

**Diktat Kuliah**

# **Pengantar Open Source dan Aplikasi**



**Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
2014**

## Kata Pengantar

Alhamdulillah, puji syukur penyusun panjatkan ke hadirat Allah SWT yang telah memberikan hidayah-Nya kepada penyusun sehingga berhasil menyelesaikan diktat kuliah Pengantar Open Source dan Aplikasi edisi 2014 ini. Diktat ini disusun untuk pegangan kuliah Pengantar Open Source dan Aplikasi bagi mahasiswa semester I program studi Sistem Informasi dan Teknik Informatika di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.

Diktat ini dirancang untuk pembelajaran dengan bobot 2 SKS (kuliah) atau 3 SKS (kuliah 2 SKS dan praktikum 1 SKS). Setelah mengikuti kuliah dengan panduan diktat ini, mahasiswa diharapkan memahami secara utuh tentang Open Source Software (perangkat lunak kode terbuka) dan Free Software (perangkat lunak bebas). Garis besar topik diktat ini meliputi definisi Open Source Software, Free Software, dan Proprietary Software, contoh-contoh produk teknologi informasi yang dibuat dengan cara Open Source, konsep pengembangan produk Open Source, keuntungan-kuntungan Open Source, lisensi-lisensi Open Source untuk software dan non software, isu-isu keamanan produk Open Source, model bisnis Open Source, instalasi sistem operasi Open Source (Linux/Unix), dan penggunaan sistem operasi Open Source dan beberapa aplikasi dasar Open Source untuk pekerjaan sehari-hari.

Sebagian isi diktat ini merupakan hasil kompilasi dari buku-buku bacaan dan artikel di internet yang penyusun masukkan dalam daftar pustaka, ditambah beberapa tulisan Toto Harjendro, S.Kom (Network Engineer STT NF) dalam buku “Ubuntu Troubleshooting”. Sebagian besar gambar dalam buku ini ditambahkan oleh Saudara Irfan Prasetyo (mahasiswa STT NF angkatan 2013).

Penyusun mengucapkan terima kasih kepada Saudara Toto Harjendro dan Irfan Prasetyo, serta semua pihak yang telah membantu dalam bentuk penulisan artikel dan buku dengan lisensi terbuka. Penyusun juga mengucapkan terima kasih kepada para mahasiswa dan dosen program studi Teknik Informatika dan Sistem Informasi di Sekolah Tinggi Teknologi Terpadu Nurul Fikri angkatan 2013, karena telah memberikan motivasi kepada penyusun untuk menyelesaikan dikat ini. Semoga karya sederhana ini berguna bagi para mahasiswa, dosen, dan siapa saja yang ingin menguasai teknologi informasi berbasis Open Source dan Openness lainnya.

Depok, September 2014

Penyusun,

**Drs. Rusmanto, M.M.**

# Daftar Isi

Kata Pengantar.....	ii
Daftar Isi .....	iii
Bab 1 Definisi dan Sejarah Open Source.....	1
1.1 Definisi Open Source.....	1
1.2 Sejarah Open Source .....	2
1.3 Filosofi Free Software dan Open Source Software.....	4
1.3.1 Filosofi Free Software Foundation.....	5
1.3.2 Filosofi Open Source Initiative.....	5
1.3.3 Persamaan dan Perbedaan Filosofi.....	7
Bab 2 Contoh Produk Open Source: Sistem Operasi .....	8
2.1 Pendahuluan .....	8
2.2 Software Sistem .....	9
2.2.1 Kernel, Inti Sistem Operasi.....	9
2.2.2 Sistem Operasi (Operating System) .....	9
2.2.3 Sejarah Linux .....	10
2.2.4 Distro (Distribution).....	11
2.2.5 Repo (Repository) .....	12
2.2.6 Ciri dan Spesifikasi Sistem Operasi Linux.....	13
2.2.7 Antaramuka Grafis atau GUI (Graphical User Interface) .....	13
2.2.8 Antarmuka Teks atau Command Line Interface.....	15
Bab 3 Contoh Produk Open Source selain Sistem Operasi .....	17
3.1 Pendahuluan.....	17
3.2 Middleware (Server Aplikasi, Web, Database, Infrastruktur).....	17
3.3 Software Aplikasi .....	19
3.4 Software Pemrograman .....	19
3.5 Aplikasi Enterprise.....	20
3.6 Web Application & Native Application.....	21
3.7 Catatan tentang Definisi Kelompok Software .....	21
Bab 4 Konsep Pengembangan Open Source .....	22
4.1 Model Pengembangan Open Source “Bazaar”.....	22
4.2 Studi Kasus Pengembangan Distro Linux.....	24
4.2.1 Sejarah Distro Linux .....	24
4.2.2 Perbandingan Distro Linux.....	25
4.3 Studi Kasus Pengembangan Linux BlankOn.....	26
4.4 Contoh Pengembangan Open Source yang Berawal dari Closed.....	29
4.5 Rangkuman.....	30
Bab 5 Komunitas Open Source .....	32
5.1 Sejarah dan Definisi Komunitas Open Source.....	32
5.2 Jenis Komunitas Open Source dan Contohnya.....	33
5.3 Alasan dan Manfaat Bergabung ke Komunitas .....	36
5.4 Organisasi Komunitas Open Source.....	37
Bab 6 Keuntungan/Kelebihan Open Source Software.....	38
6.1 Pendahuluan.....	38
6.2 Keamanan (Security).....	38
6.3 Ketersediaan/Kestabilan (Reliability/Stability).....	39
6.4 Standar Terbuka dan Tidak Bergantung Vendor.....	39
6.5 Mengurangi Ketergantungan Impor.....	40

6.6 Pengembangan Perangkat Lunak Lokal.....	40
6.7 Pelanggaran HaKI dan WTO.....	41
6.8 Bahasa dan Budaya Lokal (Localization).....	42
Bab 7 Openness di bidang Teknologi Informasi .....	43
7.1 Pendahuluan.....	43
7.2 Open Standard dan Open Format.....	43
7.2.1 Open Standard untuk Internet .....	43
7.2.2 Open Format.....	44
7.3 Open Content, Open Publication, Open Access.....	45
7.4 Open Hardware.....	46
7.5 Open Movie .....	46
Bab 8 Lisensi-lisensi Software Open Source .....	47
8.1 Pengertian Hak Cipta, Lisensi, dan Kategori Software .....	47
8.1.1 Hak Cipta dan Lisensi.....	47
8.1.2 Katagori Software Berdasar Kelompok Lisensi.....	48
8.2 Siapa yang Mengakui Lisensi Free/Open Source Software .....	49
8.3 Perbandingan Lisensi FOSS.....	50
8.4 GNU General Public License (GNU GPL atau GPL).....	52
8.4.1 Copyleft.....	53
8.4.2 Syarat dan Ketentuan Utama GPL .....	53
8.5 GNU Lesser General Public License (GNU LGPL atau LGPL).....	54
8.6 Model Lisensi BSD (Berkeley Software Distribution).....	55
8.7 Lisensi Gabungan (Dual/Multiple Licensing).....	55
Bab 9 Lisensi-lisensi Open untuk Bukan Software .....	56
9.1 GNU Free Documentation License (GNU FDL atau FDL).....	56
9.2 Creative Commons Licenses .....	56
Bab 10 Aspek Keamanan Open Source.....	59
10.1 Pendahuluan.....	59
10.2 Keamanan Komputer .....	59
10.3 Bentuk, Cara, dan Target Gangguan Keamanan Komputer.....	60
10.4 Mengapa Open Source Relatif Lebih Aman? .....	61
10.5 Enam Mitos (Salah Pengertian) Keamanan Open Source.....	62
10.6 Beberapa Kesalahan dan Prosedur Keamanan yang Baik.....	63
Bab 11 Model Bisnis Open Source .....	65
11.1 Model Bisnis Produk Proprietary.....	65
11.2 Bisnis (di atas) Open Source.....	65
11.3 Bisnis Kombinasi Open Source dan Tidak Open Source .....	66
11.4 Bisnis Proprietary di atas Open Source.....	67
11.5 Studi Kasus .....	67
Bab 12 Perbandingan Sistem Open Source dan Proprietary.....	68
12.1 Pendahuluan.....	68
12.2 Perbandingan Linux dengan Windows dan OS X .....	68
12.3 Perbandingan Android dengan Windows Phone dan iOS.....	71
12.4 Perbandingan File/Direktori Linux, Windows, dan OS X .....	72
Bab 13. Pengantar Jaringan Komputer dengan Linux.....	74
13.1 Pendahuluan: Bekerja dalam Modus Teks.....	74
13.2 Mematikan Network Manager.....	74
13.3 Jaringan Kabel Ethernet.....	75
13.4 Perintah ifconfig.....	75
13.5 Perintah route.....	76
13.6 Perintah echo untuk Resolver.....	77

13.7 Perintah dhclient.....	77
13.8 Perintah ping.....	77
13.9 Mengedit File Resolver.....	77
13.10 Perintah host dan dig.....	78
13.11 Setting Jaringan Melalui Menu Desktop.....	78
Bab 14 Instalasi Sistem Operasi Ubuntu dan Troubleshooting.....	81
14.1 Backup Data.....	81
14.2 Memulai Install Ubuntu.....	82
14.3 Perlu tidaknya Akses Internet.....	82
14.4 Mengedit Partisi Hard disk.....	83
14.5 Mengatur Waktu dan Keyboard.....	85
14.6 Username dan Password.....	85
14.7 Login Pertama setelah Instalasi.....	86
14.8 Troubleshooting .....	87
14.8.1 Tidak bisa login sebagai root .....	87
14.8.2 Tidak bisa melakukan sudo .....	87
14.8.3 Username dan password sudah benar tetapi tidak bisa login.....	87
14.8.4 Lupa password user admnistrator.....	88
14.8.5 Program identifikasi software .....	88
Daftar Pustaka.....	94

# Bab 1 Definisi dan Sejarah Open Source

Tujuan Umum: Mampu memahami definisi dan sejarah Open Source, Free Software, dan Proprietary Software.

Tujuan Khusus:

- a) Dapat menjelaskan arti Open Source Software, Free Software, FOSS, dan Proprietary Software.
- b) Dapat menjelaskan secara ringkas sejarah Open Source, mulai dari era sebelum 1984, 1984 – 1991, 1991 – 1998, 1998 – sekarang.

## 1.1 Definisi Open Source

Free/Open Source Software (FOSS) adalah istilah yang dapat memiliki tiga makna, yakni **cara pengembangan software**, **jenis lisensi software**, dan **model bisnis software**. Sebelum membahas tiga hal ini dalam beberapa bab kemudian, kita mulai dengan membahas definisi umum Open Source.

Istilah "Open Source" merupakan penyebutan singkat dari istilah Free/Open Source Software (FOSS) atau Free/Libre/Open Source Software (FLOSS). Menurut David Wheeler, secara umum program yang dinamakan free software (perangkat lunak bebas) atau open source software (perangkat lunak sumber terbuka) adalah program yang pembuatnya memberi izin kepada pengguna menjalankan program itu untuk apa saja, mempelajari dan memodifikasi program itu, dan menyebarluaskan kopi program asli atau program yang sudah dimodifikasi, tanpa harus membayar royalti kepada pengembang sebelumnya. (Sumber: [http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)).

Program yang dimaksud di sini adalah kode atau perintah yang dibuat manusia untuk dijalankan pada komputer. Komputer yang dimaksud di sini tidak hanya berbentuk komputer meja (desktop) atau komputer jinjing (laptop), namun juga komputer dalam arti luas, mulai dari komputer berbentuk telepon (smartphone), komputer tablet, televisi, jam tangan, kacamata, kamera, konsol game, hingga komputer besar (*mainframe*) dan komputer "raksasa" (*supercomputer*).



Gambar 1.1 Lambang Open Source Initiative (Wikipedia)

Dengan kata lain, open source adalah cara suatu program dikembangkan, yaitu dengan tidak merahasiakan kode sumber (source code) programnya. Source code adalah kode program yang dapat dimengerti manusia. Kode program yang hanya dimengerti mesin komputer disebut binary code (kode biner, misal dilambangkan dengan angka 0 dan 1 atau off dan on).

Kode biner dapat dihasilkan oleh suatu program, misal compiler, yang mengompilasi source code menjadi binary. Program yang hanya tersedia dalam bentuk binary disebut juga Closed Source.

FOSS juga dapat diartikan sebagai jenis lisensi atau pernyataan hak cipta pengembang atau pemilik hak cipta suatu program yang memberikan hak (izin) menggunakan, mengubah, dan menyebarluaskan program tersebut kepada orang lain, tanpa mewajibkan orang lain itu membayar izin atau royalti.

Banyak jenis lisensi FOSS, salah satunya yang paling terkenal saat ini adalah GNU GPL (General Public License) yang antara lain menyatakan bahwa pembuat program berlisensi GPL memberikan hak kepada siapa pun untuk menggunakan dan mengubah program tersebut dengan syarat tidak mengubah lisensinya. Turunan software GPL akan tetap GPL. Hak cipta umumnya disebut copyright, sedangkan hak cipta GPL ini diberi sebutan juga copyleft. Jenis-jenis lisensi software dan penjelasannya akan dibahas pada bab tersendiri.



Gambar 1.2 Logo GNU (*Wikipedia*)

Kebalikan dari Free/Open Source Software adalah Proprietary Software / Closed Source Software, yakni program yang kode sumbernya dirahasiakan oleh pengembang. Proprietary dapat diterjemahkan sebagai "berpemilik" yang artinya hak menggunakan, hak memodifikasi, dan hak menyebarluaskan program proprietary "hanya ada pada" pemilik hak cipta. Orang lain harus mendapatkan izin dari pemilik hak cipta untuk menggunakan, memodifikasi, dan menyebarluaskan program proprietary. Disebut "kebalikan" karena pemilik hak cipta FOSS tidak melarang penggunaan, pemodifikasian, dan penyebarluasan program ciptaannya atau program hasil modifikasi/turunan. Jika ada program yang boleh dikopi/digunakan, tapi pemilik hak cipta tidak menyediakan kode sumber sehingga tidak dapat diubah, atau kode sumber tersedia tapi tidak boleh diubah, atau dapat diubah tapi tidak boleh disebarluaskan, maka program itu bukan FOSS.

## 1.2 Sejarah Open Source

Gerakan FOSS dimulai dalam budaya "hacker" yang terjadi pada beberapa laboratorium ilmu komputer (Stanford, Berkeley, Carnegie Melon, dan MIT) di tahun 1960-an dan 1970-an. Komunitas pemrogram jumlahnya masih sedikit dan saling terkait secara dekat. Kode program disebarluaskan di antara anggota komunitas. Jika Anda membuat perbaikan, Anda diharapkan untuk mengirim kode Anda ke komunitas pengembang.

Sistem operasi yang dikembangkan dengan cara Open Source atau Free Software telah ada sejak dekade 1970-1980, sebelum dua organisasi Free Software Foundation dan Open Source Initiative didirikan. Sistem operasi Unix pertama dikembangkan dari 1979 hingga 1974,

menggunakan cara berbagi kode sumber program, alias cara free software atau open source. Namun di akhir 1970-an dan awal 1980-an Unix berkembang menjadi sistem operasi proprietary, yang diikuti sistem operasi lain seperti MS DOS, MS Windows, Novell Netware, dan Apple Macintosh.

Gerakan FOSS boleh dikatakan dimulai sejak awal mula industri komputer, meskipun tidak dinyatakan secara formal atau dengan konsep yang jelas. Hanya saja pada akhir 1970-an dan awal 1980-an terjadi konflik antara konsep saling berbagi perangkat lunak dengan konsep perangkat lunak berpemilik (proprietary). Acuan awal konflik ini dibuat oleh William H. Gates III (Bill Gates), dalam pernyataannya yang terkenal "An Open Letter to Hobbyists" (Surat Terbuka kepada para Penghobi). Dalam surat tertanggal 3 Februari 1976 itu ia mencemooh budaya berbagi perangkat lunak yang telah umum berlaku sebelum itu, "...*Perangkat keras harus dibeli, tetapi perangkat lunak menjadi sesuatu untuk dibagi. Siapa yang mau peduli jika orang yang bekerja untuk itu mengambil bayaran? ...*"

Pengembang perangkat lunak proprietary ingin mengambil kesempatan pada tahun-tahun berikutnya. Di laboratorium kecerdasan buatan (Artificial Intelligence) MIT pada awal 1980-an, sebuah perusahaan bernama Symbolics didirikan, lalu mengambil kode-kode yang tersedia secara bebas (bahasa pemrograman LISP) dan menjadikannya proprietary (tidak tersedia bebas alias berpemilik). Dalam prosesnya, ini berarti menghapus budaya berbagi perangkat lunak di laboratorium MIT saat itu. Namun, perusakan ini akhirnya akan menghasilkan kreasi FSF dan budaya FOSS saat ini.

Richard Stallman, salah satu anggota laboratorium MIT saat itu, terkejut atas lanjutan persitiwa tersebut. Ini kemudian membentuk pandangannya terhadap perangkat lunak proprietary, dan membangkitkan keinginannya untuk membuat sistem operasi yang free (bebas). Projek GNU (GNU is Not UNIX) berdiri pada Januari 1984. Dalam dekade berikutnya projek GNU menghasilkan berbagai program atau tool penting merupakan bagian dari sistem operasi. Yayasan perangkat lunak bebas (FSF) didirikan setahun kemudian untuk mempromosikan perangkat lunak dan projek GNU. Namun, hingga 1991 projek GNU belum menghasilkan sistem operasi lengkap karena masih ada kekurangan pada bagian kritis, yaitu kernel.

Kernel merupakan inti atau jantung dari sistem operasi. Linus Torvalds yang saat itu mahasiswa tahun kedua Universitas Helsinki membuat dan mendistribusikan kernel seperti UNIX. Sejalan dengan tujuan pengembangan FOSS, kernel yang kemudian diberi nama Linux itu tersebar secara luas, dikembangkan, dan diaplikasikan menjadi inti dari sistem operasi GNU/Linux.



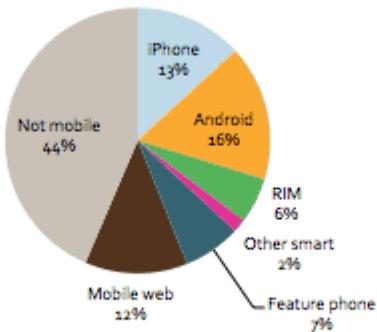
Gambar 1.3 Linus Torvalds (kiri) dan Richard Stallman (kanan) (Wikipedia)

Ada beberapa projek FOSS yang sedang berjalan dalam waktu bersamaan, antara lain server DNS BIND, bahasa pemrograman Perl, dan sistem operasi BSD. Sebagian besar projek itu kemudian bergabung atau saling menguatkan.

Sistem operasi GNU/Linux terus tumbuh secara cepat dengan makin lengkap fitur dan kemampuannya. Pada 1997, Linux meledak menjadi berita media, sesuai dengan perkiraan IDC (International Data Corporation) bahwa Linux telah menguasai 25% sistem operasi server dan memiliki pertumbuhan 25% per tahun.

Pada 1998, sebagai tanggapan terhadap Netscape yang merilis kode sumber program Netscape Navigator sebagai FOSS, sekelompok pengembang FOSS bergerak bersama dan label "Open Source" digulirkan. Gerakan ini lalu membentuk OSI (Open Source Initiative) dan OSD (Open Source Definition). Tujuan utama gerakan ini untuk mengajak dunia bisnis memberi penekanan kepada proses pengembangan FOSS, dan mengalihkan perhatian dari gerakan perangkat lunak bebas (Free Software) yang kontroversial saat itu.

Pada 1999, perusahaan distributor GNU/Linux Red Hat berhasil go public atau IPO (Initial Public Offering) dengan meraup dana dari pasar saham senilai US\$ 4,8 miliar atau Rp 48 trilyun jika 1US\$ = Rp 10.000,-. Sebagai anak baru dari FOSS, kesuksesan GNU/Linux menunjukkan bahwa era FOSS telah benar-benar tiba.



[Source: Facebook, Enders Analysis]

Gambar 1.4 Grafik Pengguna Facebook 2012 (<http://www.trenologi.com>)

Era 2010-an ini terjadi ledakan besar penggunaan inti sistem operasi atau kernel Linux di perangkat bergerak (*mobile devices*) seperti smartphone dan komputer tablet, sejak Google memimpin pengembangan Android secara open source (2008). Pada 2013, jumlah perangkat komputer dengan sistem operasi Linux-Android diprediksi telah melewati angka satu miliar. Menurut Linux Foundation pada awal Oktober 2013, setiap hari ada 1,3 juta perangkat Android baru.

Sejarah singkat itu memberikan bukti bahwa produk FOSS dapat dijadikan bisnis, bukan software gratis (Freeware/Shareware), meskipun tidak ada biaya izin atau royalti, karena orang dapat mendapatkan uang dari mengembangkan software, memodifikasi software atau menjual suatu software dipadukan dengan software lain, menyediakan jasa dukungan teknis, menyediakan jasa pelatihan, menjual software dalam bentuk kemasan CD dan dokumentasinya, menjual software disatukan dengan perangkat keras, dan sebagainya.

### 1.3 Filosofi Free Software dan Open Source Software

Dua istilah free software dan open source software sering disatukan dalam bentuk singkatan

FOSS (Free/Open Source Software), karena dua organisasi yang mencetuskan dua istilah itu memiliki misi sama dalam mengembangkan dan menyebarluaskan program, namun secara filosofi sedikit berbeda.

Ada dua filosofi pokok pada kata FOSS, yaitu filosofi dari FSF (Free Software Foundation) atau Yayasan perangkat Lunak Bebas, dan filosofi dari OSI (Open Source Initiative) atau Inisiatif Sumber Terbuka. Kita mulai pembahasan dengan filosofi FSF, sesuai dengan urutan sejarah dan karena posisi FSF sebagai pionir dalam gerakan FOSS ini. Tokoh utama gerakan FSF adalah Richard M. Stallman, sedangkan tokoh gerakan OSI adalah Eric S. Raymond dan Bruce Perens.



*Gambar 1.5 FSF (Free Software Foundation) atau Yayasan perangkat Lunak Bebas (Wikipedia)*

### **1.3.1 Filosofi Free Software Foundation**

Menurut FSF, perangkat lunak bebas mengacu pada kebebasan para penggunanya untuk menjalankan, menggandakan, menyebarluaskan/mendistribusikan, mempelajari, mengubah dan meningkatkan kinerja perangkat lunak. Tepatnya, mengacu pada empat jenis kebebasan bagi para pengguna perangkat lunak, lisensi Free Software memberikan:

1. Kebebasan untuk menjalankan programnya untuk tujuan apa saja (kebebasan 0).
2. Kebebasan untuk mempelajari bagaimana program itu bekerja serta dapat disesuaikan dengan kebutuhan pengguna (kebebasan 1). Akses pada kode program merupakan suatu prasyarat.
3. Kebebasan untuk menyebarluaskan kembali hasil salinan program tersebut sehingga dapat membantu sesama (kebebasan 2).
4. Kebebasan untuk meningkatkan kinerja program, dan dapat menyebarkannya ke khalayak umum sehingga banyak orang menikmati keuntungannya (kebebasan 3). Akses pada kode program merupakan suatu prasyarat juga.

### **1.3.2 Filosofi Open Source Initiative**

Filosofi OSI agak berbeda. Ide dasar open source sangat sederhana. Jika para pemrogram dapat mempelajari, mendistribusikan ulang, dan mengubah kode sumber sebagian perangkat lunak, maka perangkat lunak itu berkembang. Masyarakat mengembangkannya, mengaplikasikannya, dan memperbaiki kelemahannya. Berikut ini terjemahan bebas dari syarat sebuah program memiliki lisensi Open Source menurut OSI ([opensource.org/osd](http://opensource.org/osd)):

#### *1. Pendistribusian Ulang Secara Bebas (Free Redistribution)*

Lisensi tersebut tidak akan menghalangi pihak manapun dalam menjual atau memberikan software tersebut sebagai sebuah komponen dari suatu distribusi atau kumpulan software dari beberapa sumber yang berbeda. Lisensi itu juga tidak memerlukan pembayaran royalti atau biaya lain untuk penjualan software. Dibolehkan menjual software open source dalam berbagai bentuk, tapi bukan menjual lisensi (surat izin).

## *2. Kode Sumber (Source Code)*

Program harus dilengkapi kode sumber dan mengizinkan distribusi dalam bentuk kode sumber maupun bentuk jadi (binary). Jika produk tidak didistribusikan dengan kode sumber, sebuah sarana publikasi (pemberitahuan) yang baik harus disediakan untuk memperoleh kode sumber tersebut dengan biaya reproduksi yang wajar, atau memindahkan dari internet tanpa biaya, misalnya. Kode sumber tersebut harus dalam bentuk yang membuat programer dapat memodifikasinya. Kode sumber yang sengaja dibuat untuk memperdaya atau menipu tidak diizinkan.

## *3. Karya-karya Turunan (Derived Works)*

Lisensi tersebut harus memperbolehkan karya-karya modifikasi atau turunan, dan mengizinkannya untuk didistribusikan dalam bentuk yang sama seperti lisensi software asalnya.

## *4. Integritas Kode Sumber Pencipta (Integrity of the Author's Source Code)*

Lisensi dapat mencegah "pendistribusian kode sumber hanya dalam bentuk modifikasi" jika lisensi mengizinkan pendistribusian dalam bentuk "patch files" (file tambahan) disertai kode sumber yang bertujuan memodifikasi program pada masa pembuatan. Lisensi harus secara tersurat mengizinkan pendistribusian software yang dibuat dari hasil modifikasi. Lisensi dapat mensyaratkan ada modifikasi atau karya turunan jika program akan menggunakan nama baru atau versi berbeda dari software asal.

## *5. Tidak Ada Diskriminasi terhadap Individu atau Kelompok (No Discrimination Against Persons or Groups)*

Lisensi tidak diperbolehkan menciptakan diskriminasi terhadap pengguna individu/personal atau kelompok.

## *6. Tidak Ada Diskriminasi terhadap Bidang Pekerjaan (No Discrimination Against Fields of Endeavor)*

Lisensi tersebut tidak boleh membatasi seseorang dari menggunakan program itu dalam suatu bidang pekerjaan tertentu. Sebagai contoh, tidak ada pembatasan program tersebut terhadap penggunaan dalam bidang bisnis, atau terhadap pemanfaatan dalam bidang riset genetik.

## *7. Pendistribusian Lisensi (Distribution of License)*

Hak-hak yang dicantumkan pada program harus dapat diterapkan pada semua yang menerima pendistribusian program, tanpa perlu dikeluarkan lisensi tambahan untuk pihak-pihak penerima program.

## *8. Lisensi Tidak Boleh Bersifat Spesifik terhadap Suatu Produk (License Must Not Be Specific to a Product)*

Hak-hak yang tercantum pada lisensi program tidak boleh tergantung distribusi software tertentu. Jika program dipisahkan dari distribusi tersebut dan digunakan atau didistribusikan dalam bentuk lain sesuai lisensi program itu, maka semua pihak yang menerima program harus memiliki hak yang sama seperti ketika didistribusikan dalam bentuk asalnya.

## *9. Lisensi Tidak Boleh Membatasi Software lain (License Must Not Restrict Other Software)*

Lisensi tidak boleh membatasi software lain. Sebagai contoh, lisensi itu tidak boleh memaksakan bahwa program lain yang didistribusikan pada media yang sama harus bersifat open source.

## *10. Lisensi Harus Teknologi-Netral (License Must Be Technology-Neutral)*

Tidak ada ketentuan dalam lisensi yang dapat didasarkan pada teknologi tertentu (individual) atau tipe antaramuka tertentu saja.

### **1.3.3 Persamaan dan Perbedaan Filosofi**

Bila dibandingkan definisi Free Software, definisi Open Source yang terdiri dari 10 klausul itu relatif lebih longgar dan detail dibandingkan 4 klausul Free Software. Klausul tentang akses terhadap kode sumber (1), hak pemakai untuk memperbanyak dan menyebarkan program aslinya (3), dan tidak diskriminatif (5, 6, 8, 9, 10), meski tidak dinyatakan secara eksplisit sebenarnya semua klausul itu juga terkandung dalam definisi Free Software. Klausul 7 dalam Open Source mencegah agar kode sumber tidak tertutup lagi, merupakan konsep inti dari Free Software. Pengakuan terhadap pembuat program juga tidak secara eksplisit disebutkan pada Free Software, sedangkan ini masuk pada klausul 4 definisi Open Source.

OSI difokuskan pada nilai-nilai teknis dalam pembuatan perangkat lunak yang berdaya guna dan dapat dihandalkan, dan pendekatan istilah OSI ini dinilai oleh beberapa pihak lebih sesuai kebutuhan bisnis daripada filosofi FSF. OSI tidak terlalu fokus pada isu moral seperti yang ditegaskan FSF, dan lebih fokus pada manfaat praktis metode pengembangan terdistribusi FOSS.

Meskipun filosofi dasar kedua gerakan ini berbeda, FSF dan OSI berbagi area yang sama dan bekerja sama dalam hal-hal praktis, seperti pengembangan perangkat lunak, usaha melawan perangkat lunak proprietary, paten perangkat lunak, dan sejenisnya. Richard Stallman mengatakan bahwa gerakan perangkat lunak bebas dan gerakan open source merupakan dua "partai politik" dalam komunitas yang sama.

## Bab 2 Contoh Produk Open Source: Sistem Operasi

Tujuan Umum: Mampu memahami fungsi dasar salah satu sistem operasi Open Source, distribusi, dan repositori.

Tujuan Khusus:

- Dapat menjelaskan arti GNU/Linux sebagai sistem operasi dan menggunakannya.
- Dapat menjelaskan arti distro GNU/Linux atau varian Linux.
- Dapat menjelaskan arti repo, termasuk contoh isinya yang Open Source dan tidak Open Source.

### 2.1 Pendahuluan

Pengelompokan secara sederhana, ada dua jenis software, yaitu sistem dan aplikasi. Definisi lebih luas, bahasa pemrograman bukan aplikasi, sehingga ada tiga jenis software, yakni sistem, aplikasi, dan pemrograman. Namun berdasarkan tingkatan/lapisan pada sistem komputer atau sistem informasi berbasis komputer, software dapat dikelompokkan menjadi empat, yakni *system software*, *middleware software*, *application software*, dan *programming tool software*. Pengelompokan itu berlaku untuk produk proprietary maupun Free/Open Source Software (FOSS).



Gambar 2.1 Perangkat Lunak Open Source (Wikipedia)

Contoh-contoh produk atau karya FOSS yang diberikan dalam tulisan ini lebih ditujukan untuk sistem operasi Linux. Namun hampir semua contoh software open source ini juga tersedia untuk sistem operasi open source lain seperti FreeBSD. Sebagian contoh produk FOSS juga tersedia untuk sistem operasi proprietary seperti Mac OS X dan MS Windows, misal aplikasi perkantoran LibreOffice dan browser web Firefox.

Berikut ini beberapa data pendukung:

- Hasil survei IDC yang diumumkan pada November 2013, 81% telepon cerdas (*smartphone*) menggunakan sistem operasi Android.
- Google mengungkapkan pada September 2013 sudah lebih dari satu miliar komputer

- bergerak (telepon cerdas dan komputer tablet) menggunakan sistem operasi Android.
- Yayasan Linux ([linuxfoundation.org](http://linuxfoundation.org)) menyebutkan 8 dari 10 transaksi dunia menggunakan Linux, 98% supercomputer di dunia menggunakan Linux, 1,3 juta Linux Android baru diaktifkan tiap hari, dan 700 ribu TV berisi Linux terjual tiap hari.

## 2.2 Software Sistem

Software sistem adalah program yang mengendalikan perangkat keras komputer dan perlengkapannya, misal sistem operasi dan utilitasnya. Contoh sistem operasi inti adalah kernel Linux, yang bila dipasang dengan berbagai utilitas membentuk sistem operasi lengkap. Contoh utilitas Linux adalah perintah untuk mengubah password (*passwd*), membuat direktori (*mkdir*), melihat isi direktori (*ls*), dan melihat ukuran ruang harddisk (*df*). Kumpulan software sistem yang lengkap pada sistem operasi Linux disebut *Linux Tools & Utilities*.

Sistem operasi Open Source yang paling banyak digunakan orang di seluruh dunia saat ini (awal 2014) adalah Linux. Linux pada dasarnya berupa kernel, yaitu inti sistem operasi komputer. Kernel Linux digabung dengan program-program lainnya menjadi sistem operasi varian Unix (*Unix-like*). Kernel Linux saat ini paling banyak terpasang di komputer, terutama karena kernel Linux digunakan oleh sistem operasi komputer bergerak (*mobile devices*) Android yang sangat populer saat ini, Chrome/Chromium, Tizen, Sailfish, Ubuntu, dan WebOS. Linux juga menjadi inti sistem operasi server-server perusahaan terkenal di dunia seperti Google, Facebook, IBM, Oracle, Twitter, Amazon, dan Microsoft.

### 2.2.1 Kernel, Inti Sistem Operasi

Inti sistem operasi komputer disebut kernel. Saat pertama dikembangkan oleh Linus Torvalds dan dirilis pada 1991, Linux hanya nama kernel atau inti sistem operasi. Kemudian kernel Linux dilengkapi pustaka program dari GNU (Glibc) dan program utilitas lainnya membentuk sistem operasi GNU-Linux atau GNU/Linux, dengan panggilan singkat Linux. Linux tersedia untuk berbagai jenis komputer, mulai dari laptop, desktop, server, mainframe, hingga super komputer.

### 2.2.2 Sistem Operasi (*Operating System*)

Di 2010-an ini kernel Linux merambah luas ke perangkat bergerak seperti handphone dan tablet, melalui sistem operasi Android. Android tidak menggunakan Glibc tapi Bionic-libc sehingga Android adalah Linux yang tidak termasuk GNU-Linux.



Gambar 2.2 Tampilan sistem operasi Unix FreeBSD (Wikipedia)

Sistem operasi komputer dapat dikelompokkan berdasarkan jenis komputer saat ini menjadi dua, yakni sistem operasi komputer “biasa” (*laptop*, *desktop*, *server*, dsb.) dan sistem operasi komputer bergerak (*tablet pc*, *smart phone*, dsb.) atau disebut juga sistem tertanam (*embedded system*). Kata “biasa” untuk menunjukkan jenis komputer yang sudah ada sebelum era smart phone dan tablet pc yang dikenal sebagai perangkat bergerak (*mobile devices*).

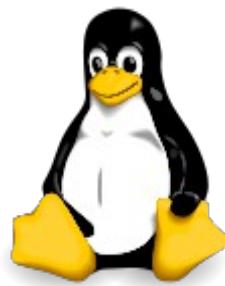
Selain Linux, beberapa sistem operasi open source yang juga populer antara lain dari keluarga BSD (Berkeley Software Distribution), yakni FreeBSD, NetBSD, OpenBSD, dan DragonFly BSD. Jika hak cipta Linux dinyatakan atau dilisensikan sebagai GPL sehingga turunannya harus tetap GPL, maka lisensi BSD mengizinkan pembuatan turunan (misal hasil modifikasi FreeBSD) tidak harus BSD, sehingga FreeBSD bisa berubah menjadi tidak open source. Contoh, FreeBSD dimodifikasi oleh perusahaan komputer Apple menjadi sistem operasi Mac OS X, yang kemudian juga merambah ke hp (iPhone) dan tablet (iPad) dengan nama sistem operasi iOS. Mac OS X dan iOS tidak open source, meskipun berasal dari open source.

Berikut ini beberapa contoh sistem operasi Open Source selain Linux.

- GNU Hurd, dikembangkan oleh Richard Stallman dan kawan-kawan melalui Yayasan Free Software, termasuk Unix-like. Setelah ada Linux yang lisensinya GPL, GNU Linux jauh lebih cepat berkembang dan kemudian sangat populer dibandingkan GNU Hurd.
- Minix (yang menjadi inspirasi bagi Linus mengembangkan Linux), termasuk Unix-like, dikembangkan oleh Prof. Tanenbaum dan dirilis pertama kali pada 1987 dengan lisensi Open Source BSD.
- Keluarga BSD (FreeBSD, OpenBSD, NetBSD, PC BSD, DragonFly), juga termasuk Unix-like. Darwin disebut juga turuan FreeBSD yang kemudian menjadi dasar sistem operasi Tidak Open Source dari perusahaan Apple Mac OS X untuk komputer desktop dan server, dan iOS untuk komputer tablet dan handphone.
- OpenSolaris, varian Open Source dari Solaris yang proprietary Sun Microsystem (sekarang dimiliki Oracle), termasuk Unix-like.
- ReactOS, bukan Unix-like tapi Windows-like, dibuat dari dasar tanpa menggunakan kode sumber sistem operasi MS Windows.
- FreeDOS, bukan Unix-like, seperti MS DOS.

### 2.2.3 Sejarah Linux

**Linux** (diucapkan 'linəks atau '/linoks/) adalah nama yang diberikan kepada inti (kernel) sistem operasi komputer bertipe Unix yang pertama kali dikembangkan oleh Linus Torvalds dan dirilis pada 5 Oktober 1991. Linux merupakan salah satu contoh hasil pengembangan perangkat lunak bebas dan sumber terbuka utama. Seperti perangkat lunak bebas dan sumber terbuka pada umumnya, kode sumber Linux dapat dimodifikasi, digunakan, dan didistribusikan kembali secara bebas oleh siapa saja.



Gambar 2.3 Pinguin Tux sebagai Maskot Linux (Wikipedia)

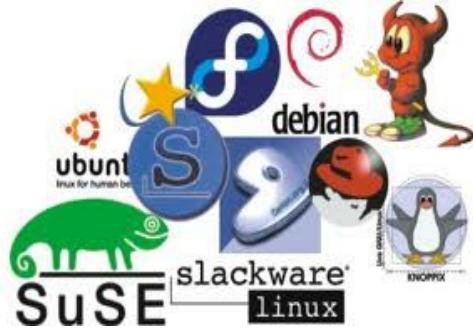
Nama "Linux" berasal dari nama pembuatnya, Linus Torvalds. Pada waktu itu, selain inti sistem operasi yang dibuat Linus, beberapa program sistem dan pustaka (*Library*) bahasa C berasal dari program-program GNU (*GNU is Not Unix*) yang dikembangkan sejak 1983 oleh Richard Stallman dan kawan-kawan, sehingga muncul nama sistem operasi **GNU/Linux**. Beberapa sistem operasi Linux yang tidak menggunakan program pustaka dari GNU (GNU Library C), misal Android yang menggunakan Bionic Library C tidak disebut GNU/Linux.

Linux telah lama dikenal untuk digunakan sebagai sistem operasi komputer server, dan didukung oleh perusahaan-perusahaan komputer ternama seperti Intel, Dell, Hewlett-Packard, IBM, Novell, Oracle, Red Hat, dan Google. Linux digunakan sebagai sistem operasi di berbagai macam jenis perangkat keras komputer, termasuk komputer desktop, superkomputer, dan sistem benam (*embedded system*) seperti pembaca buku elektronik, sistem permainan video (PlayStation 2, PlayStation 3 dan Xbox), komputer tablet, telepon genggam, router, dan lain-lain. Para pengamat teknologi informatika beranggapan kesuksesan Linux dikarenakan Linux tidak bergantung kepada vendor (*vendor independence*), biaya operasional yang rendah, dan kompatibilitas yang tinggi dibandingkan versi Unix tak bebas (*Proprietary*), serta faktor keamanan dan kestabilannya yang tinggi Unix-like dibandingkan dengan sistem operasi bukan Unix seperti Microsoft Windows. Ciri-ciri ini juga menjadi bukti atas keunggulan model pengembangan perangkat lunak sumber terbuka (*Open Source Software*).

Sistem operasi Linux yang digabungkan dengan berbagai program lain dikenal dengan istilah distribusi Linux (*Linux distribution*) atau distro Linux. Paket CD/DVD distro Linux umumnya sudah termasuk perangkat-perangkat lunak pendukung seperti server web, bahasa pemrograman, basisdata, tampilan desktop (*desktop environment*) seperti GNOME, KDE dan Xfce, paket aplikasi perkantoran (*office suite*) seperti LibreOffice, OpenOffice.org, Caligra Office (dulu bernama KOffice), Abiword, Gnumeric, dan lain-lain.

#### 2.2.4 Distro (*Distribution*)

Sistem operasi keluarga Unix seperti Linux dan FreeBSD dapat dikemas dalam bentuk mulai dari yang sangat kecil, CD, hingga beberapa DVD. Misalnya Linux untuk menjalankan software web server tidak mengharuskan menjalankan aplikasi desktop atau aplikasi GUI (Graphical User Interface), sehingga cukup dikemas dalam sebuah CD. Desktop tidak termasuk software sistem tapi software aplikasi. Web server juga tidak termasuk software sistem, tapi software menengah (middleware). Kemasan sistem operasi Linux dan program-program yang disatukan dalam suatu kemasan, misal DVD, disebut distro, singkatan dari *distribution*.



Gambar 2.4 Beberapa Contoh Distro Linux (Wikipedia)

Contoh distro yang sangat tua dan terkenal di dunia adalah Debian, Red Hat, dan Slackware. Setiap distro tua memiliki "distro turunan" atau hasil modifikasi. Distro Debian memiliki turunan yang menjadi lebih terkenal dari Debian, yakni Ubuntu (internasional) dan BlankOn (Indonesia). Contoh turunan Red Hat adalah CentOS dan Fedora yang di Indonesia diturunkan menjadi IGN (Igos Nusantara). Contoh turunan Slackware adalah SUSE (openSUSE) dan VectorLinux. Daftar distro Linux dan sistem operasi FOSS lainnya tersedia sangat lengkap di [www.distrowatch.com](http://www.distrowatch.com).

### 2.2.5 Repo (*Repository*)

Sebuah distro yang biasanya hanya berupa CD atau DVD belum tentu memenuhi semua kebutuhan pengguna yang bervariasi antara pengguna satu dengan yang lain. Untuk itu pengembang Linux menyediakan server *repository*, disingkat repo. Repo juga dapat berupa kumpulan DVD. Sebagai contoh, distro Debian, Ubuntu dan BlankOn memiliki repo yang kalau dijadikan DVD menjadi lebih dari 10 DVD. Anda dapat membeli repo dalam bentuk DVD itu melalui web, misal [juragan.kambing.ui.ac.id](http://juragan.kambing.ui.ac.id). Repo Ubuntu 13.04 berjumlah 13 keping DVD, atau 14 keping DVD termasuk DVD *installer*.

The screenshot shows the homepage of the KAMBING.ui.ac.id website. The header features a logo of a sheep and the text "KAMBING.ui.ac.id mirror aja koq repot?™". Below the header is a navigation bar with links: Terbaru, Mirror, ISO, Link Alternatif, Kontak, Status, Traffic, Pesan, CD/DVD, and Training Linux. The main content area includes sections for "Migrasi Domain", "Tentang Kambing", and "Mirror". The "Tentang Kambing" section describes KAMBING.ui.ac.id as a mirror of F/OSS. The "Mirror" section lists various mirrors such as Apache, BlankOn, Buku Sekolah Elektronik, Centos, CPAN, CRAN, and CTAN.

Gambar 2.5 Situs Repo Linux [kambing.ui.ac.id](http://kambing.ui.ac.id)

Tidak semua distro dan repo berisi 100% program/file free/open, atau tidak semua distro bebas digandakan karena ada distro yang berisi program/file tidak free/open. Contoh distro yang tidak bebas digandakan adalah Red Hat Enterprise Linux (RHEL). Software yang ada dalam DVD RHEL sebenarnya open source, tapi ada bagian dari hak atas kekayaan intelektual Red Hat seperti logo yang tidak bebas digandakan tanpa izin perusahaan Red Hat, Inc. Karena software yang ada dalam distro RHEL itu FOSS, maka pihak lain dapat mengembangkan turunan, hasil modifikasi RHEL, misal distro Linux CentOS.

Manajemen paket distro dan repo bisa berbeda antara distro satu dengan yang lain. Banyak distro Linux menggunakan *Advanced Packaging Tool* yang umumnya ditemukan di distro berbasis Debian, Ubuntu, dan BlankOn. Sebagain distro menggunakan yum yang biasa ditemukan di distro berbasis Red Hat, misalnya Fedora dan Igos Nusantara. Ada juga beberapa sistem manajemen paket independen, seperti Pacman, digunakan dalam Arch Linux dan equo, ditemukan di Sabayon Linux.

### **2.2.6 Ciri dan Spesifikasi Sistem Operasi Linux**

Linux memiliki ciri beda secara non teknis dibandingkan dengan sistem operasi proprietary seperti MS Windows dan Apple Mac OS X. Misal dari aspek hukum dan bisnis, sistem operasi Linux (GNU-Linux, Android, dll.) berlisensi FOSS, sehingga tidak seperti sistem operasi proprietary yang dijual dalam bentuk biaya izin penggunaan atau lisensi, tapi Linux dijual atau dibisniskan dalam bentuk lain.

Contoh bisnis Linux antara lain dukungan teknis (*technical support*) secara *off-line* (luring: luar ring) atau *on-line* (daring: dalam ring), modifikasi atau *customization*, migrasi, training, dll.

Linux dan Unix-like lainnya memiliki spesifikasi teknis multi user, multi tasking, dan networking, sejak tahun-tahun pertama dikembangkan. Ini berbeda dengan sistem operasi proprietary seperti MS Windows dan Apple Macintosh yang awalnya tidak multi user, sehingga dalam waktu bersamaan tidak bisa beberapa pengguna login dengan nama user (username) berbeda. Linux sejak 20 tahun lalu dapat diakses bersamaan (multi user) dan menjalankan banyak program secara bersamaan (multi tasking) sebagai komputer desktop maupun server (networking), sehingga Linux/Unix sejak dulu disebut *Network Operating System*, meskipun juga dapat difungsikan sebagai *Stand Alone Operating System*.

Istilah *Stand Alone* digunakan untuk sistem operasi bukan server, misal Windows 3x, 9x, XP, Vista, 7, dan 8. Artinya, hingga saat ini tidak semua versi Windows dan Mac dapat dijadikan sebagai server jaringan yang baik, misal Windows 7/8 dan Mac OS X. Jika ingin menggunakan sebagai server jaringan, Anda harus membeli Windows Server atau Mac server. Sedangkan umumnya distro Linux dapat digunakan sebagai stand alone (*workstation*) atau desktop maupun server. Sebagai contoh, Ubuntu Desktop dapat dijadikan server dengan menambahkan program-program untuk server, meskipun tersedia pula Ubuntu versi server yang belum memiliki sistem window atau desktop GUI. Ubuntu Server juga dapat diubah menjadi desktop dengan menambahkan program desktop, misal GNOME. Kalaupun ada Linux yang khusus dibuat untuk server, biasanya untuk tujuan server skala besar (enterprise), misal Red Hat Linux Enterprise (RHEL) dan SUSE Linux Enterprise Server (SLES).

### **2.2.7 Antaramuka Grafis atau GUI (Graphical User Interface)**

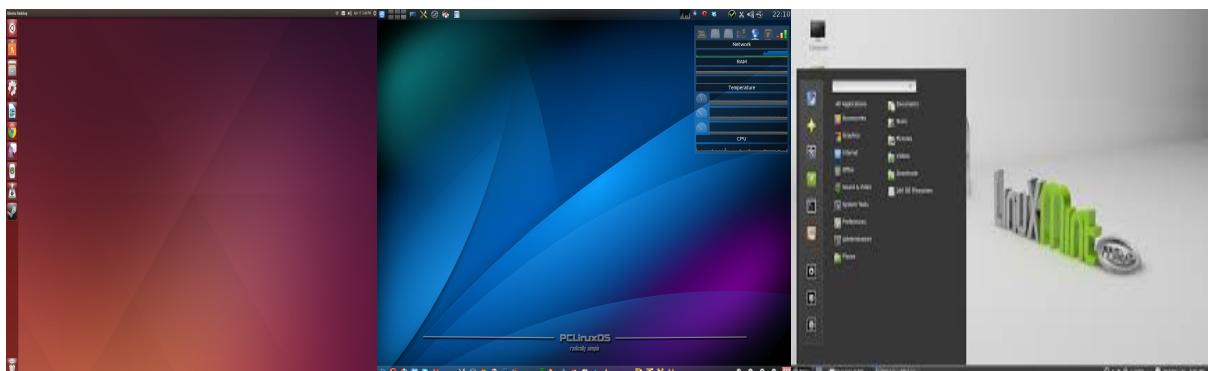
Antaramuka pengguna dalam modus grafis atau GUI (Graphical User Interface) adalah program-program yang menjadi penghubung pengguna dengan komputer dalam format grafis. Layar GUI disebut juga desktop (istilah lengkap: desktop environment). Dalam GUI

ada istilah Shell, yakni sebutan untuk software antaramuka yang menghubungkan pengguna dan kernel sistem operasi. Ada dua jenis Shell, grafis dan teks. Contoh Shell grafis atau GUI adalah GNOME Shell, GUI bawaan GNOME3. Linux Ubuntu menggunakan GNOME3, namun tidak menggunakan GNOME Shell sebagai bawaan. Shell GNOME3 diganti dengan Unity di Ubuntu.

Dalam sistem operasi Unix-like seperti Linux yang lengkap terdiri atas beberapa program terkait desktop ini, antara lain:

- *Windowing System* atau *X-Window*, misal X.Org, program dasar GUI yang berisi beberapa program bersifat *client-server* seperti program server X Window xserver-xorg dan pengatur kartu grafis atau driver VGA seperti xserver-xorg-video-intel.
- *Desktop Environment (DE)*, misal GNOME, KDE, LXDE, XFCE, yang berisi kumpulan program termasuk *Shell*, *Panel*, dan *Widgets* untuk mengatur ikon, panel, toolbar, window, menu, wallpaper, clock, calendar, messaging, dll. Sebuah produk lingkungan desktop dapat memiliki beberapa turunan/varian, misal GNOME3 Shell memiliki varian Unity (Distro Linux Ubuntu), Cinnamon (Linux Mint), dan Manokwari (Linux BlankOn).
- *Window Manager (WM)*, misal metacity, yang berfungsi mengatur tampilan jendela, termasuk dukungan terhadap 3D (misal compiz). Paket program window manager pada Linux atau UNIX sejenis dapat dipisahkan dari desktop. Misal GNOME dapat menggunakan window manager Metacity atau Enlightenment atau window manager lainnya. Sedangkan pada MS Windows dan Mac OSX, pengguna diberikan window manager yang telah menyatu dengan desktop, kecuali pengguna mengganti Shell bawaannya. Misal Shell MS Windows diganti Classic Shell dan KDE for Windows.
- *Desktop Manager (DM)*, misal gdm dan kdm, untuk mengatur jendela login ke desktop. Karena GUI Linux (X Window) itu termasuk server dan dapat diakses melalui jaringan, maka komputer lain (X Window client) dapat login ke komputer yang menjalankan desktop manager (X Window server) bila diizinkan oleh administrator.

Dalam sistem operasi GUI proprietary MS Windows, sistem X-Window, window manager, dan desktop environment sudah menjadi satu dengan sistem operasi. Demikian pula Mac OS X, iOS, dan Linux yang dibuat khusus untuk komputer bergerak seperti Android. Sedangkan pada umumnya distro GNU/Linux, sistem X-Window, window manager, dan desktop environment terpisah dari sistem operasi. Lingkungan desktop yang lengkap seperti GNOME, KDE, dan LXDE, telah dilengkapi window manager dan aplikasi sederhana. Misal untuk GNOME tersedia window manager **Metacity**, text editor **Gedit**, file browser (file manager) **Nautilus**, audio player **Audacious**, archive manager (pengekstrak/pengompres file) **File-roller**, image viewer **EOG**, document viewer (pdf reader) **Evince**, dan lain-lain.



Gambar 2.6 Desktop Unity (kiri), KDE (tengah), dan GNOME (kanan) (Wikipedia)

GUI di Linux tidak menyatu dengan sistem operasi, sehingga memudahkan pengguna memilih tampilan desktop sesuai seleranya. Bahkan pengguna dapat mengubah jenis desktop, dengan lebih dahulu menginstal paket-paket desktop lain melalui repositori. Misal desktop GNOME dapat diubah menjadi KDE atau LXDE, dan sebaliknya. Tampilan GUI (shell atau panel) GNOME juga dapat diubah dengan tetap menggunakan dasar program GNOME 3 (gtk3), misal dari Unity diubah menjadi GNOME Classic, GNOME Shell, GNOME Manokwari, atau “OS X-like” Cairo-Dock, dsb. Desktop GNOME (Shell, Unity, Manokwari, dll.) dapat diubah menjadi seperti OS X dengan perintah “sudo apt-get install cairo-dock”. Jika sukses menginstal cairo-dock dan program lain yang dibutuhkannya, logout dan login kembali dengan memilih jenis desktop Cairo-Dock.

### **2.2.8 Antarmuka Teks atau *Command Line Interface***

Antarmuka pengguna dalam modus teks atau TUI (Text User Interface), yang disebut juga CLI (*Command Line Interface*). Dalam modus teks, kernel Linux dan tool atau utilitasnya telah dapat digunakan sebagai sistem operasi, meskipun belum memiliki antarmuka grafis atau GUI. Salah satu kumpulan perintah yang dijalankan pada modus teks dinamakan BASH (Bourne Again SHell). Di Linux/Unix-like, BASH termasuk aplikasi. Karena perintah-perintah shell ini juga dapat digunakan untuk membuat program atau dimasukkan dalam program, maka ada ilmu “Pemrograman Shell.”

Kelebihan TUI atau CLI dibandingkan GUI adalah cepat dan ringan. Kekurangan TUI/CLI adalah tampilan yang kurang menarik bagi umumnya pengguna komputer, dan tidak mudah digunakan untuk pengolahan gambar, misal mengedit foto atau video. Sebaliknya, kelebihan GUI adalah kemudahan digunakan untuk komputer desktop, dan kelemahannya adalah memakan banyak memori dan tugas prosesor, sehingga lebih lambat daripada modus teks saja.

Program dasar untuk menjalankan perintah teks disebut *virtual console*. Pada GUI Linux, program untuk menjalankan perintah ini disebut terminal emulator. Misal, desktop GNOME dilengkapi program console bernama gnome-terminal, KDE dilengkapi program console bernama konsole, LXDE dengan lx-terminal, dsb.

```
[root@esams ~]# pwd
/home/mars
mars$ mainmain `` cd /usr/portage/app-shells/bash
mars$ mainmain /usr/portage/app-shells/bash $ ls -al
total 0
drwxr-xr-x 3 portage portage 1924 Jul 25 19:06 .
drwxr-xr-x 3 portage portage 1924 Aug 7 22:39 ..
-rw-r--r-- 1 root root 20902 Jul 25 19:06 ChangeLog
-rw-r--r-- 1 root root 27082 Jul 25 19:06 Manifest
-rw-r--r-- 1 portage portage 4045 Mar 23 21:37 bash-3.1_p17.ebuild
-rw-r--r-- 1 portage portage 5977 Mar 23 21:37 bash-3.2_p39.ebuild
-rw-r--r-- 1 portage portage 5988 Mar 23 21:37 bash-3.2_p40.ebuild
-rw-r--r-- 1 portage portage 5643 Apr 5 14:37 bash-4.0_p10+1.ebuild
-rw-r--r-- 1 portage portage 6238 Apr 5 14:37 bash-4.0_p10.ebuild
-rw-r--r-- 1 portage portage 14056 Apr 5 14:37 bash-4.0_p10.ebuild
-rw-r--r-- 1 portage portage 5532 Apr 8 18:21 bash-4.0_p17.ebuild
-rw-r--r-- 1 portage portage 5668 May 30 03:35 bash-4.0_p24.ebuild
-rw-r--r-- 1 portage portage 5668 May 30 03:35 bash-4.0_p28.ebuild
drwxr-xr-x 2 portage portage 2048 May 30 03:35 files
-rw-r--r-- 1 portage portage 466 Feb 9 04:35 metadata.xml
mars$ mainmain `` cd /usr/portage/app-shells/bash $ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmeta SYSTEM "http://www.centoo.org/dtd/metadata.dtd">
<pkgmeta>
    <herd>base-system</herd>
    <use/>
    <flag name="bashlogger">Log ALL commands typed into bash; should ONLY be
        used in restricted environments such as hotpots</flag>
    <flag name="bluetooth">Add support for loading bluetooth at runtime</flag>
    <flag name="plugins">Add support for loading builtins at runtime via
        enable</flag>
    <use/>
</pkgmeta>
mars$ mainmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
Password:
[...]
bluetooth is started
mars$ mainmain /usr/portage/app-shells/bash $ ping -q -c1 en.wikipedia.org
PING en.esams.wikimedia.org (91.199.174.2) 56(84) bytes of data.
[...]
--- rr.esams.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49.820/49.820/49.820/0.000 ms
mars$ mainmain /usr/portage/app-shells/bash $ grep -l /dev/sda /etc/fstab | cut --fields=-3
/dev/sda1          /boot
/dev/sda2          none
/dev/sda3          none
mars$ mainmain /usr/portage/app-shells/bash $ date
Sat Aug  8 02:42:24 MSK 2009
mars$ mainmain /usr/portage/app-shells/bash $ lsmod
Module      Size  Used by
rndis_wlan   23424  0
rndis_host   8696  1 rndis_wlan
cdc_ether   55574  2 rndis_wlan
ustorage   18688  3 rndis_wlan, rndis_host, cdc_ether
portpoint_pc 36424  0
fglrx     2368128  28
portpoint   35224  1 portpoint_pc
ITCO_wdt   12272  0
i2c_i801   9308  0
mars$ mainmain /usr/portage/app-shells/bash $ █
```

*Gambar 2.6 Contoh penggunaan perintah (Command) BASH (Wikipedia)*

Console di sistem operasi MS Windows disebut *Command Prompt*. Berbeda dengan command prompt Windows yang terbatas fungsinya, hampir semua program untuk GNU/Linux dapat dijalankan melalui console atau terminal. Hampir semua program GUI di Linux sebenarnya dapat dijalankan dengan perintah di console/terminal, namun biasanya sudah tersedia menu untuk menjalankan program GUI yang perintahnya tidak dilihat oleh pengguna. Misal untuk mulai menjalankan browser web GUI Firefox, pengguna dapat mengeksekusinya melalui console/terminal dengan mengetikkan firefox diikuti menekan Enter.

Kelebihan Shell CLI dibandingkan Shell GUI adalah cepat dan ringan. Kekurangan Shell CLI adalah tampilan yang kurang menarik di mata pengguna biasa.

Contoh perintah (command) BASH yang penting difahami pengguna Linux/Unix: clear, ls, cd, pwd, rm, mv, mkdir, rmdir, cat, less, more, chmod, chown, passwd, grep, tail, du, df, free, dll. Berikut ini penjelasan beberapa command dan utilitas di Linux yang mudah dan cepat digunakan untuk pekerjaan sehari-hari:

- ls, perintah untuk melihat isi direktori.
- cp, perintah untuk mengopi file atau direktori.
- mv, perintah untuk mengganti nama atau memindahkan file/direktori.
- rm, perintah untuk menghapus file/direktori.
- mkdir, perintah untuk membuat direktori.
- rmdir, perintah untuk menghapus direktori yang kosong.
- du, perintah untuk melihat besar ruang direktori yang telah digunakan.
- df, perintah untuk melihat besar ukuran partisi hard disk dan ruang yang tersisa.
- free, melihat penggunaan memori nyata (RAM) dan memori virtual (SWAP).
- mount, melihat daftar sistem file dan jenisnya.
- less, perintah untuk melihat isi file teks per halaman teks.
- cat, perintah untuk menampilkan isi file teks ke layar secara keseluruhan atau memindahkan data dari input standar, misal file teks, ke output standar, misal layar.
- who, perintah untuk melihat siapa saja yang sedang login di komputer.
- su, perintah untuk berganti user.
- sudo, perintah untuk menjalankan perintah sebagai super user.
- ping, perintah untuk mengecek sambungan jaringan.
- ps ux, melihat daftar semua program yang digunakan dan nomor prosesnya (PID).
- top, melihat semua program yang sedang berjalan, termasuk pemakaian prosesor, RAM, dll.
- exit atau Ctrl+D, keluar atau logout dari user yang sedang digunakan.

## Bab 3 Contoh Produk Open Source selain Sistem Operasi

Tujuan Umum: Mampu memahami fungsi dasar beberapa aplikasi FOSS dan implementasinya di dunia kerja dan industri.

Tujuan Khusus:

- a) Mampu menjelaskan beberapa manfaat aplikasi desktop dan mobile untuk internet, perkantoran, multimedia, dll.
- b) Mampu menjelaskan beberapa contoh fungsi aplikasi FOSS sebagai infrastruktur bisnis/industri dan middleware.
- c) Mampu menjelaskan beberapa contoh fungsi aplikasi FOSS sebagai SIM.

### 3.1 Pendahuluan

Sebagai garis besar pengelompokan jenis software, Martin Fink membuat tabel sebagai berikut (*Open Source – A Multidisciplinary Approach, halaman 96*).

*Tabel 3.1 Daftar kategori, tipe software, dan contoh-contohnya*

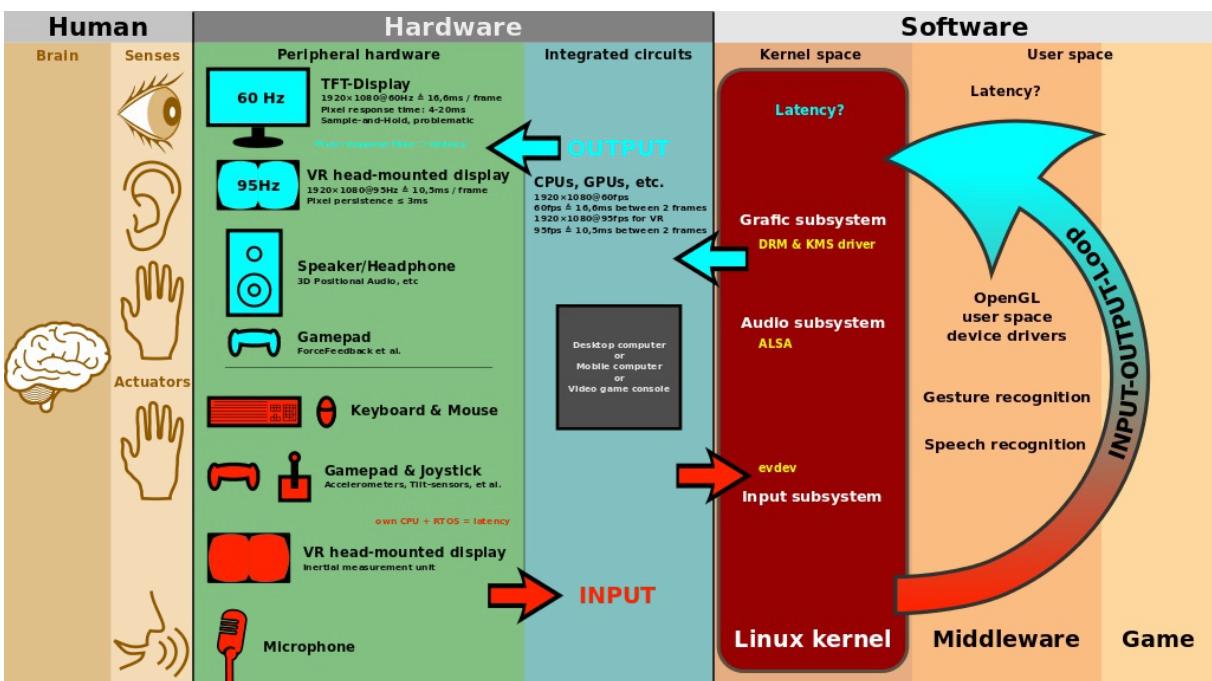
Kategori	Tipe Software	Contoh
Software Sistem	Sistem operasi	Linux dan FreeBSD
Middleware	Software server email	Sendmail, Postfix, Qmail
	Software server web	Apache
	Software server aplikasi	JBoss, Zope
	Sistem Manajemen Basis Data	MySQL, PostgreSQL
	Protokol	Samba, OpenLDAP, OpenSSH
Software Aplikasi	Office	LibreOffice, OpenOffice
	Lingkungan desktop	GNOME, KDE
	Browser web	Mozilla Firefox
	Sistem manajemen	Compiere
	Software manajemen projek	Achievo
	Groupware	Phprojet
Bahasa Pemrograman		C/C++, Perl, PHP, Python, Java

### 3.2 Middleware (Server Aplikasi, Web, Database, Infrastruktur)

Di atas sistem operasi dapat berjalan berbagai jenis program yang disebut aplikasi (dalam

pengertian umum). Ada kelompok aplikasi disebut middleware. Menurut ObjectWeb.org, middleware adalah program yang berada di antara sistem operasi dan aplikasi pada tiap sisi sistem komputasi terdistribusi dalam jaringan. Wikipedia.org mendefinisikan middleware sebagai software yang memberi layanan (yang tidak disediakan sistem operasi) untuk menjalankan aplikasi.

Middleware dapat berupa software tunggal seperti server web Apache, *Content Management System* (CMS) Joomla, dan server aplikasi JBoss. Middleware dapat pula berupa gabungan software seperti server aplikasi JBoss (WildFly), berisi server web Apache dan bahasa pemrograman Java. Java pada Apache dapat disambungkan dengan server database MySQL oleh JBoss. Artinya, JBoss dapat digunakan untuk mengembangkan aplikasi web berbasis Java di atas sistem operasi Linux atau sistem operasi lain dengan server web Apache dan software pengembangan Java.



Gambar 3.1 Middleware berada di antara kernel Linux dan aplikasi game (Wikipedia)

Contoh lain middleware, pada sistem smartphone atau komputer tablet Linux Android terdapat middleware berisi antara lain Dalvik Virtual Machine, library aplikasi Java, dan framework, sehingga aplikasi Android dapat dibuat dengan bahasa pemrograman Java.

Contoh lain lagi middleware adalah mesin game (*game engine*), misalnya Doom (Free Doom) untuk menjalankan aplikasi permainan, misalnya vavoom, prboom, freedm, yaitu game 3d FPS (First Person Shooter).

Beberapa contoh bagian dari software server jaringan atau infrastruktur, server database, dan middleware open source berikut ini ada yang telah disatukan dalam kemasan CD/DVD distro sistem operasi, misalnya Linux Ubuntu Server, dan ada yang disimpan dalam server repo atau DVD repo.

- **Web Server:** Apache, server web yang sangat terkenal dan paling banyak digunakan di seluruh dunia (menurut netcraft.com). Berbagai software middleware dikembangkan oleh Yayasan Software Apache, antara lain Axis2, Cocoon, Geronimo, dan Jakarta Tomcat.

- *Database Server*: MySQL, server database yang paling banyak digunakan di internet (didukung Oracle), dan PostgreSQL yang memiliki penyedia dukungan berbentuk software tambahan (yakni EnterpriseDB). Middleware seperti Apache menjadi penghubung aplikasi dengan server database, atau server database satu dengan server database yang lain.
- *Email Server*: Sendmail, Qmail, Postfix (yang juga dimodifikasi sebagai aplikasi webmail lengkap untuk kolaborasi, Zimbra), dll.
- *Application Server*: GlassFish, JBoss, Talend, Zope, dll.
- *Networking / Infrastructure Server*: BIND Domain Name Server, Squid proxy server, Firewall, dll.

### 3.3 Software Aplikasi

Aplikasi dalam istilah ini adalah program yang dibuat untuk digunakan sehari-hari. Aplikasi dapat dijalankan langsung pengguna di atas sistem operasi, di atas middleware, di atas desktop, atau di atas aplikasi lain. Istilah yang serupa dengan ini adalah app (singkatan dari application), program, dan aplikasi software (*software application*). Istilah app menjadi sangat populer sejak era komputer bergerak (*mobile*), terutama sejak Google merilis Android dengan kumpulan aplikasi dalam gudang program atau repositori yang dinamakan Play Store. Sebelum era komputer bergerak, di bidang ilmu komputer telah dikenal yang namanya aplikasi perkantoran, aplikasi internet, aplikasi multimedia, dan lain-lain. Berikut ini contoh aplikasi yang terdapat dalam DVD Linux Ubuntu dan repositori-nya.

- *Desktop Environment*: GNOME, KDE, LXDE, XFCE, dll.
- *File Browser*: Nautilus, dll.
- *Office Suite*: LibreOffice, OpenOffice, dll.
- *Web Browser*: Mozilla, Firefox, Chromium, dll.
- *PDF/Document Viewer*: Evince, dll.
- *Graphics Editor*: Gimp, Inkscape, dll.
- *3D Design/Animation*: Blender
- *Desktop Publishing*: Scribus
- *Audio-Video Player*: Audacious, Rhythmbox, Mplayer, Totem, VLC, dll.
- *Audio-Video Editor*: Audacity, Avidemux, OpenShot, Pitivi, dll.
- *Finance/Accounting*: GnuCash, KMyMoney, dll.
- *Enterprise (Information System)*: FrontAccounting, Compiere, openERP (*Enterprise Resource Planning*), Orange HRIS (*Human Resource IS*), Sugar CRM (*Customer Relationship Management*), dll.
- *Project Management*: Planner, DotProject, dll.
- *Content Management System*: Drupal, Joomla, Wordpress, Moodle (e-learning), dll.

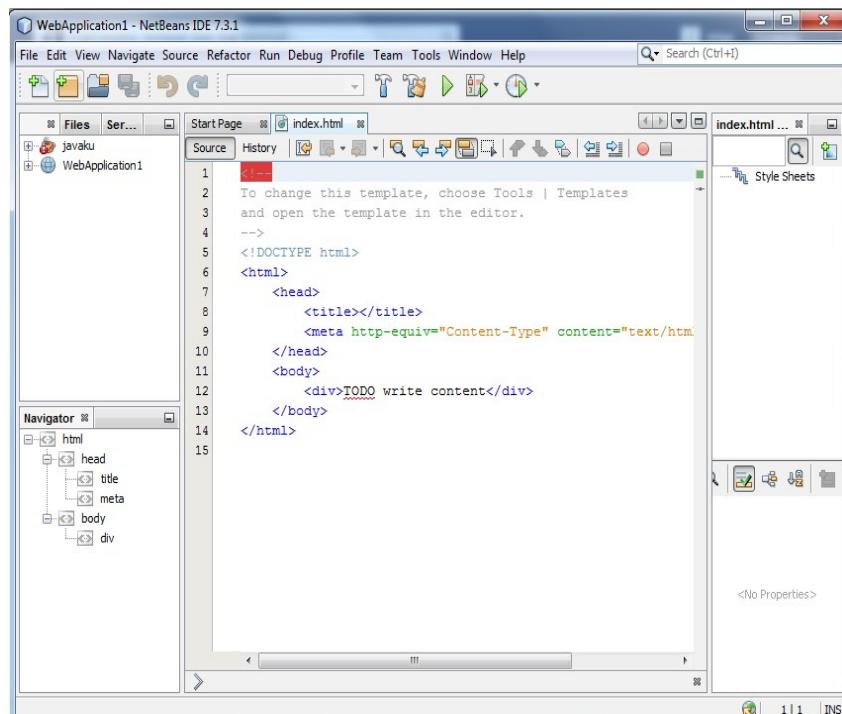
### 3.4 Software Pemrograman

Kelompok program yang tidak termasuk sistem operasi, middleware, dan aplikasi adalah bahasa pemrograman, yakni program untuk menghasilkan program lain. Berikut ini contoh bahasa pemrograman dan aplikasi bantu pembuatan program yang terdapat dalam DVD distro

Linux Ubuntu dan repositorinya.

- C/C++ dengan GCC (GNU Compiler Collection) untuk C/C++ dan lain-lain, Pascal dengan Free Pascal, Java dengan JDK (Java Development Kit) atau yang sejenis, Perl, PHP, Python, Basic, Fortran, dll.
- Beberapa aplikasi untuk memudahkan pembuatan program disebut IDE (*Integrated Development Environment*), misal Lazarus sebagai IDE untuk Free Pascal, Gambas untuk Basic, dan Eclipse untuk Java/Android, C/C++, JavaScript, Python, PHP, dll.

Bahasa pemrograman open source dan aplikasi IDE seperti contoh-contoh di atas dapat digunakan untuk membuat program lain. Lisensi program yang dihasilkan dengan bahasa pemrograman open source tidak harus juga open source. Lisensi program yang dibuat pemrogram (programer) dengan mengambil sebagian atau seluruh program lain bergantung pada lisensi kode sumber program lain yang digunakan sebagai acuan tersebut. Dengan kata lain, jika programer membuat program tidak dari nol, misalnya mengembangkan program yang sudah ada, maka lisensi program harus mengikuti pedoman pada lisensi yang dibuat oleh pemrogram sebelumnya.



Gambar 3.2 Membuat Aplikasi dengan IDE Netbeans di Windows

Sebagai contoh, seorang programer menggunakan kompiler bahasa C (gcc) untuk memodifikasi kernel Linux yang berlisensi GPL. Hasil modifikasi kernel Linux yang disebarluaskan harus juga GPL, kecuali kernel hasil modifikasi itu digunakan sendiri (tidak disebarluaskan). Tapi jika gcc yang berlisensi GPL itu digunakan untuk menghasilkan program baru yang dibuat dari nol, maka lisensi program baru itu diserahkan sepenuhnya kepada pembuat program.

### 3.5 Aplikasi Enterprise

Aplikasi disebut enterprise jika ditujukan untuk organisasi, misal perusahaan atau pemerintahan, dengan jumlah pengguna lebih dari satu, dan ada sistem database. Aplikasi

yang ditujukan untuk digunakan bekerja sendiri, misalnya word processor biasa, tidak disebut enterprise. Berikut ini beberapa contoh aplikasi enterprise berbentuk sistem informasi manajemen.

- HRM (Human Resource Management): OrangeHRM, IceHRM, dll.
- CRM (Customer Relationship Management): SugarCRM, EPESI CRM, dll.
- ERP (Enterprise Resource Planning) OpenERP (Odoo), Compiere, Adempiere, OpenBravo, dll.
- CMS (Content Management System): Joomla, Drupal, Alfresco, dll.



Gambar 3.3 Tampilan login aplikasi enterprise OrangeHRM

### 3.6 Web Application & Native Application

*Web App:* aplikasi yang menggunakan program berbasis web (html, css, javascript, php, dll.) dan dijalankan dengan web client (web browser). Aplikasi web dipasang di server web dalam komputer jaringan atau komputer lokal atau tempat penyimpanan lain. Contoh aplikasi perkantoran web: FengOffice (open source) dan Google Drive (drive.google.com). Contoh aplikasi email web: SquirrelMail (open source) dan Google Mail (gmail.com).

*Native App:* aplikasi yang dijalankan secara alami tanpa bantuan web browser. Istilah Native biasanya untuk mobile, misal aplikasi web browser pada Android, Firefox, dll. Contoh aplikasi native pada komputer desktop: LibreOffice dan Thunderbird (email client), dll.

### 3.7 Catatan tentang Definisi Kelompok Software

Definisi software sistem, middleware, dan aplikasi memiliki banyak variasi. Definisi umum aplikasi mencakup middleware, sehingga software Apache Cocoon dapat dimasukkan sebagai aplikasi maupun middleware. Beberapa software sistem juga dapat disebut sistem maupun aplikasi, misal software sistem manajemen konten (*Content Management System*) Joomla. Jika dilihat dari namanya, Joomla tergolong software **sistem** dalam pengertian luas, namun jika dilihat dari fungsinya, Joomla termasuk **aplikasi** untuk mengelola konten web. Joomla berfungsi menghubungkan aplikasi web dengan server database, sehingga Joomla dapat disebut **middleware**.

## Bab 4 Konsep Pengembangan Open Source

Tujuan Umum: Mampu memahami konsep yang mendasari pengembangan Open Source.

Tujuan Khusus:

- a) Mampu menjelaskan konsep sederhana yang diyakini dan dijalankan para pengembang Open Source.
- b) Mampu memberikan contoh-contoh organisasi pengembang Open Source dan karyanya.

### 4.1 Model Pengembangan Open Source “Bazaar”

#### 4.1.1 The Cathedral & The Bazaar

Konsep dasar atau model pengembangan FOSS (Free/Open Source Software) itu unik, dan menjadi model yang sukses digunakan karena muncul pada waktu yang bersamaan dengan berkembangnya internet dan efek internet yang luar biasa di bidang komunikasi. Analogi Katedral dan Bazar digunakan untuk membedakan model pengembangan FOSS dengan metode pengembangan perangkat lunak tradisional. Perumusan dua metode ini pertama kali dipresentasikan oleh pendiri Open Source Initiative **Eric S. Raymond** pada 1997, yang dipublikasikan dalam buku berjudul “The Cathedral & The Bazaar” pada 1999.

Pengembangan perangkat lunak tradisional diibaratkan dengan cara sebuah katedral dibangun pada masa lalu. Kelompok kecil tukang batu secara hati-hati merencanakan sebuah desain dalam tempat yang terisolasi, dan segala sesuatunya dibuat dalam sebuah usaha tunggal. Sekali katedral berhasil dibangun, maka dianggap selesai, dan hanya sedikit dilakukan perubahan lanjutan.



Gambar 4.1 Pencetus gerakan Open Source Eric S. Raymond (Wikipedia)

Perangkat lunak secara tradisional dibuat dengan gaya yang serupa itu. Sekelompok pemrogram bekerja dalam suatu isolasi (misalnya di sebuah perusahaan), dengan perencanaan dan manajemen yang hati-hati hingga selesai dan program dirilis ke publik. Sekali dirilis,

program dianggap selesai, dan selanjutnya hanya ada pekerjaan terbatas untuk program itu. Model katedral umumnya untuk software proprietary, namun Eric S. Raymond juga menyebut free software **GCC** (GNU C Compiler) dan GNU **Emacs** pada awal pengembangannya menggunakan cara katedral (en.wikipedia.org/wiki/The\_Cathedral\_and\_the\_Bazaar). GCC dan Emacs dikembangkan oleh pendiri Free Software Foundation **Richard M. Stallman** dan kawan-kawan sebagai bagian dari proyek GNU (GNU is Not Unix), yang sekarang menggunakan konsep bazar pula.

Umumnya pengembangan FOSS sekarang lebih mirip dengan sebuah bazar, yang tumbuh secara organis. Dalam sebuah bazar, pedagang awal datang, membangun struktur, dan memulai bisnis. Pedagang-pedagang berikutnya datang dan membangun strukturnya masing-masing. Perkembangan bazar nampak menjadi gaya yang tidak teratur. Pada dasarnya para pedagang diarahkan untuk membangun struktur minimal yang dapat berfungsi sehingga mereka bisa memulai berjualan. Tambahan dibuat sesuai kebutuhan dan keadaaan selanjutnya.

Dengan model serupa, pengembangan FOSS dimulai dari yang tidak terstruktur. Pengembang merilis kode programnya ke publik meskipun baru berfungsi secara minimal, dan kemudian mengubah programnya sesuai umpan balik yang diberikan pengguna atau publik. Pengembang lain bisa ikut mengembangkan program itu berdasar kode-kode yang telah ada. Pada periode waktu tertentu, keseluruhan sistem operasi dan aplikasi menjadi tumbuh dan berkembang secara terus menerus. Eric S. Raymond memberi contoh model bazar digunakan oleh **Linus B. Torvalds** dan kawan-kawan dalam pengembangan kernel **Linux** dan Eric dan kawan-kawan dalam pengembangan **Fetchmail** – program teks atau CLI (Command Line Interface) untuk mengambil atau mengunduh email. Konsep bazar menjadi sangat populer. Bahkan ada produk untuk menyimpan dan mengembangkan kode program secara open source diberi nama Bazaar (bzr).

#### 4.1.2 Kelebihan Konsep Bazar

Konsep pengembangan bazar telah dijadikan pilihan banyak pengembang FOSS untuk mendapatkan beberapa kelebihan berikut ini:

- **Mengurangi duplikasi kerja:** Dengan merilis program lebih cepat, dan memberikan izin kepada pengguna untuk megubah dan meredistribusi kode sumber, kemudian pengembang FOSS memanfaatkan kembali karya yang dihasilkan oleh teman pengembang yang lain. Skala ekonomi pengembangan open source dapat menjadi sangat besar. Sebagai gambaran, lebih baik memanfaatkan peluang potensi 50 pengembang dari 10 perusahaan untuk melakukan pengembangan suatu produk secara bersama (membentuk kerja sama atau melakukan kombinasi), daripada 50 pengembang software pada sepuluh perusahaan mengembangkan aplikasi jaringan yang sama secara terpisah. Mengurangi duplikasi kerja akan membuat skala pengembangan FOSS menjadi besar, karena ribuan pengembang di seluruh dunia dapat bekerja sama.
- **Membangun di atas karya lain:** Dengan ketersediaan kode sumber untuk membangun program, waktu pengembangan menjadi pendek. Banyak projek FOSS berbasis kode program yang dihasilkan projek lain untuk menambah fungsionalitas yang diperlukan. Sebagai contoh, projek *server* web Apache lebih memilih memanfaatkan projek OpenSSL daripada menulis sendiri kode kriptografi, sehingga menghemat jutaan jam untuk pembuatan program dan pengujiannya. Bahkan jika kode sumber tidak dapat secara langsung digabungkan, ketersediaan kode sumber memudahkan pengembang untuk belajar bagaimana projek lain telah dilakukan untuk

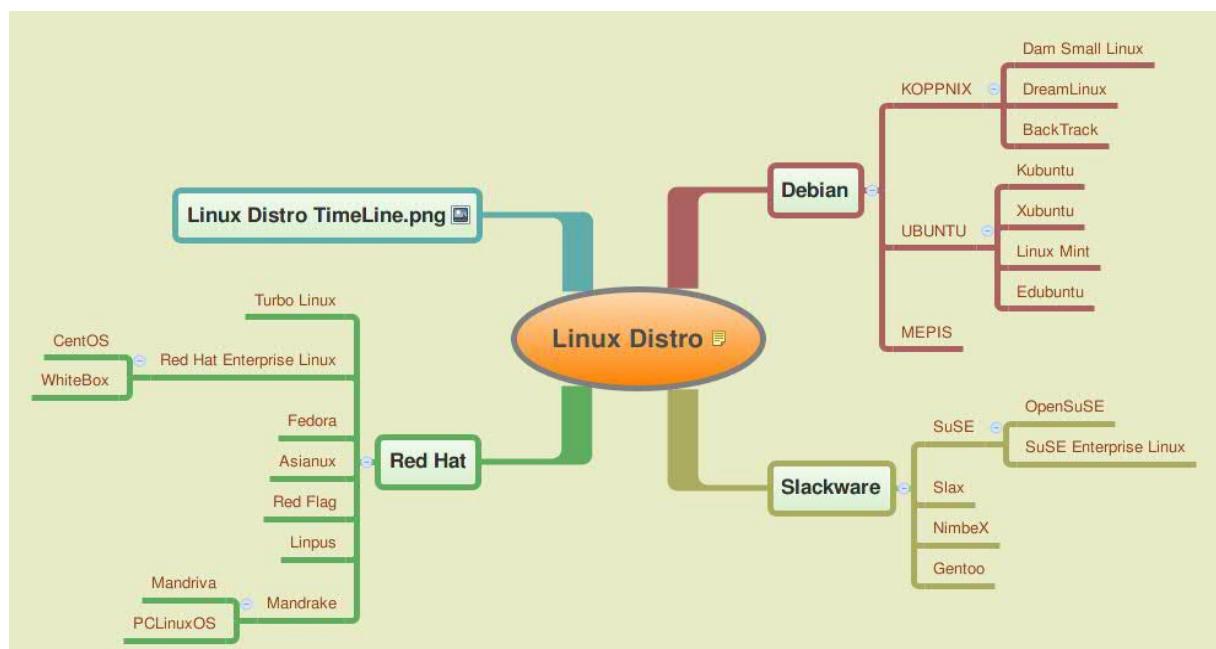
memecahkan masalah yang sama.

- **Kendali mutu yang lebih baik:** Semakin banyak orang menggunakan dan mengevaluasi kode sumber, maka kesalahan yang ada akan mudah ditemukan dan diperbaiki secara cepat. Aplikasi *proprietary* bisa saja menerima laporan kesalahan, tetapi karena pengguna tidak dapat mengakses kode sumber, maka pengguna hanya bisa sebatas melaporkan. Pengembang FOSS sering menemukan bahwa pengguna yang memiliki akses ke kode sumber tidak hanya bisa melaporkan kesalahan, namun juga menjelaskan lebih tajam apa penyebabnya. Bahkan dalam beberapa kasus pengguna dapat mengirimkan kode perbaikannya. Ini sangat mengurangi waktu pengembangan dan kontrol terhadap kualitas.
- **Mengurangi biaya perawatan:** Biaya perawatan software sering sama atau lebih besar dari biaya pengembangan awal. Jika sebuah perusahaan merawat software sendirian, maka pekerjaan itu menjadi sangat mahal. Dengan menggunakan model pengembangan FOSS, biaya perawatan dapat dibagi di antara ribuan pengguna potensial, sehingga mengurangi biaya perawatan per orang atau organisasi. Demikian pula peningkatan kemampuan software dapat dilakukan oleh banyak organisasi atau individu, yang hasilnya akan lebih efisien dalam menggunakan sumber daya.

## 4.2 Studi Kasus Pengembangan Distro Linux

### 4.2.1 Sejarah Distro Linux

Linus Torvalds mengembangkan kernel Linux dan didistribusikan versi pertama yaitu versi 0.01 pada 1991. Saat pertama kali didistribusikan hanya berupa source code kernel saja, kemudian menjadi dapat diunduh dalam bentuk *image* disket atau cakram flopi (bahasa Inggris: *floppy disk*), satu disket *bootable* yang berisi kernel, dan satu set utilitas GNU dan program untuk menyiapkan sistem file.



Gambar 4.2 Main Mapping Distro Linux ([www.xmind.net](http://www.xmind.net))

Karena prosedur instalasi yang rumit, terutama dalam menghadapi pertumbuhan jumlah perangkat lunak yang tersedia, distribusi lain mulai bermunculan untuk menyederhanakan masalah instalasi ini. Berikut ini beberapa distribusi Linux awal:

- H. J. Lu's "Boot-root", sebuah *dist images* yang terdiri atas kernel dan alat minimal untuk memulai menjalankan komputer.
- MCC Interim Linux, yang dapat diunduh untuk umum pada server FTP *University of Manchester* pada Februari 1992.
- TAMU, dibuat di *Texas A&M University* pada waktu yang hampir bersamaan dengan MCC Interim Linux.
- SLS (*Softlanding Linux System*). SLS tidak terpelihara dengan baik, sehingga Patrick Volkerding merilis distribusi baru berdasarkan SLS bernama Slackware, yang pertama dirilis pada 1993. Slackware merupakan distribusi tertua yang masih ada hingga saat ini.
- Yggdrasil Linux/GNU/X atau LGX, distribusi Linux pertama berbasis CD-ROM, yang dirilis pada 1993.
- Red Hat Linux, yang dirintis mulai 1993 dan dirilis pertama pada 1994. Pada 11 Agustus 1999 Red Hat Inc *go-public* dan kemudian menjadi distro Linux besar hingga sekarang.

#### 4.2.2 Perbandingan Distro Linux

Distribusi-distribusi Linux dapat dibedakan berdasarkan antara lain sebagai berikut:

- Manajemen paket program, misal distro dengan paket berbasis RPM RedHat Packace Manager, berbasis DEB Debian, atau berbasis TXZ Slackware.
- Bebas atau tidak bebas diunduh: misal CentOS bebas, RedHat Enterprise Linux tidak bebas.
- Tujuan pembuatan, misal Ubuntu untuk desktop dan ClearOS untuk server jaringan.
- Lingkungan desktop yang digunakan, Ubuntu dengan Gnome, Kubuntu dengan KDE.
- Perangkat lunak dasar yang disertakan dalam CD/DVD, Ubuntu tanpa GIMP dan Inkscape, BlankOn dengan GIMP dan Inkscape untuk mengedit foto dan menggambar.
- Ada juga perbedaan dari program instalasi awal dan program admininstrasi sistem (system settings), misal Linux Mint menggunakan program administrasi sistem beda dengan Ubuntu.
- Target negara pengguna, misal BlankOn untuk Indonesia, dan UbuntuKylin untuk China.
- Target agama pengguna, misal Sably dan BlankOn Sajadah untuk muslim, Ubuntu Christian Edition.

Berikut ini daftar sebagian distribusi Linux yang populer:

- Arch Linux, merupakan distribusi yang ditargetkan untuk pengguna Linux yang sudah berpengalaman. Arch Linux dikelola oleh komunitas.
- Manjaro, adalah distribusi turunan Arch Linux yang cepat, ramah pengguna awam, dan berorientasi desktop
- Debian, distribusi ini dikelola oleh komunitas. Debian merupakan distribusi tua yang masih ada hingga saat ini dan banyak menghasilkan turunan distribusi besar 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 forensic dan uji keamanan komputer.
- Linux Mint (Ubuntu Edition), turunan Ubuntu yang popularitasnya hampir menyamai Ubuntu.
- Lubuntu (varian Ubuntu dengan desktop LXDE).
- Xubuntu (varian Ubuntu dengan desktop XFCE).
- Kubuntu (varian Ubuntu dengan desktop KDE).
- BlackBuntu (varian Ubuntu khusus untuk penetrasi jaringan, keamanan, dan *digital forensic*).
- BlankOn, distribusi Linux yang dikembangkan oleh para pengembang Open Source Indonesia. BlankOn menggunakan antarmuka desktop yang disebut Manokrawi dan tersedia dalam beberapa bahasa daerah lokal di Indonesia. BlankOn 1.x berbasis Fedora, 2.x hingga 7.x berbasis Ubuntu, dan 8.x hingga sekarang (9.x) berbasis Debian.
- Fedora, distribusi komunitas yang disponsori oleh perusahaan Amerika, Red Hat.
- Red Hat Enterprise Linux, distribusi yang dikelola dan dikomersialkan oleh Red Hat.
- CentOS, distribusi komunitas yang dibangun dari kode sumber Red Hat Enterprise Linux.
- Oracle Enterprise Linux, merupakan turunan dari Red Hat Enterprise Linux, yang dipelihara dan komersialnya didukung oleh Oracle.
- IGOS Nusantara, dikembangkan oleh Puslit Informatika Lembaga Ilmu Pengetahuan Indonesia, berdasarkan distro Fedora.
- 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, dirilis pertama pada 2010.
- PCLinuxOS, turunan dari Mandriva yang dikhawasukan untuk komputer desktop.
- Slackware, merupakan distribusi besar pertama, yang lahir pada 1993.
- SUSE Linux Enterprise, dibangun berdasarkan OpenSUSE dan dikelola oleh perusahaan.
- OpenSUSE, distribusi komunitas yang disponsori oleh perusahaan pengembang SUSE.

### 4.3 Studi Kasus Pengembangan Linux BlankOn

Linux BlankOn adalah distribusi Linux yang dikembangkan oleh Yayasan Penggerak Linux Indonesia (YPLI) dan Tim Pengembang BlankOn. Distribusi ini dirancang sesuai dengan kebutuhan pengguna komputer umum di Indonesia. Saat tulisan ini disusun, versi rilis (final) adalah 8.0 yang dirilis pada 17 Agustus 2012 dengan nama kode Rote, dan versi berikutnya adalah 9.0 dengan nama kode Suroboyo yang akan dirilis pada 10 November 2013, bertepatan dengan Hari Pahlawan.

Linux BlankOn dikembangkan secara terbuka dan bersama-sama untuk menghasilkan distro Linux khas Indonesia untuk berbagai kebutuhan di pendidikan, perkantoran, pemerintahan, dan personal. Linux BlankOn merupakan bagian dari Proyek BlankOn yang memiliki

cakupan lebih luas dari sekedar distro. Proyek BlankOn bertujuan untuk membangun ekosistem pengembangan produk dan kompetensi SDM Indonesia dalam konteks perangkat lunak bebas dan terbuka.

Pengembangan BlankOn menggunakan model bazar, dengan memanfaatkan jaringan internet, sistem operasi Linux sebagai server web, dan beberapa aplikasi khusus pengembangan software seperti Git/Bzr untuk berbagi kode program, Trac untuk manajemen proyek dan penelusuran bug, Wiki untuk membuat dokumentasi, SSH untuk login ke server, IRC untuk diskusi, dan lain-lain.

#### 4.3.1 Sejarah BlankOn

BlankOn versi pertama (1.0) dikembangkan sejak September 2004 dan dirilis pada Februari 2005, dengan sponsor UNESCO, organisasi badan dunia PBB di bidang pendidikan, ilmu pengetahuan, kebudayaan, dan komunikasi. BlankOn pertama ini memiliki nama kode Bianglala, sehingga disebut juga BlankOn 1.0 Bianglala. BlankOn 1.0 dikembangkan berdasarkan distro Linux Fedora. Fedora adalah distro yang disponsori Red Hat Inc setelah perusahaan tersebut tidak lagi mengeluarkan Linux Red Hat versi “gratis” atau bebas di-download.



Gambar 4.3 Logo BlankOn ([blankonlinux.or.id](http://blankonlinux.or.id))

Pada 2007 pengembang BlankOn 2.0 mengubah distro acuan, dari Fedora ke Ubuntu, karena popularitas Ubuntu yang jauh lebih tinggi dari Fedora pada saat itu. BlankOn 2.0 hingga 7.0 berbasis Ubuntu, sedangkan BlankOn 8.0 berbasis Debian, induk dari Ubuntu. BlankOn 8.0 sudah menggunakan perangkat lunak desktop atau GUI (Graphical User Interface) buatan sendiri yang diberi nama Manokwari, diambil dari nama kota di provinsi Papua Barat. Manokwari merupakan shell desktop modern GNOME3 berbasis Gtk. Desktop Manokwari dibuat dengan HTML5.

Berikut ini versi-versi BlankOn berbasis Ubuntu/Debian dan nama kode serta penjelasannya.

- BlankOn 2.0 Konde – bagian belakang dari tutup kepala khas Jawa.
- BlankOn 3.0 Lontara – aksara daerah Bugis Sulawesi Selatan.
- BlankOn 4.0 Meuligoe – gapura atau bagian depan rumah khas Aceh.
- BlankOn 5.0 Nanggar – palu godam dalam bahasa Batak Sumatera Utara.
- BlankOn 6.0 Ombilin – nama daerah di Sumatera Barat.
- BlankOn 7.0 Pattimura – nama pahlawan nasional dari Maluku.
- BlankOn 8.0 Rote – nama pulau dan kabupaten di Nusa Tenggara Timur.
- BlankOn 9.0 Suroboyo – nama kota di provinsi Jawa Timur.
- BlankOn 10 Tambora – nama gunung di pulau Sumbawa Nusa Tenggara Barat.

Sejak BlankOn 8.0 Tim Pengembang BlankOn mengubah basis pengembangan dari Ubuntu ke Debian (distro induk Ubuntu), dengan alasan pengembangan mengikuti distro Debian lebih mudah dan fleksibel dibandingkan menginduk ke Ubuntu. Sejak 8.0 itu BlankOn sejajar dengan Ubuntu dari sisi keturunan distro. Ubuntu dikembangkan oleh perusahaan Canonical.

Sedangkan Debian dikembangkan oleh Yayasan Debian bersama komunitas Debian yang sangat besar.

#### 4.3.2 Arti Kata BlankOn

Mulanya nama BlankOn diambil dari kata belangkon, tutup kepala khas beberapa suku/budaya di Indonesia, antara lain suku Jawa (sebagian besar berasal dari provinsi Jawa Tengah, DIY, dan Jawa Timur), suku Sunda (sebagian besar berasal dari provinsi Jawa Barat dan Banten), suku Madura, suku Bali, dan lain-lain. Kata BlankOn dipilih, bukan Belangkon, untuk melambangkan *blank* (bilangan biner 0) dan *on* (bilangan biner 1), sehingga dapat berarti topi digital (modern) dengan tampilan klasik (kuno). Arti lain kata BlankOn adalah perubahan dari *blank* (kosong) menjadi *on* (menyala atau berisi).

Arti filosofi BlankOn adalah harapan agar pengguna komputer di Indonesia berubah dari belum tahu (kosong) menjadi tahu (berisi) bahwa ada Linux dan FOSS yang bisa dimanfaatkan untuk meningkatkan *skill* di bidang TI, martabat, dan kemandirian bangsa Indonesia. BlankOn juga diharapkan menjadi penggerak (activator) atau meningkatkan motivasi masyarakat Indonesia untuk menggunakan dan mengembangkan Linux dan produk FOSS lainnya. BlankOn juga sebagai pelindung (tutup kepala) dari ketergantungan terhadap *software proprietary*.

#### 4.3.3 Tim Pengembang BlankOn

Tim pengembang BlankOn terdiri atas beberapa kelompok sesuai dengan bidang tugasnya. Pemimpin pengembang BlankOn 1.0 adalah **Achmad Sofyan**, yang sebelumnya bergabung ke pengembang Linux WinBI (BPPT Indonesia), kemudian Trustix Linux (Norwegia), dan mengembangkan distro sendiri Rimba Linux. Mulai BlankOn 2.0, pemimpin tim pengembang BlankOn adalah **Mohammad Dhani Anwari**. Sebelum menjadi direktur eksekutif Tim BlankOn, Anwari alias MDAMT selama beberapa tahun bekerja di Nokia Finlandia sebagai pengembang Linux/FOSS untuk perangkat bergerak (mobile device) khusus Nokia, seperti Maemo/Meego.



Gambar 4.4 Tampilan OpenOffice pada BlankOn Lontara (Wikipedia)

Berikut ini daftar kelompok tim pengembang BlankOn 9.0 Suroboyo dan nama

koordinatornya, sesuai info yang tersedia di [dev.blankonlinux.or.id/wiki/9/TimPengembang](http://dev.blankonlinux.or.id/wiki/9/TimPengembang) pada 29/10/2013:

- Menejer Rilis: **Ahmad Haris** alias princeofgiri, yang bertugas membuat jadwal rilis, membentuk tim rilis, menyelenggarakan lokakarya-lokakarya pengembangan BlankOn yang terkait tim rilis , memantau pengembangan BlankOn, memberikan motivasi kepada tim rilis, dan memastikan jadwal rilis tepat waktu.
- Tim Infrastruktur: **Budi Wijaya** dan kawan-kawan yang bertanggung jawab mengelola serta mengembangkan infrastruktur yang digunakan dalam pengembangan proyek BlankOn. Beberapa server dikelola oleh tim infrastruktur, antara lain mengelola administrasi sistem secara umum seperti akun SSH, layanan web, bsr/git, GPG, pabrik-pabrik atau mesin-mesin program di server untuk menghasilkan paket, CD, dan repositori (lumbung paket).
- Tim Riset dan Aplikasi: **Fitra Aditya** alias Rabitya dan kawan-kawan yang bertugas melakukan riset untuk pengembangan program, antara lain penyunting partisi hard disk (partition editor) untuk BlankOn 9.0.
- Tim Pemaket: **Mahyudin Ramli** alias dotoVR dan kawan-kawan yang bertugas membuat dan merawat paket-paket program yang ada pada distribusi BlankOn, baik paket impor, khas, maupun paket mentah.
- Tim Kesenian: **Kukuh Syafaat** alias cho2 dan kawan-kawan yang bertugas membangun tampilan visualisasi, garfis dan audio, untuk distro BlankOn. Para pembuat artwork ini diberi nama Tim Kesenian.
- Tim Jaminan Kualitas: **Adrianus Yoza** dan kawan-kawan yang bertugas menjamin setiap aspek pada proyek BlankOn berjalan mulus sesuai dengan standar, spesifikasi, dan kebutuhan pengguna.
- Tim Dokumentasi: **Shokibi** alias Istana dan kawan-kawan yang bertugas memberikan informasi pada pengguna BlankOn sehingga pengguna mudah mempelajari BlankOn dan mendapatkan informasi yang akurat dan terpercaya. Hasilnya antara lain berbentuk buku panduan lengkap tentang BlankOn, panduan instalasi, dan panduan kustomisasi.
- Tim Humas: **Ainul Hakim** dan kawan-kawan yang bertugas menyusun dan menjalankan strategi proses komunikasi bertujuan untuk membangun citra produk dan komunikasi yang saling menguntungkan antara pengembang dan pengguna, serta masyarakat umum lainnya.

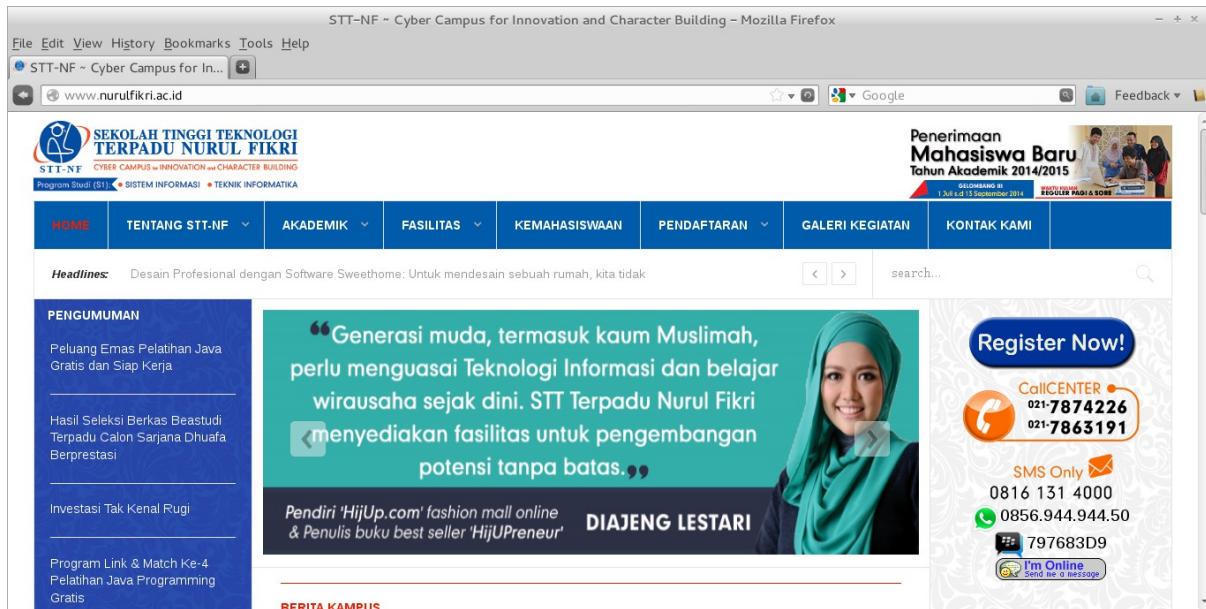
## 4.4 Contoh Pengembangan Open Source yang Berawal dari Closed

Dua produk FOSS yang sangat sukses diterima pengguna komputer dan smartphone/tablet adalah web browser Firefox dan sistem operasi Linux Android. Firefox atau Mozilla menjadi tonggak sejarah browser web FOSS setelah kejayaan Microsoft Internet Explorer selama beberapa tahun. Demikian pula Android menjadi tonggak sejarah sistem operasi smartphone setelah kejayaan Nokia Symbian, BlackBerry, dan Microsoft Windows Phone/Mobile selama bertahun-tahun.

### 4.4.1 Mozilla Firefox

Sejarah Mozilla Firefox bermula dari pengumuman oleh perusahaan bernama Netscape (Netscape Communications Corporation) pada 31 Maret 1998, bahwa Netscape telah membuka kode sumber program browser web Netscape Navigator melalui proyek

Mozilla.org (Mozilla Organization). Proyek Mozilla ini dibentuk Netscape pada 23 Februari 1998. Kemudian pada 15 Juli 2003 terbentuk yayasan Mozilla (Mozilla Foundation). Firefox 1.0 dirilis pada 9 November 2004, dan pada 3 Agustus 2005 Mozilla membuat perusahaan Mozilla Corporation untuk mengelola bisnis yang terkait produk-produk Mozilla seperti web browser Firefox dan email client Thunderbird. Netscape pada 1999 diakuisisi oleh AOL (American On Line).



Gambar 4.5 Tampilan Mozilla Firefox

Berbeda dengan kernel Linux yang sejak pertama dikembangkan sudah dirilis sebagai produk open source, browser web Mozilla berawal dari Netscape Navigator yang closed source, produk proprietary dari Netscape/AOL. Semua pengembang Mozilla awalnya adalah karyawan yang digaji oleh Netscape/AOL. Setelah cara pengembangan berubah dari proprietary (katedral) menjadi open source (bazar), Mozilla berkembang menjadi lebih cepat, meskipun tidak ada lagi karyawan Netscape/AOL yang bekerja untuk Mozilla.

#### 4.4.2 Linux Android

Sejarah Android mirip namun tidak sedahsyat perubahan Netscape Navigator menjadi Mozilla. Android berawal dari pengembangan tertutup (katedral), meskipun Android dikenal menggunakan kernel Linux yang open source. Android dikembangkan sejak 2003 oleh perusahaan Android Inc. Pada 17 Agustus 2005 Android Inc dibeli Google.

Google mengembangkan Android bersama pengembang lain dari berbagai perusahaan yang tergabung dalam organisasi Open Handset Alliance (OHA). OHA didirikan pada 5 November 2007. Android 1.0 dirilis pada 22 Oktober 2008. Pada 2013 Android menjadi jadi sistem operasi terbesar di dunia karena menguasai lebih dari 70 persen pasar komputer tablet dan komputer genggam (smartphone).

### 4.5 Rangkuman

Paling tidak ada tiga jalan sebuah produk menjadi open source. Pertama, produk yang sejak awal dikembangkan secara open source dan dilisensikan sebagai produk open source, misal kernel Linux. Kedua produk yang awalnya tidak dibuat secara open source dan tidak akan dijadikan open source, namun kemudian diubah menjadi open source, misal browser web Netscape Communicator yang di-open-source-kan menjadi Mozilla. Contoh lain yang seperti

Mozilla adalah StarOffice, produk proprietary yang kemudian di-open-source-kan menjadi OpenOffice dan memiliki varian bernama LibreOffice. Ketiga, produk yang direncanakan menjadi open source namun awalnya dikembangkan secara tertutup, misal Android, yang baru dirilis/disebarluaskan setelah produk matang.

Setiap produk open source besar memiliki organisasi dan tim pengembang yang bekerja dengan cara pengembangan open source (bazar). Contoh: Kernel Linux dikembangkan di bawah organisasi Linux Foundation, Firefox dikembangkan di bawah Mozilla Corporation, Apache dan OpenOffice di bawah Apache Foundation, LibreOffice di bawah The Document Foundation, Java dan MySQL di bawah Oracle, Android di bawah Google, Debian di bawah Debian Foundation, Ubuntu di bawah Canonical, Fedora di bawah Red Hat, BlankOn di bawah Yayasan Penggerak Linux Indonesia dan Tim Pengembang BlankOn, dan lain-lain.

Biaya pengembangan produk open source umumnya ditanggung banyak pihak dari berbagai institusi (perusahaan, pemerintahan, yayasan, dsb.) dan personal. Pengembang atau pemberi dukungan pengembang produk open source mendapatkan manfaat dari hasil pengembangan. Google diuntungkan, karena semakin banyak pengguna Android semakin banyak pula orang yang mengakses produk internet Google, yang berakibat menaikkan pemasukan iklan Google. Samsung, LG, Sony, Huawei, dan produsen hardware lainnya diuntungkan karena produknya terjual dengan harga lebih murah daripada membuat sendiri semua software seperti yang dilakukan BlackBerry. Akibatnya, harga jual perangkat BlackBerry lebih mahal dibandingkan Android untuk spesifikasi hardware (seperti jenis prosesor, ukuran memory, ukuran dan jenis layar) yang setara.

# Bab 5 Komunitas Open Source

Tujuan Umum: Memahami arti, tujuan, dan jenis komunitas Open Source.

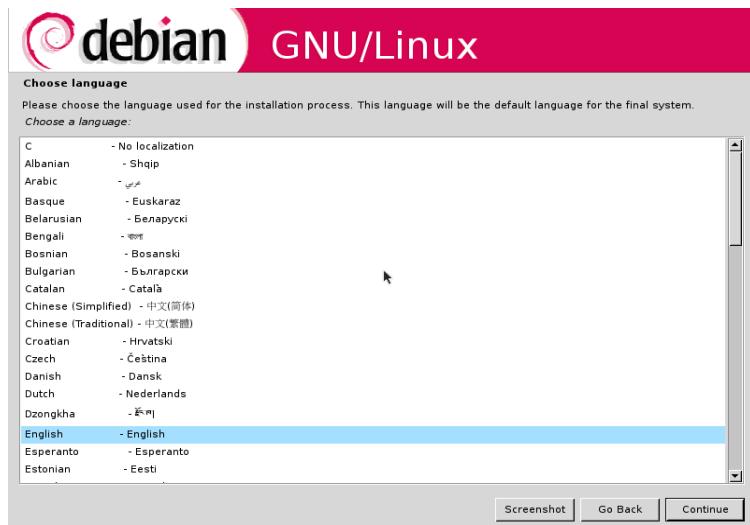
Tujuan Khusus:

- Mampu membedakan beberapa komunitas Open Source.
- Mampu mendeskripsikan beberapa motivasi orang bergabung ke komunitas.

## 5.1 Sejarah dan Definisi Komunitas Open Source

### 5.1.1 Sejarah Komunitas Open Source

Komunitas open source sudah ada sejak awal perkembangan komputer hingga awal 1970-an. Namun karena saat itu umumnya software komputer dikembangkan secara bersama dengan berbagai (*open*) kode sumber program (*source code*), maka belum “ada” istilah open source maupun free software. Kemunculan sistem operasi Unix yang dikembangkan bersama sejak 1969, disusul pengembangan bahasa pemrograman C pada 1972 dan penemuan teknologi jaringan internet TCP/IP pada 1973, menjadi titik awal berkembangnya komunitas komputer. Mereka telah berhasil mengubah dunia dari terisolasi oleh wilayah geografis menjadi satu “sambungan” yang disebut internet, karena Unix/Linux, C/C++, TCP/IP, dan berbagai program open source lainnya menjadi tulang punggung internet dunia. Sebagian besar anggota komunitas saat itu berasal dari lembaga pendidikan dan riset. Mereka adalah komunitas pengembang yang sekaligus juga komunitas pengguna.



Gambar 5.1 Debian Linux yang dikembangkan komunitas (Wikipedia)

Setelah Richard Stallman membangun Proyek GNU dan mendirikan Yayasan Free Software pada 1984/1985 ([www.fsf.org](http://www.fsf.org)), muncul berbagai komunitas pengembang dan pengguna open source. Proyek GNU berhasil membuat beragam program komputer, namun belum mampu menghasilkan inti atau kernel sistem operasi. Baru setelah Linus Torvalds menyebarluaskan kode sumber program kernel Linux pada 1991, sistem operasi dan aplikasi open source menjadi siap digunakan dan dikembangkan secara berar-besaran. Istilah open source baru dipromosikan secara serius pada 1997/1998 oleh Eric S. Raymond dengan membentuk organisasi Open Source Initiative ([www.opensource.org](http://www.opensource.org)).

### **5.1.2 Definisi Komunitas Open Source**

Kata komunitas merupakan istilah bentukan dari kata bahasa Inggris *community* yang berasal dari bahasa latin *communitas*, yakni *com* berarti bersama (*with/together*) dan *munus* berarti pemberian (*gift*). Komunitas dapat diartikan sekelompok orang yang saling berbagi untuk kepentingan bersama. Jadi, komunitas open source adalah sekelompok orang yang saling berinteraksi dan berbagi (*sharing*) dalam berbagai hal di bidang open source (software maupun non software).

Dalam hal pengembangan suatu produk software open source, komunitas pengembangnya terdiri atas personal-personal independen, yang berinteraksi dengan beragam kepentingan dan motivasi untuk menghasilkan produk open source. Personal independen itu dapat mewakili diri sendiri atau institusi tempatnya bekerja. Sebagai contoh, komunitas pengembang browser web Mozilla awalnya berasal dari karyawan perusahaan Netscape, kemudian bertambah banyak pengembang dari luar Netscape. Contoh lain, komunitas pengembang kernel Linux terdiri atas personal-personal sebagai pribadi dan personal-personal yang mewakili perusahaan seperti IBM dan lain-lain.

## **5.2 Jenis Komunitas Open Source dan Contohnya**

Dalam buku “Open Source, A Multidisciplinary Approach” Moreno Muffatto menjelaskan komunitas terdiri atas lima kategori atau pemain dan tiga peran. Lima kategori itu adalah pengguna biasa (*users*), pengguna yang juga pengembang (*prosumers: producers & consumers*), pengembang utama (*leader teams*), perusahaan (*companies*), dan institusi non perusahaan (*institutions*) seperti perguruan tinggi dan pemerintah. Kelima jenis komunitas itu dapat memiliki tiga peran dalam proses pengembangan software, yakni sebagai pelanggan (*customer*), pelaku pengembangan (*actor*), dan pembuat keputusan dalam pengembangan (*decision maker*). Artinya, pengguna juga berperan sebagai pelaku pengembangan dengan cara memberi umpan-balik atau usulan kepada pengembang utama, dan berperan juga dalam pengujian kualitas produk open source. Demikian pula perusahaan, selain menggunakan di internal perusahaan juga dapat berkontribusi dengan menugaskan karyawan sebagai pengembang dan turut mengambil keputusan dalam proses pengembangan produk lebih lanjut atau pendistribusianya.

### **5.2.1 Contoh Komunitas Pengguna**

Ada komunitas open source yang lebih fokus dalam penggunaan produk, meskipun sebagian anggotanya juga ikut menjadi pengembang atau *prosumers*. **Linux User Group** atau Kelompok Pengguna Linux merupakan komunitas pengguna yang tersebar luas, ada di berbagai belahan dunia, termasuk di Indonesia yang dinamakan KPLI (Kelompok Pengguna Linux Indonesia) tersebar di berbagai kota dan provinsi. Komunitas ini berdiri sejajar satu dengan yang lain. Misal KPLI Jakarta memiliki posisi sejajar dengan KPLI Depok, KPLI Bogor, KPLI Bekasi, KPLI Bandung, KPLI Jogja, KPLI Malang, KPLI Surabaya, dan lain-lain. Beberapa KPLI memiliki nama khas daerah, misal LinuxJak untuk KPLI Jakarta, JogXer untuk KPLI Jogja, KOLAM (Komunitas Linux Arek Malang Raya) untuk KPLI Malang, dan KLAS (Kelompok Linux Arek Suroboyo) untuk KPLI Surabaya.

Organisasi maya (*virtual organization*) yang menjadi penghubung KPLI-KPLI ini disebut Linux Indonesia dengan alamat domain untuk web dan email [linux.or.id](http://linux.or.id). Komunitas ini berkoordinasi melalui email (mailing list). Pertemuan darat diadakan setahun sekali, keliling dari satu kota ke kota lain, dengan nama acara Indonesia Linux Conference (ILC) dan KPLI Meeting. Ada beberapa KPLI yang aktif mengadakan pertemuan di darat, antara lain KPLI Jogja dan KPLI Surabaya. KPLI Jogja beranggotakan personal dan KSL-KSL perguruan tinggi di Yogyakarta.



Gambar 5.2 KPLI Kediri

Komunitas pengguna produk open source yang sedang tumbuh pesat adalah Android, antara lain berbagi tip-trik penggunaan hingga pengembangan aplikasi Android untuk hp dan tablet. Komunitas Blender Indonesia juga tumbuh pesat karena popularitas software itu dalam pembuatan film animasi dan desain 3D. Setiap distro Linux populer juga memiliki komunitas pengguna, misal komunitas Ubuntu ID, BlankOn, Fedora, dan sebagainya. Produk open source untuk pengembangan software juga memiliki komunitas, misal PHP User Group, Java User Group, dan lain-lain.

### 5.2.2 Contoh Komunitas Pengembang

Komunitas pengembang open source adalah kumpulan orang yang bergabung dalam suatu kegiatan pengembangan produk open source. Meskipun membuat program (*coding*) merupakan inti dari pengembangan software, komunitas pengembang tidak hanya berisi para programer, karena proses pengembangan produk software membutuhkan juga orang yang memiliki kemampuan teknis dan non teknis di luar pemrograman. Contoh profesi atau peminatan selain programer yang dapat berbagi dalam komunitas pengembang adalah desainer grafis, pembuat dokumen atau buku manual, penguji produk atau penjamin kualitas, pengelola infrastruktur atau server dan jaringan untuk pengembangan software, helpdesk yang membantu pengguna, humas dan pemasaran, pelokalan (*localization*), dan lain-lain.

Meskipun pada awalnya suatu produk software hanya dikembangkan oleh satu orang, setelah produk dipublikasikan akan muncul komunitas pengembang jika produk itu digunakan banyak orang. Saat ini dapat dipastikan ada komunitas pengembang pada tiap produk open source yang telah matang dan banyak penggunanya. Berikut ini hanya empat contoh komunitas dari ribuan komunitas pengembang produk open source yang ada.

- **Pengembang Apache ([www.apache.org](http://www.apache.org)).** Apache terkenal sebagai nama server web paling banyak digunakan di internet. Menurut situs [netcraft.com](http://netcraft.com), pada November 2013 jumlah server web yang mencapai hampir 1 milyar, sekitar 44 persen di antaranya menggunakan Apache. Komunitas pengembang Apache sangat besar, dan produk yang dihasilkan juga banyak, tidak hanya server web. Pengembang produk-produk Apache, termasuk aplikasi perkantoran OpenOffice, menginduk di bawah The Apache Software Foundation. Yayasan ini didukung dana oleh banyak perusahaan besar seperti Citrix, Facebook, Google, IBM, Microsoft, Yahoo, dan lain-lain, termasuk orang kaya menyumbang sebagai pribadi Matt Mullenweg pengembang WordPress (aplikasi open source untuk web blog atau portal).
- **Pengembang Kernel Linux ([www.kernel.org](http://www.kernel.org)).** Linus Torvalds awalnya membuat program inti (*core*) sistem operasi atau kernel Linux sendirian, ketika ia masih menjadi mahasiswa komputer di Universitas Helsinki Finlandia. Pada September 1991 Linus membagi kode sumber kernel melalui jaringan internet pada suatu kelompok diskusi (News Group). Kernel buatan Linus berlisensi open source GNU

GPL dan dapat digabung dengan program open source lain seperti GNU Libc dan shell (BASH) menjadi sistem operasi lengkap. Tanggapan programer komputer sangat baik dan cepat, sehingga pengembang kernel Linux menjadi besar dan terus membesar hingga pada 2013 ini menjadi sistem operasi yang paling banyak digunakan orang di seluruh dunia melalui Android. Organisasi pendukung komunitas pengembang kernel Linux adalah Linux Foundation ([www.linuxfoundation.org](http://www.linuxfoundation.org) dan [www.linux.com](http://www.linux.com)) yang didanai oleh banyak perusahaan besar seperti IBM, Intel, Oracle, Samsung, HP, NEC, Fujitsu, Huawei, AMD, CISCO, Google, Nissan, Toyota, dan lain-lain. Yayasan Linux bersama Intel dan Samsung – saat tulisan ini disusun – sedang mengembangkan sistem operasi komputer Tizen.

- **Pengembang Mozilla ([www.mozilla.org](http://www.mozilla.org)).** Komunitas pengembang browser web Mozilla Firefox awalnya berasal dari sebuah perusahaan Netscape/AOL (American On Line) yang mengembangkan browser web Netscape Navigator atau Netscape Communicator. Setelah Netscape mengubah lisensi web browser-nya menjadi open source dengan nama proyek Mozilla ([www.mozilla.org](http://www.mozilla.org)), komunitas pengembangnya membesar dengan cepat. Selain browser web, komunitas pengembang Mozilla juga mengembangkan produk open source untuk email Thunderbird, sistem operasi perangkat bergerak Mozilla OS, dan lain-lain. Organisasi yang mendukung komunitas Mozilla adalah Mozilla Corporation.
- **Pengembang BlankOn ([www.blankonlinux.or.id](http://www.blankonlinux.or.id)).** Pengembang BlankOn melakukan kolaborasi pengembangan distro Linux BlankOn melalui internet, seperti umumnya pengembang open source besar dunia. Beberapa fasilitas yang banyak digunakan Tim Pengembang BlankOn adalah email melalui mailing list BlankOn-Dev@googlegroups.com, chatting melalui IRC, serta software khusus pengembangan bersama seperti git/bzr dan wiki. BlankOn didukung YPLI dan lain-lain. Tim Pengembang BlankOn juga bertemu darat khusus pengembang (BlanKopdar) dan konferensi untuk umum (BlanKonf) setahun sekali.

### 5.2.3 Contoh Komunitas Lainnya

Komunitas open source lainnya dapat berupa gabungan antara pengguna, pengembang, pebisnis, dan lain-lain. Anggota komunitas gabungan ini belum tentu menggunakan produk open source untuk pribadi maupun perusahaannya. Anggota bergabung ke dalam komunitas untuk mengambil manfaat bagi bisnis atau institusi atau negara atau alasan lain. Contoh komunitas gabungan ini adalah AOSI (Asosiasi Open Source Indonesia), yang beranggotakan perorangan maupun institusi bisnis, pendidikan, dan pemerintahan.



Gambar 5.3 Komunitas open source AOSI ([aosi.or.id](http://aosi.or.id))

### 5.3 Alasan dan Manfaat Bergabung ke Komunitas

Setiap orang – sebagai pribadi atau mewakili institusinya – bergabung ke komunitas pasti memiliki alasan atau motivasi terkait dengan manfaat yang ingin diperolehnya setelah bergabung dengan sebuah komunitas. Untuk mengetahui alasan atau motivasi seseorang bergabung ke komunitas (khususnya pengembang) open source, Boston Consulting Group (BCG) bekerja sama dengan Open Source Development Network (OSDN) melakukan survei pada 2002 terhadap ribuan pengembang open source. Rangkuman hasilnya berupa persentase alasan utama bergabung ke komunitas open source adalah sebagai berikut<sup>4)</sup>.

- 93% untuk meningkatkan pengetahuan dan skill.
- 50% untuk mengembangkan reputasi atau pengakuan secara luas.
- 33% untuk mendapatkan pekerjaan baru.

Ketiga alasan utama itu menunjukkan manfaat bergabung ke komunitas open source adalah mendapatkan tambahan ilmu pengetahuan dan skill dalam penggunaan dan pengembangan software, sehingga mendapatkan pengakuan dan pekerjaan baru atau pekerjaan yang lebih baik dari sebelumnya.

Manfaat bergabung ke komunitas bagi organisasi bisnis adalah mendapatkan keuntungan secara bisnis secara langsung maupun tidak langsung. Misal para perusahaan yang bergabung dalam komunitas pengembang Android OHA (Open Handset Alliance) mendapatkan produk yang bagus dengan biaya pengembangan murah karena biaya ditanggung bersama. Itu terbukti dengan Samsung yang melejit mengalahkan Apple, BlackBerry, dan Nokia dalam penjualan produk hp dan tablet setelah ikut mengebangkan dan menggunakan Android. Demikian pula HP, IBM, Oracle dan lain-lain bergabung ke komunitas pengembang kernel Linux untuk mendapatkan sistem operasi server mereka dengan biaya pengembangan yang rendah dibandingkan mengembangkan sendiri.

Bergabung ke komunitas juga menguntungkan perusahaan yang membutuhkan banyak pengembang dan penyedia jasa untuk produknya yang terjual. Perusahaan tersebut tidak harus investasi sendiri banyak divisi atau cabang penyedia layanan purna jual atau dukungan teknis (*technical support*), pelatihan (*training*), pengembangan aplikasi khusus, pengintegrasikan sistem, dan lain-lain.

Empat alasan utama perusahaan mendukung komunitas open source (Moreno Muffatto):

- Menemukan karyawan teknis yang kompeten.
- Meningkatkan kualitas software.
- Memanfaatkan komunitas open source untuk riset dan pengembangan.
- Meningkatkan citra perusahaan.

Empat alasan utama asosiasi atau pemerintah mendukung komunitas open source menurut Moreno Muffatto<sup>4)</sup>:

- Memanfaatkan open standard untuk mendorong kompetisi, penurunan biaya, dan peningkatan kualitas produk.
- Melawan kekuatan monopoli.
- Mendapatkan akses ke pengembangan software berbiaya rendah.
- Mengurangi kesenjangan digital (akses informasi secara elektronik).

## 5.4 Organisasi Komunitas Open Source

Umumnya komunitas berbentuk “cair” meskipun dikoordinasi oleh sebuah organisasi. Moreno Muffatto menyebutkan ciri utama organisasi komunitas pengembang open source adalah partisipasi terbuka (*open participation*). Penentuan siapa saja yang memimpin organisasi ditentukan berdasarkan besarnya kontribusi yang telah diberikan atau sistem merit. Tidak sembarang orang dapat menjadi tim inti pengembang suatu produk open source.

Organisasi komunitas pengembang open source dapat berbentuk formal dan non formal. Di Indonesia, organisasi disebut formal atau legal-formal jika memiliki akta notaris dan terdaftar di sistem administrasi pemerintahan, misalnya Kementerian Hukum dan HAM. Contoh organisasi formal komunitas open source adalah Asosiasi Open Source Indonesia (AOSI) dan Yayasan Penggerak Linux Indonesia (YPLI) yang mewadahi organisasi pengembang Linux BlankOn.

Organisasi disebut non formal atau non legal-formal jika tidak memiliki akta notaris, dan/atau tidak terdaftar di sistem administrasi pemerintahan, atau tidak menginduk ke organisasi formal. Contoh organisasi non formal adalah Linux User Group atau umunya Kelompok Pengguna Linux Indonesia yang ada di beberapa kota di Indonesia. Kelompok Pengguna Linux dapat menjadi legal-formal jika menginduk atau berada di bawah organisasi formal, misalnya Tim Pengembang BlankOn berada di bawah Yayasan Penggerak Linux Indonesia. Kelompok Studi Linux di kampus-kampus disebut formal jika menginduk ke organisasi yang resmi di perguruan tinggi.



Gambar 5.4 Beberapa Komunitas Linux Indonesia (Sumber: [internetpositif.com](http://internetpositif.com))

Komunitas pengembang open source besar dunia biasanya dalam koordinasi organisasi formal berbentuk yayasan (not for profit, laba tidak dibagikan ke pendiri) atau perusahaan (for profit). Contoh organisasi berbentuk yayasan yang telah disebut sebelumnya antara lain Free Software Foundation, Linux Foundation, Debian Foundation, The Apache Software Foundation, Mozilla Foundation, The Document Foundation (LibreOffice), dan lain-lain. Contoh komunitas pengembang yang dikoordinasi atau disponsori perusahaan adalah Linux Ubuntu (Canonical), Android (Google), Fedora (Red Hat), Tizen (Samsung), MySQL dan Java (Oracle), dan lain-lain.

# Bab 6 Keuntungan/Kelebihan Open Source Software

Tujuan Umum: Mampu memahami keuntungan-keuntungan institusi bila memilih menggunakan produk Open Source dibandingkan produk Proprietary.

Tujuan Khusus:

- a) Mampu menjelaskan keuntungan cara Open Source secara umum, untuk pengembang dan pengguna.
- b) Mampu menjelaskan paling tidak lima keuntungan memilih produk Open Source bagi institusi bisnis, pemerintah, pendidikan, LSM, dan personal/keluarga.

## 6.1 Pendahuluan

Salah satu manfaat atau kelebihan pengembangan software secara open source (terbuka) adalah biaya yang lebih kecil daripada menggunakan cara proprietary (tertutup), seperti telah dibahas pada bab sebelumnya. Demikian pula manfaat bagi pengguna, produk FOSS (Free/Open Source Software) memiliki total biaya kepemilikan atau *TCO (Total Cost of Ownership)* yang lebih rendah daripada TCO produk proprietary. Total biaya kepemilikan itu mencakup biaya pengadaan (termasuk jika ada biaya lisensi), biaya pemasangan, biaya perawatan atau maintenance (termasuk biaya update atau upgrade), dan lain-lain.

Selain penghematan biaya, beberapa manfaat produk FOSS antara lain sebagai berikut:

- Keamanan (*Security*) lebih baik
- Ketersediaan/Kestabilan (*Realibility/Stability*) lebih tinggi
- Lebih sesuai standar terbuka dan tidak tergantung vendor
- Mengurangi ketergantungan terhadap impor
- Meningkatkan kemampuan mengembangkan perangkat lunak lokal
- Mengurangi pelanggaran HaKI (Hak atas Kekayaan Intelektual)
- Memudahkan penggunaan bahasa dan budaya lokal (*localization*)

## 6.2 Keamanan (*Security*)

Meskipun tidak ada sistem operasi atau platform yang aman secara sempurna aman, faktor-faktor seperti metode pengembangan, arsitektur program, dan pasar target dapat berpengaruh besar terhadap keamanan, dan konsekuensinya dapat berakibat lebih mudah ditembus atau sebaliknya sulit ditembus. Ada beberapa indikator bahwa sistem FOSS memiliki nilai lebih baik dalam hal keamanan dibandingkan sistem proprietary:

- Gartner Group merekomendasikan agar perusahaan bermigrasi dari server Microsoft IIS (Internet Information Services) ke Apache atau server web lainnya, karena IIS memiliki rekaman perjalanan keamanan yang buruk. Menurut Gartner Group, pada Juli 2002, perusahaan-perusahaan di Amerika Serikat menghabiskan US\$ 1,2 miliar (sekitar Rp 13 trilyun) hanya untuk mengatasi kelemahan IIS terkait Code Red.
- “Jaminan Hacker” yang dikeluarkan J.S. Wuzler Underwriting Managers membutuhkan biaya 5 hingga 15 persen lebih besar jika menggunakan MS Windows, dibandingkan jika menggunakan sistem Linux/UNIX. Walter Kopf, *senior vice president of underwriting* di J.S. Underwriting Managers berkata, “Kami telah menemukan kemungkinan rugi lebih besar jika menggunakan sistem operasi server Windows.”

Aspek keamanan telah mendorong banyak organisasi publik untuk bermigrasi, atau mempertimbangkan untuk migrasi dari Windows ke solusi open source. Lembaga pajak dan kepabeanan Perancis migrasi ke Red Hat Linux secara besar-besaran karena alasan keamanan ini.

Tiga hal yang sering dijadikan alasan bahwa FOSS lebih aman:

- Ketersediaan kode sumber (*open source*).
- Lebih fokus kepada keamanan (*security*) daripada keindahan (*user friendly*).
- Sebagian besar sistem operasi open source berbasis multi user dan UNIX yang siap untuk jaringan dan relatif lebih aman.

### 6.3 Ketersediaan/Kestabilan (*Reliability/Stability*)

Sistem FOSS sangat dikenal dengan ketersediaan dan kestabilannya (tidak mudah *hang* atau minta *restart*). Ada banyak cerita anekdot, bahwa server FOSS bekerja baik bertahun-tahun tanpa minta perawatan khusus. Dua studi yang terkait hal itu:

- Pada 1999, Zdnet menjalankan pengujian reliabilitas selama 10 bulan antara Red Hat Linux, Caldera Open Linux, dan Microsoft Windows NT Server 4 SP3. Ketiga server menggunakan perangkat keras yang identik dengan memberikan layanan atau fungsi print server, web server, dan file server. Hasilnya, server NT hang atau crash setiap 6 minggu, tapi tidak ada satu pun server Linux yang hang atau crash selama 10 bulan pengujian.
- Uji beban dengan pengujian secara acak telah dilakukan terhadap tujuh sistem komersial dan sistem GNU/Linux pada 1995. Hasilnya, sistem komersial memiliki kegagalan rata-rata sebesar 23%, sedangkan sistem Linux hanya 9%. Utilitas GNU (perangkat lunak yang dihasilkan projek GNU) gagal hanya 6%. Beberapa tahun kemudian, studi lanjutan menemukan bahwa kegagalan yang terjadi dalam sistem FOSS dapat teratasi, namun tidak demikian dengan perangkat lunak proprietary karena tidak bisa menyentuh kode sumber-nya.

### 6.4 Standar Terbuka dan Tidak Bergantung Vendor

Standar terbuka (*open standard*) memberikan fleksibilitas dan kebebasan kepada pengguna, baik individu, perusahaan, atau pemerintahan. Pengguna dapat berganti paket software, berganti platform, atau vendor yang berbeda, tanpa menimbulkan masalah. Standar proprietary yang biasanya bersifat rahasia mengunci pengguna untuk menggunakan software hanya dari sebuah vendor.



Gambar 6.1 LibreOffice software open source pendukung format ODF (Wikipedia)

Pada tahap selanjutnya pengguna proprietary sangat mengharap belas kasihan vendor jika semua datanya dalam format yang hanya dimiliki vendor dan kesulitan untuk mengubahnya menjadi standar terbuka. Dua contoh standar terbuka adalah X Window dan format dokumen perkantoran (Open Document Format).

Penyusun artikel “*Free/Libre and Open Source Software: Survey and Study*” yang diterbitkan Institut Infonomika Internasional di Belanda juga menentang penggunaan software proprietary di pemerintah, dengan alasan utama menentang implementasi *proprietary software* di sektor publik adalah ketergantungan terhadap vendor software tersebut. Jika standar proprietary telah digunakan secara mapan, keharusan untuk mengikutiinya tidak bisa dihindari. Bahkan dalam sebuah pengadaan sistem dengan tender terbuka, adanya keharusan kompatibel dengan standar proprietary membuat pengadaan sistem menjadi bias ke arah sebuah vendor, sehingga meneruskan ketergantungan.

Kelebihan lain dari FOSS adalah hampir selalu menggunakan standar terbuka. Pilihan standar terbuka ini sesuai dengan dua alasan utama:

- **Ketersediaan kode sumber.** Adanya kode sumber program selalu memungkinkan untuk rekayasa ulang dan mendokumentasikan standar yang digunakan oleh aplikasi. Semua variasi pengembangan dapat terlihat secara transparan dalam kode sumber, sehingga tidak memungkinkan ada standar proprietary tersembunyi dalam sistem FOSS.
- **Sesuai dengan standar yang aktif.** Jika ada standar aktif yang telah digunakan, misalnya HTML (*HyperText Markup Language*) yang mengontrol bagaimana web ditampilkan, maka projek FOSS berjalan secara aktif mengikuti standar yang tepat. Salah satu karya FOSS browser web Mozilla adalah sangat sesuai (*compliant*) dengan banyak standar yang dikeluarkan W3C (*World Wide Web Consortium*). Webstandards.org mencatat bahwa Mozilla merupakan browser web yang paling mengikuti standar. Kesesuaian dengan standar adalah salah satu budaya gerakan FOSS. Berbagi dan bekerja sama dengan projek atau aplikasi lain adalah hal biasa. Bekerja dalam kelompok pengembang global yang tersebar juga menjadi lebih mudah jika ada standar yang menyertainya.

## 6.5 Mengurangi Ketergantungan Impor

Alasan utama yang mendorong negara-negara berkembang mengadopsi sistem FOSS adalah biaya lisensi yang sangat besar jika memilih perangkat lunak proprietary. Karena secara virtual perangkat lunak proprietary di negara berkembang adalah impor. Belanja perangkat lunak itu akan menghabiskan cadangan devisa. Cadangan devisa yang dihemat dengan menggunakan FOSS ini dapat digunakan untuk tujuan pengembangan lainnya.

Riset “*Free/Libre and Open Source Software: Survey and Study*” di Eropa juga mencatat bahwa biaya dalam model bisnis open source yang lebih berorientasi jasa ini normalnya hanya dikeluarkan untuk bisnis dalam negeri, tidak harus menggunakan perusahaan multinasional. Ini berdampak positif terhadap masalah tenaga kerja, investasi dalam negeri, pemasukan dari pajak, dan lain-lain.

## 6.6 Pengembangan Perangkat Lunak Lokal

Ada korelasi positif antara pertumbuhan pengembang FOSS dengan kemampuan inovasi dalam sistem ekonomi. Laporan Institut Infonomika Internasional menyebutkan tiga alasan untuk itu:

- **Mudah masuk industri perangkat lunak.** Mudah untuk mendapatkan,

menggunakan, dan mempelajari FOSS karena kebebasannya untuk dimodifikasi dan didistribusikan. Perangkat lunak proprietary cenderung membatasi, tidak hanya ketersediaan kode sumber program yang terbatas, tapi juga pembatasan terkait lisensi, paten, dan hak cipta. FOSS mendorong pengembang untuk membangun produk baru berdasarkan pengetahuan dan komponen pengembangan yang telah ada.

- **FOSS sebagai sistem pelatihan yang sempurna.** Keterbukaan dan kolaborasi yang menjadi ciri khas FOSS mendorong mahasiswa untuk menguji dan mencoba konsep-konsep perangkat lunak, tanpa perlu biaya atau membebani masyarakat. Misalnya, mahasiswa dapat bergabung ke jaringan pengembangan FOSS yang saling bekerja sama secara global, yang telah dilengkapi arsip lengkap informasi teknis dan fasilitas diskusi interaktif, seperti git/bzr dan wiki.
- **FOSS sebagai sumber standar.** FOSS sering menjadi standar de-facto karena dominansinya yang baik dalam sektor khusus suatu industri. Dengan bergabung dalam menetapkan standar untuk aplikasi FOSS khusus, suatu daerah dapat memastikan bahwa standar yang dihasilkan sesuai dengan kebutuhan dan budaya daerah.

Pendekatan pengembangan FOSS tidak hanya memfasilitasi inovasi tapi juga penyebaran hasil inovasi itu secara besar-besaran. Memo internal Microsoft menyatakan, “Projek-projek riset dan pengajaran di atas Linux menyebar secara mudah karena ketersediaan luas dari kode Linux. Dalam beberapa kasus, ini menjelaskan mengapa hasil ide atau riset baru biasanya pertama tersedia untuk Linux sebelum tersedia untuk platform lainnya.”

## 6.7 Pelanggaran HaKI dan WTO

Pelanggaran HaKI seperti pembajakan software menjadi masalah di hampir semua negara di dunia. Menurut studi pembajakan software global yang dibuat IDC dan BSA (*Bussines Software Alliance*), pembajakan software pada 2005 menimbulkan kerugian US\$34 miliar (sekitar Rp 350 trilyun). Bahkan di negara maju yang secara teori mampu membeli lisensi software, nilai pembajakannya masih tinggi, sekitar 21% di Amerika Serikat dan 36% di Uni Eropa. Di negara-negara berkembang yang penghasilannya rendah harga software dinilai mahal, sehingga angka pembajakan software sangat tinggi, misalnya di atas 80% untuk Indonesia.



Gambar 6.2 WTO dan HAKI (Wikipedia)

Pembajakan software dan lemahnya hukum dapat merugikan suatu negara dalam banyak hal. Sebuah negara yang perlindungannya terhadap HaKI (Hak atas Kekayaan Intelektual) rendah menjadi tidak menarik bagi investor asing. Keanggotaan dalam organisasi perdagangan dunia atau WTO (*World Trade Organization*) dan manfaat yang didapatkan dari keanggotaan itu juga ditentukan oleh tingkat penghargaan suatu negara terhadap HaKI. Akhirnya, budaya pembajakan software merugikan pengembang software lokal, di samping rendahnya insentif bagi pengembang software lokal untuk menghasilkan produk lokal.

## **6.8 Bahasa dan Budaya Lokal (*Localization*)**

“Pelokalan di bidang software menghasilkan penguasaan suatu produk dan membuatnya sesuai dengan bahasa dan budaya target pasar lokal yang dijadikan sasaran pengguna software.” (Sumber: *Localization Industry Standards Association*).

Pelokalan merupakan salah satu bidang yang membuat FOSS bersinar karena keterbukaannya. Pengguna dapat mengubah FOSS agar sesuai dengan kebutuhan budaya lokal, termasuk sesuai dengan skala ekonominya. Kelompok kecil atau individu yang memiliki kemampuan teknis dapat membuat versi lokal produk FOSS skala kecil. Kemudian sangat dimungkinkan jika suatu saat dibutuhkan skala lebih besar atau lokalisasi lengkap dengan menggabungkan yang kecil-kecil itu. Sebagian besar pengembangan FOSS di wilayah Asia-Pasifik dimulai dengan lokalisasi, seperti Linux BlankOn di Indonesia.

# Bab 7 Openness di bidang Teknologi Informasi

Tujuan Umum: Mampu memahami berbagai bentuk keterbukaan (Openness) di bidang TI.

Tujuan Khusus:

- a) Mampu menjelaskan arti Open Standard dan beberapa contohnya.
- b) Mampu menjelaskan arti Open Hardware.
- c) Mampu menjelaskan arti Open Content dan Open Publication dan contohnya.
- d) Mampu menjelaskan arti Open Movie, Open Education, Open Data, dll.

## 7.1 Pendahuluan

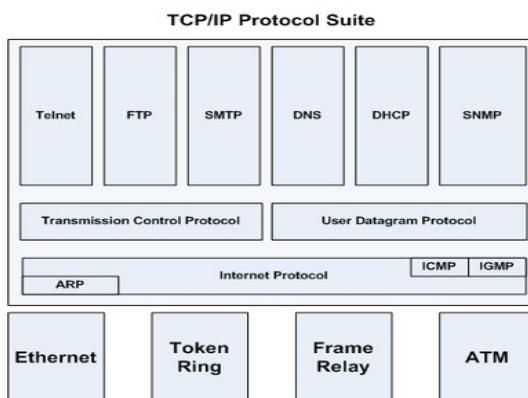
*Openness* (keterbukaan) telah membuat teknologi informasi dan komunikasi cepat berkembang, karena banyak konsep dan aplikasi di bidang teknologi informasi dikembangkan secara bersama dan terbuka. Salah satu bidang teknologi informasi yang menggunakan prinsip openness ini adalah perangkat lunak bebas/open source atau *Free/Open Source Software* yang telah dibahas pada bab-bab sebelumnya. Bab ini membahas beberapa penerapan openness selain software, yakni *open standard* dan *open format*, *open content* atau *open publication*, *open hardware*, *open movie* dan lain-lain.

## 7.2 Open Standard dan Open Format

*Standard* (dalam bahasa Indonesia ditulis “standar” tanpa “d”) didefinisikan sebagai acuan spesifikasi yang diterima secara luas oleh industri atau yang disetujui organisasi standar resmi, misalnya ISO (International Organization for Standardization) dan IEC (International Electrotechnical Commission). Open Standard (Standar Terbuka) adalah acuan spesifikasi yang dapat dipelajari dan digunakan oleh siapa saja, tanpa biaya royalti dan tidak bergantung kepada pihak tertentu. Salah satu contoh Open Standard adalah acuan teknis untuk jaringan internet atau protokol internet *IP* (*Internet Protocol*), yang sering ditulis lengkap sebagai *TCP/IP* (*Transmission Control Protocol/Internet Protocol*).

### 7.2.1 Open Standard untuk Internet

Aturan atau acuan spesifikasi TCP/IP antara lain mencakup cara pemberian alamat jaringan (IP Addressing) untuk jaringan terbuka (publik) maupun tertutup (privat). Jaringan terbuka (Open Network) mencakup jaringan internet yang siapa saja dapat terhubung ke dalamnya.



Gambar 7.1 Arsitektur TCP/IP (Wikipedia)

Jaringan tertutup (Private Network) mencakup jaringan perusahaan/organisasi atau personal yang tidak mengizinkan pihak lain terhubung ke dalamnya. Jaringan privat yang menggunakan jaringan publik disebut Virtual Private Network (VPN).

Apa hebatnya karya Openness TCP/IP yang disusun oleh IETF (Internet Engineering Task Force) dari komunitas internet internasional ISOC (Internet Society) ini? Standar terbuka internet ini membuat jaringan internet cepat berkembang dan menyebar luas ke seluruh wilayah dunia, termasuk angkasa luar. Teknologi internet dapat dipelajari dan digunakan oleh siapa saja dan untuk apa saja, tanpa harus membayar izin atau royalti kepada para penemunya.

Berikut ini beberapa dari banyak standar terbuka yang masih terkait internet:

- HTTP (Hypertext Transfer Protocol), yang menjadi standar penyediaan halaman web atau WWW (World Wide Web) yang disusun oleh W3C (World Wide Web Consortium) dan organisasi internasional di bidang standardisasi atau ISO. Contoh produk Open Source yang menggunakan standar ini adalah server web Apache.



Gambar 7.2 Penulisan alamat http di browser web (Wikipedia)

- HTML (Hypertext Markup Language), yaitu standar terbuka pemformatan halaman web, sehingga disebut juga salah satu Open Format. Contoh produk Open Source yang terkait ini adalah browser web Firefox.
- SMTP (Simple Mail Transfer Protocol), yaitu aturan untuk pengiriman email dari program email pengirim sampai ke alamat email penerima (yang disimpan dalam server email). Contoh produk Open Source yang terkait ini adalah server email Postfix.
- POP/IMAP (Post Office Protocol / Internet Message Access Protocol), yaitu aturan untuk pengambil email dari server email. Contoh produk Open Source yang terkait ini adalah program email Thunderbird.

Open Standard terkait internet seperti TCP/IP, HTTP, HTML, dan lain-lain ini mirip dengan Open Source Software, yakni tidak perlu biaya royalti untuk menggunakan dan mengembangkan lebih lanjut. Beberapa standar internet yang terbuka itu membuat produk-produk Open Source seperti Linux, Apache, Firefox, dan LibreOffice menjadi sangat cepat berkembang pula.

### 7.2.2 Open Format

Standar terbuka untuk pemformatan file atau dokumen disebut Open Format. Selain format halaman web HTML, berikut ini beberapa format terbuka yang telah digunakan secara luas:

- OpenDocument Format (ODF), yaitu spesifikasi untuk pembuatan dan pengaksesan dokumen teks ODT (OpenDocument Text), spreadsheet ODS (OpenDocument Spreadsheet), presentasi ODP (OpenDocument Presentation), gambar ODG (OpenDocument Graphic), dan rumus ODF (OpenDocument Formula). Produk Open

Source yang terkait ini adalah LibreOffice, OpenOffice, Calligra Office, AbiWord, Gnumeric, dll. Standar internasional untuk ini adalah ISO/IEC 26300:2006, yang dijadikan standar nasional Indonesia SNI ISO/IEC 26300:2011.

- Portable Document Format (PDF), yaitu standar terbuka untuk pemformatan dokumen portabel yang mudah dilihat dan dicetak tanpa mengubah tampilan. Contoh produk Open Source terkait ini adalah Evince (Document Viewer) yang serupa dengan produk proprietary Adobe Acrobat Reader.
- Format file grafis: PNG, JPG, TIFF, SVG, GIF, dll. Contoh program: GIMP dan Inkscape.
- Format file audio/musik: OGG, MP3, AAC, dll. Contoh program: Audacity audio editor dan Audacious audio player.
- Format file video/film: OGV (OGG Video), MPG, MP4, dll. Contoh programnya OpenShot Video Editor dan VLC video player.

### 7.3 Open Content, Open Publication, Open Access

Open Content (konten terbuka) atau Open Publication (penerbitan terbuka) merupakan salah satu Openness yang juga membuat teknologi informasi berkembang cepat dan sistem ekonomi dunia juga berubah cepat, terutama karena berkembangnya internet. Banyak dokumen (teks, gambar, suara, video) yang dulu tertutup di tempat penyimpanan privat sekarang banyak terbuka untuk diakses oleh siapa saja dan dari mana saja. Open Content kadang disebut juga Open Access. Contoh aplikasi web yang menyediakan konten terbuka adalah Wikipedia.org dan Wikibooks.org. Banyak orang terlibat dalam pembuatan dan pengeditan halaman Wikipedia, sehingga kolaborasi komunitas Open Content sangat mirip dengan komunitas Open Source. Namun, tidak semua dokumen Open Access berlisensi Open Content, misal hanya dapat diakses untuk dibaca tapi tidak boleh dimodifikasi.



Gambar 7.3 Wikipedia.org penyedia konten terbuka

Open Content diterapkan dalam dokumen yang menyertai software, sehingga umumnya Open Source Software dilengkapi dokumen Open Content, misalnya FAQ (Frequently Asked Questions), HowTo, manual atau buku panduan penggunaan dan pengembangan software terkait. Open Content juga diterapkan untuk penerbitan buku atau karya penulisan lainnya, sehingga disebut juga Open Publication. Berikut ini beberapa contoh buku Open Content.

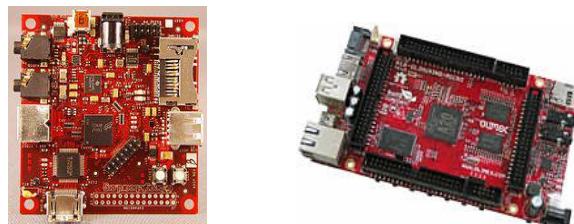
- Server Linux – Membangun Linux sebagai Internet/Intranet Server, ditulis oleh

Ahmad Sofyan (<http://sdn.vlsm.org/share/ServerLinux/>).

- Buku-buku panduan Linux BlankOn yang terdapat dalam CD/DVD BlankOn (<http://blankonlinux.or.id/dokumentasi/>).
- Beberapa referensi untuk buku ini, antara lain manual yang diterbitkan UNDP seperti “FOSS Open Standards” ([http://en.wikibooks.org/wiki/FOSS\\_Open\\_Standards](http://en.wikibooks.org/wiki/FOSS_Open_Standards)) dan “FOSS Open Content” ([http://en.wikibooks.org/wiki/FOSS\\_Open\\_Content](http://en.wikibooks.org/wiki/FOSS_Open_Content)).
- Buku acuan lain yang Open Content: “Linux Essentials” (<http://shop-download.linupfront.de/cc/lxes-en-manual-cc.pdf>) dan “Introduction to Free Software” ([http://ftacademy.org/files/materials/fta-m1-intro\\_to\\_FS-v1.pdf](http://ftacademy.org/files/materials/fta-m1-intro_to_FS-v1.pdf)).

## 7.4 Open Hardware

Open Hardware disebut lengkap sebagai Open-source Hardware, yang memiliki pengertian dasar seperti Open Source Software. Kriteria perangkat keras disebut Open-source Hardware didefinisikan oleh Open Hardware and Design Alliance, seperti Free Software, yakni: Bebas menggunakan perangkat keras untuk tujuan apa saja, bebas mempelajari cara kerjanya dan memodifikasinya, bebas menyebarkan perangkat atau membuat ulang dengan desain yang sama, dan bebas mengembangkan perangkat atau desain lebih lanjut serta merilis ke publik agar banyak orang yang dapat memperoleh manfaatnya. Contoh produk Open Hardware adalah Beagle Board dan Olinuxino, keduanya berupa komputer papan tunggal dengan prosesor Arm.



Gambar 7.4 Beagle Board (kiri) dan Olinuxino (kanan) (Wikipedia)

## 7.5 Open Movie

Open Movie adalah film yang disebarluaskan tanpa biaya lisensi. Pembuatan Open Movie secara bersama dengan berbagi teks, gambar, dan suara atau desain film secara keseluruhan. Contoh film yang tergolong Open Movie adalah Big Bug Bunny, Sintel, dan Tears of Steel (ketiganya merupakan film animasi yang dibuat dengan software Open Source Blender).



Gambar 7.5 Film Sintel (Wikipedia)

## Bab 8 Lisensi-lisensi Software Open Source

Tujuan Umum: Mampu memahami lisensi-lisensi Open Source untuk produk software.

Tujuan Khusus:

- a) Mampu menjelaskan pengertian lisensi Open Source utama atau popular yang sama menurut FSF dan OSI.
- b) Mampu menjelaskan pengertian beberapa lisensi yang berbeda menurut FS dan OSI.

### 8.1 Pengertian Hak Cipta, Lisensi, dan Kategori Software

#### 8.1.1 Hak Cipta dan Lisensi

UU No. 19 Tahun 2002 tentang Hak Cipta:

Pasal 1 ayat 1: Hak Cipta adalah hak eksklusif bagi Pencipta atau penerima hak untuk mengumumkan atau memperbanyak Ciptaannya **atau memberikan izin** untuk itu dengan tidak mengurangi pembatasan-pembatasan menurut peraturan perundang-undangan yang berlaku.

Pasal 2 ayat 1: Hak Cipta merupakan hak eksklusif bagi Pencipta atau Pemegang Hak Cipta untuk mengumumkan atau memperbanyak Ciptaannya, **yang timbul secara otomatis setelah suatu ciptaan dilahirkan** tanpa mengurangi pembatasan menurut peraturan perundang-undangan yang berlaku.

Pasal 3 ayat 2: Hak Cipta dapat beralih atau dialihkan, baik seluruhnya maupun sebagian ...dst.

Pasal 1 ayat 8: Program Komputer adalah sekumpulan instruksi yang diwujudkan dalam bentuk bahasa, kode, skema, ataupun bentuk lain, yang apabila digabungkan dengan media yang dapat dibaca dengan komputer akan mampu membuat komputer bekerja untuk melakukan fungsi-fungsi khusus atau untuk mencapai hasil yang khusus, termasuk persiapan dalam merancang instruksi-instruksi tersebut.

Pasal 1 ayat 14: Lisensi adalah izin yang diberikan oleh Pemegang Hak Cipta atau Pemegang Hak Terkait kepada pihak lain untuk mengumumkan dan/atau memperbanyak Ciptaannya atau produk Hak Terkaitnya dengan persyaratan tertentu.



Gambar 8.1 Copyright (Wikipedia)

Dalam penjelasan UU No. 19 Tahun 2002: Hak Cipta terdiri atas hak ekonomi (*economic rights*) dan hak moral (*moral rights*). Hak ekonomi adalah hak untuk mendapatkan manfaat ekonomi atas Ciptaan serta produk Hak Terkait. Hak moral adalah hak yang melekat pada diri

Pencipta atau Pelaku yang tidak dapat dihilangkan atau dihapus tanpa alasan apa pun, walaupun Hak Cipta atau Hak Terkait telah dialihkan.

Hak cipta software dimiliki oleh pengembang software secara perorangan atau kelompok, berlaku untuk software Open Source (tersedia kode sumber program) maupun tidak Open Source. Hak cipta suatu software melekat pada tiap pengembang software tersebut, namun secara ekonomi dapat ditentukan hanya dimiliki pihak tertentu (misal pihak yang membiayai atau membeli hak cipta secara ekonomi). Pemilik hak cipta suatu software (misal programmer) dapat mengalihkan hak ciptanya secara ekonomi kepada pihak lain (misal perusahaan tempat programmer bekerja atau pihak lain yang membeli hak cipta software tersebut untuk disebarluaskan secara gratis atau berbayar).

Jika suatu lisensi software isinya melarang penggunaan dan penyebarluasan tanpa izin, serta melarang pihak lain melihat kode sumber-nya (source code) untuk dipelajari/dikembangkan, maka software itu dipastikan tidak Open Source. Sebaliknya jika suatu lisensi software isinya mengizinkan pihak lain menggunakan dan menyebarluaskan, serta melihat kode sumber-nya untuk dipelajari/dikembangkan, maka software itu disebut Open Source. Ada banyak jenis lisensi software Open Source, karena ada perbedaan dalam memberikan batasan-batasan.

### 8.1.2 Katagori Software Berdasar Kelompok Lisensi

Moreno Muffatto membuat 5 katageori software yang dikelompokkan sesuai jenis lisensi software yang ada, yakni Public Domain, FOSS, Freeware, Shareware, dan Proprietary. Tabel 7.1 ini menggambarkan perbandingan masing-masing katagori dan contoh produknya.

**Tabel 8.1 Perbandingan Katagori Software Berdasar Kelompok Lisensi**

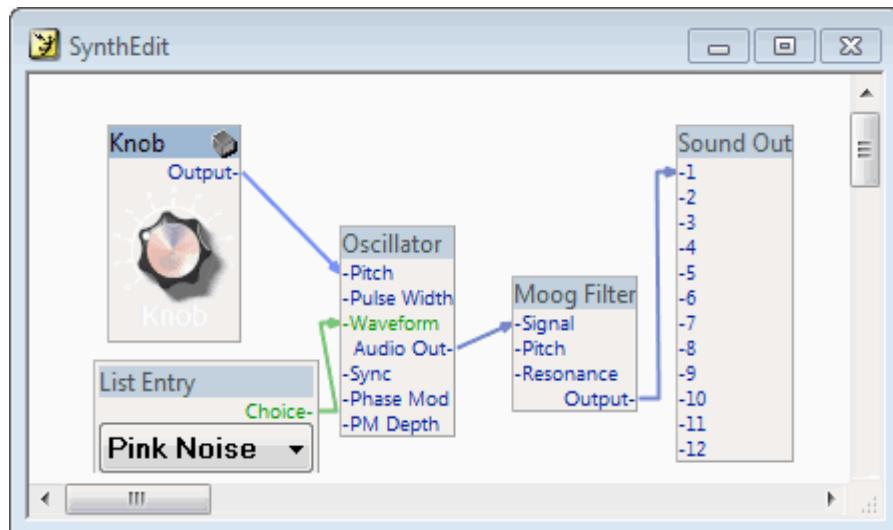
Katagori software	Public Domain	Free/Open Source	Freeware	Shareware	Proprietary
<b>Contoh produk</b>	HTML	Linux	Acrobat Reader	Apogee Software	MS Windows
<b>Tanpa biaya lisensi</b>	Ya	Ya	Ya	Terbatas	Tidak
<b>Bebas digunakan</b>	Ya	Ya	Tidak	Tidak	Tidak
<b>Tersedia kode sumber</b>	Tidak selalu	Ya	Tidak	Tidak	Tidak
<b>Bebas dimodifikasi</b>	Ya	Ya	Tidak	Tidak	Tidak
<b>Perlindungan hukum terhadap kebebasan</b>	Tidak	Ya	Tidak	Tidak	Tidak

Katagori software Public Domain berlaku untuk produk software yang sudah tidak terikat oleh pemilik hak cipta sehingga dapat digunakan dan dimodifikasi serta disebarluaskan tanpa batasan. Kode sumber tidak selalu tersedia, dan tidak ada biaya lisensi.

Free/Open Source Software (FOSS) berbeda dengan Public Domain karena pemilik hak cipta tetap memberikan batasan, misalnya pencantuman namanya, tidak mengubah lisensi, dsb. Kode sumber selalu tersedia (di internet atau di tempat yang ditunjukkan pemilik hak cipta),

dan tidak ada biaya lisensi. Hanya produk berlisensi FOSS yang memberi perlindungan hukum secara penuh terhadap kemerdekaan menggunakan dan mengembangkan software.

Freeware adalah software gratis, dalam pengertian tidak ada biaya lisensi, namun tidak menyediakan kode sumber, dan tidak membebaskan penggunaan untuk apa saja (misal hanya untuk non komersial). Freeware berbeda dengan Public Domain karena masih ada batasan dari pemilik hak cipta. Freeware berbeda dengan Free/Open Source karena tidak tersedia kode sumber dan tidak bebas dimodifikasi.



Gambar 8.2 SynthEdit aplikasi berlisensi shareware (Wikipedia)

Shareware seperti software gratis Freeware, tapi hanya untuk periode tertentu, misal gratis (tidak ada biaya lisensi) digunakan selama satu bulan. Setelah periode gratis habis, pengguna harus membayar lisensi atau mengunduh kembali versi lengkap atau lanjutannya. Apogee itu pembuat game video yang hanya memberikan gratis (tidak ada biaya lisensi) untuk sebagian produknya, tapi meminta bayaran lisensi untuk sebagian produk yang lain.

Proprietary software adalah umumnya produk yang pemilik hak ciptanya membatasi penggunaan, pemodifikasian, dan penyebarluasan karya software-nya, dengan cara menarik biaya lisensi. Kode sumber proprietary software tidak disediakan oleh pemilik hak cipta. Beberapa produk Proprietary software juga menyediakan versi gratis dalam waktu terbatas atau Shareware, biasanya untuk promosi atau demo sehingga disebut juga versi demo.

## 8.2 Siapa yang Mengakui Lisensi Free/Open Source Software

Ada dua organisasi terkenal yang mendefinisikan, menerima pendaftaran, mengakui atau mensertifikasi lisensi software Open Source, yakni Free Software Foundation (FSF – [www.gnu.org/licenses/licenses.html](http://www.gnu.org/licenses/licenses.html)) dan Open Source Initiative (OSI – [www.opensource.org/licenses](http://www.opensource.org/licenses)). FSF dan OSI mengakui dan mensertifikasi lisensi yang diajukan pihak sebagai lisensi Open Source setelah melalui beberapa tahap. Tahap-tahap sertifikasi dilakukan sesuai permintaan. Jika suatu lisensi baru saja diakui sebagai Open Source maka ia akan masuk daftar lisensi Open Source yang dipelihara oleh FSF atau OSI.

Meskipun berbeda organisasi dan cara mengakui lisensi, baik FSF maupun OSI sepakat satu sama lain bahwa pada banyak kasus lisensi-lisensi yang diakui mereka dapat diklasifikasikan sebagai lisensi FOSS (Free/Open Source Software), yang jumlahnya lebih dari 50 jenis. Berikut ini contoh 6 lisensi yang tidak diakui salah satu pihak (FSF atau OSI) dan sebaliknya.

- Apple Public Source License version 1.x (diakui OSI saja)
- Artistic License 1.0 (diakui FSF saja)
- Original BSD License (diakui FSF saja)
- Netscape Public License (diakui FSF saja)
- Reciprocal Public License 1.5 (diakui OSI saja)
- Sybase Open Watcon Public License 1.1 (diakui OSI saja)

### 8.3 Perbandingan Lisensi FOSS

Jumlah lisensi yang sama-sama diakui FSF dan OSI lebih dari lima puluh jenis, namun sebagian besar mirip. Untuk mengetahui apa saja perbedaannya, kita lihat Tabel 7.2 dan 7.3 yang memberikan contoh perbandingan enam lisensi FOSS terkenal. (N = Tidak, Y = Ya).

**Tabel 8.2 Perbandingan Lisensi FOSS dari sisi produk awal (Program Asli)**

<b>Program Asli</b>				
	Apakah diharuskan kode sumber diberikan ketika perangkat lunak disebarluaskan kembali?	Apakah Anda setuju bila perangkat lunak Anda menjadi bagian-lisensi dari lisensi-lisensi?	Apakah diizinkan orang lain untuk memakai paten dalam perangkat lunak program anda?	Jika program yang disebarluaskan tidak termasuk kode sumber dan ketersediaan kode sumber dapat dilakukan lewat cara lain, apakah Anda mengizinkan bayaran yang ditagih lebih tinggi dari biaya kirim kode sumber?
MIT	N	Y	N	Y
BSD	N	N	N	Y
Apache 2.0	N	Y	Y	Y
MPL 1.1	Y	Y	Y	Y
LGPL 2.1	Y	N	N	N
GPL 2.0	Y	N	N	N

**Tabel 8.3 Perbandingan Lisensi FOSS dari sisi hasil modifikasi (Program Turunan)**

<b>Program Turunan</b>				
	Apakah Anda mengizinkan program turunan anda memakai lisensi lain	Lanjutan pertanyaan sebelumnya dalam kondisi seperti apakah anda mengizinkan program turunan memakai lisensinya sendiri? 0. Lisensi turunannya harus sama	Apakah Anda mengizinkan orang lain untuk memakai paten dalam perangkat	Jika program yang disebarluaskan tidak termasuk kode sumber dan ketersediaan

	dalam kondisi yang berbeda?	dengan lisensi aslinya. 1. Lisensinya dapat memilih lisensi apapun yang disukai pengembang, meskipun lisensinya tidak bebas. 2. Lisensi turunan hanya boleh memakai lisensi yang disebut oleh lisensi asal. 3. Para pemodifikasi boleh menentukan mana yang mereka pakai selama tidak kontradiktif dengan lisensi asalnya.	lunak program Anda?	kode sumber dapat dilakukan lewat cara lain, apakah anda mengizinkan bayaran yang ditagih lebih tinggi dari biaya kirim kode sumber?
MIT	Y	1	N	N
BSD	Y	1	N	N
Apache 2.0	Y	3	N	Y
MPL 1.1	Y	2	Y	Y
LGPL 2.1	N	0	Y	Y
GPL 2.0	N	0	Y	Y

Semua lisensi yang dibahas di tabel di atas per definisi mengandung kesamaan fitur:

- Kode Sumber dari program asal harus terbuka.
- Diizinkan membuat kopi dari program asal.
- Penyebarluasan dari program asal dimungkinkan dan pemberitahuan hak cipta harus disertakan pada semua kopi.
- Lisensi bersifat umum, global, bebas royalti, untuk tujuan apa pun.
- Tidak ada garansi. Garansi dapat diberikan oleh pihak lain, biasanya secara berbayar dalam bentuk dukungan teknis (*technical support*).

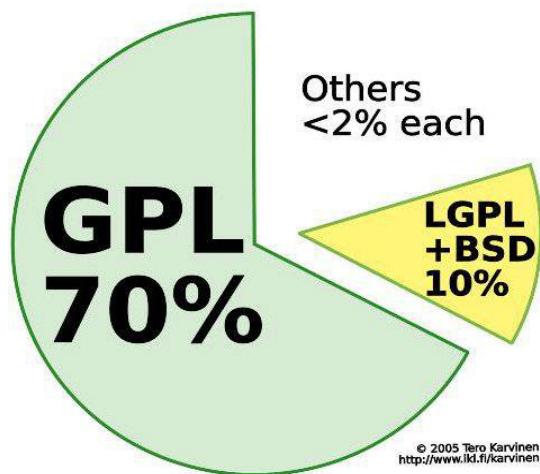
Meskipun semua hak harus diberikan kepada pemakai, lisensi-lisensi FOSS ini berbeda satu sama lain dalam level pelaksanaan. Contoh, meskipun ketersediaan kode sumber merupakan prasyarat bagi semua lisensi FOSS, program turunan berbeda-beda antara satu lisensi dengan lisensi lainnya dalam hal menyediakan kode sumber.

Misalnya pada saat mendistribusikan program modifikasi dari sebuah program dengan lisensi model BSD, kita tidak diwajibkan menyertakan kode sumber.

Meskipun yang menyebarkan kembali atau program turunannya diwajibkan untuk menyertakan kode sumber, cara penyertaan dan pembayaran penyebaran berlainan antar lisensi. GPL dan LGPL misalnya mengatur sangat detail tentang harga yang bisa dikenakan pada proses penyebaran. Ini karena GPL dan LGPL menginginkan proses distribusi masih sejalan dengan semangat perangkat lunak bebas. Anda dapat menghargai distribusi Anda

berapapun sepanjang di dalamnya sudah mengandung kode sumber. Tetapi bila mana Anda tidak memaket kode sumber dengan paket distribusi, maka bila ada yang meminta kode sumber harganya tidak boleh lebih mahal dari ongkos kirim.

Tentang produk turunan terdapat perbedaan antar lisensi lebih besar lagi. Meskipun kode sumber merupakan prasyarat produk pertama, tidak demikian untuk produk turunan. Meskipun produk turunan harus membuka kode sumber tapi bisa jadi tidak memakai lisensi yang sama, hingga produk turunannya lagi bisa jadi tidak menyertakan kode sumber. Misal GPL mensyaratkan produk turunannya harus memakai lisensi GPL juga, tapi lisensi BSD tidak begitu, produk turunan BSD bisa memakai lisensi lain atau bahkan tidak menyertakan kode sumber.



Gambar 8.3 Prosentasi penggunaan lisensi menurut freshmeat.net (<http://terokarvinen.com>)

Kemungkinan untuk menggabungkan lisensi FOSS dengan program proprietary juga tidak sama. Ini biasanya terjadi pada proyek yang besar, hampir tidak bisa dihindari ada interaksi dengan produk proprietary. Misalnya apabila ada sebuah proyek yang menggunakan program A yang GPL, program B yang BSD, dan C yang proprietary, ketiganya kita memiliki kode sumber. Begitu digabung dan menjadi program ABC maka program ABC harus berlisensi GPL karena ada komponen yang berasal dari program A. Bila mana program ABC ingin dijadikan proprietary maka satu-satunya cara mencari alternative program A yang berlisensi model BSD karena BSD tidak mengharuskan program turunannya berlisensi BSD juga. Atau cari alternative program A yang proprietary juga.

Pada contoh di atas GPL disebut sebagai 'viral efek' dan sangat tidak ramah dengan program proprietary dalam pengembangannya. Oleh karena itu diterbitkan LGPL yang lebih ramah dan untuk mendorong pemakaian pustaka bebas secara lebih luas.

Tiga jenis umum lisensi FOSS yaitu GPL, LGPL dan BSD dibahas secara detail berikut ini:

#### 8.4 GNU General Public License (GNU GPL atau GPL)

GNU General Public License (GNU GPL atau GPL) adalah lisensi klasik perangkat lunak bebas. GPL juga merupakan yang paling terkenal dan banyak digunakan di antara lisensi FOSS lainnya. GPL adalah temuan yang dihadirkan untuk memenuhi konsep kebebasan FSF. Dia adalah lisensi sekaligus dokumen yang memanifestasikan ide dasar dari perangkat lunak bebas.

#### 8.4.1 Copyleft

Cara yang dipakai GPL untuk menjamin kebebasan ini biasa dikenal sebagai “copyleft”. Ketika perusahaan proprietary biasanya memakai “Copyright, All Rights Reserved” maka FSF juga memakainya sebagai “Copyleft, All Rights Reserved”.

Copyleft mencegah perangkat lunak bebas diubah menjadi perangkat lunak proprietary. Ia menggunakan hukum hak cipta tapi dengan isi berubah 180 derajat dari yang biasanya. Tidak seperti biasanya copyright digunakan untuk memprivatisasi software, copyleft dapat dipakai untuk menjaga agar perangkat lunak tetap bebas menjadi milik masyarakat.



Gambar 8.4 Copyleft (Wikipedia)

Tidak seperti *public domain*, setiap orang tetap dapat menggunakan karya GPL atau copyleft dengan tetap menjaga karya itu sebagai karya yang di-copyright-kan. Pada dasarnya pemegang copyleft tetap terikat secara hukum. Bilamana karya GPL dipakai dan dilanggar maka yang melanggar lisensi GPL dapat dituntut secara hukum.

Anda sebagai pembuat karya cipta yang ingin software Anda bebas, tidak cukup hanya menyatakan Anda sebagai pemegang hak cipta dan merilis software anda sebagai *public domain*, karena akan memungkinkan karya Anda itu diperlakukan oleh orang lain. Anda harus menyatakan dalam bentuk lisensi yang mengatur bagaimana orang menggunakan karya Anda. Dengan melisensikan karya anda sebagai GPL, Anda mengizinkan pengguna memiliki hak yang diizinkan oleh gerakan free software, dan meminta pengguna menyetujui perjanjian untuk menjaga software dan karya turunannya tetap bebas sesuai dengan keinginan Anda.

#### 8.4.2 Syarat dan Ketentuan Utama GPL

##### a. Kebebasan Pengguna

Ketika Anda memilih program berlisensi GPL, selain memiliki akses ke kode sumber, Anda juga bebas bebas untuk:

- Menggunakan atau menjalankan program.
- Memperbanyak atau menggandakan program.
- Menyebarluaskan program, bahkan untuk tujuan komersial sekalipun, tapi pemberian hak cipta dan pernyataan jaminan harus disertakan. Penyebaran dalam bentuk kode objek diperbolehkan selama kode sumber tersedia untuk semua penerima atau pengguna.

- Program turunan harus sama berlisensi GPL demikian juga bila memakai sumber pihak ketiga maka keseluruhan pada produk turunannya juga harus di bawah lisensi GPL.

#### **b. Tanpa Jaminan**

Meskipun produk turunan bisa dikomersialisasikan, lisensinya sendiri tidak boleh dihargai/dijual. Karena lisensi tidak dijual maka tidak ada garansi pemberi lisensi terhadap pengguna perangkat lunak GPL. Distributor atau perusahaan jasa Open Source boleh menjual layanan untuk memberi garansi atau menyediakan dukungan teknis.

#### **c. Lisensi Dikeluarkan oleh Pencipta**

Lisensi GPL tidak dapat di-sub-lisensikan. Ketika program disebarluaskan, penerima program memperoleh lisensi secara langsung dari pencipta atau pemilik pertama. Pendistribusi tidak boleh membatasi atau menghalangi hak yang sudah diberikan GPL.

#### **d. Penerimaan dan Pembatalan**

Dengan memodifikasi dan menyebarkan program GPL, seseorang dianggap menerima lisensi GPL sebagai lisensi produknya. Ketika seseorang melanggar lisensi GPL maka semua hak yang diberikan oleh GPL batal demi hukum, tetapi siapapun yang sudah menerima distribusi dari orang yang sudah dibatalkan haknya tidak serta-merta kehilangan haknya karena lisensi diberikan dari penulis bukan penyebar (distributor), sepanjang tidak melanggar ketentuan GPL.

#### **e. Keterkaitan dengan Aturan Hukum lainnya**

GPL tidak berlaku pada kondisi yang tidak sesuai dengan yang disebutkan di dalamnya. Program GPL tidak boleh dipakai pada program dengan lisensi yang bukan GPL, bila terjadi yang bersangkutan tidak boleh menyebarkan produk turunannya tersebut. Sebuah program GPL tidak boleh menjadi bagian dari program proprietary ataupun berhubungan dengan pustaka proprietary.

Teks lengkap GPL terbaru tersedia di <http://www.gnu.org/licenses/gpl.html>

FSF juga mengelola halaman FAQ tentang GPL yang dapat diakses pada <http://www.fsf.org/licensing/licenses/gpl-faq.html>

### **8.5 GNU Lesser General Public License (GNU LGPL atau LGPL)**

Selain GPL, proyek GNU menawarkan jenis lain dari copyleft untuk pustaka program (*libraries*), LGPL. Pustaka program yang berlisensi LGPL diizinkan berhubungan (*link*) dengan program berlisensi proprietary atau selain GPL/LGPL. Kode sumber program proprietary dihubungkan (*link*) dengan kode sumber pustaka (*library*) pada saat kompilasi bersama sebagai static-library, atau sebagai dinamic-library. Binary program hasil kompilasi itu menjadi proprietary, meskipun menggunakan library Open Source, dengan syarat kode sumber pustaka program tetap disertakan.



Gambar 8.5 LGPL (Wikipedia)

## **8.6 Model Lisensi BSD (Berkeley Software Distribution)**

Lisensi *Berkeley Software Distribution* (BSD) pertama kali dipakai oleh *Berkeley System Distribution*, sebuah sistem UNIX yang dibuat oleh University of California di Berkeley. Mudah sekali untuk menggunakan BSD untuk membuat lisensi sendiri dengan mengubah pemilik organisasi untuk ditampilkan pada pemberitahuan hak cipta dalam lisensinya. Tidak seperti copyleft, BSD hanya memiliki batasan yang berhubungan dengan pemakaian.

Berikut ini tiga kebebasan pemakai program berlisensi BSD.

- Memperbanyak program dan menyebarluaskan program baik berbentuk kode sumber atau kode biner. Penyebarluasan tidak dituntut menyertakan kode sumber.
- Membuat karya turunan dan mendistribusikannya dalam bentuk kode sumber atau biner. Pembuat perubahan bebas memilih lisensi, FOSS atau proprietary, sehingga karya turunan software berlisensi BSD belum tentu FOSS, misal FreeBSD diturunkan menjadi Mac OS X.
- Kode sumber berlisensi BSD dapat dimasukkan dalam program proprietary.

## **8.7 Lisensi Gabungan (*Dual/Multiple Licensing*)**

Program komputer dapat memiliki lebih dari satu lisensi. Lisensi adalah pilihan Anda pemilik hak cipta, tergantung bagaimana Anda memandang hubungan antara karya Anda dengan para pengguna atau pengembang lainnya. Bisa jadi ada lebih dari satu jenis pemakai dengan lebih dari satu hubungan, misal untuk komunitas umum dan untuk perusahaan yang butuh layanan khusus. Untuk situasi seperti ini Anda dapat memilih lisensi yang berbeda-beda sesuai dengan situasinya.

Contoh: MySQL dilisensikan oleh ORACLE sebagai GPL untuk umum dan non GPL atau proprietary (disebut juga lisensi komersial) untuk perusahaan yang membutuhkan layanan khusus dari ORACLE.

## Bab 9 Lisensi-lisensi Open untuk Bukan Software

Tujuan Umum: Mampu memahami lisensi-lisensi Open Source untuk produk bukan software.

Tujuan Khusus:

- a) Menjelaskan salah satu lisensi GNU untuk dokumen atau bukan software.
- b) Menjelaskan 6 jenis lisensi CC untuk bukan software.

### 9.1 GNU Free Documentation License (GNU FDL atau FDL)

Dokumentasi dan manual yang baik amat penting untuk FOSS. Apabila dokumen ini tidak memiliki lisensi yang sama dengan FOSS, dampaknya akan sangat sulit untuk bisa memperoleh manfaat penuh dari program FOSS. Meski biasanya digunakan sebagai lisensi untuk perangkat lunak, GNU GPL juga bisa dipakai untuk hal yang bukan perangkat lunak, sepanjang definisi perangkat lunak dijelaskan.



Gambar 9.1 GNU Free Document License (Wikipedia)

FSF (Free Software Foundation) juga menyediakan lisensi khusus untuk dokumentasi, yakni GNU Free Document License (<http://www.gnu.org/licenses/fdl.html>) GNU FDL atau FDL adalah bentuk lain copyleft untuk manual, book teks dan dokumen lainnya yang menjamin pemakai mendapat kebebasan memperbanyak, menyebarluaskan dengan modifikasi atau tanpa modifikasi, untuk kepentingan komersial maupun non-komersial.

Dengan menggunakan FDL pada dokumen seperti pada GPL, Anda sebagai penulis memberikan hak untuk menggandakan dan menyebarluaskan. Produk turunannya pun harus berlisensi FDL, bedanya pada segi jumlah penggandaan ada aturan di FDL.

Persamaan bentuk lisensi produk GPL (misal software) dan FDL (misal buku):

- Pertama: Produk software GPL bebas digunakan, buku FDL bebas dibaca, untuk kepentingan apa saja, termasuk untuk komersial.
- Kedua: Produk software GPL bebas dimodifikasi, buku FDL juga bebas dimodifikasi/diedit, untuk kepentingan apa saja, termasuk untuk komersial, dengan tetap menyebutkan nama pembuat sebelumnya, dan tidak mengubah lisensinya, sehingga lisensi hasil modifikasi harus juga GPL atau FDL.
- Ketiga: Produk software GPL dan buku FDL bebas dicopy/disebarluaskan, untuk kepentingan apa saja, termasuk untuk komersial.
- Keempat: Produk hasil modifikasi software GPL dan buku hasil modifikasi buku FDL bebas disebarluaskan, untuk kepentingan apa saja, termasuk untuk komersial.

### 9.2 Creative Commons Licenses

Terinspirasi oleh perkembangan FOSS, *Creative Commons Cooperation* menganjurkan keterbukaan dari konten digital. Ini meningkatkan interkoneksi dan operasi antar konten

digital sehingga tidak terikat oleh aturan dari para pengusaha besar yang menentukan standar. Konten digital yang dimaksud dapat berupa karya tulisan, gambar, video, dan sebagainya.

Di tahun 2002 versi pertama dari lisensi *Creative Commons Public Licenses* (lisensi CC) dirilis. Ada sebelas jenis lisensi yang mengidentifikasi perhatian penting utama dari para penulis, di antaranya:

- *Attribution (BY)*: Untuk semua penggandaan dan penyebaran, menyebutkan pengarang asli diperlukan.
- *Non Commercial (NC)*: Karya ini tidak dapat dikomersialisasikan.
- *Non Derivative Works (ND)*: Karya tidak boleh diubah.
- *Share Alike (SA)*: Diizinkan mengubah, memodifikasi dan membuat produk turunan, dengan syarat lisensi yang dipakai pada karya turunan sama dengan lisensi ini.

Sebelas lisensi itu adalah kombinasi unik ke empat lisensi di atas. Para penulis berhak untuk memilih lisensi mana yang paling sesuai dengan karya mereka.

Pada 2004, Creative Commons meluncurkan versi kedua dari lisensi CC, karena kebutuhan dan pemakaian Attribution begitu luas maka Attribution menjadi default, yang berakibat hanya tinggal 6 lisensi yang tersisa (<http://creativecommons.org/licenses/>).

Icon	Description	Acronym
	Attribution alone	BY
	Attribution + NoDerivatives	BY-ND
	Attribution + ShareAlike	BY-SA
	Attribution + Noncommercial	BY-NC
	Attribution + Noncommercial + NoDerivatives	BY-NC-ND
	Attribution + Noncommercial + ShareAlike	BY-NC-SA

Gambar 9.2 Beberapa Jenis Lisensi CC (Wikipedia)

Lisensi CC digunakan dan didesain untuk semua jenis konten digital kecuali perangkat lunak, termasuk karya seni, fotografi, karya musik, dan tulisan. Tapi lisensi CC tidak berhubungan dengan hal yang terkait Open Source karena semua yang dimasukkan dalam CC sudah transparan dan tidak dapat dikompilasi. Beberapa lisensi CC tidak membolehkan modifikasi tapi dalam kontek CC tidak disebut tidak bebas. Lisensi CC sangat bermanfaat dalam penyaluran ide kebebasan dan keterbukaan tapi dalam dunia pengembangan perangkat lunak konsep ini tidak pas.

**Tabel 8.1 Daftar lisensi open source untuk karya yang bukan software**

	Nama penulis asli harus disebutkan?	Boleh untuk kepentingan komersial?	Boleh dibuat turunan (boleh diubah)?	Karya turunan harus mamakai lisensi yang sama seperti karya aslinya?
CC BY	Yes	Yes	Yes	No

CC BY-NC	Yes	No	Yes	No
CC BY-NC-ND	Yes	No	No	No
CC BY-NC-SA	Yes	No	Yes	Yes
CC BY-ND	Yes	Yes	No	No
<b>CC BY-SA</b>	Yes	Yes	Yes	Yes
CC NC	No	No	Yes	No
CC NC-ND	No	No	No	No
CC NC-SA	No	No	Yes	Yes
CC ND	No	Yes	No	No
CC SA	No	Yes	Yes	Yes
<b>GNU FDL</b>	Yes	Yes	Yes	Yes

# Bab 10 Aspek Keamanan Open Source

Tujuan Umum: Mampu memahami isu-isu keamanan terkait Open Source.

Tujuan Khusus:

- a) Mampu menjelaskan isu-isu keamanan produk-produk Open Source, antara lain Open Source tidak otomatis membuat produk tidak aman.
- b) Mampu menjelaskan mengapa software Open Source relatif lebih aman terhadap virus dan worm.

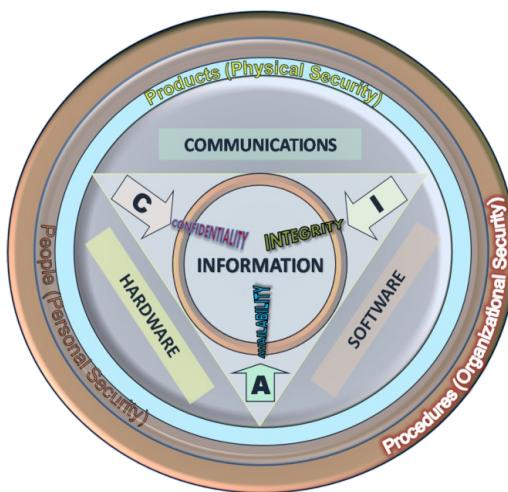
## 10.1 Pendahuluan

Salah satu syarat produk Free/Open Source Software (FOSS) adalah ketersediaan kode sumber (source code) program yang disebarluaskan melalui internet atau cara pendistribusian lainnya. Ketersediaan kode sumber ini tidak berarti membuat produk FOSS tidak aman, bahkan sebaliknya ketersediaan kode sumber memungkinkan banyak orang dapat memperbaiki jika ditemukan kelemahannya. Banyak faktor yang menentukan keamanan suatu produk software, selain tergantung produk software itu sendiri juga tergantung orang yang menggunakan dan merawat, serta prosedur keamanan yang harus dijalankan pengguna.

## 10.2 Keamanan Komputer

Keamanan komputer (*computer security*), sering disebut *IT (information technology) security* atau *cybersecurity*, merupakan penerapan keamanan informasi (*information security*) pada komputer dan jaringan komputer. Keamanan informasi yang terkait komputer ini mencakup perangkat keras (*hardware*), perangkat lunak (*software*), dan komunikasi (*communications*).

Keamanan informasi dibangun untuk menjaga kerahasiaan (*confidentiality*) termasuk privasi orang, keutuhan (*integrity*) informasi, dan ketersediaan (*availability*) informasi, serta pengendalian (*control*) dan evaluasi (*audit*) untuk pengembangan lebih lanjut dan manfaat audit lainnya.



Gambar 10.1 Diagram Keamanan Informasi (Wikipedia)

Keamanan informasi tidak hanya mencakup teknologi hardware, software, dan komunikasi yang merupakan keamanan produk atau *Physical Security*, namun juga terkait dengan orang atau *Personal Security*, dan prosedur/pedoman/aturan atau *Organizational Security*.

Keamanan Open Source yang dibahas dalam bab ini hanya fokus keamanan produk atau keamanan fisik software, namun tidak dapat terlepas dari orang yang menggunakan dan merawat (keamanan personal), serta prosedur yang dijalankan (keamanan organisasi).

### 10.3 Bentuk, Cara, dan Target Gangguan Keamanan Komputer

Komputer atau IT di perusahaan, pemerintahan, lembaga pendidikan, atau bahkan komputer personal harus dijaga keamanannya, karena ada banyak pengganggu atau perusak komputer, antara lain berbentuk program atau orang (langsung). Secara umum, dua contoh ini tidak tergantung produk targetnya itu tersedia kode sumbernya (Open Source) atau tidak (Proprietary).

- Program: Virus, worm, malware, atau yang sejenis, yakni program yang dibuat untuk mengganggu atau merusak. Program disebut virus atau worm karena bisa menular atau menyebar dari satu komputer ke komputer lain.
- Penyerang: *Attacker* atau *cracker* (hacker jahat), yakni orang yang mencari kelemahan suatu sistem IT atau komputer, lalu memanfaatkan kelemahan itu untuk menyerang atau merusak. Misal cracker yang mengubah tampilan web (*defacing*), mengambil informasi rahasia, atau menghapus data sehingga sistem informasi atau komputer tidak bekerja dengan baik.

Berikut ini beberapa contoh cara dan target dari orang penyerang/perusak (*attacker/cracker*) yang memanfaatkan ketidakamanan komputer. Secara umum, contoh-contoh ini juga tidak tergantung targetnya menggunakan Open Source atau Closed Source, karena kelemahan program dapat terjadi pada produk Open Source maupun Proprietary.

- *Defacing* (pengrusakan tampilan). Misal sebuah web portal yang dibangun dengan program sistem manajemen konten. Ketika ada ahli keamanan yang menemukan kelemahan pada program tersebut, misal tampilan dapat diubah oleh orang yang tidak berhak, maka orang yang ingin mengrusak (*deface*) dapat memanfaatkan kelemahan tersebut.
- *Planting malware* (pemasangan program jahat atau berbahaya, *malicious software*) atau *Trojan*. Orang yang tidak berhak dapat memasuki komputer orang lain untuk memasang program jahat, atau mengunggah program jahat, dengan memanfaatkan kelemahan sistem komputer. Program jahat itu kemudian dijalankan untuk mengganggu keamanan.
- *Phishing* (penipuan), berasal dari kata *fishing* (memancing), yakni suatu cara untuk mendapatkan nama dan password, misalnya untuk akses ke suatu program di internet, kartu kredit, dan lain-lain yang dapat merugikan pengguna (yang terpancing) atau sistem.
- *Hijacking*, yaitu mengambil alih (membajak) akses pengguna pada proses komunikasi untuk tujuan jahat. Ini mirip membajak pesawat terbang untuk diambil alih oleh penjahat.
- *Sniffing*: pembacaan data yang bukan tujuan. Standarnya hanya komputer dengan alamat yang bersesuaian dengan alamat tujuanlah yang akan mengambil data tersebut.
- Tetapi pada saat snif, komputer dengan alamat bukan alamat tujuan tetap mengambil

data tersebut. Dengan adanya sniffer ini, maka usaha untuk melakukan kriptografi dalam database (dalam hal ini login user dan password) akan kurang berguna.

- *IP Spoofing*: pemalsuan alamat IP sehingga sasaran menganggap alamat IP attacker adalah alamat IP dari host di dalam network, bukan dari luar network. Misal attacker mempunyai IP address 12.34.xx.xx ketika attacker melakukan serangan jenis ini, maka network yang diserang akan menganggap IP attacker adalah bagian dari network yang diserang, misal 192.xx.xx.x.



Gambar 10.2 Teknik IP Spoofing (Sumber:www.termpedia.com)

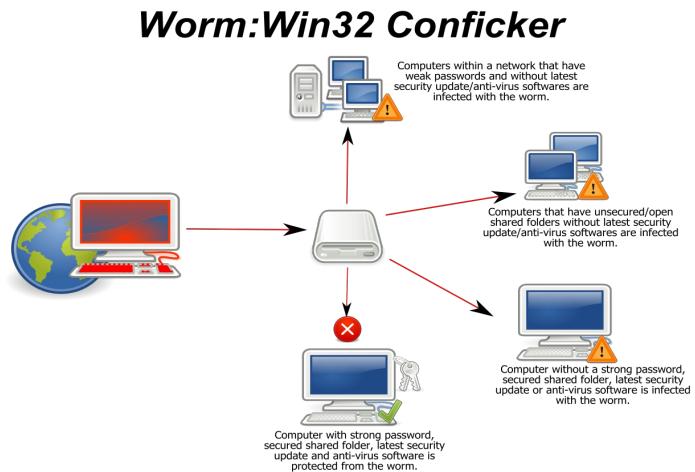
- *Exploit*: program yang khusus dibuat untuk menyerang keamanan komputer tertentu. Biasanya exploit digunakan untuk menguji keamanan atau menemukan kelemahan komputer, namun dapat digunakan untuk memasuki (penetration) ke sebuah komputer secara tidak legal.
- *Brute Force*: salah satu metode dalam penjebolan keamanan yang menggunakan password. Brute force adalah salah satu bagian dari *password guessing*, bedanya adalah waktu yang dipakai dalam brute force lebih singkat dari password guessing karena metode brute force menggunakan beberapa tool cracking untuk mendapatkan password.
- *Password Cracking*: metoda untuk melawan perlindungan password yang dienkripsi. Dengan anggapan bahwa attacker telah masuk ke sistem, ia bisa saja mengubah kekuasaannya dengan cara meng-crack password file menggunakan metode *brute-force dictionary attack* (mencocokan kata-kata yang berada dalam kamus dengan kata-kata yang dienkripsi dalam file password). Keberhasilan menggunakan cara ini bergantung pada kecepatan prosesor dan program yang dimiliki oleh attacker. Cara yang terbaik untuk menghindari serangan jenis ini adalah dengan memonitor kewenangan akses pada file.
- *DoS (Denial of Service)*: mengirim data yang sangat besar ke komputer atau jaringan yang diserang, atau sengaja menghabiskan sumber daya komputer yang terbatas, sehingga komputer atau sistem tidak dapat diakses oleh yang berhak.

## 10.4 Mengapa Open Source Relatif Lebih Aman?

Meskipun semua penjelasan sebelumnya menunjukkan bahwa keamanan komputer tidak

tergantung produk itu Open Source atau tidak, namun beberapa hal berikut ini menunjukkan produk Open Source relatif lebih aman dibandingkan produk Proprietary, dengan catatan semua orang yang terkait sudah mengikuti prosedur keamanan. Produk Open Source menjadi tidak aman jika tidak dijaga sesuai prosedur, sebaliknya produk Proprietary dapat menjadi lebih aman jika dijaga sesuai prosedur, meskipun dalam batas-batas tertentu pengguna tidak mampu melakukan penjagaan karena tidak tersedianya kode sumber.

- Dalam kasus virus/worm, sistem operasi Open Source seperti Linux relatif lebih aman karena sistem otentikasi yang lebih ketat dan sebagian besar virus/worm dibuat untuk selain Linux. Ini disebut relatif untuk kondisi sekarang, karena saat ini ada yang bukan Open Source seperti sistem operasi Windows dibuat menjadi lebih aman terhadap gangguan virus/worm.



Gambar 10.3 Conficker, worm yang menyerang Windows (Wikipedia)

- Ketersediaan kode program memungkinkan banyak orang membuat perbaikan (update) jika ditemukan kelemahan produk Open Source, sehingga ketersediaan update bisa lebih cepat. Sedangkan produk proprietary hanya dapat diperbaiki oleh pemilik hak cipta (pengembang yang memiliki kode program). Ini juga relatif atau tidak berlaku untuk semua produk, karena bisa jadi perbaikan keamanan produk Open Source tertentu menjadi lebih lambat daripada produk proprietary tertentu.

## 10.5 Enam Mitos (Salah Pengertian) Keamanan Open Source

Berikut ini tulisan Nick Heath yang mengutip pernyataan Dr. Ian Levy, penasihat pemerintah Inggris di bidang keamanan IT. Tulisan Nick dikutip beberapa website, seperti <http://www.tuxmachines.org/node/60151> dan

<http://lxer.com/module/newswire/view/184471/index.html> yang tulisan aslinya ada di <http://www.zdnet.com/six-open-source-security-myths-debunked-and-eight-real-challenges-to-consider-7000014225/>

1. *Open source software is more/less secure than proprietary.* Tidak ada data statistik yang menunjukkan perbedaan kelemahan produk-produk Open Source dan Proprietary, karena sangat tergantung pengembangannya dan perawatannya, atau tergantung orang dan prosedurnya.
2. *Many eyes makes for secure code.* Jumlah orang yang melihat kode program juga tidak menjamin keamanan, karena belum tentu orang yang melihat itu memahami audit keamanan.

3. *Bad people can look at the source code, so it's less secure.* Orang yang tidak baik bisa melihat kode sumber juga tidak otomatis membuat program itu tidak aman, karena keamanan tidak ditentukan siapa yang melihat, tapi siapa yang mengembangkan dan menggunakan.
4. *Anyone can contribute to the code so it's all bad.* Banyak orang berkontribusi tidak otomatis membuat kode buruk, karena produk yang sukses telah melalui seleksi siapa saja yang boleh dan tidak boleh memberi kontribusi terhadap kode sumber yang dirilis.
5. *Open source software means it's open for your organisation to use.* Tidak semua produk yang dinyatakan sebagai Open Source pasti berisi kode yang dapat diubah secara bebas. Karena beberapa program Open Source berisi kode yang tidak Open, misalnya mengandung kode yang dipatenkan sehingga penggunaannya terbatas.
6. *All software must be evaluated for security.* Hanya software tertentu atau untuk penggunaan tertentu yang membutuhkan evaluasi keamanan sebelum digunakan, sehingga tidak ada hubungan dengan Open atau tidak.

## 10.6 Beberapa Kesalahan dan Prosedur Keamanan yang Baik

Keamanan software ternyata sangat terkait dengan jaringan komputer, karena era internet menjadikan hampir semua komputer tersambung ke internet yang terbuka (Open Network). Software sebagai sebuah aplikasi jaringan atau sistem informasi harus diamankan mulai dari pada saat perencanaan pengembangan (desain software), pembuatan program aplikasi/database, pemasangan sistem operasi, pemasangan aplikasi/database pada sistem operasi, prosedur dan keahlian yang harus dimiliki orang yang menjaga keamanan (system/network administrator), dan pengguna/pengakses yang diberi hak akses dan kewajiban mengikuti prosedur untuk menjaga keamanan pada saat menggunakan/mengakses.

Berikut ini beberapa kesalahan yang biasa dilakukan oleh mulai dari pengguna biasa hingga administrator sistem/network, dan prosedur yang seharusnya dijalankan.

- Tidak memasang anti virus pada sistem yang menerima data dari luar. Anti virus harus dipasang pada server email dan server file karena dunia menjadi pusat pertukaran data, meskipun server-server itu menggunakan sistem operasi yang relatif aman terhadap virus, misal Linux. Jaringan dan/atau komputer pengguna yang rentan terhadap virus akan terkena dampak negatif server yang tidak dipasangi anti virus.
- Membuka email yang berisi data atau program berbahaya. Pastikan hanya email yang jelas siapa pengirimnya dibuka attachment-nya, atau diikuti perintahnya, karena bisa jadi email itu akan menularkan virus/worm atau memberikan akses ke program berbahaya. Email juga digunakan penjahat untuk melakukan penipuan (phishing), misal pengirim mengaku dari bank atau penyedia jasa berbasis internet, yang meminta penerima untuk memasukkan data seperti nama, password, tanggal lahir, nomor kartu kredit, dan sebagainya, sehingga pengirim mendapatkan data penting untuk disalahgunakan.
- Tidak mengikuti berita di web atau mailing list terkait keamanan produk yang digunakan, misalnya untuk produk server web Apache di [www.apache.org/security](http://www.apache.org/security) atau email [announce-subscribe@httpd.apache.org](mailto:announce-subscribe@httpd.apache.org).
- Tidak segera melakukan *security patch* atau update software terkait keamanan ketika suatu produk ditemukan kelemahan dan tersedia patch-nya.
- Tidak membuat dan menguji sistem cadangan data (*backup*).
- Menyambungkan sistem ke internet sebelum sistem itu dibuat sangat aman (*hardening*).

- Menyambungkan sistem ke internet dengan password bawaan atau mudah ditebak.
- Menjalankan program server (daemon) yang tidak aman (misal telnetd, harusnya sshd yang menggunakan enkripsi password), atau program yang tidak dibutuhkan sehingga membuka celah keamanan lebih banyak.
- Tidak segera mengupdate program atau sistem jika ditemukan lubang keamanan (*security hole*).
- Tidak menerapkan sistem *firewall* yang bagus untuk mencegah masuknya orang atau program jahat.
- Tidak menerapkan sistem pendeteksian intrusi (usaha pihak lain masuk ke sistem).
- Tidak melakukan update virus pada program antivirus secara otomatis.
- Tidak membuat panduan yang baik untuk diajarkan kepada pengguna dalam hal keamanan sistem yang dibuat atau dikelolanya.
- Memberikan password kepada pengguna melalui telepon atau saluran komunikasi yang kurang aman.

# Bab 11 Model Bisnis Open Source

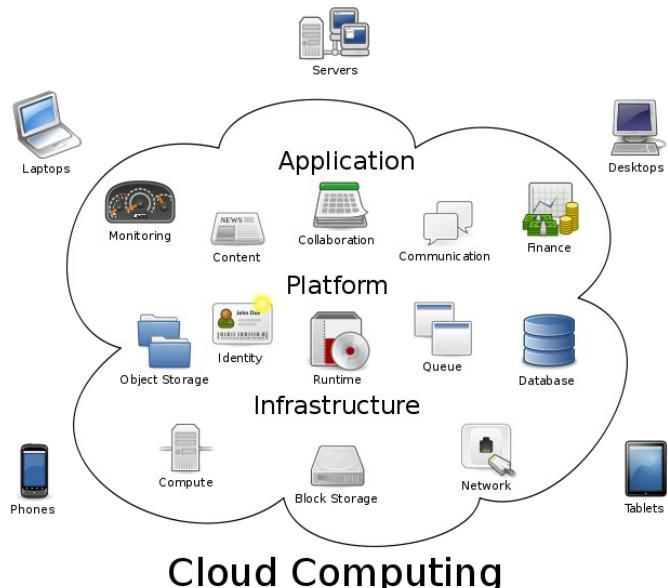
Tujuan Umum: Mampu memahami apa model bisnis software Proprietary dan Open Source.

Tujuan Khusus:

- Mampu menjelaskan beberapa model bisnis software Proprietary dan contoh produknya.
- Mampu menjelaskan beberapa model bisnis Open Source dan contoh produknya.

## 11.1 Model Bisnis Produk Proprietary

Umumnya pengembang (vendor) produk software Proprietary melakukan bisnis dengan menjual lisensi (surat izin menggunakan software). Misal lisensi MS Windows dijual seharga Rp 1 juta untuk dipasang pada sebuah komputer selamanya. Biaya lisensi itu tidak termasuk biaya support/garansi, biaya upgrade, biaya pelatihan, dan tidak boleh disewakan atau dijual kembali tanpa izin.



Gambar 11.1 Bisnis Cloud Computing (Wikipedia)

Bisnis produk proprietary lain yang masih terkait lisensi adalah menyewakan software (lisensi jangka waktu tertentu) seperti yang terjadi akhir-akhir ini dengan istilah komputasi awan (*Cloud Computing*) jenis SaaS (*Software as a Services*). Jasa yang tidak terkait langsung dengan lisensi antara lain menjual jasa instalasi, jasa technical support atau maintenance (kadang disebut biaya garansi), jasa pelatihan, biaya upgrade, dll.

## 11.2 Bisnis (di atas) Open Source

Pebisnis Open Source yang murni menjual produk software Open Source tidak melakukan penjualan lisensi seperti pada bisnis produk Proprietary, karena lisensi-lisensi produk Open Source membebaskan siapa saja menggunakan dan memodifikasi tanpa membayar biaya lisensi. Perbedaan bisnis Open Source bagi pengembang produk Open Source dibandingkan bisnis Proprietary bagi pengembang produk Proprietary selain bisnis jasa lisensi/penyewaan adalah bisnis jasa modifikasi. Untuk produk proprietary, jasa modifikasi tidak berlaku karena

tidak tersedia kode sumber, sehingga pengembang menjual hasil modifikasi dalam bentuk lisensi produk baru (upgrade).

Berikut ini beberapa model bisnis yang murni untuk produk Open Source.

- **Competency-based services:** Jasa berbasis kompetensi di bidang Open Source tanpa punya produk sendiri. Contoh: jasa system support, integrator, training, dll.
- **Distribution, services, and branding:** Mengembangkan produk Open Source dan menjual jasa terkait produk tersebut, termasuk menjual merek. Contoh: RedHat yang menjual paket distribusi RedHat Enterprise Linux dalam bentuk DVD, termasuk jasa dukungan teknis, meskipun software di dalamnya Open Source. Merek RedHat tidak free sehingga tidak dapat dijual oleh pihak lain tanpa izin RedHat. Catatan: Karena RedHat menjual merek, ada pengembang yang membuat turunan RedHat, salah satunya distro Linux CentOS (Community Enterprise Operating System).



Gambar 11.2 DVD Distro Linux Ubuntu ([linux-magazine.com](http://linux-magazine.com))

- **Widget frosting:** Bisnis ini berupa penjualan produk bukan Open Source, misal hardware (disebut *widget*) yang berisi software Open Source (disebut *frosting*). Contoh: IBM yang menjual hardware server berisi Linux. Google-Motorola, Samsung, Huawei, Axioo, IMO, dan vendor Android lainnya yang menjual hp/tabler berisi sistem operasi Android.
- **Accessorizing:** Menjual aksesoris yang berhubungan dengan Open Source, misalnya majalah Linux, buku-buku tentang produk Open Source, t-shirt, topi, dll.

### 11.3 Bisnis Kombinasi Open Source dan Tidak Open Source

- **Loss Leader:** Mengubah produk yang awalnya tidak FOSS menjadi FOSS agar tetap memimpin pasar. Contoh: Mozilla Firefox (FOSS) dari Netscape Navigator (Proprietary).
- **Free the software, sell the brand:** Membebaskan biaya lisensi (FOSS), dan menjual merek. Contoh: Google dengan sistem operasi Android-nya. Bisnis Google tidak menjual lisensi Android tapi menjual merek agar semakin banyak pengakses aplikasi

Google sehingga menghasilkan iklan, selain juga menjual perangkat keras hp Motorola.

- **Dual licensing/mission:** Merilis produk dalam dua lisensi FOSS & Proprietary. Contoh: MySQL (program database). Oracle sebagai pemilik utama hak cipta MySQL menyediakan dua jenis lisensi, yaitu GPL dan Proprietary.

## 11.4 Bisnis Proprietary di atas Open Source

- Menjual software Proprietary untuk dijalankan di sistem operasi FOSS dan ikut mengembangkan FOSS. Contoh: IBM dan Oracle menjual database proprietary untuk dipasang pada Linux, dan vendor cloud computing VMWare untuk Linux, dll.
- Menggabungkan software Proprietary dengan FOSS untuk aplikasi khusus. Contoh: Cloudera yang menjual solusi Cloudera Enterprise (Proprietary) dengan Apache-Hadoop Open Source.

## 11.5 Studi Kasus

- PT Nurul Fikri Cipta Inovasi menggunakan model bisnis *competency-based services* di bidang pelatihan, sertifikasi, development, dan technical support.
- Tim Pengembang BlankOn menggunakan model bisnis Linux distribution, services, branding Linux BlankOn, dan mengembangkan aplikasi khusus dengan cara Open Source, misal Teman Wisata (aplikasi jejaring sosial GIS untuk panduan pariwisata).
- IMO, Axioo, dan vendor Android di Indonesia menggunakan model widget frosting.

# Bab 12 Perbandingan Sistem Open Source dan Proprietary

Tujuan Umum: Mampu memahami persamaan dan perbedaan dasar Linux/Android dengan sistem operasi terkenal yang Closed Source (Windows, OSX/IOS).

Tujuan Khusus:

- a) Mampu menjelaskan perbedaan Linux dengan Sistem Operasi Closed Source Windows.
- b) Mampu menjelaskan perbedaan Linux dengan Sistem Operasi Closed Source OSX.
- c) Mampu menjelaskan perbedaan Linux Android dengan IOS dan Windows Mobile.

## 12.1 Pendahuluan

Sebagai pengantar perbandingan sistem operasi komputer Open Source dan Proprietary, bab ini hanya membahas perbandingan salah satu sistem operasi Open Source dengan dua sistem operasi Proprietary pada dua kelompok komputer “biasa” (*laptop, desktop, server, dsb.*) dan komputer bergerak (*tablet pc, smart phone, dsb.*)

Untuk komputer biasa, kita bahas perbandingan antara Linux (GNU/Linux) dengan OS X (Apple Macintosh Operating System 10) dan Windows (Microsoft Windows). Untuk komputer bergerak, kita bahas perbandingan antara Android dengan iOS (iPhone dan iPad) dan Windows Phone. Sistem operasi populer lainnya termasuk keluarga Unix (*Unix-like operating systems*) tidak dibahas, karena bisa diwakili Linux dan OS X yang keduanya juga Unix-like. Demikian pula sistem operasi komputer bergerak lainnya tidak dibahas karena juga Unix-like. Misal BlackBerry 10 merupakan turunan sistem operasi Unix QNX, ChomeOS dan Tizen turunan GNU/Linux, dan sebagainya.

## 12.2 Perbandingan Linux dengan Windows dan OS X

Sistem operasi Linux untuk komputer biasa memiliki nama lengkap GNU/Linux karena selain kernel Linux ada beberapa program dari GNU dalam sistem operasi Linux ini. Penulisan Linux tanpa GNU hanya untuk menyingkat atau memudahkan penyebutan.



Gambar 12.1 Perbandingan Linux, Windows, OS X ([www.okezone.com](http://www.okezone.com))

Membandingkan GNU/Linux (saja) dengan Windows dan OS X kurang tepat karena GNU/Linux merupakan nama sistem operasi (saja) yang belum lengkap jika belum dipaket menjadi distro atau distribusi. Sedangkan Windows dan OS X sudah berupa sistem operasi lengkap seperti distro. Oleh karena itu, perbandingan berikut ini mengarah ke salah satu atau beberapa distro Linux popular seperti Ubuntu, RedHat, dan BlankOn, untuk dibandingkan dengan Windows dan OS X.

*Tabel 12.1 Perbandingan Linux, Windows, OS X*

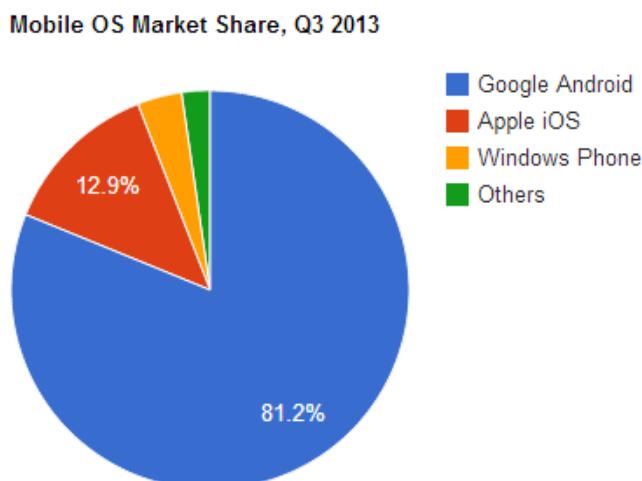
Topik	Linux	Windows	OS X
<b>Harga Lisensi</b>	Mayoritas distro Linux bebas diunduh atau dicopy dalam bentuk CD/DVD, sehingga biaya lisensi = 0, sudah termasuk aplikasi Office, Gimp photo editor, dll. Ada biaya bukan lisensi, misal biaya support/garansi, dll.	Harga lisensi Windows bervariasi antara Rp 500.000,- (bundel dengan penjualan komputer) hingga jutaan rupiah, belum termasuk aplikasi seperti MS Office, Adobe Photoshop, dll.	OS X dijual bersamaan dengan penjualan komputer Apple, sehingga harga tidak terlihat. Mulai versi OS X 10.9 tersedia gratis untuk upgrade dari versi lama.
<b>Kemudahan Digunakan</b>	Bagi pengguna komputer pertama kali, umumnya Linux desktop (yang dilengkapi GUI) seperti Ubuntu dan BlankOn mudah digunakan. Tapi bagi pengguna yang sudah terbiasa dengan Windows/Mac, maka Linux akan dirasakan lebih sulit karena belum biasa. Linux lebih sulit jika pada saat terpasang belum disertai lingkungan desktop lengkap seperti GNOME atau KDE.	Secara umum produk Windows dinilai lebih mudah digunakan karena masyarakat umum sudah terbiasa dengan Windows, sehingga kemudahan ini relatif. Kecuali untuk distro Linux tertentu, misal yang tidak menyertakan lingkungan desktop lengkap seperti GNOME atau KDE, Windows lebih mudah karena desktopnya telah menyatu dengan sistem operasi dan biasa dijual bersama komputer.	OS X dinilai lebih mudah dari Linux dan Windows untuk pengguna awam di bidang teknologi komputer. Tapi bagi pengguna yang biasa dengan Windows, OS X dinilai lebih sulit. OS X lebih mudah dari Linux yang belum disertai lingkungan desktop lengkap, karena desktop OS X sudah menyatu dengan sistem operasi, dan dibundel dalam penjualan komputer Apple Mac.
<b>Reliability</b>	Mayoritas distro Linux popular, misal RedHat dan turunannya atau Debian dan turunannya, terbukti dapat bekerja terus-menerus dalam waktu lama, terutama ketika diuji sebagai server.	Secara umum Windows kalah reliabel atau tidak mampu kerja lama dibandingkan Linux, meskipun Windows terus dikembangkan agar menjadi lebih reliabel.	OS X juga lebih reliabel dibandingkan Windows, dan setara Linux, karena dasarnya adalah FreeBSD yang setara Linux dalam hal reliabilitas.
<b>Software Aplikasi yang</b>	Untuk penggunaan desktop perkantoran, akses internet, grafis, dan	Untuk pengguna desktop perkantoran, internet, grafis, dan multimedia,	Untuk pengguna desktop perkantoran, internet, grafis, dan multimedia,

<b>Tersedia</b>	<p>multimedia, tersedia sangat banyak paket program untuk Linux, namun ada beberapa paket populer di Windows, misal MS Office, belum tersedia di Linux, meskipun ada pengganti setara, LibreOffice/OpenOffice.</p> <p>Aplikasi tertentu di Linux lebih baik, misal ketersediaan perintah terminal (<i>commands and tools</i>).</p>	<p>Windows memiliki aplikasi yang lengkap. Kelebihan Windows adalah didukung mayoritas pembuat game, meskipun dua tahun terakhir mulai banyak pengembang mendukung Linux, misalnya melalui Steam.</p> <p>Kemampuan Windows untuk diakses melalui perintah terminal (<i>command prompt</i>) masih terbatas.</p>	<p>Windows memiliki aplikasi yang lengkap seperti Linux dan Windows. OS X didukung beberapa aplikasi yang khusus dibuat untuk OS X. Seperti Linux, OS X yang berbasis Unix dapat dikendalikan penuh melalui terminal.</p>
<b>Biaya Lisensi Software Aplikasi</b>	Hampir semua software untuk kebutuhan sehari-hari tersedia untuk Linux dengan lisensi Free/Open Source.	Hampir semua software Free/Open Source yang tersedia untuk Linux juga tersedia untuk Windows, meskipun ada software populer yang dibuat untuk Windows tidak Free.	Seperti Linux dan Windows, tersedia banyak software Free/Open Source untuk OS X, meskipun tidak selengkap Linux dan Windows.
<b>Dukungan terkait Hardware</b>	<p>Banyak pilihan distro Linux yang dapat menyesuaikan kemampuan hardware, misal Linux tersedia untuk hardware spesifikasi rendah.</p> <p>Dukungan penjual/vendor hardware masih kalah dibandingkan dukungan mereka terhadap Windows, misal beberapa printer dan scanner tidak menyediakan driver untuk Linux. Dukungan vendor hardware terhadap Linux terus berkembang seiring pertumbuhan pengguna Linux.</p>	<p>Windows dibuat untuk hardware terbaru sehingga lemah atau bahkan tidak dapat digunakan untuk hardware lama atau yang spesifikasinya rendah.</p> <p>Microsoft sebagai pengembang Windows telah lama menjalin perjanjian dengan pembuat hardware tertentu, misal beberapa printer dan scanner, sehingga dukungan drivernya lebih baik dibandingkan terhadap Linux untuk hardware tertentu.</p>	Apple sebagai pengembang OS X juga telah lama menjalin perjanjian dengan pembuat hardware tertentu, misal beberapa printer dan scanner, sehingga dukungan drivernya setara Windows, lebih baik dibandingkan terhadap Linux untuk hardware tertentu.
<b>Keamanan</b>	Keamanan dari sisi virus dan maware lainnya, Linux lebih baik daripada Windows. Keamanan dari sisi perbaikan lubang keamanan juga lebih baik (cepat), karena banyak pihak dapat melakukan	Windows relatif kurang aman dari Linux karena banyaknya virus dan malware. Kelemahan yang ditemukan sangat tergantung perbaikan yang hanya dapat dibuat oleh Microsoft.	OS X berada di antara Linux dan Windows, yaitu keamanan terhadap virus seperti Linux, namun perbaikan kelemahan menunggu dari Apple. OS X dinilai paling aman, karena ketersediaan

	perbaikan. Salah satu yang membuat Linux relatif lebih aman karena pengguna diharapkan hanya menginstal program dari sumber utama (Repository) yang dijaga keamanannya.	Salah satu penyebab Windows relatif kurang aman adalah kemudahan pengguna menginstal program dari mana saja yang belum tentu dijaga keamanannya.	program untuk pengguna biasanya sangat terjaga, yaitu dari sumber (Repository) yang disediakan Apple.
<b>Tersedia kode sumber (Open Source)</b>	Sebagian besar software utama (sistem operasi) dan aplikasi yang dibutuhkan pengguna biasa di Linux tersedia versi Open Source-nya.	Majoritas software yang dibuat untuk pengguna Windows biasa tidak Open Source, meskipun tersedia versi Open Source yang setara.	OS X lebih dekat ke Linux daripada Windows karena OS X turunan FreeBSD, meskipun beberapa program telah diubah menjadi Closed.
<b>Support (Dukungan Teknis kepada Pengguna)</b>	Dukungan teknis terhadap pengguna Linux tersedia gratis melalui komunitas, atau berbayar melalui perusahaan penyedia support, misal RedHat Inc atau mitranya. Banyak perusahaan dapat menyediakan dukungan teknis secara penuh karena Linux FOSS, secara gratis atau berbayar.	Dukungan teknis penggunaan Windows juga tersedia gratis melalui sesama pengguna, atau berbayar melalui perusahaan penyedia support, misal mitra Microsoft. Hanya Microsoft yang dapat menyediakan dukungan teknis penuh berbayar, tidak secara gratis.	Dukungan teknis penggunaan OS X juga tersedia gratis melalui sesama pengguna, atau berbayar melalui perusahaan penyedia support, misal mitra Apple. Hanya Apple yang dapat menyediakan dukungan teknis penuh berbayar, tidak secara gratis.

### 12.3 Perbandingan Android dengan Windows Phone dan iOS

Secara umum perbandingan sistem operasi untuk komputer bergerak Open Source Android dengan proprietary Windows Phone dan iOS hampir sama dengan perbandingan untuk komputer biasa di atas, karena Android berbasis Linux, Windows Phone berbasis Windows, dan iOS berbasis OS X.



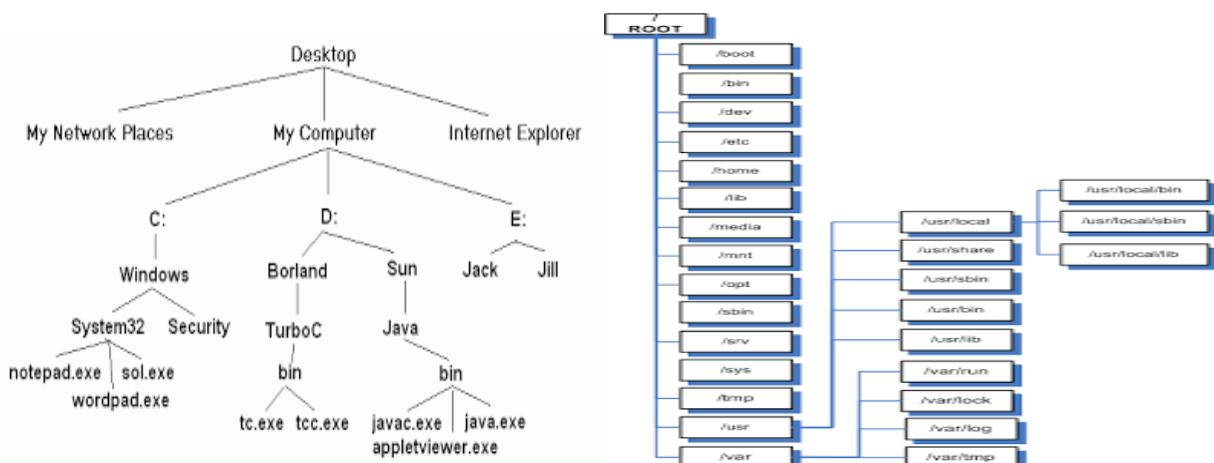
Gambar 12.2 Grafik pangsa pasar smartphone Android 81,2% (IDC, 2013)

Beberapa kelemahan Linux di komputer biasa menjadi hilang atau berkurang di komputer bergerak.

- **Kemudahan Digunakan:** penggunaan Android untuk telepon dan tablet sama mudahnya dengan iOS untuk iPhone dan iPad serta Windows Phone untuk telepon, karena tampilan, fungsi, dan kemampuan ketiga sistem operasi itu setara. Tidak seperti distro Linux komputer biasa yang memiliki variasi sangat banyak, distro Linux Android hanya memiliki distro tunggal dari Google dan turunan yang sangat serupa, misal CM (CynogenMod). Tidak seperti Linux di komputer biasa yang umumnya tidak dibundel dalam penjualan komputer, Linux Android dijual bersamaan (terinstal) pada hp, tablet, netbook dan laptop seperti Windows, Windows Phone, OS X, dan iOS dijual bersama hardware hp atau tablet.
- **Software Aplikasi yang Tersedia:** aplikasi untuk Android, berlisensi Freeware, Shareware, Free/Open Source Software, maupun Proprietary, jauh lebih banyak tersedia daripada aplikasi untuk iOS dan Windows Phone. Ini seiring dengan jumlah perangkat hp dan tablet Android yang telah diproduksi dan terjual sangat jauh lebih banyak daripada penjualan perangkat iOS dan Windows Phone, termasuk software game yang selama ini menjadi kelemahan Linux juga tersedia berlimpah untuk Android.
- **Dukungan Hardware:** Seperti dukungan pengembang aplikasi, dukungan vendor atau pembuat hardware juga sangat besar terhadap Android.
- **Support:** Android disupport penuh oleh perusahaan raksasa Google, ditambah beberapa perusahaan besar lain yang tergabung dalam Open Handset Alliance seperti Samsung, LG, Sonny, dll. Karena berlisensi Open Source, Android juga disupport oleh komunitas yang besar. Kekurangan Android secara teknis dilengkapi oleh komunitas pengembang Open Source, antara lain menghasilkan sistem operasi turunan Android, CynogenMod.

## 12.4 Perbandingan File/Direktori Linux, Windows, dan OS X

Linux dan OS X memiliki struktur sistem file yang sama, karena satu keluarga Unix, yaitu berbentuk pohon dengan direktori paling tinggi disebut root directory (direktori akar), slash (/). Windows memiliki direktori atau folder paling atas lebih dari satu berupa huruf C, D, E, dsb.



Gambar 12.3 Struktur Direktori Windows dan Unix ([purwakarta.linux.or.id](http://purwakarta.linux.or.id))

Penamaan file/direktori di Linux case sensitive (beda antara huruf kecil dan besar), sedangkan di Windows dan OS X tidak (sama antara huruf kecil dan besar). Di Linux, direktori termasuk file dengan kode d. Sedangkan di Windows, direktori disebut folder (bukan file). Pengembang dan pengguna OS X dan Linux biasanya juga menggunakan istilah folder untuk direktori.

Penamaan direktori di Linux diawali dengan garis miring atau slash (/) sedangkan di Windows diawali huruf diikuti titik dua dan garis miring terbalik atau backslash (\). Misal pengguna Linux “rus” punya direktori di /home/rus, maka pengguna “rus” di Windows punya folder C:\Users\Rus.

# Bab 13. Pengantar Jaringan Komputer dengan Linux

Tujuan Umum: Mampu mengkonfigurasi jaringan dengan sistem operasi Open Source melalui terminal (modus teks) dan menu (GUI).

Tujuan Khusus:

- a) Mampu menjelaskan cara setting jaringan dengan perintah-perintah.
- b) Mampu menjelaskan cara setting jaringan di desktop.

## 13.1 Pendahuluan: Bekerja dalam Modus Teks

Mengakses jaringan komputer melalui Linux desktop atau GUI seperti GNOME dan KDE tidak sulit, karena biasanya telah terpasang program Network Manager sehingga cukup klik ikon jaringan di panel desktop untuk memilih jaringan apa yang akan digunakan. Bahkan untuk jaringan kabel yang terhubung ke server DHCP (penyedia layanan setup jaringan otomatis), komputer langsung terhubung ke jaringan ketika kabel UTP ditancapkan ke komputer.

Namun, perintah perlu difahami oleh mahasiswa komputer atau Anda yang bekerja di bidang komputer, karena banyak komputer server yang digunakan perusahaan biasanya hanya bisa diakses melalui jaringan, sehingga hanya bisa disetting melalui konsol atau modus teks. Perintah-perintah berikut ini dapat digunakan untuk berbagai jenis distro sistem operasi Linux/Unix, termasuk Ubuntu dan BlankOn.

Salah satu yang luar biasa jika Anda mampu memberi perintah dan mengedit file konfigurasi adalah saat Anda bekerja di komputer jauh yang harus dikelola melalui jaringan. Anda akan kesulitan kalau harus menggunakan desktop secara remote. Sehingga jalan terbaik dan tercepat adalah mengakses dengan ssh, lalu memberikan perintah atau mengedit file teks di *terminal* atau *konsole*. Anda juga akan menghemat waktu dan tidak membutuhkan *bandwidth* yang besar atau tidak perlu akses jaringan/internet yang cepat.

Jadi, langkah pertama untuk mengakses komputer Linux dalam modus teks adalah menjalankan Terminal atau program virtual conlose lainnya.

## 13.2 Mematikan Network Manager

Jika Anda bekerja di komputer yang menjalankan desktop GUI, misal GNOME Unity, Shell, atau Manokwari, maka langkah kedua adalah mematikan pengatur jaringan otomatis atau network-manajer dengan perintah sebagai berikut:

```
$ sudo service network-manager stop
```

atau

```
$ sudo /etc/init.d/network-manager stop
```

Jika ingin kembali setting jaringan secara otomatis, jalankan perintah berikut:

```
$ sudo service network-manager start
```

atau

```
$ sudo /etc/init.d/network-manager start
```

### 13.3 Jaringan Kabel Ethernet

Panduan ini hanya untuk jaringan kabel ethernet karena hampir semua komputer memiliki jaringan kabel ethernet, dan server biasanya juga dihubungkan melalui kabel, bukan wireless. Jika Anda hanya menggunakan jaringan wireless, cara info cara menggunakan wireless melalui konsol atau hubungi penulis untuk mendapatkan bantuan melalui email.

Anda diharapkan telah memahami dasar-dasar jaringan TCP/IP, antara lain jenis-jenis peralatan jaringan, pemasangan kabel dan kartu jaringan, pengalamatan jaringan, dan lain-lain. Penjelasan dalam bab ini langsung ke penggunaan praktis setiap perintah atau file konfigurasi jaringan. Buku atau dokumen tentang jaringan TCP/IP banyak tersedia di internet. Namun jika Anda langsung ingin mencoba tanpa harus tahu teori jaringan TCP/IP, Anda dapat mengikuti alamat-alamat jaringan yang diberikan oleh pengelola jaringan di kampus atau kantor Anda, misalnya:

Alamat IP komputer Anda: 192.168.1.250

Netmask atau Mask: 255.255.255.0 atau 24

Alamat IP Gateway/Router: 192.168.1.1

Alamat IP Server DNS: 192.168.1.1 dan 8.8.8.8

### 13.4 Perintah ifconfig

Misal komputer Anda memiliki kartu jaringan kabel ethernet. Nama *device* (alat yang dikenali oleh sistem operasi) ethernet adalah eth0. Perintah ifconfig untuk memberi alamat IP, misalnya 192.168.1.250, harus dilakukan oleh user Administrator atau *root*, dengan ditambahkan perintah *sudo* di depannya atau didahului perintah *sudo su -*. Akhiri setiap perintah dengan menekan *enter*.

```
$ sudo /sbin/ifconfig eth0 192.168.1.250
```

atau

```
$ sudo su -
```

```
Password:
```

```
# ifconfig eth0 192.168.1.250
```

Perintah di atas akan memberikan alamat IP 192.168.1.250 dengan alamat broadcast atau Bcast: 192.168.1.255 dan netmask atau Mask: 255.255.255.0. Untuk melihat hasilnya, ketikkan perintah ifconfig saja. Hasilnya sebagai berikut.

```
# ifconfig
eth0    Link encap:Ethernet  HWaddr 00:26:22:11:D7:A0
        inet  addr:192.168.1.250  Bcast:192.168.1.255  Mask:255.255.255.0
              inet6 addr: fe80::226:22ff:fe11:d7a0/64 Scope:Link
                    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                    RX packets:873 errors:0 dropped:0 overruns:0 frame:0
                    TX packets:941 errors:0 dropped:0 overruns:0 carrier:1
                    collisions:0 txqueuelen:1000
                    RX bytes:857017 (836.9 KiB)  TX bytes:155001 (151.3 KiB)
                    Interrupt:28

lo     Link encap:Local Loopback
        inet  addr:127.0.0.1  Mask:255.0.0.0
              inet6 addr: ::1/128 Scope:Host
```

```
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:45 errors:0 dropped:0 overruns:0 frame:0
TX packets:45 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:4180 (4.0 KiB) TX bytes:4180 (4.0 KiB)
```

Catatan:

- Selain eth0 ada lo (singkatan dari localhost), yakni jaringan di dalam komputer yang tidak membutuhkan kabel. Localhost berguna untuk komunikasi program jaringan dalam komputer.
- User biasa hanya dapat menampilkan status atau konfigurasi yang ada, dan kadang harus menuliskan *path* (nama direktori dan file secara lengkap), misal /sbin/ifconfig.
- Konfigurasi jaringan pada Fedora biasanya dikendalikan oleh program Network Manager, sehingga harus dimatikan jika ingin mengubah jaringan secara manual. Perintahnya sebagai berikut:

```
# /etc/init.d/network-manager stop
```

Jika Anda ingin memberikan alamat broadcast dan netmask secara manual, contoh perintahnya sebagai berikut:

```
# ifconfig eth0 192.168.1.250 broadcast 192.168.1.255 netmask
255.255.255.0
```

Untuk mengetahui cara menggunakan perintah ifconfig lebih lanjut, ketikkan perintah man ifconfig yang artinya menampilkan manual atau petunjuk penggunaan ifconfig.

```
# man ifconfig
```

## 13.5 Perintah route

Perintah route berguna antara lain untuk memberikan akses komputer ke jaringan lain, misal ke internet, melalui gateway atau router. Perintah ini juga harus dijalankan oleh user root, kecuali hanya untuk menampilkan konfigurasi. Sebagai contoh komputer Anda memiliki kartu jaringan eth0 dengan alamat 192.168.1.250 dan akan disambungkan ke internet melalui router atau gateway dengan alamat 192.168.1.1. Perintahnya adalah:

```
# route add default gw 192.168.1.1 eth0
```

Jika hanya sebuah kartu, Anda tidak harus menuliskan eth0. Untuk melihat hasilnya, ketikkan perintah route -n (lihat juga Gambar 10.2 yang menunjukkan tabel routing):

```
# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref Use Iface
192.168.1.0    0.0.0.0        255.255.255.0 U      0      0      0 eth0
0.0.0.0         192.168.1.251  0.0.0.0       UG     0      0      0 eth0
```

Untuk menghapus sebuah baris dari tabel routing, jalankan perintah route del default. Jika terdapat beberapa kartu jaringan dengan route berbeda, Anda dapat menghapus salah satunya

dengan cara menuliskan gw diikuti alamat IP dan nama device-nya.

```
# route del default  
# route del default gw 192.168.1.1 eth0
```

## 13.6 Perintah echo untuk Resolver

File konfigurasi untuk mengakses server DNS (Domain Name System) atau disebut juga *resolver* adalah /etc/resolv.conf. Anda harus memasukkan alamat IP server DNS yang diawali dengan kata nameserver dan spasi ke dalam file resolv.conf itu. Misalnya alamat server DNS di jaringan Anda 192.168.1.1 (contoh ini kebetulan sama dengan alamat router/gateway) atau 8.8.8.8, perintahnya sebagai berikut:

```
# echo "nameserver 192.168.1.1" > /etc/resolv.conf
```

atau

```
# echo "nameserver 8.8.8.8" >> /etc/resolv.conf
```

Arti perintah di atas adalah membuat file /etc/resolv.conf yang isinya sebuah baris “nameserver 192.168.1.1” tanpa tanda petik, dan perintah kedua dengan dua panah (>>) akan menambahkan “nameserver 8.8.8.8” sebagai baris baru.

## 13.7 Perintah dhclient

Jika suatu saat Anda ingin kembali mengakses server DHCP untuk mendapatkan alamat IP, gateway, dan resolver secara otomatis, perintahnya adalah dhclient. Untuk menjalankan ulang dhclient, lebih dahulu matikan dhclient dengan perintah killall dhclient, lalu kembali jalankan dhclient diikuti nama kartu jaringan.

```
# dhclient eth0
```

## 13.8 Perintah ping

Perintah ping berguna untuk menguji jaringan, misalnya apakah komputer telah tersambung dengan router/gateway. Perhatikan perintah dan hasilnya di bawah ini.

```
$ ping 192.168.1.1  
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.  
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.272 ms  
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.261 ms  
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.264 ms
```

Jika Anda salah memberi perintah sebelumnya, misal alamat IP Linux tidak dalam satu jaringan dengan alamat IP router/gateway, atau perintah benar tapi komputer yang dituju mati atau kabel jaringan putus maka akan muncul pesan “*Network is unreachable*” atau “*Destination Host Unreachable*”.

```
$ ping 192.168.1.1  
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.  
From 192.168.1.250 icmp_seq=2 Destination Host Unreachable  
From 192.168.1.250 icmp_seq=3 Destination Host Unreachable  
From 192.168.1.250 icmp_seq=4 Destination Host Unreachable
```

## 13.9 Mengedit File Resolver

Misalnya Anda ingin memgubah alamat server DNS dari hanya satu baris nameserver 192.168.1.1 menjadi tiga baris dengan menambahkan alamat DNS server untuk ke internet

yang aman 180.131.144.144 serta 180.131.145.145, maka isi file /etc/resolv.conf harus diedit dengan isi tiga baris berikut ini.

```
nameserver 192.168.1.254  
nameserver 180.131.144.144  
nameserver 180.131.144.145
```

Catatan:

- Editor teks vi dapat digunakan untuk mengedit file secara remote dari terminal. Setelah menjadi root dengan perintah su, ketikkan vi /etc/resolv.conf diikuti enter. Untuk mulai mengedit, tekan i atau I yang artinya *insert*. Untuk menyimpan tekan Esc, titik dua (:), dan wq diikuti enter. Jika ingin membatalkan, karena banyak salah misalnya, tekan Esc diikuti titik dua, 1, dan tanda seru (!) serta enter.
- Server DNS bertugas menerjemahkan nama domain (misalnya nurulfikri.ac.id) menjadi alamat IP (misalnya 110.232.72.124), dan sebaliknya. Resolver bertugas menanyakan ke server DNS setiap ada permintaan akses ke nama domain. Tanpa server DNS atau tanpa ada resolver yang benar isinya, komputer tidak tahu alamat IP infolinux.co.id.
- Server DNS dengan alamat 180.131.144.144 dan 180.131.145.145 adalah milik Nawala yang juga berguna untuk menyaring akses internet yang tidak baik seperti pornografi, perjudian, penipuan, virus, dan sebagainya.

### 13.10 Perintah host dan dig

Kedua perintah ini, host dan dig, berguna untuk mengetahui alamat IP dari sebuah nama domain. Perintah host dan dig menggunakan isi file resolv.conf untuk mengakses server DNS. Berikut ini contoh perintah host dan hasilnya:

```
$ host nurulfikri.ac.id  
nurulfikri.ac.id has address 110.232.72.124  
nurulfikri.ac.id mail is handled by 10 mail.nurulfikri.ac.id.  
$  
$ dig nurulfikri.ac.id  
  
; <>> DiG 9.8.1-P1 <>> nurulfikri.ac.id  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15769  
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0  
  
;; QUESTION SECTION:  
;nurulfikri.ac.id. IN A  
  
;; ANSWER SECTION:  
nurulfikri.ac.id. 71351 IN A 110.232.72.124
```

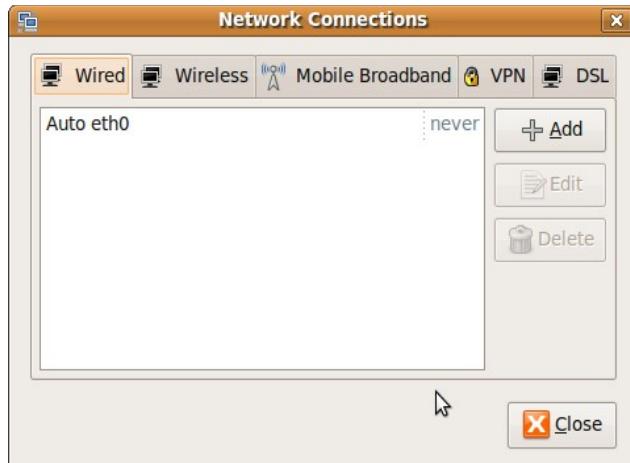
Hasil perintah host lebih sederhana, hanya menampilkan alamat IP, sedangkan perintah dig menampilkan banyak informasi yang berhubungan dengan domain nurulfikri.ac.id.

### 13.11 Setting Jaringan Melalui Menu Desktop <sup>10)</sup>

- Buka aplikasi administrasi jaringan di System → Preferences → Network

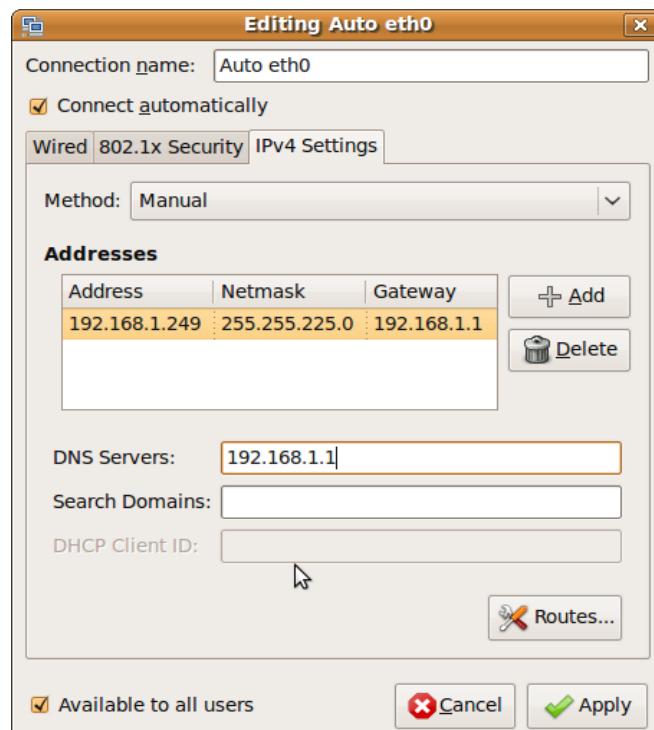
## Connections.

- Klik pada *Auto eth0* pada tab *Wired* (pengaturan koneksi untuk device ethernet) dan pilih *Edit*. Bila diminta password masukkan password dari user anda. User anda harus masuk sebagai user administrator.



Gambar 13.1 Aplikasi manajemen jaringan

- Pada jendela *Edit Auto eth0* anda pilih tab *IPv4 Settings* untuk memulai pengeditan IP address.



Gambar 13.2: Pengaturan IP address secara manual

- Pada jendela *IPv4 Settings* anda klik pada pilihan *Method*. Bila Anda hendak mengatur IP address secara otomatis anda pilih pada *Automatic (DHCP)* bila anda hendak menggunakan pengaturan secara statik maka pilih *Manual*. Bila anda klik

pada manual klik pada *Add* dan Masukkan IP Address, Netmask, dan Gateway. Untuk DNS anda tuliskan di bawah. Ingat setiap mengetikkan IP address harus diakhiri dengan menekan tombol Enter.

- Bila sudah klik *Apply* dan *Close* untuk menutup jendela pengaturan jaringan. Maka konfigurasi jaringan anda akan langsung berubah.

# Bab 14 Instalasi Sistem Operasi Ubuntu dan Troubleshooting

Tujuan Umum: Mampu menginstal Linux Ubuntu Desktop hingga kembali restart dan memahami cara mengatasi beberapa masalah (troubleshooting) dasar.

Tujuan Khusus:

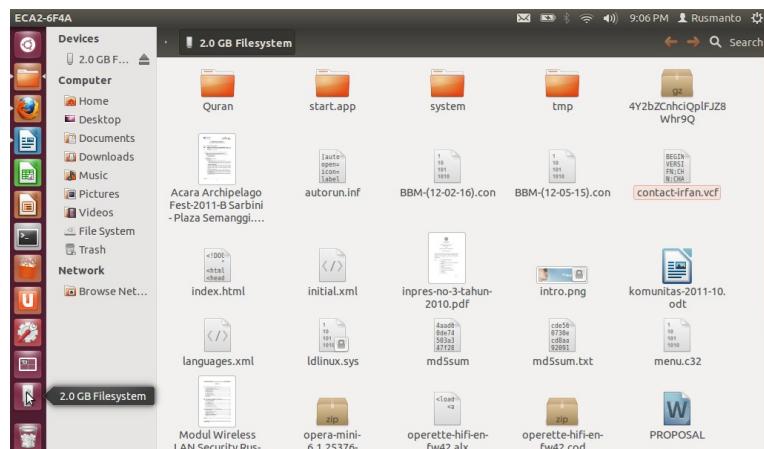
- Mampu menjelaskan langkah demi langkah menginstal Linux Ubuntu atau yang sejenis dari sebelum booting hingga kembali komputer direboot setelah instalasi selesai.
- Mampu menjelaskan cara mengatasi masalah (troubleshooting) dasar seperti gagal login, tidak bisa akses prompt root, lupa password, dll.

## 14.1 Backup Data

Sebelum memulai instal Ubuntu ke hard disk, lebih dahulu Anda harus menjawab pertanyaan, “Adakah ada data di hard disk yang sangat penting dan belum dikopi atau dibackup ke tempat lain?” Jika ada dan belum dibackup, lakukan backup segera dengan sistem operasi lain yang telah terpasang, atau backup menggunakan Ubuntu LiveCD/USB yang dibahas di bagian sebelumnya.

Cara backup data paling praktis adalah menggunakan USB stick atau flash disk atau kartu memori. Tancapkan USB stick atau kartu memori yang biasanya telah diformat sebagai FAT (DOS/Windows), sehingga USB stick atau kartu memori itu dikenali Ubuntu. Launcher akan menampilkan ikon USB stick di bagian bawah (di atas ikon Trash). Artinya, Launcher mengenali devais penyimpanan USB stick dan kartu memori. Jika ikon itu diklik, akan menampilkan program Files (nautilus) seperti pada Gambar 14.1.

Untuk melakukan backup, pindahkan data penting dari partisi Windows ke flash disk atau kartu memori. Ciri partisi Windows, di dalamnya ada folder Windows dan Program Files, dan cari nama file dan folder yang akan dibackup.



Gambar 14.1 USB Flash disk untuk backup data

Catatan:

- Ubuntu 13.04 dan versi yang lebih baru secara bawaan dapat membuka isi hard disk Windows yang diformat NTFS, tanpa harus instal driver NTFS lebih dahulu.
- Jika Anda akan mencabut USB flash disk, lebih dahulu klik mouse kanan pada ikon di File Browser atau ikon USB flash disk di Launcher, lalu pilih “Safely remove”.

Selain membackup data ke flash disk atau CD/DVD atau hard disk lokal, jika ukuran data

tidak terlalu besar, misal hanya beberapa MB, Anda juga dapat membackup data dengan cara mengirim ke email Anda di internet. Bahkan Anda dapat mengupload ke tempat penyimpanan di internet (Cloud Computing), misalnya dengan di server cloud Ubuntu One yang dibahas di bagian akhir edisi ini.

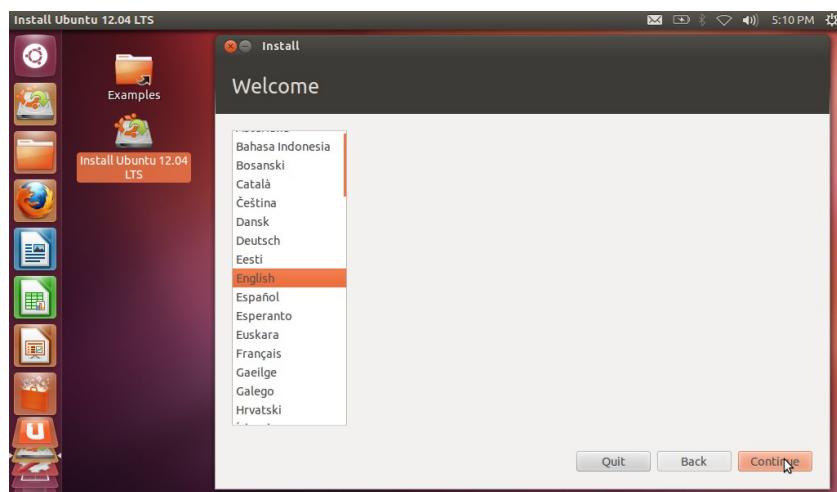
Setelah data dibackup, pastikan sisa ruang hard disk minimal 5,4 GB. Atau jika kurang dari 5,4 GB, pastikan Anda bisa menghapus data yang tidak penting. Lebih aman lagi jika Anda telah memiliki hard disk kosong atau paling partisi kosong dengan ukuran di atas 5,4 GB.

Backup data ini untuk jaga-jaga terjadi kesalahan yang disengaja atau tidak Anda, misal menghapus atau memformat partisi yang berisi data penting. Dalam kondisi normal, pemotongan partisi oleh Ubuntu pada proses instalasi tidak menghilangkan data yang ada.

## 14.2 Memulai Install Ubuntu

Setelah data dibackup, kembali pastikan komputer dapat menjalankan Ubuntu LiveCD atau LiveUSB dengan baik untuk memastikan tidak ada masalah dengan CD/DVD atau flash disk yang digunakan untuk booting. Jika komputer masih menjalankan Ubuntu LiveCD/USB, program instalasi dapat dijalankan dengan klik ikon Install pada Launcher (pinggir kiri desktop) kedua dari atas, atau klik dua kali ikon “Install Ubuntu 13.04” di desktop.

Jendela pertama yang muncul adalah untuk memilih bahasa untuk instalasi. Secara bawaan (default) LiveCD yang digunakan buku ini menggunakan bahasa Inggris (English) seperti Gambar 14.2. Anda dapat mengubahnya menjadi Bahasa Indonesia atau yang lain, lalu klik Continue.



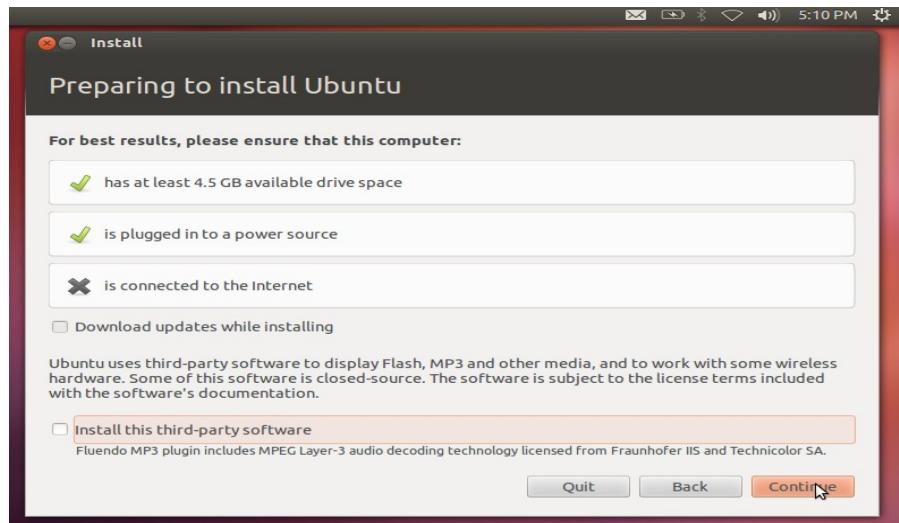
Gambar 14.2 Memilih bahasa untuk instalasi

## 14.3 Perlu tidaknya Akses Internet

Program instal Ubuntu secara otomatis mendeteksi ada tidaknya akses internet. Jika ada akses internet, dalam proses instal Ubuntu akan mengdownload daftar paket dari internet untuk memperbarui (update) sistem manajemen paket atau Ubuntu Software Center. Proses update paket ini memakan waktu lama jika akses internet Anda lambat, karena total data yang didownload sekitar 17 MB. Dengan kecepatan akses internet 128 kbps membutuhkan waktu download sekitar 25 menit. Namun jika Anda memiliki akses internet cepat, misalnya 1 Mbps, maka hanya membutuhkan waktu sekitar 3 menit.

Setelah memilih bahasa (Gambar 14.2) Anda diberikan hasil cek program installer terhadap keadaan komputer, yaitu ukuran ruang hard disk yang tersedia, sambungan daya listrik, dan pilihan untuk menggunakan jaringan akses internet atau tidak (Gambar 14.3). Untuk proses

instalasi yang cepat, Anda dapat mematikan akses internet dengan mencabut kabel jaringan atau klik ikon jaringan di panel kanan atas dan pilih Disconnect.



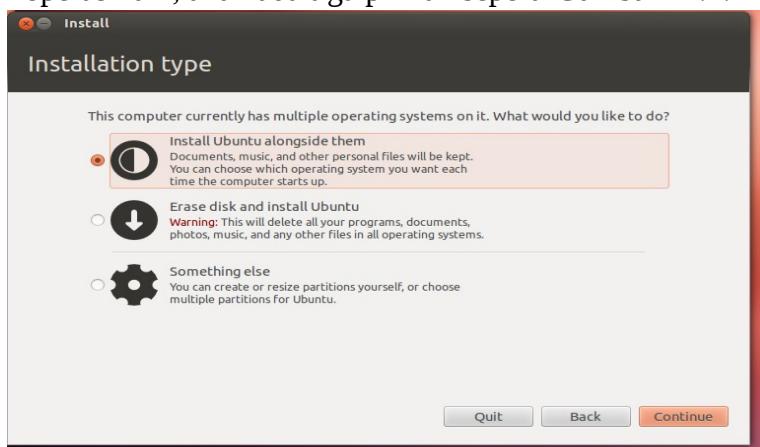
Gambar 14.3 Kondisi hard disk, listrik, dan internet

Anda juga diberi pilihan langsung mendownload paket-paket updates pada saat instalasi dan menginstal software dari pihak ketiga yang tidak termasuk FOSS (Free/Open Source Software), misalnya plugin pemutar lagu MP3 dari Fluendo. Untuk mempercepat proses instalasi, penulis menyarankan Anda tidak memilih semuanya karena kita dapat melakukannya setelah instalasi selesai.

Jika Ubuntu mendetect ada akses wireless, jendela berikutnya akan menawarkan setup akses ke wireless. Anda dapat menolak dengan memilih "I dont want to connect to a wi-fi network right now."

#### 14.4 Mengedit Partisi Hard disk

Langkah mengedit partisi hard disk ini sebaiknya dilakukan dengan hati-hati, terutama jika Anda tidak ingin menghapus sistem operasi yang telah terpasang di hard disk. Jika hard disk telah berisi sistem operasi lain, akan ada tiga pilihan seperti Gambar 14.4.



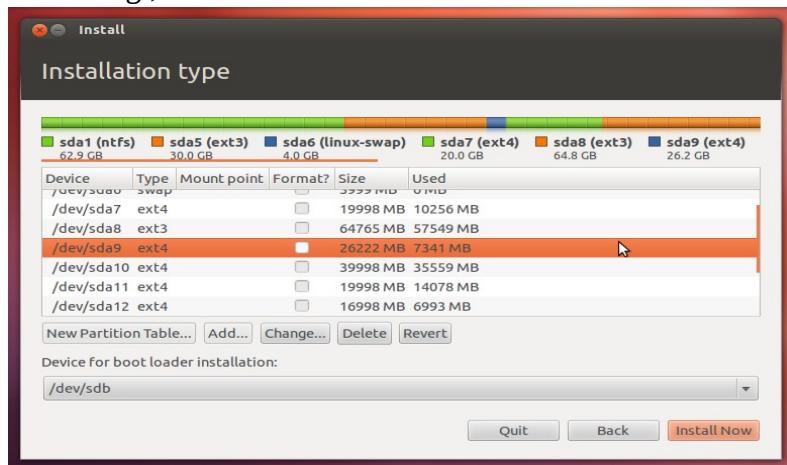
Gambar 14.4 Mengedit partisi hard disk

Saran penting penulis, Anda jangan memilih yang kedua "Erase disk and install Ubuntu" kecuali Anda ingin menghapus semua data di hard disk dan diganti dengan Linux. Dua

pilihan yang kami sarankan adalah salah satu dari dua pilihan ini:

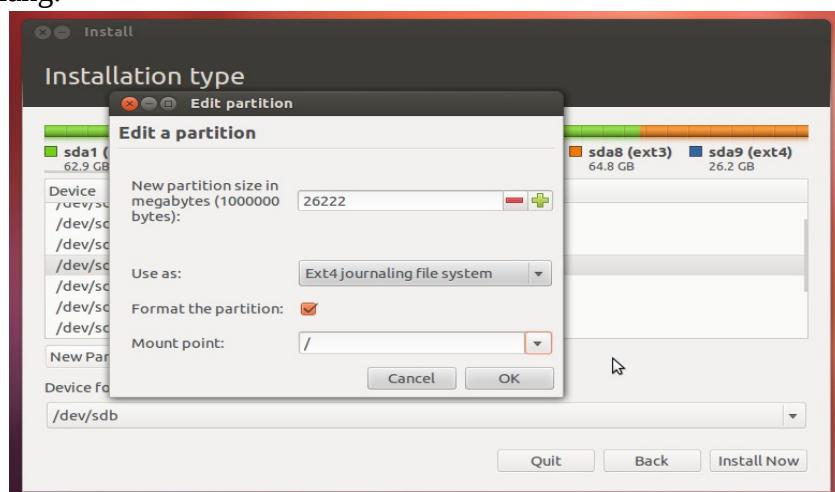
- *Install Ubuntu along side them*. Pilihan ini akan memotong partisi yang telah digunakan sistem operasi lain, untuk menghasilkan partisi kosong baru yang akan digunakan untuk menginstal Ubuntu. Program instal akan membuat paling tidak dua partisi baru, yakni partisi Swap untuk virtual memory sebagai penambah RAM dan partisi utama (/) untuk dijadikan tempat menginstal sistem operasi dan aplikasi.
- *Something else*. Pilihan ini akan memberi kebebasan Anda mengedit partisi secara manual, misalnya memotong partisi seperti pilihan pertama, menghapus salah satu partisi, atau membuat partisi baru dari sisa ruang hard disk yang belum dipartisi.

CATATAN: Jika Anda memilih “Something else”, Anda perlu kembali hati-hati agar tidak menghapus partisi yang ada sistem operasi (untuk dipertahankan) atau ada data penting (belum dibackup). Langkah pertama adalah memastikan partisi yang akan dihapus benar-benar tidak dibutuhkan lagi, misal /dev/sda9.



Gambar 14.5 Mengedit manual untuk menghapus sebuah partisi

Klik Change untuk mulai mengedit sebuah partisi. Dalam contoh ini partisi /dev/sda9 akan diformat ulang dijadikan sistem utama (titik mount /) sebagai ext4 Journaling file system, seperti Gambar 14.6. Klik OK untuk melihat rangkuman sebelum memutuskan benar-benar memformat ulang.



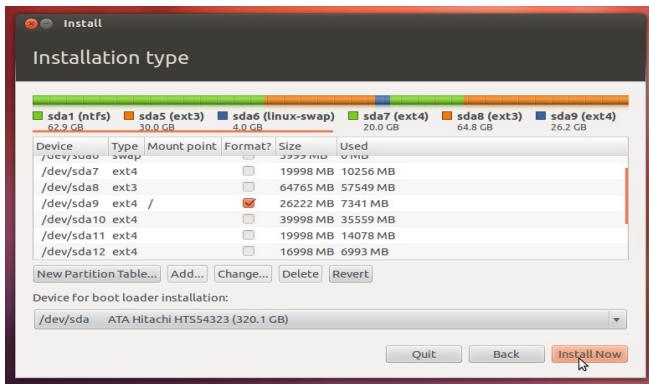
Gambar 14.6 Menandai untuk diformat dan dijadikan partisi utama

Cara manual (Something else) ini juga memberi kesempatan Anda mengubah letak boot loader Ubuntu 13.04 di master boot record hard disk (/dev/sda) atau di partisi utama (dalam

contoh ini /dev/sda9). Lihat bagian bawah Gambar 14.7. Jika memilih instal boot loader di partisi utama, ketika reboot Anda tidak langsung dapat menekan Enter untuk booting ke Ubuntu 13.04 ini, tapi harus memanggilnya melalui Linux lain atau perintah boot loader (Grub2).

PERHATIAN: Jika Anda ragu-ragu dan ingin membatalkan atau mengedit ulang, masih ada kesempatan seperti tampilan Gambar 14.7, dengan cara klik Back. Jika Anda sudah yakin, klik Install Now, yang artinya partisi terpilih diformat ulang dan semua data lama dihapus.

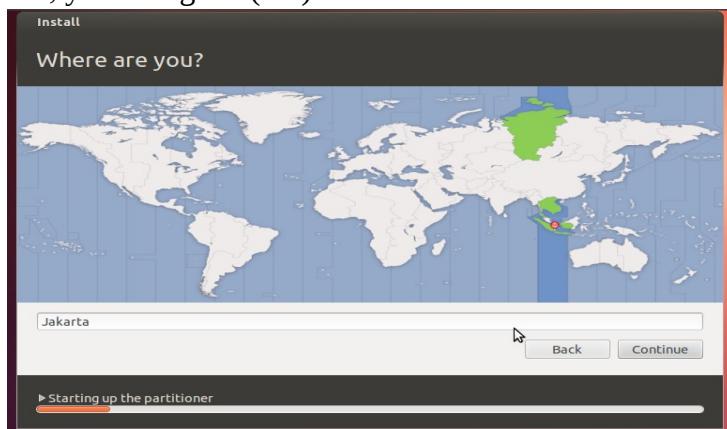
Jika Anda melakukan kesalahan di langkah ini, maka untuk menyelamatkan data yang telah diformat tidak bisa dilakukan dengan LiveCD ini. Anda harus menggunakan program khusus atau membawa hard disk Anda ke orang yang ahli menyelamatkan data pada partisi hard disk yang telah diformat.



Gambar 14.7 Melihat kembali perubahan sebelum dieksekusi

## 14.5 Mengatur Waktu dan Keyboard

Pada langkah pengaturan waktu, Anda dapat menggunakan mouse untuk memilih kota terdekat, misalnya Jakarta. Waktu dapat diatur kembali setelah instalasi, dengan opsi waktu universal (internet) atau waktu lokal (komputer). Pada langkah pengaturan keyboard, Anda tidak perlu mengubahnya jika Anda menggunakan tata letak (layout) keyboard umum/standar yang ada di Indonesia, yakni English (US).

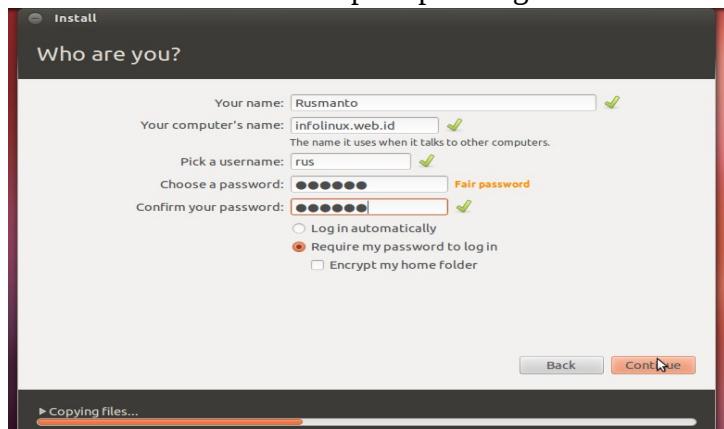


Gambar 14.8 Setting wilayah waktu

## 14.6 Username dan Password

Langkah terakhir yang penting juga adalah menuliskan nama lengkap Anda sebagai pengguna Ubuntu ini, nama komputer, username dan password untuk login. Password ini akan ditanyakan setiap kali Anda melakukan perubahan sistem atau pekerjaan yang berhubungan

dengan administrasi sistem, misalnya setup jaringan dan menginstal atau menghapus paket software. Username pengguna yang dibuat pada saat instalasi ini dapat berubah menjadi root dengan perintah “sudo su” melalui command prompt atau gnome-terminal.

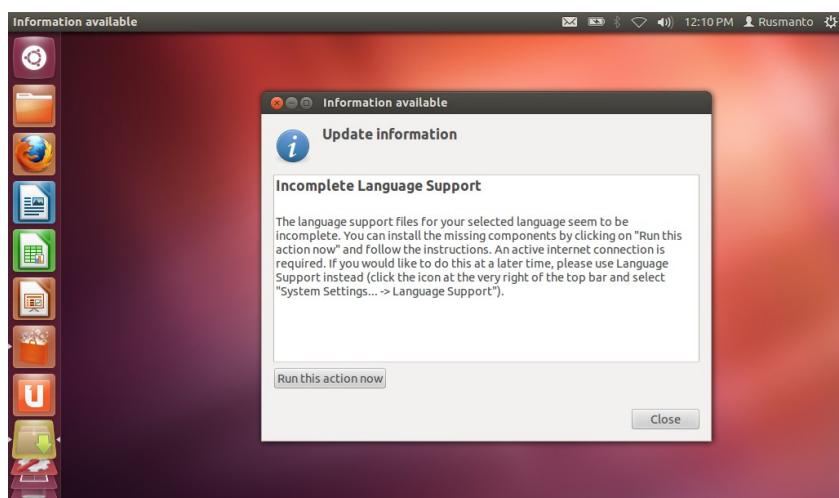


Gambar 14.9 Membuat username dan password

## 14.7 Login Pertama setelah Instalasi

Jika Anda telah berhasil menginstal ke hard disk, ada tawaran mereboot atau restart komputer. Tunggu beberapa saat komputer shutdown dan startup kembali hingga masuk ke jendela login atau langsung desktop Ubuntu.

- Untuk login pertama setelah instalasi, gunakan username dan password yang dibuat pada saat instalasi, kecuali Anda memilih login otomatis tanpa password.
- Akan muncul jendela yang menunjukkan bahwa paket bahasa belum semuanya terpasang, antara lain untuk Spelling bahasa Inggris.
- Pilihan ini dapat diabaikan saat ini dan dapat dilakukan pada lain kesempatan ketika Anda membutuhkannya dan komputer terhubung ke internet. Anda akan ditanya password user yang dibuat pada saat instalasi.
- Jika tidak ingin menginstal paket bahasa sekarang, tutup jendela itu dengan klik Close.



Gambar 14.10 Menawarkan tambahan dukungan bahasa

## 14.8 Troubleshooting <sup>10)</sup>

Dalam contoh di bawah diasumsikan user yang digunakan adalah user ubuntu dengan home direktori di /home/ubuntu. Pada tulisan ini digunakan istilah user administrator yang berarti user biasa yang diberikan akses untuk bisa menjalankan sudo atau gksudo di sistem.

### 14.8.1 Tidak bisa login sebagai root

- Diagnosa:

Di Ubuntu user root tidak diperbolehkan login.

- Solusi:

Tetapi anda bisa login dengan menggunakan perintah sudo, misalnya:

```
$ sudo su -
```

Anda juga bisa menjalankan perintah-perintah sebagai user root. Sebagai contoh:

```
$ sudo fdisk -l
```

Tetapi bila anda tetap ingin login sebagai root maka anda bisa lakukan dengan memberikan password untuk user root. Gunakan perintah:

```
$ sudo passwd root
```

Maka sekarang anda bisa login sebagai user root di terminal, tetapi untuk dari grafis anda tetap bisa login sebagai root.

### 14.8.2 Tidak bisa melakukan sudo

- Diagnosa:

Tidak semua user bisa melakukan sudo dan menjalankan program sebagai root. Anda harus masuk sebagai user administrator dengan masuk ke dalam group admin.

- Solusi:

Masuklah sebagai user administrator dan jalankan perintah :

```
$ sudo gpasswd -a ubuntu admin
```

### 14.8.3 Username dan password sudah benar tetapi tidak bisa login

- Diagnosa:

Cobalah login dari Virtual Console dengan menekan tombol Ctrl + Alt + F1. Bila gagal juga maka user atau password sudah tidak valid.

Masalah ini bisa dua kemungkinan:

- Bila anda bisa login sebagai user lain, maka user dan password anda tidak valid. Loginlah user yang lain. Kemudian cek apakah user anda ada atau tidak. Gunakan perintah:

```
$ grep ubuntu /etc/passwd
```

- Bila anda tidak bisa login sebagai user yang lain. Hal ini bisa disebabkan oleh kerusakan pada file-file yang berhubungan dengan login. File /etc/passwd, /etc/shadow, /etc/group atau /etc/gshadow hilang atau rusak. Biasanya disebabkan kecerobohan user yang dengan tanpa sengaja menghapus file-file tersebut.

Solusi:

- Untuk kasus pertama, loginlah sebagai user administrator dan login sebagai user root. Bila user anda telah ada maka ubahlah password user.

```
$ sudo passwd ubuntu
```

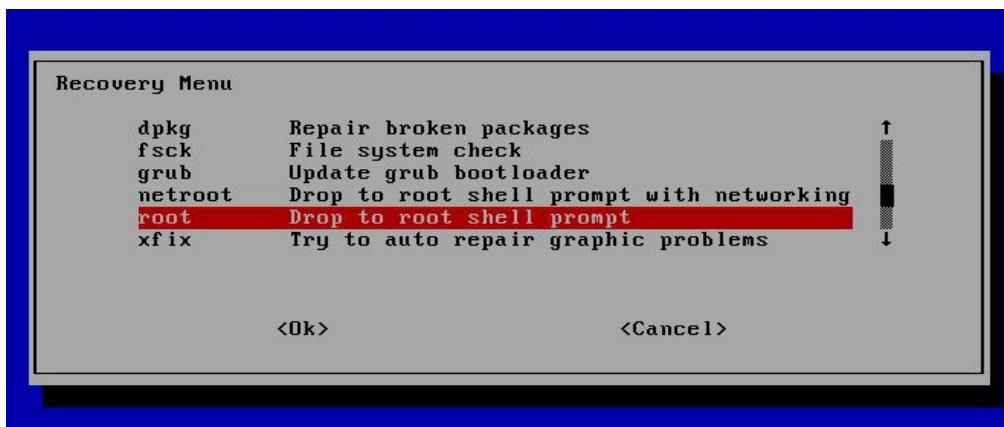
Bila belum ada buatlah user tersebut dan berikan password

```
$ sudo useradd -m ubuntu
```

```
$ sudo passwd ubuntu
```

- Untuk kasus kedua, anda harus merecovery file-file tersebut. Jangan khawatir, file-file ini memiliki backup sesuai dengan nama file-file tersebut ditambah dengan karakter "-". Seperti /etc/passwd-, /etc/group-, /etc/shadow-, dan /etc/gshadow-. Anda tinggal mengcopy ulang ke file aslinya.

Untuk melakukannya gunakanlah recovery mode Ubuntu yang dapat diakses dari GRUB dan pilih root.



Gambar 14.11 Tampilan menu recovery Ubuntu

- Pada terminal salinlah file-file backup tersebut ke file-file yang asli, perintahnya:

```
# cp /etc/passwd- /etc/passwd
# cp /etc/group- /etc/group
# cp /etc/shadow- /etc/shadow
# cp /etc/gshadow- /etc/gshadow
```

#### 14.8.4 Lupa password user administrator

- Solusi:

Restart Ubuntu anda dan masuk ke recovery mode Ubuntu. Pada menu recovery pilih root. Misalkan user administrator anda ubuntu. .Ubahlah password untuk user tersebut, dengan perintah:

```
# passwd ubuntu
```

#### 14.8.5 Program identifikasi software

Software adalah semua aplikasi dan perangkat lunak yang diinstall di komputer Anda. Setiap aplikasi yang diinstall di dalam sistem Anda membutuhkan paket-paket pendukung yang berbeda satu sama lain. Untuk mengatur ini semua di dalam Linux digunakan suatu aplikasi yang disebut package manager. Untuk Ubuntu dikarenakan merupakan keturunan dari distro

Debian maka digunakanlah dpkg.

Perintah dpkg ini dapat Anda gunakan untuk melihat software-software yang ada. Contoh penggunaannya adalah:

- Menampilkan list paket yang telah diinstall.

```
$ sudo dpkg -l
```

ii libterm-readkey-perl	2.30-3ubuntu1	A perl module for simple terminal control
ii libtext-charwidth-perl	0.04-4build1	et display widths of characters on the terminal
ii libtext-iconv-perl	1.4-3	converts between character sets in Perl
ii libtext-wrapi18n-perl	0.06-5	internationalized substitute of Text::Wrap
ii libthai-data	0.1.9-1	Data files for Thai language support library

- Mencari file-file paket yang mengandung suatu pola tertentu.

Penggunaan : \$ sudo dpkg -S <pattern>

```
$ sudo dpkg -S postfix
```

postfix-doc: /usr/share/doc/postfix/html/generic.5.html
postfix-dev: /usr/include/postfix/attr.h
postfix-dev: /usr/include/postfix/dict_sdbm.h
postfix-dev: /usr/include/postfix/mail_stream.h
postfix-doc: /usr/share/doc/postfix/html/postsuper.1.html
postfix-dev: /usr/include/postfix/header_body_checks.h

- Menampilkan list file dari suatu paket.

Penggunaan : \$ sudo dpkg -L <package\_name>

```
$ sudo dpkg -L squid
```

/etc/squid
/etc/logrotate.d
/etc/logrotate.d/squid
/etc/init.d
/etc/init.d/squid
/etc/resolvconf
/etc/resolvconf/update-libc.d
/etc/resolvconf/update-libc.d/squid

- Melihat status dari suatu paket.

Penggunaan : \$ sudo dpkg -s <package\_name>  
\$ sudo dpkg -s samba

```
Package: samba
Status: install ok installed
Priority: optional
Section: net
Installed-Size: 9204
Maintainer: Ubuntu Core Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: i386
Version: 3.0.28a-1ubuntu4
Replaces: samba-common (<= 2.0.5a-2)
```

- Melihat informasi suatu paket.

Penggunaan: \$ sudo dpkg -p pkg

```
$ sudo dpkg -p postfix
```

```
Description: High-performance mail transport agent
Postfix is Wietse Venema's mail transport agent that started life as an
alternative to the widely-used Sendmail program. Postfix attempts to
be fast, easy to administer, and secure, while at the same time being
sendmail compatible enough to not upset existing users. Thus, the outside
has a sendmail-ish flavor, but the inside is completely different
```

Selain menggunakan dpkg Anda juga bisa menggunakan aplikasi repository untuk melakukan pengecekan aplikasi-aplikasi yang ada di dalam komputer Anda atau yang ada dalam repository. Tool repository yang digunakan di dalam Ubuntu adalah APT (Advanced Packaging Tools). Contoh cara penggunaannya sebagai berikut:

- Mencari paket-paket tertentu yang berhubungan dengan kata yang dicari.

Penggunaan: \$ sudo apt-cache search <pattern>

```
$ sudo apt-cache search picture
```

```
exiv2 - EXIF/IPTC metadata manipulation tool
gnome-cards-data - data files for the GNOME card games
```

```
gnome-games-data - data files for the GNOME games
groff - GNU troff text-formatting system
hevea - translates from LaTeX to HTML, info, or text
kamera - digital camera support for KDE 4 applications
kipi-plugins - image manipulation/handling plugins for KIPi
aware programs
libdjvu-libre-dev - Development files for the DjVu image format
```

- Melihat semua paket dalam repository.

```
$ sudo apt-cache pkgnames
```

```
pipenightdreams
osgcal-doc
openoffice.org-110n-mn
mumudvb
mpg123-alsa
tbb-examples
libgammu4-dbg
libc6-xen
gcc-4.1-doc
snort-rules-default
davical
libevolution5.0-cil
aspell-am
python-gobject-doc
```

- Melihat informasi dari suatu paket.

Penggunaan: \$ sudo apt-cache show <package\_name>

```
$ sudo apt-cache show gimp
```

```
Package: gimp
Priority: optional
Section: graphics
Installed-Size: 12472
Maintainer: Ubuntu Desktop Team <ubuntu-
desktop@lists.ubuntu.com>
Original-Maintainer: Ari Pollak <ari@debian.org>
Architecture: i386
Version: 2.6.6-0ubuntul
```

```
Replaces: gimp-data (<< 2.3.17-2), gimp-gnomevfs (<< 2.6.0),
gimp-helpbrowser, gimp-libcurl (<< 2.6.0), gimp-print (<=
5.0.1-3), gimp-python (<< 2.6.0), gimp-svg, gimp-wget (<<
2.3.12-1), libgimp-perl (<= 2.0.dfsg+2.2pre1.dfsg-2)

Provides: gimp-helpbrowser, gimp-python
```

- Melihat dependensi dari suatu paket.

Penggunaan: \$ sudo apt-cache depends <package\_name>

```
$ sudo apt-cache depends gimp
```

```
gimp
```

```
  Depends: libgimp2.0
  Depends: libgimp2.0
  Depends: gimp-data
  Depends: gimp-data
  Depends: python-gtk2
  Depends: libaa1
  Depends: libatk1.0-0
  Depends: libbabl-0.0-0
  Depends: libc6
```

- Melihat reverse dependensi dari suatu paket.

Penggunaan: \$ sudo apt-cache rdepends <package\_name>

```
$ sudo apt-cache rdepends coreutils
```

```
coreutils
```

```
Reverse Depends:
```

```
| linux-image-2.6.28-16-virtual
| linux-image-2.6.28-16-server
| linux-image-2.6.28-16-generic
| linux-image-2.6.28-15-virtual
| linux-image-2.6.28-15-server
| linux-image-2.6.28-15-generic
| linux-headers-2.6.28-16-server
| linux-headers-2.6.28-16-generic
| linux-headers-2.6.28-16
| linux-headers-2.6.28-15-server
```

Selain itu Anda bisa menggunakan beberapa perintah lain yang dapat digunakan untuk melihat informasi dari suatu aplikasi:

- Mencari file-file program.

```
$ which gimp
/usr/bin/gimp
```

```
$ whereis gimp
gimp: /usr/bin/gimp /etc/gimp /usr/lib/gimp /usr/share/gimp
/usr/share/man/man1/gimp.1.gz
```

- Mencari library yang digunakan oleh suatu program.

Penggunaan: \$ ldd <file>

```
$ ldd /usr/bin/gimp
linux-gate.so.1 => (0x00130000)
libgimpwidgets-2.0.so.0 => /usr/lib/libgimpwidgets-2.0.so.0
(0x00133000)
libgimpmodule-2.0.so.0 => /usr/lib/libgimpmodule-2.0.so.0
(0x00251000)
libgimpcolor-2.0.so.0 => /usr/lib/libgimpcolor-2.0.so.0
(0x00256000)
libgimpmath-2.0.so.0 => /usr/lib/libgimpmath-2.0.so.0
(0x0026a000)
libgimpconfig-2.0.so.0 => /usr/lib/libgimpconfig-2.0.so.0
(0x0027d000)
libgimpbase-2.0.so.0 => /usr/lib/libgimpbase-2.0.so.0
(0x0027e000)
libgtk-x11-2.0.so.0 => /usr/lib/libgtk-x11-2.0.so.0
(0x00292000)
libgdk-x11-2.0.so.0 => /usr/lib/libgdk-x11-2.0.so.0
(0x00678000)
```

## Daftar Pustaka

1. Tobias Elsner, Thomas Erker, Anselm Lingnau, **Linux Essentials, The LPI Introductory Programme**. Darmstadt - Germany: Linup Front GmbH, 2012.
2. Jesús M. González Barahona, Joaquín Seoane Pascual, Gregorio Robles, **Introduction to Free Software**. Barcelona: Free Technology Academy, 2010.
3. Nah Soo Hoe, **Free/Open Source Software, Open Standard**, ([http://en.wikibooks.org/wiki/FOSS\\_Open\\_Standards](http://en.wikibooks.org/wiki/FOSS_Open_Standards)), United Nation Development Programme – Asia-Pacific Development Information Programme, 2006.
4. Lawrence Liang, **Free/Open Source Software, Open Content**, ([http://en.wikibooks.org/wiki/FOSS\\_Open\\_Content](http://en.wikibooks.org/wiki/FOSS_Open_Content)), United Nation Development Programme – Asia-Pacific Development Information Programme, 2007.
5. Gaurab Raj Upadhaya, **Free/Open Source Software, Network Infrastructure and Security**, ([http://en.wikibooks.org/wiki/FOSS\\_Network\\_Infrastructure\\_and\\_Security](http://en.wikibooks.org/wiki/FOSS_Network_Infrastructure_and_Security)), United Nation Development Programme – Asia-Pacific Development Information Programme, 2007.
6. Shun-ling Chen, **Free/Open Source Software, Licensing**, ([http://en.wikibooks.org/wiki/FOSS\\_Licensing](http://en.wikibooks.org/wiki/FOSS_Licensing)), United Nation Development Programme – Asia-Pacific Development Information Programme, 2006.
7. -, **Open Source Training Tool Kit - Free/Open Source Software**. Open Source Resource Center - Pakistan Software Export Board, 2007.
8. Moreno Muffatto, **Open Source – A Multidisciplinary Approach**. London: Imperial College Press, 2006.
9. Martin Fink, **The Business and Economics of Linux and Open Source**. New Jersey: Prentice Hall PTR, 2002.
10. Toto Harjendro & Tim Pengajar LP3T Nurul Fikri, **Ubuntu Troubleshooting**. Depok: IGOS Center, 2010.
11. [www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)
12. [en.wikipedia.org](http://en.wikipedia.org)
13. [www.fsf.org](http://www.fsf.org)
14. [www.opensource.org](http://www.opensource.org)