

Hacking Aplikasi dan Sistem Linux¹

Iwan Setiawan <stwn@jogja.linux.or.id>

Kelompok Pengguna Linux Indonesia (KPLI) Jogjakarta

<http://jogja.linux.or.id>

Daftar Isi

Hacking Aplikasi dan Sistem Linux.....	1
1. Hacker dan Cracker.....	1
2. Software Hacking.....	2
3. Linux/PLBOS sebagai Produk Hacking.....	2
4. Model Pengembangan Linux/PLBOS.....	3
5. Perkakas Pengembangan di Sistem Linux.....	3
5.1. KDevelop.....	4
5.2. Motor.....	4
6. Langkah-Langkah Mengembangkan PLBOS.....	5
7. Membangun Distribusi Linux.....	6
7.1. Berbasis Debian GNU/Linux dan KNOPPIX.....	7
7.1.1. Membuat Paket Debian.....	8
7.1.2. KuliAx.....	9
8. Daftar Pustaka yang Patut Dibaca.....	10

1. Hacker dan Cracker

Hacker menurut Eric Steven Raymond (ESR) dalam “*How To Become A Hacker*” yang mengacu pada *Jargon File* didefinisikan sebagai orang yang pandai dalam hal teknis, menyenangkan pemecahan masalah, dan dapat melebihi/mengatasi batasan. Sebaliknya Cracker disebut dengan orang yang membobol sistem komputer atau telepon (*phreaking*). Ada pula yang menambahkan kategori Abu-Abu atau *Grey* untuk orang-orang yang melakukan atau mempelajari *Cracking* dengan tujuan agar mereka mengetahui bagaimana seorang Cracker

¹ Dipresentasikan pada “Seminar & Demo: Hacking dan Networking with Linux” yang diselenggarakan oleh ELiT dan Keluarga Mahasiswa Teknik Fisika (KMTF) UGM, 11 November 2006, Jogja Expo Center (JEC), Jogjakarta

membobol sebuah sistem komputer atau memecahkan proteksi dalam perangkat lunak, baik itu untuk memproteksi sistem miliknya, hanya sebagai studi, keinginan mencari tahu, atau yang lainnya.

Banyak orang sekarang sudah mengistilahkan Hacker dengan seseorang yang jahat, suka merusak, dan lain-lain. Sebenarnya tidak, Hacker adalah pembangun sedangkan Cracker adalah perusak. Seperti yang disebut oleh ESR dalam tulisannya yang sama:

“The basic difference is this: hackers build things, crackers break them”

2. Software Hacking

Hacking merupakan kegiatan yang dilakukan oleh Hacker, dalam istilah gaulnya adalah “ngoprek” atau “utak-utik”. Apa saja yang kita lakukan untuk membangun, melewati batasan-batasan sistem, memberikan solusi pada permasalahan bisa disebut Hacking. Software Hacking adalah salah satu aktivitas yang berhubungan dengan utak-atik perangkat lunak baik itu mengembangkan aplikasi ataupun sistem. Istilah Hacking pun menjadi melekat pada Software Hacking karena kebanyakan orang banyak menggunakannya pada bidang ini.

3. Linux/PLBOS sebagai Produk Hacking

Linux sebagai kernel dan sebagai “sebutan singkat” dari sistem dengan aplikasi-aplikasi plus kernel Linux merupakan sebuah produk *hacking*. Bagaimana tidak, Linus Torvalds pada awalnya hanya *mengoprek* sistem operasi atau kernel pada saat belajar bahasa pemrograman bahasa C dan mempelajari buku “*Operating Systems: Design and Implementation*” karya Andrew S Tanenbaum yang juga disertakan didalamnya MINIX², kemudian menjadi *oprekan* bagi para *hacker* di dunia.

Dari tahun 1991, ketika Linus merilis kode sumber dari kernel Linux versi 0.01, sekarang hanya sekitar 2% saja yang merupakan hasil *hacking*-nya. Artinya? sekitar 98% dikembangkan bersama-sama oleh dunia dengan ribuan orang yang berkolaborasi lewat Internet untuk menghasilkan sistem operasi tangguh yang dapat kita nikmati sekarang.

Free and Open Source Software atau dalam Bahasa Indonesia menjadi Perangkat Lunak Bebas dan Open Source (PLBOS) adalah perangkat lunak yang dibuat dan

² Sebuah sistem operasi pembelajaran di lingkungan universitas, sekarang telah sampai pada versi 3 merupakan sebuah mikro kernel, <http://minix3.org>

dikembangkan oleh satu atau kelompok orang kemudian dirilis ke publik untuk dapat dipakai, dipelajari, dimodifikasi, serta didistribusikan secara **bebas**. Kernel Linux adalah salah satu dari PLBOS yang dirilis di bawah GNU General Public License (GPL), salah satu dari lisensi yang ada di dunia PLBOS dan paling banyak digunakan.

4. Model Pengembangan Linux/PLBOS

Model pengembangan Linux dan PLBOS dikenal dengan model terdistribusi (*distributed*) karena *hacker-hacker* yang terlibat tersebar di seluruh penjuru dunia. Umumnya, terdapat pemimpin proyek pada masing-masing proyek PLBOS tersebut, terkadang memiliki sebutan sesuai dengan tingkatannya jenderal, letnan, dan seterusnya, walaupun tidak formal.

Selain itu jika kita perhatikan lagi ada model struktur organisasi seperti pada sebuah lembaga atau perusahaan profesional terutama pada proyek besar seperti Debian, Fedora, Mozilla, termasuk yayasan yang didirikan untuk mendukung pengembangannya.

Dengan model pengembangan yang terdistribusi, hubungan saling percaya, kolaborasi kuat menjadikan PLBOS menjadi model yang sekarang banyak diadopsi oleh banyak orang atau perusahaan untuk menghasilkan perangkat lunak fungsional dan berkualitas tinggi.

5. Perkakas Pengembangan di Sistem Linux

Banyak utilitas atau perkakas yang telah tersedia untuk dapat dijalankan di atas sistem Linux, sebagian besar dirilis dibawah lisensi PLBOS jadi kita dapat dengan bebas menggunakan, menyalin, mempelajari, memodifikasi jika perlu, dan mendistribusikannya. Beberapa dari perkakas-perkakas tersebut adalah:

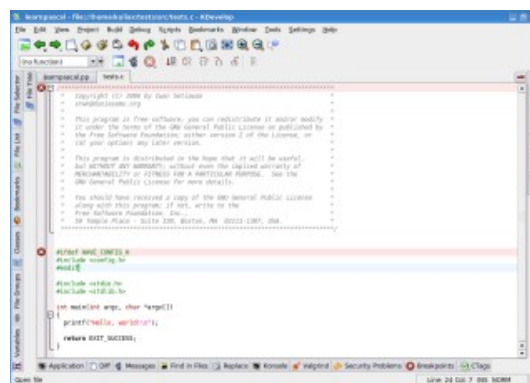
- C, C++: GNU Compiler Collection (GCC), Anjuta (IDE)
- Java: GNU Compiler Java (GCJ), Sun Java (*closed-source/proprietary*)
- Python: Python, Eric (IDE), IDLE (IDE), Boa-Constructor (IDE)
- Pascal: Freepascal, termasuk IDE ada di dalamnya. Mirip Turbo Pascal.
- Basic: Gambas, Real Basic (*closed-source/proprietary*)
- Perl: Perl dan modulnya
- Project Scheduler: Ganttproject, kplato, planner

- ➔ Integrated Development Environment (IDE): Eclipse, KDevelop, Motor, Quanta+
- ➔ Unified Modeling Language (UML): ArgoUML, BoUML, Umbrello, gaphor
- ➔ Documentation: doxygen
- ➔ Tools: cscope, kscope, quilt, valgrind, dan lain-lain

Berikut akan sedikit dibahas mengenai dua (2) IDE di sistem Linux yaitu KDevelop dan Motor.

5.1. KDevelop

KDevelop merupakan salah satu IDE untuk banyak bahasa pemrograman, lebih khususnya C++ di lingkungan KDE (Qt). Dukungan bahasa pemrograman KDevelop cukup banyak yaitu C, C++, Fortran, Haskell, Java, PHP, Pascal, Perl, Python, Ruby, juga skrip Shell. Selain itu pada KDevelop disediakan semacam *plugin* di dalamnya untuk *tool* lain seperti *valgrind*, *ctags*, dan lain-lain.



Gambar 5.1. Tampilan KDevelop

5.2. Motor

Motor adalah IDE *powerfull* untuk beragam bahasa pemrograman, berbeda dengan KDevelop yang menggunakan GUI, Motor menggunakan mode teks, karena itu lebih ringan. Template yang disediakan secara default adalah library static, C, C++, Java, dan Perl.

```
#!/bin/sh
include: cpplib.h
int main(int argc, char **argv) {
    const char *name = "sh";
    const char *author = "John DeLaney";
    printf("program by %s (%s), name: %s\n",
        argv[0], argv[1], name, author);
    return 0;
}
Project shu has been loaded
```

Gambar 5.2. Tampilan Motor

6. Langkah-Langkah Mengembangkan PLBOS

Banyak orang yang awam masih merasa bingung bagaimana kita mau mengembangkan atau memodifikasi aplikasi yang sudah dibuat oleh orang dan dirilis sebagai PLBOS? Umumnya para *hacker* menyimpannya pada portal atau situs komunitas yang menyediakan penyimpanan perangkat lunak, selain itu ada pula *review* dan *rating* dari aplikasi yang disimpan di sana. Beberapa contohnya adalah *sourceforge.net*, *freshmeat.net*, *savannah.gnu.org*.

Jika kita rangkum langkah-langkah mendapatkan aplikasi dari kode sumbernya yang selanjutnya bisa kita modifikasi seperti di bawah ini:

- ➔ Cari informasi lewat mesin-pencari atau situs penyedia repository PLBOS, mengenai aplikasi yang kita inginkan.
- ➔ Setelah dapat, unduhlah versi yang terakhir dari aplikasi tersebut.
- ➔ Kemudian ekstrak, biasanya format arsip aplikasinya adalah tar.gz atau tar.bz2. Ketikkan perintah:

```
tar zxvf aplikasi-0.8a.tar.gz
```

atau

```
tar jxvf aplikasi-0.8a.tar.bz2
```

atau lewat menu ekstrak pada *filemanager* di Window Manager atau Desktop Environment Anda.

- ➔ Masuk ke direktori hasil ekstrak.
- ➔ Baca berkas *README*, *ChangeLog*, *NEWS*, *INSTALL*, dan sejenisnya untuk mendapatkan informasi mengenai paket, perubahan apa yang sudah dilakukan

pembuat, cara menginstalnya, dan lain-lain.

- Jika perlu modifikasi kode sumber aplikasi yang umumnya ada di direktori *src/* di dalam direktori hasil ekstrak, sesuaikan dengan kebutuhan.
- Konfigurasi otomatis aplikasi ke dalam sistem dengan skrip *configure*.
- Kompilasi dengan perintah *make*.
- Setelah tidak ada pesan *Error*, lakukan instalasi aplikasi dengan perintah *make install*.

Untuk perangkat lunak yang tidak perlu dilakukan kompilasi seperti aplikasi web dapat langsung dimodifikasi, diinstal, dan dicoba. Jika kita melakukan perubahan pada kode sumber dari salah satu PLBOS, kita harus pula merilisnya secara bebas ke publik, sesuai dengan apa yang tercantum pada lisensi. Minimal kita mengirimkan hasil modifikasi atau *patch* ke pembuatnya, supaya aplikasi bisa berkembang cepat dan semakin bagus kualitasnya, itupun jika modifikasi/*patch* diterima oleh si pembuat. Kita berusaha untuk tidak melakukan *forking*³, kecuali jika memang ada alasan atau tujuan yang berbeda dengan pembuat aslinya, dan ini jarang terjadi di dunia PLBOS.

7. Membangun Distribusi Linux

Membangun distribusi atau distro Linux cukup mudah, yang kita butuhkan adalah pemahaman terhadap sistem Linux, sedikit mengetahui pemrograman terutama *shell* skrip, waktu, dan kesabaran :-). Secara umum yang dilakukan para pengembang dan perusahaan pembuat distro adalah:

- Membuat sistem pengembangan, bisa mulai dari awal atau berbasis atau menggunakan distro yang sudah ada.
- Dengan sistem pengembangan tersebut, pengembang bisa melakukan “pengumpulan” paket-paket yang akan disertakan pada distro. Tentu saja disesuaikan dengan tujuan pembuatan, misal untuk penggunaan umum/*general*, *desktop*, *server*, keamanan, pendidikan, ilmu pengetahuan, atau yang lainnya.
- Yang dilakukan pada paket-paket terkumpul bisa berupa *rebuild* ulang, optimasi arsitektur, modifikasi tampilan, melakukan translasi, menambahkan konfigurasi

³ Membuat proyek baru dari proyek di atasnya, berbasis kode sumber yang sama dengan tujuan, alasan, dan *roadmap* berbeda

spesifik, dan seterusnya. Bisa pula kita membuat aplikasi baru semacam utilitas untuk misal administrasi sistem ataupun pusat kontrol bagi aplikasi yang disediakan. Ada pula pengembang yang langsung menggunakan paket-paket dari distro basisnya tanpa mengubah apapun, hanya menyortir dan meng*index*-nya saja.

- Setelah paket-paket siap, tinggal menggabungkannya dengan *installer*, dan ini tergantung pada basis distro kita.
- Membuat ISO *image* CD/DVD, bila perlu membakarnya ke CD/DVD, kemudian dibundel dengan boks dan dokumentasi.

Seperti yang kita lakukan sebelumnya dengan cara menkonfigurasi, mengkompilasi, serta menginstal menggunakan skrip `configure` dan perintah `make` menyulitkan kita dikemudian hari jika akan melakukan manajemen paket yaitu menghapus, menambahkan, meng-*update*, atau meng-*upgrade* ke versi yang lebih tinggi. Untuk itu kita harus membuat aplikasi tersebut menjadi sebuah paket sesuai dengan standar atau format paket yang dipakai oleh distribusi kita.

Ada tiga standar/format paket di dunia Linux yang banyak dipakai yaitu:

- Debian GNU/Linux dan turunannya: `.deb`.
- Red Hat/Fedora Core dan turunannya: `.rpm`.
- Slackware dan turunannya: `.tgz`.

7.1. Berbasis Debian GNU/Linux dan KNOPPIX

Untuk membuat distro berbasis Debian/KNOPPIX cukup mudah, secara umum yang kita perlukan adalah:

- ✓ CD atau DVD KNOPPIX versi terbaru.
- ✓ Remaster dari CD yang di-*boot* atau lewat instalasi ke *harddisk* terlebih dulu dengan skrip `knoppix-installer`.
- ✓ Kustomisasi, *rebuild*, dan modifikasi: sistem, paket, tampilan.
- ✓ Buat utilitas atau sesuatu yang menonjolkan tujuan dari pembuatan distro.
- ✓ Kompres sistem berkas, bisa menggunakan *cloop* ataupun *squashfs*.
- ✓ Buat ISO *image*-nya setelah itu jadikan CD/DVD (*burn*).

7.1.1. Membuat Paket Debian

Untuk membuat sebuah paket berformat .deb kita disarankan menggunakan Debian GNU/Linux atau turunannya, beberapa tool yang perlu diinstal dalam sistem adalah:

- ➔ Paket penting: `build-essential` (`libc6-dev`, `gcc`, `g++`, `make`, `dpkg-dev`), `perl`, `file`, `patch`
- ➔ Paket rekomendasi: `debhelper`, `dh-make`, `devscripts`, `fakeroot`, `linda`, `lintian`, `gnupg`
- ➔ dependensi aplikasi tersebut, lihat README dan sejenisnya atau pesan kesalahan pada saat kompilasi :-)

Hal-hal yang kita lakukan secara umum untuk membuat sebuah paket Debian, misalnya kita ambil contoh `ntfs-3g`:

- ➔ Kita unduh dari situs penyediaanya di <http://mlf.linux.rulez.org/mlf/ezaz/>.
- ➔ Akan kita dapatkan berkas `ntfs-3g-20070920-BETA.tgz` –ini adalah versi yang penulis unduh. Untuk mengikuti standar nama berkas asli Debian maka kita harus mengubahnya menjadi `ntfs-3g-20070920-BETA.tar.gz` yang nantinya akan berhubungan dengan skrip pembantu `dh_make`.

```
$ mv ntfs-3g-20070920-BETA.tgz ntfs-3g-20070920-BETA.tar.gz
```

- ➔ Mari mengekstraknya.

```
$ tar zxvf ntfs-3g-20070920-BETA.tar.gz
```

- ➔ Kemudian masuk ke direktori hasil ekstrak.

```
$ cd ntfs-3g-20070920-BETA/
```

- ➔ Untuk memudahkan pembuatan paket, kita gunakan skrip `dh_make` untuk menghasilkan direktori `debian/` dan berkas-berkas yang akan digunakan dalam pemaketan. Jangan lupa kita harus menginstal paket `dh-make`. Contoh perintahnya:

```
$ dh_make -e stwn@duniasemu.org -f ../ntfs-3g-20070920-BETA.tar.gz
```

- ➔ Edit berkas yang ada di direktori `debian/` seperti *changelog*, *control*, *copyright*, *dirs*, *docs*, dan *rules*. Pastikan sesuai standar pemaketan yang ada di **Debian Policy Manual**.

- ➔ Setelah semuanya selesai, buat paket dengan perintah `debuild`. Kita sebelumnya harus menginstal `devscripts`, `debhelper`, dan `fakeroot`.

```
$ fakeroot debuild
```

- ➔ Jika pembuatan paket selesai hasilnya akan ada di luar direktori kode sumber, tepatnya di direktori atasnya dengan nama `ntfs-3g_20070920-BETA-1_i386.deb` beserta berkas pendukung lainnya.

7.1.2. KuliAx

KuliAx adalah salah satu distribusi Linux turunan Debian GNU/Linux dan KNOPPIX. Tujuan dari pengembangan distro ini adalah untuk membuat Linux digunakan di perkuliahan atau kampus. Untuk itu disediakan beragam aplikasi dan sistem yang disesuaikan dengan silabus/kurikulum dan kondisi kampus. Versi terakhir dari distro ini adalah 6.0 RC1.



Gambar 7.1. Logo KuliAx

Saat ini KuliAx diarahkan ke jurusan Teknik Informatika, Teknologi Informasi, Teknik Elektro, atau yang berhubungan, walaupun begitu jurusan lain pun dapat menggunakannya sebagai sarana komputasi sesuai dengan kebutuhan, misal dalam bidang statistik, KuliAx 6.0 menyediakan aplikasi R dari R Project dan RKWard atau bidang Matematika dan Pemodelan dengan Octave dan GNU PLOT.



Gambar 7.2. Tangkapan layar KuliAx 6.0 RC1

8. Daftar Pustaka yang Patut Dibaca

- ✓ Debian New Maintainers' Guide, Josip Rodin
<http://debian.org/doc/maint-guide/>
- ✓ Debian Policy Manual, The Debian Policy Mailing List
<http://packages.debian.org/debian-policy/>
- ✓ How To Become A Hacker, Eric Steven Raymond
<http://catb.org/~esr/faqs/hacker-howto.html>
- ✓ Innovation Happens Elsewhere, Ron Goldman dan Richard P. Gabriel
<http://dreamsongs.com>