

**BUKU PETUNJUK (MODUL) PRAKTIKUM**

**SISTEM PENGOPERASIAN**

**(MII 2612)**



**LABORATORIUM KOMPUTER DASAR  
DEPARTEMEN ILMU KOMPUTER DAN  
ELEKTRONIKA**

**Fakultas Matematika dan Ilmu Pengetahuan Alam**

**Universitas Gadjah Mada**

**Yogyakarta**

**2017**

## **TATA TERTIB PRAKTIKAN**

1. Masuk sesuai jadwal/waktu yang telah ditentukan/disepakati bersama, terlambat lebih dari 15 menit dipersilahkan tidak memasuki ruang LABORATORIUM KOMPUTER DASAR.
2. Praktikan yang tidak mengikuti praktikum lebih dari 3 kali tanpa keterangan resmi, tidak berhak mengikuti responsi/ujian.
3. Tidak diadakan ujian susulan baik ujian tengah semester maupun akhir semester, kecuali atas persetujuan dosen/pengampu mata kuliah bersangkutan.
4. Memakai Pakaian rapi dan sopan:
  - € **Pria** :
    - Kemeja Lengan panjang atau pendek
    - Celana panjang Rapi
    - Bersepatu
    - Tidak boleh memakai T-shirt tanpa krah atau tanpa lengan
  - € **Wanita** :
    - Kemeja Lengan panjang/pendek (tidak ketat dan atau transparan)
    - Rok atau Celana panjang (tidak ketat dan atau transparan)
    - Bersepatu
    - Tidak boleh memakai T-shirt tanpa krah atau tanpa lengan
5. Mahasiswa tidak diperbolehkan merokok, makan dan minum pada saat kuliah praktikum.
6. Tas dan perlengkapan lain harus diletakkan pada tempat yang telah disediakan (hanya diperbolehkan membawa barang berharga, Ex: Handphone, dompet dan alat yang diperlukan untuk praktikum)
7. Barang berharga milik peserta kuliah praktikum menjadi tanggung jawab sendiri (laboran tidak bertanggungjawab atas kehilangan barang tersebut).
8. Dering HP harus dimatikan (silent) pada saat kuliah praktikum.
9. Selesai Praktikum, Komputer dimatikan dan kursi dirapikan kembali.
10. Mahasiswa diwajibkan menjaga kebersihan dan ketertiban serta ketenangan belajar.
11. Mahasiswa tidak diperbolehkan menggunakan komputer untuk bermain games.
12. Mahasiswa tidak diperkenankan men-install program/software tanpa petugas lab.
13. Mahasiswa tidak diperkenankan memindah posisi hardware (mouse, keyboard, monitor, CPU)
14. Mahasiswa tidak diperbolehkan membawa atau mengambil (secara sengaja atau tidak sengaja) perlengkapan praktikum yang ada di Laboratorium komputer).
15. Mahasiswa wajib menjaga keutuhan semua peralatan yang ada di Laboratorium Komputer serta tidak diperbolehkan memakai komputer Pengajar/Instruktur.
16. Mengisi daftar hadir dan mencatat nomor kursi/komputer yang digunakan selama praktikum berlangsung ke dalam form daftar hadir yang telah disediakan.
17. Melaporkan keadaan komputer dan atau peralatan yang digunakan (yang rusak/tidak berfungsi) sebelum, sesaat dan atau sesudah penggunaan ke Pengajar/Instruktur atau kepada laboran.

18. Praktikan wajib mematikan Komputer yang telah selesai digunakan dan merapikan kembali kursi, meja dan perlengkapan pendukung praktikum lainnya setelah praktikum selesai dilaksanakan/berakhir.
19. Laboran berhak mencatat, memberikan sanksi atau melakukan tindakan seperlunya terhadap praktikan yang melanggar tata tertib.

LABORATORIUM KOMPUTER DASAR

DIKE F.MIPA UGM

## **KATA PENGANTAR**

Assalamu'alaikum Wr. Wb.

Alhamdulillah, atas berkat rahmat Allah Yang Maha Kuasa dengan didorongkan oleh keinginan luhur memperluas wawasan dalam pengembangan pengetahuan tentang Sistem Pengoperasian, maka buku petunjuk (modul) praktikum ini disusun/dibuat.

Buku petunjuk (modul) Praktikum Sistem Pengoperasian ini dibuat untuk membantu jalannya Praktikum Sistem Pengoperasian prodi S1 Ilmu Komputer. Buku petunjuk (modul) ini dibuat sedemikian rupa sehingga dapat dengan mudah dipahami dan dipelajari oleh mereka yang belum pernah mengenal Sistem Pengoperasian sekalipun, dan bagi mereka yang pernah mengenal membaca buku ini akan menyegarkan ingatan.

Terima kasih kami ucapkan kepada semua pihak yang telah membantu dan mendukung pembuatan buku ini.

Wassalamu'alaikum Wr. Wb.

LABORATORIUM KOMPUTER DASAR  
DIKE F.MIPA UGM

## Daftar Isi

TATA TERTIB PRAKTIKAN.....	2
KATA PENGANTAR.....	iii
VMWare dan Windows.....	1
TUJUAN PRAKTIKUM .....	1
VIRTUAL MACHINE .....	1
WINDOWS 7 .....	1
INSTALASI SISTEM OPERASI PADA VMWARE .....	1
INSTALASI SISTEM OPERASI WINDOWS 7 .....	5
DYNAMIC LINK LIBRARIES.....	8
TUGAS.....	9
INSTALASI DAN KONFIGURASI UBUNTU .....	10
TUJUAN PRAKTIKUM .....	10
INSTALASI UBUNTU .....	10
DISTRO LINUX.....	15
TUGAS.....	17
Computer Management, Task Manager, dan Control Panel Windows 7 .....	18
TUJUAN PRAKTIKUM .....	18
COMPUTER MANAGEMENT .....	18
TASK SCHEDULER.....	18
MEMBUAT TASK .....	19
EVENT VIEWER .....	21
SHARED FOLDERS .....	22
LOCAL USERS AND GROUPS .....	22
PERFORMACE .....	23
DEVICE MANAGER.....	24
DISK MANAGEMENT.....	25
SERVICES.....	25
TASK MANAGER.....	26
CONTROL PANEL.....	27
TUGAS.....	27
PROCESS SCHEDULING DAN PEMROGRAMAN MULTITHREAD .....	28
TUJUAN PRAKTIKUM .....	28
PROCESS SCHEDULING .....	28
Pemrograman Multi-thread .....	34
Tugas .....	37
PERINTAH-PERINTAH COMMAND RUN, PROMPT, DAN BERBAGI RESOURCE DI WINDOWS 7.....	38
TUJUAN PRAKTIKUM .....	38
RUN .....	38
SHARING .....	41

Tugas .....	20
File System Management dan Disk Scheduling (OSSIM) .....	21
TUJUAN PRAKTIKUM .....	21
FILE SYSTEM MANAGEMENT.....	21
Disk scheduling.....	29
Tugas .....	34
Group Policy Editor dan Registry Editor.....	35
TUJUAN PRAKTIKUM .....	35
GROUP POLICY EDITOR .....	35
REGISTRY EDITOR .....	37
TUGAS.....	39
TERMINAL LINUX UBUNTU DAN SECURITY.....	40
TUJUAN PRAKTIKUM .....	40
GUI.....	40
STRUKTUR SISTEM FILE DI UBUNTU.....	41
COMMAND LINE LINUX .....	41
Tugas .....	50
MANAJEMEN MEMORY .....	51
TUJUAN PRAKTIKUM .....	51
SIMULASI .....	51
Tampilan awal .....	52
Tugas .....	59
MENGENAL SISTEM OPERASI MACHINTOS DAN ANDROID (OPTIONAL).....	60
TUJUAN PRAKTIKUM .....	60
MENGENAL SISTEM OPERASI MACINTOSH (MAC).....	60
MENUBAR.....	60
DOCK .....	61
Cara menjalankan aplikasi.....	61
Tugas .....	64

# PERTEMUAN 1

## VMWare dan Windows

### TUJUAN PRAKTIKUM

- Praktikan dapat mengetahui cara menginstallasi VMWare dan system operasi
- Praktikan dapat memahami fungsi system operasi Windows 7
- Praktikan dapat menguasai dan mempraktikkan cara menginstall Windows 7
- Praktikan dapat memahami fungsi dynamic link libraries (.dll)

### VIRTUAL MACHINE

Virtual Machine (Indonesia : mesin virtual) pada mulanya didefinisikan oleh Gerard J. Popek dan Robert P. Goldberg pada tahun 1974 sebagai *sebuah duplikat yang efisien dan terisolasi dari suatu mesin asli*. Pada masa sekarang ini, virtual machine dapat mensimulasikan perangkat keras walaupun tidak ada perangkat keras aslinya sama sekali.

Virtual machine terdiri dari dua kategori besar, dipisahkan menurut cara penggunaan dan tingkat keterhubungannya dengan mesin-mesin aslinya. Sebuah System Virtual Machine adalah perangkat yang berupa platform sistem yang lengkap dan dapat menjalankan sebuah sistem operasi yang lengkap. Sebaliknya, Process Virtual Machine didesain untuk menjalankan sebuah program komputer tertentu (tunggal), yang berarti mesin virtual ini mendukung proses tertentu juga.

Disini akan dikenalkan bagaimana cara instalasi dan menggunakan suatu System Virtual Machine. Yaitu VMware Workstations 12 Player (VMware). VMware adalah salah satu System Virtual Machine yang dapat menjalankan suatu sistem operasi secara lengkap.

### WINDOWS 7

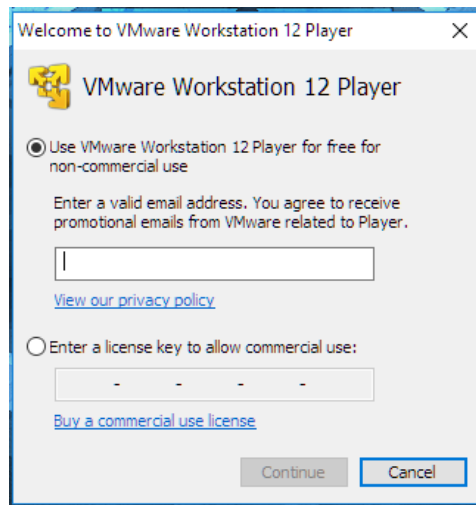
Windows 7 adalah suatu sistem operasi yang banyak dipakai sampai saat ini karena kemudahan dalam pemakaiannya. Untuk melakukan penginstalan Windows 7 diperlukan ketelitian dan kesabaran dalam prosesnya karena memerlukan waktu yang lumayan lama. Ada beberapa jenis Windows 7 diantaranya Windows 7 Starter, Home Basic, Home Premium, Professional, Ultimate, dan Enterprise.

Penggunaan dan fitur- fitur bawaan dari Windows 7 cukup lengkap dan mudah dioperasikan, hal ini membuat Windows 7 masih tetap digemari dan masih layak konsumsi bagi masyarakat banyak. Walaupun sudah terbit versi – versi Windows terbaru seperti Windows 8 ataupun Windows 10.

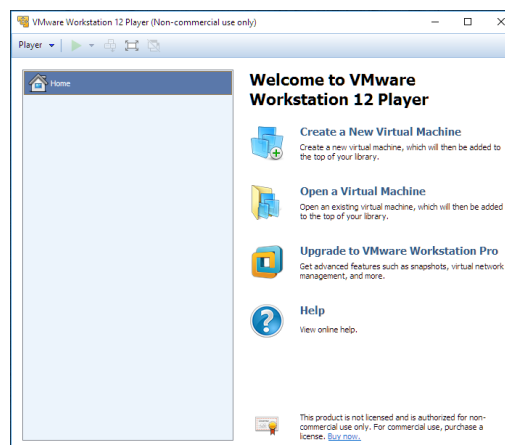
### INSTALASI SISTEM OPERASI PADA VMWARE

1. Install VMware Workstation 12 Player.

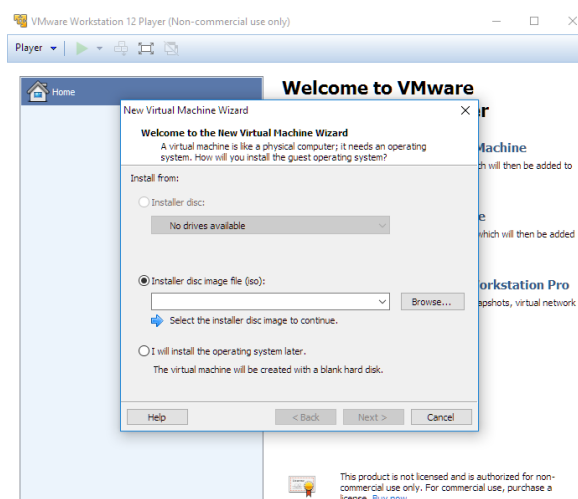
2. Pilih “Use VMware Workstation 12 Player for free for non – commercial use” dan input alamat email.



3. Pilih “Create a New Virtual Machine”.

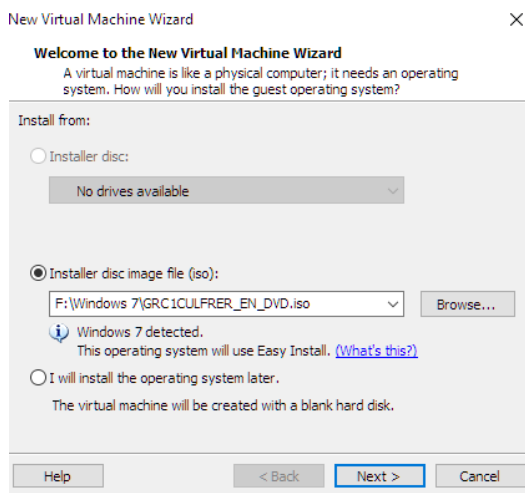


4. Akan muncul windows kecil seperti gambar di bawah. Klik Browse... dan pilih file .iso sistem operasi yang akan diinstal.

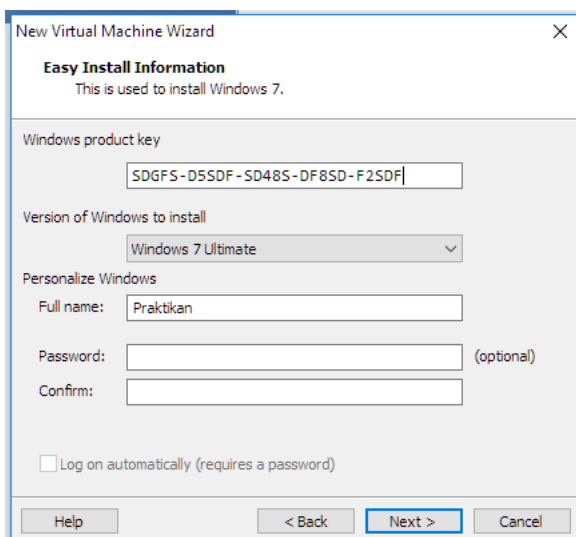




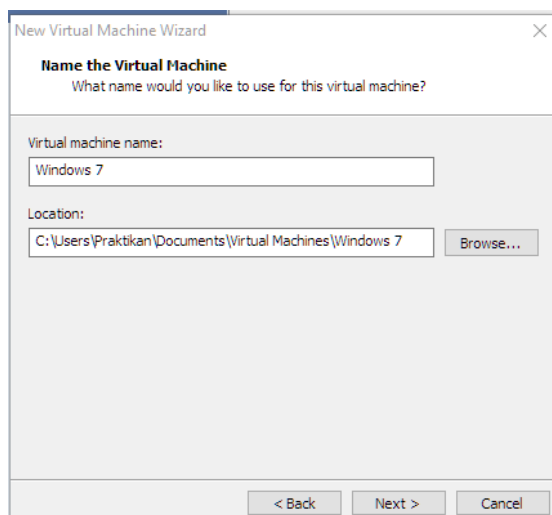
5. Jika berhasil, Sistem Operasi yang akan diinstal akan langsung terdeteksi.



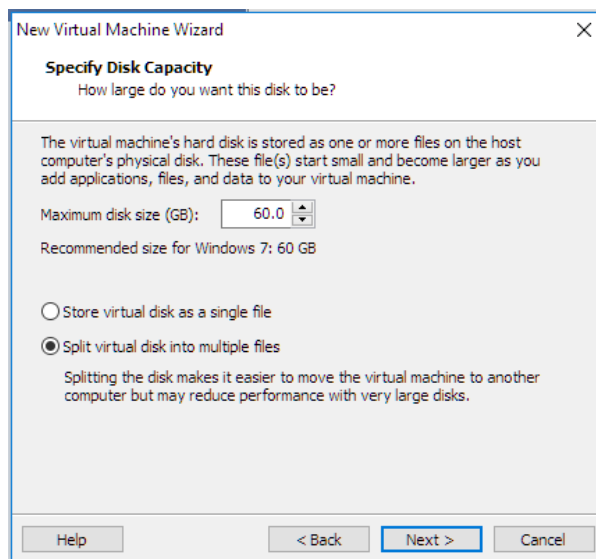
6. Tuliskan serial number Windows dan username yang diinginkan. Dan pilih juga versi windows yang sesuai.



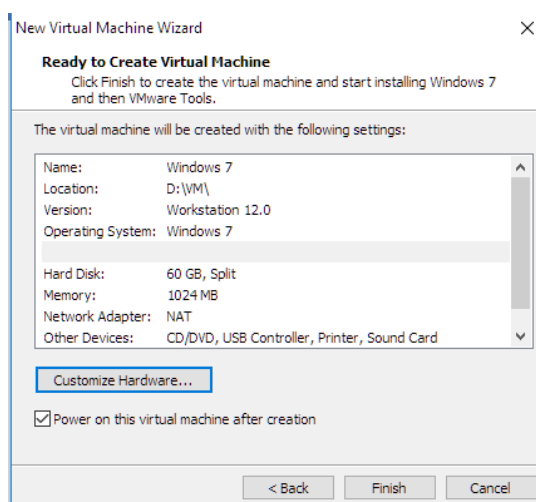
7. Tuliskan Nama Virtual Machine dan lokasi dimana sistem operasi akan di-install.



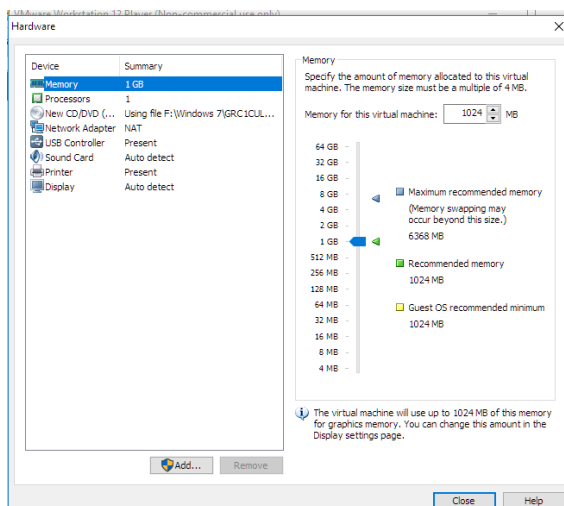
## 8. Sesuaikan ukuran Hard Disk untuk sistem operasi.



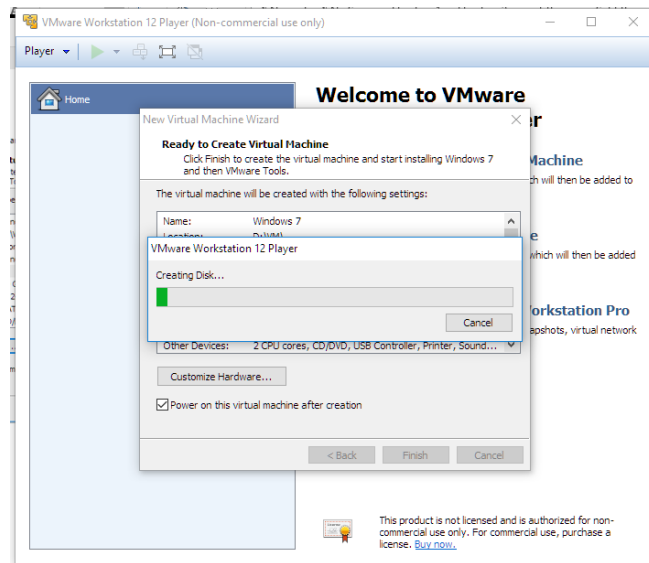
## 9. Dapat dilakukan perubahan hardware untuk virtual machine dengan men-klik “Customize Hardware...” atau jika hardware dirasa sudah sesuai.



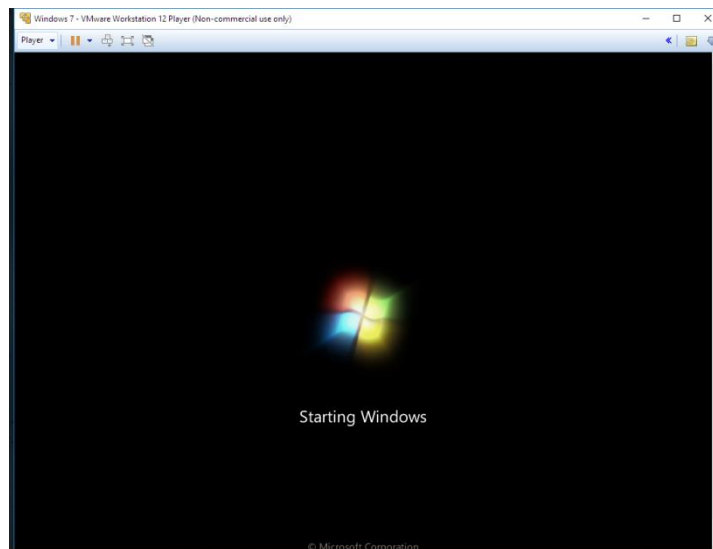
## 10. Sesuaikan hardware dengan keinginan.



11. Tunggu proses instalasi selesai.

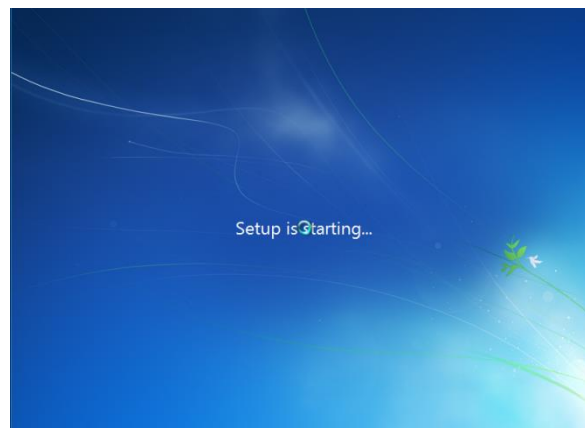


12. Sistem operasi siap dipakai.

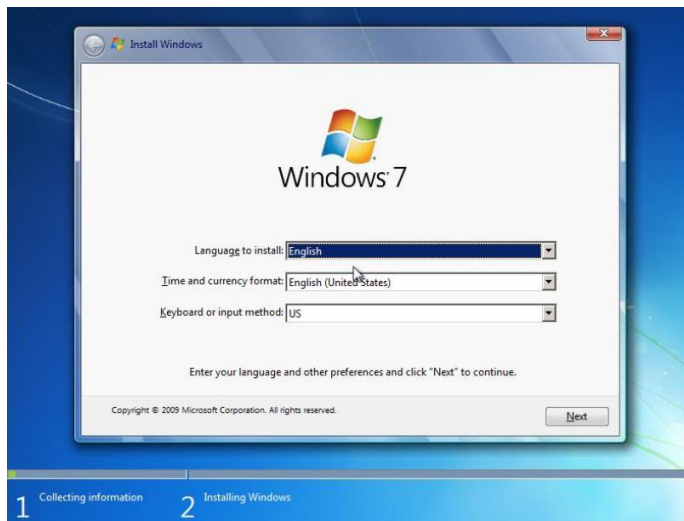


## INSTALASI SISTEM OPERASI WINDOWS 7

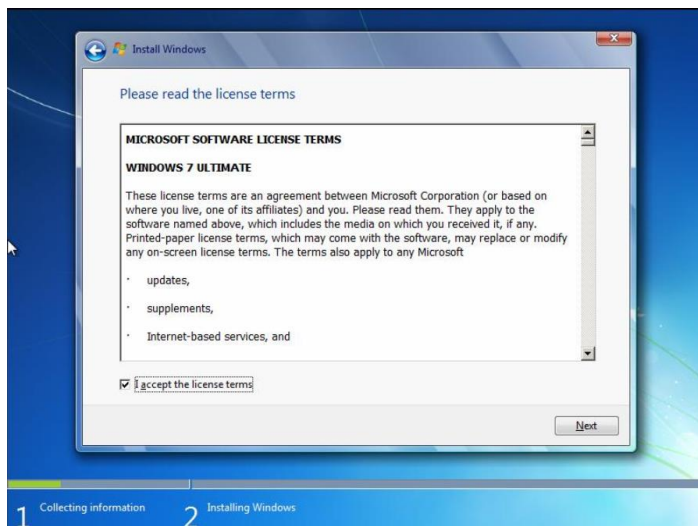
1. Mulai instalasi windows 7, terserah dari CD, DVD, atau Flashdisk.



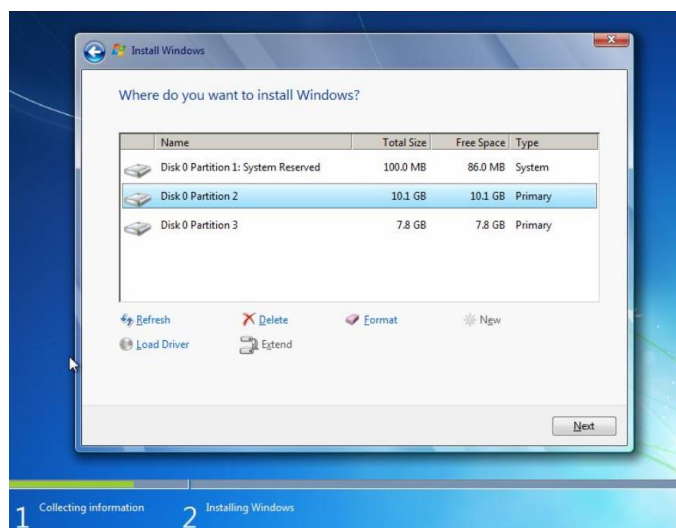
2. Kemudian pilih bahasa yang akan dipakai.



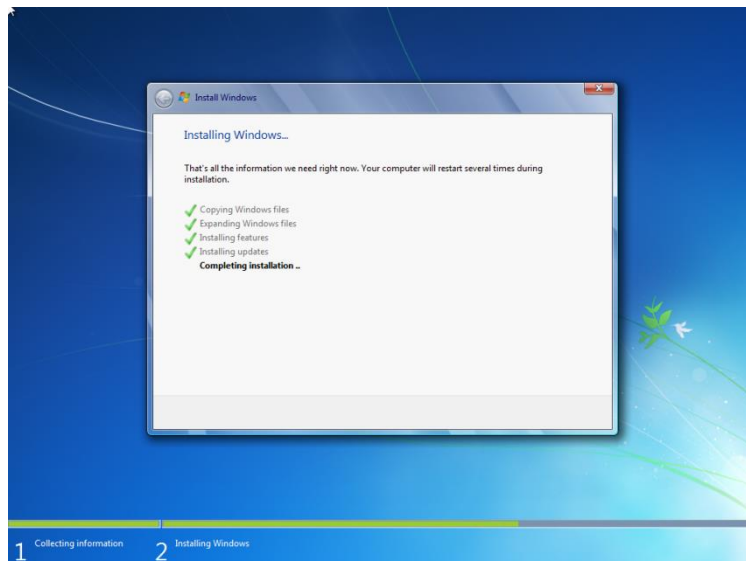
3. Masukkan serial number windows.



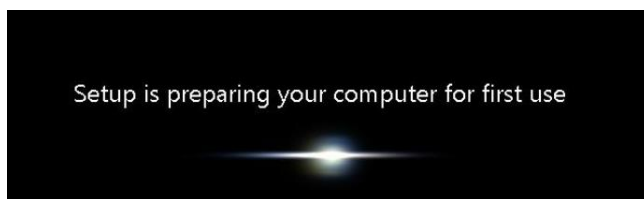
4. Buat partisi Harddisk dan tentukan drive mana, sistem operasi akan di-instal.



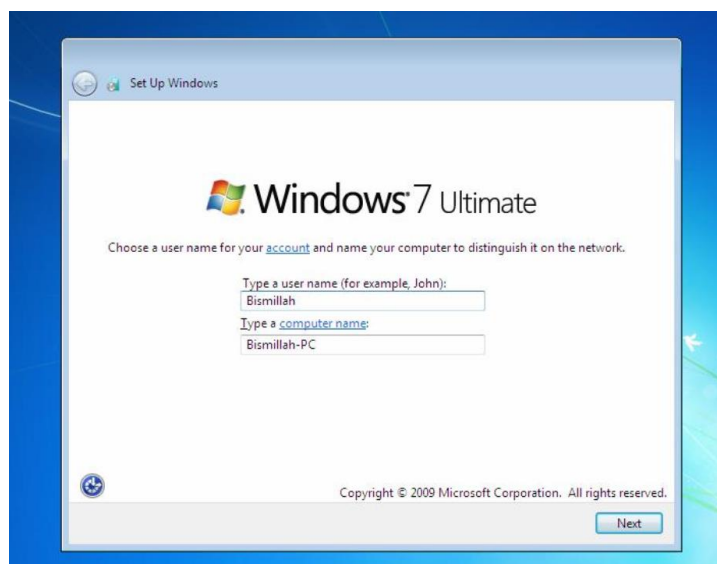
5. Tunggu Proses instalasi hingga selesai. Komputer mungkin akan melakukan restart beberapa kali.



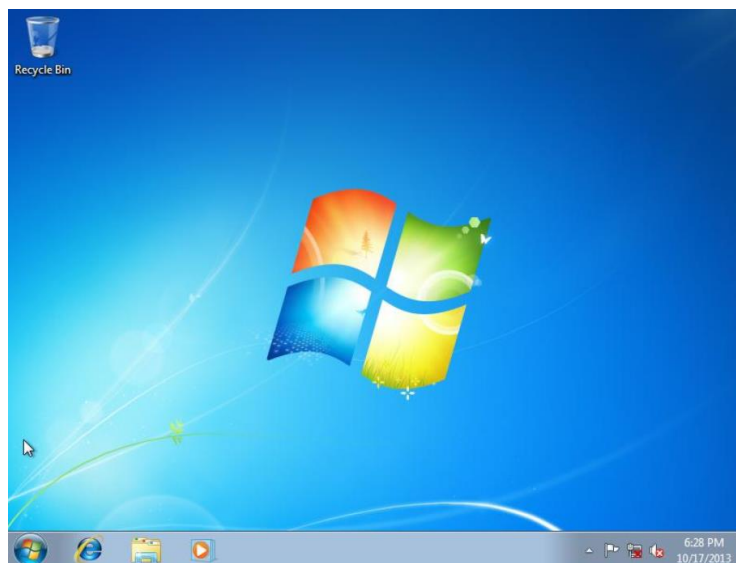
6. Jika melihat proses ini, berarti Windows akan melakukan setup untuk penggunaan pertama.



7. Isikan nama user dan nama komputer.



8. Windows 7 siap digunakan.



## DYNAMIC LINK LIBRARIES

Semua proses dari Windows 7, disimpan dalam bentuk dynamic link libraries (.dll). Segala fungsi di Windows dijalankan dalam fungsi – fungsi kecil dalam file dll. File dll sendiri terdapat di C:\Windows\System32 untuk sistem operasi 32bit atau C:\Windows\SysWOW64 untuk sistem operasi 64bit.

File .dll berisi kode dan data yang dapat dipakai lebih dari satu proses pada sat yang sama. Sebagai contoh, Comdlg32.dll berfungsi untuk membuka dialog box suatu fungsi. Jadi setiap program dapat menggunakan fungsi dll ini. Fungsionalitas yagn tersimpan dalam dll ini adalah implementasi dialog box.

Untuk Windows 7 sendiri, terdapt lebih dari 3000 file dll yang siap dipakai untuk melakukan proses-nya masing – masing. Disini terdapat beberapa file dll yang penting dan fungsinya:

<i>File name</i>	<b>Fungsi</b>
<i>COMDLG32</i>	Mengontrol dialog box
<i>GDI32</i>	Berisi grafik seperti teks dan font
<i>KERNEL32</i>	Berisi manajemen memori dan proses
<i>USER32</i>	Berisi user interface dan interaksi antar windows
<i>SHELL32</i>	Berisi Windows Shell Common
<i>MSVCRT</i>	Berisi kode program agar program dapat di-run dalam Microsoft Visual C++ dengan benar
<i>NTDLL</i>	Mengontrol messages, timing, threading, dan sinkronisasi antar program di dalam kernel-mode
<i>ADVAPI32</i>	Berisi Windows 32 Base API
<i>SETUPAPI</i>	Mengontrol instalasi suatu program
<i>CRYPT32</i>	Berisi modul Certificate dan Cryptographic Messaging
<i>WMP</i>	Mengontrol Windows Media Player

<i>WMPLOC</i>	Berisi Resources untuk Windows Media Player
<i>MSVBVM60</i>	Berisi Visual Basic Virtual Machine
<i>IEFRAME</i>	Mengontrol jalannya Internet Explorer
<i>PROPSYS</i>	Berisi System Properties
<i>COMRES</i>	Berisi Resources COM+
<i>DISPLAY</i>	Menampilkan Control Panel
<i>EXPLORERFRAME</i>	Mengontrol Frame dari Windows Explorer
<i>DINPUT8</i>	Mengontrol Direct Input pada sistem operasi
<i>D3DX10</i>	Mengontrol DirectX 10 pada Windows

### TUGAS

1. Buatlah tutorial cara instalasi Windows 8 atau 10 boleh dalam bentuk laporan dengan screenshot ataupun video.
2. Buatlah tabel yang berisi minimal 20 file dll dan fungsinya selain yang sudah ada di atas.

## PERTEMUAN 2

### INSTALASI DAN KONFIGURASI UBUNTU

#### TUJUAN PRAKTIKUM

- Praktikan dapat mengetahui cara menginstallasi VMWare dan system operasi
- Praktikan dapat memahami fungsi system operasi Linux
- Praktikan dapat menguasai dan mempraktikkan cara menginstall Linux Ubuntu
- Praktikan dapat memahami perbedaan beberapa jenis distro linux

#### INSTALASI UBUNTU

1. Tampilan awal ketika akan menginstall Ubuntu versi 12.0.4 adalah seperti pada screenshot di bawah ini :

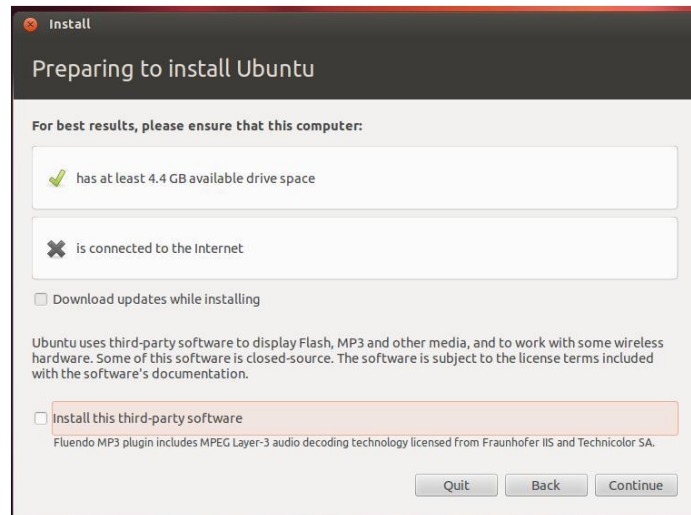


2. Tunggu (sekitar 1 menit) hingga layar berubah seperti pada gambar di bawah. Ketika tampilan di bawah ini telah muncul, pilih bahasa yang sesuai dengan keinginan anda (pemilihan bahasa akan mempengaruhi petunjuk instalasi ubuntu selanjutnya). Di sini misalkan kami memilih bahasa Inggris, lalu klik **Install Ubuntu** untuk menginstall ke dalam komputer anda.

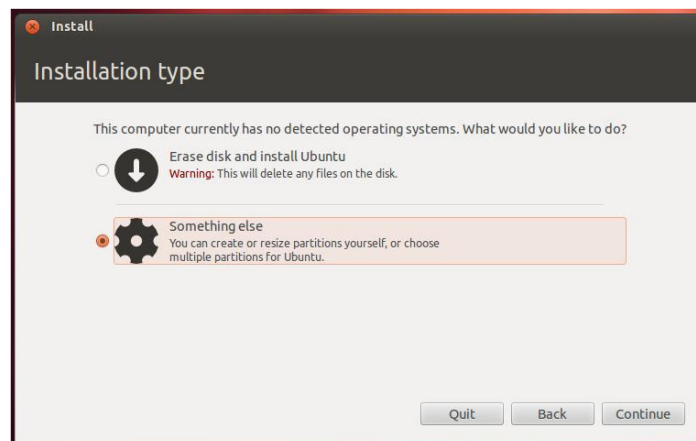




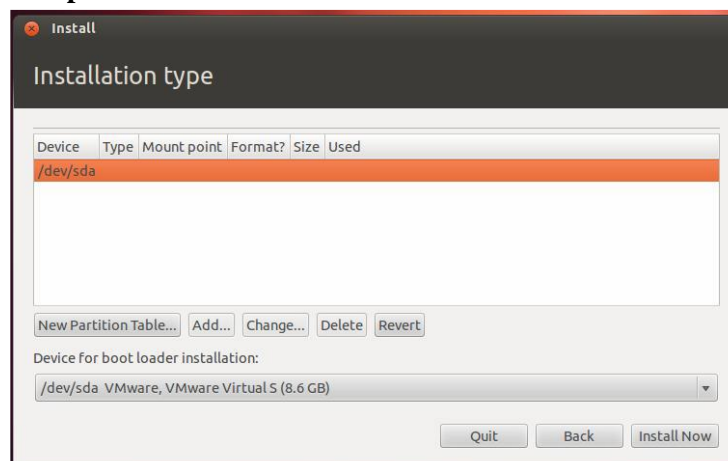
3. Ketika gambar seperti di bawah ini muncul, ada baiknya membaca terlebih dahulu kebutuhan dasar yang dibutuhkan dalam menginstall Ubuntu seperti misalnya memiliki drive space sebesar 4.4 GB. Untuk melanjutkan klik “Continue”.



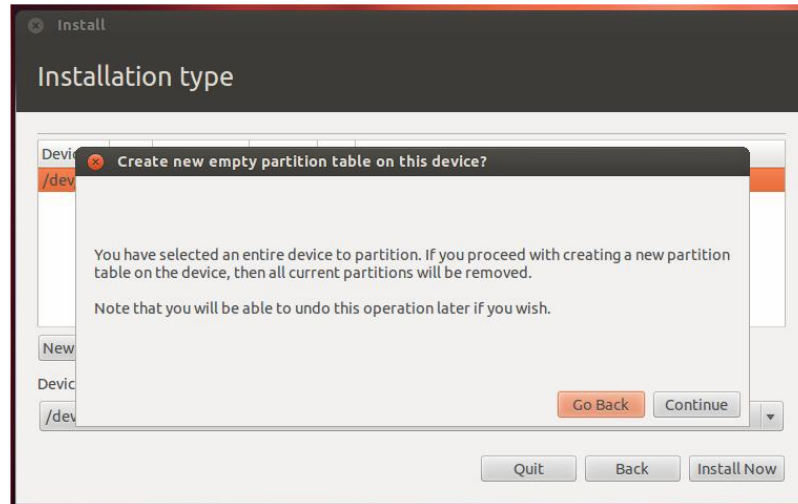
4. Selanjutnya akan muncul 2 buah opsi seperti gambar di bawah ini. Opsi **Erase disk and Install Ubuntu** memiliki dampak menghapus seluruh data yang sebelumnya tersimpan dalam HDD komputer anda lalu menginstall Ubuntu dalam keadaan HDD sudah kosong (Data anda sebelumnya sudah hilang). Untuk keamanan data sebaiknya pilih opsi **Something else** untuk melakukan setting manual mengenai apa yang akan anda lakukan terhadap komputer anda selanjutnya.



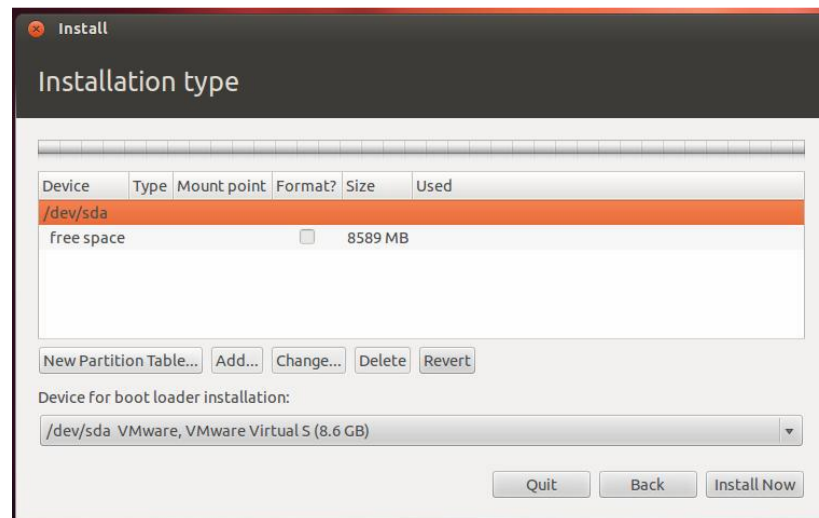
5. Pilih `/dev/sda` klik **new partition table**



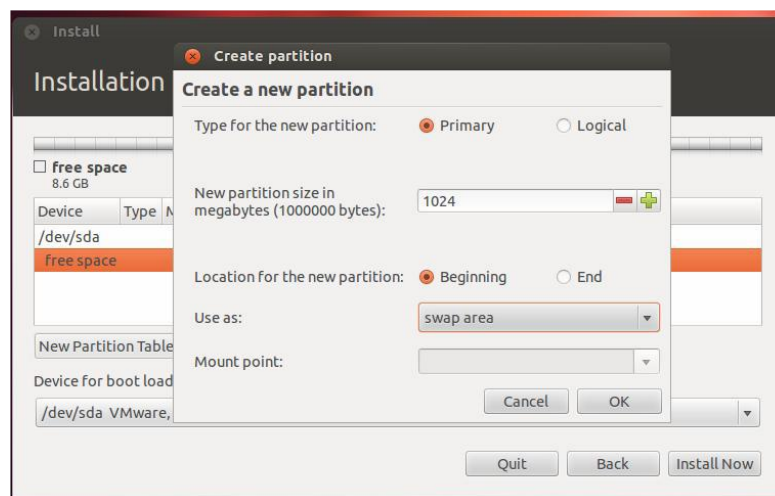
6. Ketika muncul command box seperti di bawah ini klik saja **continue**



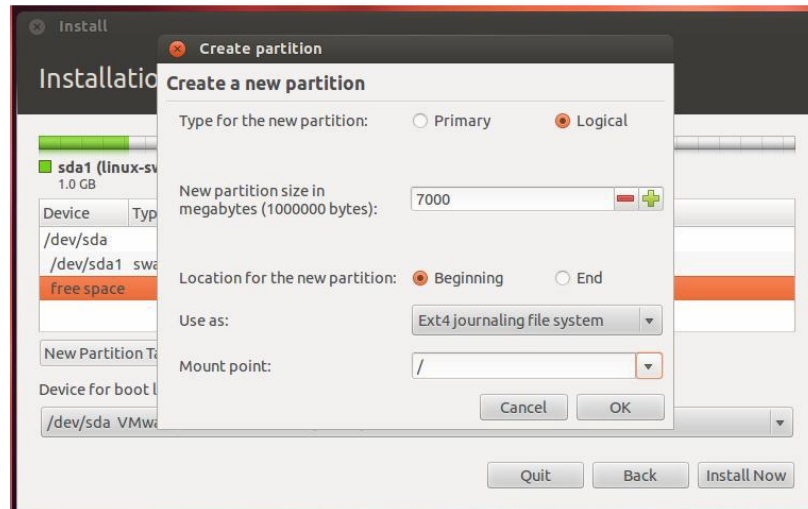
7. Pilih dan double klik **free space**



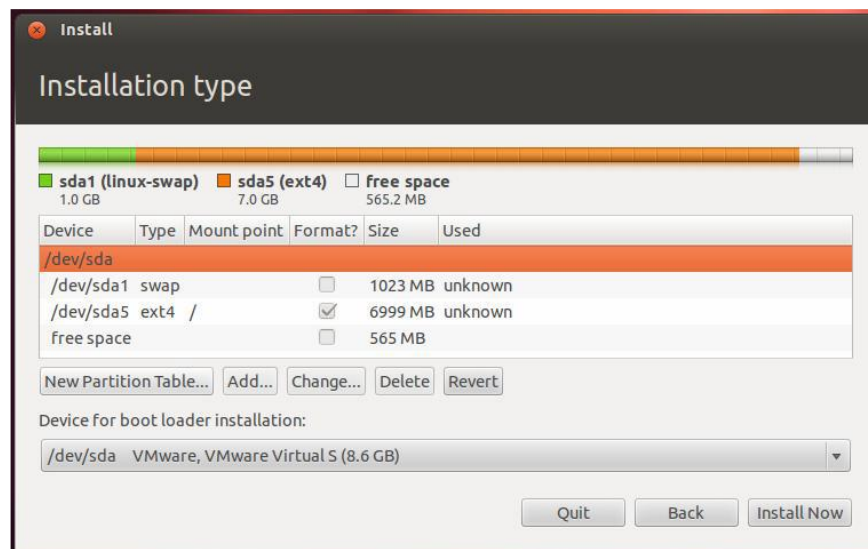
8. Edit value pada **partition size** menjadi sebesar 2x RAM anda (RAM kami sebesar 512 MB maka pada **Partition Size** kami masukkan value 1024 MB), edit kolom Use as menjadi **swap area**> klik OK



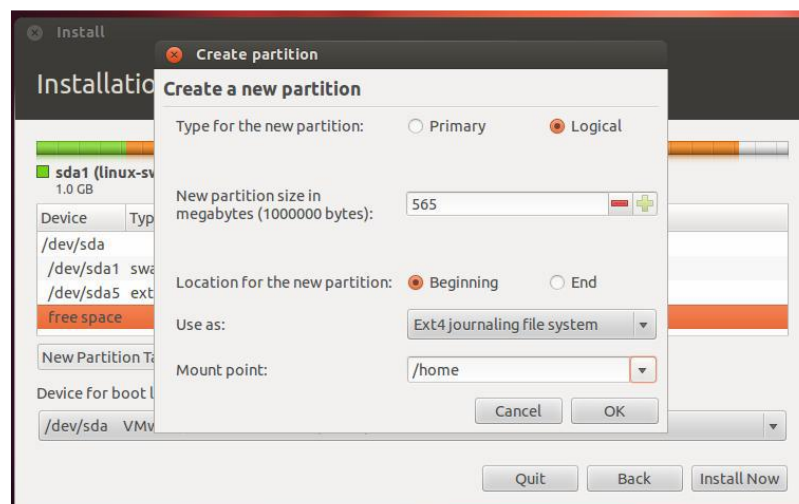
9. Selanjutnya, double klik **free space** yang tersisa, edit **partition size** menjadi 7000 mb, lalu ubah **mount point** menjadi “/”(root) > klik OK



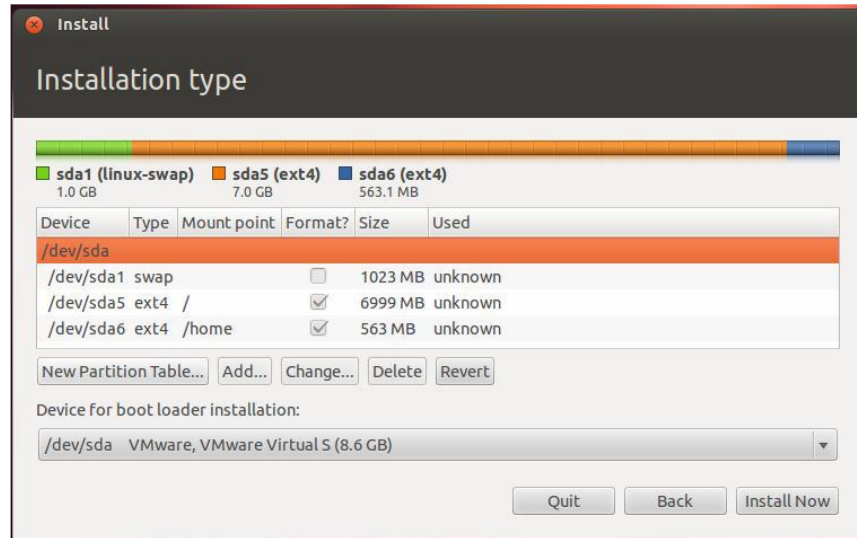
10. Pilih dan double klik **free space** sekali lagi untuk mendayagunakan sisa kapasitas HDD yang tersisa.



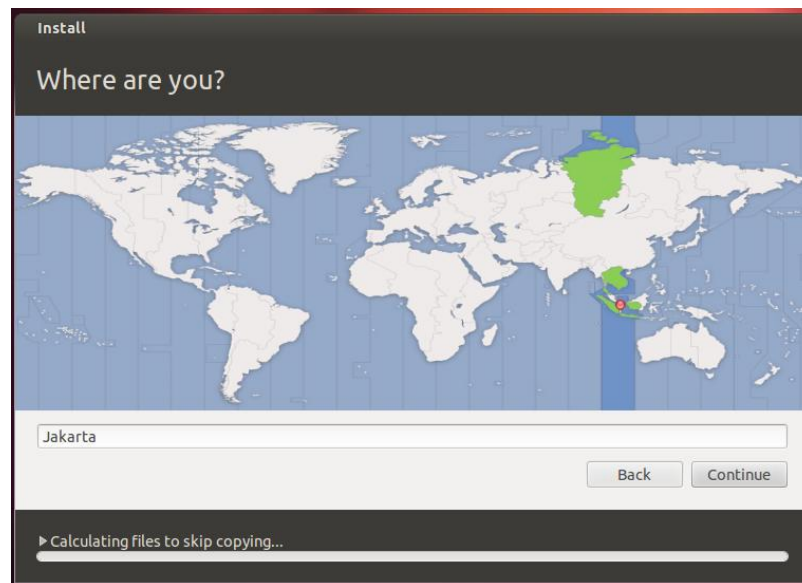
11. Ubah **mount point** menjadi “/home”



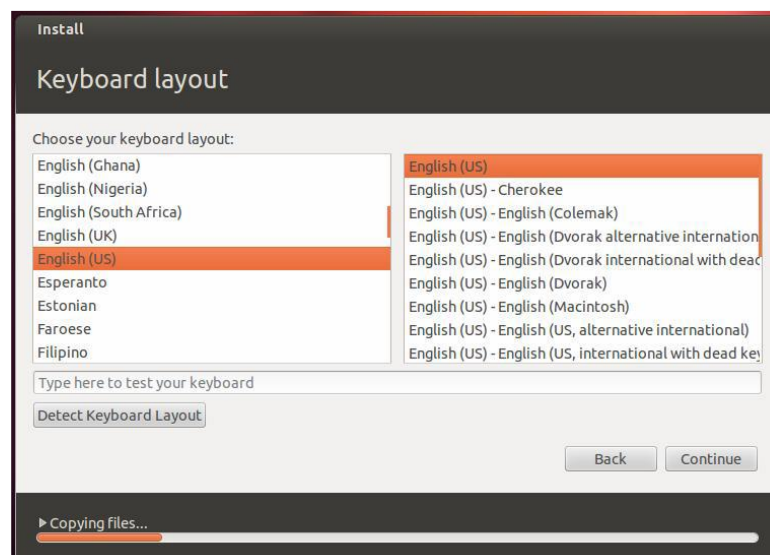
12. Setelah kira-kira sudah tidak ada sisa kapasitas HDD yang terbuang, langsung saja klik **Install Now**



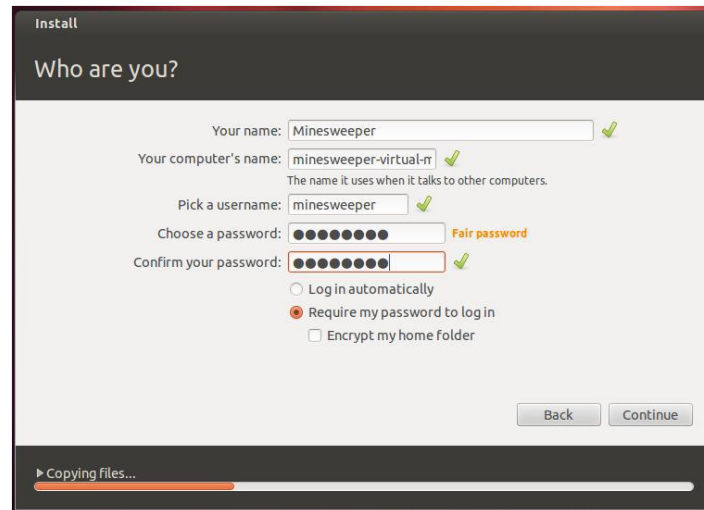
13. Pilih region sebagai acuan waktu GMT di komputer anda (kami memilih Jakarta, Indonesia sebagai contohnya) lalu klik **Continue**



14. Pada bagian Keyboard layout pilih default saja (english) lalu klik **Continue**



15. Masukkan data diri anda sebelum melakukan finalisasi instalisasi Ubuntu 12.0.4 anda, pastikan anda masukkan password juga sebagai tambahan sekuritas di komputer anda lalusebagai proses terakhir klik **Continue**



16. Sekarang proses Installasi Ubuntu 12.0.4 telah dimulai, anda tinggal menunggu sampai prosesnya berakhir dan selamat menikmati Ubuntu 12.0.4 di komputer anda!



## DISTRO LINUX

**Distro Linux** (singkatan dari **distribusi Linux**) adalah sebutan untuk sistem operasi komputer dan aplikasinya, merupakan keluarga Unix yang menggunakan kernel Linux. Distribusi Linux bisa berupa perangkat lunak bebas dan bisa juga berupa perangkat lunak komersial seperti Red Hat Enterprise, SuSE, dan lain-lain.

Ada banyak distribusi atau distro Linux yang telah muncul. Beberapa bertahan dan menjadi distro besar, bahkan sampai menghasilkan distro turunan, contohnya distro Debian GNU/Linux. Distro ini telah menghasilkan puluhan distro turunan, antara lain Ubuntu, Knoppix, Xandros, DSL, dan sebagainya.

Berikut macam-macam distro Linux:

- **Arch Linux**, merupakan distribusi jenis rolling release yang ditargetkan pada pengguna Linux yang sudah berpengalaman, Arch Linux dikelola oleh komunitas.
  - **Manjaro**, adalah distribusi turunan Arch Linux yang cepat, ramah pengguna dan berorientasi-desktop
- **Debian**, distribusi ini dikelola oleh sukarelawan di komunitas. Debian merupakan distribusi tua yang masih ada hingga saat ini dan banyak menghasilkan turunan berupa distribusi baru. Berikut ini merupakan contoh distribusi populer yang diturunkan dari Debian.
  - **Canaima**, adalah sebuah proyek sosial-teknologi terbuka. Canaima dibangun dan dikembangkan di Venezuela berdasarkan Debian.
  - **Knoppix**, distribusi pertama yang menggunakan Live CD untuk menjalankan sistem dari *removable media* tanpa harus melakukan instalasi ke hard disk.
  - **Linux Mint Debian Edition (LMDE)**, yang didasarkan langsung dari distribusi Debian.
  - **Ubuntu**, merupakan distribusi yang paling populer yang berasal dari debian, dikembangkan oleh perusahaan Canonical Ltd.
    - **Backtrack**, distribusi yang dikembangkan oleh tim Offensive Security yang dibangun khusus untuk tujuan digital forensik dan uji keamanan.
    - **Linux Mint**, distribusi Linux berbasis Debian (Linux Mint Debian Edition) dan Ubuntu.
    - **Lubuntu**, varian Ubuntu yang menggunakan antarmuka desktop LXDE (untuk komputer dengan spesifikasi rendah)
    - **Xubuntu**, varian Ubuntu yang menggunakan antarmuka desktop XFCE (untuk komputer dengan spesifikasi rendah)
    - **Kubuntu**, varian Ubuntu yang menggunakan antarmuka desktop KDE
    - **Edubuntu**, varian Ubuntu khusus untuk edukasi atau pembelajaran.
    - **BlackBuntu**, varian Ubuntu khusus untuk Penetrasi jaringan, keamanan, digital forensic.
    - **Sabily**, Varian ubuntu cocok bagi muslim(lengkap dengan software yang islami)
    - **Elementary OS**, distribusi turunan Ubuntu yang dikembangkan oleh elementary LLC. Elementary OS menggunakan antarmuka desktop Pantheon
  - **BlankOn**, merupakan distribusi yang dikembangkan di Indonesia. BlankOn menggunakan antarmuka desktop yang disebut Manokwari dan tersedia dalam beberapa bahasa daerah lokal di Indonesia.
- **Fedora**, distribusi komunitas yang disponsori oleh perusahaan Amerika, RedHat.
  - **Red Hat Enterprise Linux**, distribusi yang dikelola dan dikomersialkan oleh RedHat.
    - **CentOS**, distribusi komunitas yang dibangun dari kode sumber yang sama oleh RedHat.
    - **Oracle Enterprise Linux**, merupakan turunan dari Red Hat Enterprise Linux, dipelihara dan komersial didukung oleh Oracle.
    - **IGOS Nusantara**, dikembangkan oleh Lembaga Ilmu Pengetahuan Indonesia. Dibangun berdasarkan Red Hat Enterprise Linux.
  - **Mandriva**, merupakan distribusi komersial turunan RedHat yang populer di beberapa negara Eropa dan Brazil. Distribusi Mandriva dikelola oleh perusahaan Prancis dengan nama yang sama, Mandriva.
    - **Mageia**, merupakan distribusi komunitas yang merupakan *percabangan* dari Mandriva pada tahun 2010.

- **PCLinuxOS**, turunan dari Mandriva yang mengkhususkan kepada penggunaan komputer desktop.
- **Slackware**, merupakan distribusi pertama yang lahir pada tahun 1993 dan merupakan salah satu distribusi Linux tertua saat ini.
  - **SUSE Linux Enterprise**, dibangun berdasarkan OpenSUSE dan dikelola oleh perusahaan SUSE.
  - **OpenSUSE**, distribusi komunitas yang disponsori oleh perusahaan Jerman, SUSE.

## **TUGAS**

1. Pilih salah satu distro linux yang ada saat ini lalu tulis kelebihan dan kekurangan distro linux tersebut.
2. Buatlah tutorial untuk instalasi distro linux tersebut.

## PERTEMUAN 3

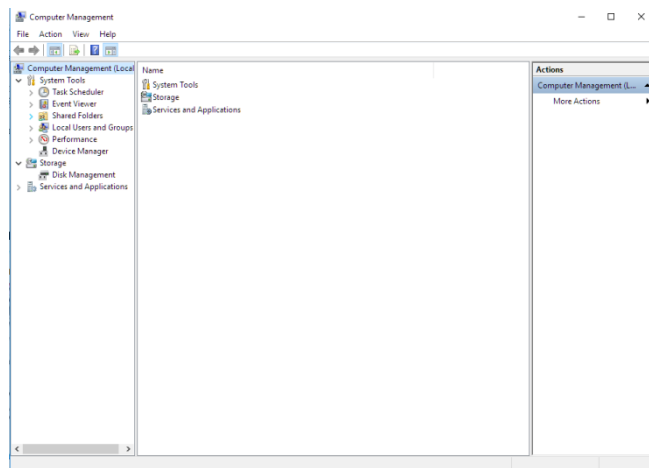
### Computer Management, Task Manager, dan Control Panel Windows 7

#### TUJUAN PRAKTIKUM

- Praktikan dapat mengatur manajemen sebuah komputer
- Praktikan dapat memahami fungsi Task Manager
- Praktikan dapat menguasai dan mengenal isi dari control panel

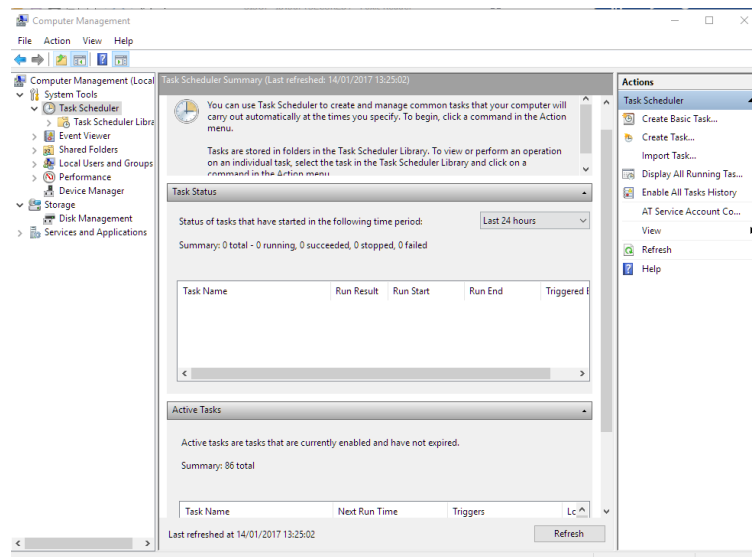
#### COMPUTER MANAGEMENT

Computer Management adalah kumpulan Administrative Tools Windows yang digunakan untuk mengelola komputer lokal ataupun remote.



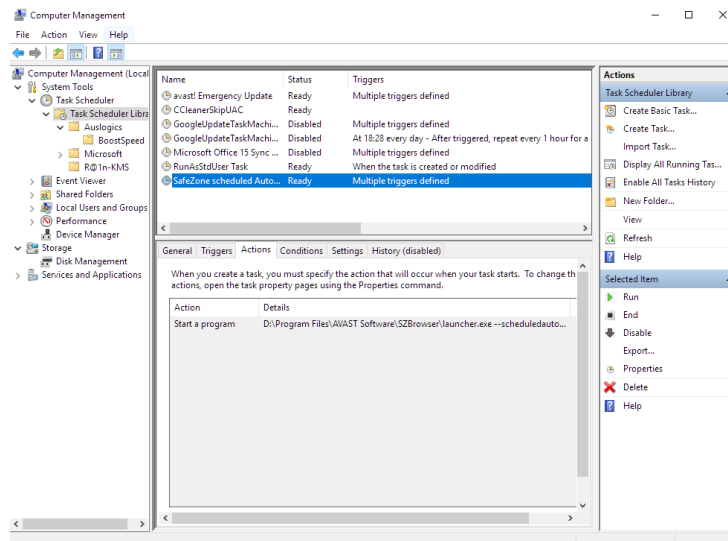
#### TASK SCHEDULER

Task Scheduler berguna untuk mengatur task – task yang sedang berjalan, akan berjalan, dan yang sudah tidak berjalan. Dapat juga membuat task – task sederhana dengan mengklik bagian “Create Basic Task...”, jika ingin membuat task yang lebih kompleks, bisa memilih “Create Task”.



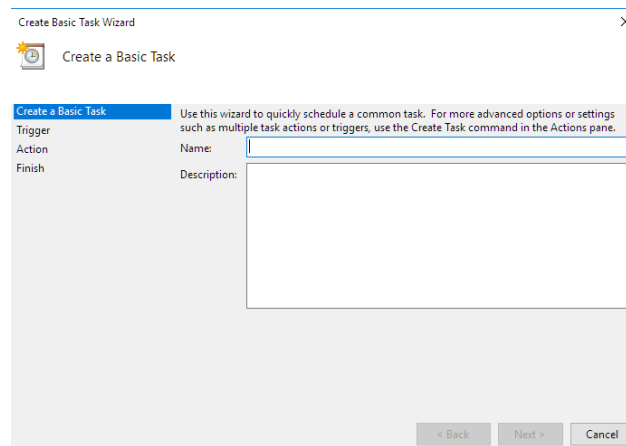


Pada Task Scheduler Library, dapat dilihat bermacam – macam Task yang sedang berjalan. Name adalah nama dari task tersebut. Status berarti kondisi task sekarang, bisa saja Ready, Disabled, atau Waiting Lalu ada Triggers, yaitu pemacu task akan dijalankan atau berhenti.

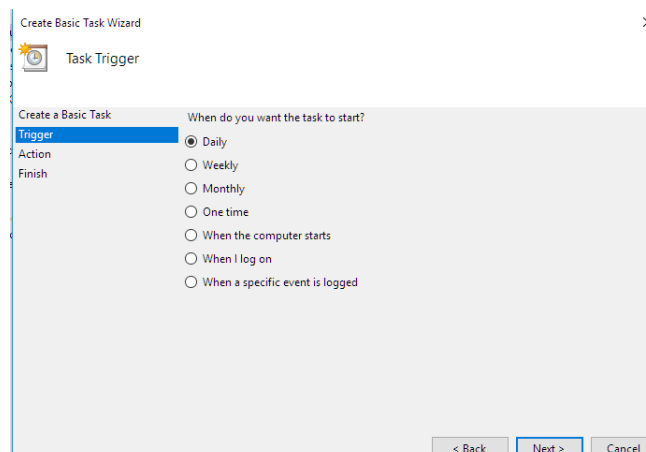


## MEMBUAT TASK

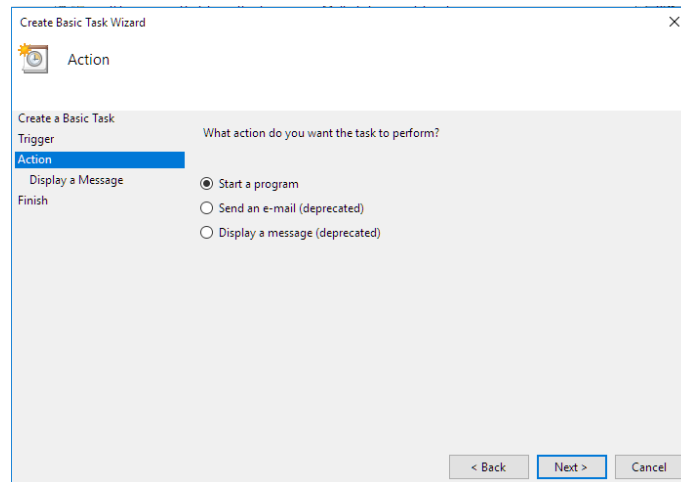
1. Klik “Create Basic Task...”
2. Isikan nama dari task yang akan dibuat, dan deskripsi tentang task itu.



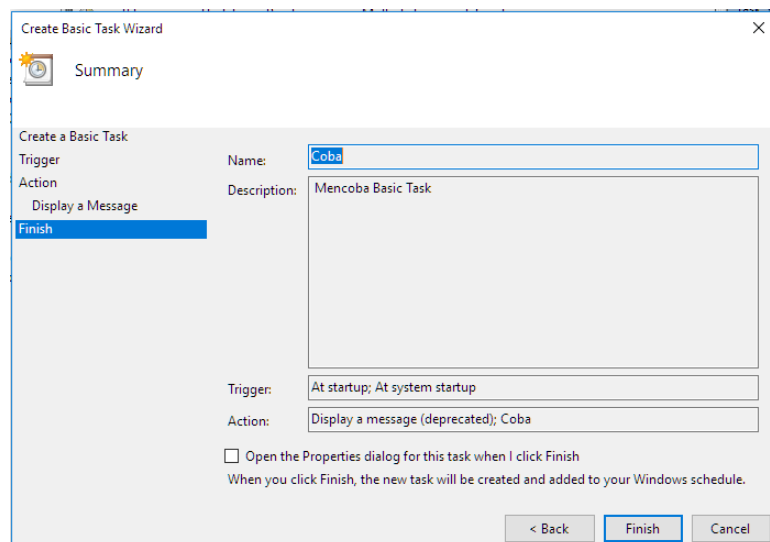
3. Tentukan trigger dari task, apakah task akan dilakukan setiap hari, setiap minggu, hanya sekali, atau ketika ada event yang spesifik



4. Tentukan kegiatan yang akan dilakukan, apakah memulai program lain, mengirim email, atau menampilkan sebuah message



5. Klik Finish dan task akan masuk ke dalam library dan siap berjalan

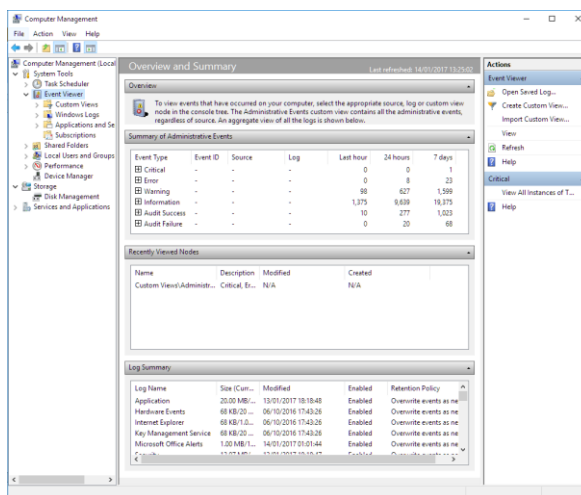



6. Task sudah masuk dalam Task Scheduler Library



Name	Status	Triggers
avast! Emergency Update	Ready	Multiple triggers defined
CCleanerSkipUAC	Ready	
Coba	Ready	At system startup
GoogleUpdateTaskMachi...	Disabled	Multiple triggers defined
GoogleUpdateTaskMachi...	Disabled	At 18:28 every day - After triggered, repeat every 1 hour for a
Microsoft Office 15 Sync ...	Disabled	Multiple triggers defined
RunAsStdUser Task	Ready	When the task is created or modified
SafeZone scheduled Auto...	Ready	Multiple triggers defined

## EVENT VIEWER

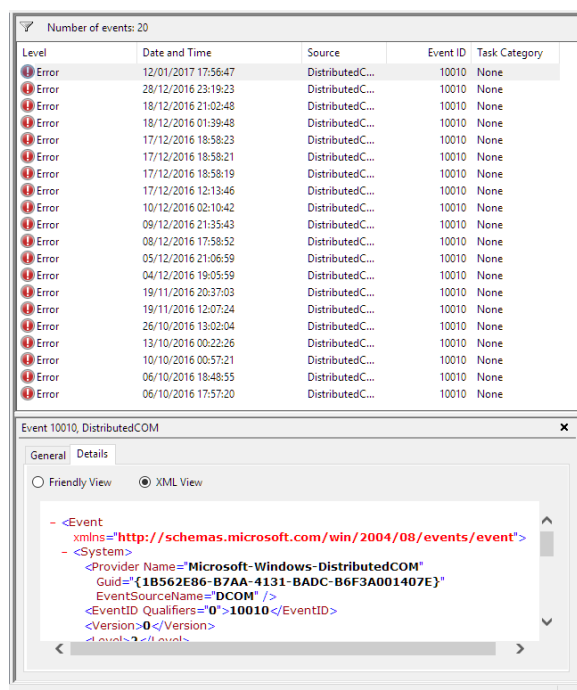
Event Viewer berguna untuk melihat bermacam – macam event yang terjadi di komputer. Dari event yang hanya berupa informasi saja sampai event suatu error dari program.



Untuk melihat event – event apa saja yang ada dalam suatu kategori tertentu (misalkan Error) klik  pada bagian kiri kategori.

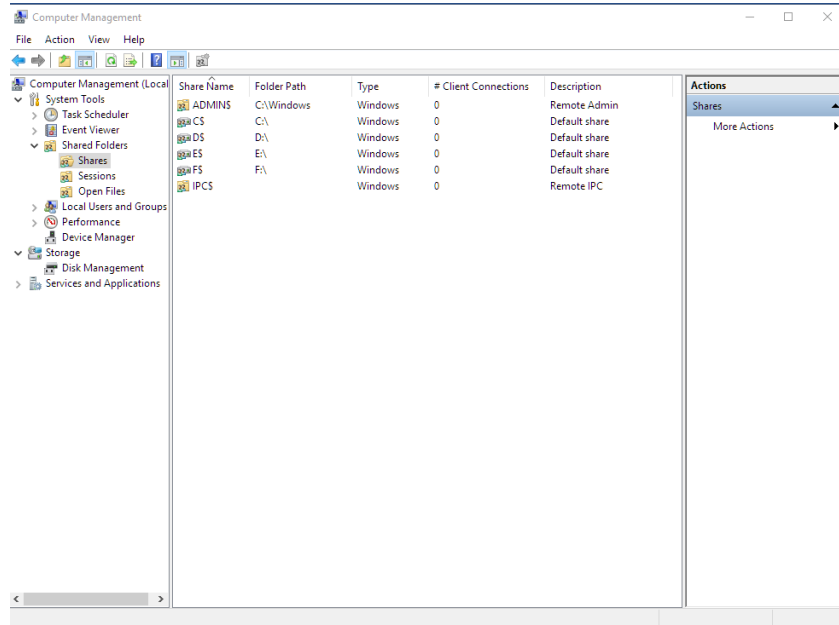
Event Type	Event ID	Source	Log	Last hour	24 hours	7 days
 Critical	-	-	-	0	0	1
 Error	-	-	-	0	8	23
	3	Kernel-EventTr...	Microsoft...	0	1	2
	512	PrintService	Microsoft...	0	1	2
	1000	Application Err...	Application	0	1	1
	2484	Apps	Application	0	1	1

Untuk melihat detail dari suatu event, double click pada nama event tersebut. Lalu akan langsung menuju details dari event yang bersangkutan



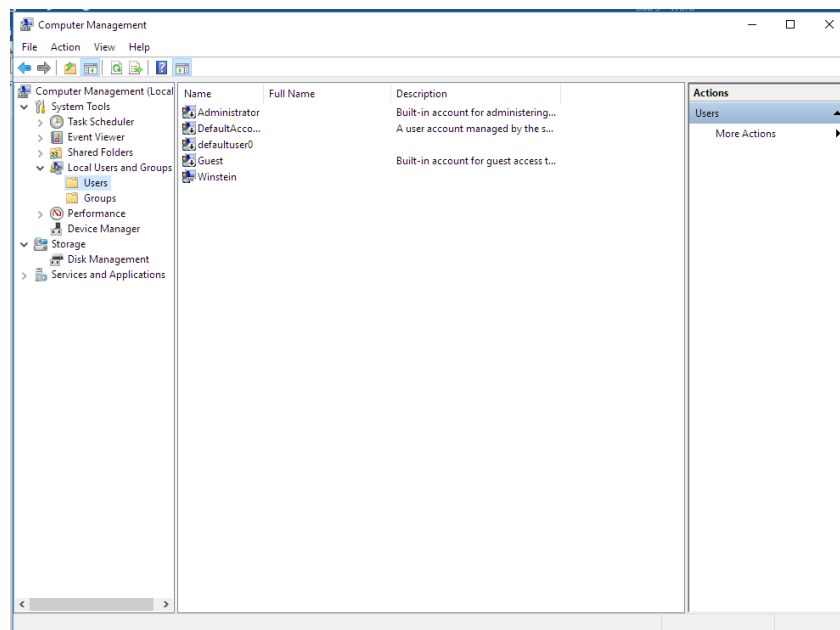
## SHARED FOLDERS

Pada Shared Folders, Dapat dilihat folder – folder apa saja yang ter-share dengan komputer yang bersangkutan



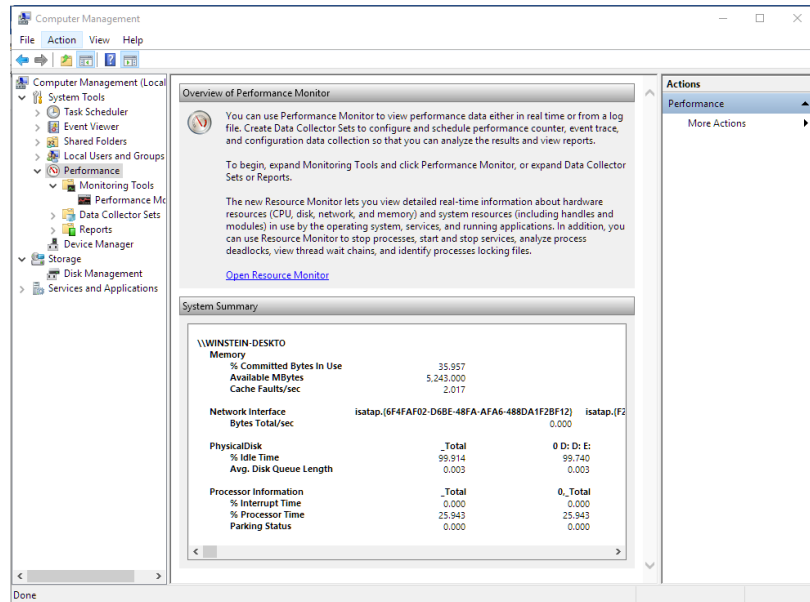
## LOCAL USERS AND GROUPS

Melihat user dalam komputer yang sama ataupun user yang tershare dalam suatu jaringan dalam grup.

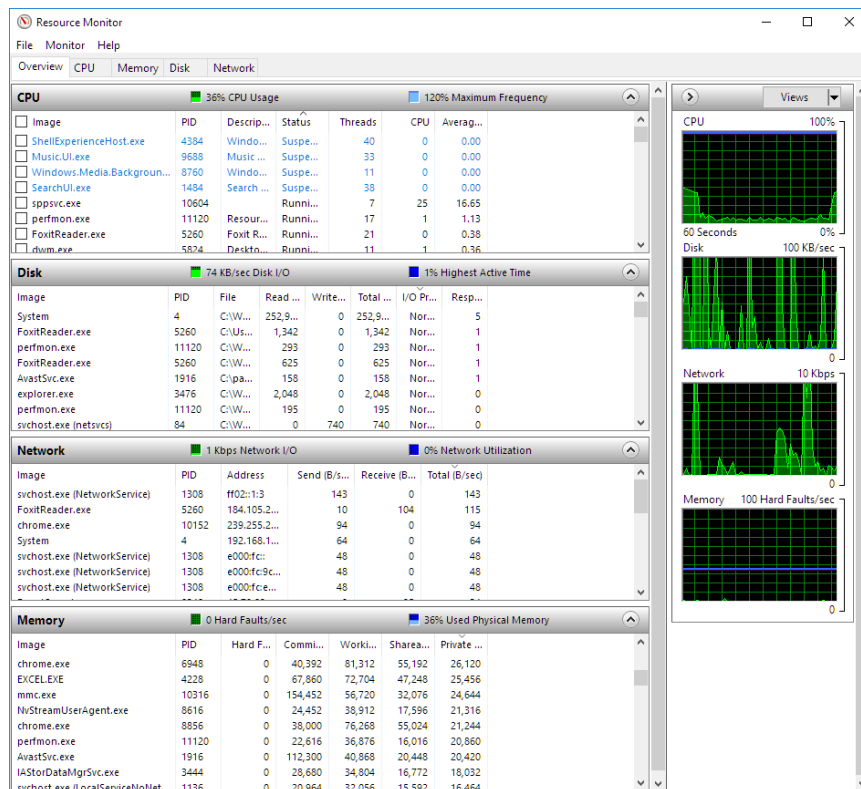


## PERFORMANCE

Melihat performa dari komputer. Untuk melihat detail dari proses – proses yang terjadi, klik “Open Resource Monitor”



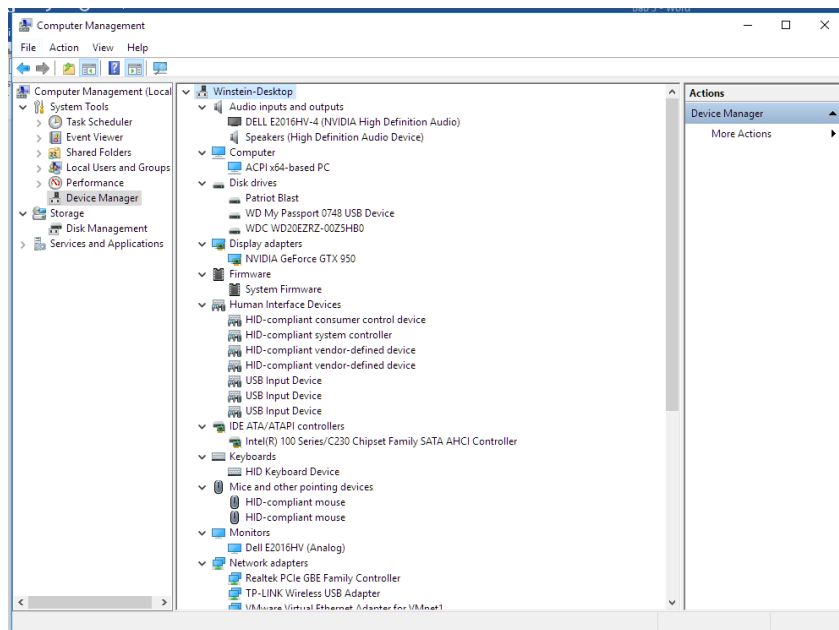
Pada Resource Monitor, Dapat terlihat secara detail penggunaan Processor yang dipakai komputer. Lalu dapat dilihat performa Disk yang ada. Dapat dilihat juga penggunaan RAM atau Memory yang sedang dipakai komputer dan juga Networking. Disisi kanan dapat dilihat grafik penggunaan CPU, Disk, Network, dan Memory dalam beberapa waktu terakhir.



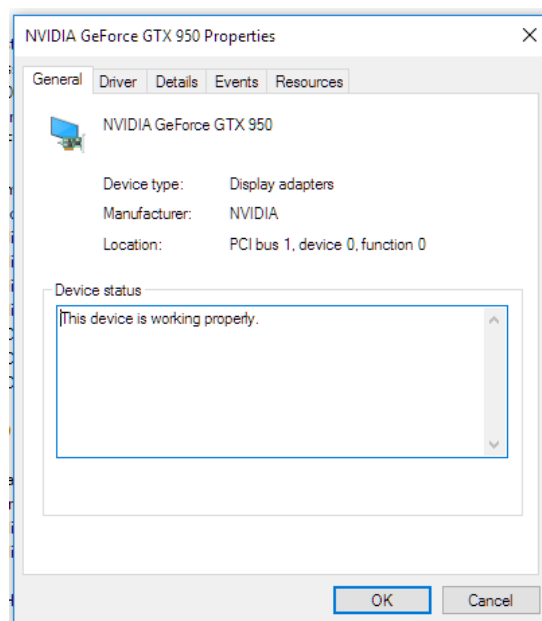
Untuk melihat lebih detail dari CPU, Disk, Network, atau Memory, dapat dilihat pada Tab yang bersangkutan.

## DEVICE MANAGER

Device Manager menampilkan perangkat keras apa saja yang terhubung dengan komputer. Misalnya seperti mouse dan keyboard, printer, atau processor.

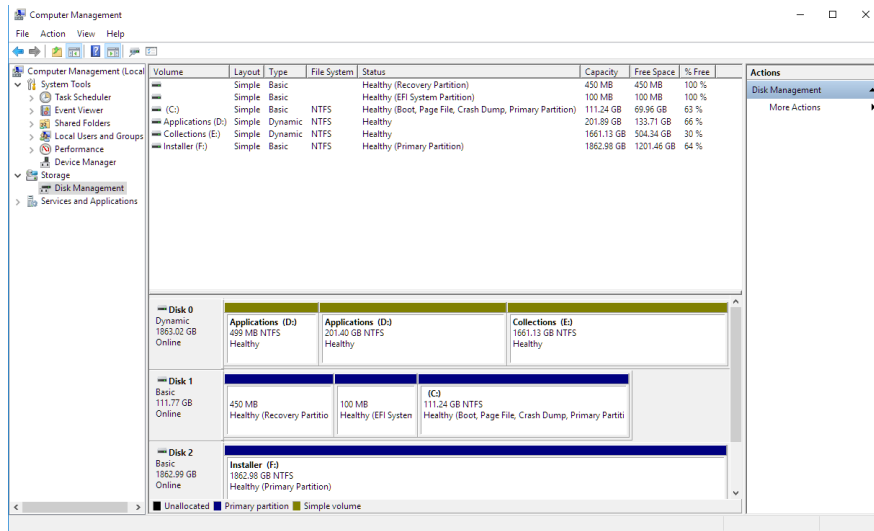


Dari gambar diatas, dapat dilihat bahwa komputer terhubung dengan audio, harddisk, graphics adapter, keyboard, mouse, dan hardware – hardware lainnya. Untuk melihat detail dari salah satu device, hanya perlu double click pada nama devicenyaa. Contohnya kita ingin melihat Graphics Adapter dari komputer tersebut, hanya perlu double click pada nama-nya (NVIDIA GeForce GTX 950) dan akan muncul properties dari device tersebut.



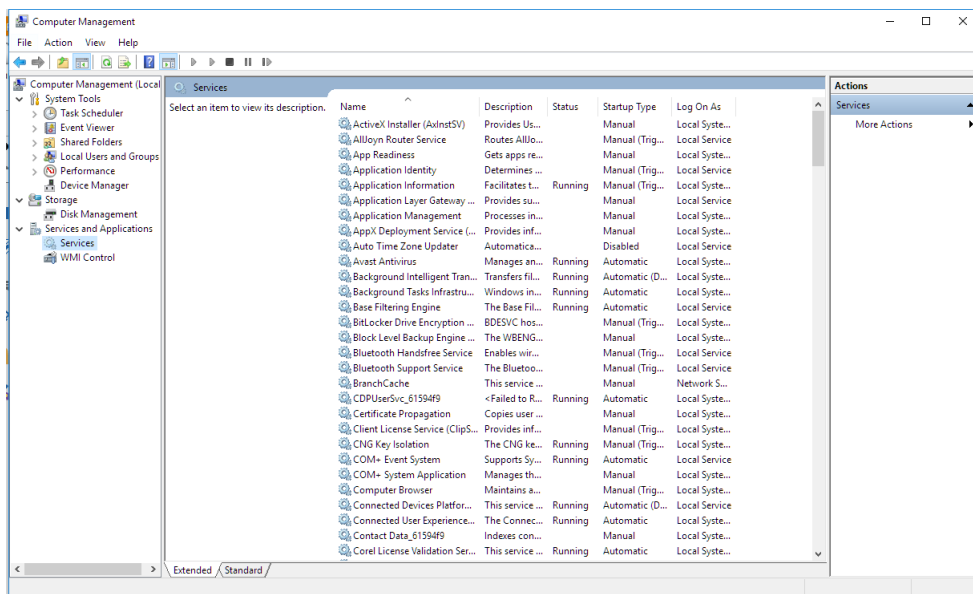
## DISK MANAGEMENT

Disk Management berfungsi untuk melihat kondisi dari semua disk yang terpasang pada komputer. Di Disk Management juga, dapat dilakukan partisi pada disk yang diinginkan.



## SERVICES

Pada Services, ditampilkan services dari semua aplikasi yang ada di Windows. Disini ditampilkan nama dan deskripsi dari suatu services, status services tersebut (apakah sedang berjalan / running atau tidak) dan Startup Type-nya.

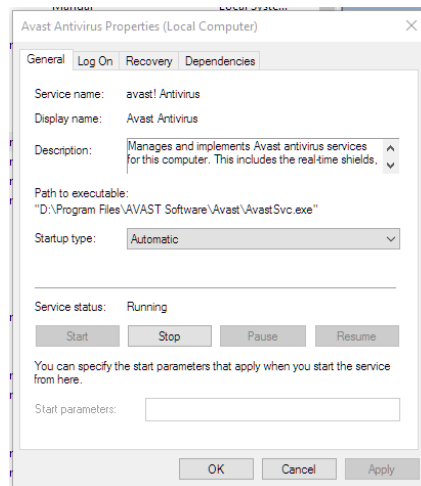


Jenis – jenis Startup Type :

- Manual, maka services tidak akan dijalan pada saat komputer dinyalakan.
- Automatic, maka services akan berjalan otomatis ketika komputer dijalankan.
- Automatic (Delayed Start), maka services akan berjalan setelah services Automatic yang lain berjalan.

- Jika terdapat Trigger Start, maka services dapat berjalan dengan trigger tertentu.
- Jika Startup Type-nya Disabled, maka services tidak akan dijalankan dan harus diaktifkan dulu.

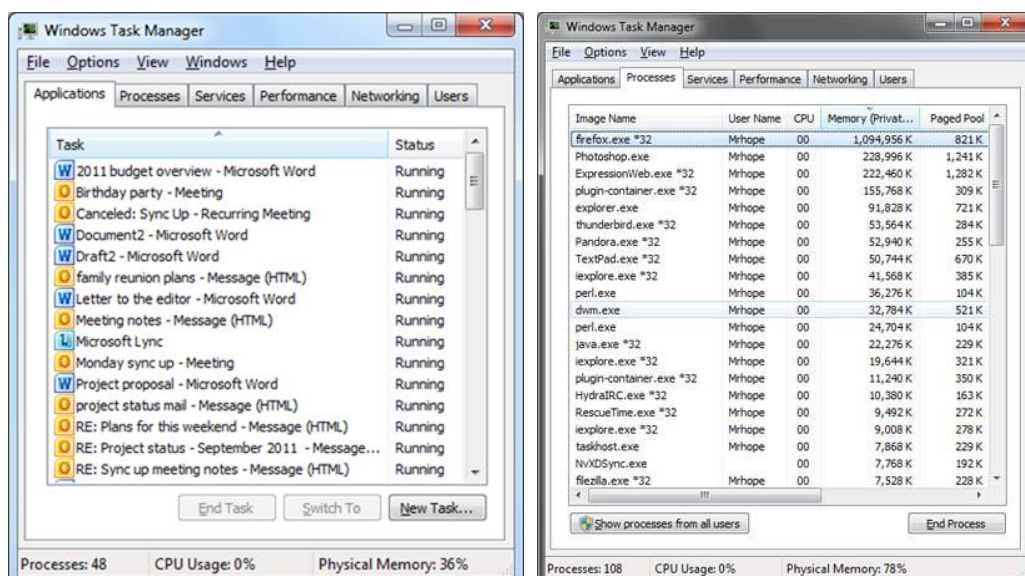
Untuk melihat detail dari salah satu services, hanya perlu double click pada nama services-nya. Contohnya kita ingin melihat services dari Avast Antivirus, hanya perlu klik kanan pada nama-nya (Avast Antivirus) lalu pilih properties. Maka akan muncul properties dari services tersebut. Disini juga, dapat diubah Startup Type dari services.



Di Services juga, kita dapat menonaktifkan services yang sedang berjalan dengan melakukan klik kanan pada nama services lalu pilih stop. Atau memberhentikan sementara dengan memilih pause dan untuk menjalankannya lagi pilih Resume. Dan dapat juga me-reset services dengan memilih Restart.

## TASK MANAGER

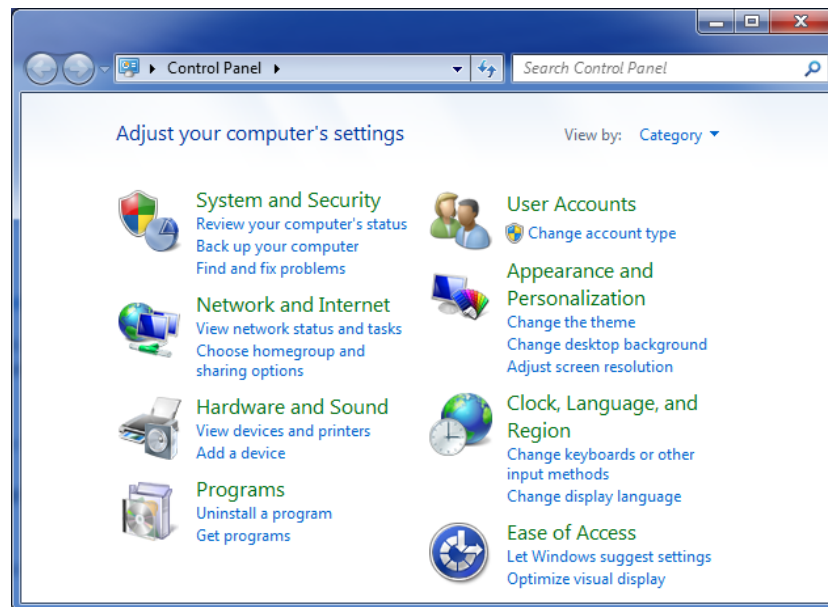
Windows Task Manager (TaskMgr) adalah sebuah aplikasi yang menyediakan informasi rinci tentang kinerja komputer dan menjalankan aplikasi, proses dan penggunaan CPU, melakukan informasi muatan dan memori, aktivitas jaringan dan statistik, log-in pengguna, dan sistem pelayanan. Task Manager juga dapat digunakan untuk menetapkan prioritas proses, afinitas prosesor, secara paksa menghentikan proses, dan mematikan, Aktifkan Ulang, hibernasi atau log off dari Windows. Windows Task Manager diperkenalkan dengan Windows NT 4.0. Versi sebelumnya dari Windows NT termasuk aplikasi Task List, yang memiliki fitur jauh lebih sedikit.





## CONTROL PANEL

Definisi Control Panel adalah bagian dari sistem operasi Microsoft Windows yang memungkinkan pengguna untuk melihat dan memanipulasi dasar sistem pengaturan dan kontrol melalui applet, seperti menambahkan hardware, menambahkan dan menghapus perangkat lunak, mengendalikan pengguna account, aksesibilitas mengubah pilihan, melakukan konfigurasi, personalisasi, kustomisasi, tweak, setting dari sistem operasi dapat dilihat ditetapkan, diubah, atau didefinisikan.



## TUGAS

1. Buatlah tutorial cara mempartisi suatu disk boleh dalam bentuk laporan dengan screenshot ataupun video.
2. Buatlah penjelasan setiap kategori dari Control Panel pada Windows 7.

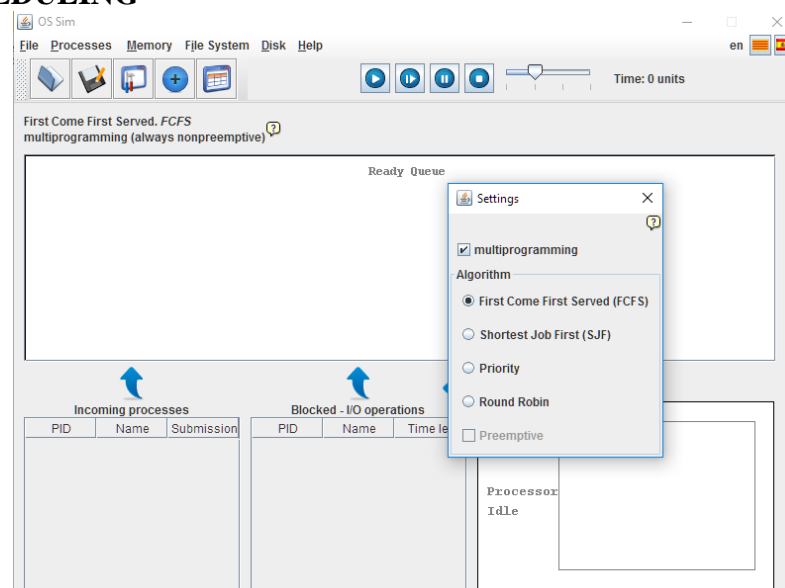
## PERTEMUAN 4

### PROCESS SCHEDULING DAN PEMROGRAMAN MULTITHREAD

#### TUJUAN PRAKTIKUM

- Praktikan dapat memahami proses penjadwalan pada suatu system operasi
- Praktikan dapat memahami cara pemrograman multithread
- Praktikan dapat menguasai dan mengenal macam-macam jenis proses penjadwalan.

#### PROCESS SCHEDULING



Process Scheduling menjadwalkan bermacam – macam proses yang akan di kerjakan oleh CPU berdasarkan algoritma penjadwalan tertentu. Ada 4 algoritma penjadwalan yang akan dibahas :

- First-Come, First-Served (FCFS)
- Shortest-Job-First (SJF)
- Priority
- Round Robin(RR)

Algoritma penjadwalan ada yang preemptive dan non-preemptive. Algoritma non-preemptive dibuat agar jika ada proses yang masuk ke running state, maka proses itu tidak bisa diganggu sampai menyelesaikan prosesnya, sedangkan penjadwalan preemptive menggunakan prioritas dimana jadwal dapat mengganggu proses dengan prioritas rendah ketika proses dengan prioritas tinggi sedang running.

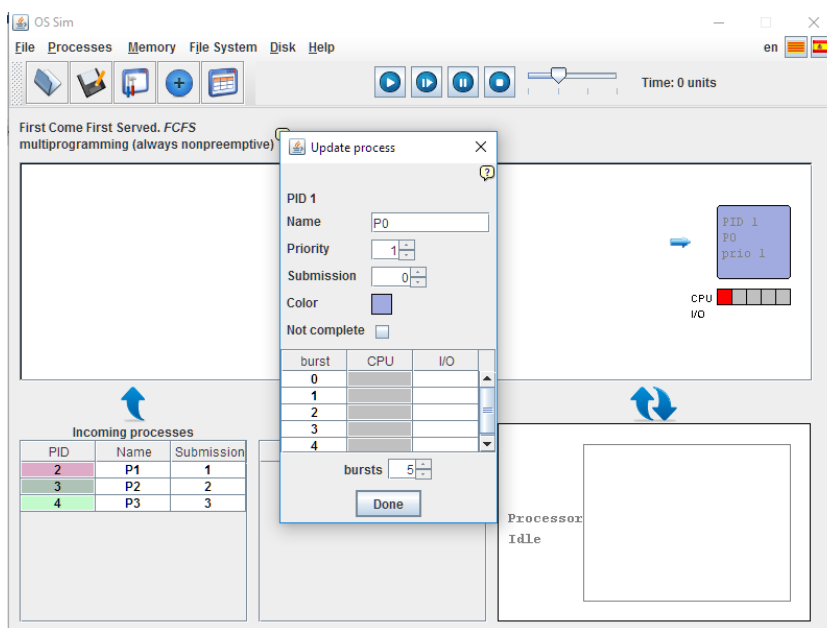
#### First Come First Serve (FCFS)

- Proses dieksekusi berdasarkan yang pertama kali datang
- Ini adalah algoritma penjadwalan non-preemptive
- Mudah untuk dipahami dan digunakan
- Implementasinya berdasarkan antrian FIFO (First In, First Out)

- Performanya buruk dan average waiting time yang tinggi

Process	Arrival Time	Burst Time	Service Time
<b>P0</b>	0	5	0
<b>P1</b>	1	3	5
<b>P2</b>	2	8	8
<b>P3</b>	3	6	16

Di OS Sim, Proses 0 akan berbentuk seperti ini :



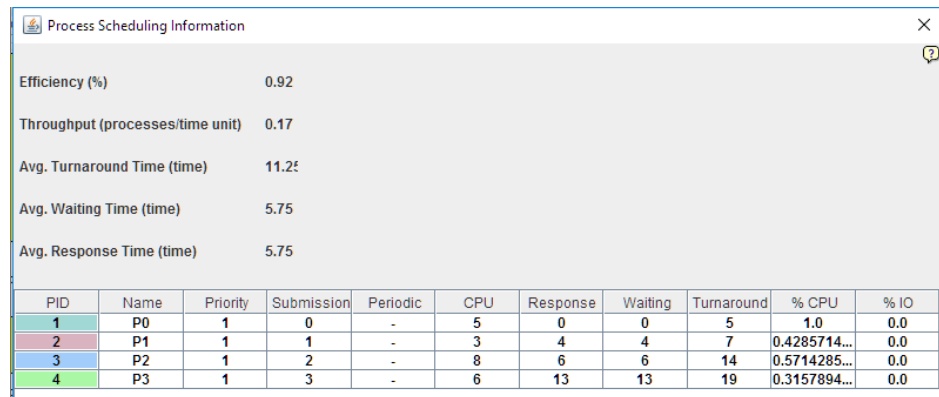
Wait time setiap proses :

Process	Wait time : Service Time – Arrival Time
<b>P0</b>	0 – 0 = 0
<b>P1</b>	5 – 1 = 4
<b>P2</b>	8 – 2 = 6
<b>P3</b>	16 – 3 = 13

Average Wait

Time:  $(0+4+6+13) / 4 = 5.75$

Hasil di OS Sim sama dengan hasil penghitungan manual.



PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	0	5	1.0	0.0
2	P1	1	1	-	3	4	4	7	0.4285714...	0.0
3	P2	1	2	-	8	6	6	14	0.5714285...	0.0
4	P3	1	3	-	6	13	13	19	0.3157894...	0.0

### Shortest Job First (SJF)

- Pendekatan terbaik untuk mengurangi waiting time
- Mudah diimplementasikan di Batch systems dimana kebutuhan CPU sudah diketahui
- Tidak mungkin diimplementasikan pada sistem yang interaktif dimana kebutuhan CPU tidak diketahui.
- Processor harus sudah tahu berapa lama waktu yang akan dipakai tiap prosesnya.

Non – preemptive

Process	Arrival Time	Burst Time	Service Time
<b>P0</b>	0	5	0
<b>P1</b>	1	3	5
<b>P2</b>	2	8	14
<b>P3</b>	3	6	8

Wait time setiap proses :

Process	Wait time : Service Time – Arrival Time
<b>P0</b>	$0 - 0 = 0$
<b>P1</b>	$5 - 1 = 4$
<b>P2</b>	$14 - 2 = 12$
<b>P3</b>	$8 - 3 = 5$

Average Wait Time:  $(0+4+12+5) / 4 = 5.25$

Process Scheduling Information

Efficiency (%) 1.00

Throughput (processes/time unit) 0.18

Avg. Turnaround Time (time) 10.75

Avg. Waiting Time (time) 5.25

Avg. Response Time (time) 5.25

PID	Name ^	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	0	5	1.0	0.0
2	P1	2	1	-	3	4	4	7	0.4285714...	0.0
3	P2	1	2	-	8	12	12	20	0.4	0.0
4	P3	3	3	-	6	5	5	11	0.5454545...	0.0

**Preemptive**

Wait time setiap proses :

Process	Wait time : Service Time – Arrival Time
<b>P0</b>	$(0 - 0) + (4 - 1) = 3$
<b>P1</b>	$1 - 1 = 0$
<b>P2</b>	$14 - 2 = 12$
<b>P3</b>	$8 - 3 = 5$

Average Wait Time:  $(3+0+12+5) / 4 = 5.00$

Process Scheduling Information

Efficiency (%) 0.96

Throughput (processes/time unit) 0.17

Avg. Turnaround Time (time) 10.50

Avg. Waiting Time (time) 5.00

Avg. Response Time (time) 4.25

PID	Name ^	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	3	8	0.625	0.0
2	P1	2	1	-	3	0	0	3	1.0	0.0
3	P2	1	2	-	8	12	12	20	0.4	0.0
4	P3	3	3	-	6	5	5	11	0.5454545...	0.0

Perbedaan antara penjadwalan preemptive dan non-preemptive.

Basis	Preemptive	Non Preemptive
Dasar	Resources dialokasikan pada proses dalam waktu tertentu	Sewaktu resources dialokasikan kepada proses. Proses akan memegang resources sampai selesai burst time atau berganti menjadi waiting state
Interrupt	Process dapat di-interrupt	Process tidak dapat di-interrupt hingga selesai atau berganti menjadi waiting state
Starvation	Jika proses berprioritas tinggi sering datang dalam antrian, maka proses berprioritas rendah dapat starve	Jika proses dengan burst time yang lama sedang berjalan, maka proses dengan burst time kecil dapat starve
Overhead	Ya	Tidak
Fleksibilitas	Lebih Fleksibel	Kaku
Cost	Cost associated	Tidak cost associated

### Priority

- Penjadwalan Priority merupakan algoritma non-preemptive dan salah satu algoritma penjadwalan yang paling sering dipakai di batch system.
- Setiap proses diberi prioritas. Proses dengan prioritas tertinggi akan dikerjakan pertama kali.
- Proses dengan prioritas yang sama akan dijalankan dengan prinsip FIFO.
- Prioritas dapat ditentukan berdasarkan kebutuhan memori, kebutuhan waktu, atau kebutuhan resources yang lain.

Process	Arrival Time	Burst Time	Priority	Service Time
<b>P0</b>	0	5	1	0
<b>P1</b>	1	3	2	11
<b>P2</b>	2	8	1	14
<b>P3</b>	3	6	3	5

Wait time setiap proses :

Process	Wait time : Service Time – Arrival Time
<b>P0</b>	$0 - 0 = 0$
<b>P1</b>	$11 - 1 = 10$
<b>P2</b>	$14 - 2 = 12$
<b>P3</b>	$5 - 3 = 2$

Average Wait Time:  $(0+10+12+2) / 4 = 6.0$

PID	Name ^	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	0	5	1.0	0.0
2	P1	2	1	-	3	10	10	13	0.2307692...	0.0
3	P2	1	2	-	8	12	12	20	0.4	0.0
4	P3	3	3	-	6	2	2	8	0.75	0.0

### Round Robin

- Round Robin adalah algoritma penjadwalan preemptive.
- Setiap proses dilakukan dalam interval waktu yang sama yang disebut **quantum**.
- Ketika proses sedang dikerjakan dalam jangka waktu yang ditentukan, maka proses akan di-interrupt dan proses lain akan dikerjakan dalam jangka waktu yang sama.
- Context switching dipakai untuk menyimpan state dari proses yang di-interrupt.

Wait time setiap proses dengan nilai quantum = 3.

Process	Wait time : Service Time – Arrival Time
<b>P0</b>	$(0 - 0) + (12 - 3) = 9$
<b>P1</b>	$3 - 1 = 2$
<b>P2</b>	$(6 - 2) + (14 - 9) + (20 - 17) = 12$
<b>P3</b>	$(9 - 3) + (17 - 12) = 11$

Average Wait Time:  $(9+2+12+11) / 4 = 8.5$

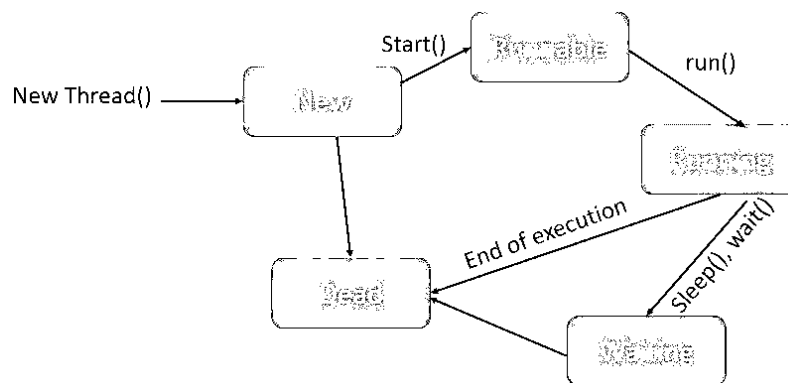
PID	Name ^	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	9	14	0.3571428...	0.0
2	P1	2	1	-	3	2	2	5	0.6	0.0
3	P2	1	2	-	8	4	12	20	0.4	0.0
4	P3	3	3	-	6	6	11	17	0.3529411...	0.0

## Pemrograman Multi-thread

Program multi-thread berisi 2 atau lebih proses yang dapat di run secara bersamaan dan setiap proses dapat meng-handle task yang berbeada dalam waktu yang sama agar penggunaan resource yang ada lebih optimal jika computer kita memiliki beberapa CPU.

Multi-threading memperluas pemikiran dari multitasking pada aplikasi dimana kita dapat membagi beberapa operasi yang spesifik dalam satu aplikasi menjadi thread. Dimana setiap thread berjalan secara parallel. Sistem Operasi membagi processing time tidak hanya aplikasi yang berbeda, tetapi juga setiap thread dalam satu aplikasi.

Dengan multi-threading, kita menulis program dengan banyak aktivitas yang harus berjalan bersamaan dalam satu program.



- **New** – Thread baru muncul dalam new state. Thread akan tetap seperti ini sampai ada program yang memulau thread tersebut. Biasa disebut **born thread**.
- **Runnable** – Setelah thread baru dimulai, thread menjadi runnable. Thread yang berada dalam kondisi ini berarti sedang menjalankan suatu task.
- **Waiting** – Terkadang, thread ber-transisi ke waiting state ketika thread menunggu thread lain menyelesaikan task-nya. Thread dapat kembali ke runnable state ketika thread lain memberi sinyal agar waiting state dapat kembali dijalankan.
- **Timed Waiting** – Runnable thread dapat menjadi timed waiting state pada interval waktu tertentu.
- **Terminated (Dead)** – Runnable thread dapat menjadi terminated state ketika sudah menyelesaikan tasknya.

Kita harus meng-override method **run( )** di class Thread. Method ini memberi titik masuk untuk thread dan kita menulis kode program di method ini. Ini contoh syntax sederhana method run().

```

public void run() {
    /**
     * Your Multithread codes goes here
     */
}
  
```

Ketika Thread sudah terbentuk, kita dapat memulainya dengan menggunakan method **start()**, dimana akan menjalankan method run(). Ini contoh syntax sederhana method start().

```

public void start () {
    System.out.println("Starting " + threadName );
    if (t == null) {
        t = new Thread (this, threadName);
        t.start ();
    }
}
  
```



Lalu, untuk membuat thread.

```
class RunnableDemo implements Runnable {
    private Thread t;
    private String threadName;
    int a = 0;

    RunnableDemo( String name) {
        threadName = name;
        System.out.println("Creating " + threadName );
    }
}

public class JavaApplication1 {

    public static void main(String args[]) {
        RunnableDemo R1 = new RunnableDemo( "Thread-1");
        R1.start();

        RunnableDemo R2 = new RunnableDemo( "Thread-2");
        R2.start();
    }
}
```

Berikut adalah syntax sederhana yang mamakai multi threading dan output-nya :

```
package javaapplication1;

/**
 *
 * @author Winstein
 */
class RunnableDemo implements Runnable {
    private Thread t;
    private String threadName;
    int a = 0;

    RunnableDemo( String name) {
        threadName = name;
        System.out.println("Creating " + threadName );
    }

    public void run() {
        System.out.println("Running " + threadName );
        try {
            for(int i = 0; i < 2; i++) {
                a++;
                System.out.println("Nilai : " + a + " , " + threadName);
                Thread.sleep(50);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread " + threadName + " interrupted.");
        }
        System.out.println("Thread " + threadName + " exiting.");
    }

    public void start () {
        System.out.println("Starting " + threadName );
        if (t == null) {
            t = new Thread (this, threadName);
            t.start ();
        }
    }
}

public class JavaApplication1 {

    public static void main(String args[]) {
        RunnableDemo R1 = new RunnableDemo( "Thread-1");
        R1.start();

        RunnableDemo R2 = new RunnableDemo( "Thread-2");
        R2.start();
    }
}
```

Output :

```

run:
Creating Thread-1
Starting Thread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Nilai : 1 , Thread-1
Running Thread-2
Nilai : 1 , Thread-2
Nilai : 2 , Thread-1
Nilai : 2 , Thread-2
Thread Thread-2 exiting.
Thread Thread-1 exiting.
BUILD SUCCESSFUL (total time: 0 seconds)

```

Seperti yang bisa kita lihat, int a dipakai oleh 2 proses (Thread-1 dan Thread-2) masing – masing 2 kali. Tetapi nilai a bukan 4, melainkan 2. Hal ini karena 2 proses tadi berjalan bersamaan.

Thread-1	Thread-2	Value of a by Thread -1	Value of a by Thread -2
CREATE		NULL	NULL
START		0	NULL
	CREATE	0	0
	START	0	0
RUN : a++		1	0
	RUN : a++	1	1
a++		2	1
	a++	2	2
	EXIT	2	2
EXIT		2	2

### Thread Methods

Ini beberapa method penting dalam class Thread.

No.	Method & Description
1	<b>public void start()</b> Memulai thread, lalu memicu method run() dalam Thread.
2	<b>public void run()</b> Jika thread dimulai menggunakan runnable target yang berbeda, maka method run() akan memulai runnable object.
3	<b>public final void setName(String name)</b> Mengubah nama Thread. Terdapat method getName() untuk mengambil nama Thread.

4	<b>public final void setPriority(int priority)</b> Mengatur prioritas Thread. Nilainya dapat diatur dari 1 sampai 10.
5	<b>public final void setDaemon(boolean on)</b> Parameter true untuk thread yang Daemon
6	<b>public final void join(long millisec)</b> Thread sekarang memulai thread kedua dan selanjutnya, menimbulkan thread sekarang untuk mengblok sampai thread kedua selesai atau setelah beberapa milisekon.
7	<b>public void interrupt()</b> Interrupt thread ini. Membuat thread berjalan jika diblok.
8	<b>public final boolean isAlive()</b> Return true jika thread masih berjalan, yaitu kapanpun semenjak thread dimulai tetapi sebelum selesai.

Method – method diatas dipakai pada Thread – thread tertentu. Method dibawah ini bersifat statis. Menggunakan satu method statis memulai operasi pada thread yang sedang berjalan.

No.	Method & Description
1	<b>public static void yield()</b> Membuat thread yang sedang berjalan untuk memberikan waktunya untuk thread lain dengan prioritas yang sama yang sedang waiting state.
2	<b>public static void sleep(long millisec)</b> Membuat thread yang sedang berjalan untuk berhenti selama beberapa milisekon.
3	<b>public static boolean holdsLock(Object x)</b> Return true jika thread sekarnag memakai lock dari objek x
4	<b>public static Thread currentThread()</b> Return suatu reference pada thread yang sedang berjalan, yaitu thread yang memulai method ini.
5	<b>public static void dumpStack()</b> Mencetak jejak stack yang dipakai thread yang sedang berjalan, berguna ketika debugging aplikasi multithread.

### Tugas

1. Jawab pertanyaan pada latihan OS-SIM multiprogramming atau mono programming.
2. Jawab pertanyaan pada latihan OS-SIM scheduling algorithm performance comparison

## PERTEMUAN 5

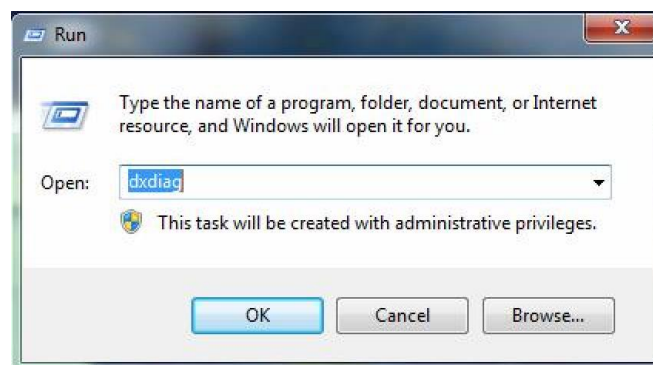
### PERINTAH-PERINTAH COMMAND RUN, PROMPT, DAN BERBAGI RESOURCE DI WINDOWS 7

#### TUJUAN PRAKTIKUM

- Praktikan dapat mengetahui perintah – perintah pada command run, prompt
- Praktikan dapat mensetting parameter dalam berbagi file
- Praktikan dapat mengatur hak akses pada suatu file di windows 7

#### RUN

Pengertian Run adalah suatu Tools pada Windows yang bertujuan untuk memanggil suatu Program atau Aplikasi dengan menggunakan Perintah atau Comment. Cara membuka Run ada beberapa cara Klik Start -> Run, atau dengan mengetik Logo Windows pada Keyboard + R



- **Program Run Command**
- **WINDOWS EXPLORER**

#### *Pengelola File pada Sistem Operasi*

Setiap sistem operasi umumnya dilengkapi dengan pengelola file. Di Linux tersedia File Manager dan di Windows tersedia Windows Explorer. Windows juga menyediakan folder My Documents sebagai tempat baku penyimpanan file dokumen.

- **Shortcut**

Cara Membuat Shortcut di Windows 7 cukup mudah. Sesuai namanya, 'shortcut' yang berarti jalan pintas adalah fitur yang memungkinkan pengguna untuk mengakses file, folder, maupun aplikasi di komputer dengan hanya 1 klik saja.

Beberapa contoh perintah yang dapat digunakan pada Run adalah sebagai berikut.

<b>Perintah</b>	<b>Keterangan</b>
logoff	Log off from Windows
calc	Launch Calculator
cmd	Command Prompt
control	Control Panel
compmgmt.msc	Computer Management
comexp.msc	Component Services
devmgmt.msc	Device Manager
dxdiag	DirectX Diagnostic Tool
cleanmgr	Disk Cleanup
dfrgui	Disk Defragmenter
diskmgmt.msc	Disk Management
utilman	Ease of Access Center
eventvwr.msc	Event Viewer
iexplore	Launch Internet Explorer
gpedit.msc	Local Group Policy Editor
secpol.msc	Local Security Policy
lusrmgr.msc	Local Users and Groups
magnify	Launch Magnifier
narrator	Launch Narrator
notepad	Launch Notepad
regedit	Registry Editor
mspaint	Launch Paint
powercfg.cpl	Power Options
appwiz.cpl	Programs and Features
mstsc	Remote Desktop Connection
snippingtool	Snipping Tool
msconfig	System Configuration
sysedit	System Configuration Editor
msinfo32	System Information
taskmgr	Task Manager
explorer	Launch Windows Explorer
firewall.cpl	Windows Firewall
powershell	Windows Powershell
write	Launch WordPad

- Command Prompt (cmd)



Perintah-perintah yang dapat dijalankan pada Command Prompt dapat dilihat dengan cara membuka Command Prompt lalu menuliskan “help” dan menekan tombol Enter. Sedangkan untuk mendapatkan informasi lebih spesifik mengenai sebuah perintah, dapat mengetik “help command” dimana command adalah perintah yang ingin kita dapatkan informasinya. Berikut ini adalah beberapa contoh perintah pada Command Prompt beserta fungsinya.

Perintah	Keterangan
cd	Displays the name of or changes the current directory
chdir	Displays the name of or changes the current directory
chkdsk	Checks a disk and displays a status report
cls	Clears the screen
cmd	Starts a new instance of the Windows command interpreter
comp	Compares the contents of two files or sets of files
copy	Copies one or more files to another location
date	Displays or sets the date
del	Deletes one or more files
dir	Displays a list of files and subdirectories in a directory
echo	Displays messages, or turns command echoing on or off
erase	Deletes one or more files
exit	Quits the CMD.EXE program (command interpreter)
help	Provides help information for windows commands
md	Creates a directory
mkdir	Creates a directory
move	Moves one or more files from one directory to another directory
print	Prints a text file
rd	Removes a directory
ren	Renames a file or files
rename	Renames a file or files
rmdir	Removes a directory
shutdown	Allows proper local or remote shutdown of machine
start	Starts a separate window to run a specified program or command
systeminfo	Displays machine specific properties and configuration

tasklist	Displays all currently running tasks including services
taskkill	Kill or stop a running process or application
time	Displays or sets the system time
type	Displays the contents of a text file

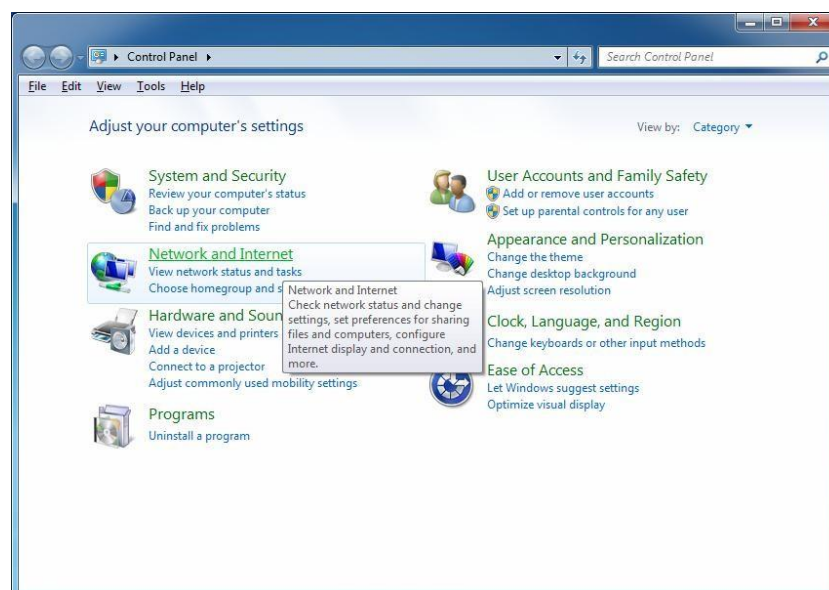
## SHARING

Sistem yang diperlukan :

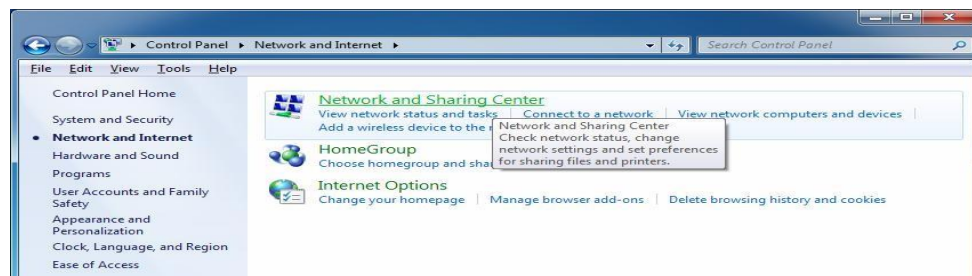
- Dua buah PC atau lebih dengan sistem operasi Windows 7
- Jaringan Local Area (LAN)
- Start Control Panel



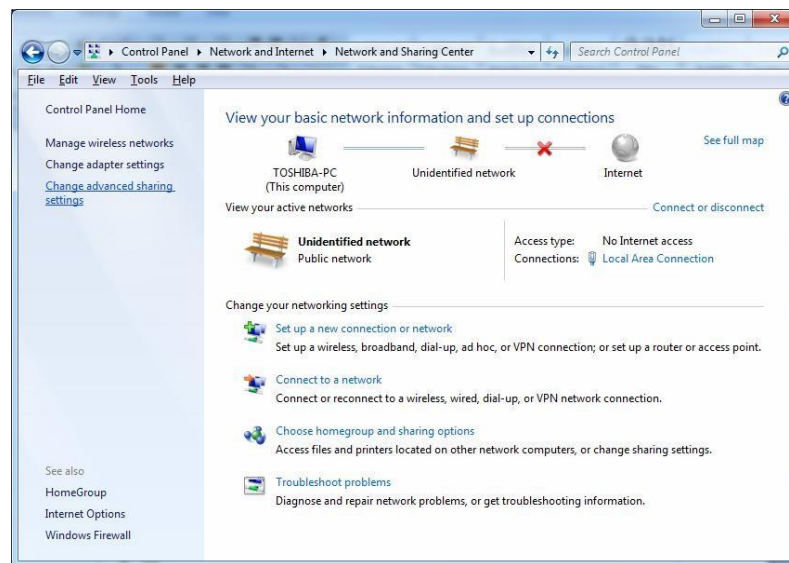
- Network and Internet



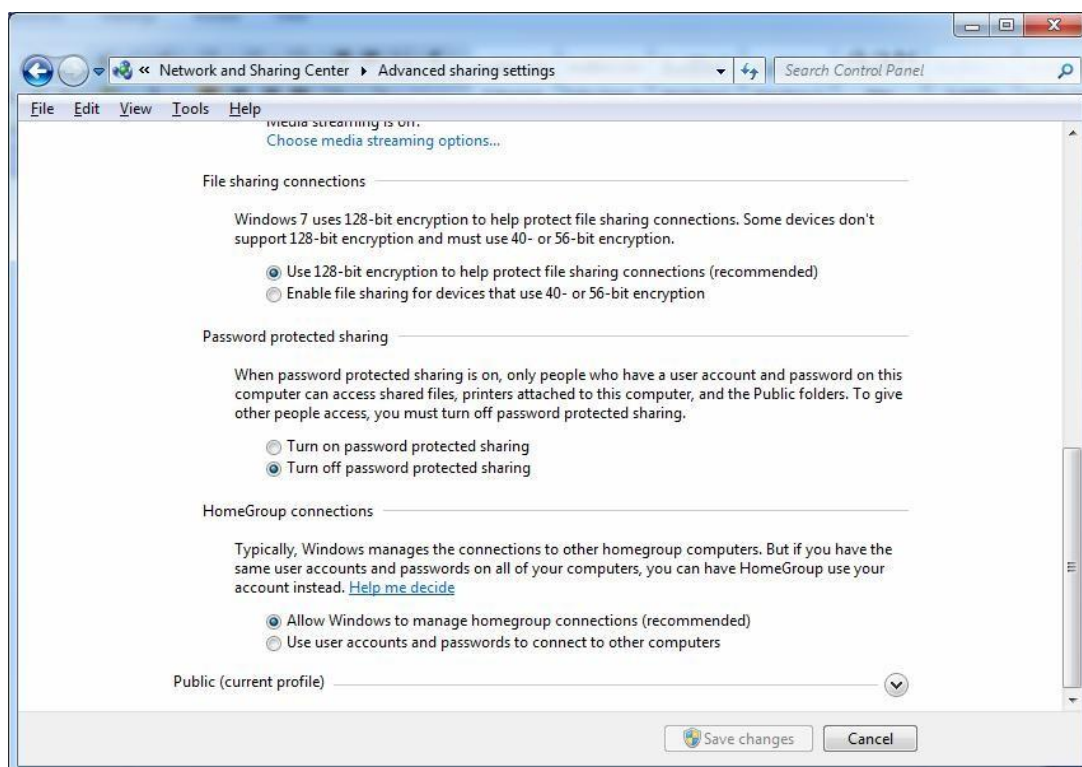
- Network and Sharing Center



- Change advanced sharing settings

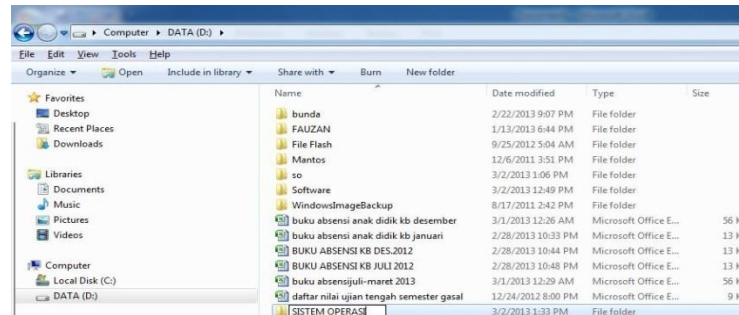


- Turn off password protected sharing

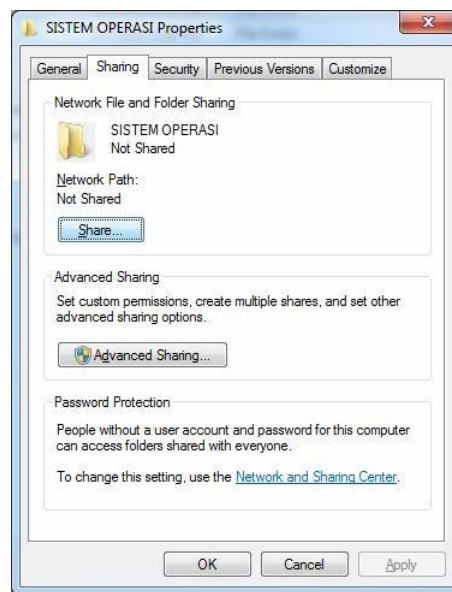




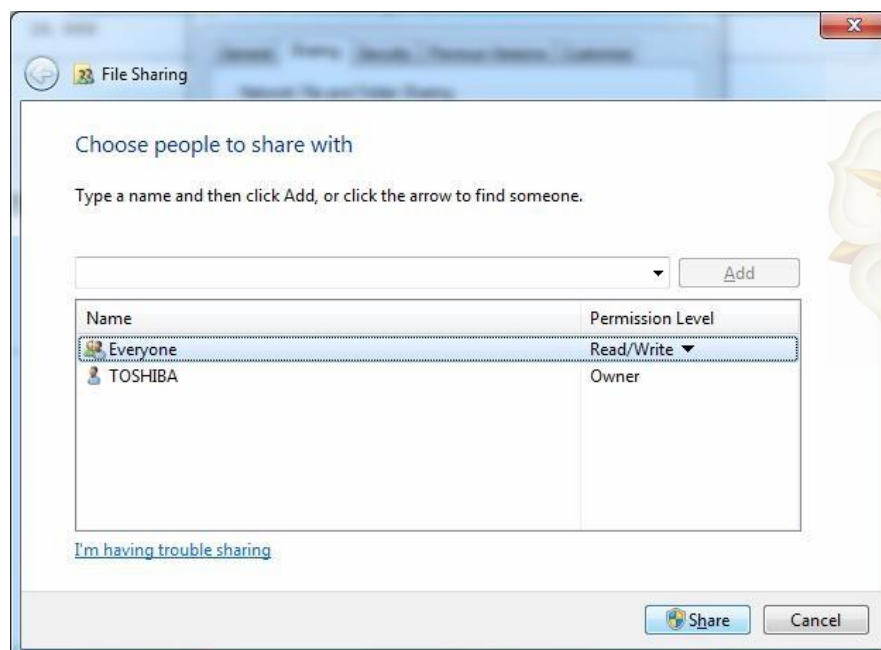
- Buat Folder melalui Windows Explorer



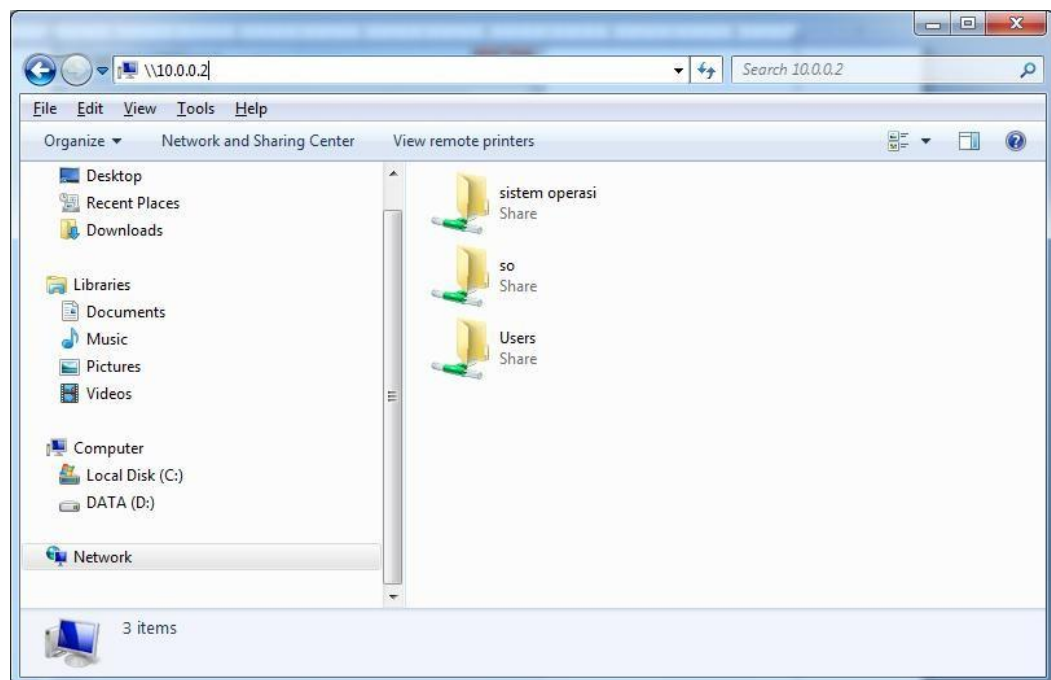
- Klik kanan sharing share



- Everyone Read/Write share Done Close



- Coba akses folder sharing tersebut melalui Windows Explorer



### Tugas

1. Buat tutorial berupa screenshot dan keterangan untuk dapat membuat sebuah folder baru di hardisk dengan menggunakan command prompt dengan alamat sebagai berikut:
  - D:\Latihan CMD\SATU\2\tiga
  - D:\Latihan CMD\DUA\1
  - D:\Latihan CMD\TIGA\2\satu
  - D:\Latihan CMD\EMPAT\3\dua
2. Buat tutorial berupa screenshot dengan keterangan untuk dapat membuat sharing folder yang dapat diakses jika memiliki password.

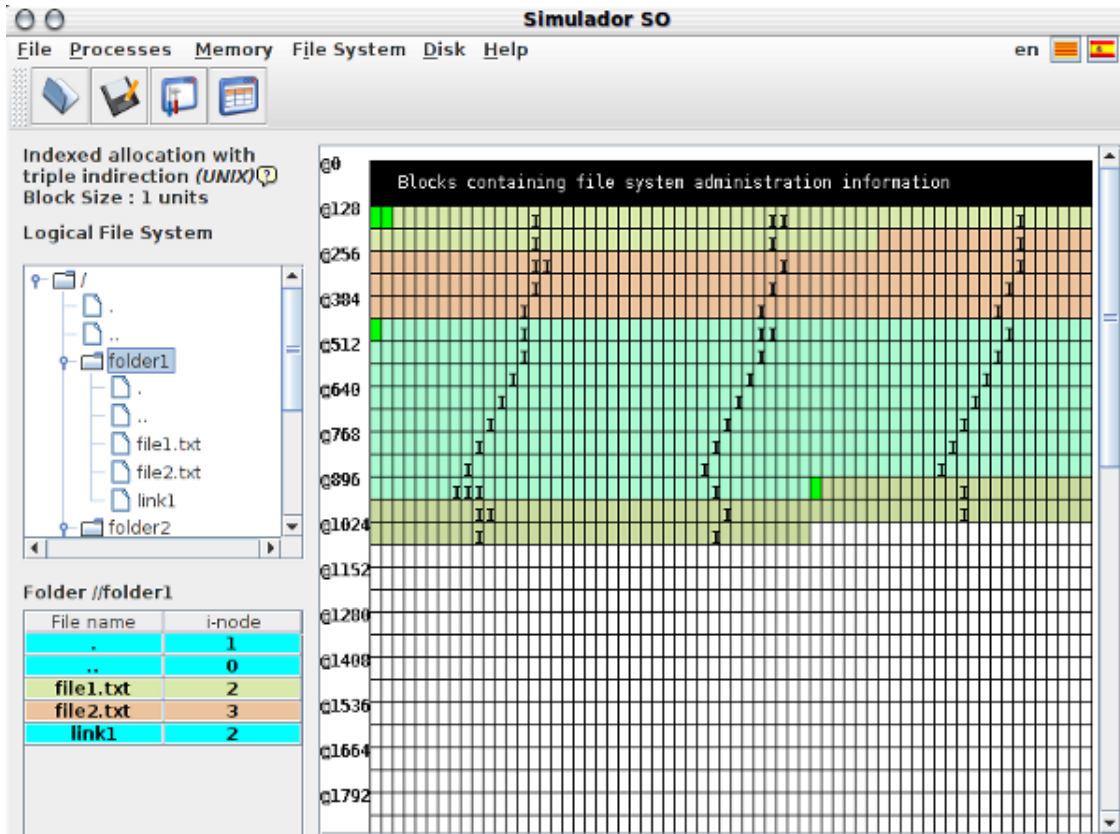
## PERTEMUAN 6

### File System Management dan Disk Scheduling (OSSIM)

#### TUJUAN PRAKTIKUM

- Praktikan dapat memahami proses manajemen file system.
- Praktikan dapat mensetting parameter dalam manajemen file system.
- Praktikan dapat memahami proses penjadwalan disk.

#### FILE SYSTEM MANAGEMENT



Sistem file mengatur hubungan antara logical space sebagai pengguna melihat (file, direktori, link) dengan struktur yang sesuai dan lokasi fisik (nyata) obyek dalam memori sekunder.

Unit kerja minimal sistem file adalah blok, memori dan sistem file objek dibagi menjadi blok dengan ukuran yang sama, dan sistem manajemen adalah untuk mengalokasikan blok-blok objek ke dalam blok yang tersedia pada memori sekunder dan menjaga kontrol.

Simulasi menerapkan dua teknik utama untuk Manajemen File System:

- Alokasi Link dengan tabel alokasi file (FAT, MS-DOS environments).
- Alokasi Index dengan tiga tingkat tidak berarah (sistem UNIX).

Parameter simulasi adalah:

- Ukuran blok (1, 2 or 4 unit).
- Total ukuran disk (4, 6 or 8 Mega Unit).

128 blok disk pertama tidak digunakan untuk mensimulasikan tugas-tugas administratif OS dan tidak digunakan.

Dua algoritma Manajemen File System adalah:

- **Alokasi dilink dengan tabel alokasi file**, sistem mempertahankan tabel dengan entri sebanyak blok yang dimiliki sistem ((File Allocation Table, FAT), untuk setiap objek hanya referensi ke blok pertama yang disimpan. Referensi ini berada di folder yang berisi objek.

Di tabel setiap blok memiliki referensi ke blok berikutnya sehingga mereka bisa pergi bersama secara blok file berantai dari awal sampai akhir.

Ukuran tabel FAT = ukuran disk / ukuran blok

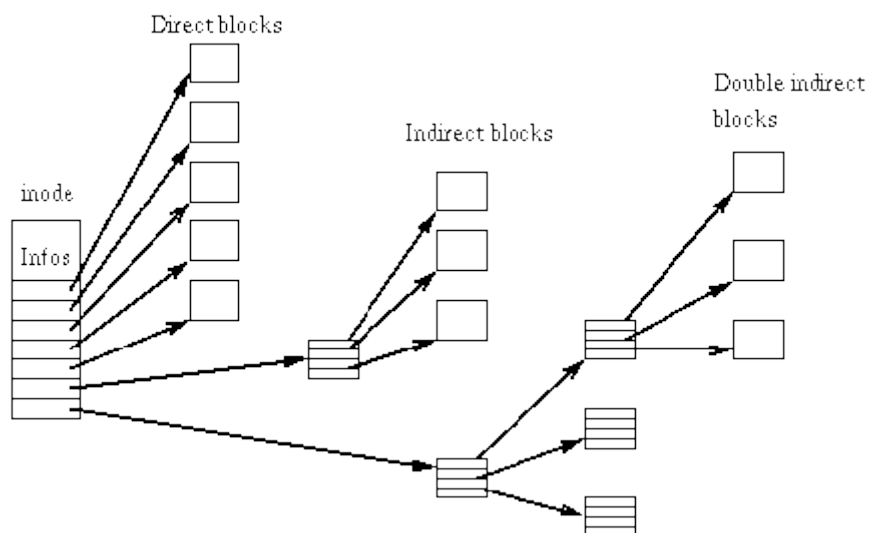
Blok yang dibutuhkan untuk setiap file = ukuran file / ukuran blok

Untuk mempermudah dalam simulasi folder menempati hanya satu blok, dan link tidak (point ke item yang ada dan informasi yang hanya disimpan di direktori yang mengandung link)

- **Alokasi diindex dengan tiga tingkat tidak berarah**, alokasi diindex didasarkan pada struktur yang disebut i-node, dan masing-masing objek terkait dengan i-node. Struktur dari i-node adalah:
  - information fields.
  - 12 pointers to data blocks.
  - 1 pointer to 1 indirect block.
  - 1 pointer to 1 double indirect block.
  - 1 pointer to 1 triple indirect block.

Titik blok tunggal tidak langsung untuk blok data, titik blok ganda tidak langsung untuk blok yang pada titik gilirannya untuk blok data titik blok triple menunjuk ke blok saat itu untuk memblokir yang mengarah ke blok data.

Untuk membuat file baru pertama blok data i-node yang digunakan, maka blok tunggal tidak langsung, double dan triple untuk mengisi informasi yang diperlukan tergantung pada ukuran file. Untuk penyederhanaan folder menempati blok (data), dan link tidak.



Dalam simulasi jumlah maksimum i-node yang dapat dibuat adalah 128, blok tidak langsung memiliki 20 pointer. Dalam tabel di bawah ini menunjukkan bagaimana menghitung blok yang dibutuhkan untuk membuat sebuah file.

	Available Data blocks	Administration Blocks	Total data Blocks	Total Adm. Blocks	Total Blocks
<b>One node</b>	12 blocks	Head	12	0	12
<b>First indirection</b>	Past more ... 20 blocks	1 indirect block	32	1	33
<b>Second indirection</b>	Past more ... 20 * 20 blocks	1 indirect block 1 +20 Indirect b.	432	22	454
<b>Third indirection</b>	Past more ... 20 * 20 * 20 blocks	1 indirect block 1 +20 Indirect b. 1 +20 +400 Indirect b.	8.432	433	8.875

### ***Sistem file Logical:***

Menampilkan sistem file seperti yang terlihat oleh pengguna, pohon folder, dan file, link dan subfolder dalam setiap folder. Root tergantung pada manajer sistem file, UNIX adalah "/" MD-DOS adalah "C:\".

Klik kanan pada objek apapun untuk menampilkan menu pop up yang memungkinkan anda melakukan berbagai tindakan:

- Tambahkan sebuah file pada folder objek yang sama.
- Tambahkan sebuah subfolder pada folder objek yang sama.
- Tambahkan sebuah link pada folder objek yang sama.
- Update objek.
- Hapus objek.

### ***(Device) Sistem file Physical:***

Menunjukkan perangkat (sistem file fisik) dibagi menjadi blok, dan pada alamat disk yang tepat.

Alamat awal (128 unit) dicadangkan untuk tugas-tugas administrasi OS dan tidak digunakan.

Blok diberi warna dengan warna yang berbeda tergantung pada jenis:

<b>Colors:</b>	<b>Block use</b>
<i>black</i>	Administration (File System)
<i>white</i>	Available
<i>Green</i>	Folder
<i>Blue</i>	Soft link
<i>[Other colors]</i>	Occupied by a file UNIX "I" means it is an indirect block

**Melihat folder terpilih:**

Tampilan ini tergantung pada folder yang dipilih sistem file logis menampilkan isi dari folder ini, setiap objek digambar dengan warna yang sesuai: berkas (file bervariasi sesuai dengan warna), link (biru), folder (hijau).

Untuk setiap objek itu menunjukkan referensi ke sistem file fisik. Untuk i-node UNIX objek, untuk MS-DOS awal entri dalam tabel FAT.

Klik kanan pada setiap objek sistem berkas untuk menampilkan menu pop up yang memungkinkan Anda untuk melihat informasi rinci objek.

**Pengaturan.**

Mengatur perilaku simulasi ini:

- *Block Size* (Values 1, 2 or 4 pieces) (Values between 64 and 256 units).
- *Memory size* (values 4, 6 or 8 x 1024 units).
- Algoritma:
  - Alokasi di link dengan file allocation table (*FAT, MS-DOS*).
  - Alokasi di index dengan triple indirection (*UNIX*).

Setiap perubahan algoritma, ukuran blok atau ukuran memori membutuhkan restart simulasi dan menghapus semua objek yang ada.

**Menambah sebuah file:**

File tersebut akan ditambahkan ke folder yang sedang dipilih, untuk setiap file baru anda harus memasukkan informasi berikut:

- Nama (*dibutuhkan*)teks apapun.
- Ukuran file (Range: 1 to 4096 units).
- Warna, warna yang akan menarik elemen grafis yang terkait dengan file tersebut.

**Menambah sebuah folder:**

Folder ditambahkan ke folder yang sedang dipilih,

Hal ini hanya diperlukan untuk menunjukkan nama folder baru (diperlukan).

**Menambah sebuah link:**

Link tersebut ditambahkan ke folder yang sedang dipilih untuk setiap link baru perlu untuk memasukkan informasi berikut:

- Nama (*dibutuhkan*)teks apapun.
- *Soft*, link type, soft link (terpilih) atau hard link (tidak terpilih). MS-DOS hanya membolehkan soft links
- *Target*, link target.

**Informasi rinci dari sebuah objek:**

Dari tampilan folder, pengguna dapat membuka informasi rinci dari objek apapun dalam folder, informasi ini tergantung pada sistem file.

UNIX menunjukkan struktur i-node:

- Informasi, dalam hal ini hanya jumlah link yang menunjuk ke i-node.
- 12 blok data, jumlah blok di mana data tersebut, atau "nihil" jika tidak digunakan.

- Indirections, ke-1, ke-2 dan ke-3, nomor blok dimana blok tidak langsung, atau "nihil" jika tidak digunakan.
  - Untuk setiap indirection juga bisa mendapatkan informasi konkret:
    - Index pointer.
    - Nomor blok Pointed, atau "nihil" jika tidak digunakan.
    - Type: "blok" jika data atau "indirection" jika tidak langsung.

Untuk MS-DOS menampilkan entri tabel FAT untuk objek:

- Nomor blok (FAT tabel entry).
- Objek selanjutnya nomor blok atau "nihil" jika terakhir.

#### ***Data dan statistik:***

Menampilkan informasi perangkat. Informasi ini bervariasi dari satu algoritma yang lain.

Dalam UNIX, menunjukkan tabel dengan semua blok perangkat data, untuk setiap blok:

- Blok, nomor blok (dimulai dari data blok pertama, @128).
- Tipe "data blok" atau "indirect blok."

Dalam MS-DOS, menunjukkan tabel FAT, yang merupakan snapshot dari semua blok perangkat, setiap entri tabel berisi informasi berikut:

- Blok, nomor blok (dimulai dari data blok pertama, @128).
- Next, nomor blok objek berikutnya ketika blok berisi data obyek, "nihil" jika blok berisi blok terakhir dari suatu obyek atau 0 jika tersedia.
- Kondisi dari blok, "used" atau "free."

#### ***Kesalahan dalam simulasi:***

1. **Memori penuh**, sebagai objek baru yang dibuat blok perangkat ditempati tetapi ada batasan jumlahnya. Jika tidak memungkinkan untuk membuat objek baru karena kurangnya ruang perlu untuk mengurangi ukuran yang sudah ada atau menghapusnya.
2. **Ada objek lain dengan nama yang sama** dalam direktori tidak mungkin ada beberapa objek (file, link atau subdirektori) dengan nama yang sama.

#### **Contoh:**

#### **Alokasi Linked dengan FAT. Sistem File Objek**

Dalam contoh ini pohon folder dibuat dengan berbagai jenis objek sistem file: folder, file dan link.

**Peringatan 1:** Warna folder adalah hijau dan link berwarna biru. File warna didefinisikan ketika dibuat.

**Peringatan 2:** Dalam simulasi ukuran folder adalah 1 blok, sedangkan link tidak menempati ruang disk, mereka hanya entri data folder nya.

Tree				Type	Size Units
C:				Folder	1
	folder1			Folder	1
		file1.txt		File	101

		file2.txt		File	200
	folder2			Folder	1
		file3.txt		File	35
		Folder3		Folder	1
			file4.txt	File	120
		link1		Soft link (to file1.txt)	0

Ukuran blok adalah 1 unit.

Komentar:

- File sistem root adalah "C:" dan separator jalan adalah "\"
- Dalam setiap folder baru, dua link secara otomatis dibuat "." dan ".." menunjuk ke direktori itu sendiri dan induknya masing-masing
- Blok pertama dari objek yang berbeda sesuai dengan entri pertama dalam tabel FAT
- Jika file yang direferensikan oleh link dihapus (link1 -> file1.txt), Link tetap tapi menjadi tidak dapat digunakan
- Jika folder dihapus, semua benda yang menggantung dari itu juga akan dihapus

#### **Alokasi Linked dengan FAT (ukuran blok: 4 units)**

Dalam contoh ini pohon folder dibuat dengan berbagai jenis objek sistem file: folder, file dan link.

**Peringatan 1:** Warna folder 'adalah hijau dan link berwarna biru. File warna didefinisikan ketika dibuat.

**Peringatan 2:** Dalam simulasi ukuran folder adalah 1 blok, sedangkan link tidak menempati ruang disk, mereka hanya entri data folder nya.

Tree				Type	Size Units	Blocks
C:				Folder	1	1
	folder1			Folder	1	1
		file1.txt		File	101	26
		file2.txt		File	200	50
	folder2			Folder	1	1
		file3.txt		File	35	9.
		Folder3		Folder	1	1
			file4.txt	File	120	30
		link1		Soft link (to file1.txt)	0	0

Ukuran blok adalah 4 units.



Komentar:

- Blok terakhir dari file "file1.txt" dan "file3.txt" tidak penuh, tapi ruang tambahan tidak dapat dimanfaatkan, seluruh blok yang digunakan
- (Dibandingkan dengan Contoh 1) Folder berisi informasi yang sama namun memakan lebih banyak ruang, dalam contoh ini 4 unit

### Alokasi Indexed (UNIX). Referensi ke i-nodes

Dalam contoh ini pohon folder dibuat dengan berbagai jenis objek sistem file: folder, file dan link.

Pada UNIX setiap objek terkait dengan struktur yang disebut i-node, referensi ini berada di entri folder di mana objek berada.

Link merujuk ke objek lain, dan berkaitan dengan i-node yang sama dari objek yang mereka berhubungan, i-node melacak benda yang berhubungan dengan mereka.

**Peringatan 1:**Warna folder 'adalah hijau dan link berwarna biru. File warna didefinisikan ketika dibuat.

**Peringatan 2:**Dalam simulasi ukuran folder adalah 1 blok, sedangkan link tidak menempati ruang disk, mereka hanya entri data folder nya.

**Peringatan 3:**Struktur dari i-node

- Informasi, yang misalnya berisi sejumlah referensi ke i-node
- 12 pointer ke blok data
- 1 pointer ke blok tidak langsung.
- 1 pointer ke blok tidak langsung ganda.
- 1 pointer ke blok tidak langsung tiga.

Tree			Type	Size Units
/			Folder	1
	folder1		Folder	1
		file1.txt	File	101
		file2.txt	File	200
		link1	Hard link (to file1.txt)	1
	folder2		Folder	1
		file3.txt	File	460
		Folder3	Folder	1
		file4.txt	File	120
		link2	Hard link (to link1)	0

Ukuran blok adalah 1 unit.

Komentar

- Objek-objek yang terkait, link2 -> link1 -> file1.txt, merujuk sama i-node.
- Pembukaan informasi yang sesuai i-node (klik kanan pada tampilan folder, pop up menu) menunjukkan jumlah link yang 3, sehingga memiliki referensi 3: file1.txt, link1 dan link2

### Alokasi Indexed (UNIX). Indirections

Dalam contoh ini pohon folder dibuat dengan berbagai jenis objek sistem file: folder, file dan link.

Pada UNIX setiap objek terkait dengan struktur yang disebut i-node, referensi ini berada di entri folder di mana objek berada.

**Peringatan 1:** Warna folder 'adalah hijau dan link berwarna biru. File warna didefinisikan ketika dibuat.

**Peringatan 2:** Dalam simulasi ukuran folder adalah 1 blok, sedangkan link tidak menempati ruang disk, mereka hanya entri data folder nya.

**Peringatan 3:** Struktur dari i-node

- Informasi, yang misalnya berisi sejumlah referensi ke i-node
- 12 pointer ke blok data
- 1 pointer ke blok tidak langsung.
- 1 pointer ke blok tidak langsung ganda.
- 1 pointer ke blok tidak langsung tiga.

Untuk setiap blok file data baru diisi di urutan i-node, pertama langsung, maka indirections tunggal, dua dan tiga.

Tree				Type	Size Units
/				Folder	1
	folder1			Folder	1
		file1.txt		File	101
		file2.txt		File	200
		link1		Hard link (to file1.txt)	1
	folder2			Folder	1
		file3.txt		File	460
		Folder3		Folder	1
			file4.txt	File	120
		link2		Hard link (to link1)	0

Ukuran blok adalah 1 unit.

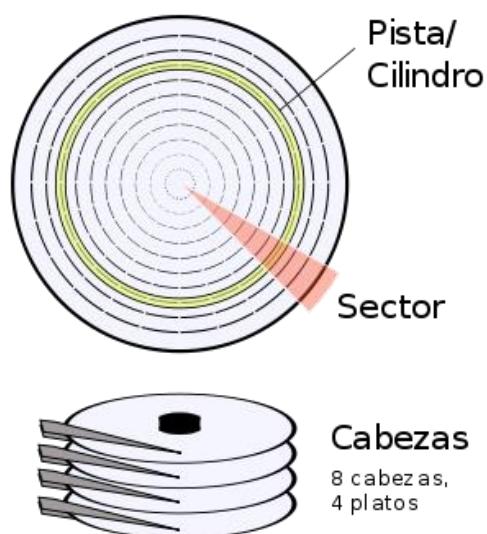
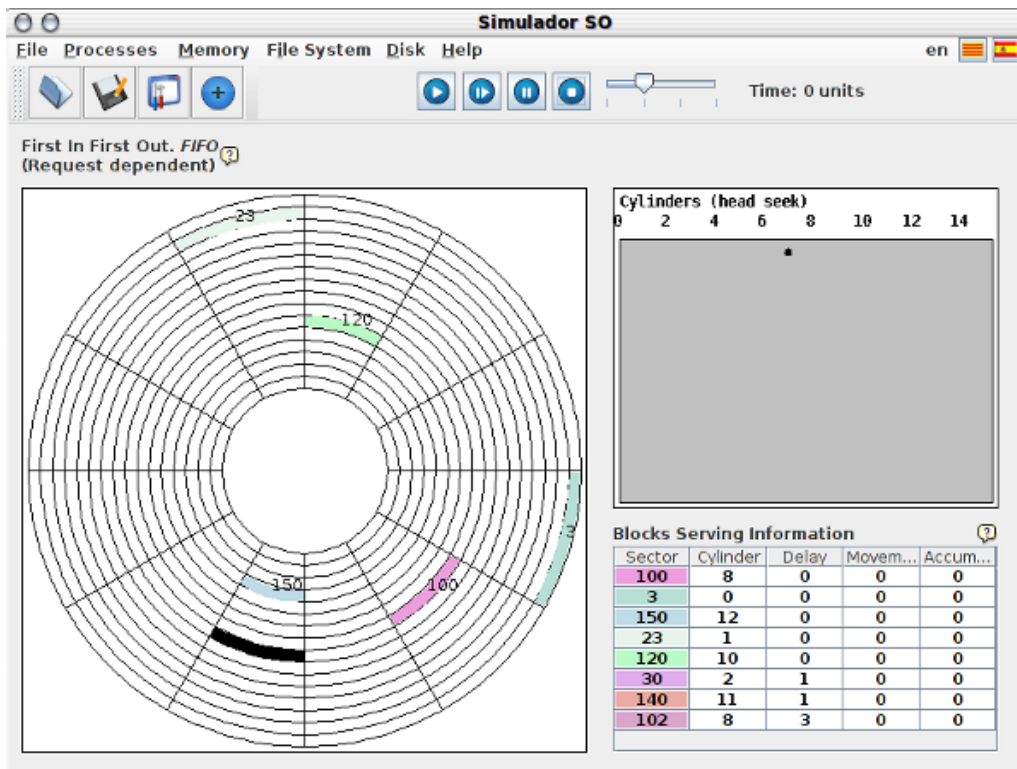
Komentar:

- Blok tidak langsung ditampilkan dengan karakter "I". Blok ini berisi data tidak ada, hanya pointer ke blok data atau blok tidak langsung lainnya

- Dua blok tidak langsung diikuti ("I", "I") berarti bahwa indirection kedua dimulai mengisi, dan tiga blok tidak langsung ("I", "I", "I") ketiga
- Hal ini dimungkinkan untuk melacak indirections dan blok data dengan membuka informasi yang sesuai i-node (klik kanan pada tampilan folder, pop up menu, dan kemudian klik kanan lagi pada blok tidak langsung).

### Disk scheduling

Berikut adalah kebijakan penjadwalan disk di dalam Sistem Operasi I / O manajemen perangkat.



Meskipun heterogenitas mereka, optical disk atau magnet, kaku atau fleksibel, read-only atau membaca / menulis CD, DVD atau HDD misalnya, keseluruhan dapat diperlakukan sama.

Disk diatur dalam sector (*sector*), mereka dikelompokkan ke dalam trek(*tracks*), trek yang di satu sisi atau permukaan di piring, dan setiap disk dapat memiliki beberapa piring-piring (*platters*).

Himpunan trek yang berada dalam posisi kepala yang sama disebut silinder. Piring-piring berputar terus-menerus dan masing-masing memiliki satu kepala tepat di atas, yang membaca dan menulis sektor, kepala bergerak tegak lurus ke trek.

Sektor disk yang direferensikan berurutan, sektor pertama adalah 0, dan itu adalah di lagu pertama dari piring pertama dalam silinder luar.

Dalam geometri simulasi disk: piring, 16 silinder, 12 sektor per silinder, benar-benar 192 sektor atau blok data.

Sistem Operasi terus menerima permintaan untuk menulis dan membaca dari disk, bagaimana melayani permintaan tersebut ditentukan oleh kebijakan penjadwalan disk.

Sebuah kebijakan penjadwalan yang baik harus meminimalkan gerakan head, yang berarti kehidupan yang lebih panjang untuk perangkat, dan permintaan-to-respon waktu.

Dalam simulasi ini ada tersedia kebijakan penjadwalan berikut:

- *FIFO. First In First Out* penjadwalan ini cukup sederhana, melayani permintaan agar tepat dari kedatangan. Head selalu bergerak mencari permintaan berikutnya di mana pun itu, jangan mencoba untuk meminimalkan gerakan head tapi itu benar karena tidak memprioritaskan permintaan apapun.
- *LIFO. Last In First Out* tidak seperti FIFO, melayani permintaan dalam urutan terbalik kedatangan, yang terakhir datang pertama dilayani. Permintaan terbaru dilayani segera, tetapi dapat menyebabkan tidak dilayani, permintaan pertama tidak akan pernah dilayani jika yang baru tiba terus-menerus.
- *STF. Shortest (Seek) First Time* selalu melayani permintaan terdekat pertama yang posisi head ini. penjadwalan seperti meminimalkan gerakan kepala, tetapi juga dapat menyebabkan tidak dilayani.
- *SCAN*. Juga disebut "Elevator", head bergerak dari silinder terluar untuk terdalam ketika mencapai akhir itu berubah arah dan kembali ke awal, dan seterusnya tanpa batas. Menangani permintaan ketika mereka muncul selama gerakan.
- *C-SCAN. Circular SCAN*. Sama dengan SCAN, Perbedaannya adalah bahwa ketika mencapai silinder bagian di disk akhir ini, bergerak langsung ke awal dan dimulai lagi, jadi selalu menangani permintaan dalam arah yang sama. Meningkatkan salah satu kelemahan SCAN, yang berjalan di atas track yang baru saja dilihat ketika perubahan arah.
- *LOOK*. Adalah variasi yang meningkatkan penjadwalan SCAN, perbedaannya adalah bahwa hal itu hanya naik ke permintaan terakhir dan tidak sampai silinder terakhir. Pada saat itu mendeteksi bahwa tidak ada lagi permintaan ke depan, maka perubahan arah.
- *C-LOOK. Circular LOOK*. Termasuk kedua LOOK dan perbaikan C-SCAN hanya pergi ke permintaan terakhir dan juga saat mencapai luar, bergerak ke yang pertama dan mulai lagi, itu selalu melayani permintaan dalam arah yang sama.

#### **Disk:**

Menunjukkan sektor dan silinder divisi disk. Sektor 0 (pertama) adalah di luar silinder, atas dan kanan vertikal. Setiap saat simulasi posisi kepala ditampilkan hitam, dan permintaan dalam warna yang sesuai Sementara simulasi dihentikan, klik kanan pada permintaan untuk menampilkan menu pop up yang memungkinkan Anda untuk mengubah dan menghapus.

### **Grafik gerakan head:**

grafik mewakili gerakan head melalui silinder sebagai permintaan dilayani. Dalam silinder sumbu x dan ordinat waktu. Setiap permintaan disajikan ditandai dengan titik (warna yang sesuai) menunjukkan silinder di mana sektor ini disajikan dan waktu, poin bergabung dengan garis-garis yang mewakili gerakan head melalui silinder.

### **Data dan statistik:**

Lihat semua permintaan, informasi dalam tabel adalah sebagai berikut:

- *Sector*, jumlah sektor yang diminta.
- *Cylinder*, di mana sektor ini.
- *Delay*, waktu yang dibutuhkan untuk permintaan tiba. Misalnya, permintaan 3 unit tertunda, masuk ke dalam antrian permintaan pada simulasi waktu 3.
- *Movement*, hanya untuk permintaan dilayani, menunjukkan gerakan head sejak permintaan sebelumnya, yaitu jumlah total silinder yang sudah naik sampai permintaan saat silinder.
- *Accumulated* Gerakan Jumlah silinder akumulasi sejak simulasi dimulai.

Permintaan dapat di tiga negara yang berbeda:

- *Served*, telah disajikan.
- *Waiting*, mereka antri menunggu untuk dilayani.
- *Delayed*, permintaan yang datang kemudian.

Disajikan ditampilkan pertama, maka mereka yang menunggu dan akhirnya mereka yang belum tiba.

Sementara simulasi dihentikan, klik kanan pada permintaan untuk menampilkan menu pop up yang memungkinkan Anda untuk mengubah dan menghapus.

### **Pengaturan:**

Set perilaku simulasi:

- *Head Start position*: pengguna dapat memilih mana head awalnya (Range: 0-192 sektor).
- Algoritma:
  - *FIFO*, First in first out.
  - *LIFO*, Last in First Out.
  - *STF*, Shortest Seek Time First.
  - *SCAN*.
  - *C-SCAN*, Circular SCAN.
  - *LOOK*.
  - *C-LOOK*, LOOK circular.

Pengaturan hanya tersedia saat simulasi dihentikan

### **Menambahkan permintaan:**

Permintaan dapat ditambahkan hanya sementara simulasi dihentikan, untuk setiap permintaan baru harus memasukkan informasi berikut:

- *Sector*, nomor sektor di mana permintaan dihasilkan (Nilai: 1 - 192).
- *Delay* ketika permintaan akan memasuki antrian permintaan, jika 0 ditambahkan langsung ke antrian, jika tidak maka ditambahkan kemudian (Range: 1-100).

- *Color*, warna yang akan menarik elemen grafis yang terkait dengan permintaan.

Selain itu, sistem menghitung dan menunjukkan silinder yang sesuai dengan sektor yang diminta.

### **Common Tasks**

#### **Membuka dan menyimpan simulasi:**

Sementara simulasi dihentikan, pengguna dapat menyimpan simulasi mereka dan membuka mereka nanti untuk menyelesaikan, review, dll ...

Menyimpan semua objek simulasi dan pengaturan.

Hal ini tidak mungkin untuk membuka atau menyimpan simulasi saat menjalankan OS Sim sebagai Applet memiliki beberapa keterbatasan karena masalah keamanan java.

#### **Kontrol waktu**

Untuk mengelola kemajuan simulasi, run normal, langkah demi langkah (satuan waktu setiap langkah), stop dan restart simulasi.

Selain itu Anda dapat bervariasi tingkat kecepatan dari 0,5 detik untuk 2 detik per satuan waktu.

simulasi waktu selalu ditampilkan.

#### **Contoh:**

#### **Fair scheduling. FIFO**

Permintaan untuk sektor disk yang disajikan dalam rangka kedatangan tepat.

**Peringatan:** geometri harddisk adalah salah satu piring, 16 silinder, 12 sektor per silinder, 192 sektor total.

Sector	Cylinder
100	8
3	0
150	12
23	1
120	10

Head awalnya diposisikan di sektor 0 (Cylinder 0).

Semua permintaan tiba di urutan tabel pada saat 0.

Komentar:

- Head selalu bergerak mencari permintaan berikutnya
- Head melewati silinder yang berisi permintaan, tetapi tidak dilayani karena mereka bukan yang akan dicari

#### **STF scheduling contoh Starvation**

Permintaan paling dekat dengan head dilayani pertama, hal ini dapat menyebabkan bahwa sementara permintaan tiba dekat dengan head, yang lebih jauh tidak dilayani (Starvation)

**Peringatan:** geometri harddisk adalah salah satu piring, 16 silinder, 12 sektor per silinder, 192 sektor total.

Sector	Cylinder	Delay
170	14	0

150	12	0
120	10	0
40	3	1
20	1	2
35	2	5
2	0	6
10	0	8
50	4	15
0	0	15

Head awalnya diposisikan di sektor 50 (Cylinder 4)

Semua permintaan tiba di urutan tabel, menurut delay yang ditunjukkan.

Komentar:

- Permintaan pertama berada di silinder bagian dalam, dan tidak melayani sebagai permintaan baru yang datang dekat dengan head
- Situasi ini bisa dilanjutkan tanpa batas

### Elevator (SCAN)

Gerakan head tidak tergantung pada permintaan, scan permintaan di kedua arah, permintaan dilayani seperti yang ditemukan.

**Peringatan:** geometri harddisk adalah salah satu piring, 16 silinder, 12 sektor per silinder, 192 sektor total.

Sector	Cylinder
100	8
3	0
150	12
23	1
120	10

Head awalnya diposisikan di sektor 90 (Cylinder 7), bergerak maju silinder terdalam.

Semua permintaan tiba di urutan meja pada saat 0.

Komentar:

- Permintaan terdalam di silinder 12, tapi head terus lanjut sampai silinder terakhir
- Pada saat mencapai silinder terakhir, seperti yang baru saja dilayani terdalam, dan mereka yang tetap berada di silinder terluar, tetapi head berubah arah dan bergerak dari dalam ke luar.

### Perbaikan elevator 1. Circular SCAN

Gerakan head tidak tergantung pada permintaan, scan dalam satu arah, dari luar ke silinder bagian

dalam, permintaan dilayani seperti yang ditemukan.

**Peringatan:** geometri harddisk adalah salah satu piring, 16 silinder, 12 sektor per silinder, 192 sektor total.

Sector	Cylinder
100	8
3	0
150	12
23	1
120	10

Head awalnya diposisikan di sektor 90 (Cylinder 7), bergerak maju silinder terdalam.

Semua permintaan tiba di urutan meja pada saat 0.

Komentar:

- Permintaan terdalam dalam silinder 12, tapi head terus sampai silinder terakhir.
- (Peningkatan SCAN) Pada saat mencapai silinder terakhir, seperti yang baru saja menjabat permintaan terdalam, dan orang-orang yang tetap berada di silinder terluar, sehingga kepala bergerak langsung ke silinder pertama dan mulai lagi

### **Perbaikan elevator 2. Look circular**

Gerakan head tidak tergantung pada permintaan, dan scan dalam satu arah, dari luar ke silinder bagian dalam, dan hanya mencapai permintaan terdalam, permintaan dilayani seperti yang ditemukan.

**Peringatan:** geometri harddisk adalah salah satu piring, 16 silinder, 12 sektor per silinder, 192 sektor total.

Sector	Cylinder
100	8
3	0
150	12
23	1
120	10

Head awalnya diposisikan di sektor 90 (Cylinder 7), bergerak maju silinder terdalam.

Semua permintaan tiba di urutan meja pada saat 0.

Komentar:

- (Peningkatan SCAN) Permintaan terdalam di silinder 12, ketika disajikan, head tidak berlanjut sampai silinder terakhir
- (Peningkatan SCAN) Dalam saat mencapai permintaan terakhir, seperti yang baru saja dilayani terdalam, dan orang-orang yang tetap berada di silinder terluar, sehingga head bergerak langsung ke permintaan terluar dan dimulai lagi

### **Tugas**

1. Jawab pertanyaan pada latihan OS-SIM Linked Allocation FAT
2. Jawab pertanyaan pada latihan OS-SIM Disk Scheduling, Algorithm Comparison



## PERTEMUAN 7


### Group Policy Editor dan Registry Editor

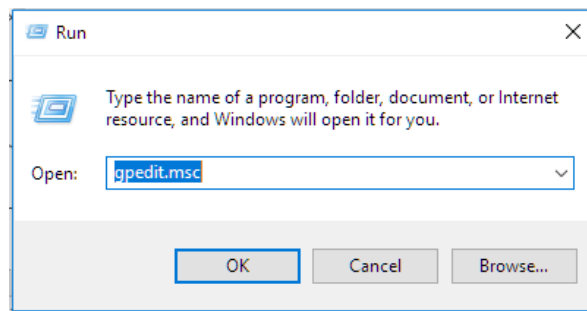
#### TUJUAN PRAKTIKUM

- Praktikan dapat mengatur Group Policy suatu system operasi
- Praktikan dapat memahami fungsi register dan group policy pada suatu system operasi
- Praktikan dapat mengatur register

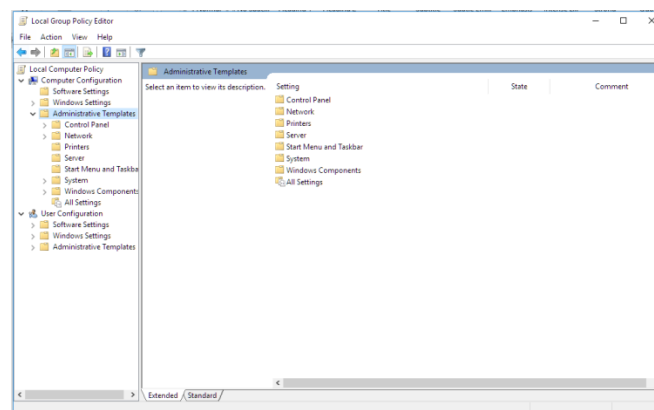
#### GROUP POLICY EDITOR

Group Policy Editor (Gpedit.msc) adalah sebuah program yang telah disediakan Windows untuk mengkonfigurasi semua pengaturan yang terdapat pada sistem operasi. Beberapa fungsi dari program ini adalah sebagai cara alternatif, cepat dan rahasia untuk mengganti pengaturan pada Windows. Fungsi gpedit.msc adalah untuk melihat, membuat dan memodifikasi Group Policy, yang digunakan untuk mengelola beberapa pengaturan di dalam setiap pengguna dan lingkungan Windows serta menyimpannya di dalam registry.

Untuk membuka gpedit, cukup buka RUN (  + R ) lalu ketikkan gpedit.msc, atau buka saja di C:\Windows\System32\gpedit.msc



Berikut adalah tampilan dari Group Policy Editor



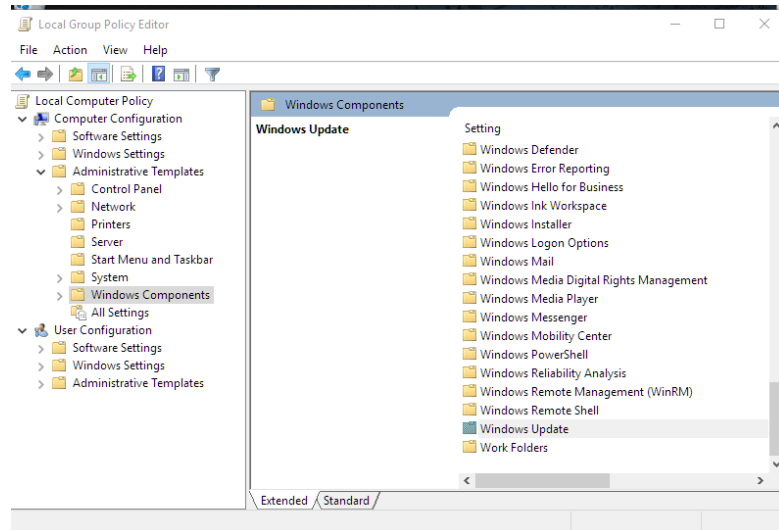
Dalam gpedit, terdapat 7 kategori yang dapat kita konfigurasi yaitu : Control Panel, Network, Printers, Server, Start Menu and Taskbar, System dan Windows Component.

Konfigurasi utama dalam semua policy ada 3 pilihan yaitu :

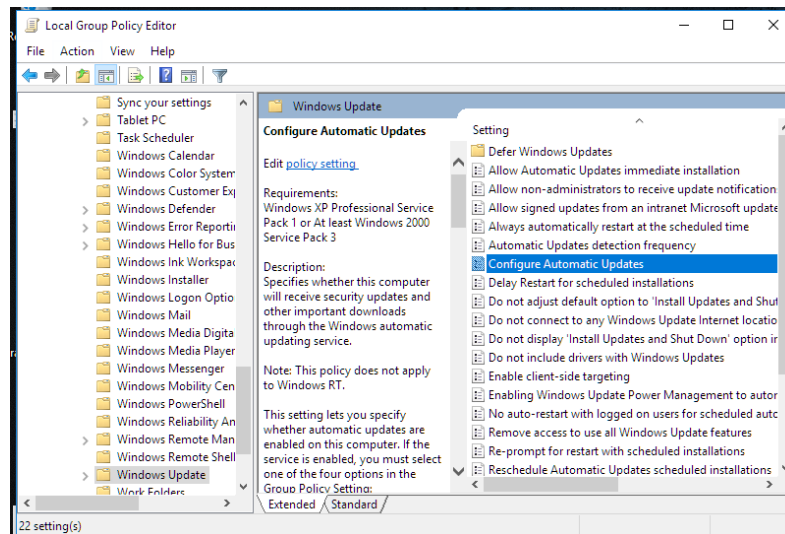
1. Enabled ( Aktif )
2. Disabled ( Non – Aktif )
3. Not Configured ( Tidak Diatur )

Misalnya kita akan mengatur Automatic Updates pada Windows Update

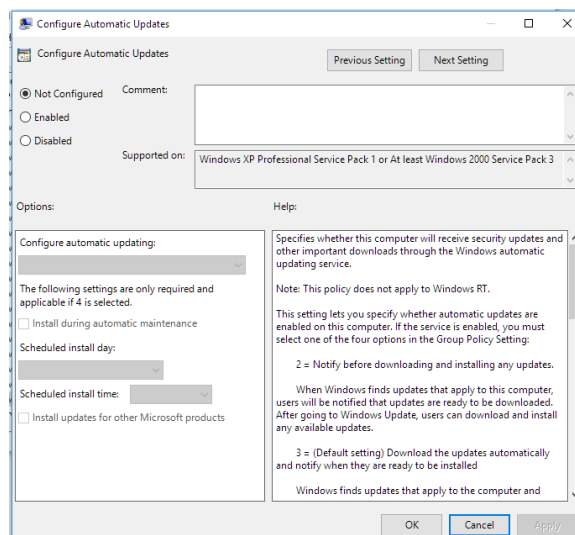
1. Buka Group Policy Editor.
2. Pilih Computer Configuration → Administrative Templates → Windows Components.
3. Pilih Windows Update.



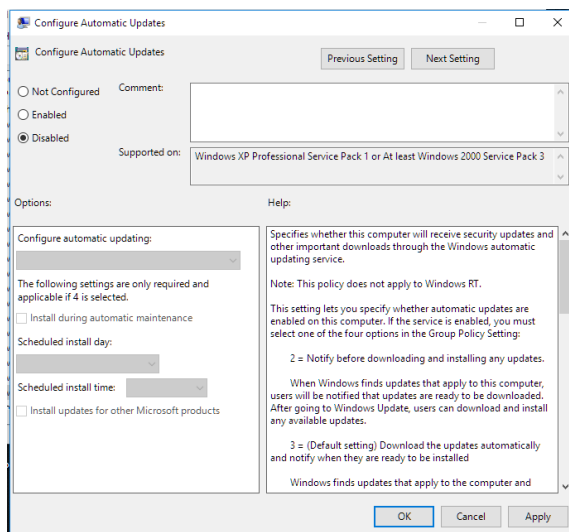
4. Double click pada “Configure Automatic Updates”.



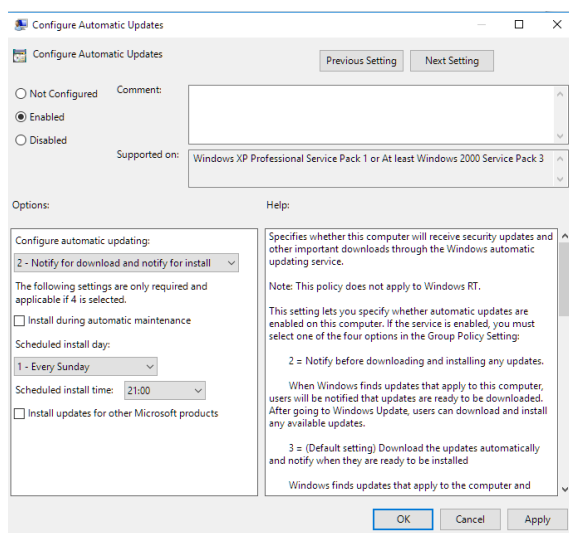
5. Akan muncul tampilan konfigurasi untuk policy “Configure Automatic Updates”.



6. a. Karena Automatic Updates akan dinonaktifkan, ubah Not Configured menjadi Disabled lalu pilih OK.



7. Jika ingin menyesuaikan opsi Automatics Update, pilih Enabled, sesuaikan Options-nya. Jika perlu bantuan, terdapat Help pada sisi kanan bawah.

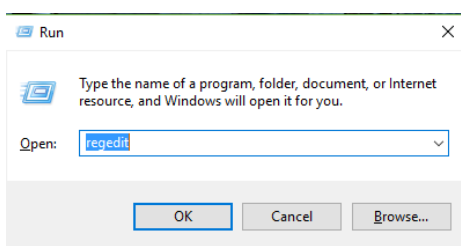


8. Jika sudah sesuai, klik OK.

## REGISTRY EDITOR

Registry Editor (regedit) adalah database hirarki yang digunakan untuk mengatur informasi yang dibutuhkan oleh Sistem Windows, regedit menampilkan Registry ke dalam format yang hampir mirip dengan Windows Explorer yang sering anda gunakan. Di dalam regedit tersebut anda dapat mengedit informasi untuk sistem semau anda, tetapi tidak bisa sembarangan. Dan harus mengetahui apa fungsi dari setiap informasi yang ada pada regedit.

Untuk membuka regedit, cukup buka RUN (  + R ) lalu ketikkan regedit.

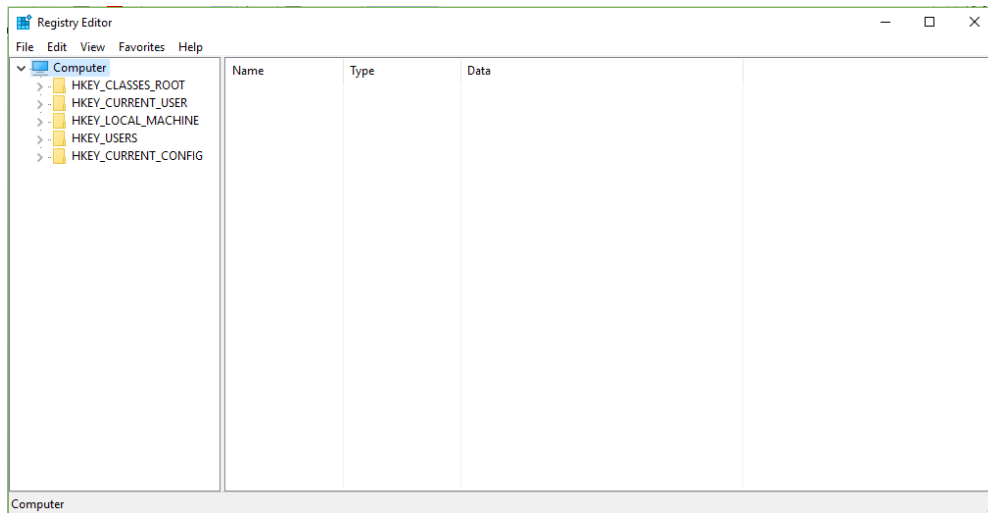


Struktur registry agak mirip dengan struktur direktori dalam sistem berkas. Selain itu, registry juga dapat diakses dengan menggunakan sintaks yang sama dengan cara mengakses berkas, dengan menggunakan karakter garis miring terbalik (backslash) untuk menandakan tingkatan hierarkis.

Susunannya adalah seperti \\. Sebagai contoh :

**MyComputer\HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows**

Sintaks diatas merujuk kepada sebuah subkey yang memiliki nama “Windows” yang terdapat di dalam subkey Microsoft, yang terdapat di dalam key dengan nama Software, yang terdapat di dalam subtree HKEY\_LOCAL\_MACHINE.



Terdapat 5 macam key utama pada registry, yaitu :

1. **HKEY\_LOCAL\_MACHINE**  
Berisi file association dan COM object registration
2. **HKEY\_CURRENT\_USER**  
Berisi data association dengan akun yang sedang login
3. **HKEY\_USERS**  
Berisi informasi tentang sistem
4. **HKEY\_CLASSES\_ROOT**  
Berisi informasi tentang semua akun yang ada di komputer
5. **HKEY\_CURRENT\_CONFIG**  
Berisi informasi tentang profil komputer yang sedang aktif

#### CONTOH PENGGUNAAN REGEDIT

1. Merubah Nama Pemilik Komputer  
Regedit\MyComputer → Edit → FIND → RegOwner atau RegCompany → isikan Nama
2. Membersihkan Page file saat shut down demi keamanan

HKEY\_LOCALMACHINE\SYSTEM\CurrentControlSet\Control |Session Manager. Edit value pada key Clear-PageFileAtShutdown menjadi 1. Konsekuensinya, proses shut down akan berlangsung sedikit lebih lama.

3. Menghilangkan tanda panah pada icon shortcut di desktop  
HKEY\_CLASSES\_ROOT\Lnkfile. Hapus value IsShortcut.
4. Untuk mengurangi waktu booting yang terasa lama  
Atur di BIOS agar booting dimulai dari harddisk baru kemudian CD atau floppy drive pada pilihan berikutnya.
5. Untuk mempercepat tampilan Start Menu  
HKEY\_CURRENT\_USER\Control Panel\Desktop. Klik ganda value yang bernama MenuShowDelay ganti angka yang ada menjadi 100.

## **TUGAS**

1. Buat tabel perubahan Group Policy Editor. Tuliskan nama policy, path, fungsi, dan perubahan yang terjadi dalam bentuk before – after.

**Policy**

**Path**

**Fungsi**

**Before**

**After**

## PERTEMUAN 8

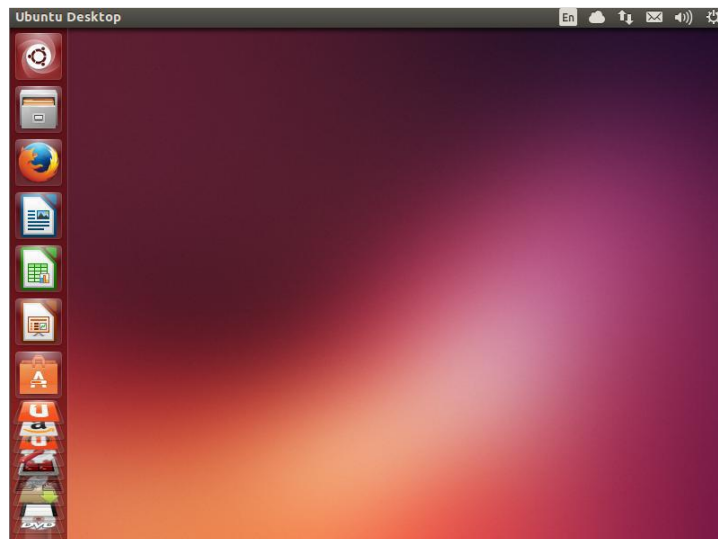
### TERMINAL LINUX UBUNTU DAN SECURITY

#### TUJUAN PRAKTIKUM

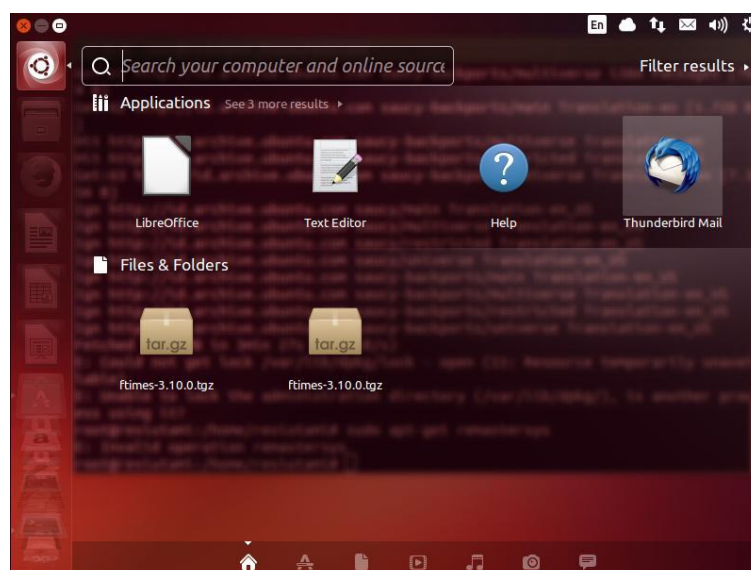
- Praktikan dapat mengoperasikan system operasi menggunakan command pada terminal
- Praktikan dapat memahami struktur dan cara kerja terminal pada ubuntu
- Praktikan dapat memahami struktur file system pada ubuntu

#### GUI

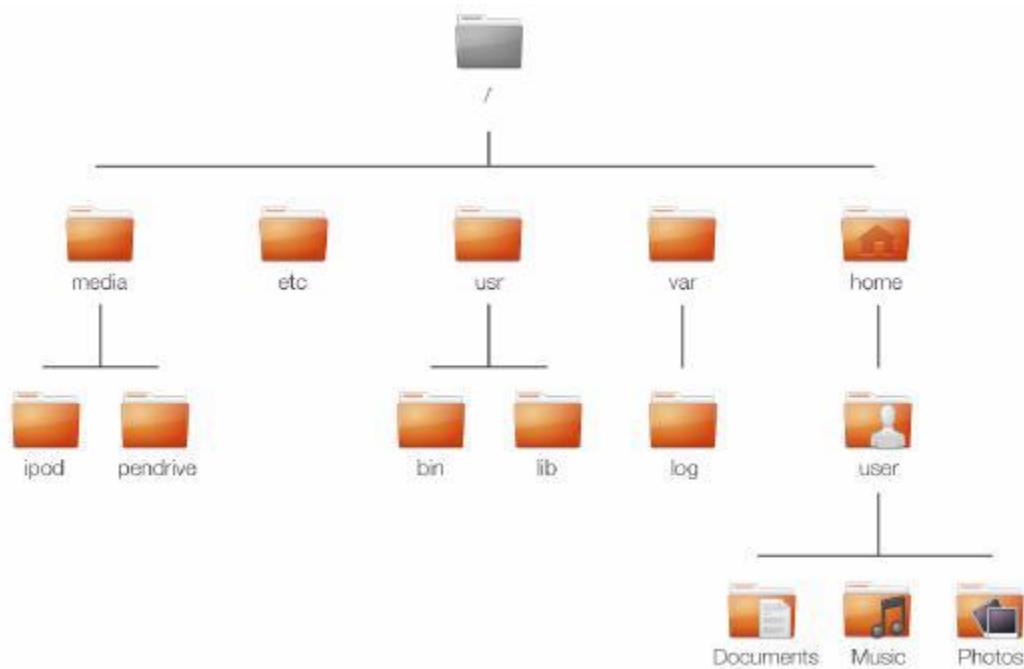
Pada dasarnya Ubuntu sama dengan sistem operasi yang lain, seperti Windows atau Macintosh. Hal ini, dikarenakan menggunakan konsep yang sama yaitu berbasis GUI (*Graphical UserInterface*). Pada setiap sistem operasi ada yang berbasis GUI (ada tampilan) atau berbasis Console (layar hitam putih).



Untuk menjalankan suatu aplikasi, bisa memulai dengan mengklik dash home yang terletak pada menu paling atas ubuntu. Dash home bisa berfungsi sebagai tombol start pada Windows 7 atau search pada windows 8.



## STRUKTUR SISTEM FILE DI UBUNTU



1. /bin & /sbin : tempat sistem aplikasi yang penting (sama seperti C:\Windows).
2. /etc : tempat konfigurasi file.
3. /home : setiap user mempunyai subdirectory untuk file pribadi (contoh /home/resiutami, sama dengan C:\Users atau C:\Documents and Settings).
4. /lib : file library, sama seperti file .dll di Windows.
5. /media : Removable media (CDROM dan USB drives) akan diletakkan di direktori ini.
6. /root : tempat file root user.
7. /usr : Pronounced "user," adalah tempatnya file program (sama dengan C:\Program Files).
8. /var/log : berisi banyak log files yang ditulis oleh banyak aplikasi.

## COMMAND LINE LINUX

Berikut ini adalah kumpulan perintah-perintah yang digunakan pada OS Linux :

Perintah	Keterangan
any_command --help	Menampilkan keterangan bantu tentang pemakaian perintah. "--help" sama dengan perintah pada DOS "/h".
ls	Melihat isi file dari direktori aktif. Pada linux perintah "dir" hanya berupa alias dari perintah "ls". Untuk perintah "ls" sendiri sering dibuatkan alias "ls -color", agar pada waktu di "ls" ditampilkan warna-warna sesuai dengan file-filenya, biasanya hijau untuk execute, dsb.
ls -al	Melihat seluruh isi file pada direktori aktif beserta file hidden, lalu ditampilkan layar perlayar.
cd (directory)	Change directory. Menggunakan "cd" tanpa nama direktori akan menghantarkan anda ke home direktori.

	Dan "cd -" akan menghantarkan anda ke direktori sebelumnya.
cp (source destination)	Mengcopy suatu file.
mcopy	Mengcopy suatu file dari/ke dos filesystem. Contoh mcopy a:autoexec.bat/junk . Gunakan "man mtools" untuk command yang sejenis : mdir, mcd, mren, move, mdel, mmd, mrd, mformat ....
mv source destination	Memindahkan atau mengganti nama file
ln -s source	Membuat Symbolic Links, contoh ln -sf /usr/X11R6/bin/XF86_SVGA /etc/X11/X, membuat Symbolic link dari file XF86_SVGA ke X
rm files	Menghapus file
mkdir directory	Membuat direktori baru
rmdir directory	Menghapus direktori yang telah kosong
rm -r files	Menghapus file, direktori dan subdirektorinya. Hati-hati menggunakan perintah ini apabila anda login sebagai root, karena root dengan mudah dapat menghapus seluruh file pada sistem dengan perintah di atas, tidak ada perintah untuk un-deletedi Linux (belum)
more	Untuk melihat isi suatu file, dengan tambahan perintah more, maka isi file tersebut ditampilkan layar per layar.
less filename	Melihat suatu file layar per layar, dan tekan tombol "q" apabila ingin keluar,pico filename Edit suatu text file.
pico -w filename	Edit suatu text file, dengan menonaktifkan fungsi word wrap, sangat berguna untuk mengedit file seperti /etc/fstab.
lynx file.html	Melihat file html atau browse ke net dengan text mode, dimana gambar/image tidak dapat ditampilkan, tapi lynx adalah suatu browser yang sangat cepat, sangat berguna bila anda hanya menginginkan suatu artikel tanpa image.
tar -zxvf file.tar.gz	Meng-antar sebuah file tar sekaligus meng-uncompress file tersebut (*.tar.gz or *.tgz), untuk meletakkannya direktori yg diinginkan tambahkan option -C direktori, contoh tar -zxvfilename.tar.gz -C /opt (meletakkan file tersebut di direktori /opt)
tar -xvf filename.tar	Meng-antar sebuah file tar yang tidak terkompres (*.tar).
gunzip filename.gz	Meng-uncompress sebuah file zip (*.gz" or *.z). dengan menggunakan gzip (juga zip atau compress) jika anda menginginkan mengompress file.
bunzip2 filename.bz2	Meng-uncompress file dengan format (*.bz2) dengan utiliti "bzip2", digunakan pada file yang besar.
unzip filename.zip	Meng-uncompress file dengan format (*.zip) dengan utiliti "unzip" yang kompatibel dengan pkzip for DOS.
find / -name "filename"	Mencari "namafile" pada komputer anda dimulai dengan direktori /. Namafile tersebut mungkin saja berisi wildcard (*,?).
locate filename	Mencari file dengan string "filename". Sangat mudah dan cepat dari perintah di atas.
pine	Email reader yang sangat mudah digunakan, dan menjadi favorit banyak pemakai mesin Unix. Atau anda bisa pakai email yang sangat customize yaitu "mutt",
talk username1	Berbicara dengan keyboard dengan user lain yg sedang login pada mesin kita (atau gunakan "talk username1 @machinename" untuk berbicara dengan komputer lain) . Untuk menerima undangan percakapan, ketikkan"talk username2". Jika seseorang mencoba untuk berbicara dengan anda dan itu dirasakan mengganggu, anda bisa menggunakan perintah"mesg n" untuk menolak pesan tersebut. Dan gunakan perintah "who" atau "rwho" untuk melihat siapa user yang mengganggu tersebut.
mc	Menjalankan "Midnight Commander" sebagai file manager, cepat dan bagus.



telnet server	Untuk menghubungkan komputer kita ke komputer lain dengan menggunakan protokol TELNET. Gunakan nama mesin atau Nomor IP mesin, dan anda akan mendapatkan prompt login name dari mesin tersebut, masukkan passwordnya, oh ya .. anda juga harus punya account di mesin remote tersebut. Telnet akan menghubungkan anda dengan computer lain dan membiarkan anda untuk mengoperasikan mesin tersebut. Telnet sangat tidak aman, setiap yang anda ketik menjadi "open text", juga dengan password anda! Gunakan ssh alih-alih telnet untuk mengakses mesin secara remote.
rlogin server	Menghubungkan anda ke komputer lain. Loginname dan password, tetapi apabila account anda tersebut telah dipakai, maka anda akan mendapatkan pesan kesalahan pada password anda. Sangat tidak aman juga, gunakan ssh sebagai gantinya. rshserver (=remote shell) Jalan lain untuk menghubungkan anda ke remote machine. Apabila loginname/password anda sedang dipakai di remotemesin tsb, maka password anda tidak akan berlaku. Idem dengan rlogin, gantikan dengan ssh. ftpserver Ftp ke mesin lain, ini sangat berguna untuk mengcopy file ke/dari remote mesin. Juga tidak aman, gunakan scp dari keluarga ssh sebagai gantinya.
minicom	Program Minicom (dapat dikatakan seperti "Procomm/Hyperterminal for Linux").
./program_name	Menjalankan program pada direktori aktif, yang mana tidak terdapat pada PATH anda
xinit	Menjalankan X-window server (tanpa windows manager).
startx	Menjalankan X-window server dan meload default windows manager. Sama seperti perintah "win" under DOS dengan Win3.1
startx -- : 1	Menjalankan sesi X-windows berikutnya pada display 1 (default menggunakan display 0). Anda dapat menjalankan banyak GUI terminal secara bersamaan, untuk pindah antar GUI gunakan <Ctrl><Alt><F7>, <Ctrl><Alt><F8>, etc, tapi ini akan lebih banyak memakan memori.
x-term	(pada X terminal), menjalankan X-windows terminal. Untuk keluar ketikkan "exit"
xboing	(pada X terminal). Sangat lucu deh ....., seperti games-games lama .....
gimp	(pada X terminal) Program image editor yang sangat bagus, bisa disamakan dengan Adobe Photoshop, yang membedakannya adalah program ini gratis.
netscape	(pada X terminal) menjalankan netscape, versi pada waktu tulisan ini dibuat telah mencapai versi 4.7
netscape -display	(pada X terminal) menjalankan netscape pada mesin yang aktif dan menampilkan outputnya pada mesin yang bernama "host" display 0 screen 0. Anda harus memberikan akses untuk mesin aktif untuk menampilkannya pada mesin "host" dengan perintah "xhost"
shutdown -h now	(sebagai root) Shut down sistem. Umumnya digunakan untuk remote shutdown. Gunakan <Ctrl><Alt> untuk shutdown pada konsol (dapat dijalankan oleh user).
halt	reboot (sebagai root) Halt atau reboot mesin. Lebih simple dari perintah di atas.
man topic	Menampilkan daftar dari sistem manual pages (help) sesuai dengan topic. Coba "man man". Lalutekan "q" untuk keluar dari viewer. Perintah "info topic" Manual pages dapat dibaca dilihat dengan cara "any_command --help".
apropos topic	Menampilkan bantuan manual berdasarkan topik..
pwd	Melihat direktori kerja saat ini
hostname	Menampilkan nama local host (mesin dimana anda sedang bekerja). Gunakan perintah "netconf" (sebagai root) untuk merubah nama host dari mesin tersebut, atau edit file /etc/hosts
whoami	Mencetak login name anda
id username	Mencetak user id (uid) atau group id (gid)

date	Mencetak atau merubah tanggal dan waktu pada komputer, contoh merubah tanggal dan waktu ke 2000-12-31 23:57 dengan perintah; date 123123572000
time	Melihat jumlah waktu yg ditangani untuk penyelesaian suatu proses + info lainnya. Jangan dibingungkan dengan perintah "date"
who	Melihat user yang login pada komputer kita.
rwho -a	Melihat semua user yg login pada network anda. Layanan perintah rwho ini harus diaktifkan, jalankan setup sebagai root utk mengaktifkannya.
finger username	Melihat informasi user, coba jalankan; finger root
last	Melihat user sebelumnya yang telah login di komputer.
uptime	Melihat jumlah waktu pemakaian komputer oleh seseorang, terhitung proses reboot terakhir.
ps (=print status)	Melihat prosesproses yang dijalankan oleh user
ps axu	Melihat seluruh proses yang dijalankan, walaupun tanpa terminal control, juga ditampilkan nama dari user untuk setiap proses.
top	Melihat proses yang berjalan, dengan urutan penggunaan cpu.
uname -a	Informasi system kernel anda
free	Informasi memory (dalam kilobytes).
df -h	(=disk free) Melihat informasi pemakaian disk pada seluruh system (in human-readable form)
du / -bh	(=disk usage) Melihat secara detil pemakaian disk untuk setiap direktori, dimulai dari root (in human legible form).
cat /proc/cpuinfo	Cpu info. Melihat file pada /proc direktori yang bukan merupakan file nyata (not real files).
cat /proc/interrupts	Melihat alamat interrupt yang dipakai.
cat /proc/version	Versi dari Linux dan informasi lainnya.
cat /proc/filesystems	Melihat filesystem yang digunakan.
cat /etc/printcap	Melihat printer yang telah disetup
lsmod	(as root) Melihat module-module kernel yang telah di load.
set	Melihat environment dari user yang aktif
echo \$PATH	Melihat isi dari variabel "PATH". Perintah ini dapat digunakan untuk menampilkan variable environmen lain dengan baik. Gunakan "set" untuk melihat environmen secara penuh.
dmesg	Mencetak pesanpesan pada waktu proses boot. (menampilkan file: /var/log/dmesg).
clear	Membersihkan layar.
adduser	Menambah pengguna.

Keamanan merupakan hal utama dalam jaringan. Apalagi jika seluruh host tersambung ke Internet. Melindungi jaringan, berarti melindungi setiap host yang ada dalam jaringan, baik *workstation* maupun server. Pada artikel ini akan dijelaskan beberapa komponen terkait keamanan pada sistem dalam Linux. Keamanan di Linux mempunyai enam komponen antara lain:

- Account Pemakai (*User Account*)
- Kontrol Akses secara Diskresi (*Discretionary Access Control*)
- Kontrol Akses Jaringan (*Network Access Control*)
- Enkripsi (*Encryption*)
- *Logging*
- Deteksi penyusupan (*Intrusion Detection*)

- **Account Pemakai (user account)**

Keuntungan :

- Kekuasaan dalam satu account yaitu root, sehingga mudah dalam administrasi system.
- Kecerobohan salah satu user tidak berpengaruh kepada system secara keseluruhan.
- Masing-masing user memiliki privacy yang ketat.
- Macam User :
  1. Root : kontrol system file, user, sumber daya (devices) dan akses jaringan.
  2. User : account dengan kekuasaan yang diatur oleh root dalam melakukan aktifitas dalam system.
  3. Group : kumpulan user yang memiliki hak sharing yang sejenis terhadap suatu devices tertentu.

- **Hak Akses dalam Linux**

Linux mengenal hak akses yang mengatur setiap user sehingga tiap user hanya dapat mengakses file atau direktori tertentu saja. Hal ini digunakan untuk kepentingan keamanan system. Setiap file dan direktori yang ada dalam sistem linux memiliki tiga buah hak akses, satu untuk user itu sendiri, kemudian untuk user dalam grup yang sama dengan pemilik file dan yang terakhir untuk user lainnya.

Anda dapat melihat hak akses sebuah file dengan menggunakan perintah `ls -l <direktori>`, perhatikan contoh dibawah ini :

```
drwxr-xr-x 2 kvrp user 4096 2013-11-14 07:20 Destop
drwxr-xr-x 2 kvrp user 4096 2013-11-14 07:19 Dokumen
-rw-r--r-- 1 kvrp user 179 2013-11-14 03:29 examples.desktop
drwxr-xr-x 2 kvrp user 4096 2013-11-14 07:27 Gambar
drwxr-xr-x 2 kvrp user 4096 2013-11-14 03:38 Musik
```

**Karakter pertama** menunjukkan jenisnya, jika berisi karakter d, berarti itu adalah direktori sedangkan jika kosong berarti file. **Sembilan karakter berikutnya** menunjukkan hak aksesnya, dengan tiga karakter pertama menunjukkan hak akses untuk user tersebut, tiga karakter berikutnya menunjukkan hak akses untuk grup nya dan tiga karakter terakhir menunjukkan hak akses untuk user lain. Masing-masing arti karakter tersebut dapat dilihat pada tabel berikut:

Karakter	Nilai	Arti
r (read)	4	Hak akses untuk membaca
w (write)	2	Hak akses untuk menulis
x (execute)	1	Hak akses untuk menjalankan

Dengan mengkombinasikan nilai pada masing-masing hak akses tersebut, akan didapatkan hak akses masing-masing. Perhatikan tabel berikut:

Hak Akses	Nilai	Arti
--	0	Tidak memiliki hak akses (tidak dapat di akses)
r--	4	Dapat dibaca dan ditulis (diedit)
rw-	6	Dapat dibaca dan ditulis (diedit)
rwX	7	Dapat dibaca, ditulis dan dieksekusi (dijalankan)
r-x	5	Dapat dibaca dan dijalankan, tetapi tidak dapat diedit
-x	1	Hanya dapat dijalankan

Berikut contoh penerapannya:

```
drwxr-xr-x 2 kvrp user 4096 2013-11-14 07:27 Gambar
755 ----- user kvrp memiliki hak akses baca, tulis dan menjalankan
folder Gambar, sedangkan group dan user lain hanya bisa membaca dan
menjalankannya.
```

- **Mengubah Hak Akses File/ Folder**

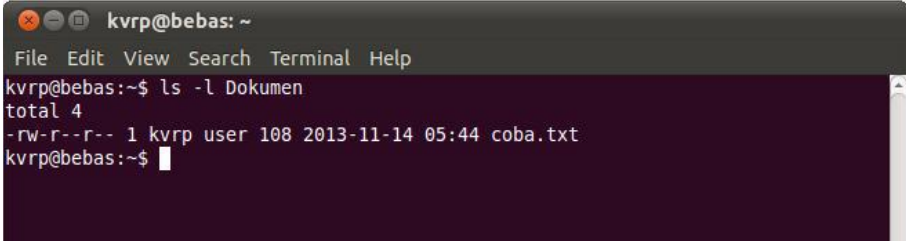
Sebuah file/folder dapat diubah hak aksesnya dengan cara membuka terminal lalu mengetikkan :

**chmod <hak\_akses><nama\_file\_atau\_direktori>**

Contoh :

Di folder Dokumen terdapat file bernama coba.txt dengan nilai hak akses 644, yang berarti user kvrp dapat membaca dan mengedit file tersebut, sedangkan group dan user lain hanya dapat membacanya saja.

Perhatikan gambar di bawah ini!

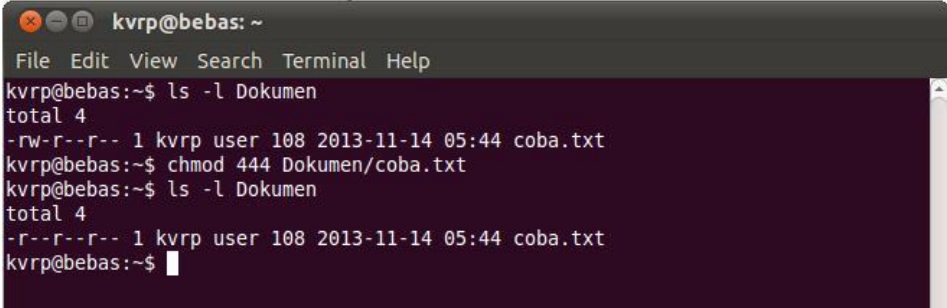


```
kvrp@bebas: ~
File Edit View Search Terminal Help
kvrp@bebas:~$ ls -l Dokumen
total 4
-rw-r--r-- 1 kvrp user 108 2013-11-14 05:44 coba.txt
kvrp@bebas:~$
```

Kita ubah hak aksesnya agar tidak ada user lain yang bisa mengakses file tersebut selain pemiliknya (user kvrp).

Ketikkan `chmod 600 Dokumen/coba.txt` di terminal.

Cek perubahan hak akses dengan mengetik `ls -l Dokumen`



```
kvrp@bebas: ~
File Edit View Search Terminal Help
kvrp@bebas:~$ ls -l Dokumen
total 4
-rw-r--r-- 1 kvrp user 108 2013-11-14 05:44 coba.txt
kvrp@bebas:~$ chmod 444 Dokumen/coba.txt
kvrp@bebas:~$ ls -l Dokumen
total 4
-r--r--r-- 1 kvrp user 108 2013-11-14 05:44 coba.txt
kvrp@bebas:~$
```

Untuk mengganti hak akses sebuah direktori beserta dengan isinya, gunakan parameter R, dengan parameter tersebut, chmod akan dijalankan secara rekursif, misalnya seluruh file yang ada pada direktori /home/user/public\_html akan dirubah hak aksesnya menjadi 755, maka perintahnya adalah sebagai berikut:

```
# chmod R 755 /home/user/public_html
```

- **Mengubah Kepemilikan File/Folder**

Untuk mengganti pemilik sebuah file, kita menggunakan perintah chown. Perintah ini hanya dapat digunakan oleh user root. Perintahnya adalah sebagai berikut:

```
chown <nama_user.nama_grup> <nama_file>
```

```
misalnya: chown user.user coba.txt
```

perintah chown juga dapat digunakan dengan menggunakan parameter R, contohnya adalah sebagai berikut:

```
chown R apache.apache /var/www/html
```

- **Perintah chattr**

Seringkali secara tidak sengaja kita menghapus atau mengedit sebuah file penting. Di dalam Linux tidak ada fasilitas undelete, jadi file yang telah terhapus tidak dapat dikembalikan lagi. Perintah (program) chattr digunakan untuk melindungi sebuah file sehingga tidak akan dapat dihapus ataupun dirubah dengan perintah apapun. Perintah chattr memberikan atribut i pada file yang dilindungi, perintahnya:

```
chattr +i <nama_file>
```

```
misal: chattr +i penting.txt
```

Setelah perintah tersebut dijalankan, gunakan perintah ls -l untuk melihat hasilnya. Jika suatu ketika file ini akan diedit atau dihapus, terlebih dahulu atribut diatas harus dilepas, untuk melepasnya gunakan perintah berikut:

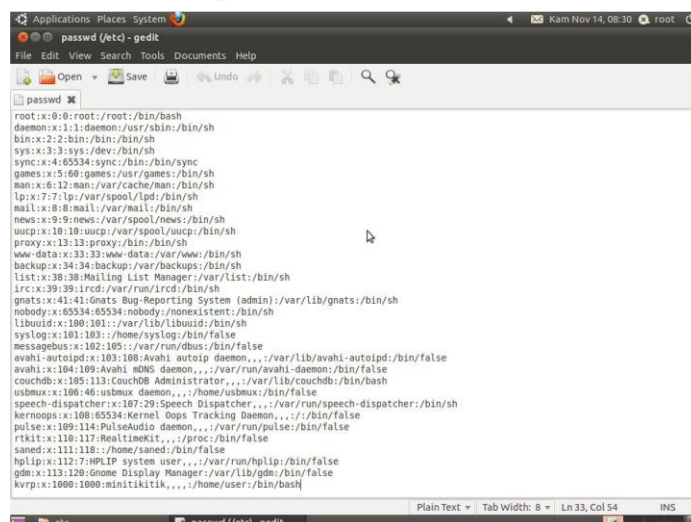
```
chattr i <nama_file>
```

- **Keamanan Host (*host security*) pada linux**

### Password

Untuk dapat mengakses sistem operasi Linux digunakan mekanisme password. Pada distribusi-distribusi Linux yang lama, password tersebut disimpan dalam suatu file teks yang terletak di /etc/passwd. File ini harus dapat dibaca oleh setiap orang (*world readable*) agar dapat digunakan oleh program-program lain yang menggunakan mekanisme password tersebut.

Berikut ini adalah contoh isi file /etc/passwd :



```
passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:MailList Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuid:x:100:101:/var/lib/libuid:/bin/sh
syslog:x:101:103:/home/syslog:/bin/false
messagebus:x:102:105:/var/run/dbus:/bin/false
avahi-autoipd:x:103:108:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:104:109:Avahi mDNS daemon,,:/var/run/avahi-daemon:/bin/false
couchdb:x:105:113:CouchDB Administrator,,:/var/lib/couchdb:/bin/bash
usbmux:x:106:46:usbmux daemon,,:/home/usbmux:/bin/false
speech-dispatcher:x:107:29:Speech Dispatcher,,:/var/run/speech-dispatcher:/bin/sh
kernoops:x:108:65534:Kernel Oops Tracking Daemon,,:/bin/false
pulse:x:109:114:PulseAudio daemon,,:/var/run/pulse:/bin/false
rtkit:x:110:117:RealtimeKit,,:/proc:/bin/false
saned:x:111:118:/home/saned:/bin/false
hplip:x:112:7:HPLIP system user,,:/var/run/hplip:/bin/false
gdm:x:113:120:Gnome Display Manager:/var/lib/gdm:/bin/false
kvrp:x:1000:1000:inittik,,:/home/user:/bin/bash
```

Satu baris mencerminkan satu user, lengkap dengan atributnya. Berikut keterangan tiap field:

Username:password:UserID:GroupID:keterangan:home:direktori:shell

Contoh:

```
kvrp:x:1000:1000:minitikitik,,,:/home/user:/bin/bash
```

Keterangan dari line tersebut :

- Username : kvrp
- Password : x
- User ID (UID) : 1000
- Group ID (GID) : 1000
- Keterangan tambahan : minitikitik,,,
- Home Direktori : /home/user
- Shell Default : /bin/bash

Password login yang terdapat pada file /etc/passwd dienkripsi dengan menggunakan algoritma DES yang telah dimodifikasi. Meskipun demikian hal tersebut tidak mengurangi kemungkinan password tersebut dibongkar (*crack*). Karena penyerang (*attacker*) dapat melakukan *dictionary-based attack* dengan cara :

- menyalin file /etc/passwd tersebut
- menjalankan program-program yang berguna untuk membongkar password, contohnya adalah Crack dan John the Ripper.

Untuk mengatasi permasalahan ini pada distribusi-distribusi Linux yang baru digunakan program utility **shadow password** yang menjadikan file /etc/passwd tidak lagi berisikan informasi password yang telah dienkripsi, informasi tersebut kini disimpan pada file /etc/shadow yang hanya dapat dibaca oleh root.

Berikut ini adalah contoh file /etc/passwd yang telah **dishadow** :

```
root:$6$WmG5wxDsZ8TmnuEk3Fj75XV7uc59r14nIw7C2oLx1A0p5x95Y1nKbKBVgy6biy39dst3Z1N9Yg05BYJ25rhAxcZC4R0p.:16022:0:99999:7:::
daemon:*:14889:0:99999:7:::
bin:*:14889:0:99999:7:::
sys:*:14889:0:99999:7:::
sync:*:14889:0:99999:7:::
games:*:14889:0:99999:7:::
man:*:14889:0:99999:7:::
lp:*:14889:0:99999:7:::
mail:*:14889:0:99999:7:::
news:*:14889:0:99999:7:::
uucp:*:14889:0:99999:7:::
proxy:*:14889:0:99999:7:::
www-data:*:14889:0:99999:7:::
backup:*:14889:0:99999:7:::
list:*:14889:0:99999:7:::
irc:*:14889:0:99999:7:::
gnats:*:14889:0:99999:7:::
nobody:*:14889:0:99999:7:::
libuuid:l:14889:0:99999:7:::
syslog:*:14889:0:99999:7:::
messagebus:*:14889:0:99999:7:::
avahi-autoipd:*:14889:0:99999:7:::
avahi:*:14889:0:99999:7:::
couchdb:*:14889:0:99999:7:::
usbmux:*:14889:0:99999:7:::
speech-dispatcher:l:14889:0:99999:7:::
kernoops:*:14889:0:99999:7:::
pulse:*:14889:0:99999:7:::
rtkit:*:14889:0:99999:7:::
saned:*:14889:0:99999:7:::
hplip:*:14889:0:99999:7:::
gdm:*:14889:0:99999:7:::
kvrp:$1$bxDYCSMBg.P72byndnEWTshybr1:16022:0:99999:7:::
```

Dengan demikian, penggunaan *shadow password* akan mempersulit *attacker* untuk melakukan *dictionary-based attack* terhadap file password. Sebagai salinan dari /etc/passwd, shadow menyimpan satu record dalam satu baris yang dibagi menjadi 9 bagian (dipisahkan oleh titik dua):

- Nama user

- Password user
- Jumlah hari digantinya password sejak 1 Januari 1970
- Jumlah hari sebelum user diijinkan untuk mengubah password
- Jumlah hari sebelum user dipaksa untuk mengganti password
- Jumlah hari sebelum user diingatkan untuk mengganti password
- Jumlah hari yang tersisa di mana user harus mengganti password sebelum user tersebut dihapus.
- Jumlah hari dihapusnya password sejak 1 Januari 1970
- Field terakhir dipakai sendiri oleh shadow

Di /etc/passwd

```
kvrp:x:1000:1000:minitik,,,:/home/user:/bin/bash
```

Di /etc/shadow

```
kvrp:$1$bxDYC$MBg.P72byndnEWTShybRr1:16022:0:99999:7:::
```

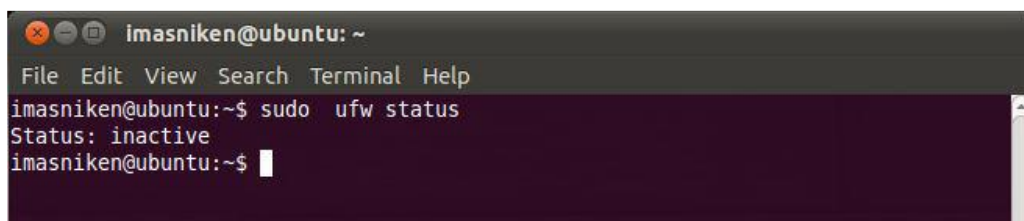
- **Kontrol akses jaringan (Network Access Control)**

Firewall merupakan sebuah alat yang membatasi suatu sistem jaringan yang ada di baliknya dari berbagai macam ancaman dan gangguan yang biasa muncul melalui jaringan internet yang rentan terhadap berbagai macam serangan. Fungsinya untuk membatasi hak akses dan mengatur aturan antara jaringan internal terhadap eksternalnya dan sebaliknya. Hal ini sangat penting mengingat tidak semua orang diperbolehkan untuk bisa mengakses ke dalam jaringan yang kita miliki. Konfigurasi dari firewall bergantung kepada kebijaksanaan dari user yang bersangkutan, secara umum terbagi menjadi dua jenis : *Deny* (menolak) dan *Allow* (memperbolehkan). Cara kerja firewall sebenarnya hanyalah dengan mengamati paket data yang dilewatkan dan kemudian berdasarkan konfigurasi dari firewall maka akses dapat diatur berdasarkan alamat IP, Port, dan arah atau tujuan informasi.

Iptables adalah suatu tools dalam sistem operasi linux yang berfungsi sebagai alat untuk melakukan filter (penyaringan) terhadap (traffic) lalu lintas data. Secara sederhana digambarkan sebagai pengatur lalu lintas data. Dengan iptables inilah kita akan mengatur semua lalu lintas dalam komputer kita, baik yang masuk ke komputer, keluar dari komputer, ataupun traffic yang sekedar melewati komputer kita.

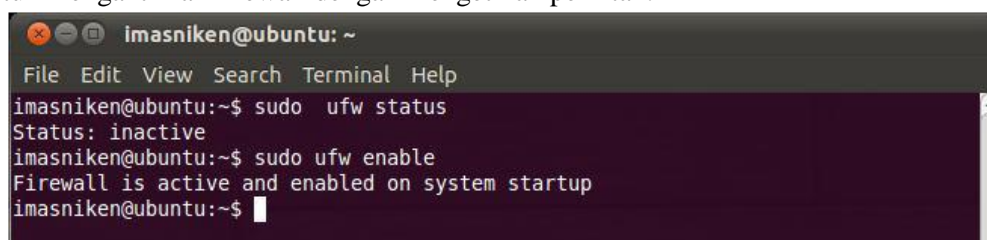
Cara mengaktifkan firewall pada Linux:

- Untuk mengecek aktif atau tidaknya firewall



```
imasniken@ubuntu: ~
File Edit View Search Terminal Help
imasniken@ubuntu:~$ sudo ufw status
Status: inactive
imasniken@ubuntu:~$
```

- Untuk mengaktifkan firewall dengan mengetikkan perintah.



```
imasniken@ubuntu: ~
File Edit View Search Terminal Help
imasniken@ubuntu:~$ sudo ufw status
Status: inactive
imasniken@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
imasniken@ubuntu:~$
```



- Untuk memblokir akses.

```

imasniken@ubuntu: ~
File Edit View Search Terminal Help
imasniken@ubuntu:~$ sudo ufw default DENY
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
imasniken@ubuntu:~$ █

```

- Untuk memberikan akses.

```

imasniken@ubuntu: ~
File Edit View Search Terminal Help
imasniken@ubuntu:~$ sudo ufw default ALLOW
Default incoming policy changed to 'allow'
(be sure to update your rules accordingly)
imasniken@ubuntu:~$ █

```

- **Enkripsi (encryption)**

Penerapan Enkripsi di linux:

- Enkripsi password menggunakan DES ( Data Encryption Standard )
- Enkripsi komunikasi data :

1. **Secure Shell (SSH)** Program yang melakukan logging terhadap komputer lain dalam jaringan, mengeksekusi perintah lewat mesin secara remote dan memindahkan file dari satu mesin ke mesin lainnya. Enkripsi dalam bentuk Blowfish, IDEA, RSA, Triple DES. Isi SSH Suite :

- scp (secure shell copy) -> mengamankan penggandaan data
- ssh (secure shell client) -> model client ssh seperti telnet terenkripsi.
- ssh-agent -> otentikasi lewat jaringan dengan model RSA.
- sshd (secure shell server) -> di port 22
- ssh-keygen -> pembuat kunci (key generator) untuk ssh

Konfigurasi dilakukan di :

- /etc/sshd\_config (file konfigurasi server)
- /etc/ssh\_config (file konfigurasi client)

2. **Secure socket Layer (SSL)** mengenkripsi data yang dikirimkan lewat port http. Konfigurasi dilakukan di : web server APACHE dengan ditambah PATCH SSL.

### Tugas

1. Buatlah tutorial untuk membuat dua directory folder pada linux, dengan permission untuk satu user, dan untuk semua user



## PERTEMUAN 9

### MANAJEMEN MEMORY

#### TUJUAN PRAKTIKUM

- Praktikan dapat memahami proses manajemen memory.
- Praktikan dapat mensetting parameter dalam manajemen memory.

#### SIMULASI

Simulasi ini memperlihatkan permuatan proses pada memori utama sesuai dengan sistem manajemen memori yang beragam. Simulasi ini juga memperlihatkan versi sederhana dari proses **swap** dan perilaku **virtual memory**.

Algoritma manajemen memori dibagi menjadi dua grup, yaitu:

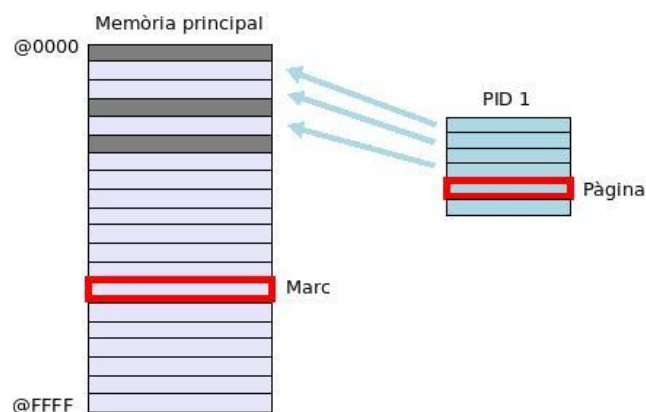
- *Contiguous memory management*, yaitu seluruh proses dialokasikan ke memori. Untuk setiap proses, partisi yang sedang tidak dipakai dan cukup besar untuk mengakomodasi proses yang membutuhkan akan dipilih.

Terdapat dua algoritma manajemen memori:

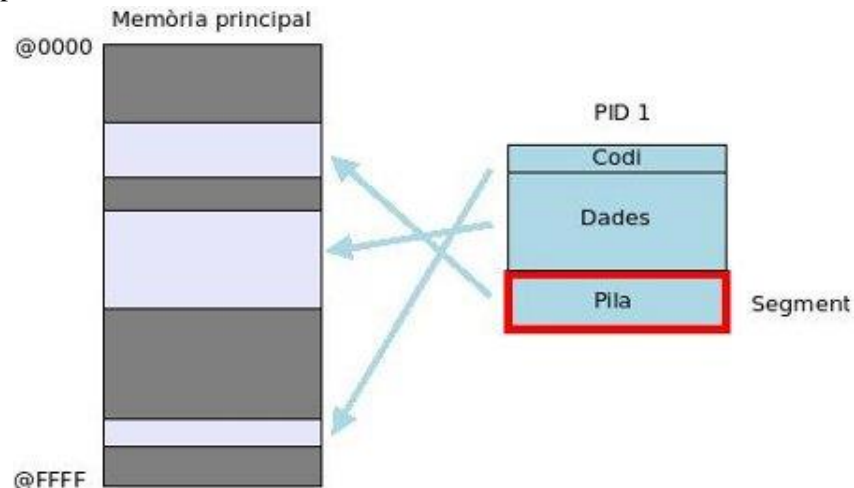
- Partisi tetap (*fixed-sized partitions*), memori awalnya dibagi menjadi beberapa partisi dengan ukuran tetap (pengguna yang membuat). Pada umumnya proses yang dialokasikan kedalam sebuah partisi memiliki ukuran yang lebih kecil daripada ukuran partisinya, dan oleh karena itu ada sebagian ukuran partisi yang tidak terpakai, yang disebut sebagai *internal fragmentation*.
- Partisi tidak tetap (*variable-sized partitions*), memori awalnya merupakan sebuah partisi besar. Lalu untuk setiap proses akan dialokasikan ukuran partisi yang sesuai dengan ukuran proses, dan setelah selesai maka partisi tersebut dapat digunakan ulang. Partisi yang telah selesai digunakan tersebut dapat disebut sebagai *external fragmentation*.

Konsep lain yang menjadi perhatian adalah aturan alokasi memori, yaitu bagaimana sistem operasi memilih satu dari beberapa partisi yang tersedia. Beberapa diantaranya adalah:

- *First fit*, memilih partisi pertama kali yang dapat memuat proses.
- *Best fit*, memilih partisi yang dapat memuat proses secara optimal.
- *Worst fit*, memilih partisi yang dapat memuat proses dengan pemilihan paling buruk.
- *Non-contiguous memory management*, yaitu seluruh proses dibagi menjadi bagian-bagian yang dapat dialokasikan secara terpisah di memori, antara lain:
  - *Pagination*, memori dan proses dibagi menjadi bagian-bagian yang berukuran sama (frame dan page), dimana page dari proses akan dialokasikan pada frame dari memori.



- *Segmentation*, proses dibagi menjadi bagian-bagian secara logical (misalnya code, data, atau stack) dan masing-masing akan dialokasikan ke partisi memori secara independen.



## Tampilan awal

### Main memory

Proses-proses nantinya akan dialokasikan disini, baik itu secara keseluruhan maupun per bagian page atau segmen.

Secara definisi, pada setiap sistem manajemen memori, akan ada selalu terdapat satu proses pertama yang akan dialokasikan ke memori yang mendapatkan alamat dasar (base addresses), yaitu Operating System, yang ukurannya bervariasi dapat diantara 1, 2, atau 4 unit.

Alamat, proses, partisi yang tersedia maupun partisi yang sedang digunakan akan selalu terlihat, dan juga fragmentasi mengikuti kode warna yang memungkinkan pemahaman mahasiswa lebih cepat.

Warna	Keterangan
Abu-abu	Operating System
Putih	Memori belum digunakan
Titik biru	Fragmentasi internal
Titik merah muda	Fragmentasi eksternal
Warna lain...	Proses (masing-masing)

Ukuran memori dari simulasi ini bervariasi antara 64 sampai dengan 256 unit.

Klik kanan pada memori untuk menampilkan **pop up menu** yang memungkinkan untuk melakukan beberapa pilihan:

- Saat simulasi berhenti (hanya untuk partisi tetap) >> untuk update dan delete partisi
- Saat waktu simulasi untuk setiap proses yang dialokasikan ke memori >> delete proses, swap proses, melihat address translation, melihat informasi detail (hanya untuk pagination dan segmentation) dan untuk melakukan defragment memori (hanya untuk partisi tidak tetap dan segmentation).

### Process queue

Memuat proses yang tersedia untuk dialokasikan ke memori diurutkan berdasarkan waktu kedatangan proses.

Untuk setiap proses menunjukkan informasi yang relevan (PID, nama, dan durasi), disamping dari ukuran dan distribusinya, dimana setiap kotak bersesuaian ke satu unit space, yang dapat digunakan oleh satu alamat memori.

Pada pagination dan segmentation, proses-proses dibedakan menjadi komponen-komponennya, dan di-highlight oleh warna abu-abu untuk komponen yang belum dialokasikan ke memori tetapi dialokasikan ke swap area.

Saat simulasi berhenti, klik kanan pada sebarang proses untuk melihat pop up menu yang memungkinkan untuk meng-update dan delete.

### Settings

- *Memory Size* (Ukuran memori, bervariasi antara 64 s/d 256 unit).
- *Operating System Size* (Ukuran memori untuk OS, bervariasi dari 1, 2 atau 4 unit).
- Algorithm:
  - *Fixed-sized partitions* (contiguous memory management) >> required untuk mempartisi seluruh memori.
  - *Variable-sized partitions* (contiguous memory management).
  - *Pagination* (non-contiguous memory management) >> harus menentukan ukuran page sistem (antara 1, 2 atau 4).
  - *Segmentation* (non-contiguous memory management).
- Allocation policy, memilih algoritma pemilihan partisi, yaitu:
  - *First fit*
  - *Best fit*
  - *Worst fit*

Perubahan pada algoritma, ukuran memori, atau ukuran OS membutuhkan untuk me-restart simulasi dan menghapus seluruh proses yang ada.

### Menambah proses

Proses hanya dapat ditambahkan selagi simulasi berhenti, untuk setiap proses baru dapat ditambahkan dengan cara berikut:

- *PID* >> proses identifier, bersifat *automatically calculated* dan *incremental*.
- *Name (required)* >> nama proses.
- *Size* >> ukuran proses (nilai: 1 s/d 64 unit).
- *Duration* >> proses dapat mempunyai durasi tak terhingga (diisi -1) atau terhingga (1-100), tapi tidak boleh 0.
- *Color* >> warna yang akan digunakan untuk menandai proses yang dimaksud.

Tambahan untuk pagination:

- *Page table* >> jumlah page yang tergantung pada ukuran proses, untuk setiap page mahasiswa harus menentukan apakah sudah dialokasikan ke memori secara langsung (initially allocated) atau tidak (swapped).

Tambahan untuk segmentation:

- *Segment table* >> setiap proses mempunyai tiga segmen: code, data, dan stack. Untuk setiap segmen, mahasiswa harus menentukan ukuran dan apakah sudah dialokasikan ke memori (initially allocated) atau tidak (swapped). Ukuran dari jumlahan segmen harus sama dengan ukuran dari proses.

### **Data dan statistik**

Menunjukkan informasi alokasi memori pada setiap waktu simulasi. Informasi ini beragam antara satu algoritma dengan algoritma yang lain.

Contiguous memory management, tabel dengan informasi berikut:

- *Direction*, alamat partisi mula-mula.
- *Size*, ukuran dari partisi.

(jika pada partisi terdapat sebuah proses)

- *PID*, proses ID (warna background menunjukkan warna proses).
- *Name*, nama proses.
- *Size*, ukuran proses.
- *Duration*, durasi proses.

Pada pagination, tabel dengan informasi berikut:

- *Address*, alamat frame mula-mula.
- *Frame*, nomor frame.

(jika partisi sedang digunakan)

- *PID*, proses ID (warna background menunjukkan warna proses).
- *Page*, proses yang menggunakan frame yang bersesuaian.
- *Name*, nama proses.
- *Dimensions*, total proses.
- *Duration*, durasi proses.

Pada segmentation, tabel dengan informasi berikut:

- *Address*, alamat frame mula-mula.
- *Size*, ukuran partisi.

(jika pada partisi terdapat sebuah proses)

- *PID*, proses ID (warna background menunjukkan warna proses).
- *Segment*, segmen proses, yaitu: code, data atau stack.
- *Name*, nama proses.
- *Size*, ukuran proses.
- *Duration*, durasi proses.

### **Tabel page dan tabel segment**

Menunjukkan informasi alokasi memori dari sebuah proses pada non-contiguous memory management.

Pada pagination, konten tabel adalah sebagai berikut.

- *Page*, proses yang menggunakan frame yang bersesuaian.

- *Frame*, nomor frame.
- *Valid*, bit valid dari page (v valid, i invalid). Sebuah page dikatakan tidak valid ketika page tersebut tidak dialokasikan ke memori.

Pada segmentation, konten tabel adalah sebagai berikut.

- *Segment*, segmen dari proses: code, data, atau stack.
- *Size*, ukuran dari segmen.
- *Address*, alamat frame mula-mula.
- *Valid*, bit valid dari segmen (v valid, i invalid). Sebuah segmen dikatakan tidak valid ketika segmen tersebut tidak dialokasikan ke memori.

### **Membuat partisi (untuk partisi berukuran tetap)**

Menentukan alamat mula-mula dan ukuran partisi.

Partisi tidak bisa saling overlap, dan tidak bisa melebihi batas akhir memori.

### **Defragment memori (untuk partisi berukuran tidak tetap dan segmentation)**

Partisi berukuran tidak tetap dan segmentation mempunyai kesamaan pada partisi yang dibuat secara dinamis dan diukur secara otomatis untuk mengakomodasi proses atau segmen secara tepat. Operating System memilih sebuah partisi yang sama atau sedikit lebih besar dan dibagi menjadi dua bagian, satu untuk mengalokasikan proses atau segmen, satu untuk yang masih tersedia.

Setelah selesai digunakan (proses sudah selesai atau swapped out), partisi tersebut menjadi kosong dan berubah menjadi lebih kecil (memorinya ter-degradasi).

Proses defragmentasi dapat menanggulangi untuk me-reorganize memori, mengatur semua proses menjadi menggunakan memori dengan alamat rendah (lower memory addresses) dan menggabungkan seluruh partisi kosong menjadi satu partisi kosong yang besar.

Ini adalah tanggung jawab dari OS, dan ini diserahkan kepada mahasiswa untuk simulasi (pop up menu dari memori).

### **Swap**

Melakukan swap dapat memungkinkan kita untuk menjalankan proses yang lebih banyak dari kapasitas memori utama, dan biasanya diimplementasikan pada memori sekunder.

Hal ini muncul atas dasar untuk mempunyai space ekstra untuk memindah proses atau bagian darinya (page atau segmen) yang sedang kurang aktif dan mengalokasikan ulang ke memori saat sedang dibutuhkan.

Simulasi digunakan untuk menyederhanakan proses ini, mensimulasikan swap in dan swap out antar memori dan swap dilakukan oleh mahasiswa secara manual.

*Swap out*, dari pop up menu memori (klik kanan pada memori). Melakukan swap out sebuah proses atau bagian darinya.

*Swap in*, dari pop up menu memori (klik kanan pada memori). Mencoba untuk melakukan swap in pada sebuah proses atau bagian darinya kepada memori, jika memori penuh akan menghasilkan sebuah error dan tidak akan bisa swap in.

Algoritma pagination dan segmentation memungkinkan untuk melakukan swap out page atau segmen pada awal (initial). Area ini tidak dibatasi pada jumlah atau ukuran dari proses yang dimuat.

### **Error pada simulasi**

1. **Memory full**, karena sedang memproses antrian proses pada simulasi, memori mengalokasikan untuk proses dan selesai mengalokasikannya. Saat proses tidak bisa dialokasikan karena kurang space-nya, maka menyebabkan tidak dapat dilanjutkan ke proses

selanjutnya. Memori dapat dilepaskan (released) secara otomatis (karena proses sudah selesai menggunakannya) atau oleh campur tangan user (delete atau swap out proses).

2. **Memory not fully partitioned**, algoritma partisi berukuran tetap (fixed-size partition) membutuhkan semua memori untuk dipartisi (alamat akhir satu partisi adalah alamat awal partisi yang lain).

### Contoh

#### *Contiguous memory management dengan menggunakan partisi berukuran tetap (fixed-size partition) dan aturan first fit*

Memori dibagi menjadi partisi-partisi berukuran tetap, lalu seluruh proses akan dialokasikan pada satu dari partisi yang tersedia. Pada aturan first fit, partisi pertama yang dapat memuat proses akan dipilih.

Partisi		Proses	
Alamat awal	Unit ukuran	Proses	Unit ukuran
@4	10	P1	5
@14	5	P2	10
@19	4	P3	3
@23	15		
@38	10		
@48	14		

Proses akan dialokasikan sesuai dengan urutan pada tabel

Keterangan:

- Aturan first fit dapat menyelesaikan masalah secara cepat, tapi tidak efisien, partisi dapat menjadi lebih besar dari proses dan akan menimbulkan fragmentasi internal.

#### **Contiguous memory management dengan menggunakan partisi berukuran tetap (fixed-size partition) dan aturan best fit**

Memory is divided into fixed-sized partitions, entire processes are allocated in one of these available partitions, in "best fit" policy, a partition which is large enough and fits best is selected.

Memori dibagi menjadi partisi-partisi berukuran tetap, lalu seluruh proses akan dialokasikan pada satu dari partisi yang tersedia. Pada aturan best fit, partisi terbaik (yang ukurannya sama atau hampir sama) yang dapat memuat proses akan dipilih.

Partisi		Proses	
Alamat awal	Unit ukuran	Proses	Unit ukuran
@4	10	P1	5
@14	5	P2	10
@19	4	P3	3
@23	15		
@38	10		
@48	14		

Proses akan dialokasikan sesuai dengan urutan pada tabel

Keterangan:

- Aturan best fit meminimalkan fragmentasi internal.

**Contiguous memory management dengan menggunakan partisi berukuran tidak tetap (variable-size partition) >> defragmentasi**

Memori pertama tidak dipartisi, lalu partisi akan dibuat saat proses dialokasikan dan mempunyai ukuran yang sama dengan ukuran proses. Selanjutnya partisi akan dibebaskan (saat proses sudah selesai menggunakan memori) dan digunakan oleh proses lain. Jika ukuran proses lebih kecil dari ukuran partisi, maka partisi akan dibagi menjadi dua bagian, bagian pertama sama besarnya dengan proses dan bagian kedua adalah sisanya.

Proses	Unit ukuran	Durasi
P1	5	5
P2	8	4
P3	3	3
P4	15	$\infty$
P5	2	1
P6	20	$\infty$
P7	3	$\infty$
P8	10	$\infty$

Proses akan dialokasikan sesuai dengan urutan pada tabel.

Keterangan:

- Proses P7 dialokasikan pada memori setelah partisi dibebaskan oleh sebuah proses yang sudah selesai.
- Proses terakhir tidak dapat dialokasikan ke memori karena tidak ada space yang mencukupi pada memori untuk mengakomodasinya. Padahal pada keseluruhan partisi yang kosong cukup untuk mengakomodasi proses terakhir.
- Jika memori sudah di-defragment (klik kanan pada memori, pop up menu), semua space yang kosong akan digabungkan menjadi satu partisi yang besar dan proses terakhir dapat dimuat ke memori.

**Contiguous memory management dengan menggunakan partisi berukuran tidak tetap (variable-size partition) >> swap**

Memori pertama tidak dipartisi, lalu partisi akan dibuat saat proses dialokasikan dan mempunyai ukuran yang sama dengan ukuran proses. Selanjutnya partisi akan dibebaskan (saat proses sudah selesai menggunakan memori) dan digunakan oleh proses lain. Jika ukuran proses lebih kecil dari ukuran partisi, maka partisi akan dibagi menjadi dua bagian, bagian pertama sama besarnya dengan proses dan bagian kedua adalah sisanya.

Memori mempunyai ukuran yang berhingga, hanya beberapa proses saja yang bisa dimuat di memori pada saat yang sama, tetapi space swap memungkinkan kita untuk menyimpan beberapa proses yang sedang kurang aktif, sehingga space pada memori dapat dialokasikan untuk proses yang lain yang lebih aktif. Proses yang telah di swap out akan dialokasikan kembali ke memori saat dibutuhkan.

Proses	Unit ukuran
P1	5
P2	8
P3	3
P4	15
P5	2

P6	20
P7	3
P8	10

Proses akan dialokasikan sesuai dengan urutan pada tabel.

Keterangan:

- Tidak ada fragmentasi internal, partisinya selalu sesuai dengan ukuran proses.
- Dua proses terakhir (P7 dan P8) tidak dapat dimuat ke memori karena memori penuh.
- Jika beberapa proses di-swap out, akan membebaskan beberapa space pada memori untuk dialokasikan untuk proses yang lain. Contohnya P2 dan P4 (klik kanan pada proses, pop up menu).
- Untuk mengalokasikan kembali proses yang sudah di-swap out, maka pada memori harus ada partisi yang dapat memuat proses tsb, contohnya adalah swap P6. Lalu P2 dan P4 dapat di-swap in (klik kanan pada memori, pop up menu).

### Pagination (ukuran page 2 unit)

Proses akan dibagi menjadi beberapa page dengan ukuran tetap (contohnya adalah 2 unit), memori lalu dibagi menjadi beberapa frame berukuran sama. Page dari proses lalu dialokasikan pada frame memori yang kosong.

Proses	Ukuran	Durasi
P1	5 unit (3 page)	4
P2	4 unit (2 page)	$\infty$
P3	11 unit (6 page)	2
P4	5 unit (3 page)	$\infty$
P5	7 unit (4 page)	$\infty$
P6	3 unit (2 page)	$\infty$

Proses akan dialokasikan sesuai dengan urutan pada tabel.

Keterangan:

- Page terkahir dari sebuah proses tidak selalu harus penuh memenuhi page, ini dapat membuat fragmentasi internal.
- Pada pagination, aturan-aturan pemilihan frame untuk page tidak berlaku, karena page dan frame mempunyai ukuran yang sama.

### Segmentation (alokasi parsial)

Proses dibagi menjadi beberapa segmen, tiap segmen bisa mempunyai ukuran yang berbeda. Segmen ini akan dialokasikan ke memori secara independen, lalu partisi akan dibuat untuk menampung segmen tersebut dengan ukuran yang sama. Pembagian ini adalah secara fungsional dan berdasarkan pada logical structure dari proses tersebut (data, stack, code, dll).

Tidak semua segmen harus dimuat pada memori agar proses dapat berjalan, selama sebuah segmen tidak dipakai maka dapat di-swap out.



Proses	Ukuran total	Segmen	Ukuran (unit)	Dialokasikan diawal
P1	20	Code	2	Yes
		Data	10	Yes
		Stack	8	No
P2	20	Code	5	Yes
		Data	14	No
		Stack	1	Yes
P3	40	Code	10	Yes
		Data	20	Yes
		Stack	10	Yes

Proses akan dialokasikan sesuai dengan urutan pada tabel. Ukuran memori adalah 64 unit, OS mendapatkan 4 unit.

Keterangan:

- Tidak ada fragmentasi internal, partisi selalu mempunyai ukuran yang sama dengan ukuran segmen.
- Ukuran total adalah 80 unit dan total ukuran memori melebihi space pada memori (60 unit), tetapi semua proses dapat dialokasikan baik secara total maupun parsial, oleh karena itu, semua proses dapat bekerja dengan optimal.

### Tugas

1. Jawab latihan OS-SIM fixed sized partition
2. Jawab latihan OS-SIM pagination

## PERTEMUAN 10

### MENGENAL SISTEM OPERASI MACHINTOS DAN ANDROID (OPTIONAL)

#### TUJUAN PRAKTIKUM

- Praktikan dapat mengenal system operasi Machintos dan Android

#### MENGENAL SISTEM OPERASI MACINTOSH (MAC)

Macintosh, atau disingkat Mac, adalah salah satu jenis komputer personal berbasis PowerPC yang diproduksi oleh Apple. Komputer ini dinamakan berdasarkan McIntosh, jenis apel yang disukai Jef Raskin. Macintosh diperkenalkan pertama kali pada bulan Januari 1984 lewat iklan Super Bowl yang fenomenal. Macintosh adalah komputer pertama yang memperkenalkan sistem antarmuka grafis (GUI). Pada waktu itu, langkah yang dilakukan Apple adalah sebuah perkembangan revolusioner dalam dunia komputer personal.

Pembuatan Mac merupakan suatu wujud model integrasi vertikal yang mana Apple memfasilitasi seluruh aspek perangkat keras dan juga sistem operasinya yang terinstal di dalam seluruh komputer Mac. Hal ini berbeda dengan komputer tipe PC pada umumnya, di mana banyak produsen membuat dan mengintegrasikan perangkat keras dengan sistem operasi yang dibuat oleh produsen lain. Sementara itu, Apple secara eksklusif membuat perangkat keras Mac dan mengatur bagaimana sistem internalnya, desain, dan juga harganya. Tidak hanya itu, Apple juga tidak melisensikan Mac OS X untuk komputer non-Apple



Tampilan di atas adalah tampilan Mac pada umumnya. Coba kita telusuri satu per satu elemen tersebut. Kenali Desktop secara mendalam dengan membaca artikel mengenai Desktop.

#### MENUBAR



Menu Bar adalah sebuah indikator yang menunjukkan aplikasi apa yang sedang aktif. Gambar di atas menunjukkan bahwa aplikasi Finder—sejenis Windows Explorer di Windows—sedang

aktif. Di sebelahnya juga menampilkan pilihan menu yang umumnya terdapat pada Windows. Hanya saja, Mac, semuanya selalu berada di Menu Bar. Dan ini yang menjadi kelebihan Mac, yaitu konsistensi.

Menu Bar Icons itu dikategorikan sekelas dengan tray yang terdapat Windows. Banyak aplikasi yang menawarkan pilihan application control melalui menu bar. Screenshot di atas menunjukkan bahwa Battery, F.lux, dan FuzzyClock sedang aktif.



## DOCK

Dock adalah wilayah di mana aplikasi yang aktif bertempat. Sebuah aplikasi yang sedang dijalankan selalu dimasukkan ke dalam Dock. Sebuah indikator bercahaya di bawah icon aplikasi menunjukkan bahwa aplikasi tersebut sedang aktif. Dock dapat dianggap sebagai Taskbar pada Windows. Anda juga dapat memasukkan aplikasi yang sering digunakan pada Dock tanpa harus menjalankannya—seperti Pin to Taskbar. Baca ”Segala Sesuatu Mengenai Dock” untuk mengetahui lebih lanjut.

Spotlight dapat membantu Anda menjalankan aplikasi, menemukan files, mengakses konfigurasi Mac, serta melakukan kegiatan lain seperti melakukan perhitungan. Spotlight ada sebuah fitur search yang menjadi keunggulan Mac dibanding Windows. Anda dapat menggunakan Spotlight dengan menekan cmd+space atau klik tombol lensa pembesar di pojok kanan atas, mengganti Spotlight dengan Alfred App sehingga tidak ada icon Spotlight di menu bar.

## Cara menjalankan aplikasi

1. Menjalankan Aplikasi cukup dilakukan dengan klik pada aplikasi yang ingin dijalankan. Lokasi aplikasi selalu disimpan pada Finder: Application. Lebih sederhanannya lagi, Anda dapat menjalankan Finder terus menekan cmd+shift+a. Satu cara lagi menjalankan aplikasi adalah dengan menjalankan Spotlight dan memasukkan nama aplikasi yang ingin dijalankan.
2. Menghentikan Aplikasi tidak bekerja seperti di Windows di mana Anda hanya perlu memilih tombol x di window titlebar. Pada Mac, Anda harus memilih nama aplikasi di menu bar terus quit. Atau Anda cukup menekan cmd+q pada aplikasi yang aktif.
3. Instalasi Aplikasi dapat dilakukan dengan drag & drop ke folder Application yang sudah disebut di atas. Tidak ada setup, dan tidak ada install. Native application pada umumnya cukup didrag & drop. Terdapat pengecualian pada beberapa aplikasi yang memiliki Installer.
4. Uninstallasi Aplikasi sebenarnya dapat dilakukan hanya dengan delete aplikasi di dalam folder Application. Tetapi akan terdapat sisa-sisa konfigurasi dan application support yang belum dihapus. Maka, untuk melakukan uninstallasi, Anda sebaiknya menggunakan aplikasi pihak ketiga seperti AppCleaner.

## MENGENAL SISTEM OPERASI ANDROID

**Android** adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang mencipta aplikasi mereka sendiri, untuk digunakan oleh bermacam peranti bergerak. Kerasnya persaingan dalam industri ponsel dunia, tidak semata melibatkan piranti kerasnya saja. Persaingan dalam sistem operasi yang menjadi penggerak fungsi-fungsi ponsel menarik juga untuk diamati.

Android dikembangkan oleh raksasa mesin pencari Google kurang lebih 1 tahun yang lalu. Korporasi yang didirikan Larry Page dan Sergey Brin itu mengembangkan smartphone yang tidak memerlukan lagi pulsa. Menghubungi ponsel orang cukup dengan account Gmail di ponsel.

Android menawarkan akses ke layanan komputasi mobile full suite Google. Ini termasuk Gmail, Youtube, Google Search, Google Maps, Google Talk dan Google Calendar. Bagi mereka yang suka menjelajah, sudah ditanamkan sebuah GPS receiver. Jadi penggunaanya dapat memanfaatkan beragam fitur Google Maps, termasuk My Location, Google Latitude, Google Street View, maupun informasi dan penunjuk arah.

## **JENIS- JENIS Versi dari Android**

### **1. Android versi 1.1**

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

### **2. Android versi 1.5 (Cupcake)**

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

### **3. Android versi 1.6 (Donut)**

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

### **4. Android versi 2.0/2.1 (Eclair)**

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* – aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

### **5. Android versi 2.2 (Froyo: Frozen Yoghurt)**

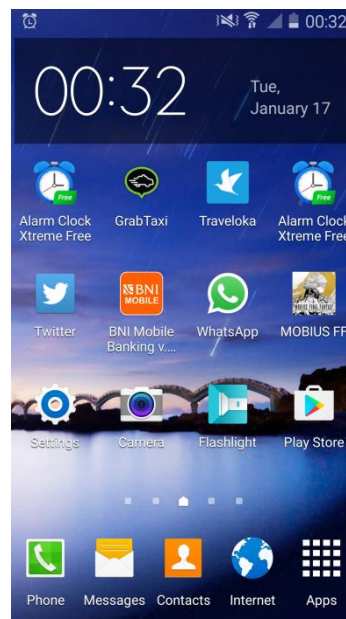
Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

## 6. Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

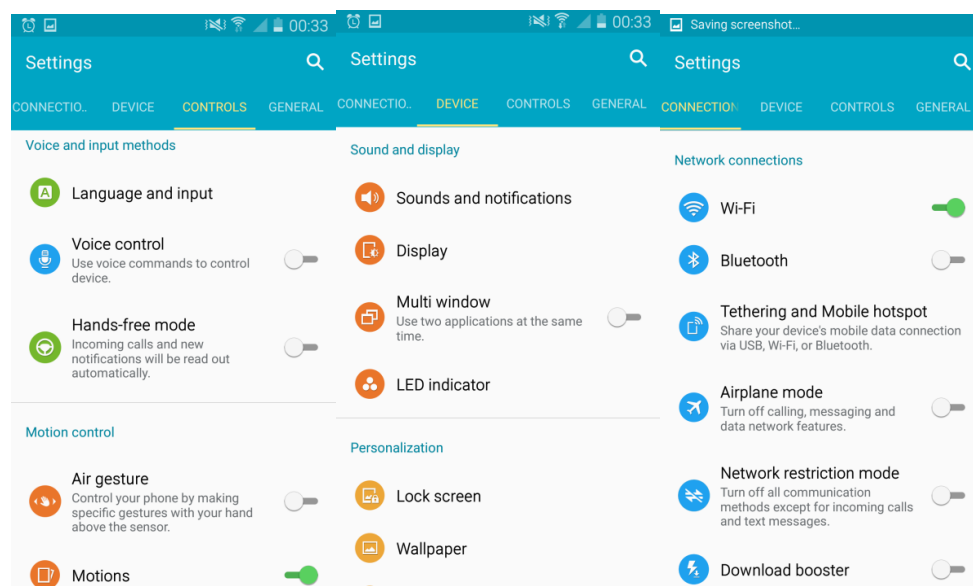
## 7. Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis.



Menu utama pada system operasi android

## Menu Setting pada android



Menu setting pada android KitKat

Pada menu setting pada android, terdapat beberapa sub menu, menu tersebut mengatur semua konfigurasi yang ada pada android. Sebagai contoh pada gambar diatas, terdapat menu connection, device, controls, dan general.

**Tugas**

1. Jelaskan kebutuhan minimum hardware untuk mac OS
2. Jelaskan kebutuhan minimum hardware untuk Android OS