MENGGUNAKAN LINUX

3

OBJEKTIF :

1. Mahasiswa Mengetahui Tentang Pengembangan Linux.
2. Mahasiswa Mengetahui Aplikasi Linux Serta Penggunaan dan Pengembangannya.
3. Mahasiswa Mengenali Keterampilan Dasar Dalam Menggunakan Linux
4. Mahasiswa Mengetahui Tentang Lisensi Pada Linux dan Para Kreatif

---

PENDAHULUAN

Untuk menjadi administrator sistem Linux, kita harus terbiasa dengan Linux sebagai sistem operasi dekstop serta memiliki kemahiran dan keterampilan Teknologi Informasi dan Komunikasi (TIK) dasar. Pemilihan Linux untuk tugas-tugas produktivitas dapat mempercepat pembelajaran dalam menggunakan Linux. Seorang administrator sistem, tidak hanya mengelola server tetapi terkadang diminta untuk membantu mengenai masalah konfigurasi, merekomendasikan perangkat lunak baru, dan lainnya.

Sebagian besar distribusi Linux, membolehkan calon penggunanya untuk mengunduh paket instalasi "*desktop*" dan dimuatkan ke USB. Sebagai calon administrator sistem Linux, kita perlu mengetahui cara instalasi dan prosesnya. *Desktop* Linux seharusnya sudah tidak asing bagi yang telah menggunakan PC atau Macintosh, karena tampilannya mirip dan untuk menjalankan program atau aplikasi cukup dengan memilih ikon yang dimaksud. Perlu diketahui, kita dapat menggunakan aplikasi "*settings*" untuk mengkonfigurasi akun pengguna, jaringan WiFi dan perangkat input. Setelah terbiasa dengan *GUI* atau *desktop* Linux, maka

langkah selanjutnya adalah mempelajari cara memberikan instruksi ke sistem melalui baris perintah atau *Command Line*.

Proyek perangkat lunak mengambil bentuk kode sumber, yang merupakan serangkaian instruksi komputer yang dapat dibaca manusia. Karena kode sumber tidak dipahami secara langsung oleh komputer, kode itu harus dikompilasi ke dalam instruksi mesin oleh kompiler. Compiler adalah program khusus yang mengumpulkan semua file kode sumber dan menghasilkan instruksi yang dapat dijalankan di komputer, seperti kernel Linux.

Pertimbangkan ini
Kode sumber yang dikompilasi ke dalam program biner adalah salah satu metode untuk membuat program dan menjalankan instruksi komputasi. Banyak jenis bahasa yang ditafsirkan, seperti PERL, Python dan bahkan skrip BASH, di mana kode tidak dikompilasi, tetapi diumpankan ke program interpreting, biasanya biner yang dapat dieksekusi yang memahami dan mengimplementasikan instruksi yang terkandung dalam kode sumber atau skrip.

Secara historis, perangkat lunak komersial telah dijual di bawah closed source license, yang berarti bahwa pengguna memiliki hak untuk menggunakan kode mesin, tetapi tidak dapat melihat kode sumber. Seringkali lisensi secara eksplisit menyatakan bahwa pengguna tidak boleh mencoba merekayasa balik kode mesin kembali ke kode sumber untuk mengetahui apa fungsinya.

Pengembangan Linux sangat memengaruhi kebangkitan perangkat lunak open source. Sebelumnya ada shareware, program yang tersedia secara bebas di mana pengguna tidak perlu memiliki akses ke kode sumber. Ada banyak hal baik tentang hal ini, tetapi juga bermasalah karena program jahat dapat disamarkan sebagai game, screensaver, dan utilitas yang tampak tidak bersalah.

Open source mengambil tampilan sumber-sentris dari perangkat lunak. Filosofi open source adalah bahwa pengguna memiliki hak untuk mendapatkan kode sumber perangkat lunak, dan untuk memperluas dan memodifikasi program untuk penggunaan mereka sendiri. Ini juga berarti kode tersebut dapat diperiksa untuk backdoors, virus, dan spyware. Dengan menciptakan komunitas

pengembang dan pengguna, pertanggungjawaban untuk bug, kerentanan keamanan, dan masalah kompatibilitas menjadi tanggung jawab bersama. Komunitas penggemar komputer global yang baru ini diberdayakan oleh meningkatnya ketersediaan layanan internet yang lebih cepat dan web di seluruh dunia.

Ada banyak varian open source yang berbeda, tetapi semua sepakat bahwa pengguna harus memiliki akses ke kode sumber. Perbedaannya adalah bagaimana seseorang dapat, atau harus, mendistribusikan kembali perubahan.

Linux telah mengadopsi filosofi ini dengan sangat sukses. Karena Linux ditulis dalam bahasa pemrograman C, dan itu mencerminkan desain dan fungsionalitas sistem UNIX yang sudah mapan, secara alami menjadi forum di mana orang dapat mengembangkan dan berbagi ide-ide baru. Terbebas dari keterbatasan platform perangkat keras dan perangkat lunak berpemilik, sejumlah besar programmer yang sangat terampil telah mampu berkontribusi pada berbagai distribusi, membuat perangkat lunak yang seringkali lebih kuat, stabil, mudah beradaptasi, dan, terus terang, lebih baik daripada yang eksklusif, ditutup penawaran sumber yang mendominasi dekade sebelumnya.

Banyak organisasi yang curiga menggunakan perangkat lunak yang dibangun dengan cara baru ini, tetapi seiring waktu mereka menyadari bahwa programmer terbaik mereka bekerja pada proyek open source berbasis Linux di waktu luang mereka. Segera, server-server Linux dan program-program open source mulai mengungguli sistem mahal yang sudah ada. Ketika tiba saatnya untuk memperbarui perangkat keras yang sudah ketinggalan zaman, para programmer, insinyur, dan administrator sistem yang sama yang mulai bekerja di Linux sebagai hobi dapat meyakinkan bos mereka untuk mencoba Linux. Sisanya, seperti kata mereka, sejarah.

Sebelum pengembangan Linux, banyak aplikasi korporat dan ilmiah dijalankan pada sistem UNIX. Perusahaan, universitas, dan pemerintah yang

menjalankan server farm besar menyukai stabilitas dan relatif mudahnya pengembangan aplikasi yang ditawarkan platform ini.

UNIX awalnya dibuat pada tahun 1969. Pada edisi keempatnya, pada tahun 1973, ia telah ditulis ulang dalam bahasa pemrograman C yang masih menonjol hingga saat ini. Pada tahun 1984 University of California Berkeley merilis 4.2BSD yang memperkenalkan TCP/IP, spesifikasi jaringan yang menopang Internet. Pada awal 1990-an, ketika pengembangan Linux dimulai, berbagai perusahaan yang mengembangkan sistem operasi UNIX menyadari sistem mereka perlu kompatibel, dan mereka mulai bekerja pada spesifikasi X/Open yang masih digunakan sampai sekarang.

Selama bertahun-tahun, para ilmuwan komputer dan organisasi yang mempekerjakan mereka telah menyadari manfaat dari sistem yang menyediakan alat yang akrab dan cara konsisten dalam menyelesaikan tugas tertentu. Standarisasi antarmuka pemrograman aplikasi (API) memungkinkan program yang ditulis untuk satu sistem operasi UNIX atau Linux tertentu untuk di-porting (dikonversi) relatif mudah dijalankan pada yang lain. Jadi, sementara sistem UNIX yang dipatenkan masih digunakan di seluruh dunia dalam lingkungan di mana solusi "bersertifikat" lebih disukai, interoperabilitas sistem ini bersama komputer Linux dinilai oleh industri, akademisi, dan pemerintah yang menggunakannya.

Pentingnya standar organisasi tidak dapat dilebih-lebihkan. Kelompok-kelompok seperti IEEE (Institute of Electrical Engineers) dan POSIX (Portable Operating System Interface), memungkinkan para profesional dari berbagai perusahaan dan lembaga untuk berkolaborasi pada spesifikasi yang memungkinkan berbagai sistem operasi dan program untuk bekerja bersama. Tidak masalah jika suatu program closed atau open source, sederhana atau kompleks, jika ditulis dengan standar ini, orang lain akan dapat menggunakan dan memodifikasinya di masa depan. Setiap inovasi dalam komputasi dibangun di atas karya orang lain yang datang sebelumnya. Perangkat lunak open source adalah kolaborasi dari orang-orang yang berbeda dengan kebutuhan dan latar belakang

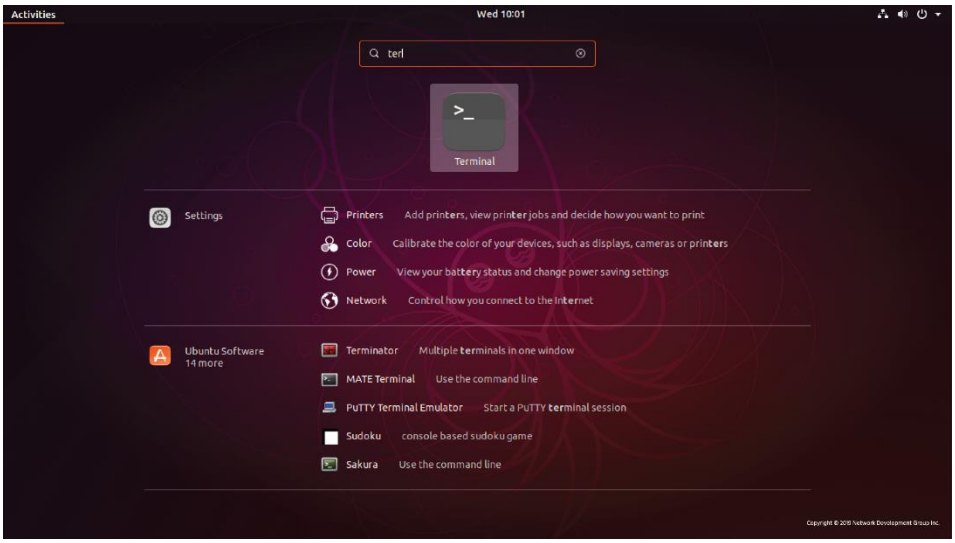
yang berbeda yang semuanya bekerja bersama untuk membuat sesuatu yang lebih baik daripada yang dapat dibuat oleh mereka secara individual. Standar adalah apa yang memungkinkan hal ini, dan banyak organisasi yang membuat, memelihara, dan mempromosikannya merupakan bagian integral dari industri.

4.1 MENAVIGASI LINUX

Antarmuka baris perintah atau *Command line interface* (CLI) adalah sistem input teks sederhana untuk memasukkan perintah atau instruksi tunggal hingga skrip yang rumit. Kebanyakan sistem operasi memiliki CLI yang menyediakan cara langsung untuk mengakses dan mengontrol komputer.

Pada sistem yang *boot* ke GUI, terdapat dua cara umum untuk mengakses baris perintah yaitu – terminal berbasis GUI dan terminal virtual:

- Terminal berbasis GUI adalah program yang mengemulsi jendela terminal. Terminal GUI dapat diakses melalui sistem menu. Contohnya, pada distro Linux CentOS, kita dapat membuka terminal dengan mengklik **Applications** di bar menu, kemudian **System Tools >** dan terakhir, **Terminal**. Selain itu, **Terminal** dapat dibuka melalui alat pencarian pada sistem, dengan mengetiknya seperti pada gambar dibawah ini.



- Terminal virtual dapat dijalankan bersamaan dengan GUI, tetapi memerlukan akun pengguna untuk *log in* (seperti *log* in saat mengakses GUI) melalui terminal virtual sebelum dapat mengeksekusi perintah.

Setiap distribusi desktop Linux memiliki sedikit perbedaan, tetapi aplikasi **Terminal** atau **x-term** akan membuka jendela terminal GUI. Meskipun ada perbedaan tipis pada *session* antara teminal *console* dan terminal jendela, tetapi keduanya sama dari sudut pandang administrator dan membutuhkan pengetahuan yang sama tentang perintah atau *command* untuk digunakan.

Tugas yang dapat dilakukan oleh baris perintah atau *command line* diantaranya adalah menjalankan program, modifikasi skrip, dan mengedit teks file yang digunakan untuk konfigurasi sistem atau aplikasi. Sebagian besar server melakukan boot langsung ke terminal, karena jika boot ke GUI dapat menghabiskan banyak sumber daya, dan umumnya tidak diperlukan untuk melakukan operasi berbasis server.

4.2 APLIKASI LINUX

Kernel dari sistem operasi berperan seperti *Air Traffic Controller* (ATC) pada bandara, sedangkan aplikasi berperan seperti pesawat yang berada dibawah kendali ATC. Kernel dapat menentukan berapa memori yang didapatkan oleh tiap-tiap aplikasi yang berjalan, menjalankan dan mematikan aplikasi, hingga menangani tampilan teks atau grafik pada monitor.

Aplikasi akan membuat permintaan ke kernel, dan sebagai balasannya aplikasi akan menerima sumber daya, seperti memori, CPU, dan ruang disk. Jika dua aplikasi meminta sumber daya yang sama, maka kernel dapat memutuskan mana yang diprioritaskan, bahkan dalam beberapa kasus, kernel akan mematikan aplikasi lain untuk menyimpan sisa sistem agar tidak terjadi *crash*.

Kernel juga mengabstraksi beberapa detail rumit dari aplikasi. Misalnya, aplikasi tidak mengetahui blok penyimpanannyta ada di *solid-state drive, harddisk,* atau jaringan penyimpanan bersama (seperti *cloud*). Aplikasi hanya perlu

mengikuti *Application Programming Interface* (*API*) kernel dan tidak perlu mengkhawatirkan detail implementasinya. Setiap aplikasi berperilaku seolah-olah memiliki blok memori yang besar pada sistem, kernel menerapkan ilusi ini dengan memtakan ulang blok memori yang lebih kecil, berbagi blok memori dengan aplikasi lain dan bahkan menukar blok yang tidak tersentuh ke disk.

Kernel juga menangani peralihan aplikasi, atau sebuah proses yang dikenal sebagai *multitasking*. Sistem komputer memiliki sejumlah kecil *Central Processing Unit (CPU),* dan jumlah memori yang terbatas. Kernel menangani pembongkaran satu tugas dan memuat tugas baru jika ada banyak permintaan daripada sumber daya yang tersedia. Ketika satu tugas telah berjalan untuk jangka waktu tertentu, CPU akan menjedanya agar tugas lain dapat berjalan. Jika komputer melakukan beberapa tugas sekaligus, kernel dapat memutuskan kapan harus mengalihkan fokus antar tugas. Dengan peralihan tugas yang cepat, komputer jadi terlihat melakukan banyak hal sekaligus.

Ketika kita, sebagai pengguna memikirkan kata aplikasi, kita cenderung menggambarkan aplikasi seperti mesin pengolah kata, *web browser*, dan klien email, namun, sebenarnya ada banyak jenis variasi aplikasi. Kernel tidak membedakan antara aplikasi yang berhadapan langsung ke pengguna, aplikasi layanan jaringan yang berbicara ke komputer jarak jauh, atau aplikasi tugas internal. Dari sini, kita mendapatkan abstraksi yang disebut *proses*. Sebuah proses hanyalah satu tugas yang dimuat dan dilacak oleh kernel. Sebuah aplikasi bahkan mungkin memerlukan banyak proses untuk berfungsi, jadi kernel akan menjalankan proses tersebut, memulai dan menghentikannya sesuai permintaan, dan membagikan sumber daya sistem.

4.2.1 APLIKASI UTAMA

Kernel Linux dapat menjalankan berbagai macam perangkat lunak di banyak platform perangkat keras. Komputer dapat bertindak sebagai *server*, yang artinya menangani data atas nama orang lain, atau sebagai *desktop*, yang berarti pengguna berinteraksi dengannya secara langsung. Mesin

tersebut dapat menjalankan perangkat lunak atau digunakan sebagai mesin pengembangan dalam proses pembuatan perangkat lunak. Sebuah mesin bahkan dapat mengadoption banyak peran karena Linux tidak membedakan; ini hanya masalah konfigurasi aplikasi mana yang berjalan.

Satu keuntungan yang dihasilkan adalah bahwa Linux dapat mensimulasikan hampir semua aspek lingkungan produksi, dari pengembangan hingga pengujian, hingga verifikasi pada perangkat keras yang diperkecil, yang dapat menghemat biaya dan waktu. Administrator Linux dapat menjalankan aplikasi server yang sama di desktop atau server virtual murah yang dijalankan oleh penyedia layanan internet besar. Tentu saja, desktop tidak akan dapat menangani volume yang sama seperti penyedia utama, tetapi hampir semua konfigurasi dapat disimulasikan tanpa memerlukan perangkat keras yang kuat atau lisensi server.

Perangkat lunak Linux umumnya termasuk dalam salah satu dari tiga kategori:

- **Aplikasi Server**

  Perangkat lunak yang tidak memiliki interaksi langsung dengan monitor dan keyboard dari mesin yang menjalankannya. Tujuannya adalah untuk melayani informasi ke komputer lain yang disebut *klien*. Terkadang aplikasi server mungkin tidak terhubung ke komputer lain tetapi hanya diam di sana dan mengolah data.

- **Aplikasi Desktop**

  Browser web, editor teks, pemutar musik, atau aplikasi lain yang berinteraksi langsung dengan pengguna. Dalam banyak kasus, seperti browser web, aplikasi berbicara ke server di sisi lain dan menafsirkan data. Ini adalah sisi "klien" dari aplikasi klien / server.

- **Tools**

  Perangkat lunak yang termasuk kategori bebas ini ada untuk mempermudah pengelolaan sistem komputer. Tools dapat membantu

mengkonfigurasi tampilan, menyediakan shell Linux yang digunakan pengguna untuk mengetikkan perintah, atau bahkan tools yang lebih canggih, yang disebut kompiler, yang dapat mengubah kode sumber menjadi program aplikasi yang dapat dijalankan oleh komputer.

Ketersediaan aplikasi bervariasi, tergantung distribusinya. Seringkali vendor aplikasi memilih subset dari distribusi untuk didukung. Distribusi yang berbeda memiliki versi pustaka kunci yang berbeda, dan sulit bagi perusahaan untuk mendukung semua versi yang berbeda ini. Namun, beberapa aplikasi seperti **Firefox** dan **LibreOffice** didukung secara luas dan tersedia untuk semua distribusi utama.

Komunitas Linux telah menghasilkan banyak solusi kreatif untuk aplikasi desktop dan server. Aplikasi ini, banyak di antaranya merupakan tulang punggung Internet, sangat penting untuk memahami, dan memanfaatkan kekuatan Linux.

Sebagian besar tugas komputasi dapat diselesaikan dengan sejumlah aplikasi di Linux. Ada banyak web browser, web server, database server, dan editor teks yang dapat dipilih. Mengevaluasi perangkat lunak aplikasi adalah keterampilan penting yang harus dipelajari oleh calon administrator Linux. Menentukan persyaratan untuk kinerja, stabilitas, dan biaya hanyalah beberapa pertimbangan yang diperlukan untuk analisis yang komprehensif.

4.2.2 APLIKASI SERVER

Linux unggul dalam menjalankan aplikasi server, karena keandalan dan efisiensinya. Kemampuan untuk mengoptimalkan sistem operasi server hanya dengan komponen yang diperlukan memungkinkan administrator melakukan lebih banyak hal dengan lebih sedikit, fiturnya disukai oleh perusahaan baru rintis dan juga perusahaan besar.

4.2.2.1 WEB SERVERS

Salah satu kegunaan awal Linux adalah untuk web server. Web Server menghosting konten untuk halaman web, yang dilihat oleh

browser web menggunakan **HyperText Transfer Protocol (HTTP)** atau versi terenkripsi, yaitu **HTTPS**. Halaman web itu sendiri bisa statis atau dinamis. Saat browser web meminta halaman statis, server web mengirim file seperti yang muncul di disk. Dalam kasus situs dinamis, permintaan dikirim oleh server web ke aplikasi, yang menghasilkan konten.

WordPress adalah salah satu contoh populer. Pengguna dapat mengembangkan konten melalui browser mereka di aplikasi WordPress, dan perangkat lunak mengubahnya menjadi situs web dinamis yang berfungsi penuh.

Apache adalah server web dominan yang digunakan saat ini. Apache awalnya merupakan proyek mandiri, tetapi grup tersebut telah membentuk **Apache Software Foundation** dan mengelola lebih dari seratus proyek perangkat lunak sumber terbuka. **Apache HTTPD** adalah daemon, atau program aplikasi server, yang "melayani" permintaan halaman web.

Web server lain adalah **NGINX** , yang berbasis di Rusia. NGINX berfokus pada kinerja dengan memanfaatkan kernel UNIX yang lebih modern dan hanya melakukan sebagian dari apa yang dapat dilakukan Apache. Lebih dari 65% situs web didukung oleh NGINX atau Apache.

4.2.2.2 PRIVATE CLOUD SERVERS

Saat individu, organisasi, dan perusahaan mulai memindahkan data mereka ke cloud, ada permintaan yang meningkat untuk perangkat lunak server cloud pribadi yang dapat diterapkan dan dikelola secara internal.

Proyek **ownCloud** diluncurkan pada tahun 2010 oleh Frank Karlitschek untuk menyediakan perangkat lunak untuk menyimpan, menyinkronkan, dan berbagi data dari server cloud pribadi. Ini

tersedia dalam lisensi open source GNU AGPLv3 standar dan versi perusahaan yang membawa lisensi komersial.

Proyek **Nextcloud** bercabang dari ownCloud pada tahun 2016 oleh Karlitschek dan telah berkembang dengan hebat sejak saat itu. Ini disediakan di bawah GNU AGPLv3 dan bertujuan untuk "proses pengembangan yang terbuka dan transparan."

Kedua proyek fokus pada penyediaan perangkat lunak cloud pribadi yang memenuhi kebutuhan organisasi besar dan kecil yang memerlukan keamanan, privasi, dan kepatuhan terhadap peraturan. Sementara beberapa proyek lain bertujuan untuk melayani pengguna yang sama, keduanya sejauh ini merupakan yang terbesar dalam hal penerapan dan anggota proyek.

4.2.2.3 DATABASE SERVERS

Aplikasi server database merupakan tulang punggung sebagian besar layanan online. Aplikasi web dinamis menarik data dari dan menulis data ke aplikasi ini. Misalnya, program web untuk melacak siswa online mungkin terdiri dari server front-end yang menyajikan formulir web. Ketika data dimasukkan ke dalam formulir, itu ditulis ke aplikasi database seperti **MariaDB** . Ketika instruktur perlu mengakses informasi siswa, aplikasi web menanyakan database dan mengembalikan hasilnya melalui aplikasi web.

MariaDB adalah fork yang dikembangkan komunitas dari sistem manajemen database relasional **MySQL** . Ini hanyalah salah satu dari banyak server database yang digunakan untuk pengembangan web karena persyaratan yang berbeda menentukan aplikasi terbaik untuk tugas yang diperlukan.

Basis data menyimpan informasi dan juga memungkinkan pengambilan dan kueri yang mudah. Beberapa database populer lainnya adalah **Firebird** dan **PostgreSQL** . Anda dapat memasukkan

angka penjualan mentah ke dalam database dan kemudian menggunakan bahasa yang disebut **Structured Query Language (SQL)** untuk menggabungkan penjualan berdasarkan produk dan tanggal untuk menghasilkan laporan.

4.2.2.4 EMAIL SERVERS

Email selalu digunakan secara luas untuk server Linux. Saat membahas server email, selalu bermanfaat untuk melihat 3 tugas berbeda yang diperlukan untuk mendapatkan email di antara orang-orang:

- **Agen Transfer Surat (MTA)**

  MTA (perangkat lunak yang digunakan untuk mentransfer pesan elektronik ke sistem lain) yang paling terkenal adalah **Sendmail** . **Postfix** adalah salah satu yang populer dan bertujuan untuk menjadi lebih sederhana dan lebih aman daripada Sendmail.

- **Agen Pengiriman Surat (MDA)**

  Juga disebut **Agen Pengiriman Lokal** , ini menangani penyimpanan email di kotak surat pengguna. Biasanya dipanggil dari MTA terakhir dalam rangkaian.

- **Server POP / IMAP**

  The **Post Office Protocol (POP)** dan **Internet Message Access Protocol (IMAP)** adalah dua protokol komunikasi yang memungkinkan klien email yang berjalan pada pembicaraan komputer Anda ke server jauh untuk mengambil email.

Dovecot adalah server POP / IMAP yang populer karena kemudahan penggunaan dan perawatannya yang rendah. **Cyrus IMAP** adalah option lain. Beberapa server POP / IMAP menerapkan format basis data emailnya sendiri untuk kinerja dan menyertakan MDA jika basis data khusus diinginkan. Orang yang menggunakan

format file standar (seperti semua email dalam satu file teks) dapat memilih MDA apa pun.

Ada beberapa perbedaan signifikan antara dunia perangkat lunak sumber tertutup dan sumber terbuka, salah satunya adalah penyertaan proyek lain sebagai komponen proyek atau paket. Dalam dunia sumber tertutup, **Microsoft Exchange** dikirim terutama sebagai paket perangkat lunak / suite yang mencakup semua komponen yang diperlukan atau disetujui, semuanya dari Microsoft, jadi hanya ada sedikit jika ada option untuk membuat pilihan individual. Di dunia open source, banyak option dapat dimasukkan secara modular atau ditukar dengan komponen paket, dan memang beberapa paket perangkat lunak atau suite hanyalah kumpulan yang dikemas dengan baik dari komponen individual yang semuanya bekerja sama secara harmonis.

4.2.2.5 FILE SHARING

Untuk berbagi file yang berpusat pada Windows, **Samba** adalah pemenang yang jelas. Samba memungkinkan mesin Linux untuk terlihat dan berperilaku seperti mesin Windows sehingga dapat berbagi file dan berpartisipasi dalam domain Windows. Samba mengimplementasikan komponen server, seperti membuat file tersedia untuk berbagi dan peran server Windows tertentu, dan juga klien berakhir sehingga mesin Linux dapat menggunakan berbagi file Windows.

Proyek **Netatalk** memungkinkan mesin Linux bekerja sebagai server file Apple Macintosh. Protokol berbagi file asli untuk UNIX / Linux disebut **Sistem File Jaringan (NFS)** . NFS biasanya merupakan bagian dari kernel yang berarti bahwa sistem file jarak jauh dapat dipasang (dibuat dapat diakses) seperti disk biasa, membuat akses file transparan ke aplikasi lain.

Ketika jaringan komputer menjadi lebih substansial, kebutuhan akan direktori meningkat. Salah satu sistem direktori jaringan tertua adalah **Domain Name System (DNS)** . Ini digunakan untuk mengubah nama seperti https://www.icann.org/ menjadi alamat IP seperti 192.0.43.7, yang merupakan pengenal unik dari sebuah komputer di Internet. DNS juga menyimpan informasi global seperti alamat MTA untuk nama domain tertentu. Sebuah organisasi mungkin ingin menjalankan server DNS mereka sendiri untuk menghosting nama publik mereka, dan juga berfungsi sebagai direktori layanan internal. The **Internet Software Consortium** mempertahankan sebagian DNS server yang populer, hanya disebut *mengikat* setelah nama dari proses yang berjalan layanan.

DNS difokuskan terutama pada nama komputer dan alamat IP dan tidak mudah dicari. Direktori lain bermunculan untuk menyimpan informasi seperti akun pengguna dan peran keamanan. The **Lightweight Directory Access Protocol (LDAP)** adalah salah satu sistem direktori umum yang juga kekuatan Microsoft Active Directory. Dalam LDAP, sebuah objek disimpan dalam sebuah pohon, dan posisi objek tersebut pada pohon dapat digunakan untuk memperoleh informasi tentang objek tersebut dan apa yang disimpannya. Misalnya, administrator Linux dapat disimpan di cabang pohon yang disebut "Departemen TI," yang berada di bawah cabang yang disebut "Operasi." Dengan demikian seseorang dapat menemukan semua staf teknis dengan mencari di bawah cabang "Departemen TI". **OpenLDAP** adalah program dominan yang digunakan dalam infrastruktur Linux.

Satu bagian terakhir dari infrastruktur jaringan yang dibahas di sini disebut **Dynamic Host Configuration Protocol (DHCP)** . Saat komputer melakukan boot, ia membutuhkan alamat IP untuk jaringan lokal agar

dapat diidentifikasi secara unik. Tugas DHCP adalah mendengarkan permintaan dan menetapkan alamat gratis dari kumpulan DHCP. Konsorsium Perangkat Lunak Internet juga mengelola server **ISC DHCP** , yang merupakan server DHCP sumber terbuka paling umum.

4.2.3 APLIKASI DESKTOP

Ekosistem Linux memiliki berbagai macam aplikasi desktop. Ada permainan, aplikasi produktivitas, alat kreatif, browser web, dan banyak lagi.

4.2.3.1 EMAIL

Mozilla Foundation keluar dengan **Thunderbird** , klien email desktop berfitur lengkap. Thunderbird terhubung ke server POP atau IMAP, menampilkan email secara lokal, dan mengirim email melalui server SMTP eksternal.

Klien email penting lainnya adalah **Evolution** dan **KMail** yang merupakan klien email proyek GNOME dan KDE. Standarisasi melalui POP dan IMAP dan format email lokal berarti mudah untuk beralih di antara klien email tanpa kehilangan data.

4.2.3.2 CREATIVE

Untuk jenis kreatif, ada **Blender** , **GIMP (GNU Image Manipulation Program)** , dan **Audacity** yang masing-masing menangani pembuatan film 3D, manipulasi gambar 2D, dan pengeditan audio. Mereka memiliki berbagai tingkat kesuksesan di pasar profesional. Blender digunakan untuk segala hal mulai dari film independen hingga film Hollywood, misalnya. GIMP mendukung manipulasi foto berkualitas tinggi, kreasi karya seni asli, elemen desain grafis, dan dapat dikembangkan melalui skrip dalam berbagai bahasa. Audacity adalah alat pengeditan audio gratis dan open source yang tersedia di beberapa sistem operasi.

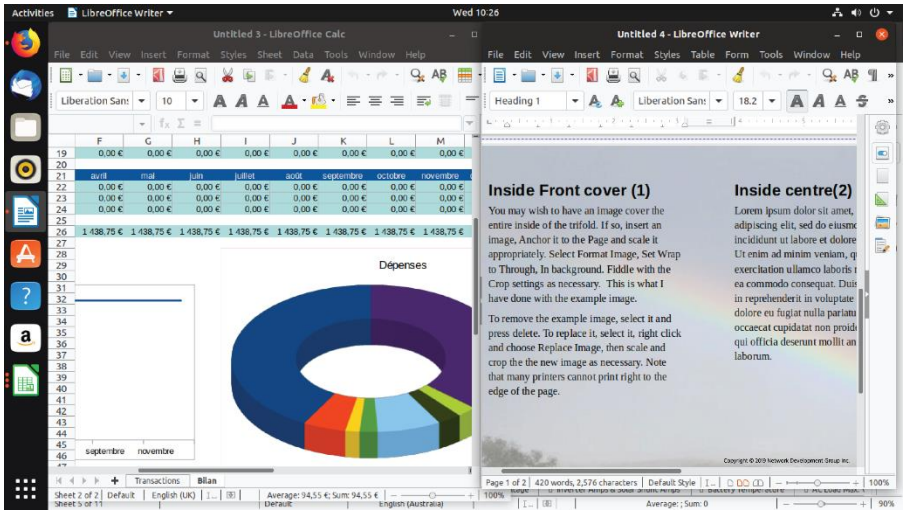
4.2.3.3 PRODUCTIVITY

Penggunaan aplikasi open source yang umum dalam presentasi dan proyek adalah salah satu cara untuk memperkuat keterampilan Linux. Aplikasi produktivitas dasar, seperti pengolah kata, spreadsheet, dan paket presentasi adalah aset berharga. Secara kolektif mereka dikenal sebagai *office suite* , terutama karena Microsoft Office, pemain dominan di pasar.

LibreOffice adalah cabang dari rangkaian aplikasi **OpenOffice** (terkadang disebut **OpenOffice.org**). Keduanya menawarkan office suite lengkap, termasuk alat yang mengupayakan kompatibilitas dengan Microsoft Office dalam fitur dan format file.

Di bawah ini adalah spreadsheet dan editor dokumen LibreOffice. Perhatikan bagaimana spreadsheet, **LibreOffice Calc** , tidak terbatas pada baris dan kolom angka. Angka-angka dapat menjadi sumber grafik, dan rumus dapat ditulis untuk menghitung nilai berdasarkan informasi, seperti menggabungkan suku bunga dan jumlah pinjaman untuk membantu membandingkan option pinjaman yang berbeda.

Menggunakan **LibreOffice Writer** , dokumen dapat berisi teks, grafik, tabel data, dan banyak lagi. Anda dapat menautkan dokumen dan spreadsheet bersama-sama, misalnya, sehingga Anda dapat meringkas data dalam bentuk tertulis dan mengetahui bahwa setiap perubahan pada spreadsheet akan tercermin di dokumen.



LibreOffice juga dapat bekerja dengan format file lain, seperti file Microsoft Office atau **Adobe Portable Document Format (PDF)** . Selain itu, melalui penggunaan ekstensi, LibreOffice dapat dibuat berintegrasi dengan perangkat lunak Wiki untuk memberi Anda solusi intranet yang kuat.

4.2.3.4 WEB BROWSER

Linux adalah warga negara kelas satu untuk **Mozilla Firefox** dan browser **Google Chrome** . Keduanya adalah peramban web sumber terbuka yang cepat, kaya fitur, dan memiliki dukungan luar biasa untuk pengembang web. Paket-paket ini adalah contoh yang sangat baik tentang bagaimana kompetisi membantu mendorong pengembangan open source - peningkatan yang dilakukan pada satu browser memacu pengembangan browser lainnya. Hasilnya, Internet memiliki dua peramban luar biasa yang melampaui batas dari apa yang dapat dilakukan di web, dan bekerja di berbagai platform. Menggunakan browser, meskipun merupakan kebiasaan bagi banyak orang, dapat menyebabkan masalah privasi. Dengan memahami dan memodifikasi option konfigurasi, seseorang dapat membatasi jumlah informasi yang mereka bagikan saat menelusuri web dan menyimpan konten.

4.3 CONSOLE TOOLS

Secara historis, pengembangan UNIX menunjukkan tumpang tindih yang cukup besar antara keterampilan pengembangan perangkat lunak dan administrasi sistem. Alat untuk mengelola sistem memiliki fitur bahasa komputer seperti loop (yang memungkinkan perintah dilakukan berulang kali), dan beberapa bahasa pemrograman komputer digunakan secara ekstensif dalam mengotomatiskan tugas administrasi sistem. Dengan demikian, seseorang harus mempertimbangkan keterampilan ini sebagai pelengkap, dan setidaknya keakraban dasar dengan pemrograman diperlukan untuk administrator sistem yang kompeten.

4.3.1 SHELLS

Pada tingkat dasar, pengguna berinteraksi dengan sistem Linux melalui *shell* apakah menghubungkan ke sistem dari jarak jauh atau dari keyboard yang terpasang. Tugas shell adalah menerima perintah, seperti manipulasi file dan memulai aplikasi, dan meneruskannya ke kernel Linux untuk dieksekusi. Linux shell menyediakan bahasa yang kaya untuk iterasi file dan menyesuaikan lingkungan, semua tanpa meninggalkan shell. Misalnya, dimungkinkan untuk menulis satu baris perintah yang menemukan file dengan konten yang cocok dengan pola tertentu, mengekstrak informasi berguna dari file, lalu menyalin informasi baru ke file baru.

Linux menawarkan berbagai shell untuk dipilih, sebagian besar berbeda dalam cara dan apa yang dapat disesuaikan, dan sintaks bahasa skrip bawaan. Dua keluarga utama adalah **Bourne shell** dan **C shell** . Bourne shell dinamai menurut penciptanya Stephen Bourne dari Bell Labs. Shell C dinamai demikian karena sintaksnya banyak meminjam dari bahasa C. Karena kedua cangkang ini ditemukan pada 1970-an, ada versi yang lebih modern, **Bourne Again Shell (Bash)** dan **tcsh** (diucapkan sebagai tee-cee-shell). Bash adalah shell default pada kebanyakan sistem, meskipun tcsh biasanya juga tersedia.

Programmer telah mengambil fitur favorit dari Bash dan tcsh dan membuat shell lain, seperti **shell Korn (ksh)** dan **shell Z (zsh)** . Pilihan cangkang sebagian besar adalah pilihan pribadi; pengguna yang merasa nyaman dengan Bash dapat beroperasi secara efektif di sebagian besar sistem Linux. Shells lain mungkin menawarkan fitur yang meningkatkan produktivitas dalam kasus penggunaan tertentu.

4.3.2 TEXT EDITOR

Sebagian besar sistem Linux menyediakan pilihan editor teks yang biasanya digunakan di konsol untuk mengedit file konfigurasi. Dua aplikasi utama adalah **Vi** (atau **Vim yang** lebih modern) dan **Emacs** . Keduanya adalah alat yang sangat kuat untuk mengedit file teks; mereka berbeda dalam format perintah dan bagaimana plugin ditulis untuk mereka. Plugin dapat berupa apa saja, mulai dari penyorotan sintaks proyek perangkat lunak hingga kalender terintegrasi.

Baik Vi dan Emacs rumit dan memiliki kurva pembelajaran yang curam, yang tidak membantu untuk pengeditan sederhana dari file teks kecil. Oleh karena itu, **Pico** dan **Nano** tersedia di sebagian besar sistem dan menyediakan pengeditan teks yang sangat dasar.

> Perlu diperhatikan!
> Nano editor dikembangkan sebagai editor open source sepenuhnya yang didasarkan pada Pico, karena lisensi untuk Pico bukanlah lisensi open source dan melarang melakukan perubahan dan mendistribusikannya.

Meskipun Nano sederhana dan mudah digunakan, Nano tidak menawarkan rangkaian lengkap fitur pengeditan dan pengikatan kunci yang lebih canggih seperti yang dilakukan editor seperti Vi. Administrator harus berusaha keras untuk mendapatkan pengetahuan dasar dengan Vi, karena Vi tersedia di hampir semua sistem Linux yang ada. Saat memulihkan sistem Linux yang rusak dengan menjalankan mode pemulihan distribusi, Vi dapat menjadi alat yang penting, dan waktu terbaik untuk mempelajari Vim atau

editor apa pun adalah sebelum Anda sangat membutuhkannya untuk memperbaiki sistem yang rusak.

4.4 PAKET MANAJEMEN

Setiap sistem Linux perlu menambah, menghapus, dan memperbarui perangkat lunak. Di masa lalu, ini berarti mengunduh kode sumber, menyiapkannya, menyusunnya, dan menyalin file ke setiap sistem yang memerlukan pembaruan. Untungnya, distribusi modern menggunakan *paket*, yang merupakan file terkompresi yang menggabungkan aplikasi dan *dependensinya* (atau file yang diperlukan), sangat menyederhanakan instalasi dengan membuat direktori yang tepat, menyalin file yang tepat ke dalamnya, dan membuat item yang diperlukan seperti tautan simbolik.

Seorang *manajer paket* menangani melacak file mana yang milik paket mana dan bahkan men-download pembaruan dari repositori, biasanya server berbagi jauh keluar update sesuai untuk distribusi. Di Linux, ada banyak sistem manajemen paket perangkat lunak yang berbeda, tetapi dua yang paling populer adalah yang berasal dari Debian dan Red Hat.

4.4.1 PAKET MANAJEMEN DEBIAN

Distribusi Debian, dan turunannya seperti Ubuntu dan Mint, menggunakan sistem manajemen paket Debian. Inti dari manajemen paket Debian adalah paket perangkat lunak yang didistribusikan sebagai file yang diakhiri dengan `.deb` ekstensi.

Tool tingkat terendah untuk mengelola file-file ini adalah perintah `dpkg`. Perintah ini bisa jadi rumit untuk pengguna Linux pemula, jadi **Advance Package Tool**, `apt-get`(program front-end tool `dpkg`), membuat pengelolaan paket lebih mudah. Alat baris perintah tambahan yang berfungsi sebagai front-end untuk `dpkg` disertakan `aptitude` dan front-end GUI seperti **Synaptic** dan **Software Center**.

4.4.2 PAKET MANAJEMEN RPM

Linux Standards Base, yang merupakan proyek **Linux Foundation**, dirancang untuk menentukan (melalui konsensus) satu set standar yang meningkatkan kompatibilitas antara sesuai sistem Linux. Menurut Linux Standards Base, sistem manajemen paket standar adalah RPM.

RPM menggunakan `.rpm` file untuk setiap paket perangkat lunak. Sistem ini adalah distribusi yang diturunkan dari Red Hat, termasuk Centos dan Fedora, digunakan untuk mengelola perangkat lunak. Beberapa distro lain yang bukan turunan Red Hat, seperti SUSE, OpenSUSE, dan Arch, juga menggunakan RPM.

Seperti sistem Debian, sistem Manajemen Paket RPM melacak ketergantungan antar paket. Ketergantungan pelacakan memastikan bahwa ketika paket diinstal, sistem juga menginstal paket apa pun yang diperlukan oleh paket itu agar berfungsi dengan benar. Dependensi juga memastikan bahwa pembaruan dan penghapusan perangkat lunak dilakukan dengan benar.

Tools back-end yang paling umum digunakan untuk Manajemen Paket RPM adalah perintah `rpm`. Meskipun perintah `rpm` dapat menginstal, memperbarui, meminta, dan menghapus paket, alat front-end baris perintah seperti `yum` dan `up2date` mengotomatiskan proses penyelesaian masalah ketergantungan.

> Catatan:
> Program atau aplikasi back-end berinteraksi langsung dengan program front-end atau "disebut" oleh program perantara. Program back end tidak akan berinteraksi langsung dengan pengguna. Pada dasarnya ada program yang berinteraksi dengan orang (front-end) dan program yang berinteraksi dengan program lain (back-end).

Ada juga tools front-end berbasis GUI seperti **Yumex** dan **Gnome PackageKit** yang juga membuat manajemen paket RPM lebih mudah.

Beberapa distribusi berbasis RPM telah menerapkan gaya manajemen paket **ZYpp** (atau **libzypp**), kebanyakan openSUSE dan SUSE Linux Enterprise, tetapi juga distribusi seluler MeeGo, Tizen dan Sailfish.

Perintah `zypper` adalah dasar dari metode zypp, dan fitur pendek dan panjang English perintah untuk melakukan fungsi, seperti yang menginstal paket termasuk dependensi yang diperlukan. `zypper in` *`packagename`*

Sebagian besar perintah yang terkait dengan manajemen paket memerlukan hak akses root. Aturan praktisnya adalah jika perintah mempengaruhi status paket, akses administratif diperlukan. Dengan kata lain, pengguna biasa dapat melakukan kueri atau penelusuran, tetapi untuk menambah, memperbarui, atau menghapus paket, perintah tersebut harus dijalankan sebagai pengguna root.

4.5 DEVELOPMENT LANGUAGES

Tidaklah mengherankan bahwa sebagai perangkat lunak yang dibangun atas kontribusi dari pemrogram, Linux memiliki dukungan yang sangat baik untuk pengembangan perangkat lunak. Shells dibuat agar dapat diprogram, dan ada editor hebat yang disertakan di setiap sistem. Ada juga banyak alat pengembangan yang tersedia, dan banyak bahasa pemrograman modern memperlakukan Linux sebagai warga negara kelas satu.

Bahasa pemrograman komputer menyediakan cara bagi programmer untuk memasukkan instruksi dalam format yang lebih dapat dibaca manusia, dan instruksi tersebut pada akhirnya diterjemahkan ke dalam sesuatu yang dimengerti komputer. Bahasa termasuk dalam salah satu dari dua kubu: *diinterpretasikan* atau *dikompilasi*. Sebuah *bahasa ditafsirkan* menerjemahkan kode yang ditulis dalam kode komputer sebagai program berjalan, dan *bahasa yang dikompilasi* diterjemahkan sekaligus.

Linux sendiri ditulis dalam bahasa yang dikompilasi disebut **C**. Manfaat utama C adalah bahasanya itu sendiri memetakan secara dekat dengan kode mesin yang dihasilkan sehingga programmer yang terampil dapat menulis kode yang kecil dan efisien. Ketika memori komputer diukur dalam kilobyte, ini sangat penting. Bahkan dengan ukuran memori yang besar saat ini, C tetap berguna untuk menulis kode yang harus berjalan cepat, seperti sistem operasi.

C telah diperpanjang selama bertahun-tahun. Ada **C++**, yang menambahkan dukungan objek ke C (gaya pemrograman yang berbeda), dan **Objective C** yang mengambil arah lain dan banyak digunakan di produk Apple.

Bahasa **Java** menempatkan putaran berbeda pada pendekatan terkompilasi. Alih-alih mengompilasi ke kode mesin, Java pertama-tama membayangkan CPU hipotetis yang disebut **Java Virtual Machine (JVM)** dan kemudian mengkompilasi semua kode ke sana. Setiap komputer host kemudian menjalankan perangkat lunak JVM untuk menerjemahkan instruksi JVM (disebut bytecode) ke dalam instruksi asli.

Terjemahan tambahan dengan Java mungkin membuat Anda berpikir itu akan lambat. Namun, JVM relatif sederhana sehingga dapat diimplementasikan dengan cepat dan andal pada apa pun mulai dari komputer yang kuat hingga perangkat berdaya rendah yang terhubung ke televisi. File Java yang dikompilasi juga dapat dijalankan di komputer mana pun yang mengimplementasikan JVM!

Manfaat lain dari kompilasi ke target perantara adalah bahwa JVM dapat menyediakan layanan ke aplikasi yang biasanya tidak tersedia di CPU. Mengalokasikan memori ke program adalah masalah yang kompleks, tetapi itu dibangun ke dalam JVM. Hasilnya, pembuat JVM dapat memfokuskan peningkatan mereka pada JVM secara keseluruhan, sehingga kemajuan apa pun yang mereka buat akan langsung tersedia untuk aplikasi.

Bahasa yang ditafsirkan, di sisi lain, diterjemahkan ke kode mesin saat dijalankan. Daya ekstra komputer yang dihabiskan untuk melakukan ini sering kali dapat dipulihkan dengan peningkatan produktivitas yang diperoleh programmer

dengan tidak harus berhenti bekerja untuk mengkompilasi. Bahasa yang ditafsirkan juga cenderung menawarkan lebih banyak fitur daripada bahasa yang dikompilasi, yang berarti bahwa kode yang dibutuhkan seringkali lebih sedikit. Penerjemah bahasa itu sendiri biasanya ditulis dalam bahasa lain seperti C, dan terkadang bahkan Java! Ini berarti bahwa bahasa yang ditafsirkan sedang dijalankan di JVM, yang diterjemahkan pada waktu proses menjadi kode mesin yang sebenarnya.

JavaScript adalah bahasa pemrograman dengan interpretasi tingkat tinggi yang merupakan salah satu teknologi inti di world wide web. Ini mirip tetapi secara

> Perlu diperhatikan!
> Istilah *berorientasi objek* mengacu pada pemrograman yang mengabstraksi tindakan dan proses kompleks sehingga pengguna akhir hanya berurusan dengan tugas-tugas dasar. Untuk memvisualisasikan konsep ini, bayangkan sebuah mesin yang melakukan serangkaian tugas kompleks hanya dengan menekan sebuah tombol.

fundamental berbeda dari Java, yang merupakan bahasa pemrograman berorientasi objek yang dimiliki oleh Oracle. JavaScript adalah bahasa skrip lintas platform untuk menambahkan elemen interaktif ke halaman web, yang digunakan secara luas di seluruh internet. Dengan menggunakan pustaka JavaScript, pemrogram web dapat menambahkan semuanya mulai dari animasi sederhana hingga aplikasi sisi server yang kompleks untuk pengguna internet. JavaScript terus berkembang untuk memenuhi kebutuhan fungsionalitas dan keamanan pengguna internet dan mampu dirilis di bawah Lisensi GNU GPL.

Perl adalah bahasa yang ditafsirkan. Perl pada awalnya dikembangkan untuk melakukan manipulasi teks. Selama bertahun-tahun, itu disukai oleh administrator sistem dan terus ditingkatkan dan digunakan dalam segala hal mulai dari otomatisasi hingga membangun aplikasi web.

PHP adalah bahasa yang awalnya dibangun untuk membuat halaman web dinamis. File PHP dibaca oleh server web seperti Apache. Tag khusus dalam file menunjukkan bahwa bagian kode harus ditafsirkan sebagai instruksi. Server web menarik semua bagian file yang berbeda bersama-sama dan mengirimkannya ke

browser web. Keunggulan utama PHP adalah mudah dipelajari dan tersedia di hampir semua sistem. Karena itu, banyak proyek populer dibangun di atas PHP. Contoh penting termasuk WordPress (untuk blogging), cacti (untuk pemantauan), dan bahkan bagian dari Facebook.

Ruby adalah bahasa lain yang dipengaruhi oleh Perl dan Shell, bersama dengan banyak bahasa lainnya. Itu membuat tugas pemrograman yang kompleks menjadi relatif mudah, dan dengan dimasukkannya shellska kerja Ruby on Rails, adalah pilihan populer untuk membangun aplikasi web yang kompleks. Ruby juga merupakan bahasa yang mendukung banyak alat otomatisasi terkemuka seperti **Chef** dan **Puppet** , yang membuat pengelolaan sejumlah besar sistem Linux menjadi lebih sederhana.

Python adalah bahasa skrip lain yang umum digunakan. Mirip seperti Ruby, ini membuat tugas yang rumit menjadi lebih mudah dan memiliki shellska kerja yang disebut **Django** yang membuat pembuatan aplikasi web menjadi sangat mudah. Python memiliki kemampuan pemrosesan statistik yang sangat baik dan menjadi favorit di dunia akademis.

Bahasa pemrograman komputer hanyalah alat yang membuatnya lebih mudah untuk memberi tahu komputer apa yang ingin Anda lakukan. Pustaka menggabungkan tugas-tugas umum ke dalam paket berbeda yang dapat digunakan oleh pengembang. **ImageMagick** adalah salah satu pustaka yang memungkinkan pemrogram memanipulasi gambar dalam kode. ImageMagick juga dilengkapi dengan beberapa alat baris perintah yang memungkinkan pemrogram memproses gambar dari shell dan memanfaatkan kemampuan skrip di sana.

OpenSSL adalah pustaka kriptografi yang digunakan dalam segala hal mulai dari server web hingga baris perintah. Ini menyediakan antarmuka standar untuk menambahkan kriptografi ke dalam skrip Perl, misalnya.

Pada tingkat yang lebih rendah banyak adalah **C library** . Library C menyediakan serangkaian fungsi dasar untuk membaca dan menulis ke file dan tampilan, dan digunakan oleh aplikasi dan bahasa lain.

4.6 SECURITY

Administrator dan pengguna komputer semakin menyadari masalah privasi baik dalam kehidupan pribadi maupun profesional mereka. Pelanggaran data profil tinggi telah menjadi berita terlalu sering baru-baru ini, dan biaya pembobolan ini dapat mencapai jutaan dolar untuk lembaga yang menjadi korban peretas dan serangan ransomware. Sering kali penyebab pelanggaran ini hanyalah kesalahan manusia seperti membuka email yang mencurigakan atau memasukkan kata sandi ke halaman login palsu.

Cookies adalah mekanisme utama yang digunakan situs web untuk melacak Anda. Terkadang pelacakan ini bagus, seperti untuk melacak apa yang ada di keranjang belanja Anda atau membuat Anda tetap masuk saat kembali ke situs.

Saat Anda menjelajahi web, server web dapat mengirim kembali cookie, yang merupakan sepotong kecil teks, bersama dengan halaman web. Browser Anda menyimpan informasi ini dan mengirimkannya kembali dengan setiap permintaan ke situs yang sama. Cookie biasanya hanya dikirim kembali ke situs asalnya, jadi cookie dari example.com tidak akan dikirim ke example.org.

Namun, banyak situs telah menyematkan skrip yang berasal dari pihak ketiga, seperti iklan spanduk atau piksel analitik Google. Jika example.com dan example.org memiliki piksel pelacakan, seperti dari pengiklan, cookie yang sama tersebut akan dikirim saat menjelajahi kedua situs. Pengiklan kemudian mengetahui bahwa Anda telah mengunjungi example.com dan example.org.

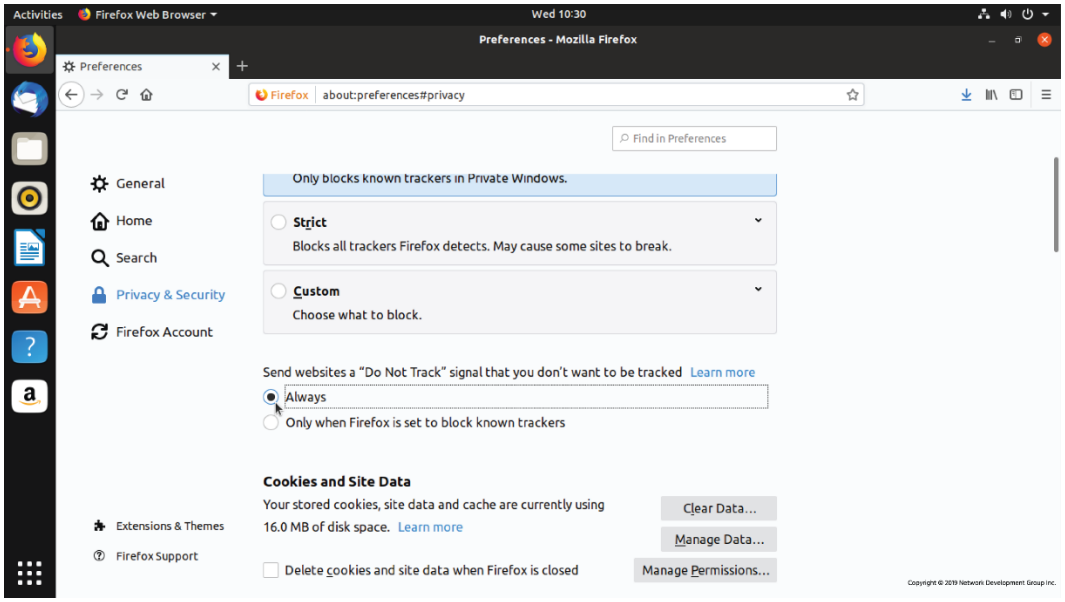
Dengan jangkauan yang cukup luas, seperti penempatan di situs jejaring sosial dengan tombol "Suka" dan semacamnya, situs web dapat memperoleh pemahaman tentang situs web mana yang sering Anda kunjungi dan mengetahui minat dan demografi Anda.

Ada berbagai strategi untuk mengatasi hal ini. Salah satunya adalah mengabaikannya. Cara lainnya adalah membatasi piksel pelacakan yang Anda

terima, baik dengan memblokirnya sepenuhnya atau menghapusnya secara berkala.

Browser biasanya menawarkan pengaturan terkait cookie; pengguna dapat memilih agar browser memberi tahu situs untuk tidak melacak. Tag sukarela ini dikirim dalam permintaan, dan beberapa situs akan menghormatinya. Browser juga dapat disetel untuk tidak pernah mengingat cookie pihak ketiga dan menghapus cookie biasa (seperti dari situs yang Anda jelajahi) setelah ditutup.

Mengubah pengaturan privasi dapat membuat Anda menjadi lebih anonim di Internet, tetapi juga dapat menyebabkan masalah pada beberapa situs yang bergantung pada cookie pihak ketiga. Jika ini terjadi, Anda mungkin harus secara eksplisit mengizinkan beberapa cookie untuk disimpan.



Browser juga menawarkan mode *pribadi* atau *penyamaran di* mana cookie dan piksel pelacakan dihapus saat keluar dari jendela. Mode ini dapat berguna jika Anda ingin mencari sesuatu tanpa memberi tahu situs web lain apa yang Anda cari.

4.6.1 PASSWORD ISSUES

Manajemen kata sandi yang baik sangat penting untuk keamanan dalam lingkungan komputasi apa pun. Administrator sistem Linux sering kali merupakan orang yang bertanggung jawab untuk menetapkan dan menegakkan kebijakan sandi untuk pengguna di semua tingkatan. Pengguna

yang paling berhak di sistem Linux adalah *root* ; akun ini adalah *administrator* utama dan dibuat saat sistem operasi diinstal. Seringkali administrator akan menonaktifkan akses root sebagai garis pertahanan pertama melcloud intrusi karena peretas komputer akan mencoba mendapatkan akses root untuk mengambil kendali sistem.

Ada banyak tingkat akses dan berbagai cara untuk mengatur kata sandi pada sistem Linux. Saat pengguna dibuat, mereka diberikan izin masuk yang berbeda tergantung pada grup mana mereka ditugaskan. Misalnya, administrator dapat membuat dan mengelola pengguna sedangkan pengguna biasa tidak bisa. Layanan yang berjalan pada sistem seperti database juga dapat memiliki izin masuk dengan kata sandi dan hak istimewanya sendiri. Selain itu, ada kata sandi khusus untuk mengakses sistem dari jarak jauh melalui SSH, FTP, atau program manajemen lainnya.

Mengelola semua akun ini, dan kata sandi yang menyertainya adalah bagian yang rumit dan perlu dari peran administrator sistem. Kata sandi harus cukup rumit agar tidak mudah ditebak oleh peretas, namun mudah diingat oleh pengguna. Semakin banyak pengguna dan administrator yang beralih ke program *pengelola kata sandi* untuk menyimpan kredensial login dalam bentuk terenkripsi. Tren lainnya adalah *otentikasi dua faktor (2FA)* , teknik di mana kata sandi dilengkapi dengan "faktor" kedua, sering kali berupa kode sandi yang dikirim ke ponsel pengguna atau perangkat lain. Mengikuti tren keamanan saat ini, sambil memastikan kemudahan akses pengguna yang sah, merupakan tantangan berkelanjutan yang harus dipenuhi.

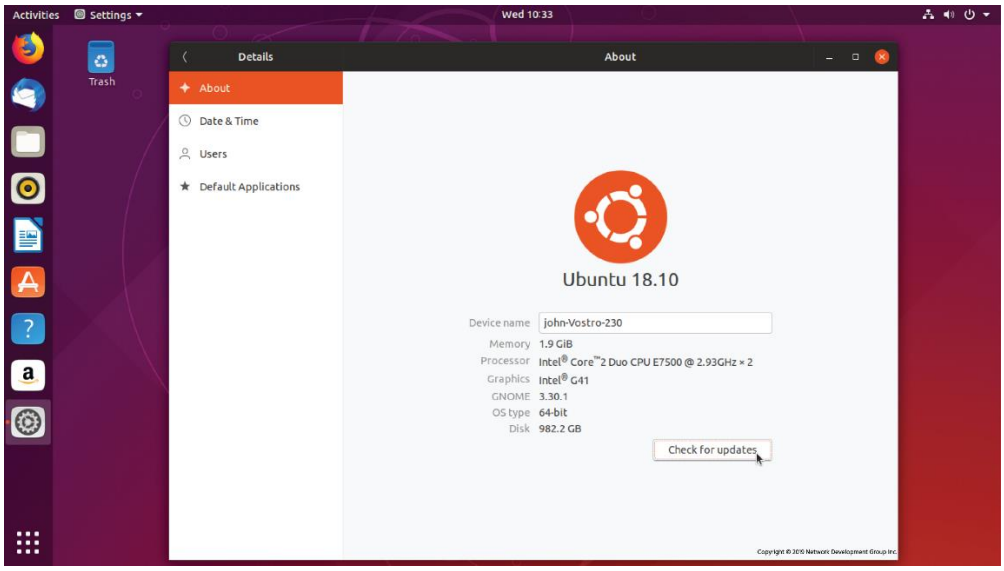
4.6.2 PROTECTING YOURSELF

Saat Anda menjelajahi web, Anda meninggalkan jejak digital. Banyak dari informasi ini diabaikan; beberapa di antaranya dikumpulkan untuk mengumpulkan statistik untuk iklan, dan beberapa dapat digunakan untuk tujuan jahat.

Hal termudah yang dapat Anda lakukan adalah menggunakan kata sandi yang bagus dan unik ke mana pun Anda pergi, terutama di komputer lokal Anda. Kata sandi yang baik setidaknya terdiri dari 10 karakter dan berisi campuran angka, huruf (huruf besar dan kecil), dan simbol khusus. Gunakan *pengelola kata sandi* seperti **KeePassX** untuk membuat kata sandi, lalu Anda hanya perlu memiliki kata sandi masuk ke mesin Anda dan kata sandi untuk membuka file KeePassX Anda.

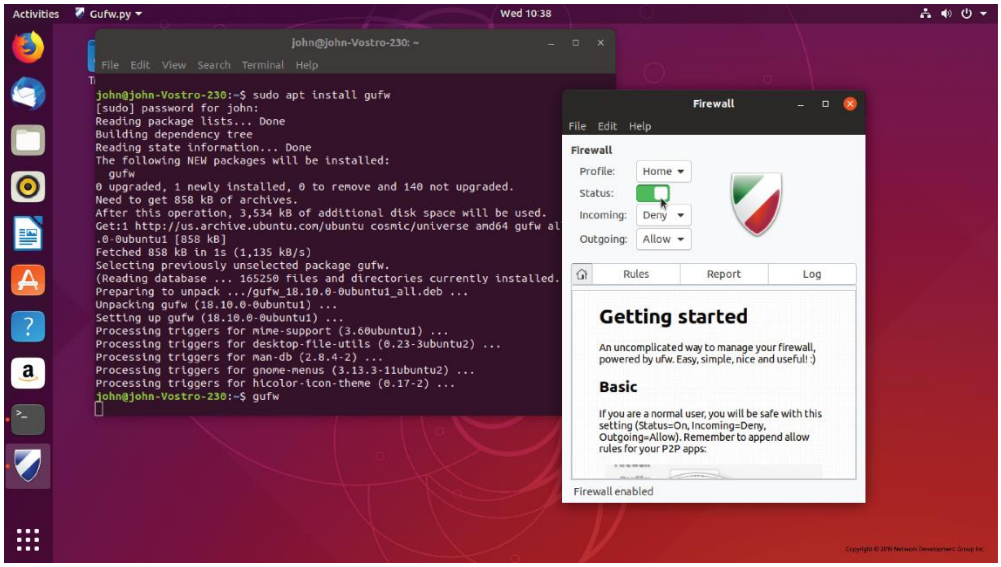
Selain itu, batasi informasi yang Anda berikan ke situs hanya pada yang diperlukan. Meskipun memberikan nama gadis dan tanggal lahir ibu Anda dapat membantu membuka kunci login jejaring sosial Anda jika Anda kehilangan kata sandi, informasi yang sama dapat digunakan untuk menyamar sebagai Anda di bank.

Setelah itu, lakukan pengecekan pembaruan secara berkala. Sistem dapat dikonfigurasi untuk memeriksa pembaruan secara teratur. Jika ada pembaruan terkait keamanan, Anda mungkin diminta segera untuk menginstalnya.



Terakhir, Anda harus melindungi komputer Anda dari menerima koneksi masuk. Sebuah *firewall* adalah sebuah perangkat yang menyaring lalu lintas jaringan, dan Linux memiliki satu built-in. Jika Anda menggunakan Ubuntu,

maka **Gufw** adalah antarmuka grafis untuk **Uncomplicated Firewall (UFW) Ubuntu** .



Di balik terpal, Anda menggunakan **iptables** , yang merupakan sistem firewall bawaan. Alih-alih memasukkan perintah iptables yang rumit, Anda menggunakan GUI. Meskipun GUI ini memungkinkan Anda membuat kebijakan yang efektif untuk desktop, GUI ini hampir tidak menggores permukaan dari apa yang dapat dilakukan iptables.

4.6.3 PRIVACY TOOLS

Penggunaan alat privasi modern, baik di tingkat server dan pengguna, dapat membantu mencegah gangguan sistem dan akses tidak sah ke data.

Kabar baiknya adalah Linux secara default adalah salah satu sistem operasi paling aman yang pernah dibuat. Banyak eksploitasi yang mengganggu sistem operasi lain tidak akan berfungsi di Linux karena arsitektur yang mendasarinya. Namun, masih banyak kelemahan yang diketahui yang dapat dimanfaatkan oleh peretas sehingga administrator sistem proaktif bijaksana untuk menyebarkan alat privasi yang melindungi pengguna mereka serta sistem yang mereka gunakan.

Enkripsi mungkin adalah alat privasi paling terkenal dan paling banyak digunakan yang digunakan saat ini. Administrator menerapkan enkripsi

dengan kunci otentikasi di hampir setiap sistem yang berkomunikasi dengan dunia luar. Salah satu contoh yang terkenal adalah standar **HyperText Transfer Protocol Secure (HTTPS) yang** digunakan pada server web untuk memastikan bahwa data yang dikirimkan antara pengguna dan sumber daya online tidak dapat dicegat saat dikirimkan di Internet terbuka.

Jaringan pribadi virtual (VPN) telah digunakan oleh perusahaan untuk menghubungkan server jarak jauh dan karycloud mereka selama bertahun-tahun. Sekarang mereka mendapatkan popularitas di antara pengguna biasa yang ingin melindungi privasi online mereka. Mereka bekerja dengan membuat saluran komunikasi terenkripsi antara dua sistem, sehingga data yang dikirimkan di antara mereka diacak oleh algoritma yang hanya diketahui oleh sistem.

Proyek **Tor** telah lama terlibat dalam pembuatan alat privasi seperti **Tor Browser** yang bekerja dengan menyampaikan permintaan internet melalui jaringan server yang mencegah situs web dan orang lain mempelajari identitas orang yang membuat permintaan tersebut.

Alat-alat ini terus berkembang dan memilih mana yang sesuai untuk pengguna dan sistem yang terlibat merupakan bagian penting dari peran administrator sistem.

4.7 CLOUD

Pasti Anda pernah mendengar tentang *Cloud*. Baik Anda menggunakan Google Docs untuk pekerjaan rumah atau menyimpan musik dan foto di iCloud, Anda mungkin memiliki setidaknya beberapa konten digital yang dihosting di server cloud di suatu tempat.

Cloud Computing telah merevolusi cara kita mengakses teknologi. Karena konektivitas dan kecepatan Internet telah meningkat, semakin mudah untuk memindahkan sumber daya komputasi ke lokasi terpencil di mana konten dapat diakses, dimanipulasi, dan dibagikan di seluruh dunia. Organisasi semakin

memandang cloud sebagai hal penting untuk bisnis dan operasi mereka. Migrasi aplikasi dan proses TI organisasi ke layanan cloud, yang dikenal sebagai *adoption cloud*, dengan cepat menjadi keputusan bisnis strategis bagi banyak orang. Dengan adoption cloud yang meningkat secara signifikan di seluruh dunia, komputasi cloud bukanlah slogannya seperti dulu. Komputasi cloud dipandang sebagai salah satu teknologi utama yang mengganggu dalam dekade mendatang yang akan secara signifikan mengubah bisnis, ekonomi, dan kehidupan secara global.

Secara fisik, cloud dapat digambarkan sebagai sumber daya komputasi dari satu atau banyak pusat data di luar situs yang dapat diakses melalui internet. Cloud dibangun berdasarkan manfaat dari pusat data dan menyediakan solusi komputasi untuk organisasi yang perlu menyimpan dan memproses data, dan memungkinkan mereka untuk mendelegasikan pengelolaan infrastruktur TI kepada pihak ketiga. Data dan sumber daya yang disimpan organisasi di cloud dapat mencakup data, server, penyimpanan, hosting aplikasi, analitik, dan berbagai layanan lainnya.

Model penerapan cloud memberikan dasar tentang bagaimana infrastruktur cloud dibangun, dikelola, dan diakses. Ada empat model penerapan cloud utama:

- **Cloud Publik** : **Cloud** publik adalah infrastruktur cloud yang digunakan oleh penyedia untuk menawarkan layanan cloud kepada masyarakat umum dan organisasi melalui Internet. Dalam model cloud publik, mungkin ada beberapa penyewa (konsumen) yang berbagi resource cloud yang sama. Kemungkinan besar, banyak dari kita telah mengakses sumber cloud publik di beberapa titik melalui penyedia seperti Amazon, Google, dan penyedia cloud publik populer lainnya.
- **Cloud Pribadi** : **Cloud** pribadi adalah infrastruktur cloud yang diatur untuk penggunaan tunggal organisasi tertentu. Jika dibandingkan dengan cloud publik, cloud pribadi menawarkan tingkat privasi yang lebih tinggi

kepada organisasi, dan kontrol atas infrastruktur cloud, aplikasi, dan data. Ini dapat dihosting baik di server yang dikelola oleh perusahaan yang menggunakannya atau melalui penyedia cloud pribadi yang dikelola seperti Rackspace atau IBM.

- **Cloud Komunitas** : **Cloud** komunitas adalah infrastruktur cloud yang disiapkan untuk penggunaan tunggal oleh sekelompok organisasi dengan tujuan atau persyaratan yang sama. Organisasi yang berpartisipasi dalam komunitas biasanya berbagi biaya layanan cloud komunitas. Option ini mungkin lebih mahal daripada cloud publik; namun, ini mungkin menawarkan tingkat kontrol dan perlindungan yang lebih tinggi terhadap ancaman eksternal daripada cloud publik.
- **Cloud Hibrida** : **Cloud** hibrida terdiri dari dua atau lebih cloud individu, yang masing-masing dapat berupa cloud pribadi, komunitas, atau publik. Cloud hibrida dapat berubah seiring waktu saat cloud komponen bergabung dan pergi. Penggunaan teknologi tersebut memungkinkan portabilitas data dan aplikasi. Ini juga memungkinkan perusahaan untuk memanfaatkan sumber daya luar sambil tetap mempertahankan kendali atas sumber daya yang sensitif.

4.7.1 LINUX IN THE CLOUD

Linux memainkan peran penting dalam komputasi cloud. Ini memberdayakan 90% dari beban kerja cloud publik, sebagian besar server virtual didasarkan pada beberapa versi kernel Linux, dan Linux sering digunakan untuk menghosting aplikasi di belakang layanan komputasi cloud. Jadi apa yang membuat Linux secara unik cocok untuk mengaktifkan komputasi cloud?

Fleksibilitas

Komputasi cloud memberikan kemampuan untuk menyediakan sumber daya TI dengan cepat dan kapan saja. Ketangkasan ini memungkinkan pengembangan dan eksperimen yang cepat yang, pada gilirannya,

memfasilitasi inovasi yang penting untuk penelitian dan pengembangan, penemuan pasar baru dan peluang pendapatan, menciptakan segmen pelanggan baru, dan pengembangan produk baru.

Akibatnya, komputasi cloud harus mengimbangi fakta bahwa setiap organisasi memiliki seperangkat persyaratan sumber daya yang unik dan terus berkembang.

Linux menonjol di sini karena sangat mudah beradaptasi. Sebagai permulaan, Linux memiliki desain modular, dan di tengah ekosistem aplikasi open source yang sangat besar yang menyediakan option konfigurasi tanpa akhir untuk menyesuaikan dengan berbagai sistem dan kasus penggunaan. Selain itu, Linux menskalakan secara efisien, memungkinkannya menjalankan apa pun dari sensor jarak jauh yang kecil ke seluruh server farm.

Aksesibilitas

Dalam lingkungan tradisional, sumber daya TI diakses dari perangkat khusus, seperti desktop atau laptop. Dalam komputasi cloud, aplikasi dan data berada secara terpusat dan diakses dari mana saja melalui jaringan dari perangkat apa pun, seperti desktop, seluler, atau klien tipis, dan ada versi Linux untuk setiap perangkat ini.

Hemat Biaya

Komputasi cloud menarik karena berpotensi bagi konsumen untuk mengurangi biaya TI mereka. Dalam komputasi cloud, konsumen dapat menskalakan sumber daya TI secara sepihak dan otomatis untuk memenuhi permintaan beban kerja, sehingga menghilangkan overhead dari sumber daya yang kurang dimanfaatkan. Selain itu, biaya yang terkait dengan konfigurasi TI, manajemen, ruang lantai, daya, dan pendinginan berkurang.

Penyedia cloud menanggung biaya infrastruktur ini tetapi harus tetap menjadi alternatif berbiaya rendah. Memilih Linux adalah salah satu penyedia solusi paling hemat biaya yang dapat diterapkan. Linux adalah salah satu sistem operasi yang paling hemat daya, dan kernel Linux benar-benar gratis,

begitu pula banyak aplikasi terkait, utilitas, dan komponen perangkat lunak tambahan.

Organisasi perusahaan dan pemerintah dapat memilih untuk membayar distribusi yang didukung secara komersial, yang masih lebih hemat biaya jika dibandingkan dengan pesaing berlisensi. Distribusi non-komersial yang mendukung komputasi cloud juga merupakan pilihan yang layak bagi banyak organisasi.

Vendor tidak hanya dapat memberikan penghematan ini kepada pelanggan, menawarkan solusi berbasis Linux dapat lebih murah untuk diterapkan oleh klien. Menyiapkan Linux di sistem mereka sendiri menghilangkan biaya lisensi pengguna yang mahal yang berpotensi terkait dengan sistem operasi pesaing.

Pengelolaan

Sementara Linux dimulai sebagai sistem operasi khusus, kehadirannya yang meluas di industri TI telah menjadikan penggunaan dan administrasi Linux sebagai keterampilan yang diperlukan bagi para profesional TI. Menjadi semakin mudah bagi vendor cloud dan konsumen untuk memperoleh bakat yang diperlukan, atau mengalokasikan kembali anggota tim yang ada.

Sifat Linux, dibangun di atas bahasa pemrograman C, juga cocok untuk alat manajemen otomatis. Sebagian besar server Linux yang beroperasi di cloud dibuat dan dikelola oleh program manajemen otomatis daripada operator manusia. Proses ini membebaskan administrator untuk memantau operasi komputasi daripada mengonfigurasi dan memperbarui sistem secara manual.

Keamanan

Saat menggunakan solusi cloud, terutama cloud publik, organisasi mungkin memiliki masalah terkait privasi, ancaman eksternal, dan kurangnya kontrol atas sumber daya dan data TI.

Linux dapat membantu mengatasi masalah ini karena ini adalah salah satu sistem operasi paling aman dan andal yang tersedia. Linux adalah open source, artinya kode sumbernya tersedia bagi siapa saja untuk diperoleh, ditinjau, dan dimodifikasi. Ini juga berarti kode dapat diperiksa untuk masalah kerentanan dan kompatibilitas, menghasilkan upaya komunitas yang luas untuk memperbaiki masalah ini dan mempertahankan reputasi Linux yang kuat.

Virtualisasi

Virtualisasi adalah salah satu kemajuan paling signifikan yang telah berkontribusi pada pemberdayaan cloud komputasi.

Linux adalah *sistem operasi multi-pengguna* , yang berarti banyak pengguna yang berbeda dapat bekerja pada sistem yang sama secara bersamaan dan sebagian besar tidak dapat melakukan hal-hal yang merugikan pengguna lain. Namun, ini memiliki batasan - pengguna dapat menghabiskan ruang disk atau menggunakan terlalu banyak memori atau sumber daya CPU dan membuat sistem menjadi lambat untuk semua orang. Berbagi sistem dalam mode multi-pengguna juga mengharuskan setiap orang menjalankan sebagai pengguna yang tidak memiliki hak istimewa, jadi membiarkan setiap pengguna menjalankan server web mereka sendiri, misalnya, merupakan tantangan.

Virtualisasi adalah proses di mana satu komputer fisik, yang disebut *host* , menjalankan banyak salinan sistem operasi, setiap salinan disebut *tamu* . Gambar tamu ini dapat dikonfigurasi sebelumnya untuk fungsi tertentu untuk memungkinkan penerapan cepat, sering kali secara otomatis, saat diperlukan. Sistem host menjalankan perangkat lunak yang disebut *hypervisor* yang mengalihkan sumber daya antara berbagai tamu seperti yang dilakukan kernel Linux untuk proses individual. Dengan hypervisor logam kosong, hypervisor berjalan langsung di perangkat keras

komputer daripada di atas OS yang membebaskan lebih banyak sumber daya untuk gambar tamu.

Virtualisasi berfungsi karena server menghabiskan sebagian besar waktunya untuk menganggur dan tidak memerlukan sumber daya fisik seperti monitor dan keyboard. Dengan perangkat lunak dari perusahaan seperti **VMWare** dan **Openbox** , Anda sekarang dapat menggunakan CPU yang kuat dan dengan menggunakannya untuk menjalankan beberapa mesin virtual, administrator dapat mengoptimalkan penggunaan sumber daya fisik dan secara dramatis mengurangi biaya dibandingkan model pusat data satu mesin satu OS sebelumnya. Batasan utama biasanya adalah memori, namun dengan kemajuan dalam teknologi hypervisor dan CPU, dimungkinkan untuk menempatkan lebih banyak mesin virtual pada satu host daripada sebelumnya.

Dalam lingkungan virtual, satu host dapat menjalankan lusinan sistem operasi tamu, dan dengan dukungan dari CPU itu sendiri, para tamu bahkan tidak tahu bahwa mereka berjalan pada mesin virtual. Setiap tamu mendapatkan sumber daya virtualnya sendiri dan berkomunikasi dengan jaringannya sendiri. Bahkan tidak perlu menjalankan sistem operasi yang sama pada semua tamu, yang selanjutnya mengurangi jumlah server fisik yang dibutuhkan.

Virtualisasi menawarkan cara bagi perusahaan untuk menurunkan penggunaan daya dan mengurangi ruang pusat data melalui armada server fisik yang setara. Tamu sekarang hanya berupa konfigurasi perangkat lunak, jadi mudah untuk menjalankan mesin baru untuk diuji dan dihancurkan ketika kegunaannya telah berlalu.

Karena dimungkinkan untuk menjalankan beberapa contoh sistem operasi pada satu mesin fisik dan menghubungkannya melalui jaringan, lokasi mesin tidak menjadi masalah. Komputasi cloud mengambil pendekatan ini dan memungkinkan administrator untuk memiliki mesin virtual di pusat data

jarak jauh yang dimiliki oleh perusahaan lain, dan hanya membayar sumber daya yang digunakan. Vendor komputasi cloud dapat memanfaatkan skala ekonomi untuk menawarkan sumber daya komputasi dengan harga yang jauh lebih rendah daripada mengoperasikan pusat data di tempat.

Kontainer dan Penerapan Bare Metal

Dengan munculnya teknologi containerization seperti **Docker** dan software aplikasi **Kubernetes** sekarang sedang ditulis yang berjalan di lingkungan *tanpa server* . Pada dasarnya, pemrogram membuat perangkat lunak yang melakukan satu fungsi sistem (seperti pemrosesan atau penyimpanan basis data) yang berjalan dalam satu wadah. Penampung ini diatur dalam *pod* yang berjalan di dalam sebuah *node* dan dapat berbicara satu sama lain, dan dengan dunia luar jika diperlukan. Node, pada gilirannya, diatur dan dikendalikan oleh *node master*yang menyediakan layanan untuk setiap komponen dalam struktur. Membangun aplikasi dengan cara ini memisahkan setiap komponen dari yang lain, dan dari overhead menjalankan OS. Karena setiap bagian dari teka-teki dapat secara otomatis dihancurkan dan dibuat ulang oleh master node, mereka tidak perlu lagi sekuat perangkat lunak yang berjalan di atas OS. Meskipun arsitektur pemrograman baru ini dalam banyak hal melewati kebutuhan akan OS tradisional, teknologi yang mendasari yang membuatnya bekerja tetaplah Linux. Jadi, bekerja di Linux akan semakin banyak bekerja dalam tim pengembangan yang mengacu pada disiplin ilmu pemrograman, desain database, jaringan, dan administrasi sistem untuk membuat sistem di masa depan.

4.8 OPEN SOURCE LICENSING

Ketika berbicara tentang membeli perangkat lunak, ada tiga komponen berbeda:

- Ownership – Siapa yang memiliki kekayaan intelektualnya?

- Money Transfer – Apakah berbayar? Bagaimana cara membayarnya?
- Licensing – Apa yang didapat? Apa yang bisa dilakukan dengan perangkat lunak tersebut? Untuk berapa banyak komputer? Apakah bisa dibagikan?

Dalam kebanyakan kasus, kepemilikan perangkat lunak tetap pada orang atau perusahaan yang membuatnya. Pengguna hanya diberikan lisensi untuk menggunakan perangkat lunak; ini adalah masalah hukum hak cipta. Transfer uang tergantung pada model bisnis pencipta. Ini adalah lisensi yang membedakan perangkat lunak sumber terbuka dari perangkat lunak sumber tertutup.
Dimulai dari dua contoh yang kontras.

Dengan Microsoft Windows, Microsoft Corporation memiliki kekayaan intelektual. Lisensi itu sendiri, Perjanjian Lisensi Pengguna Akhir (EULA), adalah dokumen hukum khusus yang harus Anda klik, menunjukkan penerimaan Anda, untuk menginstal perangkat lunak. Microsoft menyimpan kode sumber dan hanya mendistribusikan salinan biner melalui saluran yang diotorisasi. Untuk sebagian besar produk konsumen, Anda diizinkan untuk menginstal perangkat lunak di satu komputer dan tidak diperbolehkan membuat salinan disk selain untuk cadangan. Anda tidak diperbolehkan merekayasa balik perangkat lunak. Anda membayar untuk satu salinan perangkat lunak, yang memberi Anda pembaruan kecil tetapi bukan peningkatan besar-besaran.

Linux dimiliki oleh Linus Torvalds. Dia telah menempatkan kode di bawah lisensi yang disebut GNU General Public License versi 2 (GPLv2). Lisensi ini, antara lain, mengatakan bahwa kode sumber harus dibuat tersedia bagi siapa saja yang meminta dan bahwa siapa pun diizinkan untuk melakukan perubahan. Satu peringatan untuk ini adalah bahwa jika seseorang membuat perubahan dan mendistribusikannya, mereka harus meletakkan perubahan di bawah lisensi yang sama sehingga orang lain dapat memperoleh manfaat. GPLv2 juga mengatakan bahwa tidak ada yang diizinkan membebankan biaya untuk mendistribusikan kode sumber selain dari biaya sebenarnya untuk melakukannya (seperti menyalinnya ke media yang dapat dilepas).

Secara umum, ketika seseorang menciptakan sesuatu, mereka juga mendapatkan hak untuk memutuskan bagaimana itu digunakan dan didistribusikan. Perangkat Lunak Bebas dan Sumber Terbuka (FOSS) mengacu pada perangkat lunak di mana hak ini telah dilepaskan; siapa pun boleh melihat kode sumber dan mendistribusikannya. Linus Torvalds telah melakukan itu dengan Linux - meskipun ia menciptakan Linux, ia tidak dapat melarang seseorang untuk menggunakannya di komputer mereka karena ia telah melepaskan hak itu melalui lisensi GPLv2.

Lisensi perangkat lunak adalah masalah politik, oleh karena itu tidak mengherankan bahwa ada banyak pendapat yang berbeda. Organisasi telah membuat lisensi mereka sendiri yang mewujudkan pandangan khusus mereka, sehingga lebih mudah untuk memilih lisensi yang ada daripada membuat sendiri. Sebagai contoh, universitas seperti Massachusetts Institute of Technology (MIT) dan University of California telah mengeluarkan lisensi, seperti halnya proyek seperti Apache Foundation. Juga, kelompok-kelompok seperti Free Software Foundation telah membuat lisensi mereka sendiri untuk memajukan agenda mereka.

4.8.1 THE FREE SOFTWARE FOUNDATION

Dua kelompok dianggap sebagai kekuatan paling berpengaruh di dunia open source: Free Software Foundation dan Open Source Initiative.

Hanya beberapa tahun setelah pengembangan proyek GNU, Richard Stallman mendirikan Free Software Foundation (FSF) pada tahun 1985 dengan tujuan mempromosikan perangkat lunak Free. Dalam konteks ini, kata "free" tidak merujuk pada harga, tetapi pada kebebasan untuk berbagi, mempelajari, dan memodifikasi kode sumber yang mendasarinya. Menurut situs web mereka, FSF percaya bahwa pengguna harus memiliki "kendali atas teknologi yang kami gunakan di rumah, sekolah, dan bisnis kami".

FSF juga menganjurkan bahwa lisensi perangkat lunak harus menegakkan keterbukaan modifikasi. Ini adalah pandangan mereka bahwa

jika seseorang memodifikasi perangkat lunak bebas maka mereka harus diminta untuk membagikan perubahan apa pun yang telah mereka buat ketika mereka membagikannya lagi. Filosofi spesifik ini disebut copyleft. Menurut FSF, "copyleft adalah metode umum untuk membuat program (atau pekerjaan lain) free (dalam arti kebebasan, bukan"harga nol "), dan mengharuskan semua versi program yang dimodifikasi dan diperluas untuk menjadi bebas juga".

FSF juga mengadvokasi menentang paten perangkat lunak dan bertindak sebagai pengawas untuk organisasi standar, berbicara ketika standar yang diusulkan mungkin melanggar prinsip-prinsip perangkat lunak bebas dengan memasukkan item-item seperti Digital Rights Management (DRM) yang membatasi apa yang dapat dilakukan dengan program yang sesuai.

FSF telah mengembangkan serangkaian lisensi mereka sendiri yang bebas bagi siapa saja untuk digunakan berdasarkan GNU General Public License (GPL) yang asli. FSF saat ini memelihara GNU General Public License versi 2 (GPLv2) dan versi 3 (GPLv3), serta GNU Lesser General Public License versi 2 (LGPLv2) dan versi 3 (LGPLv3). Lisensi ini dimaksudkan untuk dimasukkan dalam kode sumber aktual untuk memastikan bahwa semua varian dan modifikasi di masa depan dari program asli terus memiliki kebebasan penggunaan yang sama seperti aslinya. Lisensi GPL dan variannya adalah alat hukum yang kuat untuk memajukan penyebab perangkat lunak bebas di seluruh dunia. Apa yang dimulai pada tahun 1983 sebagai keinginan satu orang untuk berbagi dan meningkatkan perangkat lunak dengan membiarkan orang lain mengubahnya, pada akhirnya mengubah dunia.

Pertimbangkan ini
Perubahan antara GPLv2 dan GPLv3 sebagian besar berfokus pada penggunaan perangkat lunak bebas pada perangkat keras yang ditutup yang telah diciptakan Tivoization. TiVo adalah perusahaan yang membangun perekam video digital televisi pada perangkat keras mereka sendiri dan menggunakan Linux sebagai basis untuk perangkat lunak mereka. Sementara TiVo merilis kode sumber ke versi Linux mereka seperti yang dipersyaratkan dalam GPLv2, perangkat keras tidak akan menjalankan binari yang dimodifikasi. Di mata FSF, ini bertentangan dengan semangat GPLv2, jadi mereka menambahkan klausa khusus untuk versi 3 lisensi. Linus Torvalds setuju dengan TiVo tentang masalah ini dan telah memilih untuk tetap dengan GPLv2.

4.8.2 THE OPEN SOURCE INITIATIVE

Open Source Initiative (OSI) didirikan pada tahun 1998 oleh Bruce Perens dan Eric Raymond. Mereka percaya bahwa Free Software Foundation terlalu bermuatan politis dan bahwa lisensi yang kurang ekstrim diperlukan, terutama di sekitar aspek copyleft dari lisensi FSF. OSI percaya bahwa sumber tidak hanya harus tersedia secara bebas, tetapi juga bahwa tidak ada batasan harus ditempatkan pada penggunaan perangkat lunak, tidak peduli apa yang digunakan. Berbeda dengan FSF, OSI tidak memiliki lisensi sendiri. Sebaliknya, OSI memiliki seperangkat prinsip dan menambahkan lisensi ke daftar itu jika memenuhi prinsip-prinsip itu, yang disebut lisensi open source. Perangkat lunak yang sesuai dengan lisensi Open Source adalah perangkat lunak open source.

Salah satu jenis lisensi Open Source adalah BSD (Berkeley Software Distribution) dan turunannya, yang jauh lebih sederhana daripada GPL. Saat ini ada dua lisensi "BSD" yang disetujui oleh OSI, 2-Klausul dan 3-Klausul. Lisensi ini menyatakan bahwa Anda dapat mendistribusikan kembali sumber dan binari selama Anda mempertahankan pemberitahuan hak cipta dan tidak menyiratkan bahwa pembuat asli mendukung versi Anda. Dengan kata lain "lakukan apa yang Anda inginkan dengan perangkat lunak ini, jangan katakan Anda menulisnya." Menurut FSF, lisensi BSD asli memiliki kelemahan serius

karena mengharuskan pengembang untuk menambahkan klausul yang mengakui University of California, Berkeley dalam setiap iklan untuk perangkat lunak yang dilisensikan dengan cara ini. Ketika orang lain menyalin lisensi sederhana ini, mereka memasukkan pengakuan untuk lembaga mereka sendiri yang menyebabkan lebih dari 75 pengakuan seperti itu dalam beberapa kasus.

Lisensi FSF, seperti GPLv2, juga merupakan lisensi open source. Namun, banyak lisensi open source seperti BSD dan MIT tidak mengandung ketentuan copyleft dan karenanya tidak dapat diterima oleh FSF. Lisensi ini disebut lisensi perangkat lunak bebas permisif karena mereka permisif dalam cara Anda dapat mendistribusikan kembali perangkat lunak. Anda dapat mengambil perangkat lunak berlisensi BSD dan memasukkannya ke dalam produk perangkat lunak tertutup selama Anda memberikan atribusi yang tepat.

Daripada memikirkan poin-poin penting Open Source dan Perangkat Lunak Bebas, komunitas telah mulai merujuk mereka secara kolektif sebagai Perangkat Lunak Bebas dan Sumber Terbuka (FOSS). Kata bahasa Inggris "gratis" dapat berarti "gratis seperti dalam makan siang" (seperti tanpa biaya) atau "gratis seperti dalam ucapan" (seperti dalam tanpa batasan). Ambiguitas ini menyebabkan dimasukkannya kata "libre" untuk merujuk pada definisi yang terakhir. Jadi, kita berakhir dengan Perangkat Lunak Bebas / Bebas / Sumber Terbuka (FLOSS).

4.8.3 CREATIVE COMMONS

Lisensi FOSS sebagian besar terkait dengan perangkat lunak. Orang-orang telah menempatkan karya seperti gambar dan rencana di bawah lisensi FOSS, tetapi ini bukan maksudnya.

Ketika perangkat lunak telah ditempatkan di domain publik, penulis telah melepaskan semua hak, termasuk hak cipta atas karya tersebut. Di beberapa negara, ini adalah default ketika pekerjaan dilakukan oleh agen

pemerintah. Di beberapa negara, karya berhak cipta menjadi domain publik setelah penulis meninggal dan masa tunggu yang panjang telah berlalu.

Organisasi Creative Commons (CC) telah menciptakan Lisensi Creative Commons yang mencoba untuk mengatasi niat di balik lisensi FOSS untuk entitas non-perangkat lunak. Lisensi CC juga dapat digunakan untuk membatasi penggunaan komersial jika itu adalah keinginan pemegang hak cipta. Lisensi CC terdiri dari serangkaian kondisi yang dapat diterapkan oleh pembuatnya untuk pekerjaan mereka:

- Atribusi (BY) - Semua lisensi CC mensyaratkan bahwa pencipta harus diberi kredit, tanpa menyiratkan bahwa pencipta mendukung penggunaan.
- ShareAlike (SA) - Ini memungkinkan orang lain untuk menyalin, mendistribusikan, melakukan, dan memodifikasi pekerjaan, asalkan mereka melakukannya dengan persyaratan yang sama.
- NonCommercial (NC) - Ini memungkinkan orang lain untuk mendistribusikan, menampilkan, melakukan, dan memodifikasi karya untuk tujuan apa pun selain komersial.
- NoDerivatives (ND) - Ini memungkinkan orang lain untuk mendistribusikan, menampilkan, dan melakukan hanya salinan asli dari karya tersebut. Mereka harus mendapatkan izin pembuat untuk memodifikasinya.

Ketentuan ini kemudian digabungkan untuk membuat enam lisensi utama yang ditawarkan oleh Creative Commons:

- Attribution (CC BY) - Sama seperti lisensi BSD, Anda dapat menggunakan konten CC BY untuk penggunaan apa pun tetapi harus memberi kredit pada pemegang hak cipta.
- Attribution ShareAlike (CC BY-SA) - Versi copyleft dari lisensi Attribution. Karya yang diturunkan harus dibagikan di bawah lisensi yang sama, seperti halnya dalam ideal Perangkat Lunak Bebas.

- Attribution NoDerivs (CC BY-ND) - Anda dapat mendistribusikan kembali konten dalam kondisi yang sama dengan CC-BY tetapi tidak dapat mengubahnya.
- Attribution-NonCommercial (CC BY-NC) - Sama seperti CC BY, tetapi Anda tidak boleh menggunakannya untuk tujuan komersial.
- Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) - Dibangun di atas lisensi CC BY-NC tetapi mengharuskan perubahan Anda dibagikan di bawah lisensi yang sama.
- Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) - Anda membagikan konten yang akan digunakan untuk tujuan non-komersial, tetapi orang tidak dapat mengubah konten.

Tanpa Hak Dilindungi Undang-undang (CC0) - Ini adalah versi publik dari domain Creative Commons.

4.8.4 OPEN SOURCE BUSINESS MODELS

Jika semua perangkat lunak ini gratis, bagaimana orang dapat menghasilkan uang darinya?

Pertama, Anda harus memahami bahwa tidak ada apa pun di GPL yang melarang penjualan perangkat lunak. Faktanya, hak untuk menjual perangkat lunak adalah bagian dari lisensi GPL. Sekali lagi, ingat bahwa kata bebas mengacu pada kebebasan, bukan harga. Perusahaan-perusahaan yang menambah nilai pada program-program gratis ini didorong untuk menghasilkan uang sebanyak yang mereka bisa, dan mengembalikan laba tersebut ke pengembangan perangkat lunak yang lebih banyak dan lebih baik.

Salah satu cara paling sederhana untuk menghasilkan uang adalah dengan menjual dukungan atau garansi di sekitar perangkat lunak. Perusahaan seperti Canonical, pengembang Ubuntu, dan Redhat telah tumbuh menjadi perusahaan besar dengan menciptakan distribusi dan alat Linux yang memungkinkan pengguna komersial untuk mengelola perusahaan mereka dan menawarkan produk dan layanan kepada pelanggan mereka.

Banyak proyek open source lainnya juga telah berkembang menjadi bisnis besar. Pada 1990-an, Gerald Combs bekerja di penyedia layanan Internet dan mulai menulis alat analisis jaringan sendiri karena alat serupa pada saat itu mahal. Lebih dari 600 orang kini berkontribusi pada proyek ini, yang disebut Wireshark. Sekarang sering dianggap lebih baik daripada penawaran komersial dan menyebabkan perusahaan dibentuk untuk menjual produk dan dukungan. Seperti banyak perusahaan lain, perusahaan ini dibeli oleh perusahaan besar yang mendukung kelanjutan pengembangannya.

Perusahaan seperti Tivo memiliki paket perangkat keras atau menambahkan perangkat lunak sumber tertutup ekstra untuk dijual bersama perangkat lunak gratis. Peralatan dan sistem tertanam yang menggunakan Linux adalah bisnis bernilai miliaran dolar dan mencakup semuanya, mulai dari DVR rumah hingga kamera keamanan dan perangkat kebugaran yang dapat dikenakan. Banyak firewall konsumen dan perangkat hiburan mengikuti model ini.

Saat ini, baik pengusaha besar dan kecil memiliki individu dan kadang-kadang seluruh kelompok yang dikhususkan untuk bekerja pada proyek-proyek sumber terbuka. Perusahaan teknologi bersaing untuk mendapatkan kesempatan mempengaruhi proyek yang akan membentuk masa depan industri mereka. Perusahaan lain mendedikasikan sumber daya untuk proyek yang mereka butuhkan untuk penggunaan internal. Karena semakin banyak bisnis dilakukan pada sumber daya cloud, peluang bagi programmer open source terus berkembang.

Referensi:

https://www.netacad.com/courses/os-it/ndg-linux-essentials