

BAB III

LANDASAN TEORI

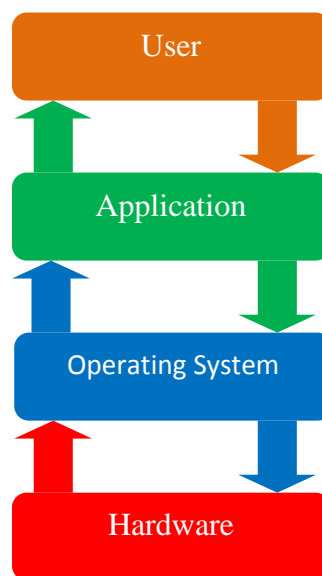
3.1. Sistem Operasi

Sistem operasi adalah sekumpulan rutin perangkat lunak yang berada di antara program aplikasi dan perangkat keras. Semua perangkat lunak berjalan di bawah kendali sistem operasi, mengakses perangkat keras lewat sistem operasi (Hariyanto, 2017:10).

Sistem operasi memiliki peranan yang penting dalam sistem berbasis komputer. Ada dua tugas utama sistem komputer. Kedua tugas tersebut yaitu sistem operasi sebagai *extended machine* dan sistem operasi sebagai *resource manager*. Sistem operasi sebagai *extended machine* berarti sistem operasi bertugas untuk menyembunyikan kerumitan *hardware* dan menyediakan antarmuka yang lebih mudah dipakai oleh pengguna. Sedangkan sistem operasi sebagai *resource manager* berarti sistem operasi berfungsi sebagai pengatur sumber daya yang ada di komputer baik itu jaringan, *software* aplikasi, *hardware* atau sumber daya lainnya.

Sistem operasi dilihat dari sistem pengembangannya sendiri bisa dibagi menjadi dua. Pertama adalah sistem operasi dengan metode pengembangan tertutup seperti *Windows* dimana pada sistem operasi ini pengguna tidak bisa melihat dan mengubah *source code* dari sistem operasi tersebut dan yang kedua adalah sistem operasi *open source* dimana pengguna bisa memakai, melihat, dan mengembangkan kode penyusun dari sistem operasi tersebut.

Contoh sistem operasi *open source* yaitu *Linux*, *Minix*, *Free BSD*. Namun sistem operasi *free* yang paling populer saat ini adalah *Linux*. *Linux* bisa berjalan di atas arsitektur prosesor yang berbeda-beda. Dari super komputer, *server*, komputer pribadi, *handheld device* sampai *embedded system* (Agung, 2010:50).



Gambar 3.1 Letak Sistem Operasi pada Sistem Komputer

3.1.1. Elemen Sistem Operasi

Sistem operasi tersusun atas beberapa elemen yang tiap elemen mempunyai tugas tertentu yang saling berkaitan. Elemen-elemen dasar yang membentuk sistem operasi diantaranya (Fredy, 2009:35).

1. *Boot Loader*

Boot loader adalah bagian dari sistem operasi yang melakukan *booting*, pengambil alihan tugas dari *bios* ke sistem operasi. Selanjutnya *boot loader* akan menjalankan *kernel*

sehingga tugas beralih pada *kernel*. Pada *boot loader* juga tersimpan informasi-informasi yang dibutuhkan oleh sistem operasi.

2. *File System*

File system berfungsi untuk melakukan fungsi manajemen terhadap file, menyediakan layanan-layanan yang berhubungan dengan penggunaan file bagi program aplikasi guna memenuhi kebutuhan manajemen data bagi pemakai.

3. *Kernel*

Kernel merupakan jantung dari sistem operasi. Semua proses *input* dan *output* yang berlangsung selama computer berjalan diatur oleh *kernel*, seperti pembacaan dan penulisan terhadap disk, manajemen memori, dan penjadwalan program aplikasi.

4. *Shell*

Shell atau bisa juga disebut *user interface* adalah sebuah program aplikasi yang menghubungkan pengguna komputer dengan kernel. Segala perintah yang diberikan oleh pengguna komputer akan melalui perantara *shell*, kemudian *shell* akan memberikan perintah tersebut kepada *kernel* untuk diproses. Ada dua macam *user interface* yang paling umum, yaitu *command line* dan *graphical user interface*.

a. *Command Line Interface*

Antarmuka dengan sistem berupa baris perintah. Pengguna mengetikkan perintah dengan *keyboard* berupa sebaris teks/*string* yang menyatakan perintah ke sistem komputer agar komputer melakukan sesuatu.

b. *Graphical User Interface (GUI)*

Dalam komputasi, *graphical user interface* (GUI) adalah jenis antarmuka pengguna yang memungkinkan pengguna untuk berinteraksi dengan perangkat elektronik dengan gambar dari pada perintah teks (Linfo, 2004).

GUI dapat digunakan pada komputer, perangkat genggam seperti MP3 *player*, *gadget*, *portable media player* atau perangkat game, peralatan rumah tangga, dan peralatan kantor. GUI menampilkan informasi dan tindakan yang tersedia bagi pengguna melalui ikon grafis dan indikasi visual sebagai lawan dari antarmuka berbasis teks. Tindakan ini biasanya dilakukan melalui manipulasi langsung dari elemen grafis.

3.2. *Linux*

Linux merupakan sistem operasi *UNIX-like* yang menggunakan *kernel Linux*. Pengembang *Linux* dilakukan oleh Linus Benedict Tovalds, Universitas Helsinki, Finlandia sebagai proyek hobi (Weeks, 2004).

Seluruh kode sumber *Linux* termasuk *kernel*, *device driver*, pustaka, program, dan perangkat pengembang disebarakan secara bebas dengan lisensi GNU GPL (*General Public License*). Nama *Linux* sendiri diambil dari *kernel Linux* yang ditulis oleh Linus, namun pendiri *Free Software Foundation*, Richard Stallman (pencetus *GNU project*) lebih memilih menyebutnya *GNU/Linux*.

Linux tidak memuat kode *UNIX*. Melainkan ditulis ulang berdasarkan standar *POSIX*. Linus mengelola *kernel*, menerima penambahan dan modifikasi dari banyak orang di seluruh dunia. Linus menerapkan kendali kualitas dan menambahkan semua kode baru ke *kernel*. *Linux* telah di-*port* ke banyak *platform* antara lain Intel x86, *Alpha AXP*, *MIPS*, *Sparc*, dan *PowerPC*. Sekitar 95% kode sumber *kernel* adalah sama untuk semua *platform* perangkat keras.

3.2.1. Perkembangan Linux

Perbedaan utama antara *Linux* dan sistem operasi populer lainnya terletak pada *kernel Linux* dan komponen-komponennya yang bebas dan terbuka. *Linux* bukan satu-satunya sistem operasi dalam kategori tersebut, walaupun demikian *Linux* adalah contoh terbaik dan terbanyak digunakan. Beberapa lisensi perangkat lunak bebas dan sumber terbuka berdasarkan prinsip-prinsip *copyleft*, sebuah konsep yang menganut prinsip, karya yang dihasilkan dari bagian *copyleft* harus juga merupakan *copyleft*. Lisensi perangkat lunak bebas yang paling umum, GNU *General Public License* 12

(GPL) adalah sebuah bentuk *copyleft* dan digunakan oleh *kernel Linux* dan komponen-komponen dari proyek GNU.

Sistem *Linux* berkaitan erat dengan standar-standar *POSIX*, *SUS*, *ISO* dan *ANSI*. Akan tetapi, baru distribusi *Linux-FT* saja yang mendapatkan sertifikasi *POSIX* (Agung, 2010:33).

3.2.2. Distro *Linux*

Sebuah distribusi *Linux*, yang umum disebut dengan “distro”, adalah sebuah proyek yang bertujuan untuk mengatur sebuah kumpulan perangkat lunak berbasis *Linux* dan memfasilitasi instalasi dari sebuah sistem operasi *Linux* (Agung, 2010:40).

Linux adalah *kernel*, program di dalam sistem yang mengalokasikan *resource* dari mesin untuk program lain yang dijalankan. *Kernel* merupakan bagian penting dari sistem operasi, namun tidak bisa berdiri sendiri. Ia hanya bisa berfungsi dalam sebuah sistem operasi yang utuh. Keseluruhan sistem operasi *Linux* pada dasarnya adalah *tools* GNU di tambah *Linux* (GNU/*Linux*). Sehingga yang disebut distribusi *Linux* sesungguhnya adalah distribusi dari GNU/*Linux*.

Berikut adalah pengelompokan distro menurut fungsi dan targetnya :

1. Distro umum

Distro umum adalah distro yang dipakai untuk keperluan umum. Distro jenis ini biasanya disebarkan dalam keeping CD/DVD dengan paket *software* yang sangat lengkap.

2. Distro khusus

Distro khusus adalah distro yang ditunjukan untuk kepentingan khusus seperti untuk keamanan jaringan, pemrograman, atau keperluan spesifik termasuk multimedia.

3.2.3. Konsep Pengembangan *Distro*

Dalam pembangunan sebuah *distro Linux*, secara garis besar ada dua metode yang biasa digunakan, yaitu *from scratch*, dan *remastering*.

1. *Linux From Scratch (LFS)*

Linux From Scratch merupakan cara pembuatan distro secara mandiri, dimana semua aplikasi dikompilasi dari kode sumber murni (Agung, 2010).

Metode ini biasanya digunakan bagi pengembang untuk membuat distro tanpa bisa untuk dikembangkan lagi dalam hal ini distro dipakai untuk diri sendiri atau tujuan khusus misalnya untuk *firewall* atau *router*. Kelemahan dari metode ini adalah distro yang dihasilkan tidak bisa dibuat satu ISO yang *installable* dan *portabilitas* yang kurang, dan sulit untuk didistribusikan

kembali, walaupun dapat dikembangkan hal tersebut memerlukan usaha yang besar serta memerlukan tim yang handal.

2. *Remastering*

Dalam bidang *software*, *remastering* dapat diartikan sebagai sebuah proses pembungkusan ulang paket aplikasi pada sistem operasi induk dimana dapat menambah bahkan dapat juga mengurangi paket aplikasi yang telah disertakan.

Remaster merupakan proses pembuatan sistem operasi baru dengan paket aplikasi yang berbeda dari sistem aslinya. Untuk Menambah dan mengurangi paket aplikasi di sistem operasi yang ada dengan paket aplikasi yang baru (Agung, 2010).

Tujuan *remastering* adalah membuat sebuah sistem operasi yang sesuai dengan kehendak dan kebutuhan pembuatnya, dalam hal ini memang ditargetkan pada lingkungan tertentu seperti di bidang multimedia (Agung, 2010).

Hasil *remastering Linux* adalah *Linux* yang mirip dengan *Linux* induk namun telah mengalami beberapa modifikasi yang membuatnya berbeda dibandingkan dengan *Linux* induk, misal tema tampilan, perangkat lunak yang terbundel dengannya dan sebagainya (Edogawa, 2010).

3.2.4. *Live CD*

Live CD adalah istilah yang dipakai untuk sistem yang bisa dijalankan lewat media CD/DVD ROM. *Live CD* umumnya dipakai

untuk menjalankan *Linux* tanpa perlu melakukan instalasi kedalam *hard disk* (Brickner, 2005).

Tujuan dibuatnya *Live CD* bermacam-macam. Umumnya *Live CD* dibuat untuk mencoba *Linux* tanpa harus melakukan instalasi. Selain itu, pengguna yang berpengalaman dengan sistem operasi juga bisa menggunakan *Live CD* untuk menentukan sejauh mana sebuah sistem operasi kompatibel dengan *hardware* dan beberapa konfigurasi tertentu.

3.3. Model Pengembangan *Software*

3.3.1. *Close Source*

Secara konsep *Close source* identik dengan Microsoft (Suntopo, 2004). Dimana mereka memiliki konsep, jika menghasilkan suatu hal baru dalam bidang teknologi, hal tersebut menjadi hak paten dari si pembuat, sehingga jika ada seseorang yang ingin menggunakan produk tersebut harus membayar sejumlah uang, sebagai suatu tanda bahwa telah diizinkan untuk menggunakan produk tersebut.

Jika melihat konsep dasar tersebut terdapat suatu kendala, dimana jika program aplikasi yang kita gunakan mengalami kerusakan akan di perlukan waktu tambahan guna memperbaikinya, dimana kita tidak memiliki wewenang dan juga konfigurasi secara teknis dari apa yang digunakan. Hal lanjut yang akan merugikan kita

yaitu diperlukan biaya dan waktu tambahan, dimana pekerjaan menjadi tertunda.

3.3.2. *Open Source*

Open source merupakan paradigma baru dalam dunia informatika. Adanya konsep *open source*, merupakan sebagai jawaban dari segala macam usaha dalam memberantas pembajakan *software* (Suntopo, 2004).

Open source jika menghasilkan suatu produk, bukan berarti orang lain tidak boleh mengetahui detail dari pengerjaan. Keuntungan yang didapat akan lebih besar disamping biaya yang lebih murah juga dalam pengembangannya banyak orang yang memberi ide.

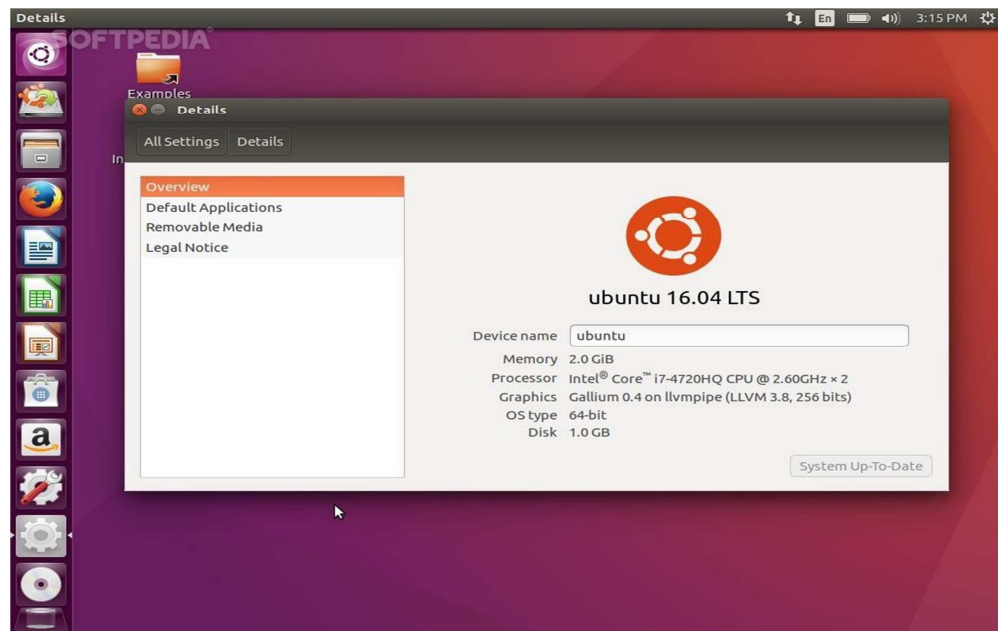
3.4. *General Public License (GPL)*

GNU *General Public License* (GNU GPL) merupakan suatu lisensi perangkat lunak bebas yang ditulis oleh Richard Stallman untuk proyek GNU (GNU, 2011).

Lisensi GPL memberikan penerima salinan perangkat lunak hak dari perangkat lunak bebas dan menggunakan *copyleft* untuk memastikan kebebasan yang sama diterapkan pada versi berikutnya dari karya tersebut. Saat ini GNU GPL sudah mencapai versi 3.

3.5. Ubuntu

Ubuntu merupakan salah satu distribusi *Linux* yang bebasiskan *Debian*. Proyek Ubuntu resmi disponsori oleh Canonical Ltd yang merupakan perusahaan milik seorang kosmot asal Afrika Selatan Mark Shuttleworth. Nama Ubuntu diambil dari nama sebuah konsep ideologi Afrika Selatan, “Ubuntu” berasal dari bahasa kuno Afrika, yang berarti “rasa perikemanusiaan terhadap sesama manusia”. Tujuan dari distribusi *Linux* Ubuntu adalah membawa semangat yang terkandung di dalam Filosofi Ubuntu ke dalam dunia perangkat lunak. Ubuntu adalah sistem operasi lengkap berbasis *Linux*, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari professional maupun komunitas (Ubuntu-id, 2010).

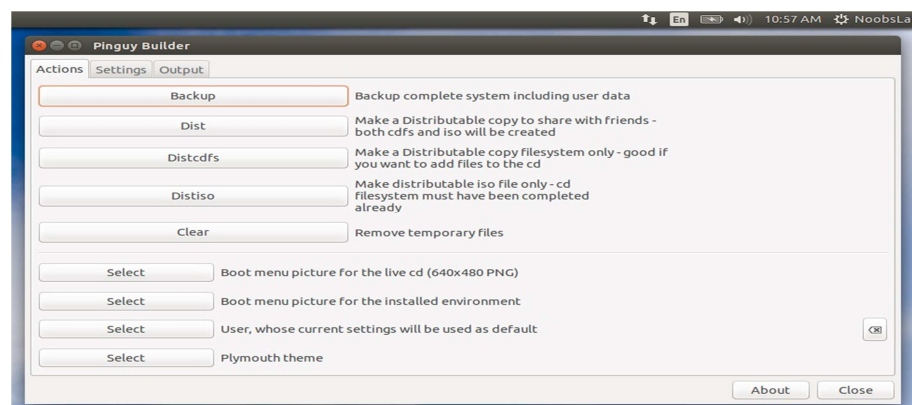


Gambar 3.2 Tampilan Ubuntu 16.04 LTS

3.6. Pinguy Builder

Pinguy Builder adalah aplikasi yang sangat berguna yang dapat digunakan oleh siapa saja untuk membuat Ubuntu *Live* CD dari awal atau untuk mendukung instalasi Ubuntu yang ada. Aplikasi ini hampir sama dengan aplikasi *Remastersys* yang ditinggalkan tahun lalu. *Pinguy Builder* memungkinkan untuk dengan mudah membackup seluruh sistem Ubuntu atau hanya membuat *ISO* diinstal dari sistem yang sedang berjalan. Kemudian, dapat membakar *ISO* untuk setiap DVD atau USB, dan menginstall OS yang pada sistem apapun. Dapat meliputi merek sendiri seperti logo, tema *plymouth*, tema *boot menu*. Aplikasi ini memiliki *front-end GUI*, dukungan *EFI* lengkap termasuk partisi sistem *EFI*. Mengisi *pool folder* dengan *grub-EFI* untuk instalasi *offline*.

Entri pembaruan *grub* dengan penamaan digunakan untuk label CD. Menggunakan *XZ* untuk kompres *filesystem*. *Squashfs* untuk ukuran *ISO* yang lebih kecil (Nurhidayat, 2015).



Gambar 3.3 Tampilan Pinguy Builder

3.7. Repository

Repository merupakan sekumpulan paket-paket software alias program untuk suatu sistem operasi (red : *Linux*) yang dipakai untuk menunjang kinerja dari suatu, program, serta sebagainya yang didapatkan dari situs paket-paket tersebut (Anan, 2015).

Bisa dibilang *Repository* ini merupakan paket-paket khusus untuk suatu sistem operasi yang kemudian paket-paket tersebut di instal untuk mendapat kinerja lebih baik dari suatu sistem operasi. Suatu sistem operasi semacam ini kan terdiri dari beberapa macam distro. mulai dari yang berbasiskan .rpm, .deb, serta lain-lain. Jadi, pasti saja disediakan paket-paket khusus untuk menunjang kinerja *software* serta program dari sistem operasi tersebut agar tak tercampur aduk oleh paket-paket untuk distro lain. Contohnya semacam paket-paket (*Repository*) dari *Mandriva* berbeda dengan paket-paket yang ada pada *Ubuntu*. Sebab kedua extensi tersebut, tak kompatibel satu sama lain.

Maka dari itu, agar pemakai bisa dengan mudah mengunduh *Repository*, dibuatlah *Mirror Repository* tersebut semacam , serta beberapa mirror-mirror lokal lainnya. Sehingga bisa diambil kesimpulan bahwa *Repository* itu merupakan paket-paket yang dikhususkan serta disediakan untuk di instal oleh beberapa macam Distro *Linux*.

```
deb http://kambing.ui.ac.id/ubuntu/ xenial main restricted universe multiverse
deb http://kambing.ui.ac.id/ubuntu/ xenial-updates main restricted universe multiverse
deb http://kambing.ui.ac.id/ubuntu/ xenial-security main restricted universe multiverse
deb http://kambing.ui.ac.id/ubuntu/ xenial-backports main restricted universe multiverse
deb http://kambing.ui.ac.id/ubuntu/ xenial-proposed main restricted universe multiverse
```

Gambar 3.4 Tampilan *Repository*

Repository perangkat lunak mengandung ribuan paket perangkat lunak yang dikelompokkan ke dalam empat “komponen”. Pengelompokan dibangun berdasarkan tipe dukungan yang bisa diberbagi serta apakah paket tersebut mengikuti Filosofi Perangkat Lunak Bebas. Komponen-komponen tersebut ialah “*main*”, “*restricted*”, “*universe*”, dan “*multiverse*”. *Repository* perangkat lunak Ubuntu dibagi menjadi empat komponen (*main*, *restricted*, *universe*, dan *multiverse*) yang dibagi berdasarkan dukungan yang diberbagi serta apakah perangkat lunak tersebut sesuai dengan tujuan yang ada dalam Filosofi Perangkat Lunak Bebas.

Instalasi standar Ubuntu berisi sebagian perangkat lunak yang berasal dari komponen *main* serta *restricted*. Hal ini bisa meperbuat instalasi perangkat lunak tambahan dengan memakai *Synaptic Package Manager* alias *Aptitude*. Komponen lainnya ditambahkan dengan cara merubah berkas */etc/apt/sources.list*.

1. Komponen “*main*”

Komponen “*main*” berisi software yang tergolong perangkat lunak bebas, bisa didistribusikan ulang dengan bebas, serta didukung penuh oleh tim Ubuntu. Dalam komponen ini terdapat *software open source* yang paling terkenal serta bisa dipercaya, yang sebagian besar telah terinstal dengan cara baku (*default*) saat Kamu meperbuat instalasi Ubuntu. Perangkat lunak dalam komponen *main* berisi *software* yang dirasakan penting oleh pengembang Ubuntu, komunitas, pemakai, tim keamanan, serta distribusi Ubuntu berniat mendukungnya. Dengan menginstall

perangkat lunak dari komponen *main*, hal ini bisa percaya bahwa perangkat lunak tersebut hadir dengan pembetulan keamanan serta dukungan teknis.

Perangkat lunak dalam komponen *main* ini ialah perangkat lunak yang diperlukan oleh tak sedikit orang dalam membangun desktop yang *manfaatonal* alias *server* internet yang hanya menjalankan perangkat lunak *open source*. Lisensi dari *software* wajiblah bebas, tetapi komponen *main* bisa berisi *firmware biner* serta font opsi yang tak bisa dimodifikasi tanpa izin dari pemiliknya. Batasan yang ada pada segala permasalahan di atas tak boleh memberatkan hak untuk meperbuat distribusi ulang.

2. Komponen “*restricted*”

Komponen *restricted* disediakan bagi perangkat lunak yang umum dipakai serta didukung oleh tim Ubuntu mesikipun tak hadir dalam lisensi leluasa dengan cara penuh. Harap dicatat bahwa perangkat lunak tersebut bisa saja tak didukung dengan cara penuh mengingat bahwa hal ini tak bisa membenahi sendiri perangkat lunak tersebut. Hal ini hanya bisa mengabarkan persoalan yang terjadi ke pemilik aslinya. Beberapa perangkat lunak dari komponen *restricted* bakal diinstal dalam CD Ubuntu, tetapi telah dipisahkan jadi bisa dihapus dengan mudah. Dengan memasukkannya perangkat lunak semacam ini sebab perangkat lunak tersebut diperlukan oleh Ubuntu agar bisa berlangsung pada beberapa mesin tertentu. Contohnya ialah *driver biner* dari *vendor video card* yang ialah satu-satunya cara agar Ubuntu bisa berlangsung diatasnya.

3. Komponen “*universe*”

Komponen *universe* ialah fotoan dari *Linux* serta dunia perangkat lunak leluasa serta *open source*. Dalam komponen *universe*, akan mendapat hampir segala perangkat lunak *open source* maupun perangkat lunak dengan lisensi *less open*. Semuanya telah dibangun dari beberapa sumber yang terdapat bagi publik. Segala perangkat lunak ini disusun memakai *library* serta *tool* yang menjadi tahap dari komponen *main*. Jadi perangkat lunak tersebut sewajibnya bisa diinstal serta bekerja dengan *main* dengan perangkat lunak yang terdapat dalam komponen *main*. Tetapi perangkat lunak dalam komponen *universe* ini tak dijamin bakal mendapat pembetulan keamanan maupun dukungan. Komponen *universe* berisi ribuan perangkat lunak. Dengan memakai komponen *universe* ini, *user* dapat merasakan keragaman serta *fleksibilitas* yang disediakan oleh dunia yang luas serta dibangun di atas inti Ubuntu yang stabil.

4. Komponen “*multiverse*”

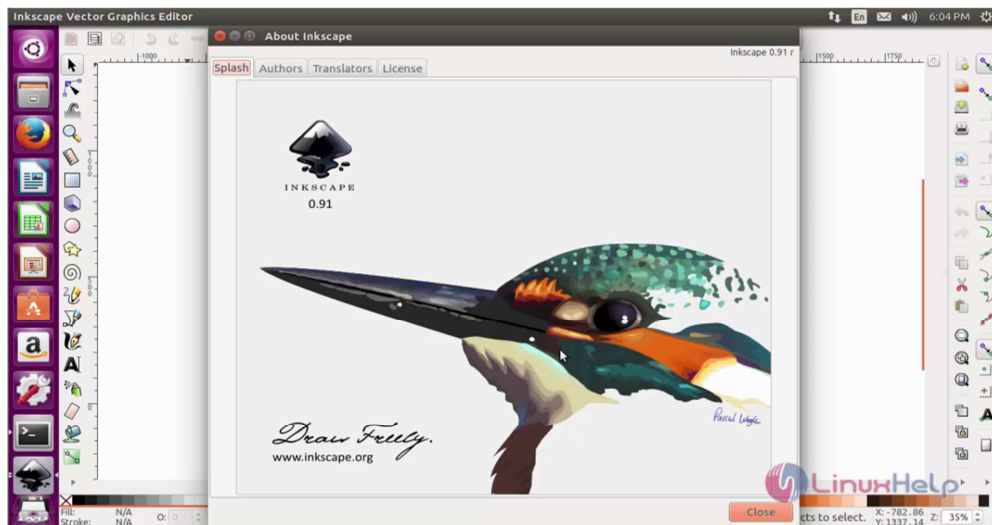
Komponen *multiverse* ialah komponen yang berisi perangkat lunak yang “tidak bebas”. Dalam artian lisensi yang disertakan pada perangkat lunak tersebut tak sesuai dengan Kebijakan Lisensi pada Komponen “*main*”.

3.8. *Inkscape*

Inkscape adalah sebuah perangkat lunak editor gambar vektor yang bersifat perangkat lunak bebas dibawah lisensi *GNU GPL*. Tujuan utama dari

Inkscape adalah menjadi perangkat grafik mutakhir yang memenuhi standar *XML*, *SVG*, dan *CSS* (Indra:2014).

Inkscape bersifat *cross-platform* dan dapat dijalankan pada *Mac OS X* (biasanya dibawah aplikasi *X11*, walaupun *toolkit GTK+* yang digunakan dapat dikompilasikan untuk beroperasi secara langsung dibawah *Quartz*), sistem operasi berbasis *Unix*, dan *Microsoft Windows*. Implementasi *SVG* dan *CSS* di *Inkscape* belum sempurna, misalnya *Inkscape* tidak mendukung animasi *SVG*, dan *font SVG*, walaupun dukungan dasar untuk pembuatan *font SVG* telah di implementasikan pada versi 0.47.



Gambar 3.5 Tampilan awal *Inkscape*

3.9. *GIMP* (*GNU Image Manipulation Program*)

GIMP adalah alat pengolah gambar *multi-platform*, *GIMP* adalah singkatan dari *GNU Image Manipulation Program*. *GIMP* cocok untuk berbagai pekerjaan pengolahan gambar, termasuk perbaikan foto,

penggabungan dan pembuatan gambar. *GIMP* memiliki berbagai kemampuan. Software ini dapat digunakan untuk program paint sederhana, program perbaikan foto berkualitas tinggi, sistem pemrosesan *batch online*, produksi massal render gambar, *konverter format* gambar dan sebutan dari *GNU Image Manipulation Program* atau yang lebih dikenal dengan sebutan *GIMP* adalah perangkat lunak untuk manipulasi grafik berbasis *raster*. *GIMP* berjalan pada desktop *GNOME* dan dirilis dengan lisensi *GNU General Public License*.

GIMP pada awalnya dikembangkan untuk desktop X11 yang berjalan di platform *Unix*. Namun saat ini software ini sudah di *porting* ke beberapa *platform* sistem operasi yang lain yaitu *MS Windows* dan *Mac OS*. Grafik yang dihasilkan oleh *GIMP* disimpan dengan format *XCF* dan bisa di *ekspor* ke berbagai format gambar seperti *bmp*, *jpg*, *gif*, *pdf*, *png*, *svg*, *tiff*, dan masih banyak lagi yang lainnya.

GIMP menyediakan banyak sekali *plugin* yang memudahkan kita dalam mengolah *image* dengan cepat. Maksudnya *multi-platform* adalah bisa dijalankan di berbagai sistem operasi. Saat ini *GIMP* sudah bisa dijalankan di *GNU/Linux*, *Apple MacOS-X*, *Microsoft Windows*, *OpenBSD*, *NetBSD*, *FreeBSD*, *Solaris*, *SunOS*, *AIX*, *HP-UX*, *Tru64*, *Digital UNIX*, *OSF/1*, *IRIX*, *OS/2* dan *BeoS*. Dinamakan *GNU Image Manipulation Program*, jika suatu program sudah ada embel-embel GNU artinya program itu dikembangkan dalam lingkungan open-source yang berlisensi *GNU-*

GPL dibawah naungan *Free Software Foundation*. Istilah gampangnya bebas, bahkan bisa saja gratis.

GIMP mendukung banyak format gambar, termasuk *JPG* dan *PNG* yang sering digunakan. Format ini dapat dibuka di hampir semua *image editor* dan *image viewer*. Untuk penyimpanan dalam format mentah, *GIMP* memiliki file format sendiri yaitu *XCF*.

GIMP dapat digunakan dengan skrip yang disebut *Script-Fu*. Hal ini dapat mengkostumisasi berbagai fungsi/fitur menjadi fungsi baru sehingga mempercepat penggunaan fungsi sering yang dilakukan atau bahkan membuat fitur yang sebelumnya tidak ada untuk penggunaan lebih *expert*, dengan sifatnya yang *source-code*. *GIMP* dapat digunakan untuk membuat pengolahan atau *render* gambar *online* dengan menggabungkan dengan *CGI* atau lainnya sehingga memungkinkan kita mengolah gambar secara *online* (Indra:2014).



Gambar 3.6 Tampilan *GIMP*

3.10. *Terminal Command*

Seperti halnya mengetikkan perintah di DOS, *command line* atau baris perintah di Linux juga diketikkan di *prompt* dan diakhiri dengan menekan tombol Enter pada *keyboard* untuk mengeksekusi perintah tersebut. Baris perintah merupakan cara yang lebih efisien untuk melakukan suatu pekerjaan. Oleh karena itu pemakai linux tetap mengandalkan cara ini untuk bekerja.

Berikut akan dijelaskan beberapa perintah dasar yang mungkin kelak akan sering digunakan terutama oleh para pemula. Penjelasan masing-masing perintah akan dipersingkat dan untuk mengetahui lebih detail lagi fungsi-fungsi suatu perintah, pengguna dapat melihat manualnya, yaitu dengan mengetikkan perintah **man**.

man adalah perintah untuk menampilkan manual dari suatu perintah.

Cara untuk menggunakannya adalah dengan mengetikkan **man** diikuti dengan **perintah** yang ingin diketahui manual pemakaiannya. Contoh:

\$ man ls

Perintah di atas digunakan untuk menampilkan bagaimana cara penggunaan perintah **ls** secara lengkap.

Berikut adalah daftar perintah secara alfabet. Sebenarnya, pengguna dapat saja menekan tab dua kali untuk melihat semua kemungkinan perintah yang dapat digunakan. Misalnya ingin mengetahui perintah apa saja yang dimulai dengan huruf a, maka cukup mengetikkan **a** lalu tekan tab dua kali.

3.10.1. Perintah &

Perintah **&** digunakan untuk menjalankan perintah di belakang (*background*). Contoh :

wget http://id.wikibooks.org &

Perintah **&** dipakai di belakang perintah lain untuk menjalankannya di *background*. Jalan di *background* maksudnya adalah membiarkan sistem untuk menjalankan perintah sendiri tanpa partisipasi pengguna dan membebaskan *shell/ command prompt* agar bisa dipergunakan menjalankan perintah yang lain.

3.10.2. adduser

Perintah **adduser** digunakan untuk menambahkan user. Biasanya hanya dilakukan oleh *root* untuk menambahkan *user* atau akun yang baru. Setelah perintah ini bisa dilanjutkan dengan perintah **passwd**, yaitu perintah untuk membuat *password* bagi *user* tersebut.

Contoh:

```
# adduser hanan
# passwd anakdesa#
```

Perhatikan bahwa semua perintah yang membutuhkan akses *root*, ditulis dengan menggunakan tanda *#*, agar lebih mudah membedakannya dengan perintah yang tidak perlu akses *root*. Jika menjalankan perintah **adduser**, pengguna akan diminta memasukkan *password* untuk *user* yang dibuat. Isikan *password* untuk *user* baru tersebut dua kali dengan kata yang sama.

3.10.3. Alias

Digunakan untuk memberi nama lain dari sebuah perintah. **Alias** digunakan untuk memudahkan agar tidak harus mengetikkan perintah yang panjang, tapi cukup aliasnya saja.

Misalnya bila ingin perintah **ls** dapat juga dijalankan dengan mengetikkan perintah **dir**, maka buatlah aliasnya :

```
$ alias dir=ls
```

Kalau Anda suka dengan tampilan berwarna-warni, cobalah bereksperimen dengan perintah berikut:

```
$ alias dir="ls -ar --color:always"
```

Untuk melihat perintah-perintah apa saja yang mempunyai nama lain saat itu, cukup ketikkan **alias** saja (tanpa argumen).

3.10.4. apt-cache

Untuk mencari aplikasi yang telah di- *download* oleh **apt-get**.

```
$ apt-cache search nginx
```

3.10.5. apt-get

Untuk meng- *install* atau *uninstall* aplikasi di linux, sebagian keluarga linux yang menggunakan adalah Debian dan turunannya.

Contoh untuk meng- *install* aplikasi nginx:

```
$ apt-get install nginx
```

Untuk mengecek pembaruan dari aplikasi yang terinstall di linux.

```
$ apt-get update
```

Untuk memperbarui aplikasi di linux, biasanya digunakan setelah **apt-get update**.

```
$ apt-get upgrade
```

3.10.6 cat

Menampilkan isi dari sebuah *file* di layar. Contoh:

```
$ cat /nama/suatu/file
```

3.10.7 cd

Berpindah lokasi folder. Contoh:

```
[ubuntu@z-tion]$ cd /usr/X11R6/bin
```

```
[ubuntu@z-tion bin]$ pwd
```

3.10.8 chgrp

Perintah ini digunakan untuk merubah kepemilikan kelompok *file* atau *direktori*. Misalnya untuk memberi ijin pada kelompok atau grup agar dapat mengakses suatu *file*. *Sintaks* penulisannya adalah sbb:

```
# chgrp <grup baru> <file>
```

3.10.9 chmod

Digunakan untuk menambah dan mengurangi izin pemakai untuk mengakses *file* atau direktori. Ada tiga jenis *permission* / perijinan yang dapat dirubah yaitu: **r** untuk *read*, **w** untuk *write*, dan **x** untuk *execute*.

Dengan menggunakan *letter coding*, dapat dirubah *permission* diatas untuk masing-masing **u** (*user*), **g** (*group*), **o** (*other*) dan **a** (*all*) dengan hanya memberi tanda plus (+) untuk menambah izin dan tanda minus (-) untuk mencabut izin.

Misalnya untuk memberikan izin baca dan eksekusi *file* **coba1** kepada *owner* dan *group*, perintahnya adalah:

```
$ chmod ug+rx coba1
```

Untuk mencabut izin-ijin tersebut:

```
$ chmod ug-rx coba1
```

Dengan menggunakan sitem *numeric coding*, *permission* untuk *user*, *group* dan *other* ditentukan dengan menggunakan kombinasi angka-angka 4, 2 dan 1, dimana 4 (*read*), 2 (*write*) dan 1 (*execute*).

Misalnya untuk memberikan izin baca(4), tulis(2) dan eksekusi(1) file **coba2** kepada *owner*, perintahnya adalah:

```
$ chmod 700 coba2
```

```
testing dicoba
```

```
$ chmod 644 coba3
```


3.10.10 chown

Perintah **chown** digunakan untuk merubah *user ID* (*owner*) sebuah *file* atau *direktori*

```
$ chown <user id> <file>
```

3.10.11 cp

Perintah **cp** digunakan untuk menyalin *file* atau *copy*. Misalnya untuk menyalin *file1* menjadi *file2*:

```
$ cp <file1> <file2>
```

3.10.12 fg

Perintah ini digunakan mengembalikan suatu proses yang dihentikan sementara (*suspend*) agar berjalan kembali di *foreground*.

3.10.13 find

Perintah ini akan mencari *file* sesuai dengan kriteria yang ditentukan. Sintaksnya adalah perintah itu sendiri diikuti dengan nama *folder* awal pencarian, kemudian nama *file* dan terakhir menentukan bagaimana hasil pencarian itu akan ditampilkan. Misalnya akan dicari semua *file* yang berakhiran *.doc* di *current direktori*serta tampilkan hasilnya di layar:

```
find . -name *.doc -print
```

Contoh hasil:

```
./public/docs/account.doc
```

```
./public/docs/balance.doc
```

```
./public/docs/statistik/prospek.doc
```

```
./public/docs/statistik/presconf.doc
```

3.10.14 grep

Global Regular Expression Parse atau **grep** adalah perintah untuk mencari baris-baris yang mengandung teks dengan kriteria yang telah ditentukan pada *file* yang diberikan. Format perintah:

```
$ grep [opsional] <teks> <file>
```

Misalnya akan dicari *file*- *file* yang mengandung teks *marginal* di *current direktori*:

```
$ grep marginal <file>
```

3.10.15 gzip

Ini adalah *software* kompresi *zip* versi GNU, fungsinya untuk mengkompresi sebuah *file*. Sintaksnya sangat sederhana:

```
$ gzip <namafile>
```

Walaupun demikian bisa diberikan parameter tertentu bila memerlukan kompresi *file* yang lebih baik

3.10.16 halt

Perintah ini hanya bisa dijalankan oleh *super user* atau pengguna harus *login* sebagai *root*. Perintah ini untuk memberitahu *kernel* supaya mematikan sistem atau *shutdown*.

3.10.17 hostname

Untuk menampilkan *host* atau *domain name* sistem dan bisa pula digunakan untuk mengatur nama *host* sistem. Contoh pemakaian:

```
[user@localhost mydirectoryname]$
```

```
hostname localhost.localdomain
```

3.10.18 kill

Perintah ini akan mengirimkan sinyal ke sebuah proses yang ditentukan. Tujuannya adalah menghentikan proses. Format penulisan:

```
$ kill <sinyal> <pid>
```

PID (*Proportional Integral Derivative*) adalah nomor proses yang akan di hentikan.

3.10.19 less

Fungsinya kebalikan perintah **more**.

3.10.20 login

Untuk masuk ke sistem dengan memasukkan *login ID* atau dapat juga digunakan untuk berpindah dari *user* satu ke *user* lainnya.

3.10.21 logout

Untuk keluar dari sistem.

3.10.22 ls

Menampilkan isi dari sebuah direktori seperti perintah **dir** di DOS. Dapat digunakan beberapa *option* yang disediakan untuk mengatur tampilannya di layar. Bila perintah ini dijalankan tanpa *option* maka akan ditampilkan seluruh *file nonhidden* (*file* tanpa awalan tanda titik) secara alfabet dan secara melebar mengisi kolom layar. *Option -la* artinya menampilkan seluruh *file/ all*

termasuk *file hidden* (*file* dengan awalan tanda titik) dengan format panjang.

3.10.23 mesg

Perintah ini digunakan oleh *user* untuk memberikan izin *user* lain menampilkan pesan dilayar *terminal*. Misalnya *mesg user* dalam posisi *y* maka *user* lain bisa menampilkan pesan di layar dengan *write* atau *talk*.

```
$ mesg y atau mesg n
```

Gunakan **mesg n** bila tidak ingin diganggu dengan tampilan pesan-pesan dari *user* lain

3.10.24 mkdir

Membuat direktori baru, sama dengan perintah **md** di DOS.

Contoh :

```
$mkdir mahasiswa
```

3.10.25 more

Mem- *paging* halaman, seperti halnya **less**

3.10.26 mount

Perintah ini akan me- *mount filesystem* ke suatu direktori atau *mount point* yang telah ditentukan. Hanya *super user* yang bisa menjalankan perintah ini. Untuk melihat *filesystem* apa saja beserta *mount-point*nya saat itu, ketikkan perintah **mount**.

```
$ mount
```

```
/dev/hda3 on / type ext2 (rw)
```

```
none on /proc type proc (rw)
```

```

/dev/hda1 on /dos type vfat (rw)
/dev/hda4 on /usr type ext2 (rw)
none on /dev/pts type devpts (rw,mode=0622)

```

3.10.27 mv

Untuk memindahkan *file* dari satu lokasi ke lokasi yang lain. Bila argumen yang kedua berupa sebuah *folder* maka *mv* akan memindahkan *file* ke *folder* tersebut. Bila kedua argumen berupa *file* maka nama *file* pertama akan ditimpa *file* kedua. Akan terjadi kesalahan bila *user* memasukkan lebih dari dua argumen kecuali argumen terakhir berupa sebuah direktori.

3.10.28 nano

Salah satu *editor* di linux, seperti halnya Notepad di Windows. Contoh untuk mengedit suatu *file*.

```
$ nano /etc/apt/sources.list
```

Setelah selesai untuk menyimpan menggunakan ctrl+o dan untuk keluar menggunakan ctrl+x.

3.10.29 passwd

Digunakan untuk mengganti *password*. *User* akan selalu diminta mengisi *password* lama dan selanjutnya akan diminta mengisi *password* baru sebanyak dua kali. *Password* sedikitnya terdiri dari enam karakter dan sedikitnya mengandung sebuah karakter.

3.10.30 Pwd

Print working directory untuk menampilkan nama direktori dimana *user* saat itu sedang berada.

```
$ pwd
```

Contoh Hasil:

```
/home/user
```

3.10.31 rm

rm atau **remove** digunakan untuk menghapus *file*. Secara *default*, tidak menghapus direktori. Contoh Penggunaan:

```
$ rm [opsi] <file>
```

3.10.32 rmdir

rmdir atau **remove directory(ies)** digunakan menghapus *folder*, apabila kosong. Contoh Penggunaan:

```
$ rmdir <direktori>
```

3.10.33 shutdown

Perintah ini untuk mematikan sistem, seperti perintah **halt**. Pada beberapa sistem, *user* bisa mematikan komputer dengan perintah **shutdown -h now** dan me- *restart* sistem dengan perintah **shutdown -r now** atau dengan kombinasi tombol Ctr-Alt-Del.

3.10.34 su

Untuk *login* sementara sebagai *user* lain. Bila *user ID* tidak disertakan maka komputer menganggap *user* ingin *login* sementara sebagai *super user* atau *root*. Bila *user* bukan *root* dan *user* lain itu memiliki *password* maka *user* harus memasukkan *password*-nya

dengan benar. Tapi bila *user* adalah *root* maka *user* dapat *login* sebagai *user* lain tanpa perlu mengetahui *password* *user* tersebut.

3.10.35 tar

Menyimpan dan mengekstrak *file* dari media seperti *tape drive* atau *hardisk*. *File* arsip tersebut sering disebut sebagai *file tar*. Sintaknya sebagai berikut:

```
$ tar <aksi> <option> <file atau direktori>
```

Contoh:

```
$ tar -czvf namaFile.tar.gz /nama/direktori/*
```

Perintah di atas digunakan untuk memasukkan semua isi *folder*, lalu dikompres dengan format tar lalu di zip dengan gzip, sehingga menghasilkan sebuah *file* bernama **namaFile.tar.gz**

```
$ tar -xzvf namaFile.tar.gz
```

Perintah di atas untuk mengekstrak *file* **namaFile.tar.gz**

3.10.36 unmount

Adalah kebalikan dari perintah **mount**, yaitu untuk meng-*unmount filesystem* dari *mount-point*-nya. Setelah perintah ini dijalankan direktori yang menjadi *mount-point* tidak lagi bisa digunakan.

```
# umount <filesystem>
```

3.10.37 unalias

Kebalikan dari perintah **alias**, perintah ini akan membatalkan sebuah alias. Jadi untuk membatalkan **alias** **dir** seperti telah dicontohkan diatas, gunakan perintah:

```
$ unalias dir
```

3.10.38 unzip

Digunakan untuk menguraikan *file* yang dikompres dengan **zip**. Sintaknya sederhana dan akan mengekstrak *file* yang ditentukan:

```
$ unzip <namafile>
```

3.10.39 who

Perintah ini akan menampilkan informasi mengenai *login name*, jenis *terminal*, waktu *login* dan *remote hostname* untuk setiap *user* yang saat itu sedang *login*. Misalnya :

```
$ who

root tty0 May 22 11:44
flory tty2 May 22 11:59
pooh typ3 May 22 12:08
```

3.10.40 zip

Digunakan untuk mengompres *file* yang ditentukan.

```
$ zip <namafile>
```

3.11. Diagram Alir (*Flowchart*)

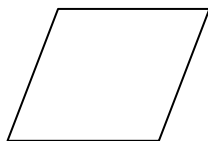
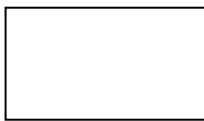
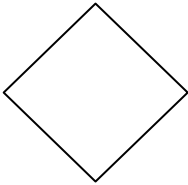
Diagram alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Diagram alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Salah satu jenis *flowchart* adalah bagan alir program. Bagan alir ini terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*),

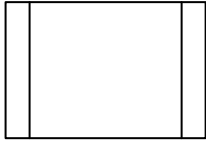
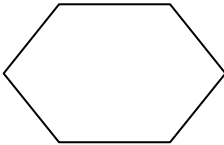

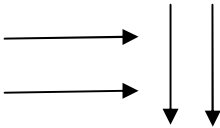
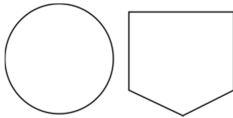
dan bagan alir program komputer terinci (*detailed computer program flowchart*).

Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah didalam program komputer secara logika. Bagan alir logika program ini dipersiapkan oleh analis sistem, sedangkan bagan alir program komputer terinci (*detailed computer program flowchart*) digunakan untuk mempersiapkan instruksi-instruksi program komputer secara terinci. Bagan alir ini dipersiapkan oleh pemrogram.

Diagram alir program dibuat dengan menggunakan simbol-simbol sebagai berikut :

Tabel 3.1 Program *Flowchart*

NO	SIMBOL	KETERANGAN
1.		Input/Output , digunakan untuk mewakili data input/output
2.		Proses , digunakan untuk mewakili suatu proses
3.		Keputusan , digunakan untuk penyeleksian suatu kondisi dalam suatu program

4.		Proses Terdefinisi , menunjukan suatu operasi yang rincian dilakukan ditempat lain dalam prosedur tertentu
5.		Persiapan , digunakan untuk menunjukan nilai awal suatu variable
6		Terminal , menunjukan awal dan akhir suatu program
7		Garis Alir , arus dari proses
8		Konektor , penghubung ke halaman yang sama atau ke halaman yang lain