

Kalau Linux Mau Maju...

Bismillahirrahmanirrahim.

Saya menulis ini sebagai seorang Linux Enthusiast User, sebagai pendukung open source, pemula, dan anggota komunitas Linux. Tulisan ini bertema masa depan. Saya merasa ada yang harus berubah. Tujuannya tulisan ini hanya memperbaiki beberapa sisi Linux. Dan tulisan ini murni opini pribadi. Jadi siapa pun boleh mengambil atau membuangnya. Selamat membaca.

Table of Contents

Kalau Linux Mau Maju.....	1
Batasan Masalah.....	1
1. Stop Bikin Distro Baru.....	1
1.1 Mari Melihat Fakta Jumlah Distro Linux.....	1
1.2 Mari Merenungkan Fakta.....	2
1.3 Mari Mengetahui Dampak Fakta.....	2
2. Fokus Aplikasi.....	2
2.1 Contoh Masalah Aplikasi Desktop.....	3
2.2 Contoh Solusi.....	3
3. Portabilitas Aplikasi.....	3
3.1 Kesulitan Portabilitas Aplikasi di Linux.....	4
3.2 Contoh Permasalahan Portabilitas Aplikasi.....	5
3.3 Contoh-Contoh Solusi Portabilitas.....	6
3.4 Contoh-Contoh Ide Solusi.....	6
4. Waktu Rilis.....	7
5. Android? Hardware!.....	8
5.1 Contoh-Contoh Masalah Hardware di Linux Desktop.....	8
5.2 Contoh Solusi Hardware di Linux Desktop.....	9

Batasan Masalah

Linux desktop, bukan server atau embedded.

1. Stop Bikin Distro Baru

1.1 Mari Melihat Fakta Jumlah Distro Linux

Perhatikan laman Distrowatch per 13 Oktober 2014 pukul 19.39:

<http://distrowatch.com/search.php?status=All>

Jumlah distro Linux di seluruh dunia versi mereka adalah 788. Tidak perlu mencari referensi lain.

1.2 Mari Merenungkan Fakta

Untuk memiliki satu buah sepeda motor, Anda diberi 700 pilihan. Orang yang memberi pilihan tentunya orang yang berniat baik. Namun apakah pilihan itu sendiri baik? Tidak.

1.3 Mari Mengetahui Dampak Fakta

1. **Waktu terbuang.** Adakah orang yang mau mencoba satu per satu seluruh distro itu? Angka 788 hanyalah angka untuk nama, sedangkan per nama distro umumnya memiliki lebih dari 1 dukungan arsitektur dan lebih dari 1 versi.
2. **Duplikasi bug.** Dengan banyaknya remasteran, maka bug yang ada dari distro asal akan terduplikasi ke dalam distro remasteran. Sedangkan profesi bug hunter adalah sulit dan jarang. Tidak mungkin setiap remasteran memiliki bug hunter yang kompeten.
3. **Tugas bertambah.** Dengan banyaknya duplikasi bug, maka tugas membetulkan bug itu pun terduplikasi dan menjadi berlipat banyaknya. Sedangkan profesi bug hunter bukan dimiliki semua orang. Lalu siapa yang bertanggung jawab kepada pengguna jika terdapat suatu bug spesifik?
4. **Tugas lain ikut bertambah.** Apakah pembuatan suatu distro (baca: sistem operasi) hanya terhenti di bugfixing? Tidak. Masih ada dokumentasi yang sudah pasti beda per nama distro. Dokumentasi itu sangat-sangat besar. Lihat contoh Wiki Archlinux yang sudah besar sekali. Membangun wiki semacam itu tentu lama sekali. Dan tentu saja dokumentasi Archlinux tidak cuma wiki. Itu belum termasuk artwork, DE, repositori, dan lain-lain.
5. **Ruang penyimpanan membengkak.** Ada distro ada repo. Distro bertambah repo berlipat. Satu distro bisa punya 2 x 20 GB repo. Tidak semua server bisa menampung + me-maintain kontinu setiap repo yang ada.
6. **Bingung memilih.** Bagi mereka yang telah memiliki pengalaman, memilih bukan hal sulit. Tetapi bagi pemula sebaliknya. Apalagi di dalam komunitas, jawaban yang muncul seringkali tidak satu suara.

2. Fokus Aplikasi

Yang hampir pasti selalu dibicarakan di forum Linux mana pun adalah aplikasi. Telah demikian banyak aplikasi *free software* dibuat untuk Linux, bahkan ada yang *proprietary*. Kita bisa melihat daftarnya di <http://alternativeto.net> atau semacamnya. Namun jelas di sebagian segi, aplikasi untuk Linux belum dapat 100% menggantikan fungsionalitas aplikasi untuk Windows. Sebutlah nama Microsoft Office, CorelDRAW, AutoCAD, FruityLoops, AfterEffects, DeepFreeze¹, dan lain-lain. Berikut penjelasannya.

¹ Nama-nama aplikasi yang disebut di sini adalah komersial dan propietari.

1. Jika memang niat dan sumber daya pengembang cukup, maka kiranya pengembangan aplikasi untuk distro tertentu lebih efisien dibandingkan berdarah-darah membuat distro baru atau remasteran. Pengembangan di sini termasuk pula penerjemahan seperti <http://translations.launchpad.net>.
2. Jika pengembangan aplikasi dirasakan sulit pula, maka jalan terakhir adalah fokus pada aplikasi dalam hal penulisan tutorial atau *review* atau hal-hal lain yang sifatnya mengajarkan aplikasi tersebut kepada masyarakat.

2.1 Contoh Masalah Aplikasi Desktop

1. Printer Resetter untuk sebagian merek dan model.
2. Aplikasi *desktop* yang portabel dan universal untuk pengajaran sekolah.
3. Aplikasi office yang kompatibel 100% dengan format Microsoft.
4. Aplikasi backup untuk aplikasi yang sudah terinstal.
5. Aplikasi backup untuk konfigurasi yang dilakukan (semacam Deepfreeze).
6. Aplikasi khusus belajar Linux.
7. Dan lain-lain.

Tentu dengan catatan, nomor 3 mendekati mustahil untuk dibuat sendiri.

2.2 Contoh Solusi

Pengembangan diarahkan lebih kepada aplikasi daripada kepada sistem operasinya.

3. Portabilitas Aplikasi

Yang dimaksud portabilitas aplikasi di sini adalah kemampuan **satu** *installer* aplikasi untuk dipasang di

1. Suatu distro tanpa meminta dependensi tambahan.
2. Distro yang sama tetapi beda versi.
3. Distro yang sama tetapi beda arsitektur.
4. Distro berbeda (tetapi sama-sama DEB/RPM) dan beda versi. Misalnya antara Mint 17 dan Ubuntu 14.04.
5. Distro berbeda dan beda arsitektur. Misalnya Mint 17 32 bit dan Ubuntu 14.04 64 bit.

Semua hal tersebut adalah sangat sulit dilakukan di Linux. Di Linux, hampir semua distro memanfaatkan sistem manajemen pakatnya sendiri-sendiri (oleh apt, yum, urpmi, zypper, slackpkg, pacman, dll.) yang mengambil aplikasi selalu dari repositori terpusat semacam *Applications Store*. Saya tidak akan jauh menjelaskan sistem manajemen paket karena telah ditulis [di sini](#), [di sini](#), dan [di sini](#). Namun saya akan menjelaskan kesulitan-kesulitan dan solusinya.

3.1 Kesulitan Portabilitas Aplikasi di Linux

Jangan buru-buru membantah. Saya membicarakan keumuman dan mayoritas. Saya tidak sedang memperhitungkan minoritas di sini. Satu paket di Linux bisa berwujud DEB atau RPM atau TGZ. Saya juga tidak sedang membicarakan manajemen paket ala Gentoo²³ dkk. di sini.

1. Di Linux, pada keumuman aplikasinya Anda melihat *dynamic linking*. Di Windows, pada keumuman aplikasinya Anda melihat *static linking*.
2. Oleh karena itu, Anda melihat *dependency resolving* di Linux. Anda tidak melihatnya di Windows.
3. Oleh karena itu, Anda mengunduh banyak paket untuk 1 aplikasi saja. Anda tidak melakukannya di Windows karena cukup unduh 1 paket EXE untuk 1 aplikasi.
4. Di Linux, satu paket membutuhkan paket yang lain supaya bisa diinstal dan berjalan. Di Windows, satu paket sudah berisi paket-paket yang lain tersebut.
5. **Keuntungan di sisi Windows:** Oleh karena itu, Anda pengguna *desktop*, menganggap instalasi aplikasi di Linux lebih rumit daripada di Windows. Atau sebaliknya, Anda merasa instalasi aplikasi di Windows lebih mudah dan sederhana daripada di Linux. Ini karena Anda cukup mengunjungi satu situs, mengunduh 1 EXE, dan menginstal aplikasi secara sempurna dari 1 EXE tersebut tanpa dependensi dan keuntungan jangka panjangnya 1 EXE ini bisa disimpan untuk diinstal di masa mendatang.
6. **Keuntungan di sisi Linux:** Oleh karena itu, Anda hemat ruang HDD karena *dynamic linking* yakni satu paket yang dibutuhkan oleh banyak aplikasi, cukup satu kali diinstal dan tidak berulang-ulang kali diinstal di Linux.
7. **Kerugian di sisi Windows:** ada kemungkinan paket diubah sebelum sampai ke pengguna, karena di Windows, tidak ada manajemen pemaketan terpusat⁴⁵⁶⁷ seperti di Ubuntu dan lain-lain. Oleh karena itu, Anda menemukan banyak paket yang berbahaya di Windows karena virus atau orang yang tidak bertanggung jawab. Kerugian lainnya adalah biasanya ukuran paketnya besar-besar (tentu karena memuat banyak paket lainnya).
8. **Kerugian di sisi Linux:** instalasi aplikasi menjadi lebih panjang langkah-langkahnya (karena ada *dependencies resolving*) alias rumit dan *installer* aplikasi tidak bisa disimpan dengan mudah alias tidak bisa disimpan⁸ untuk instalasi jangka panjang di

2 <https://www.gentoo.org/doc/en/handbook/handbook-x86.xml?part=2&chap=1>

3 [http://en.wikipedia.org/wiki/Portage_\(software\)](http://en.wikipedia.org/wiki/Portage_(software))

4 <https://wiki.ubuntu.com/MOTU>

5 <https://wiki.ubuntu.com/NewDevelopersAndMaintainers>

6 <https://wiki.ubuntu.com/UbuntuDevelopment>

7 <https://wiki.ubuntu.com/Ubuntu/ForDebianDevelopers>

8 Bisa disimpan, tetapi lebih rumit dan panjang jalannya karena Anda harus berurusan dengan `/var/cache/apt/archives` dengan satu per satu melakukan instalasi dan menyimpan satu per satu dependensi paket dengan ketelitian tinggi. Ini karena pengunduhan paket **tidak dipisahkan per nama aplikasi ke dalam folder-folder tersendiri**. Hal yang sama berlaku untuk yang selain apt, misalnya yum dan lain-lain karena cara kerjanya sama secara garis besar. Maka bahasa yang cocok untuk ini adalah "tidak bisa disimpan".

masa mendatang. Jadi, Anda tidak bisa⁹ mengunduh dahulu paket di komputer lain dari *browser* untuk diinstal kemudian di Linux, karena *dependencies resolving* tidak dapat dilakukan kalau komputer tidak tersambung langsung ke internet.

9. Sistem satu paket di Windows inilah yang saya sebut sebagai *single offline installer*. Anda bisa instal suatu paket secara *offline* 100% di komputer Anda, tanpa harus terhubung secara langsung ke server tertentu selama instalasi berlangsung. Kita tidak bisa menyebut **nonsingle offline installer** untuk sistem multipaket di Linux, karena selama instalasi berlangsung, harus terjadi koneksi langsung antara komputer Anda dan server repositori.
10. Itulah perbandingan antara sistem instalasi Windows yang 100% *offline* dengan Linux yang tidak 100% *offline*. Perhatikan penekanan-penekanan yang saya berikan. Cukup adil perbandingannya, bukan?

3.2 Contoh Permasalahan Portabilitas Aplikasi

Setelah membaca poin **Kesulitan** di atas, Anda setidaknya akan paham perbedaan sangat mendasar sistem instalasi aplikasi di antara kedua OS. Anda diharap paham mengapa orang menyebut instalasi aplikasi di Windows lebih mudah daripada Linux, dan tidak perlu *ngotot*. Dan Anda tidak bingung lagi mengapa saya membuat istilah *single offline installer* di sini. Sekarang Anda perlu melihat langsung contoh-contohnya. Inilah kesulitan yang dirasakan masyarakat Anda, entah Anda menyadarinya atau tidak.

1. Ketika seseorang ingin memasang modem di Ubuntu, karena **wvdial** belum terinstal¹⁰, maka dia akan mencari di internet paket wvdial sesuai versi Ubuntu-nya (ingat poin-poin dalam portabilitas aplikasi di atas). Dia pasti mengunduh (di warnet atau yang lain) hanya paket wvdial itu sendiri, tanpa memperhitungkan dependensinya. Sampai di rumah, dia kebingungan karena ternyata wvdial membutuhkan paket yang lain. Ini terjadi terus menerus sampai paketnya dilengkapi semua baru wvdial berhasil diinstal. Sekalipun dia tahu semua dependensinya, keumuman orang akan malas mengunduh satu per satu dari laman berbeda seperti di pkgs.org. Ini satu poin kesulitan tersendiri, akibat kelemahan dari *dynamic linking*.
2. Ketika seseorang tidak memiliki internet langsung di Linux, dia ingin menginstal satu paket, dia meminta kepada komunitas melalui *handphone* untuk diunggah aplikasi yang diminta. Jika dia menggunakan Windows, warga yang ingin membantu cukup mengunggah 1 paket EXE dan dia cukup mengunduh 1 paket EXE dan bisa diinstal sempurna di komputer *offline*. Karena dia menggunakan Linux, maka warga yang ingin membantu harus repot mencarikan dependensi dan mengunggah banyak

9 Bisa sebenarnya, tetapi sangat sulit karena 1) hanya Ubuntu yang memiliki situs semacam apt-web 2) kalau ada pkgs.org, Anda hanya bisa mengunduh 1 paket tanpa dependensinya karena sampai hari ini 17 Oktober 2014 mereka belum memiliki fitur *dependencies resolving* seperti apt-web 3) Anda harus mengunduh semua dependensi yang ditemukan satu per satu tanpa boleh ada yang terlewat atau komputer Anda akan error nantinya 4) situs hebat semacam apt-web membutuhkan admin yang bagus dan memiliki dedikasi tinggi kepada komunitas serta biaya yang cukup untuk mempertahankannya. Cukup adil?

10 Saya menulis ini bukan berarti saya tidak tahu wvdial sudah ada di dalam CD Ubuntu itu sendiri. Sekali lagi, saya hanya membicarakan keumuman dan mayoritas. Berapa banyak orang yang tahu wvdial sudah disertakan di dalam CD Ubuntu? Sudahkah kita menulis tutorialnya?

dependensi tersebut plus memberitahukan cara instalasinya. Pertanyaannya, siapa yang mau repot melakukan itu? Ini sudah satu poin kesulitan berikutnya.

3. Ketika seseorang ingin menginstal driver hardware di Linux, komputer dia harus online ketika instalasi berlangsung. Jika tidak, maka tidak ada driver yang diunduh. Pertanyaannya, jika si pengguna benar-benar *offline* 100%, bagaimana dia bisa menginstal driver yang dia butuhkan?

3.3 Contoh-Contoh Solusi Portabilitas

1. **apt-web buatan Fajran Iman Rusadi.** Ini adalah salah satu solusi paling brilian yang pernah ada. Solusi ini benar-benar telah membantu banyak orang di Indonesia (termasuk saya). Namun, ada 2 hal penting. Pertama, solusi ini butuh server yang kuat dan di-maintain dengan konsisten. Sementara di Indonesia terlalu sedikit orang yang bisa/mampu melakukan itu. Kedua, solusi ini masih berpeluang dikembangkan ulang. Misalnya saja ditambah satu opsi untuk memaketkan seluruh dependensi jadi 1 ZIP sehingga pengguna hanya perlu mengunduh 1 paket, persis seperti Windows. Ini akan memotong banyak sekali kewajiban-kewajiban sehingga menghemat waktu dan usaha. Hasilnya efisiensi lebih maksimal diperoleh. Kendala: server tidaklah gratis dan konsistensi itu mahal.
2. **apt-id dan alldeb buatan Nifa Dwi Kurniawan.** Ini adalah solusi yang begitu gemilang setelah adanya apt-web. Dua program ini bekerja di sisi client tanpa perlu server untuk mengatasi keterbatasan *dynamic linking* di Ubuntu. Begitu banyak orang yang sudah terbantu dengan keberadaannya. GUI untuk alldeb bahkan sudah ada, baik dari Nifa sendiri atau versi Qt buatan Slamet Badwi. Silakan kunjungi laman resminya [di sini](#) dan [di sini](#).
3. **dot isi buatan wagung.** Ini adalah solusi yang sejak awal kemunculannya sudah disambut dan sampai sekarang pun masih banyak di-request pembaca situs resminya sendiri (<http://twitter.web.id>). Konsepnya sama persis dengan *single offline installer* di Windows, satu buah berkas berekstensi ISI untuk menginstal satu nama program. Dependensi sudah disertakan di dalam ISI tersebut, sama persis dengan EXE di Windows. Tampak sekali kalau wagung paham betul kelebihan sistem instalasi Windows dan mengadaptasikannya dengan hampir sempurna di Linux. Sebuah cara yang jitu dan pantas saya puji.
4. **PortableLinuxApps.** Jika Anda jeli, semua solusi yang saya sebut di atas adalah buatan orang Indonesia. Jelas betul, karena yang bermasalah dengan internet kebanyakannya adalah warga Indonesia. Warga luar negeri seperti Eropa tidak ada masalah sama sekali, karena internet mereka di sana tidak seperti di sini. Namun ternyata ada warga luar negeri yang membuat sistem persis seperti Windows yakni proyek [PortableLinuxApps](#). Sepertinya dia sadar pentingnya masalah ini, hanya saja komunitas di luar negeri sana tampak kurang peduli. Tentu saja wajar, karena internet bukan lagi masalah bagi mereka.

3.4 Contoh-Contoh Ide Solusi

Ide-ide ini adalah murni dari saya pribadi. Anda bisa mempertimbangkannya.

1. **Service Pack.** Anda pernah menggunakan Windows? Tentu Anda tahu *service pack* atau *driver pack*, suatu DVD yang isinya paket-paket untuk menginstal atau memperbarui sistem di Windows. Kenapa kita tidak membuat yang semacam itu saja? Bukan hanya semata-mata untuk upgrading, tetapi untuk instal aplikasi-aplikasi tertentu untuk distro tertentu. Misalnya *service pack* khusus untuk instal *codec* pada Ubuntu 14.04, *service pack* khusus desain grafis, *service pack* khusus driver NVIDIA, *service pack* khusus LaTeX, dan sebagainya. Jadi intinya si pengguna cukup unduh 1 paket yang berisi semua kebutuhan khusus dia. Entah ukurannya besar atau kecil, pengguna pasti akan mengunduhnya karena mereka butuh. Developer BlankOn dan IGOS layak mempertimbangkan ide ini.
2. **Enhanced apt-web.** Mengulang yang saya katakan di atas, yaitu apt-web biasa dengan kemampuan pemaketan *on the fly* ke dalam 1 berkas untuk berapa pun dependensi yang ditemukan. Jadi pengguna minta 1 aplikasi, apt-web bertugas mengumpulkan itu semua ke dalam 1 berkas (misalnya ZIP atau RUN atau bahkan DEB juga), pengguna cukup unduh 1 berkas itu bukan semua dependensinya satu per satu. Kelemahan ide ini adalah kebutuhan *resource* yang sangat besar. Namun seandainya bisa diakali dengan pemrograman web yang memungkinkan *batch downloading* dalam 1 klik saja di sisi server, maka itu akan memudahkan.
3. **Distributed Dot ISI.** Distributed dalam arti *branding*, dibikin suatu brand yang jelas untuk ISI yang dipioneri oleh wagung. Disebarluaskan dan diseminarkan, serta dipakai secara nyata di komunitas-komunitas internet. Contoh paling pentingnya dibuatkan situs khusus yang berisi dot ISI dalam kategori-kategori. Pengguna cukup cari sesuai distronya dan mengunduh mana yang diperlukan.
4. **alldub dan apt-id.** Ini karya cipta Nifa Dwi Kurniawan. Ini sangat brilian. Jika ini masuk ke repo resmi Ubuntu bahkan dipaketkan dalam ISO, ini akan sangat membantu pengguna Indonesia.

4. Waktu Rilis

Ubuntu dan beberapa distro lain memiliki jadwal 6 bulan sekali untuk rilis. Agaknya hal ini terlalu cepat bagi sebagian pengguna. Walaupun versi LTS didukung selama 5 tahun, tetapi nonLTS hanya 9 bulan (harus *upgrade* jika habis masanya). Debian memiliki jadwal yang lebih longgar lagi, prinsip Debian saja *rilislah kalau sudah siap betul*¹¹. Archlinux memakai sistem *rolling release*, yang bisa dibilang *forever LTS*. Tidak ada kata *support* habis seperti Ubuntu nonLTS, karena versi lama tetap bisa 1) instal aplikasi 2) di-upgrade langsung ke versi paling baru.

Dalam hal ini, jika Anda pengembang distro, pertimbangkan kekuatan tim Anda. Bisakah Anda mengejar waktu rilis 6 bulan sekali? Sampai kapan tim sanggup melakukannya (apalagi jika tanpa *feedback* dana)? Buat saja selonggar mungkin supaya tim enak pengguna juga enak.

11 <https://wiki.debian.org/ReleaseWhenReady>

5. Android? Hardware!

Android dianggap sebagai saudara kandung Linux desktop kita. Ya, karena kernel Android juga Linux. Namun setidaknya ada satu hal yang membedakan Android dengan Linux desktop kita. Yaitu dukungan driver *hardware*. Android didukung langsung oleh Google dan [Open Handset Alliance](#). Bagaimana dengan Linux *desktop* kita? Permasalahan yang kita hadapi sebenarnya kompleks sekali. Dukungan vendor-vendor *hardware* itu yang paling penting bagi sebuah OS. Namun justru itulah yang menjadi masalah di Linux desktop. Berikut penjelasannya.

1. Android di dalam setiap perangkat, didukung oleh vendor *hardware* secara langsung. Jadi secara umum, tidak ada cerita *error* instal driver di Android. Linux desktop kita didistribusikan terpisah dari laptop/PC-nya dan tidak ada *support* secara langsung antara vendor hardware dengan laptop/PC kita.
2. **Keunggulan Linux:** Linux itu sangat-sangat mandiri. Driver-driver dibuat sendiri, disertakan di dalam kernel, dan dia bisa mengenali semua *hardware* dalam laptop/PC kita walau kita tidak menginstal CD driver dulu.
3. **Kelemahan Linux:** Linux desktop tidak seperti Windows (di ranah desktop) dan tidak seperti Android (di ranah mobile). Di dunia ini, bisa dibayangkan, seluruh hardware diciptakan dan dioptimalkan untuk Windows. Jadi jangan heran jika Anda selalu memperoleh CD driver untuk Windows setiap kali membeli suatu *hardware*. Ini karena produsen *hardware* sengaja membuat driver untuk Windows. Linux? Tidak selalu seperti itu. Ada produsen-produsen yang tidak merilis driver untuk Linux. Kalaupun merilis, drivernya tidak sesempurna driver Windows (misalnya sebagian driver VGA). Dan kalaupun merilis driver/*hardware* dikenali otomatis oleh kernel, sebagian mereka tidak merilis aplikasi bantu untuk Linux (misalnya manajer GUI untuk modem USB).

Linux desktop dengan Linux mobile adalah berbeda satu sama lain. Berbeda jalan, dukungan, dan metode pemakaian.

5.1 Contoh-Contoh Masalah Hardware di Linux Desktop

1. **Modem USB.** Kiranya Anda sudah mengerti kalau ZTE, HUAWEI, Haier, Sierra, AT&T, **tidak membuat dan/atau memasukkan** program manajer modemnya versi Linux ke dalam setiap modem. Hasilnya, banyak pengguna baru merasa kesulitan menginstal modem (walau ujung-ujungnya banyak yang berhasil melalui tutorial). Bagi profesional ataupun nonprofesional yang bekerja di bidang selain *ngoprek*, hal ini amat sangat mengganggu. Maka jelaslah bagaimana perasaan mereka yang enggan menggunakan Linux karena masalah ini.
2. **VGA.** Berapa banyak sudah pertanyaan muncul di forum masalah layar BSOD¹² di Linux karena driver VGA? Belum lagi masalah resolusi, artefak, dan *watermark*. Bagi profesional maupun nonprofesional, hal ini sangat mengganggu. Maka wajar kalau

12 Black Screen Of Death.

sebagian mereka sampai menolak karena masalah ini.

3. **WLAN.** Berapa banyak sudah pertanyaan muncul masalah WLAN yang tidak dikenali sama sekali setelah instal Linux? Atau pertanyaan seputar performa WLAN di Linux?

5.2 Contoh Solusi Hardware di Linux Desktop

1. Vendor-vendor *hardware* beraliansi untuk Linux. Seperti Google dan OHA, seperti HP dan Red Hat¹³.
2. Vendor-vendor menyediakan *hardware* kompatibel Linux dan melabeli semua produknya dengan itu.
3. Vendor-vendor menyengaja membuat *hardware* mereka kompatibel Linux dan menyengaja pula memasang label *Linux ready* di samping Windows dan Mac OS pada semua boks penjualan *hardware*.
4. Vendor-vendor menyediakan driver resmi siap unduh pada situs resminya untuk Linux. Misalnya untuk printer: HP¹⁴, Canon¹⁵, Epson¹⁶, dan Brother¹⁷. Misal untuk server: HP dengan Red Hat Enterprise Linux¹⁸ atau HP dan SUSE Linux Enterprise¹⁹ atau HP dan Canonical Ubuntu²⁰ atau HP dan CentOS²¹.
5. Distro-distro besar mengadakan sertifikasi *hardware*²² untuk Linux. Ini akan meningkatkan drastis rasa percaya diri masyarakat menggunakan Linux dan hardware-nya.
6. Vendor-vendor memasarkan PC *built-up* atau laptop dengan Linux prainstal. Ini sudah diawali oleh System76 (laptop²³ dan desktop²⁴), Dell²⁵, dan Samsung (Chromebook²⁶).
7. Vendor-vendor pihak ketiga memasarkan PC atau laptop dari vendor asli dengan Linux prainstal. Contohnya EmperorLinux²⁷, ThinkPenguin²⁸, dan LinuxCertified²⁹.
8. Vendor-vendor kartu grafis merilis driver-driver mereka sebagai *open source* atau paling tidak memberikan kesamaan fitur *driver* propietari antara Windows dan Linux. Supaya tidak ada lagi kejadian Linux panas atau kualitas grafis jelek karena kualitas driver yang "dibedakan" dari Windows.

Memang yang memegang peranan paling penting di sini adalah vendor *hardware*. Jika tidak, maka developer kernel dan driver Linux akan bekerja lebih berat membuat driver versi *open*

13 <http://h22168.www2.hp.com/us/en/partners/red-hat/>

14 <http://hplipopensource.com/hplip-web/index.html>

15 <http://www.canon-europe.com/Printers/Home-printers/>

16 <http://download.ebz.epson.net/dsc/search/01/search/?OSC=LX>

17 http://solutions.brother.com/linux/en_us/

18 http://h17007.www1.hp.com/us/en/enterprise/servers/supportmatrix/redhat_linux.aspx

19 http://h17007.www1.hp.com/us/en/enterprise/servers/supportmatrix/suse_linux.aspx

20 <http://h17007.www1.hp.com/us/en/enterprise/servers/supportmatrix/ubuntu.aspx>

21 http://h17007.www1.hp.com/us/en/enterprise/servers/supportmatrix/cent_os.aspx

22 Contohnya <http://www.ubuntu.com/certification/>.

23 <https://system76.com/laptops/>

24 <https://system76.com/desktops/>

25 <http://www.dell.com/us/business/p/xps-13-linux/pd>

26 <http://www.samsung.com/us/computer/chromebook>

27 <http://www.emperorlinux.com/>

28 <https://www.thinkpenguin.com>

29 <http://www.linuxcertified.com/>

source (yang belum tentu 100% kompatibel).