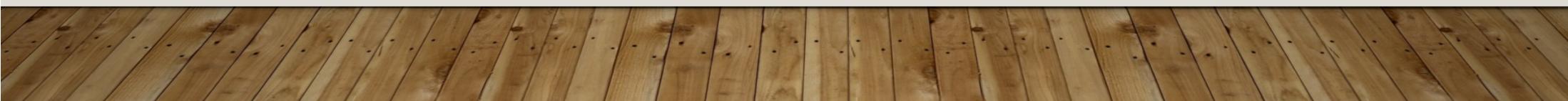
ATTRIBUT FILE

Sebuah file mempunyai atribut yang berbeda antara sistem operasi satu dengan lainnya, tetapi secara umum terdiri dari :

- **Nama**, informasi disimpan dalam bentuk yang dapat dibaca manusia
- **Tipe**, diperlukan sistem yang mendukung tipe yang berbeda.
- **Lokasi**, pointer ke lokasi file pada perangkat.
- **Ukuran**, ukuran file saat ini.
- **Proteksi**, mengontrol siapa yang dapat membaca, menulis dan mengeksekusi.
- **Waktu, tanggal** dan **identifikasi user**, data untuk monitoring proteksi, sekuritidan penggunaan.



Karakter	Arti
-	File biasa
d	Direktori
l	Symbolic link
b	Block special file
c	Character special file
s	Socket link
p	FIFO



- Ijin akses : menentukan hak user terhadap file ini.
- Jumlah link : jumlah link untuk file ini.
- Pemilik (Owner) : menentukan siapa pemilik file ini
- Group : menentukan group yang memiliki file ini
- Jumlah karakter : menentukan ukuran file dalam byte
- Waktu pembuatan : menentukan kapan file terakhir dimodifikasi
- Nama file : menentukan nama file yang dimaksud

Contoh :

```
-rw-rw-r--    1      bin      auth      1639   Oct 31 20:19  /etc/passwd
|           |      |        |          |
Ijin akses  Jml link Pemilik  Group    Jml karakter Waktu   Nama file
tipe
```

OPERASI PADA FILE

Sebagai tipe data abstrak, perlu didefinisikan operasi yang dapat dibentuk oleh file. Terdapat enam operasi dasar yg disediakan sebagai sistem call yaitu :

- Membuat file (create)
- Menulis file (write)
- Membaca file (read)
- Reposisi dalam file (file seek)
- Menghapus file (delete)
- Memotong file (truncate)

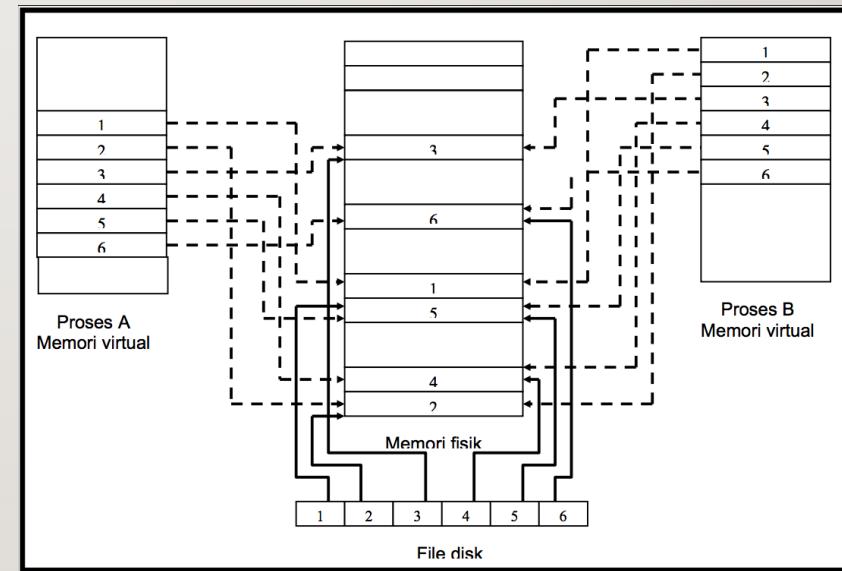
Operasi tambahan yang biasanya dilakukan terhadap file adalah :

- Menambah (append) informasi baru pada akhir file yang sudah ada
- Mengubah nama (rename) file yang sudah ada
- Membuat duplikasi (copy) file



BEBERAPA INFORMASI YANG BERKAITAN DG PEMBUKAAN FILE

- Pointer file.
- Jumlah file yang dibuka.
- Lokasi file pada disk.



TIPE FILE

- Salah satu pertimbangan penting dalam merancang sistem file dan keseluruhan sistem operasi adalah apakah sistem operasi mengenali dan mendukung sistem file.
- Bila sistem operasi mengenali tipe suatu file, maka dapat dilakukan operasi terhadap file
- Misalnya user yang mencoba mencetak file executable dapat dicegah oleh sistem operasi karena file adalah program binary.



TEKNIK UMUM TIPE FILE

- Implementasi tipe file adalah memasukkan tipe file sebagai bagian dari nama file.
- Nama file dibagi menjadi dua bagian yaitu nama dan extension (seperti pada MS-DOS)



LANJUTAN...

file type	usual extension	function
executable	exe, com, bin or none	read to run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

- Setiap file mempunyai atribut pembuat berisi nama dari program yang membuatnya (seperti pada MSWindows/ Apple Macintosh).
- Atribut ini di-set oleh sistem operasi saat menggunakan system call create.
- Bila user membuka file tersebut dengan melakukan double-clicking mouse pada icon dari file tsb, program yang dibuat ditampilkan otomatis.



STRUKTUR FILE

- Tipe file juga digunakan untuk menunjukkan struktur internal dari file.
- File tertentu harus konfirmasi ke struktur yang dibutuhkan yang dimengerti oleh sistem operasi. Misalnya sistem operasi membutuhkan file executable yang mempunyai struktur khusus sehingga dapat menentukan dimana letak memory dan lokasi dari instruksi pertama.



STRUKTUR FILE INTERNAL

- Secara internal, sistem disk mempunyai ukuran blok yang ditentukan oleh ukuran sebuah sector.
- Semua disk I/O dibentuk dalam unit satu blok (physical record) yang berukuran sama.
- Ukuran physical record tidak sama dengan panjang logical record, Logical record mempunyai panjang yang bervariasi Solusinya adalah dengan mengirim sejumlah logical record ke blok fisik.
- Ukuran logical record, ukuran blok fisik dan teknik pengiriman menentukan berapa banyak logical record yang berada pada blok fisik. Pengiriman dapat dilakukan oleh program aplikasi user atau sistem operasi.



METODE AKSES

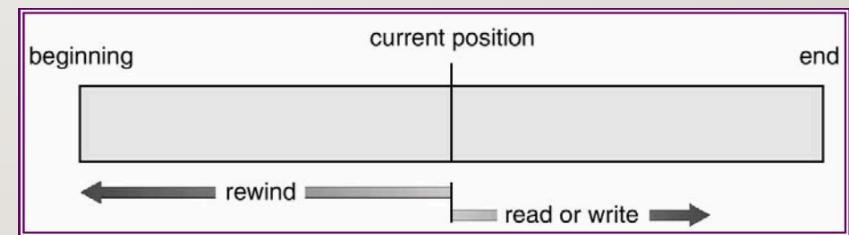
Terdapat beberapa cara mengakses informasi pada file yaitu

- akses berurutan (sequential access),
- akses langsung (Direct access atau relative access)
- metode akses lain.



AKSES BERURUTAN (SEQUENTIAL ACCESS)

- Akses berurutan merupakan metode akses paling sederhana. Informasi pada file diproses secara berurutan, satu record diakses setelah record yang lain.
- Metode akses ini berdasarkan model tape dari suatu file yang bekerja dengan perangkat sequential access atau random-access.
- Operasi pada akses berurutan terdiri dari :
 - read next
 - write next
 - reset
 - no read after last write (rewrite)



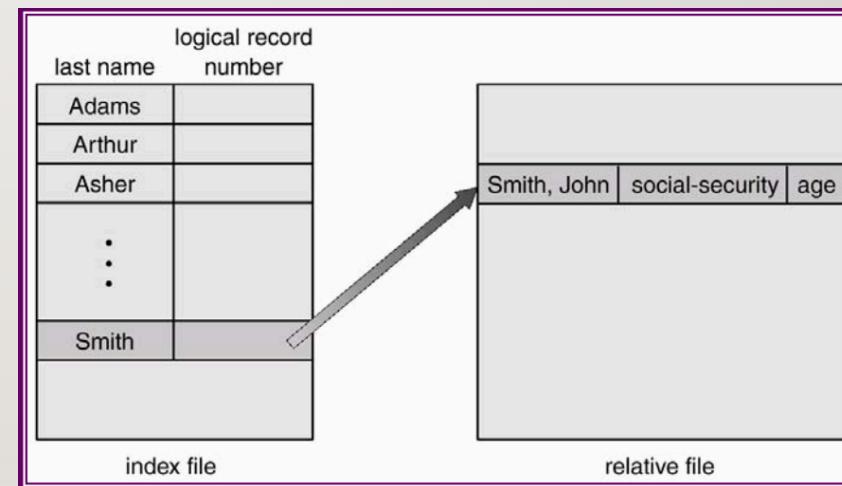
AKSES LANGSUNG (DIRECT ACCESS)

- File merupakan logical record dengan panjang tetap yang memungkinkan program membaca dan menulis record dengan cepat tanpa urutan tertentu.
- Metode akses langsung berdasarkan model disk dari suatu file, memungkinkan acak kesembarang blok file, memungkinkan blok acak tersebut dibaca atau ditulis.
- Operasi pada akses langsung terdiri dari :
 - read n
 - write n
 - position to n
 - read next
 - write next
 - rewrite n

sequential access	implementation for direct access
<i>reset</i>	$cp = 0;$
<i>read next</i>	<i>read cp;</i> $cp = cp+1;$
<i>write next</i>	<i>write cp;</i> $cp = cp+1;$

METODE AKSES LAIN

- Metode akses lain dapat dibangun berpedoman pada metode direct access.
- Metode tambahan ini biasanya melibatkan konstruksi indeks untuk file.
- Untuk menentukan masukan dalam file, pertama dicari indeks, dan kemudian menggunakan pointer untuk mengakses file secara langsung dan menemukan masukan yang tepat.



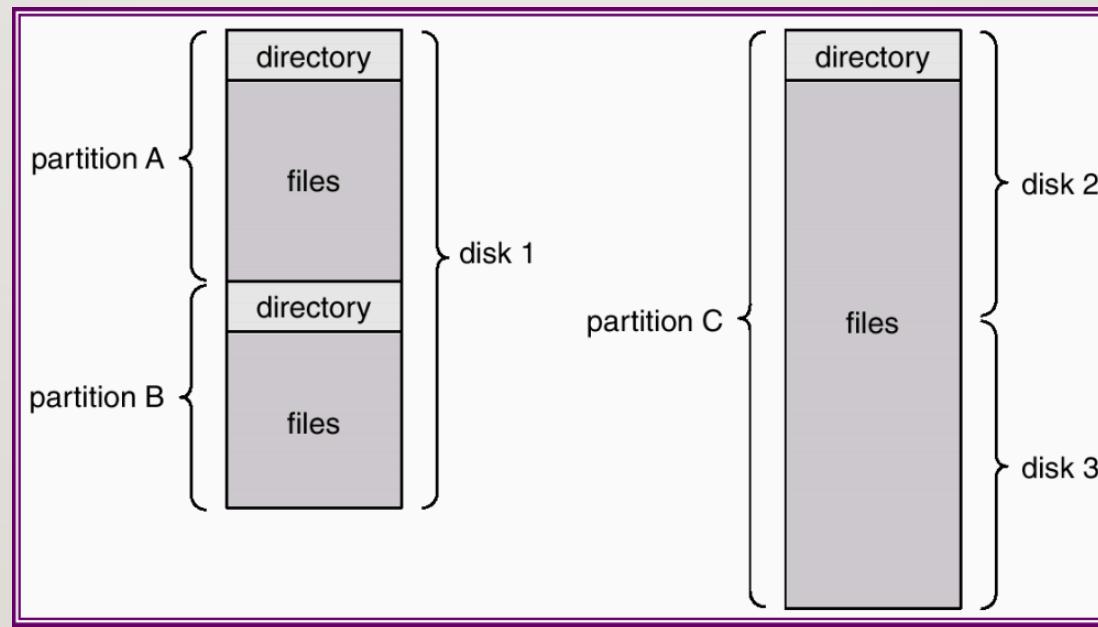
STRUKTUR DIREKTORI

Direktori adalah kumpulan titik yang berisi informasi tentang semua file, untuk mengatur semua data menggunakan organisasi yg dilakukan dalam dua bagian.

- Pertama, system file dipecah ke dalam partisi, yang disebut juga “**minidisk**”(pada mesin IBM) atau “**volume**” (pada mesin PC dan Macintosh). Setiap disk padasistem berisi sedikitnya satu partisi, merupakan struktur low-level dimana file dandirektori berada.
- Kedua, setiap partisi berisi informasi mengenai file di dalamnya. Informasi ini disimpan pada entry dalam “**device directory** atau **volume table of contents**”. Perangkat direktori (atau direktori) menyimpan informasi seperti nama, lokasi, ukuran dan tipe untuk semua file dari partisi tersebut.



ORGANISASI SISTEM FILE



Informasi yang terdapat pada direktori adalah

- Nama
- Tipe
- Alamat
- Panjang saat ini
- Panjang maksimum
- Tanggal akses terakhir
- Tanggal perubahan terakhir
- ID pemilik
- Informasi proteksi

Beberapa operasi yang dibentuk pada direktori adalah :

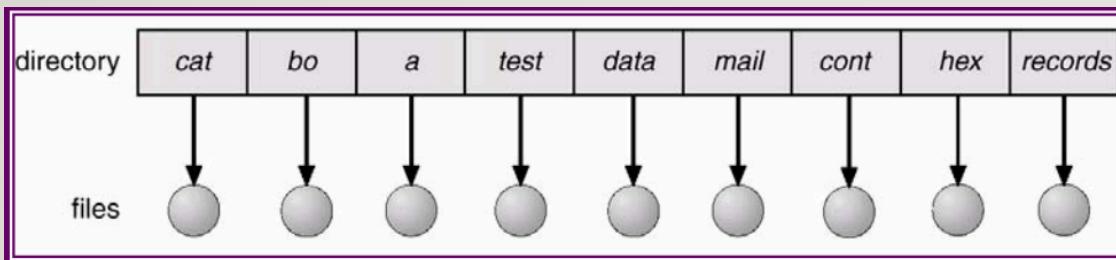
- Mencari file (search)
- ~~Membuat file (create)~~
- Menghapus file (delete)
- Mendaftar suatu direktori (list)
- Mengubah nama file (rename)
- Melintasi sistem file (traverse)

**ORGANISASI FILE DAN DIREKTORI DISARANKAN YANG SEEFISIEN MUNGKIN
SEHINGGA DAPAT MENEMPATKAN FILE DENGAN CEPAT. SELAIN ITU DALAM
PENAMAAN FILE DAN DIREKTORI HARUS NYAMAN UNTUK USER**



DIREKTORI SATU LEVEL

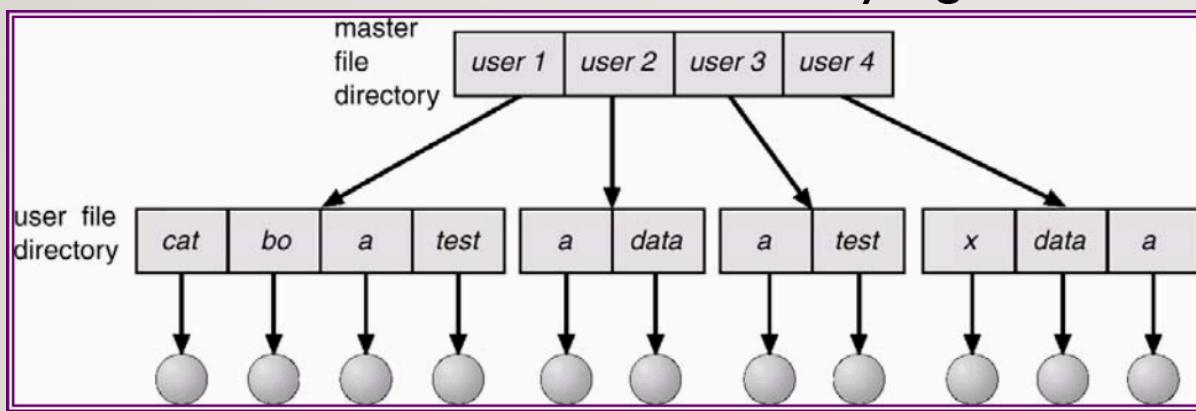
Direktori ini hanya terdiri dari satu direktori untuk setiap user, Pada direktori jenis ini terjadi permasalahan penamaan dan pengelompokan berdasarkan user.



DIREKTORI DUA LEVEL

Direktori ini terdiri dari dua level yang memisahkan direktori untuk setiap user. Setiap file diberi nama path, dapat mempunyai nama file yang sama untuk user yang berbeda,

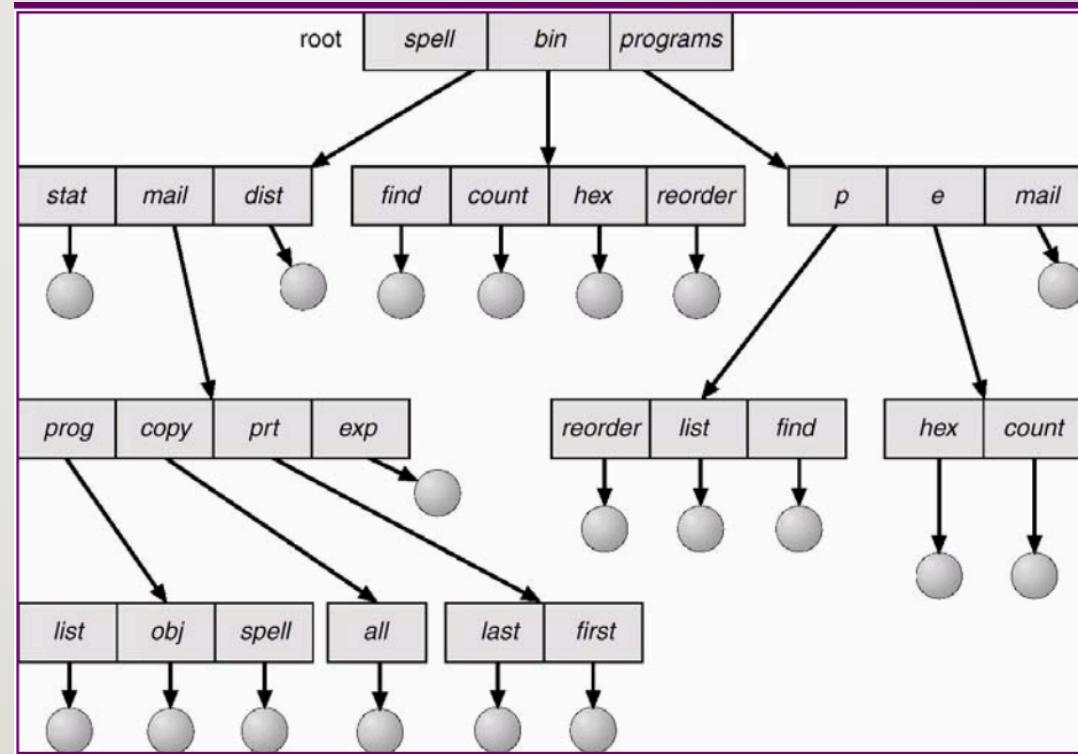
atas pencarian, tetapi belum sejalan.



DIREKTORI

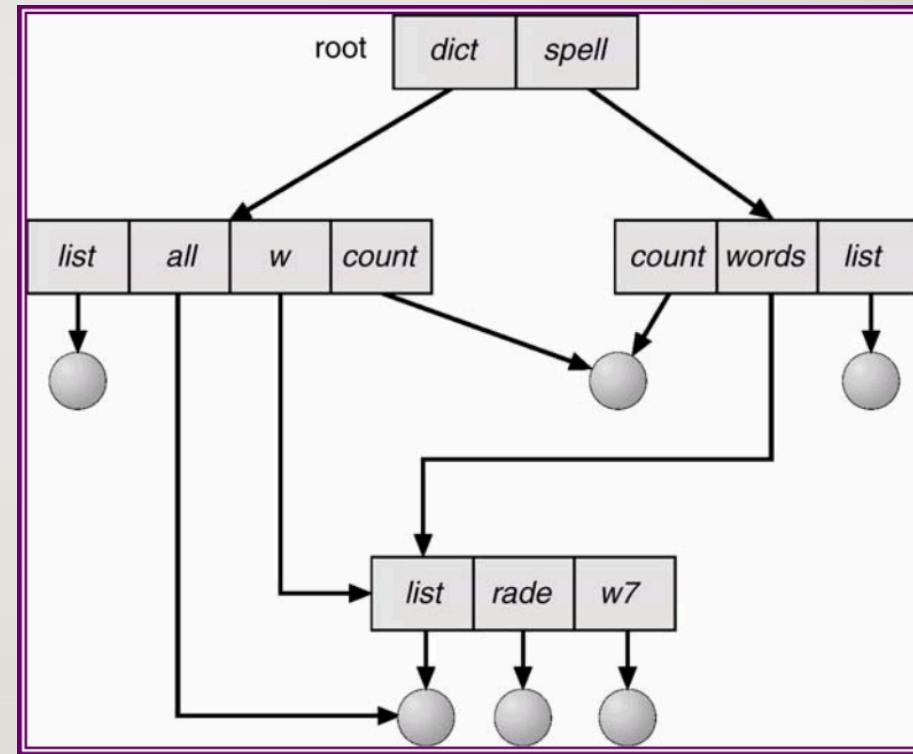
BERSTRUKTUR POHON DIREKTORI

- berstruktur pohon merupakan struktur direktori yang biasa digunakan. Pohon mempunyai direktori root. Setiap file pada sistem mempunyai nama path yang unik
- Pada direktori ini pencarian file dan direktori lebih efisien, mengelompokkan file dan dapat mengakses direktori dan sub direktori.



DIREKTORI ACYCLIC

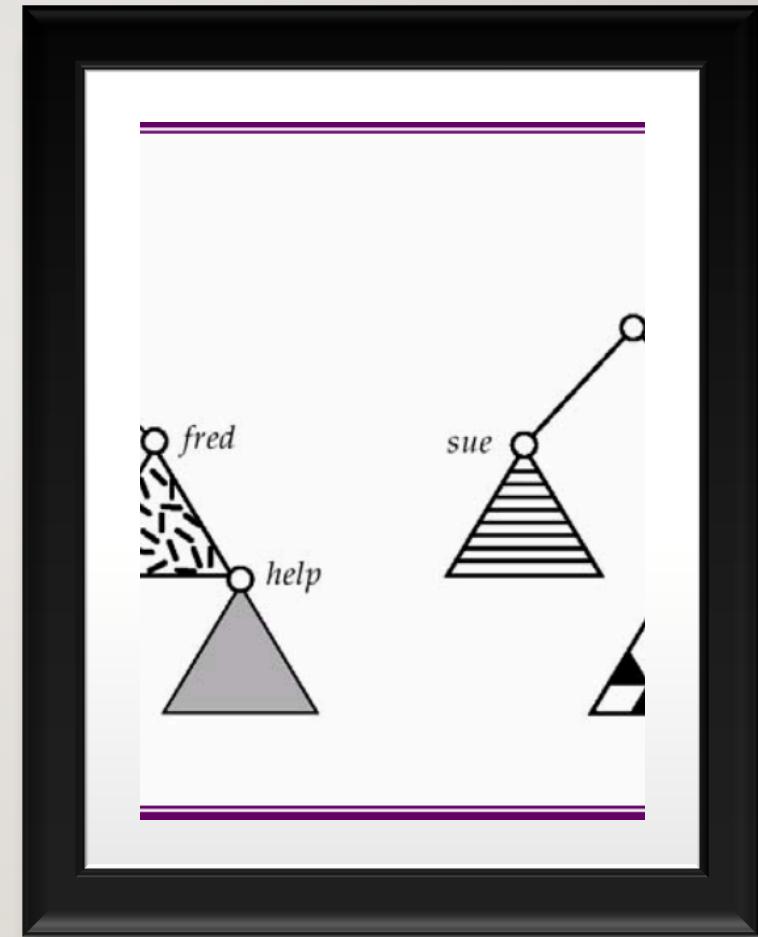
Struktur ~~graph~~ melarang menggunakan bersama-sama file dan direktori. Pada direktori acyclic graph memungkinkan direktori mempunyai subdirektori dan file yang digunakan bersama-sama. File dan subdirektori yang sama mungkin berada pada duadirektori yang berbeda



FILE SYSTEM MOUNTING

Suatu sistem file harus di-mount
sebelum diakses.

- (a) Sistem eksis
- (b) (b) partisi yang tidak di-mount



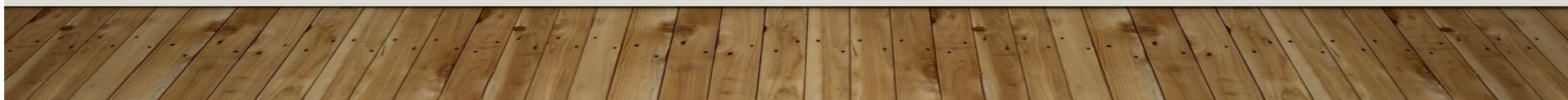
PROTEKSI

- Informasi yang disimpan dalam system komputer harus diproteksi dari kerusakan fisik (reliability) dan akses yang tidak benar (protection).
- Reliability biasanya dilakukan dengan duplikasi copy dari file.
- Beberapa sistem komputer mempunyai sistem yang secara otomatis (atau melalui intervensi operator komputer) menduplikasi file ke tape secara regular dari sistem file yang secara tiba-tiba dihapus.



TIPE AKSES

- Mekanisme proteksi dengan tipe akses file terbatas yang dapat dibuat.
- Akses diperbolehkan atau tidak tergantung beberapa faktor, satu diantaranya permintaan tipe akses.
- Beberapa operasi yang disediakan :
 - Membaca dari file (read)
 - Menulis ke file (write)
 - Menjalankan file (execute)
 - Menambah isi file (append)
 - Menghapus file (delete)
 - Melihat nama dan atribut file (list)



ACCESS LIST DAN GROUP

- Pendekatan permasalahan proteksi yang sering digunakan adalah dengan membuat akses secara **dependent** pada identifikasi user.
- Bila user meminta akses ke file khusus, sistem operasi memeriksa access list. Jika user tersebut **terdaftar**, akses diijinkan, sebaliknya terjadi **protection violation** dan dilarang mengakses file.
- Masalah pokok dengan access list adalah ukuran. Jika ingin mengijinkan user membaca file, **harus didaftarkan semua user dengan akses read**.
- Masalah ini dipecahkan dengan melakukan pemetaan terhadap access list. Beberapa system memperkenalkan tiga klasifikasi user :
 - Owner. User yang membuat file
 - Group. Kumpulan user yang menggunakan file bersama-sama dan memerlukan akses yang sama
 - Universe. Semua user lain dalam system.



CONTOH PROTEKSI : UNIX

Pada sistem UNIX, proteksi direktori ditangani sama dengan proteksi file,misalnya, diasosiasikan dengan setiap subdirektory menggunakan owner, group dan universe (others) sebagai 3 bit RWX

-rw-rw-r--	1	pbg	staff	31200	Sep	3	08:30	intro.ps
drwx-----	5	pbg	staff	512	Jul	8	09:33	private/
drwxrwxr-x	2	pbg	staff	512	Jul	8	09:35	doc/
drwxrwx---	2	pbg	student	512	Aug	3	14:13	student-proj/
-rw-r—r--	1	pbg	staff	9423	Feb	24	1993	program.c
-rwxr-xr-x	1	pbg	staff	20471	Feb	24	1993	program
drwx—x--x	4	pbg	faculty	512	Jul	31	10:31	lib/
drwx-----	3	pbg	staff	1024	Aug	29	06:52	mail/
drwxrwxrwx	3	pbg	staff	512	Jul	8	09:35	test/

PERAN IJIN AKSES

Pada File

- R (Read) Ijin untuk membaca
- W (Write) Ijin untuk mengubah / membuat
- X (Execute) Ijin untuk menjalankan program

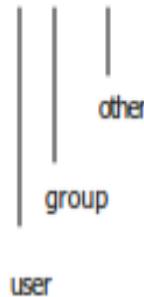
Pada Direktori

- R (Read) Ijin untuk membaca daftar file dalam direktori
- W (Write) Ijin untuk mengubah/membuat file di direktori
- X (Execute) Ijin untuk masuk ke direktori (cd)



PEMILIK FILE/DIREKTORI

```
-rwxrwxrwx 1 student test 1639 Oct 31 20:19 file
```



Format lain dari chmod adalah menggunakan bilangan octal sebagai berikut

r	w	x
4	2	1
= 7		

Format untuk mengubah ijin akses

```
chmod [ugoa] [= + -] [rwx] File(s)  
chmod [ugoa] [= + -] [rwx] Dir(s)
```

dimana u = user (pemilik)
 g = group (kelompok)
 o = others (lainnya)
 a = all