

Panduan Pendayagunaan Open Source Software :
Perangkat Lunak Bebas dan Open Source

PENGANTAR

Perkembangan Teknologi Informasi dan Komunikasi (TIK) yang sedemikian pesat telah membawa dampak yang positif dalam era peningkatan kinerja dari berbagai kegiatan yang dilakukan. Penerapan teknologi informasi dan komunikasi telah digunakan di hampir semua sektor, mulai dari transportasi, perdagangan, ekonomi, manufaktur, pemerintahan, pendidikan, kesehatan hingga pertahanan. Dengan kemajuan teknologi informasi dan komunikasi, informasi tidak lagi mengenal batas-batas negara, berbagai transaksi ekonomi, perbankan hingga komunikasi tatap muka, dengan siapa saja di belahan bumi manapun dapat berlangsung melalui teknologi ini.

Dua aspek penting dari rekomendasi para pakar di Dewan TIK Nasional (DETIKNAS) yang telah dihasilkan dalam mempercepat pembangunan teknologi informasi dan komunikasi di tanah air. Dua aspek penting tersebut adalah *leadership* dimana kepemimpinan dalam TIK menuntut komitmen dari para pemimpin pemerintahan, baik dari jajaran pemerintah pusat maupun pemerintah daerah. Aspek kedua adalah infrastruktur informasi nasional dimana aspek ini diperlukan dalam proses pengambilan keputusan yang senantiasa didasarkan pada informasi yang relevan, terkini atau *real times*, akurat dan dapat dipertanggungjawabkan.

Menyadari akan pentingnya serta semangat untuk membangkitkan inovasi pembangunan TIK di tanah air, sangatlah tepat apabila pendeklarasian program *Indonesia, Go Open Source!* (IGOS) adalah merupakan upaya dalam meraih kemandirian dalam pemanfaatan teknologi informasi dan komunikasi yang ditopang dengan tersedianya berbagai pilihan terhadap perangkat lunak (*software*) yang pada akhirnya dapat dipergunakan oleh masyarakat. Hal ini juga sesuai dengan salah satu Tugas dari DETIKNAS yaitu “*Merumuskan kebijakan umum dan arahan strategis pembangunan nasional, melalui pendayagunaan teknologi informasi dan komunikasi (TIK)*”.

Panduan Pendayagunaan Open Source Software ini tentunya diharapkan dapat menjadi salah satu solusi dalam perumusan kebijakan umum dan arahan strategis dalam penyiapan *blue print* dan *road map* teknologi informasi dan komunikasi di Indonesia. Dan dengan Buku Panduan ini dapat lebih memudahkan

masyarakat dalam mengenal perangkat lunak berbasis open source. Buku panduan ini terdiri atas delapan dokumen yaitu *Perangkat Lunak Bebas dan Open Source*, *Petunjuk Instalasi IGOS Nusantara*, *Konfigurasi Server Linux*, *Aplikasi untuk Server*, *RDBMS (Relational Database Management System): MySQL*, *Bahasa Pemrograman Open Source*, *Aplikasi Perkantoran OpenOffice.org*, dan *CMS, CRM, dan ERP*.

Akhir kata, semoga kedelapan Dokumen ini dapat bermanfaat bagi peningkatan kemampuan masyarakat dalam penguasaan teknologi piranti lunak bebas dan open source.

Dengan semangat “**Kreativitas Tanpa Batas**” dan maju bersama *Indonesia, Go Open Source!* Kita besarkan produk aplikasi perangkat lunak Indonesia.

Jakarta, 2008

Ketua Harian Detiknas

Muhammad Nuh

DAFTAR ISI

PENGANTAR.....	iii
DAFTAR ISI.....	v
I. APA ITU FREE/OPEN SOURCE SOFTWARE?.....	1
II. FILOSOSI FOSS.....	3
III. METODA PENGEMBANGAN FOSS.....	5
IV. SEJARAH FOSS.....	9
4.1. Sejarah singkat Gerakan FOSS.....	9
V. MENGAPA FOSS?.....	13
VI. KEUNTUNGAN MENGGUNAKAN FOSS.....	15
6.1. Keamanan (Security).....	15
6.2. Ketersediaan/Kestabilan (Reliability/Stability).....	16
6.3. Standar Terbuka dan Tidak Bergantung Vendor.....	16
6.4. Mengurangi Ketergantungan Impor.....	18
6.5. Pengembangan Perangkat lunak Lokal.....	18
6.6. Pembajakan, HaKI dan WTO.....	19
6.7. Bahasa dan Budaya Lokal (Localization).....	19
VII. KELEMAHAN FOSS.....	21
7.1. Aplikasi Bisnis belum lengkap.....	21
7.2. Interoperabilitas dengan Sistem Proprietary.....	21
7.3. Dokumentasi dan Tampilan	22
VIII. LISENSI FOSS DAN HAKI.....	21
8.1 Bagaimana software diproteksi?.....	24
8.2 Dasar-dasar Copyright (Hak Cipta).....	24
8.2.1 Apa saja yang dapat dilindungi hak ciptanya?.....	24
8.2.2 Apa yang dibutuhkan untuk mendapatkan hak cipta?.....	24
8.2.3 Hak apa yang diberikan kepada pemegang hak cipta?.....	25
8.3 Perluasan Perlindungan Hak Cipta.....	25
8.3.1 Legislasi pertama terhadap hak cipta (Statute of Anne, 1710).....	25

8.3.2	Perluasan proteksi hak cipta di berbagai bidang.....	25
8.4	Dari Perlindungan Nasional ke Internasional.....	26
8.4.1	Konvensi Berne.....	26
8.4.2	WTO dan TRIPs.....	26
8.4.3	Perlindungan pada kreasi, penghapusan syarat formalitas.....	27
8.4.4	Hukum hak cipta, keseimbangan antara kepentingan publik dan pribadi.....	22
8.5	Perangkat Lunak dan Proteksi Hak Cipta.....	28
8.5.1	Perluasan Hukum Proteksi Hak cipta pada perangkat lunak di tahun 1980.....	28
8.5.2	Hak cipta memproteksi kode sumber dan kode objek di bawah kesepakatan TRIPs.....	28
8.5.3	Hak Pemakai diabaikan dalam model-model lisensi proprietary	29
IX.	BAGAIMANA FOSS BERBEDA DENGAN PERANGKAT LUNAK PROPRIETARY ?.....	31
9.1.	Free Software (Perangkat Lunak Bebas).....	31
9.2.	Richard Stallman pada sebuah keputusan moral yang penting.....	31
9.3.	Definisi Free Software.....	32
X.	MEMBANGUN LINGKUNGAN FREE SOFTWARE.....	33
10.1.	Proyek GNU dan Free Software Foundation.....	33
10.2.	GNU General Public License (GNU GPL atau GPL).....	33
XI.	OPEN SOURCE SOFTWARE.....	35
11.1	Definisi Open Source.....	35
XII.	LISENSI-LISENSI YANG DIAKUI OSI.....	37
12.1.	Bebas atau Ketat?.....	37
12.2.	Bagaimana membuat kode sumber free/open?.....	38
12.3.	Perbandingan Lisensi FOSS v2.1.....	38
12.4.	GNU General Public License (GNU GPL atau GPL).....	43
12.4.1.	Copyleft.....	43
12.4.2.	Syarat dan Ketentuan Utama GPL.....	44
12.5.	GNU Lesser General Public License (GNU LGPL atau LGPL).....	45
12.5.1	Prasyarat dan ketentuan utama dari LGPL.....	46

12.6. Model Lisensi BSD (Berkeley Software Distribution).....	47
12.7. Lisensi Gabungan (Multiple Licensing).....	48
12.8. Kode sumber itu Terbuka; Bagaimana dengan Dokumentasi?.....	48
12.8.1. GNU Free Documentation License (GNU FDL atau FDL).....	48
12.8.2. Creative Commons Licenses.....	49
XIII. BEBERAPA SKENARIO.....	53
13.1. Pengguna Akhir (Individu / Bisnis / Pemerintahan).....	53
13.1.1. Isu-isu Hukum yang Terkait.....	54
13.2. Pengembang (Individu, Bisnis).....	55
13.2.1. Ketika Memulai Proyek Baru.....	56
13.2.2. Ketika Memodifikasi Modul yang Sudah Ada.....	57
13.2.3. Ketika Mengintegrasikan Modul-Modul FOSS yang Berbeda Menjadi Satu.....	59
13.3. Pengusaha/Produsen (Bisnis).....	60
13.3.1. Distribusi Sederhana.....	60
13.3.2. Distribusi dari Sistem Terintegrasi.....	61
13.3.3. Proyek yang di Sponsor Pemerintah.....	61
XIV. DAFTAR ISTILAH.....	65
14.1. Copyleft.....	65
14.2. Copyright (Hak cipta).....	65
14.3. Copyright-holder (Pemegang Hak cipta).....	65
14.4. Derivative works (Karya Turunan)	66
14.5. Distribution/Redistribution (Penyebaran/Penyebaran ulang).....	66
14.6. Fair Use (Penggunaan yang Adil).....	66
14.7. License (Lisensi atau Surat Izin).....	67
14.8. Multiple Licensing (Pemberian lisensi campuran)	67
14.9. Public Domain (Milik Umum).....	67
14.10. Source Code (Kode Sumber).....	67
14.11. Sub Licensing (Sub-lisensi).....	68
14.12. Warranty Disclaimer (Penolak Jaminan).....	68

I. APA ITU FREE/OPEN SOURCE SOFTWARE ?

Menurut David Wheeler, secara umum program yang dinamakan free software (perangkat lunak bebas) atau *open source software* (perangkat lunak sumber terbuka) adalah program yang lisensinya memberi kebebasan kepada pengguna menjalankan program untuk apa saja, mempelajari dan memodifikasi program, dan mendistribusikan penggandaan program asli atau yang sudah dimodifikasi tanpa harus membayar royalti kepada pengembang sebelumnya. (Sumber: http://www.dwheeler.com/off_fs_why.html).

Free/Open Source Software (FOSS) atau perangkat lunak bebas dan open source (PLBOS) telah menjadi sebuah fenomena internasional. Dalam beberapa tahun terakhir, FOSS mengalami perubahan besar dari sebuah kata yang relatif tidak dikenal menjadi sebuah kata populer terbaru. Namun, istilah FOSS tetap belum mudah dipahami mengingat FOSS merupakan konsep baru, misalnya apa saja pengertian FOSS dan apa saja cabang atau jenis-jenisnya.

Bab-bab selanjutnya berikut ini memberikan penjelasan yang baik tentang fenomena FOSS, filosofinya, perbedaannya dengan program yang bukan FOSS, dan metoda pengembangannya.

II. FILOSOSI FOSS

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.

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

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

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.

OSI difokuskan pada nilai-nilai teknis dalam pembuatan perangkat lunak yang berdaya guna dan dapat diandalkan, dan pendekatan istilah OSI ini lebih sesuai kebutuhan bisnis daripada filosofi FSF. OSI tidak terlalu fokus pada isu moral seperti yang ditegaskan FSF, dan lebih fokus pada manfaat praktis dari metoda pengembangan terdistribusi dari 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.

III. METODA PENGEMBANGAN FOSS

Model pengembangan FOSS adalah unik, dan menjadi sukses karena muncul bersamaan dengan berkembangnya internet dan efeknya yang luar biasa di bidang komunikasi. Analogi Katedral dan Bazar digunakan untuk membedakan model pengembangan FOSS (Bazar) dengan metode pengembangan perangkat lunak tradisional (Katedral).

Pengembangan perangkat lunak tradisional diibaratkan dengan cara 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. 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 bekerjanya selesai dan program dirilis ke publik. Sekali dirilis, program dianggap selesai, dan selanjutnya hanya ada pekerjaan terbatas untuk program itu.

Sebaliknya, pengembangan FOSS 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 keadaan 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 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.

Metoda pengembangan “bazar” telah dijadikan pilihan untuk mendapatkan beberapa kelebihan berikut ini:

- 1) **Mengurangi duplikasi kerja:** Dengan merilis program lebih cepat, dan

memberikan izin kepada pengguna untuk mengubah dan mendistribusi kode sumber, pengembang FOSS memanfaatkan kembali karya yang dihasilkan oleh compatriots. Skala ekonomi dapat menjadi sangat besar. Daripada 5 pengembang software pada sepuluh perusahaan mengembangkan aplikasi jaringan yang sama, ada potensi 50 pengembang melakukan secara bersamaan membentuk kerja sama kombinasi. Mengurangi duplikasi kerja akan membuat skala pengembangan FOSS menjadi besar, karena ribuan pengembang di seluruh dunia dapat bekerja sama.

- 2) **Membangun di atas karya lain:** Dengan ketersediaan kode sumber untuk membangun program, waktu pengembangan menjadi pendek. Banyak proyek FOSS berbasis program yang dihasilkan proyek lain untuk menambah fungsionalitas yang diperlukan. Sebagai contoh, proyek *server* web Apache lebih memilih memanfaatkan proyek 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 proyek lain memecahkan masalah yang sama.
- 3) **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 akses ke 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, dan dalam beberapa kasus pengguna dapat mengirimkan kode perbaikannya. Ini sangat mengurangi waktu pengembangan dan kontrol terhadap kualitas.
- 4) **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.

IV. SEJARAH FOSS

Gerakan FOSS dimulai dalam budaya “hacker” yang terjadi pada beberapa laboratorium ilmu komputer (Stanford, Berkeley, Carnegie Mellon, dan MIT) di tahun 1960-an dan 1970-an.

Komunitas pemrogram adalah kecil 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.

4.1. Sejarah Singkat Gerakan FOSS

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 Hobby). Dalam surat tertanggal 3 Februari 1976 itu ia mencemooh budaya berbagi perangkat lunak yang telah umum berlaku:

Mengapa ini? Hobbyists harus hati-hati, sebagian besar Anda mencuri perangkat lunak Anda. Perangkat keras harus dibeli, tetapi perangkat lunak menjadi sesuatu untuk dibagi. Siapa yang mau peduli jika orang yang bekerja untuk itu mengambil bayaran?

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, perusahaan ini akhirnya akan menghasilkan kreasi FSF dan budaya FOSS saat ini.

Richard Stallman, salah satu anggota laboratorium MIT saat itu, terkejut atas lanjutan peristiwa tersebut. Ini kemudian membentuk pandangannya terhadap

perangkat lunak *proprietary*, dan membangkitakan keinginannya untuk membuat sistem operasi yang *free* (bebas). Proyek GNU (*GNU is Not UNIX*) berdiri pada Januari 1984. Dalam dekade berikutnya proyek 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 proyek GNU. Namun, hingga 1991 proyek 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.

Ada beberapa proyek FOSS yang sedang berjalan dalam waktu bersamaan, antara lain server DNS BIND, bahasa pemrograman Perl, dan sistem operasi BSD. Sebagian besar proyek 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 milyar (sekitar Rp 43 triliun jika 1US\$ = Rp 9.000,-).

Sukses lain IPO perusahaan FOSS saat itu adalah VA Linux (US\$ 7 milyar atau Rp 63 triliun), Cobalt Networks (US\$ 3,1 milyar atau Rp 28 triliun), dan

Andover.net (US\$ 712 juta atau Rp 6,4 trilyun). Sebagai anak baru dari FOSS, kesuksesan GNU/Linux menunjukkan bahwa era FOSS telah benar-benar tiba.

V. MENGAPA FOSS?

Perangkat lunak *open source* telah disebut dengan beberapa istilah baik dan buruk, antara lain: gerakan, mode, virus, konspirasi komunis, hati dan jiwa dari internet. Tetapi ada satu poin yang sering dilupakan orang bahwa perangkat lunak *open source* juga merupakan kendaraan yang sangat efektif untuk mentransfer kekayaan dari dunia industri ke negara-negara berkembang. Itu diungkapkan Andrew Leonard dalam tulisannya “*An Alternative Voice: How the Tech-Poor Can Still Be Software Rich*” (bagaimana yang miskin teknologi dapat tetap menjadi kaya perangkat lunak).

Apakah FOSS selalu Free?

Mitos terkenal di seputar FOSS adalah selalu gratis, yang artinya tidak ada biaya sama sekali. Ini benar hanya untuk tingkatan tertentu, misalnya tidak perlu biaya izin untuk mendownload atau menggandakan, misalnya iso CD IGOS Nusantara atau Fedora. Mitos itu tidak benar untuk aplikasi FOSS yang membutuhkan biaya dalam pengemasan, instalasi, *support*, pelatihan, dan lain-lain. Banyak distro Linux seperti Red Hat, SUSE, Mandriva, Debian, Ubuntu, dan lain-lain dapat diperoleh tanpa biaya lisensi untuk mendownload-nya melalui internet. Dalam hal ada biaya lisensi, hampir semua biaya lisensi aplikasi FOSS lebih murah dibandingkan lisensi perangkat lunak proprietary.

Namun, biaya penggunaan FOSS tidak hanya biaya pemaketan atau infrastruktur. Ada juga biaya personal, biaya perangkat keras, biaya yang hilang (*opportunity costs*) misalnya karena peralihan, dan biaya pelatihan. Dengan menghitung biaya total kepemilikan atau TCO (*Total Costs of Ownership*), akan tergambar penghematan yang diperoleh jika menggunakan FOSS.

VI. KEUNTUNGAN MENGGUNAKAN FOSS

Di samping rendahnya biaya, ada beberapa alasan mengapa masyarakat, organisasi publik, atau bisnis secara agresif mengadopsi FOSS, antara lain:

- 1) Keamanan (*Security*)
- 2) Ketersediaan/Kestabilan (*Realibility/Stability*)
- 3) Standar terbuka dan tidak tergantung vendor
- 4) Mengurangi ketergantungan terhadap impor
- 5) Meningkatkan kemampuan mengembangkan perangkat lunak lokal
- 6) Pembajakan, HaKI (Hak atas Kekayaan Intelektual) dan WTO (*World Trade Organization*)
- 7) Bahasa dan budaya lokal (*localization*)

6.1. Keamanan (Security)

Meskipun tidak ada sistem operasi atau platform yang aman secara sempurna, faktor-faktor seperti metoda 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 dalam hal keamanan dibandingkan sistem proprietary:

- 1) 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 milyar (sekitar Rp 18 trilyun) hanya untuk mengatasi kelemahan IIS terkait Code Red.
- 2) “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 Windows NT.”

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

Tiga hal yang sering dijadikan alasan bahwa FOSS lebih aman:

- 1) Ketersediaan kode sumber (*open source*).
- 2) Lebih fokus kepada keamanan (*security*) daripada keindahan (*user friendly*).
- 3) Roots: Sebagian besar sistem FOSS berbasis multi user dan UNIX yang siap untuk jaringan.

6.2. Ketersediaan/Kestabilan (*Reliability/Stability*)

Sistem FOSS sangat dikenal dengan kestabilan dan ketersediaannya (tidak mudah *hang* atau minta *restart*). Ada banyak cerita anekdot, bahwa server FOSS bekerja baik bertahun-tahun tanpa minta perawatan khusus. Namun, studi kuantitatif terhadap hal itu tidak mudah dilakukan. Berikut ini dua studi yang terkait hal itu:

- 1) 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 pencetakan, 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.
- 2) 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 proyek 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.

6.3. Standar Terbuka dan Tidak Bergantung Vendor

Standar terbuka 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. Pada tahap selanjutnya pengguna proprietary sangat mengharapkan belas kasihan vendor jika semua datanya dalam format yang hanya dimiliki vendor dan kesulitan untuk mengubahnya menjadi standar terbuka.

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 mengatakan:

Alasan utama menentang implementasi *proprietary software* di sektor publik adalah ketergantungan terhadap vendor software tersebut. Bilamana standar proprietary telah digunakan secara mapan, keharusan untuk mengikutinya 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:

- 1) 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 mungkin membuat standar proprietary tersembunyi dalam sistem FOSS.
- 2) Sesuai dengan standar yang aktif: Jika ada standar aktif yang telah digunakan, misalnya HTML (*HyperText Markup Language*) yang mengontrol bagaimana web ditampilkan, proyek 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 tersedia saat ini yang paling mengikuti standar. Kesesuaian dengan standar adalah salah satu budaya gerakan FOSS, bahwa berbagi dan bekerja sama dengan aplikasi lain adalah hal biasa. Bekerja dalam kelompok pengembang global yang tersebar juga

menjadi lebih mudah jika ada standar yang menyertainya.

6.4. Mengurangi Ketergantungan Impor

Alasan utama yang mendorong negara-negara berkembang untuk 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 mata uang berharga dan cadangan devisa. Cadangan devisa ini lebih dapat digunakan untuk tujuan pengembangan lainnya.

Sebuah riset Eropa, *“Free/Libre and Open Source Software: Survery and Study”*, juga mencatat bahwa biaya dalam model 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.

6.5. 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:

- 1) Mudah masuk industri perangkat lunak: Mudah untuk mendapatkan, menggunakan, dan mempelajari FOSS karena kebebasannya untuk dimodifikasi dan didistribusikan. Perangkat lunak proprietary cenderung lebih membatasi, tidak hanya ketersediaan kode sumber program yang terbatas, tapi juga pembatasan terkait lisensi, paten, dan hak cipta. FOSS mendorong pengembang untuk membangun di atas pengetahuan dan komponen pra-pengembangan yang telah ada.
- 2) FOSS sebagai sistem pelatihan yang sempurna: Keterbukaan dan kolaborasi yang menjadi ciri khas FOSS mendorong siswa/mahasiswa untuk menguji dan mencoba konsep-konsep perangkat lunak, tanpa perlu biaya atau membebani masyarakat. Misalnya, siswa/mahasiswa dapat bergabung ke jaringan pengembangan FOSS yang saling bekerja sama secara global, yang telah dilengkapi arsip lengkap informasi teknis dan fasilitas diskusi interaktif.

- 3) FOSS sebagai sumber standar-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, “Proyek-proyek 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.”

6.6. Pembajakan, HaKI dan WTO

Pembajakan software menjadi masalah di hampir semua negara di dunia. Menurut studi pembajakan software global yang dibuat IDC dan BSA (*Business Software Alliance*), pembajakan software pada 2005 menimbulkan kerugian US \$34 milyar (sekitar Rp 300 triliun). 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.

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.

6.7. Bahasa dan Budaya Lokal (*Localization*)

“Lokalisasi 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*).

Lokalisasi 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 dan IGOS Nusantara.

VII. KELEMAHAN FOSS

Dengan berbagai kelebihanannya, FOSS saat ini belum tentu cocok untuk semua keadaan. Ada beberapa bidang yang masih membutuhkan penyempurnaan produk FOSS.

7.1. Aplikasi Bisnis belum lengkap

Meskipun saat ini telah terdapat banyak proyek FOSS dalam berbagai tingkatan pengembangan, tetap masih ada bidang yang belum tersedia produk dengan fitur lengkap, khususnya di dunia bisnis. Rilis produk ERP (Enterprise Resource Planning) seperti SAP dan Peoplesoft telah membantu pasar aplikasi high-end, tetapi pasar bisnis kecil dan menengah atau SME (Small and Medium Enterprise) masih kurang dilayani. Sebagai contoh, aplikasi akuntansi populer di Windows seperti Quickbooks, Peachtree, atau Great Plains belum memiliki penggantinya di Linux.

Masalah itu muncul karena sangat sedikit orang yang menguasai dua bidang teknis dan bisnis sekaligus. Para pengembang teknis lah yang telah memulai hampir semua proyek FOSS saat ini untuk menyelesaikan masalah dan menulis program untuk memenuhi keinginannya (*scratch an itch*). Proyek-proyek ini secara alami biasanya lebih mengarah pengembangan aplikasi teknis, seperti server web, bahasa pemrograman, dan aplikasi atau tool untuk jaringan. Sangat jarang pengembang software menghadapi masalah akuntansi, misalnya, pada saat yang sama ingin mendapatkan pengetahuan bisnis untuk menghasilkan solusi teknis.

7.2 Interoperabilitas dengan Sistem Proprietary

Sistem FOSS, khususnya untuk desktop, tidak sepenuhnya kompatibel dengan sistem proprietary. Bagi organisasi yang telah berinvestasi secara besar-besaran pada software proprietary dan format penyimpanan data, mencoba mengintegrasikannya solusi FOSS dapat menjadi pilihan yang mahal. Mengubah standar proprietary akan membuat lebih buruk persoalan ini, karena standar proprietary sering tidak ditujukan untuk berintegrasi dengan solusi alternatif.

Pada saatnya, seiring dengan pergeseran organisasi dari menggunakan standar

proprietary ke standar terbuka, masalah seperti itu harus dikurangi.

7.3. Dokumentasi dan Tampilan

Sistem atau aplikasi FOSS yang telah ada saat ini masih banyak yang lemah dalam hal kelengkapan dokumentasi dan kemudahan bagi pengguna (user-friendliness), dibandingkan dengan yang ditemukan dalam program proprietary. Fokus utama para pengembang awal FOSS adalah kegunaanya (fungsionalitas). Membuat program yang dapat bekerja dengan baik jauh lebih penting daripada kemudahan dan tampilan.

Selain kekurangan dokumentasi bermutu tinggi, juga ada isu seputar GUI (Graphical User Interface) dari FOSS. Karena elemen GUI pada banyak sistem FOSS tidak berupa elemen tunggal tapi merupakan koleksi proyek-proyek berlainan yang menyatu secara bersamaan. Perintah untuk menyimpan data berbeda untuk satu program dengan program lainnya, tidak seperti pada sistem proprietary Macintosh atau Windows. Pemotongan (*cut*) dan penempelan (*paste*) pada program-program berbeda dapat menjadi tidak konsisten, atau bahkan bisa tidak dimungkinkan. Meskipun ada usaha-usaha yang signifikan untuk menyatukan desktop, desktop di lingkungan FOSS kelihatannya masih belum konsisten untuk beberapa waktu yang akan datang.

VIII. LISENSI FOSS DAN HAKI

Belakangan ini, produk dari aktivitas kreatif manusia yang tidak terukur secara nyata (*intangible*) dapat dinilai sebagai jenis kekayaan dan diproteksi/dilindungi dalam bentuk yang terukur secara nyata (*tangible*). Secara umum ide hak atas kekayaan intelektual (HaKI) atau *intellectual property rights* dapat diterima, dan institusi hukum telah didirikan untuk memberikan perlindungan terhadap para pemilikinya. Meskipun hak cipta (*copyrights*), paten, merek dagang, dan rahasia dagang termasuk dalam kelompok besar kekayaan intelektual, masing-masing memiliki arti yang berbeda. Berikut ini penjelasannya:

Rahasia Dagang (*Trade Secrets*)

Rahasia dagang dilindungi dari akses oleh pesaing bisnis. Perlindungan ini dapat dilakukan dalam berbagai bentuk, misalnya perjanjian rahasia atau NDA (Non Disclosure Agreement) yang ditandatangani oleh mereka yang diberi akses terhadap suatu pengetahuan atau informasi.

Merek Dagang (*Trademarks*)

Merek dagang adalah nama, frasa, simbol, desain, gambar, atau gaya khusus yang digunakan oleh sebuah bisnis untuk mengidentifikasikan dirinya dan produk atau jasanya kepada pelanggannya. Di banyak negara, warna, merek berbentuk tiga dimensi, suara, dan bahkan bau, juga mendapat perlindungan sebagai merek dagang.

Paten (*Patents*)

Berbeda dengan rahasia dagang yang merahasiakan informasi tertentu dari publik, paten memberikan monopoli kepada penemunya terhadap setiap pengetahuan baru yang dikembangkan dari paten itu untuk periode waktu tertentu (misalnya 20 tahun) sebagai imbalan atas pengungkapan temuannya. Biasanya, untuk mendapatkan hak eksklusif itu pemilik paten harus mendaftarkan temuannya ke suatu badan yang akan melakukan pengujian terhadap penemuannya. Hal yang baru (*novelty*) dari suatu penemuan merupakan elemen penting dalam pemberian paten.

Hak Cipta (*Copyrights*)

Hak cipta diterapkan terhadap berbagai bentuk hasil kreasi manusia, misalnya tulisan sastra, komposisi musik, lukisan dan perangkat lunak (software). Pemegang hak cipta memiliki hak eksklusif terhadap ciptaannya dalam hal reproduksi, modifikasi, distribusi, dan publikasi atau pengumuman kepada publik. Tidak seperti paten yang dilindungi dari sisi idenya, hak cipta dilindungi dari sisi hasil kreasinya. Ide untuk menghasilkan kreasi itu (misalnya algoritma software) tidak dilindungi oleh hukum hak cipta.

8.1. Bagaimana software diproteksi?

Software sekarang diproteksi atau dilindungi oleh hukum hak cipta. Meskipun belakangan ini ada usulan agar kode sumber dan algoritma dapat dipatenkan, dan telah disetujui untuk beberapa kasus, paten software tetap dipertanyakan dan ditentang oleh banyak pihak, terutama dari komunitas FOSS. Tutorial ini hanya fokus kepada lisensi FOSS, sedangkan paten software tidak banyak dibahas di sini.

8.2. Dasar-dasar Copyright (Hak Cipta)

8.2.1. Apa saja yang dapat dilindungi hak ciptanya?

Hak cipta melindungi ekspresi ide dalam berbagai bentuk karya yang berbeda, termasuk artistik, drama, sastra, musik, dan karya intelektual lainnya. Sejak 1980-an, program atau software telah dilindungi oleh hukum hak cipta. Ide untuk mengekspresikan karya itu tidak diproteksi.

8.2.2. Apa yang dibutuhkan untuk mendapatkan hak cipta?

Saat ini, hukum hak cipta tidak memerlukan formalitas untuk mendapatkan proteksi hak cipta. Pengarang tidak harus mengumumkan, mendaftarkan, membayar registrasi dan tidak harus melampirkan catatan hak cipta pada ciptaannya. Hak cipta suatu karya kreatif secara otomatis menjadi milik penciptanya, yaitu orang yang menghasilkan karya cipta itu. Perlindungan hak cipta otomatis diberikan kepada suatu karya yang telah dihasilkan orang.

8.2.3. Hak apa yang diberikan kepada pemegang hak cipta?

“Hak cipta” merupakan hak hukum eksklusif yang diberikan pemerintah kepada pengarang (kreator) yang telah menghasilkan sebuah karya kreatif. Hak cipta ini meliputi hak mereproduksi, hak menyiapkan kreasi turunan berbasis kreasi awal, hak mendistribusikan hasil penggandaan, hak menampilkan atau mengumumkan hasil karya kepada publik dan hak-hak lain yang dirumuskan dalam undang-undang hak cipta. Tanpa persetujuan tertulis pemegang hak cipta, siapapun tidak legal jika melanggar salah satu ketentuan dari sebuah karya cipta. Perlindungan hak cipta telah diperluas secara besar-besaran sesuai perjalanan waktu.

8.3. Perluasan Perlindungan Hak Cipta

8.3.1. Legislasi pertama terhadap hak cipta (Statute of Anne, 1710)

Jika dibandingkan dengan institusi hukum lainnya, hukum hak cipta relatif lambat dalam kehidupan manusia. Perundang-undangan hak cipta yang pertama dikenal adalah Statute of Anne, yang diundangkan pada 1710 di Inggris. Untuk setiap karya baru, Statute of Anne melindungi hak pemegang hak cipta untuk mencetak dan cetak ulang buku dan karya tulis lainnya selama 14 tahun.

8.3.2. Perluasan proteksi hak cipta di berbagai bidang

Pada awalnya, cakupan perlindungan hak cipta dari Statute of Anne masih terbatas. Karya yang dilindungi terbatas untuk buku dan karya tulis lainnya. Hak yang diberikan juga masih terbatas pada pencetakan dan pencetakan ulang, dengan lama perlindungan sampai dengan 14 tahun.

Saat ini, hukum hak cipta memberi perlindungan jauh lebih luas dari itu. Karya yang dilindungi saat ini mencakup arsitektur, komposisi musik, perekaman musik, lukisan, ukiran dan software. Hak yang diberikan kepada pemegang hak cipta saat ini meluas dari hanya cetak-mencetak menjadi pengubahan atau pemodifikasian, display atau tayangan untuk publik dan pendistribusian karya cipta. Usia hak cipta juga ditambah hingga 50 tahun setelah pencipta meninggal, bahkan di Eropa dan Amerika hingga 70 tahun. UU Republik Indonesia tentang Hak Cipta No. 19 Tahun 2002 menyebutkan hak cipta program komputer berlaku selama 50 (lima puluh) tahun sejak pertama kali diumumkan.

8.4. Dari Perlindungan Nasional ke Internasional

8.4.1. Konvensi Berne

Di akhir abad ke-19, ketika karya-karya yang dilindungi hak cipta secara bertahap menjadi bagian penting dalam perdagangan internasional, perlindungan hak cipta trans-nasional secara bertahap menjadi isu yang serius. Diawali oleh negara-negara Eropa sebagai penanda tangan, Konvensi Berne 1886 pertama mengenalkan prinsip-prinsip perlakuan hak cipta secara nasional. Meskipun Konvensi Berne menghendaki para penanda tangannya untuk menegakkan hukum perlindungan hak cipta dasar, prinsip perlakuan nasional memberi perlindungan karya cipta pemegang hak cipta asing sama dengan perlindungan terhadap pemegang hak cipta dari dalam negeri. Konvensi Berne telah menghasilkan standar internasional perlindungan hak cipta.

Namun, tanpa melalui mekanisme resolusi dengan perdebatan, perlindungan yang diberikan Konvensi Berne relatif lemah. Terlalu mahal bagi Anda pemegang hak cipta untuk mengklaim hak-haknya di negara lain, sementara Anda percaya bahwa hak Anda telah dilanggarnya.

8.4.2. WTO dan TRIPs

Pada tahun 1990-an, WTO (*World Trade Organization*) dan perjanjian TRIPs (*Trade Related aspects of Intellectual Property rights*) telah menjadi struktur yang kuat untuk perlindungan hak cipta internasional. Semua anggota WTO diharuskan ikut tanda tangan perjanjian TRIPs, dan setiap penanda tangan TRIPs harus setuju dengan semua bagian penting dari Konvensi Berne. WTO juga menyediakan mekanisme penyelesaian sengketa dan pelanggaran di antara negara-negara anggota. Sekarang perlindungan hak cipta internasional menjadi standar yang lebih bisa dilaksanakan.

8.4.3. Perlindungan pada kreasi, penghapusan syarat formalitas

Sebagai kelanjutan prinsip Konvensi Berne 1908, perlindungan hak cipta diberikan kepada karya pada saat pertama kali diciptakan, tanpa memerlukan formalitas. Pencipta tidak perlu mendaftarkan hak ciptanya dan mengumumkan karyanya hanya untuk mendapatkan perlindungan hak cipta secara penuh.

Meskipun cakupan dan aturan hukum hak cipta di setiap negara berubah sesuai perkembangan waktu, tetapi bagi penanda tangan Konvensi Berne berlaku ketentuan bahwa hukum hak cipta menganggap pencipta telah mengklaim semua hak cipta atas ciptaannya, kecuali pencipta itu menyatakan lain.

Dengan standarisasi perlindungan hak cipta internasional, hukum hak cipta di berbagai negara harus direvisi agar sesuai dengan standar itu. Sebagai contoh, Amerika Serikat telah merevisi undang-undang hak ciptanya dan menghapuskan syarat formalitas pada 1976 agar dapat bergabung dengan Persatuan Konvensi Berne.

8.4.4. Hukum hak cipta, keseimbangan antara kepentingan publik dan pribadi

Jika dibandingkan dengan istitusi hukum lainnya, perlindungan hak cipta relatif merupakan penemuan baru dalam sejarah kehidupan manusia. Dalam perjalanannya, pengembangan regulasi hak cipta juga menunjukkan transformasi sosial dan teknologi, sesuai aktivitas kreatif manusia dan distribusinya. Di satu sisi, pemberian hak pribadi kepada pemegang hak cipta secara eksklusif dapat meningkatkan aktivitas kreatif manusia, hukum hak cipta juga menghambat kepentingan publik yang lebih luas, terutama di dunia pendidikan, penelitian, dan akses terhadap informasi.

Bervariasi ukuran juga telah diadopsi untuk mendapatkan keseimbangan antara kepentingan publik dan pribadi. Dalam Statute of Anne, hukum menyatakan bahwa penguasa dapat membatasi dan menetapkan harga buku cetak, sesuai dengan keputusan yang paling adil. Dalam perundang-undangan Amerika yang diputuskan oleh Kongres, hak eksklusif yang diberikan terhadap penulisan dan penemuan memiliki jangka waktu tertentu. Hak eksklusif yang diberikan oleh hukum juga tidak boleh menghambat kemudahan akses publik terhadap informasi dan ilmu pengetahuan, dalam batas-batas tertentu. Misalnya, penggunaan yang adil (*fair use*) dan doktrin penjualan pertama (*first sale doctrine*) dapat meniadakan hak pribadi eksklusif dalam jangka waktu tertentu itu.

8.5. Perangkat Lunak dan Proteksi Hak Cipta

8.5.1. Perluasan Hukum Proteksi Hak cipta pada perangkat lunak di tahun 1980

Undang-undang Komputer Amerika Serikat 1980 yang berhubungan dengan program komputer menyatakan bahwa hak cipta sebagai suatu karya yang dapat dilindungi oleh hukum . Sejak saat itu menjadi tren bahwa proteksi hak cipta dapat diaplikasikan ke program komputer juga. Kesepakatan WIPO (1996) juga menyatakan bahwa perangkat lunak komputer juga sah dilindungi oleh hukum hak cipta.

8.5.2. Hak cipta memproteksi kode sumber dan kode objek di bawah kesepakatan TRIPs

Perangkat Lunak dapat ditemukan dalam bentuk kode sumber dan kode objek. Sementara ide-ide yang diekspresikan dalam bentuk kode sumber dapat dimengerti oleh pemrogram terlatih, sedangkan dalam bentuk kode objek tidak dapat diartikan oleh manusia. Seperti yang tercantum dalam TRIPs (*Trade Related Aspects of Intellectual Property Rights*) bahwa hak cipta untuk perangkat lunak dapat diaplikasikan baik dalam bentuk kode sumber maupun kode objek, maka pada tataran praktis biasanya perusahaan perangkat lunak cenderung hanya mengeluarkan produk mereka dalam bentuk kode objek, dan menyimpan kode sumbernya sebagai rahasia dagang.

Hukum hak cipta hanya memproteksi pengekspresian ide-ide, tapi tidak memproteksi ide itu sendiri. Dalam menikmati sebuah karya intelektual, komposisi sebuah karya musik, ide yang ada di dalamnya dapat dipahami atau bahkan menjadi inspirasi untuk karya karya lain, dan karenanya berkontribusi langsung pada perkembangan intelektual umat manusia secara keseluruhan. Akan tetapi ketika perangkat lunak hanya didistribusikan dalam bentuk kode objek yang dilindungi oleh hak cipta, artinya perusahaan proprietary dapat menikmati perlindungan ini tanpa harus berbagi ide. Cara membuat perangkat lunak bersangkutan tidak dapat diketahui, bahkan oleh pengembang terlatih, apalagi oleh masyarakat umum. Perlindungan hak cipta dalam hal ini tidak sejalan dengan inti sari hukum hak cipta yaitu mendapatkan keseimbangan antara kepentingan pribadi dengan kepentingan masyarakat.

8.5.3. Hak Pemakai diabaikan dalam model-model lisensi proprietary

Dalam lisensi-lisensi proprietary tradisional, kode sumber tidak dapat diketahui. Bahkan lisensi-lisensi proprietary mencegah para pengembang untuk mempelajari perangkat lunaknya. Lisensi untuk para pengembang contohnya, seperti Microsoft End User Agreement, dan pendaftaran ke Microsoft Developer Network, tidak mengizinkan *reverse-engineering* (rekayasa-pembalikan, membalik proses produksi dari produk ke ide dasar), kompilasi ulang, dan membongkar ulang menjadi bahasa assembly. Kecuali dan hanya untuk hal-hal tertentu yang diizinkan oleh hukum yang diterapkannya.

Untuk pemakai-akhir, biasanya lisensi proprietary hanya mengizinkan satu kopi hanya untuk satu komputer atau setiap processor. Artinya kalau anda punya satu buah *laptop* dan sebuah *Desktop*, atau dua buah desktop, anda perlu membeli dua lisensi untuk menjalankannya secara legal di kedua mesin anda. Bila kemudian program anda terdapat bug (kesalahan) maka anda tidak dapat memperbaikinya sendiri karena anda tidak memiliki kode sumbernya dan tidak diizinkan untuk mempelajarinya. Anda hanya bisa pasif memberitahu dan menunggu perbaikan dari perusahaan pembuatnya. Bahkan ketika perbaikan sudah diberikan, anda tidak dapat memastikan bahwa masalah sudah terselesaikan atau hanya menutupi efek masalah tanpa menyelesaikan masalah sebenarnya. Menggunakan perbaikan dari pihak ketiga juga tidak diizinkan karena mengubah program tidak diizinkan.

Di lain pihak gerakan FOSS telah berkontribusi positif pada transformasi seperti itu. Seperti dinyatakan oleh *Free Software Foundation* (FSF) ketika didirikan pada 1985, bahwa FSF memperjuangkan hak pemakai dalam menggunakan, mempelajari, memperbanyak, mengubah dan menyebarkan program komputer. Hak-hak itu secara serius telah dilanggar atau diabaikan oleh lisensi-lisensi tradisional yang ada.

IX. BAGAIMANA FOSS BERBEDA DENGAN PERANGKAT LUNAK PROPRIETARY?

Pengembangan FOSS dijalankan sebagai sebuah reaksi dari komunitas para pengembang perangkat lunak terhadap institusi hukum dalam konteks hak cipta. Kedua gerakan *Free* dan *Open Source* mensyaratkan akses ke kode sumber. Dan lebih jauh hak-hak lain juga dimasukkan dalam hak cipta, seperti hak untuk memperbanyak, menyebarkan, dan membuat turunan dan perbaikan.

9.1. Free Software (Perangkat Lunak Bebas)

Latar belakang: Transisi dalam industri teknologi informasi dan institusi hukum.

Pada tahun 1970-an perubahan pada institusi hukum dalam industri teknologi informasi berakibat pada terbentuknya *perangkat lunak bebas*. Amerika pada saat itu merevisi hukum hak cipta-nya, dan perusahaan IT terus mengklaim perangkat lunak sebagai barang yang sesuai dengan hukum hak cipta. Di lain pihak saat itu terjadi pemisahan antara perangkat lunak dan perangkat keras. Bila sebelumnya perangkat lunak merupakan bagian dari mesin, maka mulai saat itu perangkat lunak dapat dijual sebagai paket terpisah. Perusahaan perangkat lunak mulai merekrut para pengembang dari laboratorium-laboratorium pada institusi-institusi riset dan meminta mereka menandatangani perjanjian kerahasiaan.

9.2. Richard Stallman pada sebuah keputusan moral yang penting

Sebelum masa itu, berbagi kode sumber dan saling mengcopi merupakan sebuah praktek umum di laboratorium-laboratorium. Bagi Richard Stallman, yang pada waktu itu bekerja pada laboratorium di Massachusetts Institute of Technology (MIT), perubahan seperti itu di industri teknologi informasi meniadakan komunitas, yang memiliki budaya berbagi dan nilai etika saling bisa menolong, komunitas dimana dia bangga menjadi bagian darinya. Sebagai seorang pemrogram yang sangat berbakat dia dengan mudah bisa menandatangani perjanjian kerahasiaan dengan perusahaan perangkat lunak dengan imbalan gaji besar, Stallman dihadapkan pada sebuah keputusan moral dimana dia bisa memilih menandatangani kontrak atau berusaha mempertahankan komunitas tempatnya berasal. Akhirnya dia memilih yang terakhir dan membuat

infrastruktur untuk menjaga komunitas perangkat lunak bebas (free software community).

9.3. Definisi Free Software

Free software artinya perangkat lunak yang menjamin kebebasan penggunanya untuk menjalankan, mengcopy, menyebarkan, mempelajari, mengubah dan mengembangkannya. Stallman mendefinisikan Free Software sebagai berikut:

- 1) Kebebasan menjalankan program, untuk tujuan apapun (kebebasan 0).
- 2) Kebebasan untuk mempelajari bagaimana program bekerja dan mengubahnya agar sesuai kebutuhan anda (kebebasan 1). Ketersediaan kode sumber merupakan prasyarat untuk ini.
- 3) Kebebasan untuk menyebarkan salinan sehingga anda dapat menolong tetangga anda (kebebasan 2).
- 4) Kebebasan untuk memperbaiki dan mengembangkan program, dan merilisnya ke masyarakat sehingga semua komunitas dapat mengambil manfaatnya (kebebasan 3). Akses pada kode sumber juga merupakan prasyarat untuk hal ini.

Selain pernyataan yang tegas tentang ketersediaan kode sumber, defnisi perangkat lunak bebas juga menyebut hak pengguna untuk memperbanyak, menyebarkan, memodifikasi program dan mendistribusikan hasil turunannya.

X. MEMBANGUN LINGKUNGAN FREE SOFTWARE

10.1. Proyek GNU dan Free Software Foundation

Definisi Free Software secara tegas melindungi hak dari masyarakat yang tidak memiliki hak cipta. Tapi tanpa lingkungan yang mendukung ide ini maka keempat kebebasan tentang software menjadi tidak realistis. Proyek GNU diluncurkan pada tahun 1984 untuk membuat sistem operasi UNIX-base yang berlisensi *free software* (perangkat lunak bebas) yang diberi nama sistem GNU. Bahkan aplikasi di atas sistem operasinya pun menjad bagian dari proyek GNU.

Pada 1985 Free Software Foundation didirikan untuk mengusung ide perangkat lunak bebas. FSF mempromosikan pengembangan dan pemakaian dengan menyebarkan program yang bebas tersedia. FSF juga berkonsentrasi pada pengembangan dengan tujuan untuk mencapai sistem yang mapan sebagai alternatif pengganti proprietary software. Penjelasan lebih detil tentang ini dapat diakses dari <http://www.gnu.org> dan <http://www.fsf.org>.

10.2. GNU General Public License (GNU GPL atau GPL)

Di bawah struktur hukum yang ada, sekali karya diciptakan, perlindungan hak cipta diberikan secara eksklusif kepada pemegang hak cipta. Tanpa pernyataan yang eksplisit pemegang hak cipta mengklaim semua haknya.

Orang yang tidak tahu bagaimana membuat pernyataan eksplisit, dapat menggunakan GNU General Public License (GNU GPL atau GPL) telah secara legal, yang di dalamnya disebutkan hak-hak pengguna secara eksplisit. GPL juga menyatakan batasan-batasan agar lingkungan Free Software tetap terjaga.

GPL adalah lisensi, hanya saja isinya sangat berbeda dengan lisensi proprietary. Di dalamnya dia memberikan hak-hak pemakai yang biasanya tidak diberikan oleh lisensi komersial oleh perusahaan perangkat lunak. Di dalamnya termasuk hak untuk mengakses kode sumber, hak untuk menjalankan program, hak untuk memperbanyak dan menyebarkan, hak untuk mengubah program dan menyebarkan hasil modifikasinya. Di lain pihak meskipun GPL memberikan hak-hak pemakai, GPL juga membatasi hal-hal tertentu untuk menjamin bahwa program yang memakai GPL dan turunannya tetap perangkat lunak bebas.

Turunan GPL harus GPL.

Ketika suatu karya memakai lisensi GPL, itu artinya penulis program tetap memiliki hak cipta dari karyanya, tapi juga dapat mengambil lisensi lain untuk memberi kebebasan yang lebih besar kepada pengguna daripada yang dinyatakan oleh hukum hak cipta.

XI. OPEN SOURCE SOFTWARE

Sementara kebanyakan para pendukung Free Software menitikberatkan kebebasan sebagai sebuah masalah moralitas, maka ada sebagian yang lain yang lebih mementingkan tentang “*Open Source Software*” dimana lebih terfokus pada nilai-nilai teknis dan lebih bersahabat dengan bisnis dan industri. OSI (*Open Source Initiative*) menjalankan organisasi untuk mempromosikan atau kampanye Open Source, dengan mengelola dan promosi Definisi Open Source, dan sertifikasi terhadap lisensi dan produk yang open source (kode terbuka).

11.1. Definisi Open Source

Definisi Open Source adalah revisi dari dokumen kebijakan milik distribusi Debian GNU/Linux. Dokumen ini menjelaskan lisensi-lisensi mana saja yang termasuk bebas. OSI menjelaskan ide dasar dari open source:

“Ide dasar dari open source sangat sederhana: Bilamana pemrogram dapat membaca, menyebarkan, dan memodifikasi kode sumber dari sebuah perangkat lunak, maka perangkat lunak itu akan berkembang. Masyarakat memakai, membuatnya lebih baik dan memperbaiki kelemahannya.”

Open Source juga menekankan hak-hak yang tersebut dalam definisi Free Software, termasuk hak akses terhadap kode sumber, memperbanyak, menyebarkan, memodifikasi dan menyebarkan hasil turunannya.

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

XII. LISENSI-LISENSI YANG DIAKUI OSI

Dengan arahan yang tersurat pada definisi Open Source, OSI mengakui dan mensertifikasi lisensi-lisensi sebagai lisensi Open Source setelah melalui beberapa tahap. Tahap-tahap sertifikasi dilakukan sesuai permintaan, dan jika suatu lisensi baru saja diakui sebagai Open Source maka ia akan masuk daftar lisensi Open Source yang dipelihara oleh OSI.

Jumlah lisensi yang diakui OSI tumbuh bersamaan dengan pengembangan FOSS di berbagai area. Ada lisensi yang berasal dari komunitas FOSS, misalnya GPL, Lesser General Public License (LGPL), lisensi PHP, lisensi Nethack GPL; ada juga lisensi-lisensi dari akademik atau institusi riset, contoh: *NASA Open Source Agreement*, lisensi MIT, lisensi Universitas Illinois/*NCSA Open Source License*; selain itu juga berasal dari perusahaan yang memakai FOSS sebagai strateginya, di antaranya ada *Apple Public License*, *Eclipse Public License*, *Qt Public License*, dan *Mozilla Public License*. Sebagian besar dari lisensi-lisensi yang diakui oleh OSI berasal dari perusahaan.

12.1. Bebas atau Ketat?

Meski sekilas definisi Free Software dan definisi Open Source sama, sebetulnya pada tingkat retorika berbeda, yang mana menggambarkan perbedaan di tingkat filosofi.

Beberapa orang ada yang menyebut lisensi klasik perangkat lunak bebas seperti GPL dan LGPL sebagai “sangat ketat” atau “ketat”, karena didalamnya mendefinisikan batasan-batasan yang ketat dalam mencegah supaya perangkat lunak bisa bebas dan turunannya tetap bebas. Bagi FSF sendiri batasan ini memang prasyarat untuk membuat suatu lingkungan yang baik untuk perangkat lunak bebas, maka tidak sepatasnya anggapan bahwa lisensi ini “sangat ketat” atau “ketat”.

Meskipun berbeda, baik FSF maupun OSI sepakat satu sama lain bahwa pada banyak kasus lisensi-lisensi ini dapat diklasifikasikan sebagai lisensi FOSS.

Di antara puluhan lisensi yang diakui OSI hanya dua lisensi yaitu lisensi Original Artistic dan Original Artistic License yang dicap sebagai lisensi non-free (lihat

tabel berikut).

12.2. Bagaimana membuat kode sumber free/open?

Di bawah institusi legal yang ada saat ini, perangkat lunak dilindungi oleh hukum hak cipta. Gerakan FOSS menggunakan institusi legal. FOSS memberikan hak-hak pengguna dari hak-hak yang sepenuhnya milik pemegang hak cipta. Lisensi FOSS membuat pemberitahuan yang lebih mudah dapat juga dipandang sebagai kesepakatan dengan komunitas pengembang FOSS.

Ada banyak lisensi FOSS dengan keunikan dan karakteristiknya masing-masing. Kita akan diskusikan tiga besar jenis lisensi utama: GPL, LGPL, dan lisensi bergaya BSD (Berkeley System Distribution). Ketiganya mewakili model-model yang berbeda pada lisensi FOSS dan paling banyak diadopsi. Tapi sebelumnya, untuk pandangan umum perhatikan tabel di bawah ini dari *Open Source Software Foundery* (OSSF).

12.3. Perbandingan Lisensi FOSS v2.1

Catatan: N = No (Tidak), Y = Yes (Ya).

Program Asli				
	Apakah diharuskan kode sumber di berikan ketika perangkat lunak disebarkan 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

Program Asli				
				lain, apakah anda mengizinkan bayaran yang ditagih lebih tinggi dari biaya kirim kode sumber?
MIT	N	Y	N	Y
BSD	N	N	N	Y
zlib/libpng	N	N	N	Y
Apache 1.1	N	N	N	Y
Apache 2.0	N	Y	Y	Y
Artistic	Y	N	N	Y
CPL 1.0	Y	Y	Y	Y
QPL 1.0	Y	N	N	N
MPL 1.1	Y	Y	Y	Y
LGPL 2.1	Y	N	N	N
GPL 2.0	Y	N	N	N

Program Turunan

	<p>Apakah anda mengizinkan program turunan anda memakai lisenensi lain dalam kondisi yang berbeda?</p>	<p>Lanjutan pertanyaan sebelumnya dalam kondisi seperti apakah anda mengizinkan program turunan memakai lisensinya sendiri?</p> <ol style="list-style-type: none"> 0. Lisensi turunannya harus sama 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. 	<p>Apakah diizinkan orang lain untuk memakai paten dalam perangkat lunak program anda?</p>	<p>Jika program yang disebarluaskan tidak termasuk kode sumber dan keterseediaan kode sumber dapat dilakukan lewat cara lain, apakah anda mengizinkan bayaran yang ditagih lebih tinggi dari biaya kirim kode sumber</p>
--	--	--	--	--

Program Turunan				
				?
MIT	Y	1	N	N
BSD	Y	1	N	N
zlib/lib png	Y	1	N	Y
Apache 1.1	Y	1	N	Y
Apache 2.0	Y	3	N	Y
Artistic	Y	1	Y	Y
CPL 1.0	Y	3	Y	N
QPL 1.0	Y	3	Y	N
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.

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, tapi apakah program turunan menyediakan kode sumber berbeda-beda satu lisensi dengan lainnya. Misalnya pada saat mendistribusikan program modifikasi dari sebuah program dengan lisensi model BSD kita dapat tidak diwajibkan menyertakan kode sumber.

Dan 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 didalamnya sudah mengandung kode sumber. Tetapi bila mana anda tidak membundel kode sumber dengan distribusi maka bila ada yang meminta kode sumber harganya tidak boleh lebih mahal dari ongkos kirim.

Tentang produk turunan malah perbedaan antar lisensi lebih besar lagi. Meskipun kode sumber merupakan prasyarat per definisi, produk turunan tidak demikian. 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. Misalnya meskipun GPL mensyaratkan produk turunannya harus memakai lisensi GPL juga tapi lisensi BSD tidak begitu produk turunannya bisa memakai lisensi lain atau bahkan tidak menyertakan kode sumber.

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 sumbernya begitu di gabung 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 turunnnya 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 model BSD akan dibahas secara detail berikut ini:

12.4. 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 memanisfetasikan ide dasar dari perangkat lunak bebas.

12.4.1. Copyleft

Cara yang dipakai GPL untuk menjamin kebebasan ini biasa dikenal sebagai "copyleft" ketika perusahaan 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.

Tidak seperti *public domain*, setiap orang tetap dapat menggunakan karya GPL atau copyleft dengan tetap menjaga karya itu sebagai karya yang di-copyrightkan. Pada dasarnya pemegang copyleft tetap mengikat secara hukum. Bilamana karya GPL dipakai dan dilanggar maka yang melanggar melakukan sebuah tindakan ilegal.

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 diprivatisasi 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.

12.4.2. Syarat dan Ketentuan Utama GPL

Kebebasan Pengguna

Ketika program memakai GPL, selain akses ke kode sumber, pengguna bebas untuk:

- 1) Menggunakan atau menjalankan program (klausul 0)
- 2) Memperbanyak atau menggandakan program (klausul 1)
- 3) Menyebarkan program, bahkan untuk tujuan komersial sekalipun, tapi pemberian hak cipta dan pernyataan jaminan harus disertakan (klausul 1). Penyebaran dalam bentuk kode objek diperbolehkan selama kode sumber tersedia untuk semua penerima atau pengguna (klausul 3).
- 4) Program turunan harus sama berlisensi GPL demikian juga bila memakai sumber pihak ketiga maka keseluruhan pada produk turunannya juga harus di bawah lisensi GPL (klausul 2).

Tanpa Jaminan

Meskipun produk turunan bisa dikomersialisasikan lisensinya sendiri tidak boleh dihargai/dijual. Karenanya tidak ada jaminan pada perangkat lunak GPL (klausul

11, 12). Penyebar boleh menjual layanan untuk menjamin atau mensupport (klausul 1).

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 (klausul 6).

Penerimaan dan pembatalan

Dengan memodifikasi dan menyebarkan program GPL, seseorang dianggap menerima lisensi GPL sebagai lisensi produknya (klausul 5). 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 (klausul 4).

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 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>

12.5. GNU Lesser General Public License (GNU LGPL atau LGPL)

Selain GPL proyek GNU menawarkan jenis lain dari copyleft untuk pustaka program (*libraries*). LGPL mengizinkan pustaka yang berlisensi LGPL berhubungan dengan program proprietary.

Pengecualian seperti ini bisa ditemukan dalam situasi yang berbeda. Ada kemungkinan merupakan sebuah keputusan strategis untuk memperluas

pemakaian perangkat lunak bebas karena banyak kejadian produsen perangkat keras tidak bersedia memberikan desain dari perangkat kerasnya untuk dibuat pustaka yang sesuai. Biasanya produsen memuat pustaka perangkat lunaknya secara proprietary. Untuk bisa memanfaatkan perangkat keras tersebut, LGPL adalah solusinya.

Meski ada LGPL para pendukung perangkat lunak bebas tetap menganjurkan para produsen perangkat keras menggunakan GPL pada produk mereka. Terutama beberapa pustaka yang punya kemampuan khusus dianjurkan agar memakai GPL supaya lebih banyak orang yang dapat memanfaatkan dalam lingkungan yang bebas.

12.5.1. Prasyarat dan ketentuan utama dari LGPL

LGPL identik dengan GPL pada hampir semua aspek dimana harus disebutkan tidak ada jaminan dan pernyataan bahwa lisensi diperoleh langsung dari penulis, kapan diterapkan dan kapan dibatalkan, dan hubungannya dengan ketentuan hukum lain yang berlaku bagi pengguna.

Perbedaannya ada pada hak pemakai LGPL membedakan dua buah situasi. Di satu sisi disebut “produk berdasar dari pustaka” ini berarti pustaka itu sendiri maupun pustaka turunannya. Di sisi lain penyebutan “produk yang memakai pustaka” ini artinya bukan pustaka atau turunannya melainkan program yang didesain untuk memanfaatkan pustaka.

Karya berdasarkan dari pustaka

Pada kasus ini karya atau produk merupakan pustaka maupun turunan dari pustaka yang bersangkutan. Pada intinya semua yang disebutkan di GPL berlaku semua di sini.

Kebebasan Pengguna

- 1) Menggunakan atau menjalankan program (klausul 0)
- 2) Memperbanyak atau menggandakan program (klausul 1)
- 3) Menyebarkan program, bahkan untuk tujuan komersial sekalipun, tapi pemberian hak cipta dan pernyataan jaminan harus disertakan (klausul 1). Penyebaran dalam bentuk kode objek diperbolehkan selama kode sumber tersedia untuk semua penerima atau pengguna (klausul 4).

- 4) Program turunan harus sama berlisensi LGPL demikian juga bila memakai sumber pihak ketiga maka keseluruhan pada produk turunannya juga harus di bawah lisensi LGPL (klausul 2c).

Anda juga dapat menerapkan lisensi GPL terhadap hasil penggandaan karya LGPL, khususnya ketika menggabungkan kode ke dalam program yang bukan pustaka.

Karya yang memakai pustaka

Pada kasus ini ketika sebuah produk yang berhubungan dengan pustaka GPL sedangkan produk ini memiliki lisensi lain selain GPL atau LGPL produk ini diminta untuk memaketkan kode sumber pustaka bila diminta oleh pemakainya. Apabila tidak, produk ini bisa memakai mekanisme pustaka-berbagi agar bisa diakses oleh pustaka yang LGPL.

Dengan membuat katagori demikian, LGPL memungkinkan program proprietary memakai pustaka LGPL.

12.6. 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.

Kebebasan Pemakai

- 1) Memperbanyak program dan menyebarluaskan program baik berbentuk kode sumber atau kode biner. Penyebarluasan tidak dituntut menyertakan kode sumber.
- 2) Membuat karya turunan dan mendistribusikannya dalam bentuk kode sumber atau binari. Pembuat perubahan bebas memilih lisensi, FOSS atau proprietary.
- 3) Dapat dimasukkan dalam program proprietary.

Lisensi aslinya (BSD dengan 4 klausul) memiliki satu klausul iklan sehingga

tidak dipakai lagi. Yang digunakan sekarang BSD dengan tiga klausul, mirip dengan lisensi MIT, bedanya lisensi MIT tidak punya klausa “tidak ada paksaan pada produk turunan”. Juga ada lisensi BSD 2 klausul yang menghilangkan klausa di atas sehingga menjadi sangat mirip dengan lisensi MIT.

12.7. Lisensi Gabungan (*Multiple Licensing*)

Hal penting untuk disadari bahwa program dapat memiliki lebih dari satu lisensi. Lisensi adalah pilihan anda pemilik hak cipta, tergantung bagaimana anda memandang hubungan antara karya anda dengan para penggunanya. Bisa jadi ada lebih dari satu jenis pemakai dengan lebih dari satu hubungan. Untuk situasi seperti ini anda dapat memilih lisensi yang berbeda-beda sesuai dengan situasinya.

Contohnya OpenOffice.org yang memiliki lisensi GPL dan *Sun Industrial Standart Source License* (SISSL). Meskipun OpenOffice.org menganjurkan para pemakai menggunakan GPL dan berpartisipasi penuh ke komunitasnya di openoffice.org, SISSL disediakan sebagai alternatif untuk perusahaan yang tidak bisa memakai GPL.

12.8. Kode sumber itu Terbuka; Bagaimana dengan Dokumentasi?

12.8.1. 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. FSF juga menyediakan lisensi khusus untuk dokumentasi. GNU Free Document License (GNU FDL atau FDL) adalah bentuk lain copyleft untuk manual, book teks dan dokumen lainnya yang menjamin pemakai mendapat kebebasan memperbanyak, menyebarkan 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 menyebarkan. Produk

turunannya pun harus berlisensi FDL, bedanya pada segi jumlah penggandaan ada aturan di FDL.

12.8.2. 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.

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:

- 1) *Attribution (BY)*: Untuk semua penggandaan dan penyebaran, penyebutan pengarang asli diperlukan.
- 2) *Non Commercial (NC)*: Karya ini tidak dapat dikomersialisasikan.
- 3) *Non Derivative Works (ND)*: Karya tidak boleh diubah.
- 4) *Share Alike (SA)*: Diizinkan mengubah, memodifikasi dan membuat produk turunan dari ini dengan syarat lisensi yang dipakai pada karya turunannya 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 dari sebelas, tapi kesebelas lisensi pada versi satu masih tetap berlaku.

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 konteks 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.

	Nama penulis asli disebutkan	Boleh untuk kepentingan komersial	Boleh dibuat turunan (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
CC BY-SA	Yes	Yes	Yes	Yes
CC	No	No	Yes	No

	Nama penulis asli disebutkan	Boleh untuk kepentingan komersial	Boleh dibuat turunan (diubah)	Karya turunan harus mamakai lisensi yang sama seperti karya aslinya
NC				
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
GNU FDL	Yes	Yes	Yes	Yes

XIII. BEBERAPA SKENARIO

Untuk setiap pihak yang berkepentingan, pemakaian FOSS akan dapat berbeda-beda satu sama lain. Para pengembang menggunakan sebuah program mungkin lebih intensif dan mendalam daripada pengguna biasa. Aktivitas pengembang perangkat lunak mungkin lebih merasa terbatas dengan pembatasan pemakaian dibandingkan pengguna biasa untuk sebuah program. Bagian ini mencoba untuk mengembangkan beberapa skenario sebagai contoh untuk menjelaskan perbedaan isu legalitas yang mungkin akan terjadi dalam perbedaan penggunaan FOSS.

13.1. Pengguna Akhir (Individu / Bisnis / Pemerintahan)

Budi adalah seorang guru sekolah menengah atas. Sekolah tempat ia mengajar tidak mampu membeli aplikasi perkantoran (office) yang proprietary. Meskipun perusahaan pembuat aplikasi perkantoran tersebut menawarkan paket kerjasama khusus untuk sekolah tempat ia mengajar, Budi tetap mencari solusi alternatif untuk mendidik siswanya agar tidak menjadi ketergantungan dengan perangkat lunak proprietary. Ahmad, rekan Budi adalah seorang programmer yang memiliki minat pada pengembangan FOSS, dan ia mengenalkan Budi pada aplikasi office FOSS. Ia lalu mengimplematasikan solusi aplikasi perkantoran FOSS dan mengajarkan baik aplikasi FOSS maupun yang proprietary kepada siswanya. Budi merasa sangat puas dengan program tersebut dan mengenalkannya pada rekan-rekan yang lain. Akhirnya lambat laun untuk urusan administrasi sekolah serta berbagai tugas administratif lainnya pun sudah mulai mencoba menggunakan solusi FOSS.

Dalam kasus ini sebagai contoh sederhana pengguna biasa/ pengguna akhir, baik Budi (sebagai individu) maupun sekolahnya (sebagai institusi) membuat berbagai modifikasi dari aplikasi yang mereka download dari web.

Situasi untuk pengguna biasa/ pengguna akhir relatif sederhana. Pengguna akhir dari sebuah aplikasi perangkat lunak dapat berupa individu, institusi pemerintahan, atau institusi bisnis. Pengguna individual atau institusi bisnis resmi mungkin memiliki perbedaan alasan dalam menggunakan FOSS. Alasannya mungkin memilih solusi yang lebih murah atau solusi alternatif untuk

memenuhi kebutuhannya, dan alasan yang lain mungkin menggunakan FOSS untuk dapat mengkostumisasi sesuai dengan kebutuhan. Serta yang lainnya mungkin beralasan untuk menghindari ketergantungan pada perusahaan perangkat lunak tertentu.

13.1.1. Isu-isu Hukum yang Terkait

Cara menggunakan solusi FOSS tidak jauh berbeda dengan cara menggunakan solusi proprietary. pengguna mengunduh solusi FOSS atau mendapatkan salinannya, menginstallnya pada komputer (dengan mengkopinya ke dalam hard disk), menjalankan program, dan menggunakannya sesuai kebutuhan pengguna.

Hak yang diperhatikan disini adalah hak untuk menggandakan program tersebut dan hak untuk menjalankannya (menjalankan program mungkin pula dapat disamakan dengan menggandakan program, namun dibedakan dalam lisensi FOSS. Contohnya lisensi GPL, tidak memiliki batasan dalam menjalankan program, namun memiliki aturan dalam menggandakannya). Hak-hak tersebut dijamin oleh semua lisensi FOSS, dan tidak terlalu ditekankan pada legalitas bagi pemakai biasa / pemakai akhir. Anda mesti menemukan isu-isu lain yang dapat dijadikan sebagai bahan pertimbangan, antara lain sebagai berikut.

Dukungan Teknis

Meskipun untuk pemakai akhir terdapat solusi FOSS, pengguna juga harus memperhatikan tentang dukungan teknis untuk solusi FOSS yang dibutuhkan oleh pengguna. Secara sederhana pemakai akhir dapat menghemat biaya dengan mengunduh sebuah solusi FOSS dan bisa juga dengan cara membeli paket solusi FOSS di toko yang juga menyediakan solusi proprietary. Terkadang harganya juga hampir sama dengan solusi proprietary. Namun terdapat perbedaan diantara keduanya. Misalkan pengguna membeli Linux Red Hat di sebuah toko, pengguna tidak membayar lisensinya, tetapi untuk layanan dan dukungan teknisnya. Ketika batas waktu perjanjian layanannya berakhir, pengguna dapat membayar untuk layanan yang lain atau memilih penyedia jasa layanan lainnya.

Kostumisasi

Jika solusi FOSS tidak sesuai dengan kebutuhan pengguna, maka ia dapat

meminta pengembang individual maupun perusahaan pengembang perangkat lunak untuk menjadikannya sesuai dengan permintaan dan kebutuhan pengguna. Dalam beberapa kasus, pengguna personal, institusi swasta atau pemerintah menginginkan untuk memasukan pasal pada kontrak perjanjian yang dibuat dengan perusahaan vendor atau pengembang untuk pertanggungjawaban dan kompensasi yang mungkin timbul jika terdapat pelanggaran perjanjian. Sebagai pembeli, pengguna bebas untuk menambahkan pasal dalam kontrak.

Kebutuhan Pemerintah

Karena konsep lisensi FOSS memiliki model yang berbeda dengan hak cipta, maka dalam proses pengadaan software, pemerintah harus hati-hati dalam membuka tender perangkat lunak atau menandatangani kontrak dengan vendor. Rancangan penawaran dan kontrak pemerintah harus di rancang sesuai ketentuan hukum hak cipta tradisional, dan harus diuji serta revisi, jika gagal dalam menyeimbangkan perlakuan terhadap FOSS dengan perlakuan terhadap perangkat lunak proprietary.

13.2. Pengembang (Individu, Bisnis)

Sebagai pengembang perangkat lunak (baik yang sifatnya individual maupun institusi bisnis), pengembang harus lebih berhati-hati dengan syarat dan kondisi dari berbagai perbedaan lisensi karena pengembang menggunakan FOSS secara intensif. Pengembang tidak hanya menjalankan dan menggandakan perangkat lunak, namun juga membuat turunannya lagi bahkan mendistribusikan perangkat lunak hasil turunan tersebut bersamaan dengan program aslinya.

Dalam rangka berkontribusi untuk pengembangan FOSS, pengembang harus memiliki kebebasan untuk menjalankan, menggandakan/memperbanyak, mendistribusikan, dan menyiapkan program turunannya.

Hak-hak kebebasan di atas telah dijamin oleh lisensi FOSS. Hal ini menjadi hak yang esensial dan penting baik di Free Software Definition dan juga di dalam Open Source Definition. Perbedaan diantara lisensi FOSS hanya dalam hal-hal tertentu saja, khususnya dalam hal membuat dan mengembangkan perangkat lunak turunannya. Pengembang harus memperhatikan hal ini dan mengkonsultasikannya pada pengacara berdasarkan situasi yang dihadapi dan waktu yang dibutuhkannya. Perbedaan pilihan menjadi perhatian ketika kita

berpartisipasi dalam setiap tahap yang berbeda dalam pengembangan perangkat lunak.

13.2.1. Ketika Memulai Proyek Baru

Teman Budi, Hamid adalah seorang petugas perpustakaan sekolah. Perpustakaan sekolah tidak begitu besar dan perpustakaan ini juga terbuka untuk penduduk sekitar. Untuk dapat mengetahui dengan akurat catatan buku yang keluar atau sedang dipinjam, Hamid meminta bantuan rekannya untuk membuat program perpustakaan untuk membantu pekerjaannya.

Keterkaitan Hukum Legal, Pilih Sendiri Salah Satu Lisensinya

Pengembang: Apakah artinya proyek ini untuk saya dan orang lain? Bagaimana jika saya menginginkan orang lain terlibat? Apa yang dapat dikatakan oleh lisensi FOSS? Apakah perbedaan antar lisensi FOSS?

Situasi ini akan sederhana jika pengembang memulai proyek barunya tanpa menggunakan modul-modul yang sudah ada, jika pengembang tidak terlalu memperhatikan lisensi dari semua modul yang menarik untuk digunakan.

Perbedaan karakteristik dari lisensi FOSS akan secara signifikan mempengaruhi kemungkinan jalannya perkembangan proyek. Tentukan kebutuhan sebelum memilih lisensinya.

Misalkan pengembang ini adalah seorang yang memiliki idealisme perangkat lunak bebas, maka kemungkinan besar ia akan memilih lisensi GPL atau LGPL. Jika pengembang tidak berfikir untuk membatasi orang lain untuk menggunakan perangkat lunaknya, maka pengembang mungkin akan menggunakan lisensi BSD. Lalu jika pengembang berfikir bahwa lebih baik jika pengembangan dapat dikontrol dalam sebuah perusahaan, mungkin pengembang tidak akan berfikir menggunakan lisensi BSD.

Sebaliknya apabila suatu saat akan dibuat turunan model lisensi BSD mungkin lebih cocok meskipun nanti akan menjadi kesulitan jika versi turunan ini digabungkan kembali ke proyek utama yang lisensinya nanti sudah bukan FOSS.

Pengembang: Dapatkah saya merubah lisensi proyek saya?

Meskipun pengembang telah membuat lisensi copyright untuk sebuah proyek, ia masih dapat memilih lisensi lain untuk programnya selama versi sebelumnya sudah dilisensikan di bawah lisensi FOSS. Hak-hak penerima versi sebelumnya tidak akan terpengaruh semenjak lisensinya tidak dapat dirubah. Situasi ini akan lebih kompleks lagi jika kontribusi dari komunitas tidak dimasukkan ke dalam versi yang baru, karena dalam kasus ini pemilik copyright mencakup pula kontributornya.

Pengembang: Saya tidak menyukai lisensi FOSS yang ada, dapatkah saya membuat yang baru?

Meskipun banyak lisensi FOSS yang sudah ada, masih dimungkinkan bagi pengembang menentukan lisensi FOSS yang tidak lebih disukainya. Pengembang berhak untuk memilih lisence yang manapun untuk proyeknya, termasuk lisensi baru yang sudah dirancangnya. Namun menciptakan lisensi FOSS baru memerlukan pengetahuan hukum khusus dan ketrampilan untuk menghindari kesamaran dan celah hukum. Sudah terlalu banyak lisensi FOSS dan juga membutuhkan biaya yang besar untuk mengerti dan memahaminya. Membuat satu lisensi baru tidak direkomendasikan jika pengembang memang tidak benar-benar menemukan salah satu lisensi yang cocok dan alasan yang kuat untuk mengerjakannya.

13.2.2. Ketika Memodifikasi Modul yang Sudah Ada

Aplikasi perkantoran yang ada di sekolah tempat Budi bekerja menggunakan Bahasa Inggris dan belum mendukung Bahasa Lokal. Meskipun siswa disana tidak mengalami kesulitan menggunakan antar muka berbahasa Inggris. Namun menjadi masalah ketika Budi mencoba untuk mengajarkan orang desa sekitar. Konsultan Budi, namanya Ahmad telah banyak berkontribusi dalam program FOSS dan juga sangat familiar dengan kode sumber dari perangkat lunak tersebut. Dia lalu mendiskusikannya pada beberapa rekannya untuk membentuk tim dalam rangka melokalisasi aplikasi tersebut.

Ketika kita mencoba untuk merubah modul yang sudah eksis dan tujuannya tidak hanya untuk penggunaan pribadi saja namun juga untuk didistribusikan kembali,

misalkan melokalisasi program sesuai dengan daerah tertentu, terlebih dahulu kita harus mengetahui lisensi dari modul tersebut. Sehingga kita akan mengetahui hak yang dijamin oleh lisensi tersebut.

Pengembang: di bawah lisensi ini apa hak saya yang dijamin, dan apa saja yang dibatasi dalam menggunakan hak tersebut?

Dalam mendistribusikan proyek FOSS, beberapa lisensi FOSS (GPL dan LGPL) mengharuskan distributor menyediakan baik kode objek maupun kode sumbernya, atau sedikitnya menyediakan informasi bagaimana cara mengakses kode sumber tersebut.

Dalam memodifikasi proyek FOSS, beberapa lisensi FOSS (GPL, LGPL dan BSD) mengharuskan pihak yang memodifikasi untuk menyediakan dokumentasi tentang perubahan yang telah dibuat.

Dalam mendistribusikan pekerjaan turunannya, lisensi copyleft mengharuskan turunannya untuk dilisensikan di bawah lisensi yang sama dengan asalnya, meskipun lisensi FOSS lain membolehkan kita untuk memilih lisensi yang berbeda (BSD dan MIT)

Dalam kasus Ahmad ini, mereka mencoba untuk melokalisasi open office yang memiliki dua lisensi. Dalam hal ini sesuai saran openoffice.org yang mengharuskan mereka menggunakan GPL, maka hasil akhir proyek mereka yakni open office yang sudah dimodifikasi juga harus memiliki lisensi GPL.

Beberapa lisensi FOSS seperti lisensi MIT membolehkan pengguna untuk membuat sub lisensi dari karya aslinya. Ini berarti dalam pendistribusiannya baik hasil kopi maupun program aslinya, dengan diberi ruang oleh pemilik copyrightnya, distributor dapat memilih lisensi yang berbeda dan menjadi orang yang memberikan lisensinya sendiri. Dalam beberapa contoh kasus, ketika kita menurunkan dan mendistribusikannya bersama dengan program aslinya, kita dapat memilih baik sebagai pemberi lisensi dari kedua program tadi. Dan membuat menjelaskan hubungan antara kedua bagian tersebut. Jika sub lisensi tidak diperbolehkan, pemberi lisensi dari program aslinya akan menjadi pemberi lisensi, meskipun pemberi lisensi dari program turunannya adalah kita yang menyiapkan program turunannya.

13.2.3. Ketika Mengintegrasikan Modul-Modul FOSS yang Berbeda Menjadi Satu

Ahmad bekerja di AA Software Inc. (AA). Untuk mengembangkan perangkat lunak dari berbagai proyek berbeda yang mereka kembangkan, dikembangkan sebuah sistem manajemen proyek dengan mengintegrasikan berbagai modul FOSS yang berbeda. Sistem yang dibuat ini digunakan untuk internal, namun karena sistem ini memang bagus, mereka berencana untuk mendistribusikannya secara komersial.

Ini akan menjadi kompleks situasinya, mengintegrasikan modul-modul yang diambil sekaligus juga membuat turunan dari modul-modul tersebut.

Keterkaitan Hukum Legal-- Memberikan lisensi dari program yang telah diintegrasikan dan membandingkan kompatibilitas diantara lisensi-lisensi tersebut.

Dalam kasus ini, yang paling penting bagi AA adalah memperhatikan lisensi dari tiap modul. Jika dilisensikan dengan lisensi yang sama, misalnya GPL, maka akan menjadi lebih sederhana. Karena tidak peduli modul-modul tersebut diintegrasikan ataupun dimodifikasi, maka hasil akhir sistemnya harus berlisensi GPL pula.

Sama halnya jika semua modul berlisensi BSD, namun dalam kasus ini AA dapat memilih lisensi proprietary untuk modul sistem yang telah diintegrasikan.

Namun jika beberapa modul berlisensi GPL sementara yang lainnya bukan, maka AA harus melihat kecocokan diantara lisensi-lisensi yang berbeda tersebut. Ketika dua lisensi cocok, maka dua modul yang berlisensi tersebut dapat dikombinasikan menggunakan kedua lisensi tersebut. FSF menyediakan daftar kecocokan dan ketidakcocokan lisensi FOSS GPL.

Ketika mengkombinasikan program berlisensi GPL dan BSD (GPL Compatible) menjadi program yang lebih besar, maka program tersebut akan berlisensi GPL yang menemukan syarat dari kedua lisensi tersebut (GPL dan BSD). Jika beberapa modul berlisensi GPL, namun modul yang lain GPL-incompatible, mengintegrasikannya dalam program yang lebih besar berarti lebih dari sekedar hanya menggabungkan modul. AA harus memilih modul mana yang lebih

penting dan menggantikannya dengan modul lain yang menggunakan lisensi yang sesuai (*compatible*).

Lisensi yang digunakan dalam modul-modul yang berbeda dan cara mengkombinasikannya akan menentukan lisensi sistem akhirnya dan cara pendistribusiannya

Penekanan lainnya – Pemilihan Klausula Hukum

Terakhir, untuk pengembang yang memiliki kemampuan untuk memilih lisensi untuk program yang dikembangkannya, baik memulai program sendiri maupun menurunkan kembali dari yang sudah ada, harus tetap memperhatikan bahwa banyak lisensi yang diterima OSI adalah turunan dari perusahaan perangkat lunak proprietary. Beberapa lisensi telah dirancang untuk mempertemukan kebijakan dan strategi perusahaan yang mungkin tidak cocok untuk dipilih oleh kita sebagai pengembang umum. Beberapa isu teknis seperti pemilihan klausula hukum (Dapat ditemukan di QT Public License, Mozilla Public License, dan Common Public License, dll) mungkin menjadi signifikan dalam perkara hukum dan sangat perlu untuk menjadi perhatian.

13.3. Pengusaha/Produsen (Bisnis)

Ana dan temannya telah selesai membuat versi lokal dari sebuah aplikasi office FOSS yang sudah tersedia. Perusahaan perangkat lunak AA sangat tertarik dengan aplikasi ini dan perusahaan juga telah membuat sebuah aplikasi untuk administrasi kecil. Mereka lalu memaket jadi satu menggunakan lisensi proprietary mereka. Paket ini laku terjual di pasar. Beberapa bulan kemudian AA ingin membuat distribusi dari system manajemen proyek dengan lisensi proprietary yang di dalamnya juga menggunakan komponen FOSS.

13.3.1. Distribusi Sederhana

Dalam situasi di atas, program FOSS dan proprietary didistribusikan dalam satu paket. Untuk aplikasi FOSS ini hanya media distribusi, dan lisensi FOSS menjamin bahwa aplikasi itu boleh didistribusikan. Sementara bagi program proprietary AA memegang hak ciptanya dan berhak memilih jalur distribusinya.

Adalah diperbolehkan memaketkan program proprietary dengan program FOSS dalam satu paket di distribusi sepanjang tidak terkait langsung atau berdiri sendiri.

13.3.2. Distribusi dari Sistem Terintegrasi

Dalam kasus distribusi sistem yang terintegrasi, ini sangat bergantung pada lisensi dari masing-masing module dan bagaimana cara mereka di kombinasi. Seperti diterangkan di atas AA harus memastikan bahwa lisensi masing-masing module mendukung untuk di kombinasi. Lisensi ini juga akan mengikat AA bagaimana cara distribusi sistem yang terintegrasi.

FOSS juga dipakai dalam sistem embedded. Banyak alat seperti handphone, handhelds, kamera digital, dan DVD player sekarang menggunakan FOSS. Pabrik alat-alat ini memakai FOSS untuk memotong biaya pada saat membuat produk baru, tapi tidak untuk menyebarluaskan FOSS sendiri.

13.3.3. Proyek yang di Sponsor Pemerintah

Gerakan FOSS dan pengembangan FOSS yang sangat cepat mendapat perhatian dari komunitas FOSS sendiri, akademisi, pembuat kebijakan dan sebagainya. Di beberapa negara Asia Pemerintah bekerjasama dengan produsen PC dan vendor untuk menciptakan PC yang terjangkau oleh masyarakat, biasanya di bundle dengan sistem operasi dan aplikasi perkantoran FOSS. Usaha ini menyebabkan harga kepemilikan PC menjadi rendah dan memaksa raksasa proprietary Microsoft menurunkan harganya. Pemerintah beserta dukungan komunitas FOSS membuat proyek FOSS dan memakai FOSS untuk kebijakan teknologi nasional. Beberapa institusi akademik bahkan sudah memiliki proyek FOSS jauh sebelum pemerintah menyadari potensi FOSS.

Dalam daftar FAQ (*Frequently Ask Questions* atau pertanyaan yang sering ditanyakan) yang dipelihara oleh FSF tentang GPL, terdapat pertanyaan apakah pemerintah Amerika Serikat dapat meluncurkan program dengan lisensi GPL, atau meluncurkan program turunan hasil pengembangan program GPL. Situasinya sangat berbeda dari satu negara-kenegara lain, dan dari kasus per kasus tergantung dari regulasi masing-masing pemerintah. Kebanyakan peraturan pemerintah yang menjadi dasar pada proyek-proyek pemerintah tergantung pada hukum hak cipta nasional dan internasional, dan mungkin lebih bersifat nasional,

protektif secara ekonomi, dan kadang-kadang tidak bersahabat dengan model lisensi FOSS.

Berikut adalah dua kasus studi pengembangan FOSS yang di dukung pemerintah. Yang pertama tentang studi yang diadakan pada institusi riset pemerintah tanpa melibatkan peraturan pemerintah. Yang kedua tentang sebuah proyek FOSS nasional.

Proyek FOSS yang didanai oleh pemerintah – Studi kasus dari kawasan Asia-

Pasifik

Proyek FOSS di bawah institusi riset pemerintah “Multi-lingual Editor”, Jepang.

Emacs adalah editor teks banyak bahasa (*multi-lingual*) yang pertama kali dikembangkan oleh Richard Stallman di MIT. Tidak lama setelah proyek GNU mulai berjalan pada 1984 dan pertama kali diluncurkan pada 1985. GNU Emacs diluncurkan dengan lisensi GPL.

Institusi riset Jepang, *Electrotechnical Laboratory* (ETL) memulai riset pada pemrosesan banyak bahasa dengan menyatukan GNU Emacs dan Mule (editor banyak bahasa yang berdasarkan Emacs dan pada akhirnya bergabung dengan Emacs dengan nama MULE) pada pertengahan 1990-an tapi terganjal beberapa masalah yang berhubungan dengan hak cipta. Para pengembang di ETL adalah peneliti pemerintah sedangkan model lisensi FOSS sangat berbeda dengan hukum hak cipta, jadi tidak ada satu pun yang bisa menentukan apakah ETL dapat mengembalikan kodenya ke FSF untuk dilepas dengan lisensi GPL. Hasilnya ETL tidak pernah secara resmi meluncurkan produknya melainkan hanya versi percobaan (trial).

Negosiasi antara ETL dan FSF menghasilkan perjanjian khusus. FSF setuju untuk tidak disebutkan dalam hak cipta dari kode yang dimodifikasi, sementara ETL mengizinkan FSF untuk mendapat kode sumbernya. Ini pertamakalinya ada bagian kode dari Emacs yang bukan milik FSF.

Pada tahun 2001 ETL dirombak organisasinya dan dilebur ke dalam institusi nasional Advance Industrial Science and Technology (AIST). Meskipun AIST masih institusi yang dibiayai pemerintah, AIST adalah badan independen dan aset AIST bukan milik negara. Sepertinya AIST akan bisa meluncurkan kodenya dengan lisensi GPL. Perlu setahun lagi kira-kira dilakukan negosiasi internal

untuk meluncurkan kode dalam GPL. Sulit untuk meyakinkan AIST, kelebihan yang didapat jika meluncurkan kode dengan GPL. Menurut Dr. Kenichi Handa alasan utama di belakang itu tetap misteri.

Kejadian ini terjadi sebelum pemerintah Jepang mengambil posisi tentang pengembangan FOSS. Dalam sebuah konferensi *open source* antara negara-negara Asia dimana Dr Handa diundang untuk berdiskusi tentang pengembangan Emacs, Shuichi Tashiro, kepala pengembangan FOSS di bawah Kementerian Ekonomi, Perdagangan dan Industri menambahkan bahwa pemerintah Jepang perlu mengubah regulasi untuk memberikan hak cipta kepada pengembang di institusi riset pemerintah yang oleh karenanya dapat memilih lisensi sesuai kebutuhan selama didukung oleh hukum sejak awal pengembangannya.

Proyek FOSS Nasional – Taiwan

Di bawah tekanan kongres, Pemerintah Taiwan memulai merencanakan proyek nasional FOSS pada tahun 2002, dan mengalokasikan anggaran untuk proyek FOSS selama 5 tahun. Kementerian Ekonomi (The Ministry of Economics/ MoE) telah ditugaskan untuk menstruktur, mensponsori, dan mengontrol sub proyeknya.

Di bawah regulasi pemerintahan, meskipun hasil yang menggunakan lembaga pemerintahan akan menjadi copy right dan aplikasi proyek tersebut memiliki tujuan berdasarkan prinsip yang diyakini. Sehingga setidaknya hal tersebut akan lebih menguntungkan bagi pengembangan sains dan teknologi nasional, hasilnya antara lain:

- Dilisensikan bebas
- Dilisensikan untuk lembaga atau perusahaan Taiwan
- Digunakan atau dibuat dengan otorisasi Taiwan

Meskipun pengecualian dibuat untuk proyek FOSS, dimana pengembang FOSS telah menandatangani perjanjian dengan MoE yang memiliki tugas utama dan penting dalam melindungi kompetensi ekonomi nasional. Sehingga aturan yang sangat ketat akan diterapkan dalam proyek FOSS nasional. di bawah aturan MoE, hanya pihak ketiga (Digunakan atau dirancang dengan Hukum di Taiwan) yang mendapat pengecualian. Beberapa prinsip tidak sejalan dengan model

pengembangan FOSS dan agak sulit untuk sub proyek di bawah proyek FOSS nasional untuk merilis kodenya.

Pertanyaan mengenai hal akan terhalanginya proyek FOSS oleh peraturan terus diajukan pada tahun pertama dari lima tahun proyek FOSS. Semua lembaga pemerintahan yang berhubungan dengan proyek FOSS melakukan pertemuan untuk menemukan solusinya. Namun karena model lisensi FOSS berbeda dengan model yang digunakan, maka masalahnya tidak terpecahkan dan kode yang dikembangkan dalam tahun pertama tidak dapat dirilis.

Negosiasi terakhir yang berlangsung pada bulan Mei 2004, dimana lembaga pemerintahan sampai pada suatu kesepakatan. Kementerian MoE akan mengirimkan masalah ini kepada Administrative Yuan (Badan Administrasi Tertinggi) tentang pengimplementasian aturan-aturan umum pada proyek FOSS untuk mendapatkan rekomendasi formal dari pemerintah. Bahwa FOSS memenuhi klausul pengecualian dan dapat dikecualikan dari prinsip. Sementara itu kementerian MoE sendiri mencari celah yang memungkinkan untuk merevisi peraturan dalam kementerian. Pada saat yang sama beberapa kode yang dibuat dari proyek FOSS diserahkan ke *National Science Council* mengikuti aturan umum yang ada. Diharapkan proyek FOSS ini akan dapat melepas kodenya di bawah FOSS pada waktu dekat.

XIV. DAFTAR ISTILAH

14.1. *Copyleft*

Istilah *copyleft* diajukan oleh pengacara *Free Software*. *Copyleft* adalah sebuah ide alternatif untuk menggambarkan aktivitas kreatif manusia yang didefinisikan pada hukum hak cipta. Copyright (Hak cipta) biasanya memberikan hak eksklusif pada pemilik hak cipta dan pembatasan semua akses atas karya ciptanya. Pencipta bisa memberikan lisensi *copyleft* terhadap karya mereka untuk memberikan kepastian hak pada siapa saja yang berkepentingan pada pemakaian karya mereka, juga mengizinkan orang lain untuk memberikan lisensi *copyleft* terhadap apa saja yang dihasilkan berdasarkan karya awal. *Copyright* dan *Copyleft* mewakili dua ide yang sangat berbeda tentang hubungan antara pencipta dan ciptaannya, namun *copyleft* tidak melawan hukum *copyright*. Sebaliknya, tanpa hak yang diberikan oleh hukum hak cipta, pemegang hak cipta tidak akan memiliki kekuatan untuk meng-*copyleft*-kan karya mereka.

14.2. *Copyright* (Hak cipta)

Sebuah pemasangan hak eksklusif yang resmi diberikan oleh pemerintah untuk penghargaan seorang pengarang pada penggunaan ungkapan asli (termasuk penyair, film, musik ciptaan, pelukis, software, dll.) diberikan secara eksklusif kepada pemilik. Hak cipta dihasilkan untuk sebuah pekerjaan atas ciptaannya. Kecuali dari pemasangan pembatasan oleh Hukum hak cipta. beberapa penggunaan pada sebuah pekerjaan tanpa izin pemilik hak cipta dihargai sebagai sebuah pelanggaran tidak resmi. hukum hak cipta itu hanya melindungi fakta cara pada pengungkapan ide bukan ide mereka sendiri.

14.3. *Copyright-holder* (Pemegang Hak cipta)

Pemegang hak cipta adalah individu atau badan hukum yang telah diberi hak eksklusif di bawah hukum hak cipta. Biasanya hukum hak cipta itu untuk melindungi pengarang atas pekerjaannya yang kreatif, tetapi banyak perlindungan hak diberlakukan juga sebagai hak milik dan dapat dipindahkan kepemilikannya. Banyak pemilik hak cipta bukan pencipta atas pekerjaan mereka sendiri tapi karyawan mereka.

14.4. *Derivative works* (Karya Turunan)

Hukum hak cipta dihasilkan untuk semua karya cipta satu kali penciptaan. Ketika disetujui kepemilikan hak cipta, siapa saja dapat menciptakan karya cipta turunan berdasarkan pada karya asli. Sebuah versi yang baru pada sebuah program, misalnya apakah mengandung semuanya atau berbeda dengan kode versi sebelumnya. Versi baru ini merupakan sebuah bentuk pekerjaan pada versi sebelumnya. Terjemahan sebuah dokumen juga dianggap sebagai bentuk turunan dari suatu karya tulis.

14.5. *Distribution/Redistribution* (Penyebaran/Penyebaran ulang)

Penyaluran pada peniruan pekerjaan juga sebuah hak eksklusif yang diberikan untuk pemilik hak cipta. Pada lisensi FOSS, semua menerima peniruan pada sebuah program diperbolehkan untuk membuat penyaluran lebih lanjut. Batas waktu pembagian kembali dapat digunakan ketika penekanan penyaluran telah diterima program dari mana saja dan penyaluran lebih lanjut

14.6. *Fair Use* (Penggunaan yang Adil)

Hukum hak cipta adalah sebuah keseimbangan antara kepentingan pribadi dan kepentingan umum. “*Fair Use*” dikembangkan untuk membatasi perlindungan hak cipta dan untuk memberikan akses yang lebih luas kepada publik umum terhadap karya cipta.

Jika suatu karya cipta digunakan tanpa memperhatikan izin dari pemilik hak cipta untuk tujuan member komentar, kritik, berita, laporan, penelitian, pendidikan atau pengajaran, maka tidak dinilai sebagai sebuah pelanggaran. Faktor-faktor berikut ini dapat dijadikan pertimbangan oleh kehakiman untuk menentukan apakah sebuah perkara gugur karena penggunaan secara fair atau memang sebuah pelanggaran hak cipta:

- 1) Tujuan dan karakter penggunaan, termasuk apakah digunakan untuk komersil atau pendidikan yang tidak mengambil laba (non-profit).
- 2) Nilai hakiki dari karya cipta itu.
- 3) Jumlah dan substansi dari bagian karya yang digunakan terkait dengan karya secara keseluruhan.
- 4) Efek penggunaan terhadap potensi pasar dari nilai karya cipta.

14.7. License (Lisensi atau Surat Izin)

Pemegang hak cipta atau pemilik paten biasanya meminta pengguna karyanya untuk menerima syarat dan ketentuan dari suatu lisensi atau surat izin, sebagai syarat agar dapat izin untuk menggunakan karya cipta atau karya yang telah dicopyright-kan.

14.8. Multiple Licensing (Pemberian lisensi campuran)

Pemilik hak cipta pada sebuah karya cipta dapat memiliki bermacam-macam cara agar karyanya dapat berguna. Syarat dan ketentuan yang diharapkan diterima pengguna dapat berbeda antara kasus yang satu dengan yang lain. Misalnya, pemegang hak cipta sebuah program editor ingin melisensikan karyanya sebagai lisensi akademik, yang lebih murah dan lebih terjangkau pada murid-murid, namun lisensi komersil digunakan bila menjual program untuk perusahaan. Untuk tujuan atau target pengguna yang berbeda, pemegang hak cipta dapat memilih dua lisensi, yaitu lisensi FOSS dan lisensi proprietary.

14.9. Public Domain (Milik Umum)

Public domain adalah bidang lisensi yang tidak dilindungi oleh hukum hak cipta dan dapat digunakan secara gratis. Lisensi ini kemungkinan berlaku terhadap karya yang telah ada sebelum hukum hak cipta, atau karya yang awalnya dilindungi hukum hak cipta, tetapi syarat perlindungannya telah berakhir, atau karya yang sengaja oleh penciptanya tidak diklaim sebagai hak ciptanya. Dalam kasus terakhir itu mestinya ada pernyataan secara eksplisit.

14.10. Source Code (Kode Sumber)

Kode sumber ditulis dalam bahasa khusus yang didesain untuk pemrograman. Sebuah program dalam bentuk kode sumber tidak mudah dimengerti orang awam, tetapi dapat dimengerti programmer terlatih. Jika kode sumber telah diubah menjadi kode yang dapat dibaca mesin, programmer sulit untuk mengerti dan memodifikasi program. Akses terhadap kode sumber menjadi syarat dari pengembangan FOSS, dan menjadi prinsip yang dianut semua lisensi FOSS.

Cerita lengkap terkait “source code” dapat ditemukan dalam bagian “*Glossary*” dari buku “*Free/Open Source Software, A General Introduction,*” yang tersedia

di http://iosn.net/download/foss_primer_current.pdf

14.11. Sub Licensing (Sub-lisensi)

Jika pemegang hak cipta (*licensor*) memberikan lisensi karyanya kepada seseorang (*licensee*), pemegang hak cipta juga dapat memberikan izin kepada penerima lisensi itu untuk membuat sub-lisensi karyanya. Misalnya ketika penerima lisensi menyebarluaskan karyanya dalam lingkup yang diizinkan pemberi lisensi, maka penerima lisensi tidak hanya sebagai distributor tetapi dapat juga menjadi licensor baru. Lisensi baru ini dinamakan sub-lisensi, yang berlaku antara penerima lisensi pertama (*sub-licensor*) dengan pihak lain (*licensee of sub-license*).

Namun, banyak lisensi FOSS yang tidak memberikan hak kepada orang lain untuk melakukan sub-lisensi. Misalnya, A adalah pemilik hak cipta pada program X, B penerima sebuah salinan X dan menyalurkan beberapa salinan. C menerima salinan dari B. Jika A tidak memberikan B hak untuk sub-lisensi program X, maka C dan B menerima lisensi langsung dari A. Tetapi jika A memberikan hak sub-lisensi kepada B, maka B bisa memulai sebuah lisensi baru sehingga B menjadi pemberi lisensi (*sub-licensor*) pada program X.

14.12. Warranty Disclaimer (Penolak Jaminan)

Warranty adalah garansi atau jaminan dari sebuah produk. *Warranty Disclaimer* selalu dapat ditemukan pada semua lisensi FOSS. Pernyataan ini dibuat untuk melindungi pencipta program FOSS yang dilisensikan tanpa royalti, dan perubahan dapat ditambahkan selama dalam pengembangan. Meskipun program FOSS bebas royalti dan menolak garansi, namun produsen atau vendor program FOSS bisa menyediakan garansi dan berbagai tingkatan support berbasis biaya kepada pelanggannya.