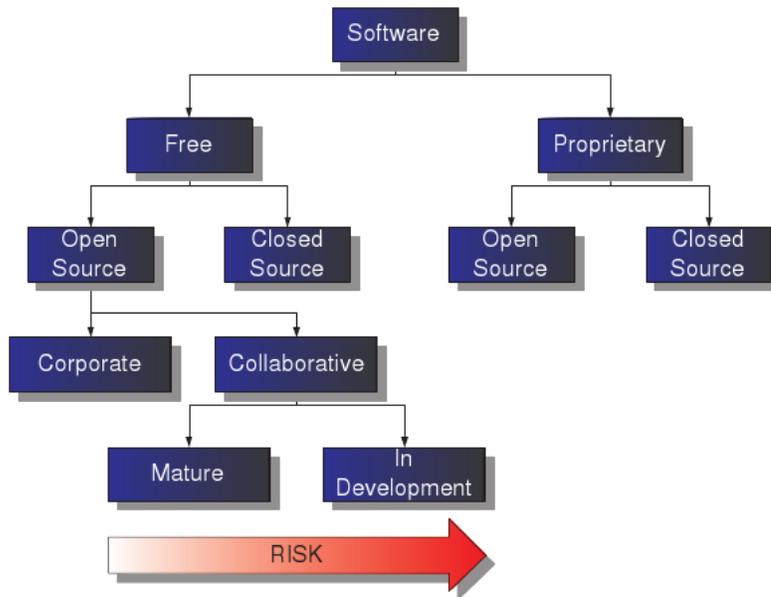


Bab 2 Berkenalan Dengan Dunia Open Source

Taksonomi Software

Sebelum berkenalan dengan dunia open source, ada baiknya kita kenali dahulu berbagai jenis software menurut biaya lisensi dan ketersediaan *source-codenya*. Menurut Robert Charpentier dan Richard Carbone (2004), berbagai jenis lisensi software dapat disusun dalam taksonomi berikut:



Taksonomi Perangkat Lunak

Pada gambar di atas, ada dua lisensi utama, yaitu free (gratis) dan proprietary yang meminta kompensasi biaya atas pembelian atau penggunaan perangkat lunak. Pengembang pada kedua jenis lisensi utama tersebut dapat menerapkan skema open source (menyediakan source code) maupun closed source (tidak menyediakan source code). Sehingga dapat ditemui adanya perangkat lunak gratis yang tidak open source, maupun perangkat lunak berbayar yang open source. Pada skema open source, ada yang dikembangkan oleh perusahaan (corporate) dan komunitas (collaborative). Perangkat lunak open source kolaboratif ada yang sudah matang (mature) dan sedang dalam pengembangan (in development). Bagi pemula, disarankan menggunakan perangkat lunak yang sudah matang. Karena mengandung resiko yang lebih kecil.

Secara sederhana, ada dua kategori lisensi software yang banyak dipakai, yaitu:

1. FOSS (Free / Open Source Software) adalah dua istilah yang maksudnya hampir sama, yakni program yang tidak perlu biaya izin (free = bebas) digunakan dan kode sumbernya

- tidak dirahasiakan (open = tersedia), sehingga cara kerjanya dapat dipelajari, lalu dikembangkan, dan disebarluaskan. Contoh: Linux, OpenOffice, GIMP, Inkscape.
2. PCSS (Proprietary / Closed Source Software) adalah program yang hanya dimiliki pembuatnya (terikat). Pengguna hanya dapat menggunakan jika membeli lisensi (mendapatkan izin). Pihak lain tidak dapat mempelajari cara kerjanya (tertutup), tidak pula mengembangkan dan menyebarkan. Contoh: Windows, MS Office, Photoshop, CorelDraw.

Seorang profesor dari universitas terkemuka di Amerika pada awal tahun 90-an pernah meramalkan, bahwa di masa depan nanti, hampir semua perangkat lunak di dunia akan diberikan secara gratis kepada siapa saja yang membutuhkan. Alasannya cukup sederhana sebagai berikut:

- perangkat lunak dibangun berdasarkan sebuah algoritma tertentu;
- algoritma pada hakekatnya berasal dari hasil ide atau buah pemikiran seseorang;
- akan semakin banyak orang yang memiliki ide dan pemikiran cemerlang sejalan dengan kebutuhan manusia yang bertambah dan berubah-ubah;
- karena banyak orang yang dengan rela dan kesadarannya kerap memberikan ide atau pemikirannya secara gratis kepada banyak orang, maka dengan kata lain perangkat lunak akan diberikan secara cuma-cuma kepada siapa saja yang membutuhkannya.

Pada mulanya profesor ini menyangka bahwa tipe perangkat lunak yang akan pertama-tama diberikan secara gratis dengan konsep “open source” adalah aplikasi, ternyata yang lahir terlebih dahulu adalah tipe sistem operasi – yaitu Linux. Karena sistem operasi merupakan dasar atau fundamental dari sebuah arsitektur perangkat lunak, maka perkembangan konsep “open source” yang diramalkan profesor tersebut percepatannya lebih tinggi dari yang dibayangkan.

Apa itu Open Source

Open Source tidak hanya bermakna kebebasan akses ke source code saja. Open source juga merupakan:

- Sebuah komunitas kuat yang terdiri dari individu-individu yang lebih mengutamakan kepentingan dan kesejahteraan umum dibandingkan dirinya sendiri;
- Seperangkat aturan lisensi software; open source bukan berarti tanpa lisensi, sebab ini berkaitan dengan hukum. Agar open source dapat menjadi legal di mata hukum, diperlukan aturan lisensi open source;
- Sebuah model pengembangan software secara kolaboratif; setiap orang dapat ikut berpartisipasi dalam mengembangkan software;
- Sebagai katalis yang membangkitkan bisnis dan model bisnis yang belum pernah ada sebelumnya; tidak ada bisnis dalam sistem open source itu sendiri, karena ia hanyalah alat; namun open source dapat digunakan untuk menjalankan bisnis dengan lebih efisien atau mengembangkan model bisnis baru di sekitar pemanfaatan open source;
- Kekuatan yang mendorong percepatan software menjadi komoditi.

Sebagai seperangkat lisensi software, ada beberapa kriteria yang harus dipenuhi sebelum suatu software dapat disebut sebagai open source. Berikut ini definisi open source menurut lembaga nirlaba *Open Source Initiative* (OSI):

1. Distribusi Ulang Secara Bebas

Lisensi yang digunakan tidak boleh membatasi siapa pun untuk menjual atau mendistribusikan ulang. Baik distribusi ulang secara terpisah maupun digabungkan dengan program lain. Lisensi tidak boleh mesyaratka royalti atau semacamnya bila program tersebut akan dijual.

Alasan Logis: Dengan mensyaratkan distribusi ulang secara bebas, hilangnya manfaat jangka panjang demi hasil penjualan jangka pendek dapat dieliminasi.



Situs web Open Source Initiative

2. Kode program (Source Code)

Distribusi program harus menyertakan source code, dan harus mengizinkan distribusi source code sebagaimana halnya bentuk yang sudah dikompilasi (bentuk binari/executable). Jika program tidak didistribusikan bersama source code, harus ada publikasi/penjelasan yang memadai bagaimana caranya mendapatkan source codenya. Biaya yang diperlukan untuk mendapatkan source code tidak boleh lebih dari biaya reproduksinya atau tersedia untuk didownload (diunduh) melalui internet. Source code harus menjadi bentuk yang lebih disukai jika programmer ingin memodifikasi programnya. Source code tidak boleh diubah atau dibuat menjadi tidak jelas dengan sengaja. Bentuk intermediate juga tidak diijinkan, misalnya keluaran dari preprosesor atau translator.

Alasan Logis: Akses ke source code yang jelas diperlukan untuk mengembangkan dan memodifikasi program. Agar hal tersebut dapat dilakukan dengan mudah, maka akses ke source code harus dimudahkan.

3. Hasil Modifikasi atau Turunan

Lisensi harus mengizinkan modifikasi atau pembuatan turunan dari program tersebut, dan harus mengizinkan program yang diturunkan untuk dilisensikan dengan lisensi yang sama dengan program aslinya.

Alasan Logis: Ketersediaan akses untuk membaca source code saja tidak cukup untuk mendukung peer review secara independen dan pengembangan evolusioner yang cepat. Agar hal tersebut dapat terjadi, diperlukan eksperimen pada source code dan distribusi ulang hasil modifikasinya.

4. Integritas Programmer Asli

Lisensi dapat melarang source code untuk didistribusikan dalam bentuk yang sudah dimodifikasi bila mengizinkan distribusi patch beserta source codenya untuk memodifikasi program pada saat

build time. Lisensi harus secara eksplisit mengizinkan distribusi program yang dibangun dari source code yang telah dimodifikasi. Lisensi dapat mensyaratkan program turunan agar menggunakan nama atau logo yang berbeda dengan program yang asli.

Alasan Logis: mendorong terjadinya banyak pengembangan dan perbaikan adalah hal yang sangat baik, namun para pengguna berhak untuk mengetahui siapa yang bertanggung jawab terhadap aplikasi yang mereka gunakan. Sebaliknya, sang pembuat program dan pemeliharanya (maintainer) juga berhak untuk mengetahui apa harus mereka dukung dan untuk menjaga reputasi mereka.

Dengan demikian, sebuah lisensi open source harus menjamin agar source code tersedia, namun dapat mensyaratkan agar source code didistribusikan dalam bentuk pristine (masih asli dari sang pembuat, belum ada modifikasi dari pihak lain) ditambah dengan patch (tambalan). Dengan cara ini, perubahan yang tidak resmi dapat dibuat dan disediakan, namun dapat dibedakan dari source aslinya.

5. Tidak Ada Diskriminasi Pada Orang Atau Kelompok Orang

Lisensi tidak boleh membatasi orang atau kelompok orang untuk menggunakan atau terlibat dalam proses pengembangan program open source.

Alasan Logis: Untuk mendapatkan manfaat yang optimal dari proses pengembangan aplikasi open source, maka tingkat perbedaan orang atau kelompok orang yang terlibat dalam prosesnya juga harus maksimal. Setiap orang harus memiliki hak yang sama untuk berkontribusi pada proyek open source apa pun. Dengan kata lain, tidak boleh ada larangan bagi siapapun untuk terlibat dalam proses pengembangan open source.

6. Tidak Ada diskriminasi Dalam Bidang Penggunaannya

Lisensi tidak boleh membatasi seseorang untuk menggunakan program yang dimaksud dalam bidang tertentu. Misalnya, lisensi tidak boleh membatasi penggunaan program dalam bidang penelitian, pendidikan, atau digunakan untuk menjalankan bisnis.

Alasan Logis: Hal ini dimaksudkan agar penggunaan open source meluas dan tidak terjebak pada batasan untuk digunakan sebagai alat bantu dalam dunia bisnis komersial. Pengguna komersial justru diharapkan bergabung dengan komunitas open source dan tidak merasa dikecualikan dalam menggunakan program open source.

7. Distribusi Lisensi

Hak-hak yang melekat pada program harus dapat diterapkan pada seluruh pengguna; tanpa memerlukan tambahan lisensi.

Alasan Logis: klausula ini dimaksudkan untuk menghindari penutupan software secara tidak langsung.

8. Lisensi Tidak Boleh Spesifik pada Produk tertentu

Hak-hak yang melekat pada program tidak boleh mensyaratkan program tersebut menjadi bagian dari distribusi software tertentu. Jika program tertentu digunakan atau didistribusikan secara terpisah dari distribusi softwarentya, namun tetap mengikuti lisensi berlaku pada program tersebut, maka seluruh pihak yang menerima atau menggunakan program tersebut harus menerima hak yang sama dengan mereka yang mendapatkannya bersama distribusi software aslinya.

Alasan Logis: klausul ini mencegah jenis jebakan lisensi yang lain.

9. Lisensi Tidak Boleh Membatasi Software Lain

Lisensi tidak boleh membatasi software lain yang didistribusikan bersama program yang dilisensikan. Misalnya, lisensi tidak boleh memaksa bahwa program lain yang didistribusikan dalam media yang sama harus merupakan software yang open source.

Alasan Logis: Distributor software open-source memiliki hak untuk menentukan pilihan mengenai software mereka. Lisensi GPL (GNU General Public License) juga mengadaptasi hal ini. Software yang menggunakan pustaka berlisensi GPL hanya diharuskan berlisensi GPL bila membentuk satu software yang sama, bukan pada software apa saja yang didistribusikan bersamanya.

10. Lisensi Harus Netral Terhadap Teknologi

Penyediaan lisensi tidak boleh mengharuskan penggunaan teknologi atau tampilan grafis tertentu.

Alasan Logis: Penyediaan lisensi ini ditujukan secara spesifik pada lisensi yang mengharuskan adanya tindakan yang secara eksplisit menunjukkan ekspresi persetujuan dan mengadakan kontrak antara pengguna software yang dilisensikan dengan pembuat lisensinya. Penyediaan lisensi yang mengharuskan "click-wrap" dapat menimbulkan konflik dengan beberapa metode penting dalam distribusi software seperti misalnya: download melalui FTP (*File Transfer Protocol*), CD-ROM berisi kumpulan aplikasi, atau mirror web; beberapa di antaranya dapat menghalangi atau mencegah penggunaan kembali kode program. Maka adaptasi penyediaan lisensi harus memungkinkan (a) distribusi software bisa dilakukan di jalur non-web yang tidak mendukung click-wrap pada proses download dan (b) kode program yang tercakup dalam lisensi (atau penggunaan kembali sebagian dari kode program yang tercakup) harus dapat dijalankan dalam lingkungan tanpa tampilan grafis yang tidak dapat mendukung dialog pop-up.

Pengertian GNU dan Free Software

Pengertian GNU

Proyek GNU (baca: guh-new) diluncurkan pada 1984 untuk mengembangkan sistem mirip Unix yang merupakan free software, yaitu sistem GNU. Kernel GNU belum selesai dibangun, sehingga sistem GNU digunakan bersama dengan kernel Linux. Kombinasinya menghasilkan sistem operasi lengkap yang disebut GNU/Linux, yang kini digunakan oleh jutaan orang. Terkadang kombinasi ini disebut kurang tepat dengan Linux saja. GNU adalah akronim dari GNU's not Unix, diucapkan sebagai g-noo sebagai satu suku kata tanpa huruf vokal di antara g dan n.

Kernel Linux memang dilepas secara cuma-cuma oleh Linus Torvalds. Begitu pula dengan berbagai software dari proyek GNU (GNU is Not UNIX) yang dirintis oleh Richard M Stallman (RMS) dengan Free Software Foundation-nya. Akan tetapi Richard sangat tidak setuju dengan pemahaman bahwa free adalah gratis. Free yang ia maksud adalah freedom atau kebebasan. Free di sini tidak ada kaitannya dengan uang. Siapa pun bisa menjual atau mengambil manfaat dari free software. Dia mendefinisikan kebebasan dalam empat tingkat, dari nol sampai tiga.

Pengertian Free Software

Untuk memahami konsep free ini, anda bisa menyamakannya dengan kebebasan mengemukakan pendapat dan gagasan atau *free speech*. Jangan disamakan dengan minuman atau makanan gratis.

Free software merujuk pada kebebasan pengguna untuk menjalankan, menyalin atau memperbanyak, mendistribusikan, mempelajari, mengubah, dan mengembangkan software. Lebih tepatnya, ia dirinci dalam empat kebebasan:

- Kebebasan nol artinya kebebasan menggunakan untuk kepentingan apa saja.
- Kebebasan satu untuk mempelajari dan memodifikasi sesuai kebutuhan anda. Akses ke source code menjadi prasyarat agar hal ini dapat dilakukan.

- Kebebasan dua untuk disebarluaskan atau bebas dikopi agar anda bisa membantu tetangga atau lingkungan sekitar anda dalam menggunakan software tanpa takut melanggar lisensi.
- Kebebasan tiga adalah untuk meningkatkan kemampuan program yang anda gunakan dan menyebarluaskan kembali hasil modifikasi anda ke publik. Dengan demikian, masyarakat luas dapat mengambil manfaat dari pengembangan yang anda buat. Akses ke source code juga menjadi prasyarat agar hal ini dapat dilakukan.

“Free software” lebih mementingkan pada kebebasan bukan harga. Hal ini yang seringkali membingungkan banyak orang. Banyak yang mengartikannya sebagai barang gratisan. Sehingga akhirnya Eric S. Raymond dan Bruce Perens mengusulkan istilah open source agar tidak disalahpahami orang. Buku ini menggunakan istilah open source juga dengan maksud yang sama. Meski demikian, pada dasarnya, free software dan open source software mengandung pengertian yang hampir sama.

Mereka yang menggunakan istilah open source seringkali memiliki niat dan penekanan yang berbeda. Istilah open source software biasanya digunakan oleh orang yang ingin menekankan pada aspek reliabilitas yang tinggi, fleksibilitas program yang dihasilkan sebagai motivasi utama dalam mengembangkan software. Sebaliknya, istilah free software lebih menekankan pada kebebasan dari kendali orang lain, atau barangkali lebih tepat bebas dari kendali vendor software.

Secara kontras, OSI menyatakan bahwa istilah open source adalah program pemasaran untuk free software dan merekomendasikan penggunaan istilah open source.

Free Software Foundation

Free Software Foundation (FSF) adalah organisasi utama yang menjadi pendukung proyek GNU. FSF menerima dukungan dana dari berbagai perusahaan atau yayasan, namun juga tergantung pada dukungan dari individu seperti anda. FSF meminta setiap orang agar bergabung menjadi anggota asosiasi, membeli manual, atau dengan menyumbangkan dana. Jika anda menggunakan free software dalam bisnis, anda bisa mempertimbangkan untuk membeli distribusi deluxe dari sistem GNU sebagai salah satu cara untuk memberi dukungan pada FSF.

Proyek GNU mendukung misi dari FSF untuk menjaga, melindungi, dan mempromosikan kebebasan untuk menggunakan, menyalin atau memperbanyak, meningkatkan kemampuannya, dan merilisnya ke publik kembali, serta mempertahankan hak-hak pengguna free software. FSF juga mendukung kebebasan berpendapat, penyebaran literatur, dan asosiasi di Internet, hak menggunakan perangkat lunak enkripsi untuk komunikasi pribadi, dan hak untuk membuat perangkat lunak yang tidak diganggu oleh monopoli.



Situs web Free Software Foundation

Dukungan untuk Open Source

United Nation Conference on Trade Development pada tahun 2003 merekomendasikan penggunaan free dan open source software di negara berkembang untuk mengurangi kesenjangan teknologi dengan negara maju.

Penggunaan GNU/Linux pun sudah meluas ke berbagai negara, diantaranya: Amerika Serikat, Estonia, Peru, Srilangka, Belanda, Cina, Filipina, Jepang, India, Malaysia, Thailand, Nigeria, Brasil, Kuba, Spanyol, Jerman, Polandia, dan Afrika Selatan. Sementara itu, pemanfaatan GNU/Linux di lembaga pemerintahan Indonesia sudah dilakukan di Depkominfo (Departemen Komunikasi dan Informatika), Depsos (Departemen Sosial), Pemerintah Provinsi Nangroe Aceh Darussalaam, Pemerintah Provinsi Daerah Istimewa Yogyakarta, Pemerintah Provinsi Jawa Tengah, BATAN, BAPETEN, dan Pemerintah Kabupaten Kebumen. Penggunaan GNU/Linux pada perusahaan swasta di Indonesia jauh lebih banyak. Hal ini dikarenakan sektor swasta benar-benar memikirkan efisiensi dan prinsip untung rugi yang ketat.

Penggunaan GNU/Linux dan open source software dipicu oleh berbagai keunggulan yang dimilikinya dibandingkan sistem operasi Windows. Keunggulan tersebut antara lain adalah ketersediaan source code. Terbukanya source code membuahakan kebebasan untuk mempelajari dan mengembangkan lebih lanjut. Tidak heran bila penerapan teknologi baru jauh lebih cepat. Misalnya 3D Desktop yang baru ada di Windows Vista sudah ada di Linux sebelumnya.

Selain itu, Linux juga lebih stabil dan aman dari serangan virus. Linux pun sudah didukung oleh berbagai vendor besar. Fujitsu, Hitachi, HP, IBM, Intel, NEC, Novell, dan Oracle telah bergabung membentuk konsorsium nirlaba bernama The Linux Foundation. Ketersediaan aplikasi di Linux pun sudah dapat memenuhi kebutuhan komputasi sehari-hari. Penggunaan Linux juga dapat meningkatkan martabat bangsa karena menurunkan tingkat pembajakan. Selain itu, penguasaan teknologi informasi akan meningkat. Kita tidak akan terus-terusan hanya menjadi pemakai, akan tetapi juga dapat berpartisipasi langsung pada pengembangannya.

Linux Foundation

The Linux Foundation merupakan pemberi dukungan utama terhadap perkembangan Linux. Lembaga ini didirikan pada tahun 2007 dengan menggabungkan Open Source Development

Labs (OSDL) dan Free Standard Groups (FSG). The Linux Foundation adalah konsorsium nirlaba yang bertujuan membantu mengembangkan pertumbuhan Linux. Kegiatan utamanya adalah mempromosikan, melindungi, dan membuat standar Linux dengan menyediakan layanan dan sumber-sumber standar yang dibutuhkan oleh open source untuk bersaing dengan *closed platform*.



Perlindungan terhadap Linux dilakukan dengan menjadi sponsor bagi Linus Torvalds dan pengembang utama kernel Linux. Hal ini dilakukan agar mereka dapat tetap independen dan bekerja secara full-time untuk mengembangkan Linux. Kekuatan sebuah platform diukur dari kekuatan aplikasi pendukungnya. Maka The Linux Foundation juga menyediakan dukungan dan layanan standarisasi bagi pengembang aplikasi di Linux, termasuk Linux Standard Base (LSB) dan Linux Developer Network. Sehingga diharapkan banyak developer akan tertarik mengembangkan aplikasi untuk platform Linux.

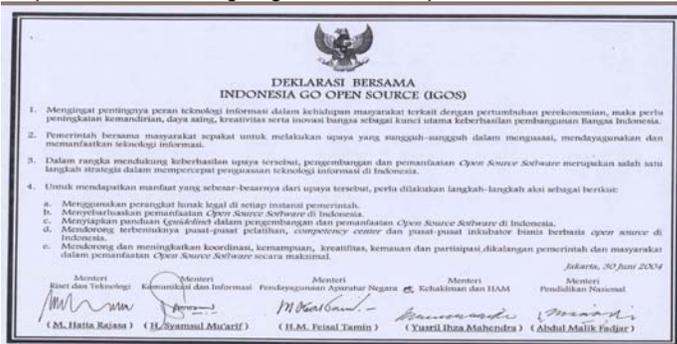
Anggota Platinum sekaligus pendiri The Linux Foundation adalah Fujitsu, Hitachi, HP, IBM, Intel, NEC, Novell, dan Oracle. Sedangkan Gold Member antara lain AMD, Cisco, Google, Motorola. Selain itu, masih ada Silver Member yang terdiri dari Adobe, Canonical, Dell, Konica-Minolta, Mitsubishi, Montavista, Nokia, Toshiba, Redhat, VMWare, Sun Microsystems, dan NTT. Jumlah anggota dapat bertambah setiap saat. (LinuxFoundation.org, 2008)

Dukungan Open Source di Indonesia

Pada 30 Juni 2004, lima menteri di pemerintahan Megawati yaitu Menteri Negara Riset dan Teknologi, Menteri Negara Komunikasi dan Informasi, Menteri Negara Pendayagunaan Aparatur Negara, Menteri Kehakiman dan HAM, serta menteri Pendidikan Nasional secara resmi menyatakan akan menggalakkan penggunaan standar software terbuka melalui gerakan Indonesia Go Open Source (IGOS) yang konon dapat menghemat belanja sampai 20 triliun rupiah.

Kemudian pada IGOS Summit II tanggal 27-28 Mei 2008, 18 kementerian dan kepolisian turut menandatangani MoU lanjutan, yaitu Departemen Komunikasi dan Informatika, Kementerian Negara Ristek, Sekretariat Negara, Kementerian Negara BUMN, Kementerian Negara PAN, Bappenas, Departemen Koperasi dan UKM, Departemen Perhubungan, Departemen Pendidikan

Nasional, Departemen Perindustrian, Departemen Sosial, Departemen Agama, Departemen Kehutanan, Departemen Keuangan, Departemen Dalam Negeri, Departemen Hukum dan HAM, Departemen Perdagangan, serta Kepolisian RI.



Deklarasi Indonesia Go Open Source

Keuntungan Pemanfaatan Open Source

Model open source memiliki banyak hal yang dapat ditawarkan kepada dunia bisnis. Open source merupakan jalan untuk membuat standar terbuka dalam dunia software, bukan hanya di atas kertas atau teori saja. Sebuah jalan di mana berbagai perusahaan maupun individu bekerja sama dalam suatu sistem kolaborasi untuk menghasilkan sebuah produk yang tidak dapat dicapai bila dilakukan sendirian oleh masing-masing pihak. Selain itu, ia juga menawarkan perbaikan *bug* secara sangat cepat, menerapkan perubahan pada software sesuai keinginan user, sesuai jadwal yang diinginkan oleh user.

Model open source juga berarti peningkatan keamanan; sebab kode program terbuka untuk publik dan dapat diteliti baris demi baris; dan permasalahan atau bug ditemukan dan diperbaiki dengan cepat; bukannya malah disembunyikan/dirahasiakan sampai ditemukan oleh orang yang salah. Ia juga merupakan jalan bagi kumpulan individu dapat bekerja sama dan memiliki kesempatan untuk mengalahkan monopoli. Peningkatan reliabilitas merupakan hal terpenting di antara keuntungan-keuntungan dari pemanfaatan open source. Pada kasus Year 2000 / Millenium bug, sistem closed source menjadikannya lebih buruk dan dapat membunuh bisnis anda dengan sangat baik.

Mengenai Reliabilitas

Gerald P. Weinberg pernah menyatakan dengan yakin bahwa, "Jika para insinyur membangun rumah-rumah sebagaimana para programmer membangun program, maka burung pelatuk pertama yang datang akan menghancurkan seluruh peradaban". Dia benar. Hingga saat ini, reliabilitas sebagian besar perangkat lunak sangatlah buruk.

Fondasi dari masuknya open source ke dunia bisnis adalah reliabilitas yang tinggi. Perangkat lunak open source adalah perangkat lunak yang telah mengalami *peer-review*. Alias telah dievaluasi secara profesional oleh berbagai pihak yang berpartisipasi. Sehingga memiliki reliabilitas lebih tinggi dibandingkan perangkat lunak closed-source atau proprietary. Kode program dari perangkat lunak open source adalah perangkat lunak setangguh rompi anti peluru yang paling bagus yang pernah ada.

Hingga beberapa waktu terakhir, hal ini dianggap sebagai ide radikal oleh banyak orang yang bergelut di dunia bisnis. Banyak yang meyakini bahwa perangkat lunak open source tidak dibuat secara profesional, yang dibuat dengan murah dan berkonotasi murahan sehingga lebih mudah mengalami kerusakan atau kegagalan daripada perangkat lunak proprietary.

Akan tetapi infrastruktur internet menjadi bukti yang paling nyata mengenai hal ini. Dan sejak OSI (Open Source Initiative) didirikan pada 1998, banyak orang mulai memberikan perhatian. Coba amati DNS, sendmail, berbagai implementasi TCP/IP open source, dan bahasa pemrograman skrip seperti Perl dan PHP yang berada di balik sebagian besar situs web yang ada. Mereka adalah mesin utama internet. Bayangkan bila berbagai produk open source tersebut tidak ada. Bisakah anda bayangkan hidup tanpa internet? Windows pun bisa menjelajah internet menggunakan produk open source tersebut sebagai mesinnya.

Bila anda masih belum percaya, silahkan buka sebuah halaman web. Anda bisa menggunakan browser dan sistem operasi apapun. Anda akan selalu bisa melihat bagaimana halaman web tersebut dibuat. Sebab, selalu ada menu View – Source atau View – Page Source. Ya, internet dan HTML adalah produk open source. Sehingga anda selalu bisa mempelajari bagaimana orang lain membuat tampilan web yang atraktif. Anda bisa melihat kode program htmlnya hanya dengan klik menu View – Source atau View – Page Source.

Program-program open source tersebut telah menunjukkan sebuah tingkatan reliabilitas dan kekuatan untuk menangani perubahan kondisi yang sangat cepat. Misalnya, pertumbuhan dan ukuran internet yang berkembang begitu cepat dan menjadi begitu besar. Sehingga tidak perlu berdecak kagum bila dihadapkan dengan perangkat lunak komersial.

Anda dapat membaca argumentasi teknis yang lebih lengkap mengenai reliabilitas perangkat lunak secara umum dalam makalah "The Cathedral and the Bazaar". Makalah ini berisi tentang contoh kasus Netscape yang menjadi pionir dalam memutuskan melepas client softwrenya menjadi proyek open source. Makalah tersebut menjelaskan metode bazaar dalam mengelola pengembangan perangkat lunak open source dan mengarahkan pada reliabilitas dan kualitas yang tinggi.

Bukti-bukti dari dunia nyata mendukung hal ini. Dalam uji reliabilitas, sistem Unix open source lebih jarang mengalami crash atau hang dibandingkan perangkat lunak proprietary sejenis.

Dampak dari fakta teknis ini bagi dunia bisnis begitu jelas. Dalam jangka panjang, peer-review dalam metode bazaar dianggap sebagai kondisi yang penting untuk memperoleh kualitas terbaik dari sebuah perangkat lunak. Di berbagai pasar yang sesuai, perangkat lunak yang belum pernah mengalami peer-review tidaklah mudah diakui sebagai perangkat lunak yang cukup bagus untuk bersaing dengan yang lain.

Keuntungan bagi Pembuat Software

Pengembangan software dengan model Bazaar tampak seolah berlawanan dengan pandangan normal mengenai pengembangan software. Semakin banyak programmer akan semakin baik, setidaknya selama pemimpin proyek (project leader) masih mampu menangani integrasinya). Bahkan proyek open source kecil pun dapat mengarahkan beberapa pikiran untuk meningkatkan kemampuan sebuah software dibandingkan dengan yang dapat dicapai oleh model retail. Ada beberapa keuntungan pada model open source baik untuk pembuatan software yang hanya akan digunakan secara internal maupun yang akan dijual kembali.

Pengembangan yang Cepat

Merupakan hal yang dapat dimaklumi jika pengembang software komersial dapat berpartisipasi dalam model bazaar, dan mendapat keuntungan besar dibandingkan dengan mereka yang tidak. Pengembang software komersial pertama dalam pasar software yang sesuai dan beralih ke model ini bisa mendapat keuntungan besar dibandingkan yang belakangan mengikutinya.

Mengapa? Sebab, sumber talenta yang tersedia untuk rekrutmen dalam model bazaar sangatlah terbatas. Maka proyek bazaar pertama dalam pasar tertentu dapat menarik perhatian co-developer terbaik untuk menginvestasikan waktunya. Ketika mereka telah menginvestasikan waktunya, umumnya mereka akan konsisten dalam proyek tersebut.

Overhead yang Lebih Rendah

Beralih ke model open source juga dapat mengurangi biaya overhead dalam biaya produksi

software per proyek. Model open source juga memungkinkan penjual software untuk meng-*outsource*-kan (sebagai efeknya) sebagian dari pekerjaan mereka, dan membayarnya dengan nilai tambah yang mungkin kurang terlihat dibandingkan dengan uang. (Namun bisa jadi tidaklah kurang signifikan secara ekonomis; kecepatan yang meningkat karena adanya co-developer dari pihak luar yang dapat memberikan perbaikan bug bisa diterjemahkan menjadi peluang besar untuk mendapat keuntungan). Ini berarti penjual software skala kecil bisa menangani proyek yang lebih besar.

Keuntungan bagi Pedagang Software

Jika anda membuat software untuk dijual, akan ada dua keuntungan tambahan yang dapat terlihat:

Kedekatan dengan Pelanggan

Salah satu bagian dari manajemen yang paling sering diulang-ulang adalah, "Tetaplah dekat dengan kustomer". Dalam dunia bisnis yang bergerak cepat seperti sekarang, iklim bisnis dalam siklus produk yang pendek, hal tersebut menjadi jauh lebih penting. Anda harus mengerti apa yang diinginkan para pelanggan dan menjawab kebutuhan tersebut secepatnya.

Jika anda menjual software, apalagi yang lebih baik daripada bekerja sama dengan programmer dari sisi pelanggan untuk membantu pengembangan software anda?

Ada contoh menarik bahwa metode bazaar, open source, mirip dengan yang dilakukan oleh banyak perusahaan Jepang dalam mengembangkan produk untuk pelanggannya. Mereka melepas produk yang sudah dapat digunakan ke pasaran, namun produk tersebut belum sempurna. Kemudian mereka melakukan perulangan yang cepat berdasarkan umpan balik dari para pelanggannya untuk menghasilkan kombinasi fitur yang dibutuhkan dan diinginkan oleh para pelanggan. Hal ini telah terbukti sangat berguna, terutama untuk produk-produk berteknologi tinggi (laptop, PDA, ponsel, dll) pada keadaan orang-orang tidak mengetahui apa yang mereka butuhkan atau fitur apa yang mereka inginkan.

Pasar yang Lebih Luas

Efek samping yang cukup berarti pada model open source adalah perluasan platform untuk produk anda. Pengembang versi orisinal dari produk open source sering kali menerima *porting* aplikasinya untuk sistem operasi dan environment yang jelas mereka ketahui adanya, namun tidak dapat menyediakan pengembang untuk memberikan dukungan. Setiap *port*, tentu saja memperluas pasar untuk produk tersebut.

Keuntungan bagi para Pengusaha

Bagi seorang pengusaha atau produsen software start-up, beralih ke open source adalah sebuah cara untuk berbagi pikiran. Konsep baru yang paling baik di dunia tidak akan menghasilkan uang kecuali orang-orang mengetahui bahwa hal tersebut menarik.

Apakah ini bisa dianggap logis sebagai sebuah strategi bergantung pada apakah anda berpikir bahwa nilai utama yang dapat dijual oleh bisnis anda adalah pada software itu sendiri atau pada layanan dan keahlian berkaitan dengan software tersebut. Seringkali, nilai lebihnya ada pada layanan dan integrasi.

Contohnya adalah Digital Creations meng-*open source*-kan proyek Zope sesuai saran dari *Venture Capitalist* mereka. Investor mereka tersebut memperkirakan bahwa beralih ke open source dapat meningkatkan nilai perusahaan mereka.

Pertanyaan dan Keraguan pada Open Source

Beberapa pihak menghindari open source atau free software karena beberapa hal yang mereka takutkan. Mari kita jawab beberapa ketakutan dan keraguan mereka.

Dukungan perangkat lunak open source

Tidak. Saat ini ada dua jenis dukungan untuk perangkat lunak open source: dukungan berbayar tradisional dan dukungan informal dari komunitas. Perlu dicatat juga bahwa vendor-vendor proprietary seringkali tidak lagi mendukung produk-produk mereka setelah beberapa waktu; tidak ada perubahan pengguna produk proprietary menjadi pemandu bagi yang lain, sedangkan di dunia open source selalu ada orang yang lebih dulu belajar dan dengan senang hati mengajari yang lain.

Banyak organisasi yang menyediakan jenis dukungan tradisional berbayar untuk produk open source; karena harga dan kualitas layanan dapat dikompetisikan (sebuah pilihan yang tidak ada pada proprietary yang biasanya berupa vendor tunggal), anda seringkali bisa mendapat harga yang sangat bagus. Seorang pengacara anti-trust akan mengatakan bahwa dukungan untuk produk open source dapat dipilih dari beberapa yang ada. Banyak distribusi GNU/Linux yang menyertakan dukungan instalasi jika anda membelinya; dan dengan biaya tertentu anda bisa mendapat dukungan lebih. Contohnya adalah perusahaan seperti Redhat, Novell SUSE, Mandriva (sebelumnya dikenal sebagai MandrakeSoft), dan Canonical Ltd (yang memberikan dukungan untuk Ubuntu, salah satu turunan Debian GNU/Linux yang lebih populer dibanding Debian sendiri).

Selain itu, banyak juga organisasi independen yang menyediakan dukungan tradisional berbayar. Beberapa proyek distribusi didukung secara aktif oleh himpunan besar yang terdiri dari berbagai perusahaan dan konsultan yang dapat dipilih. Contohnya termasuk Debian GNU/Linux dan OpenBSD. Artikel berjudul *'Team'work Pays Off for Linux* telah mengevaluasi empat layanan dukungan teknis yang berbeda untuk sistem GNU/Linux dan menemukan bahwa "tingkat respon tidak menjadi masalah bagi pelanggan mana pun" serta "tidak ada vendor yang gagal menyelesaikan masalah yang diberikan padanya".

Banyak organisasi lain yang muncul untuk mendukung produk yang spesifik. Misalnya dukungan untuk Mozilla Firebird dan Thunderbird disediakan oleh decisionOne dan MozSource. Bertahun-tahun AdaCore (aka AdaCore Technologies atau ACT) telah menjual dukungan komersial untuk kompilasi Ada GNAT dan MySQL AB menjual dukungan komersial untuk database relasional mereka. Sangat penting untuk dipahami bahwa dukungan untuk produk open source dapat bersaing secara terpisah dari produknya. Sedangkan produk proprietary biasanya terikat dengan pembelian lisensi penggunaan softwaranya.

Sementara itu, para pengguna dapat meminimalkan resiko kesesuaian produk dengan kebutuhan melalui proses evaluasi dan pengujian, serta dengan hanya menggunakan rilis untuk penggunaan produktif yang stabil, matang, dan dikenal baik dari distributor yang reputasinya baik. Bahkan, prediksi ini terlihat hampir pasti, karena ini sedang terjadi dan prosesnya semakin berkembang cepat dalam beberapa tahun ini.

Sebagai alternatif dari dukungan berbayar, anda juga bisa mendapat dukungan tidak berbayar dari komunitas pengguna dan pengembang melalui newsgroup, mailing list, web site dan forum elektronik lainnya. Meskipun ini adalah jenis dukungan non-tradisional, banyak pihak yang merasa puas dengan jenis dukungan ini. Bahkan, pada 1997 InfoWorld memberikan penghargaan "Best Technical Support" kepada komunitas pengguna Linux (Linux User Community), mengalahkan semua vendor penyedia dukungan teknis.

Banyak yang percaya bahwa ini merupakan akibat sampingan dari pertumbuhan penyebaran Internet yang semakin meluas. Pengguna dan pengembang yang semakin banyak berkomunikasi satu sama lain secara langsung dan menemukan cara-cara yang lebih efektif di antara alternatif yang ada. Penggunaan pendekatan non-tradisional seperti ini juga perlu mengikuti beberapa aturan. Misalnya panduan "How to Ask Smart Question (Bagaimana Menanyakan Pertanyaan Cerdas)" atau "How to Report Bug Effectively (Bagaimana Melaporkan Bug Secara Efektif)". Menggunakan produk open source tidak mengharuskan anda untuk memakai dukungan non-tradisional (dan mengikuti aturannya), maka bagi yang ingin mendapat jaminan dari dukungan tradisional dapat membayar seperti pada produk proprietary.

Ya, perangkat lunak proprietary juga didukung secara informal. Kelompok pengguna, majalah, dan berbagai organisasi telah berdiri bertahun-tahun untuk memberi dukungan bagi produk

proprietary, bahkan bagi yang secara teori memiliki jaringan formal. Hal ini menunjukkan bahwa dukungan formal seringkali tidak efektif. Tidak seefektif yang diklaim oleh vendor proprietary. Namun, organisasi yang memberi dukungan pada produk open source memiliki kelebihan daripada produk proprietary berupa akses langsung ke kode program dan informasi pengembangan softwarentya, yang berarti bisa jauh lebih efektif dalam menangani masalah.

Dan penting untuk diingat bahwa pada produk proprietary, vendor dapat kapan saja menghentikan dukungan untuk produk tertentu. Sementara itu selalu ada alternatif untuk pengguna produk open source. Hal ini bisa menjadi resiko bila perusahaan vendor mengalami kebangkrutan atau tidak lagi menjalankan bisnisnya, dibeli oleh perusahaan lain, berpindah pada pasar yang lain, atau bila pasarnya menjadi terlalu kecil. Namun hal ini juga bisa terjadi ketika vendor menanggung keuntungan, tidak mengubah pasarnya, pasarnya pun besar, dan banyak pengguna produknya. Bagaimana pun juga, vendor dapat memiliki prioritas yang tidak sama dengan kebutuhan anda, dan vendor tersebut biasanya merupakan satu-satunya organisasi yang dapat memperbaiki dan meningkatkan kemampuan serta menjual software tersebut.

Sebuah contoh ekstrim bagaimana sebuah vendor bisa meninggalkan penggunanya begitu saja adalah Microsoft yang menghentikan dukungan untuk Visual Basic 6 (pada Maret 2005) yang banyak dipakai oleh berbagai perusahaan. Banyak organisasi besar telah mengembangkan infrastruktur besar yang bergantung pada Visual Basic 6. Sebuah survei melaporkan bahwa 52% dari pengembang software menggunakan Visual Basic 6, setidaknya ketika dibutuhkan. Seorang pengembang piranti lunak memperkirakan bahwa keputusan Microsoft ini menelantarkan kurang lebih 18 juta pengembang software, dengan perkiraan 6 juta dari mereka adalah pekerja profesional, yang telah mengembangkan puluhan juta aplikasi.

Ketika Microsoft mengembangkan infrastruktur .NET-nya, ia juga membuat produk baru yang disebut "Visual Basic for .NET" (VB.NET). Sayangnya, VB.NET sama sekali tidak kompatibel dengan bahasa VB 6 yang telah banyak digunakan secara luas di dunia industri. Sehingga jutaan kode program yang telah dibuat dengan VB 6 tidak dapat dijalankan dengan VB.NET tanpa menulis ulang kode program-nya dari awal. Migration Wizard yang disediakan tidak membantu karena banyak sekali yang tidak kompatibel.

Kejadian ini mengakibatkan pengeluaran yang sangat banyak. Jika rata-rata biaya yang dibutuhkan untuk menulis ulang aplikasi Visual Basic adalah sebesar \$4,000, dan hanya 10% dari seluruh 30 juta aplikasi yang perlu ditulis ulang, hal ini berarti bahwa para pengguna VB harus membayar 12 milyar dolar hanya untuk menulis ulang aplikasi mereka (tanpa tambahan fungsionalitas).

Berbagai survei menunjukkan bahwa Visual Basic 6 masih jauh lebih populer daripada VB.NET; survei pada 2004 menemukan bahwa 80% masih menggunakan Visual Basic 5 atau 6, sementara hanya 19% yang menggunakan VB.NET. Sebuah petisi telah ditandatangani oleh 2000 orang (termasuk 222 MVP) untuk menyatakan protes terhadap keputusan Microsoft, dan banyak perusahaan yang menyatakan keberatan mengenai biaya yang besar dan seharusnya tidak perlu untuk menulis ulang aplikasi mereka hanya karena Microsoft tidak lagi memberikan dukungan pada bahasa asli VB 6.

Namun demikian, Microsoft telah memutuskan untuk menelantarkan Visual Basic 6 dan tidak mendengarkan ratapan dari sebagian besar pelanggannya. Karena Visual Basic tidak ada standarnya, dan implementasinya tertutup tanpa adanya alternatif, maka pengguna Visual Basic pun tidak dapat berbuat apa-apa. Mereka tidak bisa mengambil alih pengembangannya sebagaimana yang dapat terjadi pada perangkat lunak terbuka (free/open source software). Sebagai gantinya, sebagian besar pengembang Visual Basic berpaling pada bahasa pemrograman yang lain, utamanya C# dan Java.

Misalnya, survei Evans Data menemukan bahwa mereka yang memutuskan berpaling dari Visual Basic 6, hanya 37% dari mereka yang berencana menggunakan VB.NET; 31% lainnya berencana menggunakan Java dan 39% sisanya berencana menggunakan C#. Informasi lebih lanjut bisa didapatkan di ClassicVB.org.

Anda bisa saja mengatakan bahwa biaya besar yang tidak diinginkan ini adalah hukuman bagi pada pengembang software yang secara tidak bijaksana memilih sebuah bahasa pemrograman yang tidak memiliki standarisasi, tidak ada alternatif implementasi, dan tidak ada mekanisme untuk mendapatkan dukungan bila sang vendor memutuskan untuk mengakhiri dukungannya pada produk aslinya.

Sebaliknya, banyak aplikasi open source yang telah ditelantarkan atau mengalami perubahan besar dalam strategi sehingga berbeda dengan keinginan para penggunanya. Akan tetapi dukungannya tidak dihentikan. Apache lahir dari ditelantarkannya aplikasi web server NCSC. Akhirnya para pengguna bergabung bersama-sama dan mengulang proyek tersebut hingga menjadi web server nomor wahid.

The GIMP juga ditelantarkan oleh pengembang aslinya, sebelum dirilis secara penuh. Sekali lagi, pada pengguna bergabung bersama dan membangun kembali proyeknya. Proyek Xfree86 telah mengubah lisensinya sehingga tidak kompatibel dengan keinginan banyak penggunanya. Hal ini berakibat pada munculnya proyek lain yang menggantikannya. Tentu saja, jika anda adalah pengguna tunggal dari sebuah aplikasi open source, barangkali kurang berguna bila anda menjadi pemimpin dari proyek penerusnya. Namun setidaknya anda memiliki hak untuk melakukan hal tersebut. Sebuah proyek open source tidak dapat berjalan terlalu jauh dari keinginan para penggunanya, sebab para pengguna dapat mengambil alih kendali sewaktu-waktu.

Apakah Perangkat Lunak Proprietary Memberikan Legalitas yang Lebih? Atau, Bukankah Free/Open Source Software Lebih Beresiko di Mata Hukum?

Hal tersebut tidaklah benar. Beberapa orang memberikan komentar bahwa “menggunakan open source software berarti melepaskan hak-hak yang dapat dituntut bila ada yang tidak berjalan semestinya”. Kenyataannya, semua lisensi proprietary juga melanggar hukum, maka keduanya tidak berbeda. Sebagian besar pengguna juga tidak tertarik untuk menuntut vendor. Mereka hanya menginginkan sistem yang bekerja dengan baik.

Sebenarnya, dengan perangkat lunak open source, anda menyepakati lisensi dengan mata terbuka lebar. Anda tahu apa yang anda dapatkan, atau jika tidak, anda bisa mencari orang yang lebih mengetahui. Lisensi open source memungkinkan komunitas pengguna memeriksa kode program untuk menemukan kode-kode jahat dan lubang keamanan serta saling berbagi pengetahuan mengenai hal tersebut.

Beberapa lisensi mengizinkan para pengguna membuat versi turunan dari kode programnya untuk memperbaiki masalah yang mungkin muncul dan tidak terlihat oleh pembuat program yang asli. Mereka membebaskan pengguna untuk menentukan apakah programnya memiliki tingkat keamanan yang cukup atau tidak. Pada perangkat lunak proprietary, anda tidak diizinkan untuk melihat kode program dan mengubahnya. Anda tidak tahu apa yang anda dapatkan. Jika perangkat lunak anda mengalami masalah, dan harus segera diselesaikan, anda bisa memperbaikinya sendiri atau membayar orang untuk memperbaikinya. Pilihan ini sangat meminimalkan resiko, dan pilihan ini hanya ada pada perangkat lunak open source.

Ada perbedaan legalitas lain yang seringkali tidak disebutkan. Banyak program proprietary yang mensyaratkan bahwa pengguna harus mengizinkan pemeriksaan lisensi. Lalu jika organisasi yang diperiksa tidak dapat membuktikan bahwa seluruh perangkat lunak yang digunakannya memiliki lisensi, maka ia harus membayar sejumlah besar uang. Maka dalam beberapa kasus, bila anda menggunakan perangkat lunak proprietary, perbedaan legalitas terbesar adalah sang vendor bisa saja menuntut anda ke muka pengadilan.

Ada beberapa klaim bahwa program open source membawa resiko tertentu bagi penggunanya, namun hal ini tidak benar pada kenyataannya. BusinessWeek memuat opini Stuart Cohen dari Open Source Development Lab; ia menyatakan bahwa usaha SCO untuk menuntut IBM dalam isu-isu yang berkaitan dengan Linux malah berakibat meningkatnya popularitas Linux dan

menguatkan fondasi legalitasnya. Dia mencatat bahwa banyak pengembang Linux, dibantu oleh pihak-pihak yang berminat telah melakukan investigasi mendalam pada tuntutan yang diajukan SCO dan tidak menemukan pelanggaran yang dilakukan oleh Linux.

Bukankah aplikasi open source hanya menjiplak aplikasi proprietary?

Tidak. Seorang pemrogram yang memiliki akses ke kode program suatu aplikasi memang dapat saja mengambil kode program tersebut secara ilegal dan memasukkannya pada program lain yang mirip atau berkaitan. Ada alasan yang layak dijadikan dasar untuk mempercayai bahwa hal ini banyak terjadi ada perangkat lunak proprietary. Sebab, hanya beberapa orang yang bisa melihat dua kode program proprietary. Beberapa pemrogram mungkin melakukannya dengan keyakinan bahwa mereka tidak akan ketahuan.

Namun demikian, seorang pemrogram tidak akan menyalin kode program dari aplikasi proprietary ke dalam aplikasi open source tanpa izin, karena (1) kode program aplikasi open source terbuka bagi siapa saja di seluruh dunia, sehingga akan mudah bagi vendor proprietary untuk menemukan pelanggaran yang terjadi, dan (2) catatan yang jelas mengenai siapa yang menyumbangkan kode program yang dijiplak akan memudahkan penangkapan pemrogram yang melanggar hukum tersebut.

Beberapa tahun yang lalu, SCO Group mulai mengklaim bahwa kernel Linux mengandung jutaan baris kode miliknya, dan menuntut beberapa perusahaan termasuk IBM. Namun ketika diminta untuk menunjukkan bukti di pengadilan, SCO tidak dapat memberikan bukti bahwa kode program yang dimiliki SCO disalin ke kernel Linux. Bahkan, tidak jelas apakah kode program yang diklaimnya memang hak cipta milik SCO.

Sebagai tambahan, Open Source Risk Management (OSRM) telah melakukan analisis kode program secara detail dan pada April 2004 mengumumkan bahwa kernel Linux bebas dari pelanggaran hak cipta.

Pada 2004, Ken Brown, presiden ADTI (dibiayai Microsoft), mengklaim bahwa Linus Torvalds tidak menulis kode program Linux, dan mengklaim bahwa ia mencuri sebagian besar kode programnya dari Minix. Belakangan diketahui bahwa ADTI sebelumnya telah menyewa Alexey Toptygin untuk mencari bagian yang disalin pada Minix dan Linux dengan aplikasi otomatis, dan ia tidak menemukan kode program yang disalin.

Andrew Tanenbaum, sang pembuat Minix, menolak keras pernyataan Brown yang tidak berdasar. Tanenbaum dengan jelas menyatakan bahwa, "Linus Torvalds menulis sendiri kode program Linux, dan ia berhak dan layak untuk menjadi pemegang hak ciptanya". Tanenbaum juga mendiskreditkan pernyataan Brown bahwa tidak ada seorang pun yang mampu menulis kode program kernel dasar. Tanenbaum menyatakan, "ada 6 orang yang saya ketahui menulis (ulang) UNIX dan mereka semua melakukannya secara independen". Associated Press juga mencatat bahwa serangan terhadap Linux datang dari sumber yang meragukan.

Apakah Open Source Lebih Beresiko Dihentikan Dukungannya oleh Vendor?

Tidak. Banyak bisnis gagal atau merugi, dan banyak juga individu yang kehilangan minat pada produk software tertentu. Baik dalam dunia proprietary maupun dunia open source. Namun ada satu perbedaan mendasar dalam hal ini. Semua aplikasi open source, secara otomatis dapat diselamatkan. Jika pengembang aslinya berhenti memberikan dukungan, siapa saja (baik perorangan maupun kelompok) dapat mengambil alih proses pengembangan. Contohnya adalah GIMP dan Apache sebagaimana sudah disebutkan sebelumnya.

Pertanyaan selanjutnya, apakah lisensi open source dapat secara sah dipertanggungjawabkan secara hukum? Hampir semua aplikasi open source dirilis dengan lisensi tertentu. Lisensi yang paling populer adalah GPL (General Public License). Beberapa kompetitor sempat mengklaim bahwa lisensi open source tidak sah di mata hukum. Namun beberapa pengacara dan sekolah hukum berpendapat lain.

Eben Moglen, seorang profesor hukum di *Columbia University Law School*, sekaligus konsulat jenderal *Free Software Foundation*, menulis sebuah artikel yang berjudul *Enforcing the GNU GPL*. Dalam artikel tersebut, ia menjelaskan bagaimana GPL sangat mudah dipertanggungjawabkan secara hukum. Ia juga menyebutkan bahwa ia telah berhasil mempertahankan GPL secara hukum belasan kali. Bahkan tanpa harus membawanya ke pengadilan. Pada saat itu ia berkata, "kami tidak membawa GPL ke pengadilan karena belum ada yang berani mengambil resiko melawan kami di pengadilan".

Pada tahun 2004, GPL akhirnya diuji di pengadilan dan ternyata valid. Tiga hakim di pengadilan kota Munich, Jerman, memutuskan memerintahkan penghentian distribusi produk Sitecom yang diturunkan dari program GPL namun gagal memenuhi ketentuan lisensi GPL.

Persyaratan lisensi open source pada umumnya mudah dipenuhi. Meski demikian, sudah ada bukti kuat bahwa lisensinya sah di mata hukum. Hal ini merupakan kabar gembira bagi para pemakai perangkat lunak open source. Persyaratan lisensi yang jelas, sederhana, dan konsisten memudahkan pemahaman mengenai apa yang perlu dilakukan. Bagi para pengembang software ingin menjaga agar kode programnya selalu tersedia untuk dikembangkan melalui GPL, hal ini juga merupakan berita baik.

Apakah ada peraturan khusus mengenai penggunaan aplikasi open source di dalam aplikasi yang kita buat?

Tidak, pada dasarnya aturan yang sama berlaku baik anda memasukkan perangkat lunak open source ataupun proprietary dalam aplikasi yang anda buat. Pada dasarnya, anda hanya bisa memasukkan aplikasi yang dikembangkan oleh orang lain ke dalam perangkat lunak yang anda buat hanya jika anda memiliki lisensi untuk melakukannya. Selain itu, anda harus mengikuti persyaratan yang ada dalam lisensinya. Pada sebagian besar aplikasi proprietary, hal ini hanya bisa dilakukan dengan membayar royalti per unit atau membatasi penggunaannya (misalnya hanya boleh digunakan untuk pendidikan). Jika anda melanggar aturan tersebut, anda bisa dibawa ke pengadilan, tidak peduli apakah berupa aplikasi proprietary atau open source.

Banyak program proprietary yang memasukkan perangkat lunak open source di dalamnya. Hal ini menunjukkan bahwa hal ini dapat dilakukan secara legal. Misalnya, Microsoft Windows memasukkan komponen perangkat lunak open source untuk mengimplementasikan kemampuan internet (yaitu komponen dari University of California at Berkeley, dan kontributornya). Microsoft Office menggunakan pustaka open source zlib.

Walau bagaimanapun, anda harus menelaah lisensinya terlebih dahulu sebelum menggunakan kembali aplikasi orang lain dalam program yang anda buat. Hal ini juga berlaku pada dunia proprietary. Beberapa aplikasi open source menerapkan lisensi seperti MIT, BSD, dan sejenisnya secara eksplisi mengizinkan anda menggunakan aplikasi tersebut dalam sistem anda tanpa biaya harus membayar biaya royalti selama anda mengikuti aturan-aturan sederhana. Misalnya, beberapa mensyaratkan agar anda mencantumkan credit dari pengembang aslinya pada dokumentasi atau pada kode programnya. Ini adalah persyaratan yang sangat murah daripada anda harus menulis sendiri aplikasi tersebut.

Karen Faulds Copenhaver dari Black Duck Software membuat presentasi berjudul "Reviewing Use of OSS in the Enterprise" yang mendiskusikan berbagai mitos, termasuk mitos "anda tidak dapat menjalankan aplikasi open source pada lingkungan proprietary. Hal penting yang perlu diketahui dari sudut pandang pengembang software adalah sama: pahami dan penuhi persyaratan lisensi yang diminta. Ia juga meyakini bahwa pada umumnya, persyaratan yang diminta oleh perangkat lunak open source mudah dipenuhi.

Terkadang masalah lisensi ini menjadi faktor penentu dalam memilih aplikasi yang akan digunakan. Misalnya, ada dua GUI toolkit populer di Linux: Gtk+ yang digunakan dalam pengembangan proyek Gnome dan Qt yang digunakan dalam pengembangan proyek KDE. Gtk+ dirilis dengan lisensi LGPL, sehingga dapat digunakan dalam aplikasi open source ataupun proprietary tanpa harus membayar biaya lisensi. Qt tersedia secara free di bawah lisensi GPL,

dan dengan biaya royalti tertentu untuk lisensi proprietary. Maka jika anda tidak ingin membayar lisensi (atau peduli mengenai potensi biaya di masa datang dan atau bagaimana biaya tersebut dapat memperkuat sebuah perusahaan di masa datang) anda bisa memilih menggunakan pustaka Gtk+.

Sebaliknya, jika anda telah memutuskan untuk melanggar lisensi secara ilegal, jangan membuat asumsi bahwa anda tidak akan tertangkap. Karena kode program open source tersebar luas, seringkali begitu mudah menentukan apakah suatu produk mengandung kode program yang dicuri. Banyak orang benar-benar melakukan analisis mengenai hal ini. Seorang pengembang software dengan cepat menemukan dan membuktikan bahwa CherryOS mencuri kode program PearPC.

Intinya adalah, jika ingin menggunakan kode program milik orang lain, teliti dulu lisensi yang digunakan. Pastikan lisensinya mengizinkan hal tersebut dan penuhi persyaratan lisensi yang diminta. Baik pada program proprietary ataupun program open source.

Jika terjadi penggunaan kode program GPL tanpa sengaja dalam program yang dibuat, apakah keseluruhan program harus diliris sebagai GPL?

Tidak juga, meski anda bisa memutuskan untuk melakukannya. Lisensi GPL, seperti halnya berbagai lisensi pada pustaka proprietary umumnya, memberikan lisensi penggunaan dengan syarat tertentu. Banyak pustaka proprietary mensyaratkan anda membayar biaya per unit, atau sejumlah biaya besar untuk penggunaan tak terbatas. Lisensi GPL tidak membutuhkan biaya. Selain itu juga tidak meminta berbagai batasan-batasan penggunaan seperti pada lisensi proprietary umumnya. Namun lisensi GPL mensyaratkan jika anda memasukkan kode program GPL dalam program yang anda buat, anda harus merilis keseluruhan kode program sebagai GPL.

Namun sebenarnya ada tiga pilihan yang dapat dilakukan. Misalnya anda sedang mengerjakan sebuah proyek proprietary. Lalu seorang pemrogram anda, tanpa sepengetahuan anda, memasukkan kode program GPL ke dalam proyek anda. Jika ini terjadi, pilihannya adalah: (1) merilis seluruhnya sebagai GPL, (2) menghilangkan kode program GPL tersebut, (3) melakukan negosiasi dengan pemilik kode program agar kode program GPL tersebut agar dapat diliris dengan lisensi yang kompatibel. Pilihan terakhir biasanya memerlukan biaya tertentu, dan beberapa proyek open source mungkin tidak akan mengizinkannya sama sekali.

Ada berbagai cara agar aplikasi proprietary dan GPL dapat berjalan bersama, namun lisensinya harus dipenuhi dengan baik. Kernel Linux adalah program GPL, namun aplikasi proprietary dapat berjalan di atasnya. Tidak ada yang menyalahkan hal ini. Kompiler gcc yang dilisensikan GPL juga dapat digunakan untuk meng-compile program proprietary. Jadi, program GPL dapat dijalankan oleh program proprietary selama benar-benar terpisah.

Apakah Perangkat Lunak Open Source Dapat Menjadi Sumber Penghasilan?

Jawabannya adalah ya. Ada berbagai perusahaan yang menarik keuntungan dari perangkat lunak open source, atau menggunakan perangkat lunak open source untuk mendukung aktivitas perusahaan dalam mencari uang. Banyak artikel dan buku yang telah ditulis mengenai hal ini.

Misalnya, pada Januari 2003, HP melaporkan keuntungan dari bisnis GNU/Linux sebesar 2 milyar dolar. IBM juga mencatat pada 2002 bahwa investasi mereka pada Linux sebesar 1 milyar dolar telah kembali dalam satu tahun sebagai laba. IBM kini membukukan keuntungan lebih banyak dari apa yang disebut sebagai penghasilan yang berkaitan dengan Linux daripada penghasilan yang berasal dari penjualan lisensi paten tradisional. Sementara IBM adalah pemegang hak paten terbesar di dunia.

Artikel berjudul Firefox Fortune Hunter pada 2004, mencatat bahwa "bisnis-bisnis baru tumbuh dan berkembang pada penyediaan aplikasi berbasis Mozilla dengan biaya tertentu (tidak gratis). Berbagai organisasi menjadi klien, mulai dari museum hingga perusahaan software dan bahkan Departemen Pertahanan Amerika Serikat. Pete Collins, dari Mozilla Developers Group,

menyatakan, “Bisnis saat ini menggila disebabkan popularitas Firefox. Kami bahkan tidak memasang iklan. Klien sendiri yang menemukan kami dan menawarkan pekerjaan kepada kami”.

Sebagian besar perusahaan yang menghabiskan banyak uang untuk mengembangkan aplikasi open source memang sengaja melakukannya sebagai sebuah strategi bisnis yang baik bagi mereka. Hal ini sesuai dengan konsep mikroekonomi, yaitu bahwa setiap produk memiliki substitusi dan komplemen. Produk substitusi adalah produk lain yang akan anda beli jika produk pertama yang serupa terlalu mahal. Sedangkan produk komplemen adalah sesuatu yang anda beli bersamaan (untuk melengkapi) barang lain yang anda beli.

Permintaan terhadap suatu barang akan naik jika harga barang komplemennya turun. Berdasarkan hukum ini, perusahaan yang pintar akan berusaha membuat komplemen dari produknya menjadi sebuah komoditi. Misalnya sebuah pabrik mobil melakukan investasi agar bensin yang dipakai lebih irit. Sehingga perusahaan tersebut akan berhasil menjual lebih banyak mobil karena mobil yang dibuat lebih irit bensin. Bagi banyak perusahaan, misalnya pembuat perangkat keras komputer dan organisasi penyedia dukungan, memberi dukungan pada produk open source ternyata mampu mengubah produk komplemen menjadi sebuah komoditi. Hasilnya, lebih banyak penjualan (dan tentu saja uang) bagi mereka.

Banyak proyek open source yang pada awalnya memang dimulai dari seorang individu yang mengerjakannya di waktu luang. Masih banyak proyek yang berjalan dengan metode seperti ini. Akan tetapi, proyek yang sudah matang dan banyak digunakan orang, atau menjadi yang paling populer untuk proyek sejenis, cenderung tidak berjalan dengan metode seperti itu. Sebaliknya, sebagian besar proyek open source yang sudah besar atau menjadi dominan memiliki perusahaan yang menjadi penopang di belakangnya. Perusahaan tersebut memiliki dana yang signifikan untuk digunakan dalam proyek tersebut. Hal ini sudah berjalan bertahun-tahun.

Selain itu, kurang tepat jika hanya melihat berbagai perusahaan yang mengeruk keuntungan dari perangkat lunak open source. Ada sisi lain yang perlu diperhatikan, yaitu sisi permintaan atau sisi pengguna alias konsumen. Para pengguna dapat menyisihkan dana mereka untuk hal lain ditambah dengan berbagai keuntungan lainnya jika menggunakan perangkat lunak open source. Dengan demikian, ada alasan ekonomi yang kuat untuk melakukannya. Setiap orang yang mampu menyisihkan dana akan berusaha keras untuk terus dapat menyimpan dana tersebut. Seringkali jauh lebih murah bagi para konsumen untuk bergabung bersama membayar pengembangan kecil dari sebuah produk open source.

Sebuah vendor proprietary dapat menemui masalah jika bersaing dengan produk open source yang serupa. Penyebabnya adalah produk open source biasanya lebih murah dan membebaskan pengguna dari kendali vendor. Beberapa vendor open source memang gagal secara finansial. Namun, hal yang sama terjadi juga pada vendor proprietary. Jadi, hal ini sama sekali tidak menunjukkan bahwa open source tidak akan berhasil.

Pada dasarnya, software sangat berbeda dibandingkan dengan benda fisik bila dipandang secara ekonomis. Software dapat diperbanyak hingga jumlah yang tak terbatas dan hampir tidak ada biaya untuk menggandakannya. Tinggal disalin saja. Selain itu, software juga dapat dikembangkan oleh ribuan pemrogram yang bekerja bersama-sama dengan biaya investasi yang kecil. Biaya untuk menggerakkan tiap pemrogram dapat ditekan menjadi sangat kecil. Software juga tahan lama, dalam arti bahwa software (secara teori) dapat digunakan selamanya. Software yang sama juga dapat digunakan oleh dua orang secara bersamaan tanpa saling mengganggu. Masing-masing dapat menjalankan salinan software yang sama di komputer masing-masing. Hal ini tidak berlaku pada benda fisik seperti perangkat keras komputer. Dengan demikian, biaya marjinal untuk mendistribusikan sebuah paket software hampir mendekati nol. Hal ini mampu menjelaskan mengapa Microsoft menjadi kaya dalam sekejap (karena menjual sesuatu yang biayanya mendekati nol untuk digandakan). Hal ini pula yang membuat banyak pengembang software open source yang rela mendistribusikan software secara cuma-cuma.

Apakah Open Source Akan Menghancurkan Industri Software?

Tidakkah para pemrogram akan kelaparan jika banyak aplikasi menjadi open source? Tidak. Saat

ini, semakin banyak aplikasi open source yang dikembangkan dan didukung secara komersial. Ada bukti lain yang menarik bahwa 95% perangkat lunak tidak dikembangkan untuk dijual. Sebagian besar software dikembangkan oleh pengembang software yang dibayar oleh suatu organisasi untuk memenuhi kebutuhannya. Dengan demikian, meluasnya penggunaan open source tidak akan membahayakan sumber penghidupan para pemrogram. Karena jika akhirnya semua orang menggunakan software yang free (tidak membeli atau membajak software), hanya akan menghapus 5% dari seluruh pekerjaan pemrogram software.

Perangkat lunak open source tidak mensyaratkan pengembangnya bekerja tanpa dibayar. Banyak produk open source yang dibangun atau dikembangkan kemampuannya oleh orang yang memang dipekerjakan untuk hal tersebut atau dikontrak untuk menambahkan fitur tertentu. Jika suatu organisasi ingin menambahkan kemampuan tertentu pada sebuah produk open source, ia harus mencari seseorang untuk melakukannya. Bisa jadi orang internal organisasi tersebut, orang yang terlibat dengan pengembangan produk tersebut, atau pihak ketiga. Artinya, ia harus membayar orang yang akan melakukan hal tersebut. Model pembayaran seperti ini lebih masuk akal. Sebab, yang dibayar adalah penambahan fitur spesifik tersebut. Sebagaimana telah dibahas sebelumnya, biaya yang sebenarnya pada software adalah biaya pengembangan, bukan biaya pengandaannya.

Ada beberapa cara untuk menghubungkan pihak yang membutuhkan modifikasi tertentu dengan pihak yang memahami bagaimana melakukan modifikasi tersebut. Salah satunya adalah bounty system, atau biasa disebut juga sponsor system atau pledge system. Caranya adalah: seorang pengguna meminta pengembangan atau penambahan fitur tertentu dan menyatakan jumlah yang akan dibayarnya untuk fitur tersebut. Biasanya, sistem ini mengizinkan orang lain untuk bergabung. Tujuannya adalah mengumpulkan jumlah yang mencukupi untuk membuat pengembang tertarik memenuhi permintaan pengembangan atau penambahan fitur tersebut. Sistem ini bisa dijalankan oleh perorangan atau pihak ketiga. Banyak juga proyek open source yang memiliki sistem bounty sendiri. Misalnya Mozilla, GNOME, dan lainnya.

Sebaliknya, siapa saja bisa mengusulkan untuk menambahkan fitur tertentu jika orang-orang mau membayar sejumlah uang. Fundable.org menyediakan fasilitas untuk metode ini. Fasilitas ini pernah digunakan untuk mengumpulkan dana secara cepat sehingga Frederico Caldeira Knabben dapat melakukan porting html editornya ke browser Safari milik Apple Macintosh.

Pendekatan yang adalah software ransom. Pendekatan ini biasanya digunakan untuk mengubah software proprietary menjadi open source. Para pengguna mengumpulkan uang mereka untuk membayar pemilik software agar dirilis sebagai open source. Contoh untuk pendekatan ini adalah Blender.org. Ada banyak contoh lainnya. Misalnya Nokia membayar Matthew Allum untuk secepatnya mengembangkan Matchbox Window Manager agar mendukung Internet Tablet Nokia N770).

Halaman Jobs in Free Software pada situs Free Software Foundation (FSF) adalah salah satu tempat pertemuan antara perusahaan sebagai pengguna dan para pengembang yang tertarik mengembangkan proyek open source tertentu. Tentu ini bukan satu-satunya tempat. Robert Westervelt melaporkan di situs SearchVB (situs khusus tentang Microsoft Visual Basic) bahwa pekerjaan di bidang sekuriti, web services, dan Linux terus mendominasi iklan lowongan TI pada 2004.

Pengembangan piranti lunak open source memungkinkan developer pemula untuk mendapatkan dan meningkatkan pengalamannya. Sekaligus memungkinkan berbagai organisasi menemukan developer yang mereka butuhkan (dan kemudian membayar untuk menggarap lebih banyak piranti lunak). Seringkali berbagai organisasi menemukan developer yang mereka butuhkan dengan melihat proyek open source yang penting bagi mereka. Maka pemimpin suatu proyek open source memiliki kemungkinan lebih besar untuk diminta mengerjakan pembuatan extension bagi piranti lunak tersebut atau menyediakan dukungan. Hal ini memupuk harapan bagi developer pemula. Jika mereka memulai proyek baru atau berkontribusi nyata pada suatu proyek akan meningkatkan kemungkinan diminta mengerjakan modifikasi bagi kepentingan organisasi tertentu. Ada contoh nyata dalam teori ini. Salah seorang presenter Netscape pada FOSDEM (Free and Open Source Software Developers European Meeting) 2002 adalah kontributor proyek

Mozilla milik Netscape. Kontribusi beliau membuat Netscape menawarkan pekerjaan yang ternyata beliau terima.

Tentu saja open source mempengaruhi industri software. Namun pengaruhnya adalah positif. Pengguna atau pelanggan adalah pihak yang membiayai secara langsung pengembangan spesifik yang mereka inginkan. Sehingga pasar akan mendorong para developer open source mengerjakan pengembangan sesuai keinginan para pengguna/pelanggan. Vendor proprietary berusaha mengidentifikasi kebutuhan pelanggannya melalui departemen pemasaran. Namun metode tentu kurang efektif dibandingkan dengan pelanggan itu sendiri yang mengidentifikasi. Pada pengembangan secara open source, para pengguna atau pelanggan menunjukkan secara langsung fungsionalitas apa yang mereka anggap paling penting. Yaitu dengan cara memutuskan mana yang akan mereka biayai pengembangannya.

Hal lain yang terlihat kontras adalah motivasi pendanaan produk proprietary terkadang tidak sejalan dengan motivasi penggunaannya. Pengembangan secara proprietary menyediakan dana yang cukup besar untuk mencegah penggunaan dan interoperasi produk serupa (pesaing), serta mencegah penggandaan (kecuali diizinkan secara spesifik oleh vendor). Sehingga ketika suatu produk proprietary digunakan secara luas, vendor yang bersangkutan seringkali lebih memperhatikan pencegahan penggunaan dan interoperasi serta penggandaan. Bukan berfokus pengembangan fungsionalitas yang diinginkan oleh para pengguna. Mereka kadang tetap melakukannya meski tidak sejalan dengan kebutuhan pengguna.

Kecenderungan ini terlihat jelas selama beberapa dekade industri software. Protokol dan format data yang terus berubah tanpa adanya dokumentasi, media yang mengandung proteksi penggandaan, dan mekanisme registrasi adalah contoh dari perbedaan motivasi tersebut. Pengembang aplikasi open source tidak akan kehilangan apa-apa jika penggunaannya beralih ke aplikasi lain, baik proprietary atau open source. Sebab, mereka juga tidak menginvestasikan sejumlah dana untuk kepentingan semacam itu. Banyak perusahaan open source telah didirikan. Ada yang sukses dan ada yang gagal. Mendirikan perusahaan software memang bukan pekerjaan mudah dan perlu kerja keras. Saat ini, orang menggunakan dan menggabungkan aplikasi open source untuk menyelesaikan masalah yang spesifik dan menjualnya (dengan dukungan) dengan nilai keuntungan yang besar.

Karen Shaeffer telah membuat tulisan menarik berjudul *Prospering in the Open Source Software Era*, yang mendiskusikan efek piranti lunak open source. Misalnya, efek disruptif open source yang mengubah properti proprietary menjadi suatu komoditi dan mendorong inovasi. (berlawanan dengan proprietary yang cenderung mencegah inovasi. Menurutnya, pemenang utamanya adalah pengguna akhir dan developer software. Sebab, "nilai suatu software tidak lagi terletak pada kode program dasar, nilainya ada pada developer yang bisa dengan cepat beradaptasi dan memperluas aplikasi open source yang ada untuk memungkinkan berjalannya bisnis, serta mengetahui tujuan mereka berkaitan dengan peluang yang muncul. Komoditisasi kode program ini menunjukkan lompatan quantum jauh ke depan dalam efisiensi proses bisnis – membawa developer dengan keahlian langsung ke kelompok bisnis yang memiliki gagasan inovasi."

Apakah Open Source Kompatibel dengan Kapitalisme?

Ya. Beberapa tahun yang lalu ada sebagian orang yang berusaha memberikan stempel "komunis" atau "sosialis" pada open source. Namun demikian, mereka telah gagal dengan retorika tersebut. Konsep open source cukup konsisten dengan kapitalisme: meningkatkan kekayaan tanpa mengganggu prinsip-prinsip hak milik pribadi atau kebebasan berkehendak atau berkarya.

Jika dalam suatu kategori software hanya tersedia produk open source, akankah hal ini menghilangkan kompetisi?

Tidak. Cukup aneh bahwa aplikasi open source seringkali bersaing satu sama lain dalam kategori fungsionalitas yang sama. Misalnya editor teks emacs dan vi telah berduel selama beberapa dekade. Sendmail masih menjadi yang paling populer untuk kebutuhan pengiriman email. Namun

Sendmail senantiasa bersaing dengan Postfix dan Exim. Lingkungan dekstop KDE dan GNOME pun terus berlomba dengan fitur-fitur terbaru masing-masing. Demikian juga dengan kernel Linux dan BSD. Secara umum, proyek open source yang bersaing harus memiliki perbedaan khusus dengan yang lain bila ingin meraih kesuksesan. Perbedaan yang mendasar tersebut dapat berupa filosofi user interface, pendekatan yang digunakan dalam desain, karakteristik sekuriti, dan strategi lisensi yang digunakan. Hal ini juga berlaku pada aplikasi proprietary yang saling bersaing. Namun demikian aplikasi open source yang bersaing selalu berusaha agar kompatibel satu sama lain. Sebab, para pengguna menginginkan hal itu. Terkadang mereka malah saling membantu dalam urusan teknis.

Misalnya, freedesktop.org menyediakan sebuah forum untuk mendorong kerja sama antar berbagai dekstop untuk X Window System (seperti halnya KDE dan GNOME). Ia juga merupakan bagian dari Free Standards Group yang berusaha mempercepat penggunaan dan penerimaan teknologi open source. Usaha tersebut dilakukan melalui pengembangan, penerapan, dan promosi standarisasi teknologi open source. Bahkan, jika hanya ada satu produk, berbagai pihak dapat bersaing dalam menyediakan perawatan dan dukungan. Misalnya persaingan antar distribusi GNU/Linux. Dengan demikian, meski perangkat lunak open source menghilangkan aplikasi proprietary dalam kategori tertentu, kompetisi tidak akan ikut menghilang.

Apakah Program Open Source Memiliki Standarisasi? Ya.

Aplikasi open source bisa menerapkan standarisasi sebagaimana aplikasi proprietary. Bahkan seringkali aplikasi open source dapat menerapkan standarisasi dengan lebih baik. Alasannya cukup sederhana: proyek open source tidak mendapat keuntungan finansial jika tidak mengikuti standarisasi yang ada. Tugas pembuat software proprietary adalah mencapai keuntungan sebanyak-banyaknya. Untuk mencapai tujuannya, pembuat software proprietary bisa saja tidak menghiraukan standarisasi yang ada atau membuat extension proprietary dari standar yang ada; ketika pelanggan sudah bergantung pada antarmuka proprietary tersebut, mereka akan kesulitan berpindah ke produk yang lain, meski produk lain lebih baik. Sebaliknya, proyek open source umumnya didukung langsung oleh para penggunanya. Para pengguna menginginkan adanya standarisasi untuk menjaga kesinambungan akses ke data-data mereka, menyederhanakan kerjasama dengan aplikasi lain, dan menyederhanakan integrasi dengan lingkungan milik mereka.

Pada beberapa kasus, proyek open source seringkali pada akhirnya membuat spesifikasi atau standar dalam bentuk executable. Standar tradisional (tertulis dalam bentuk paper/dokumentasi) tidak dapat langsung digunakan oleh pengguna, dan selalu mengandung hal-hal yang dapat ditafsirkan berbeda serta sulit diselesaikan saat diterapkan. Sebaliknya, program open source dapat digunakan secara langsung sehingga menolong pengguna secara langsung. Karena diterapkan secara transparan, mereka dapat memperjelas hal-hal yang ambigu dalam dokumentasi standarisasinya.

Seiring pertumbuhan open source software, telah dibentuk beberapa lembaga yang bertujuan mengembangkan standarisasi untuk mendukung *interoperability*. Lembaga-lembaga yang terbentuk antara lain *Free Standards Group*, *Free Desktop.org*, *Linux Standard Base*, *Filesystem Hierarchy Standard*, dan *X.org*. Ada juga interaksi dengan kelompok pembuat standar seperti IETF dan W3C (World Wide Web Consortium).

Apakah Open Source akan Menghancurkan Hak atas Kekayaan Intelektual?

Tidak. Mari kita simak salah satu jenis lisensi open source yang paling banyak dipakai, yaitu GPL. Berikut ini penjelasan dari lisensi GPL yang disederhanakan:

Piranti lunak ini mengandung kekayaan intelektual dari beberapa orang. Kekayaan intelektual adalah sesuatu yang bernilai, dan anda tidak bisa menggunakan kekayaan intelektual orang lain dalam pekerjaan anda secara gratis. Banyak bisnis dan individu yang bersedia menukar kekayaan intelektual mereka dengan sesuatu yang bernilai, misalnya uang. Contohnya, dengan sejumlah uang, anda bisa memperoleh hak untuk menggabungkan/memasukkan kode program

orang lain ke dalam program yang anda buat.

Para developer software ini bermaksud menukar hak penggunaan kekayaan intelektual mereka pada anda dengan sesuatu yang bernilai. Namun, bukan meminta uang, mereka meminta anda bersedia menukarnya dengan hak untuk memasukkan kode program mereka dalam program yang anda buat. Sebagai gantinya, anda diminta membebaskan kode program yang anda buat untuk dapat dimasukkan dan digabungkan dengan kode program mereka. Proses pertukaran kode ini dilakukan dibawah lisensi GPL. Jika anda tidak setuju dengan hal ini, anda masih bebas menggunakan program ini. Anda bisa membuat program anda sendiri dari awal atau membeli dari pihak lain.

Microsoft menyampaikan keberatan karena GPL tidak mengizinkan mereka mengambil kode program tertentu dan membuat perubahan yang akan mereka rilis sebagai proprietary. Ini adalah sesuatu yang hipokrit. Sebab, Microsoft sendiri biasanya tidak mengizinkan orang lain untuk membuat dan mendistribusikan modifikasi dari software mereka sama sekali. Jadi, GPL memberikan hak yang jauh lebih banyak kepada para pengguna daripada Microsoft.

Pada beberapa kasus, Microsoft akan merilis kode program di bawah lisensi "shared source". Tetapi lisensi ini jauh lebih ketat bila dibandingkan dengan GPL. Misalnya, lisensi ini melarang penyebaran software dalam bentuk *source* atau *object* untuk tujuan komersial dalam keadaan apapun.

GPL terkadang juga disebut sebagai "virus" oleh vendor proprietary, khususnya Microsoft. Sebabnya tak lain adalah GPL menyebarkan pandangan dan mendorong pihak lain untuk menerapkan lisensi GPL juga pada produk atau hasil kerja masing-masing. Namun, jika ingin dianggap "fair", perlu diingat bahwa berbagai lisensi dan produk proprietary juga memiliki efek seperti virus. Banyak produk proprietary dengan protokol dan format data tertutup yang memiliki "network effect". Yaitu bila banyak orang menggunakan produk tersebut, kelompok pengguna ini membuat orang yang tidak menggunakan produk tersebut berada pada situasi yang kurang menguntungkan.

Sebagai contoh, ketika beberapa orang menggunakan produk proprietary seperti sistem operasi atau pengolah kata, semakin sulit bagi pihak lain untuk menggunakan produk yang berbeda. Seiring berjalannya waktu, hal ini dapat mengarah pada penyebaran produk proprietary yang menyebar seperti virus.

Barangkali yang dimaksud menghancurkan kekayaan intelektual oleh Microsoft adalah kemungkinan "gulung tikar"-nya produk milik pesaing GPL. Jika demikian, maka hal ini adalah sesuatu yang hipokrit. Microsoft sudah membuat banyak bisnis pesaingnya gulung tikar atau membeli mereka dengan harga jauh di bawah harga asli mereka. Bahkan kadang-kadang teknik yang digunakan Microsoft terbukti ilegal ketika dibawa ke pengadilan. Sebaliknya ada bukti yang sangat kuat bahwa GPL adalah sesuatu yang sangat legal di mata hukum.

GPL tidak menghancurkan kekayaan intelektual. Melainkan hanya menciptakan aturan main yang memungkinkan setiap orang dapat berkontribusi secara sukarela dalam pengembangan suatu proyek dengan meminimalkan kemungkinan dicurinya kontribusi tersebut oleh pihak lain. Hal ini dapat dianalogikan dengan membuat konsorsium, namun tidak ada yang diminta mendanai konsorsium tersebut. Tetapi bagi yang bergabung dalam konsorsium harus mengikuti aturan yang ada. Dapat dipahami bahwa Microsoft ingin mengambil hasil dari konsorsium tersebut dan memiliki sendiri turunan atau hasil modifikasinya. Namun tidak ada alasan untuk mempercayai bahwa dunia tanpa GPL adalah dunia yang paling diinginkan oleh para pengguna atau pelanggan.

Saat ini Microsoft secara aktif mengembangkan dan merawat beberapa proyek open source, meliputi antara lain Windows Installer XML (WiX), Windows Template Library (WTL), dan FlexWiki. Jason Matusow, direktur program Microsoft Shared Source menyatakan bahwa perusahaannya, "akan memperluas program open source seiring berjalannya waktu, namun akan bergerak perlahan-lahan sambil mencoba mempelajari bagaimana berpartisipasi dalam

komunitas open source.

Apakah memiliki akses ke source code adalah sesuatu yang penting?

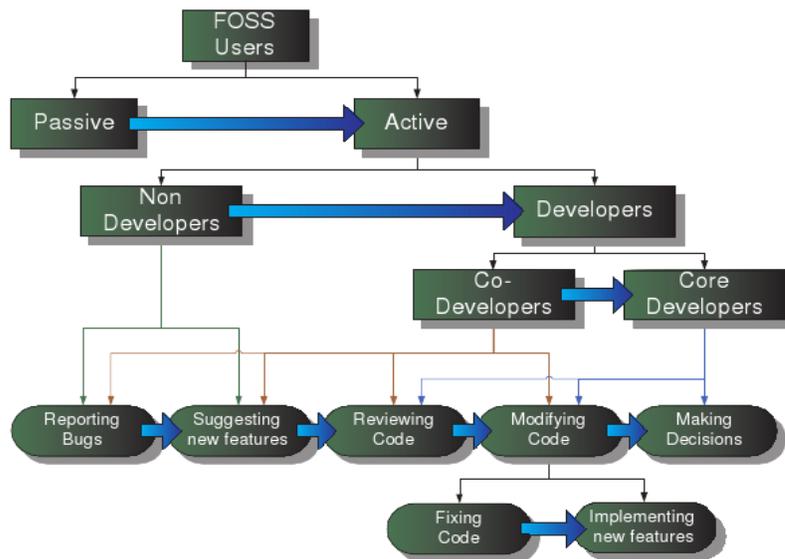
Sementara itu, ada anggapan bahwa akses ke source code tidak memberikan keuntungan signifikan. Sebab, orang yang dapat memanfaatkan akses ke source code hanya sedikit. Apakah memiliki akses ke source code adalah sesuatu yang penting? Jawabannya adalah Ya. Memang benar bahwa hanya sedikit orang yang memerlukan akses langsung; hanya pengembang atau peneliti source code yang dapat melakukannya. Namun, tidak memiliki akses pada bagaimana komputer bekerja adalah hal yang tidak baik. Bob Young dari Red Hat memiliki analogi yang paling baik untuk hal ini:

Open source gives the user the benefit of control over the technology the user is investing in... The best analogy that illustrates this benefit is with the way we buy cars. Just ask the question, "Would you buy a car with the hood welded shut?" and we all answer an emphatic "No." So ask the follow-up question, "What do you know about modern internal-combustion engines?" and the answer for most of us is, "Not much."

Hampir tidak ada orang yang akan mau membeli mobil yang mesinnya tidak boleh dibuka. Meskipun pembeli tidak mengetahui mengenai teknologi terbaru dalam mesin mobil. Pengguna memerlukan akses ke source code, agar memiliki kendali pada produk yang dibeli agar tidak tergantung pada vendor.

Pembeli dapat saja membawa mobilnya kembali ke dealer bila ada masalah. Namun bila dealer tersebut tidak meminta biaya yang terlalu tinggi, melakukan pekerjaannya dengan baik, dan menambahkan fitur yang diinginkan, maka pembeli dapat terus kembali ke dealer tersebut. Akan tetapi, bila dealer tersebut memintabiaya yang terlalu tinggi, tidak menyelesaikan masalah kita, dan tidak mau menambahkan klakson dengan musik, maka masih ada ribuan perusahaan lain yang akan dengan senang hati mengerjakan bisnis kita.

Pada bisnis software proprietary, pelanggan tidak memiliki kendali pada teknologi yang digunakan untuk membangun bisnisnya. Jika vendor meminta biaya yang terlalu tinggi, tidak memperbaiki bug yang menyebabkan sistemnya menjadi crash, atau memilih untuk tidak menerapkan fitur yang dibutuhkan, pelanggan tidak punya pilihan lain. Ketiadaan kendali ini dapat menyebabkan biaya yang tinggi, reliabilitas yang rendah, dan banyak frustrasi.



Evolusi Pengguna Non Teknik dalam Pengembangan Perangkat Lunak Open Source

Skema pengembangan pada platform open source di atas memungkinkan setiap orang berpartisipasi. Selain itu, ketersediaan aplikasi di masa datang juga terjawab oleh survei Evans Data pada Oktober 2002. survei ini melaporkan bahwa Linux terus memperluas basis

penggunanya, dan 59% dari responden merencanakan untuk membuat aplikasi berbasis Linux pada tahun berikutnya. Pada November 2001, Evans Data International Developer Survey Series melaporkan wawancara in-depth dengan 444 pengembang aplikasi dari 70 negara, 48,1 % menjawab akan mengembangkan aplikasi mereka agar berjalan pada Linux mulai tahun depan. Setengah dari responden menyatakan memiliki kepercayaan yang cukup untuk menjalankan GNU/Linux pada aplikasi kritis. Sedangkan survei yang dilakukan Saugatuck Research pada Januari 2007 menyatakan bahwa setengah dari seluruh perusahaan akan menggunakan Linux dalam waktu 5 tahun.

Apakah Open Source Hanya Sebuah Kampanye Anti Microsoft?

Tidak. Memang ada pendukung open source yang juga anti Microsoft, tetapi pandangan bahwa open source hanya anti Microsoft adalah tidak benar.

Microsoft sendiri sudah lama bergantung pada aplikasi open source dalam software yang dijualnya. Implementasi protokol dasar internet pada sistem Windows (TCP/IP) diturunkan dari kode program open source. Aplikasi Microsoft Office juga menggunakan pustaka kompresi open source "zlib". Microsoft juga mengakui bahwa infrastruktur penting dari Windows Server 2003 Compute Cluster Edition adalah aplikasi open source. Sebab, jika tidak, akan sulit bagi Microsoft untuk bersaing.

Microsoft pun bisa saja, kapan saja, merilis program (misal sistem operasi) sebagai open source, mengambil sistem operasi open source yang sudah ada dan merilis versi Microsoft, atau menyediakan aplikasi untuk sistem open source. Tidak ada lisensi yang melarang hal ini. Namun sepertinya hal ini tidak akan terjadi dalam waktu dekat ini. Mengapa? Tentu saja karena hal tersebut bertentangan dengan praktik bisnisnya saat ini. Para pemimpin open source juga sering menyatakan bahwa mereka tidak anti Microsoft *per se*. Mereka hanya tidak setuju dengan sebagian praktik bisnis Microsoft saat ini. Banyak juga yang terus mengajak Microsoft untuk bergabung dalam komunitas open source.

Data SourceForge.net menunjukkan banyak kasus pengembangan aplikasi open source dengan dan untuk teknologi Microsoft. Pada Juni 2002, terdaftar 831 proyek open source yang menggunakan Visual Basic (teknologi proprietary dari Microsoft) dan 241 yang lain menggunakan C# (bahasa pemrograman yang berasal dari Microsoft, sudah distandarisasi oleh ECMA). Ada pula 8867 proyek yang ditujukan untuk berjalan di atas Windows. Bukti bahwa banyak juga yang tidak anti Microsoft.

Anehnya, Microsoft meski pernah menyatakan bertentangan dengan GPL, namun pada saat yang sama juga menjual produk dengan komponen berlisensi GPL. Tidak ada yang salah dalam hal ini. Microsoft tetap mengikuti aturan hukum yang berlaku dalam lisensi komponen tersebut. Microsoft Windows Services for Unix yang dijualnya berisi Interix, sebuah lingkungan yang memungkinkan untuk menjalankan aplikasi dan skrip Unix di atas Windows NT dan Windows 2000. Banyak komponen Interix merupakan aplikasi GPL, seperti gcc dan g++ (kompiler untuk C dan C++). Microsoft memutuskan untuk menjual suatu cara untuk menjalankan aplikasi Unix pada produk sistem operasinya. Sebab, ada banyak aplikasi Unix di luar sana.

Masalahnya adalah, Microsoft menyatakan sebaiknya tidak ada orang yang menggunakan GPL, dan bahwa tidak ada orang yang dapat menghasilkan uang dengan GPL, sambil terus-menerus menghasilkan uang dengan GPL. Kenyataan ini membuktikan bahwa Microsoft "bisa" menggunakan aplikasi GPL tanpa kehilangan kekayaan intelektualnya.

Microsoft pun mulai berperan lebih banyak di dunia open source. Bill Hilf menjelaskan alasannya, "*Do I really care if it's open source or not if it sells our infrastructure?*" Microsoft juga lebih aktif mendukung proyek-proyek open source. Meski pada kasus tertentu produk yang dihasilkan proyek tersebut bersaing dengan produk Microsoft. Apa penyebabnya? Tidak lain karena penggunaan produk tersebut mendorong penjualan produk Microsoft yang lain. Jadi, Microsoft melakukan hal tersebut juga bukan agar terlihat "bersahabat" dengan komunitas open source. Namun Microsoft melakukannya dengan tetap sejalan dengan bisnisnya, yaitu untuk

memperbesar bisnisnya. Proyek yang tercakup antara lain WiX, IronPython, dan seluruh situs CodePlex yang disponsori oleh Microsoft untuk mengembangkan program-program open source.

Kompetitor selalu ada di bidang bisnis apa saja. Umumnya, pebisnis yang baik tidak akan menghancurkan bisnis kompetitornya, melainkan menjaganya selalu ada. Sebagaimana ditulis oleh Sun Tzu dalam "The Art of War", peliharalah musuhmu, karena mereka adalah orang yang paling jujur mengenai apa yang harus kamu perbaiki. Jika tidak ada pesaing, pebisnis mungkin akan puas dengan pencapaiannya. Namun jika ada pesaing, ia terpacu untuk terus menghasilkan yang lebih baik daripada pesaingnya.