

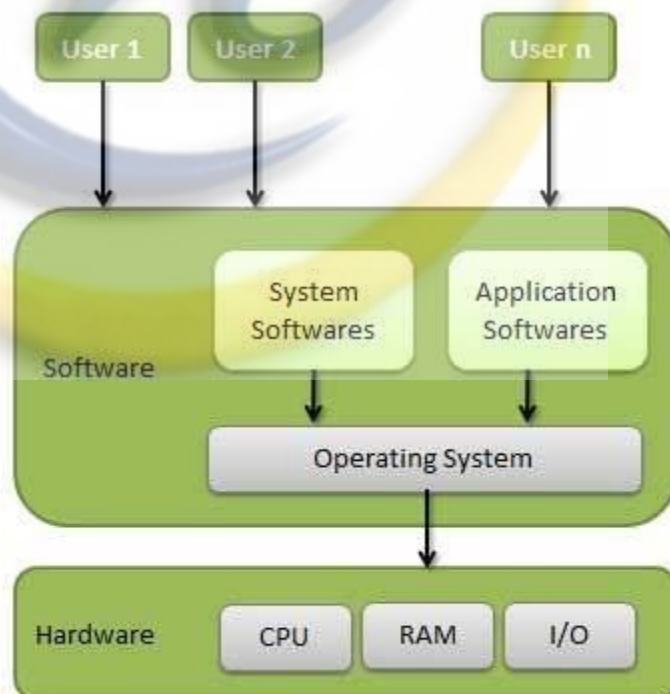
BAB II

LANDASAN TEORI

2.1 Sistem Operasi

Perangkat keras (*hardware*) dihubungkan dengan pengguna (*brainware*) melalui sistem operasi. Pengelola seluruh sumber daya yang terdapat pada sistem komputer serta penyedia sekumpulan layanan (*system calls*) kepada *user* guna mempermudah dan memberikan kenyamanan penggunaan serta pemanfaatan sumber daya sistem komputer merupakan pengertian umum dari sistem operasi.

Sistem operasi atau *operating system* disingkat *OS* merupakan perangkat lunak sistem yang memiliki fungsi kontrol serta manajemen terhadap perangkat keras, serta operasi-operasi dasar sistem, termasuk menjalankan aplikasi seperti aplikasi pengolah kata, pengolah gambar, pengolah suara, *web browser* dan lain sebagainya.



Gambar 2.1.1 Sistem Komputer [8]

Sistem operasi disimpan dalam memori komputer yang merupakan *software* lapisan pertama pada saat komputer dinyalakan. *Software* lainnya seperti *software* aplikasi akan dijalankan setelah sistem operasi bekerja. Layanan inti umum yang dibutuhkan oleh *software* aplikasi akan dikerjakan oleh sistem operasi. Layanan inti umum yang dimaksud seperti akses ke *disk*, manajemen memori, penjadwalan tugas (*scheduling task*), dan *interface* pengguna sehingga masing-masing *software* aplikasi tidak perlu melakukan tugas-tugas inti umum tersebut.

Bila dirinci, beberapa tugas penting yang dimiliki oleh sistem operasi adalah sebagai berikut :

1. Manajemen memori

Memori memiliki *words array* atau *bytes array* yang sangat besar, dan setiap *words array* atau *bytes array*-nya memiliki *address* masing-masing. Sistem operasi memiliki peran dalam melakukan manajemen memori diantaranya melakukan pemantauan memori primer, sistem operasi harus dapat mengetahui bagian memori mana yang sudah digunakan, oleh siapa digunakan, dan bagian mana yang belum digunakan.

Dalam *multiprogramming*, sistem operasi memilah mana proses yang membutuhkan memori serta mengetahui berapa besar memori yang dibutuhkan. Sehingga merupakan tugas sistem operasi pula dalam mengalokasikan memori saat suatu proses membutuhkan penggunaan memori, dan menarik kembali alokasi memori saat suatu proses tidak lagi membutuhkan memori.

2. Manajemen prosesor

Saat *user* melakukan *multiprogramming*, sistem operasi berfungsi memilah proses mana yang membutuhkan penggunaan *processor* serta berapa lama proses itu membutuhkan *processor*, itulah yang disebut dengan *process scheduling*. Sistem operasi harus memantau *processor* serta status dari proses yang berlangsung, fungsi ini disebut *traffic*

controller. Pengalokasian penggunaan *CPU* pun merupakan bagian tugas sistem operasi dalam manajemen proses.

3. Manajemen perangkat/*device*

Dalam fungsi manajemen *device* , sistem operasi memantau serta mengontrol seluruh *devices* yang terhubung hal ini sering disebut *I/O controller*. Sistem operasi pun harus memilah proses mana yang membutuhkan *device* serta melakukan alokasi *device* dengan cara yang efisien.

4. Manajemen File

Beberapa aktifitas sistem operasi dalam melakukan manajemen file antara lain melakukan kontrol *file system* serta pengalokasian *resources*.

5. Aktifitas-aktifitas penting lainnya

Aktifitas-aktifitas ini seperti keamanan, melakukan kontrol *system performance*, *job accounting*, deteksi *error*, serta melakukan koordinasi antara *software* dengan *user*.

2.1.1 Ragam Sistem Operasi

Semenjak komputer generasi pertama hingga sekarang ini, sistem operasi ada dan masih terus berkembang. Ada beberapa jenis sistem operasi yang sering digunakan, diantaranya :

1. Sistem Operasi *Batch*

Sistem operasi *Batch* ini tidak memungkinkan *user* berinteraksi langsung dengan komputer. *Off-line device* seperti *punch cards* menjadi perantara antara *user* dengan komputer, *user* memberi instruksi pada komputer dengan cara memasukan *punch card* kepada operator komputer. Namun jenis sistem operasi ini memiliki beberapa kekurangan, diantaranya kurangnya interaksi antara *user* dengan *job* yang diinstruksikan, seringkali *CPU* berstatus *idle* atau tidak digunakan, hal ini karena kecepatan mekanik *I/O devices* jauh lebih lambat dibandingkan *CPU*.

2. Sistem Operasi *Time-sharing*

Dalam sistem operasi ini memungkinkan banyak *user* menggunakan satu sistem komputer dalam waktu yang bersamaan. Waktu milik *processor* dibagi kepada beberapa *user* secara simultan. Bila sistem operasi *batch* bertugas dengan memaksimalkan penggunaan *processor*, sistem operasi *time-sharing* berfungsi dengan meminimalisir waktu respon.

Pekerjaan yang banyak dieksekusi oleh *CPU* dengan cara saling bergantian. *User* menerima respon dengan cepat karena pergantian *jobs* dilakukan dengan frekuensi yang cepat. Sistem operasi ini bisa dikatakan lebih baik dari pada sistem operasi sebelumnya. Sistem operasi ini memiliki beberapa kelebihan seperti menyediakan respon yang cepat, menghindari duplikasi dari *software*, dan mengurangi *CPU idle time* yang sebelumnya terjadi di sistem operasi *batch*. Namun sistem operasi ini pun tidak luput dari beberapa kekurangan, diantaranya masalah dalam reliabilitas, keamanan yang kurang, serta masalah dalam komunikasi data.

3. Sistem Operasi *Distributed*

Sistem operasi *distributed* menggunakan lebih dari satu *CPU* untuk menyediakan lebih dari satu *real-time application* dan *user* yang banyak. Komunikasi *lines* seperti *high-speed buses* menjadi perantara komunikasi antara *processor* dengan yang lainnya. Beberapa kelebihan dari sistem operasi ini antara lain bisa saling berbagi sumber daya, kecepatan pergantian data yang meningkat, pelayanan yang lebih baik terhadap *user*, serta mengurangi *delay* dalam pemrosesan data.

4. Sistem Operasi *Network*

Fungsi utama dari sistem operasi ini adalah untuk menyediakan layanan yang mampu melakukan manajemen data, *user*, *group*,

keamanan, aplikasi-aplikasi, serta berbagai fungsi jaringan atau *networking*. Dapat berbagi file dan akses kepada printer diantara banyaknya *user* yang terhubung merupakan tujuan utama dari adanya sistem operasi *network*.

Beberapa contoh dari sistem operasi *network* antara lain *Microsoft Windows Server, Unix, Linux, Mac OS X, Novell NetWare, BSD* dan lain sebagainya.

Kelebihan dari sistem operasi ini antara lain stabilnya *server* yang disentralkan, keamanan dapat di-*manage* oleh *server*, teknologi yang selalu ter-*upgrades* sehingga *hardware* dapat dengan mudah terintegrasi dengan sistem, serta adanya fitur *remote access* yang memungkinkan dalam mengakses server dari lokasi yang berbeda.

Namun, sistem operasi ini pun memiliki beberapa kekurangan, diantaranya harga yang tinggi, bergantung pada *central location* untuk melakukan fungsi operasi yang banyak, serta dibutuhkannya *maintenance* dan *updates*.

5. Sistem Operasi *Real Time*

Pada sistem operasi *real-time*, sistem pemrosesan data dengan interval yang dibutuhkan dalam melakukan proses serta respon sangatlah kecil. Sistem ini bekerja dengan mengutamakan respon terhadap inputan-inputan yang masuk, respon akan ditampilkan dengan cepat.

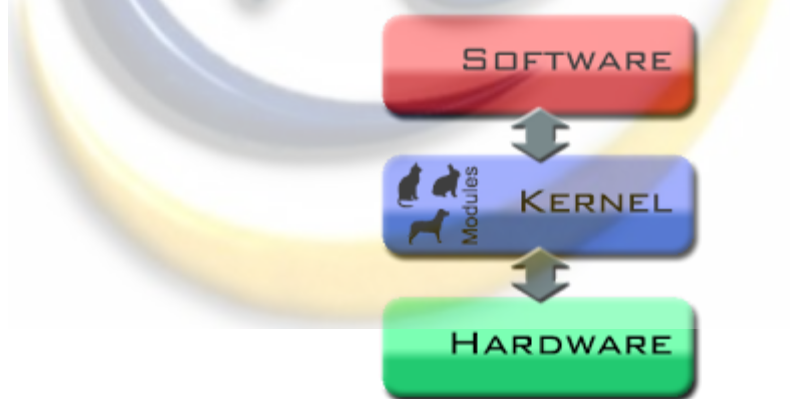
Real time system digunakan sebagai kontrol *device* pada alat atau aplikasi yang terhubung. Contohnya seperti, sistem *medical imaging*, sistem kontrol industry, sistem senjata, robot, dan lain sebagainya. Terdapat dua jenis dari sistem operasi *real-time*, diantaranya *Hard real-time systems* dan *Soft Real-Time Systems*. *Hard Real-Time Systems* menjamin tugas-tugas penting terselesaikan dengan tepat waktu. Dalam sistem ini, penyimpanan sekunder terbatas bahkan tidak ada dan data disimpan di dalam *ROM*. Sedangkan pada *Soft Real-time*

Systems sedikit terbatas, contohnya seperti multimedia, *virtual reality*, dan lain sebagainya. [8]

2.1.2 Kernel

Kernel merupakan inti dari suatu sistem operasi dan merupakan bagian kode yang melakukan tugas-tugas inti umum suatu sistem operasi. Tugas dan fungsi kernel yaitu :

1. Melakukan koordinasi aliran informasi antara perangkat *input* dan *output*.
2. Membaca serta merespon perintah-perintah yang diberikan oleh *user* melalui perangkat *input*.
3. Melakukan pengaturan program-program lain yang dijalankan oleh *user*.
4. Membawa program serta data yang diberikan oleh *user* kepada memori komputer dan memastikan prosesor bekerja dengan benar. [9]



Gambar 2.1.2 Kernel

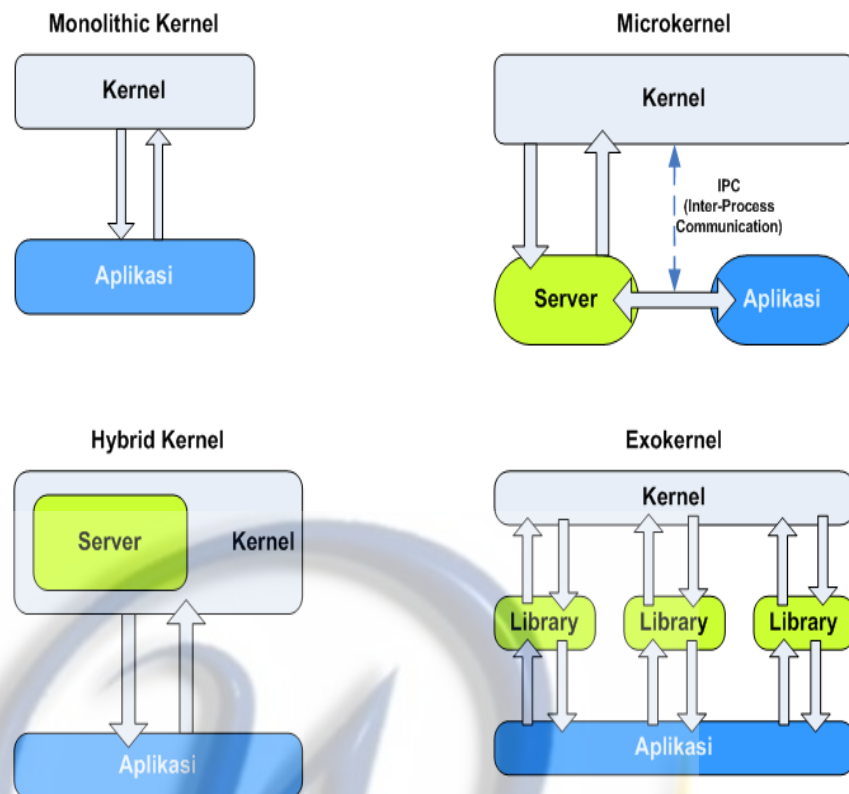
Service dasar dari sistem operasi dijalankan oleh kernel. *Service* dasar tersebut diantaranya seperti manajemen memori, manajemen proses, manajemen file, serta manajemen *I/O devices*. Isi kernel tergantung dari sistem operasinya, umumnya terdapat *scheduler*, *supervisor*, *interrupt handler*, serta *memory manager* guna menjalankan *services* dasar tersebut.

Fungsi membagi waktu dan urutan proses yang ada dilaksanakan oleh *scheduler*. Kemudian proses yang sudah dijadwalkan akan dilayani oleh *supervisor*. Seluruh permintaan dari *hardware* yang ingin dilayani oleh kernel ditangani oleh *interrupt handler*. Sedangkan pengaturan alokasi *memory address* dilakukan oleh *memory manager*.

Sebuah kernel di-*design* untuk sebuah sistem operasi yang spesifik. Semua sistem operasi memiliki kernel didalamnya. Sebagai contoh *Microsoft Windows 10* memiliki *design* kernel yang hanya dipakai oleh sistem operasi tersebut. Sedikit berbeda dengan *Linux*, *user* dapat melakukan modifikasi kernel sesuai dengan kebutuhan. Oleh karena itu *Linux* terkenal dengan istilah sistem operasi *open source* karena kernel yang ada dapat dimodifikasi, sedangkan sistem operasi lainnya seperti *Microsoft Windows* tidak dapat dimodifikasi.

2.1.2.1 Jenis-Jenis Kernel

Kernel memiliki beberapa perbedaan *design*, dan setiap *design* kernel tersebut memiliki kelebihan serta kekurangan dalam pengimplementasiannya. *Monolithic*, *microkernel*, *hybrid kernel*, dan *exokernel* merupakan jenis-jenis kernel yang ada.



Gambar 2.1.3 Ragam desain Kernel

1. *Monolithic Kernel*

Services dasar dari sistem operasi dimiliki seluruhnya oleh tipe kernel ini. *Monolithic kernel* memiliki performa sistem yang optimal karena kernel ini didisain dengan kelebihan efisiensi. Seluruh proses yang membutuhkan fungsi kernel dilakukan di dalam kernel. Namun kelemahan dari desain kernel ini adalah kurangnya stabilitas karena kemungkinan *crash* pada sistem besar terjadi. Contoh sistem operasi yang menggunakan desain kernel ini yaitu *Linux* dan *Microsoft Windows 9x*.

2. *Micro Kernel*

Desain kernel ini hanya menggunakan *services* dasar minimal yang diperlukan. *Services* dasar tersebut seperti manajemen pengalamatan memori, manajemen proses, serta *inter-process communication*. Dengan minimalnya *services* dasar kernel yang digunakan, stabilitas sistem *Micro Kernel* lebih baik

dibandingkan dengan *Monolithic Kernel*. Namun, komunikasi antar proses menjadi lebih rumit sehingga sistem kurang efisien, itulah yang menjadi kelemahan dari desain kernel ini. Contoh sistem operasi yang menggunakan desain kernel ini yaitu *Mac OS X*, *Minix*, *QNX*, dan lain sebagainya.

3. *Hybrid Kernel*

Hybrid Kernel merupakan *Micro Kernel* yang memiliki tambahan kode yang membuat kernel ini dapat berjalan lebih cepat dari *Micro Kernel*. Hal itu merupakan salah satu kelebihan yang dimiliki oleh desain kernel tersebut. Contoh sistem operasi yang berdesain *Hybrid Kernel* antara lain *Microsoft Windows NT*, *Windows 2000*, serta *Windows XP*.

4. *Exo Kernel*

Desain kernel ini masih dalam tahap penelitian dan merupakan desain kernel eksperimental, penggunaannya masih dalam lingkup tertentu. *Exo Kernel* memiliki fungsi perlindungan dan pembagian *resource* untuk *hardware*. Lebih dari satu *library* sistem operasi dapat dimasukkan kedalam desain kernel ini, hal ini membuat desain kernel ini mampu menjalankan program-program untuk sistem operasi yang berbeda dalam waktu yang sama. Contoh *Exo Kernel* terdapat pada *Nemesis* yang dikembangkan oleh *University of Cambridge*. [8]

2.2 Linux

Seorang mahasiswa Finlandia yang bernama Linus Torvalds, membuat sistem operasi *Linux*. Sistem operasi ini terinspirasi oleh *Minix* yang merupakan sistem *UNIX* kecil yang dikembangkan oleh Andrew Tanenbaum. Saat ini, *Linux* merupakan sistem *UNIX* yang sangat lengkap. *Linux* dapat digunakan untuk keperluan jaringan, pengembangan *software*, bahkan untuk pekerjaan sehari-hari. Sistem operasi ini menjadi alternatif selain sistem operasi komersial milik *Microsoft* dan *Mac OS*, hal ini dikarenakan harga sistem operasi *Linux* sangatlah murah bahkan gratis.

Perkembangan *Linux* sangatlah cepat, beragam kelompok di dunia mengembangkan sistem operasi ini. *Linux* bisa didapatkan dalam berbagai distribusi atau sering disebut dengan *Distro*. *Distro* merupakan kumpulan dari kernel *Linux*, sistem dasar *Linux*, program instalasi, *basic tools*, serta berbagai macam program yang dipasang sesuai dengan tujuan pembuatan *Distro* tersebut. Hingga saat ini *Distro Linux* sangatlah banyak, namun banyaknya *Distro Linux* tersebut dapat terwakili oleh beberapa *Distro Linux* berikut :

1. *Arch Linux*

Distribusi ini ditargetkan pada *user Linux* yang sudah berpengalaman. *Distro* turunannya seperti *Manjaro* merupakan *Distro Linux* yang cepat, *user friendly*, serta berorientasi *desktop*.

2. *Debian*

Debian merupakan *Distro* yang cukup populer, dari kalangan pengguna *Linux* yang berpengalaman, hingga pengguna *Linux* yang masih awam. *Distro* ini pun banyak menghasilkan turunan berupa *Distro baru*. Beberapa *Distro* populer turunan *Debian* yaitu *Canaima*, *Knoppix*, *Linux Mint Debian Edition*, dan *Ubuntu*. Yang menarik, turunan dari *Debian* yaitu *Ubuntu* merupakan *Distro Linux* yang sangat populer juga dan menghasilkan banyak turunan diantaranya *Backtrack*, *Linux Mint*, *Xubuntu*, *Kubuntu*, dan masih banyak lagi.

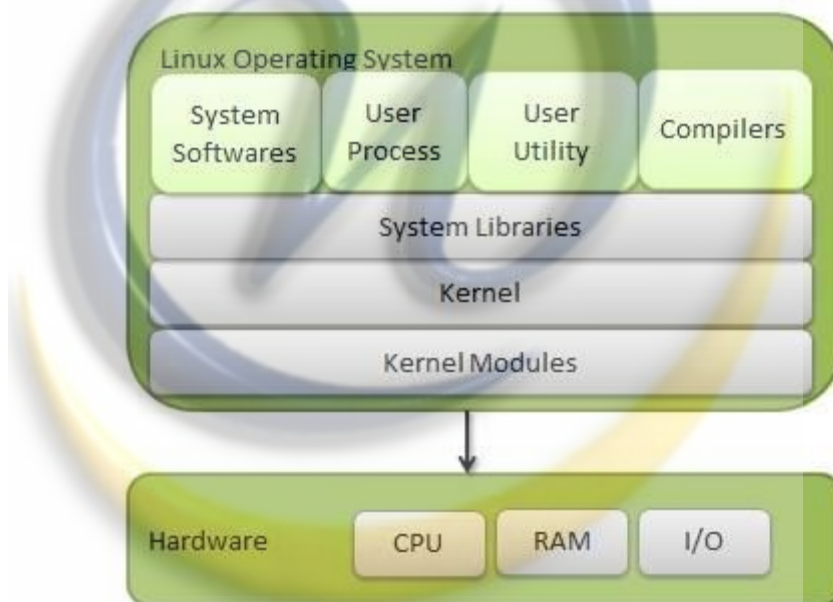
3. *Fedora*

Fedora merupakan distribusi komunitas yang disponsori oleh perusahaan Amerika yaitu *RedHat*. Hal itu membuat *Fedora* menghasilkan sistem operasi yang baik untuk server namun berbayar. *Fedora* memiliki beberapa turunan seperti *Red Hat Enterprise Linux (RHEL)* dan *Mandriva*. *RHEL* memiliki beberapa turunan lagi yaitu *CentOS*, dan *Oracle Enterprise Linux*

4. Slackware

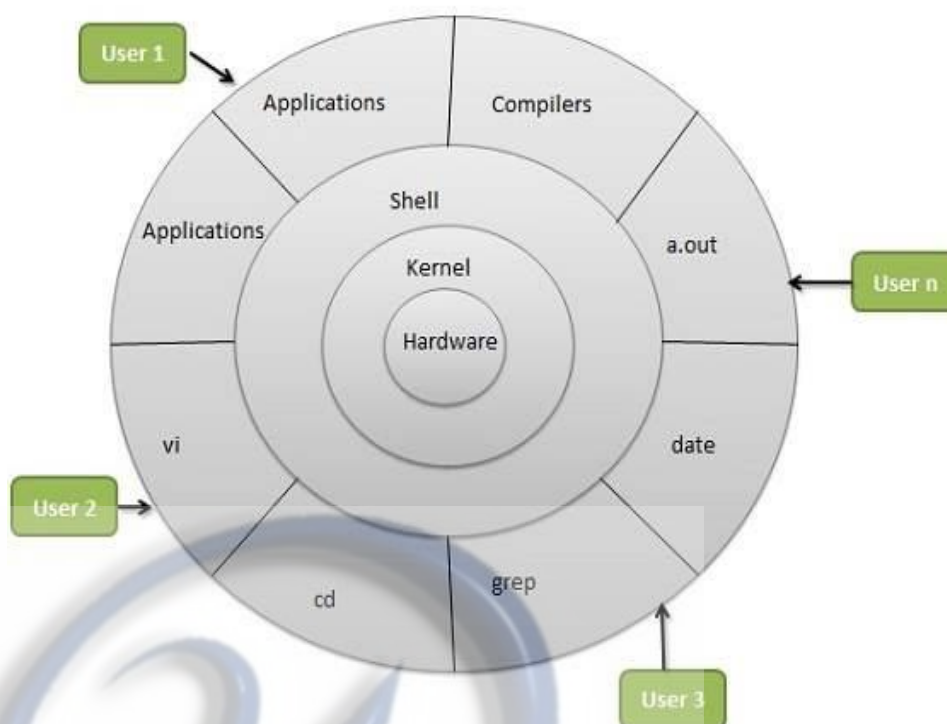
Distribusi ini merupakan salah satu distribusi tertua hingga saat ini. Terdapat turunan seperti *SUSE Linux Enterprise* yang berbayar karena dikelola oleh perusahaan *SUSE*. Adapula *OpenSUSE* yang merupakan turunan dari *SUSE Linux Enterprise* dengan versi gratis.

Sistem Operasi *Linux* memiliki komponen utama yaitu kernel, *system library*, dan *system utility*. Sebagai inti dari sistem operasi *Linux*. Kernel bertanggung jawab untuk semua aktifitas-aktifitas yang berjalan pada sistem operasi.



Gambar 2.2.1 Komponen sistem Linux [8]

System Library bertugas sebagai kumpulan kode yang akan menghubungkan *software* aplikasi atau pun inputan dari *user* untuk dapat menggunakan kernel. Sedangkan *system utility* bertanggung jawab melakukan tugas masing-masing. *Linux* sendiri memiliki arsitektur sistem yang terdiri dari beberapa layer, yaitu *Hardware layer*, *kernel*, *shell*, dan *utilities*.



Gambar 2.2.2 Arsitektur sistem Linux [8]

Pada lapisan paling dalam terdapat *hardware layer* yang terdiri dari berbagai macam perangkat keras yang terhubung. Di lapisan kedua terdapat *kernel* yang merupakan inti dari sistem operasi, *kernel* berinteraksi langsung dengan *hardware*. *Shell* terletak pada lapis ketiga, yang merupakan *interface* untuk dapat mengakses *kernel*. *Shell* menerima perintah-perintah dari *user* kemudian menghubungkannya ke *kernel* untuk dieksekusi. Pada lapis terakhir *utilities* menyediakan program-program yang lebih fungsional kepada user. [8]

2.3 Software

Perangkat lunak atau yang dikenal dengan *software* merupakan sekumpulan data elektronik yang diatur dan disimpan oleh komputer. Komputer menyimpan data elektronik berupa program atau instruksi yang akan menjalankan suatu perintah. Perintah dijalankan melalui *software* atau perangkat lunak yang ada. *Software* sendiri dikelompokkan menjadi beberapa kelompok diantaranya program utilitas, paket program, program

aplikasi, Bahasa pemrograman, bahkan sistem operasi pun masuk kedalam pengelompokan *software*.

2.3.1 Program Utilitas (*Utility Program*)

Program utilitas merupakan suatu perangkat lunak yang berfungsi membantu jalannya proses pada komputer atau suatu paket program. Program ini didisain guna membantu proses analisis, konfigurasi, optimasi, serta membantu proses pengelolaan sebuah komputer ataupun sistem. *Software* utilitas lebih memfokuskan fungsinya terhadap pengoptimasian fungsi dari infrastruktur yang terdapat dalam sebuah komputer. *Software* utilitas umumnya ditujukan bagi pengguna yang memiliki pemahaman atas cara kerja sistem komputer yang cukup baik.

Jenis-jenis program utilitas terbagia atas 2 yaitu:

1. Program Utilitas Internal (bawaan) Utility internal adalah utility yang sudah ada pada Windows, secara default sudah terinstall pada saat selesai menginstall Windows. Contoh : Norton utility
Contoh : Notepad (untuk menulis), Wordpad (untuk menulis), Windows Media Player (multimedia), Games, Calculator, Paint (menggambar) dan lain-lain. Utility internal banyak terdapat di Control Panel, misalnya fonts, add remove, printer, system dan lain-lain.

Contoh program utilitas:

- a. File Manager adalah Program utilitas yang melakukan fungsi-fungsi yang berkaitan dengan manajemen file, seperti memformat dan menyalin disk, menampilkan daftar file, memeriksa jumlah ruang yang digunakan dan ruang kosong pada media penyimpanan, mengorganisasikan, merename, menghapus, memindahkan dan mengurutkan file-file.
- b. Personal Firewall adalah program utilitas yang berfungsi untuk mendeteksi dan melindungi komputer dari usaha computer lain untuk masuk ke komputer tanpa izin.

Personal Firewall untuk mendeteksi dan mengamankan personal computer dari instruksi-instruksi yang tidak dikenal (unauthorized intrusions).

- c. Disk Defragmenter adalah program utilitas yang menata kembali file-file dan ruang-ruang kosong pada hardisk komputer agar sistem operasi dapat mengakses data dan program dapat berjalan lebih cepat. Manfaat Defrag adalah mengatur kembali tata-letak file-file yang rusak/kacau karena penghapusan dan penulisan, dan menghapus jejak file yang tidak hilang pada saat penghapusan. Defrag perlu dilakukan secara teratur untuk dalam waktu mungkin 1-2 minggu sekali.
- d. Utilitas Backup adalah suatu program utilitas yang memungkinkan pengguna untuk menyalin atau membackup file-file terpilih atau seluruh hard disk ke media lain seperti CD, DVD, dll Backup Utility untuk melakukan duplikasi file-file yang dipilih atau seluruh hard disk ke disk atau tape. Umumnya melakukan compress files pada saat melakukan backup sehingga membutuhkan tempat lebih kecil.
- e. Utilitas Diagnostik adalah untuk mengumpulkan informasi teknis mengenai hardware dan software tertentu dari komputer kemudian membuat laporan yang menggarisbawahi masalah-masalah yang teridentifikasi.
- f. Image Viewer adalah program utilitas yang memungkinkan pengguna untuk menampilkan, menyalin dan mencetak isi dari file gambar.
- g. Disk Scanner adalah utilitas yang diinstal pada sistem komputer untuk membaca kesalahan dalam rangka untuk mengoptimalkan kinerja sistem. Ini berarti mengoreksi kesalahan yang mencegah program berjalan dengan baik. Disk scanner mendeteksi dan memperbaiki

masalah-masalah fisik maupun logis pada hard disk serta mencari dan menghapus/ membersihkan file-file yang tidak diperlukan. Pada dasarnya, disk scanner meningkatkan memori system, dalam rangka untuk mencegah dari overloading sementara operasi.

- h. Uninstaller adalah program utilitas untuk menghapus program dan berkas-berkas yang berkaitan dengan file sistem.
- i. Screen Saver adalah utilitas yang menyebabkan perangkat layar tampilan menampilkan gambar bergerak atau layar kosong jika tidak ada aktivitas pada keyboard atau mouse selama waktu yang telah ditentukan.
- j. Kriptografi Utilitas dapat membantu dalam mengkonversi ke dalam kode bahasa biasa dan juga mengkonversi kode ke dalam bahasa biasa. Oleh karena itu, berguna dalam mengenkripsi dan mendekripsi file.

2. Program Utilitas External (Bukan Bawaan)

Utility External adalah utility yang bukan bawaan Windows, harus menginstall terlebih dahulu.

- a. Registry Cleaner Registry Cleaner adalah program utilitas yang berfungsi untuk membersihkan file-file yang menumpuk di registry yang membuat perlambatan pada komputer seperti booting lebih lama, sehingga jika kita telah meregistrinya maka akan menghilangkan perlambatan pada komputer. Contoh Registry Cleaner : Regcure. Cleaning System, digunakan untuk membersihkan file-file, registri, internet files yang rusak. Contoh : RegCleaner, JvPowerTools, Norton System Works.
- b. Anti virus (Virus Scanner) Antivirus adalah program untuk menscan, menghapus, dan berjaga-jaga dari serangan virus. Program ini akan selalu berjalan di

background/tray icon jika diaktifkan. Contoh Antivirus : McAfee, Norton, AntiVir, Panda, AVG, dan lain-lain.

- c. Spyware Remover Spyware Remover adalah program utilitas untuk mendeteksi dan menghapus spyware dan adware. Spyware adalah program yang ditempatkan pada sebuah komputer tanpa sepengetahuan penggunanya yang secara diam-diam akan mengumpulkan informasi mengenai pengguna terutama yang berhubungan dengan penjelajahan di web.
- d. Internet Filter Internet Filter adalah program yang menghapus atau menghentikan hal-hal tertentu agar tidak ditampilkan. Tiga jenis internet filter yang digunakan secara luas adalah anti spam (menghapus spam sebelum masuk ke inbox kita), popup blocker (menghentikan pop-up agar tidak ditampilkan pada halaman web) dan web filter (membatasi akses ke materi-materi tertentu pada web).
- e. Kompresi file Kompresi file adalah program utilitas yang berfungsi untuk menyusutkan ukuran file. File Compression Utility berfungsi untuk mengurangi ukuran file untuk menambah sedikit ruang dan memperbaiki performance. Compressed files biasa disebut dengan zipped files, Utility program yang umum digunakan: PKZIP , WinZip, Izarc, dll.
- f. CD/DVD Burning adalah utilitas untuk menulis dan memperbaiki masalah-masalah pada sistem operasi atau disk dan memperbaiki kinerja komputer. CD/DVD burning software berfungsi untuk menuliskan text, graphics, audio, dan file video ke recordable atau rewriteable CD atau DVD. Contoh : Ahead Nero 6

2.3.2 Utilitas Diagnostik

Utilitas diagnostik merupakan suatu teknik untuk mengumpulkan informasi teknis mengenai hardware dari suatu perangkat. Setelah informasi dikumpulkan maka informasi tersebut bisa dijadikan suatu laporan yang menggaris bawahi masalah-masalah yang teridentifikasi. Dalam utilitasi hardware, biasanya yang dihimpun merupakan informasi mengenai pemantauan kinerja dari empat resources inti yaitu *CPU*, *Memory*, *Network*, dan *Disk I/O*.

2.3.3 Tools Utilitas Diagnostik

Tools utilitas diagnostik merupakan suatu aplikasi yang memiliki fungsi – fungsi utilitas diagnostik. Beberapa *tools* utilitas diagnostik diantaranya sebagai berikut :

2.3.3.1 Bonnie++

Tool ini dikembangkan oleh Russel Coker, *Bonnie++* merupakan *benchmarking tool* untuk melakukan pengukuran *disk* serta *file system*. Dengan menggunakan *Bonnie++* dapat diperoleh nilai yang berguna untuk mewakili performansi dari *disk* atau *file system* yang diuji.

Bonnie++ memiliki fungsi untuk melakukan *benchmark* terhadap performansi *filesystem* dengan berbagai macam tugas. *Bonnie++* menjadi tolak ukur untuk melakukan sejumlah *test* performansi *harddisk* dan *file system*. *User* bisa memilih mana *test* yang penting serta melakukan pemilihan dalam membandingkan sistem yang berbeda saat dijalankan.

Dengan menggunakan *Bonnie++*, dapat melakukan *benchmarks* terhadap tiga hal yaitu kecepatan *read* data dan *write* data, jumlah tugas yang dapat dilakukan per detik, serta jumlah dari operasi file metadata yang ditampilkan per detik.

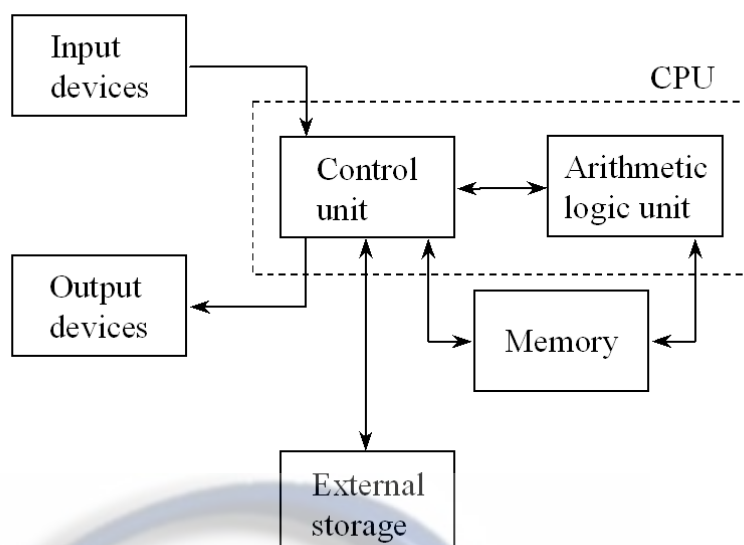
2.3.3.2 PSUTIL

psutil (sistem python dan utilitas proses) adalah perpustakaan lintas platform untuk mengambil informasi terkait menjalankan proses dan pemanfaatan sistem (*CPU*, memori, disk, jaringan, sensor) dengan Python. Hal ini berguna terutama untuk pemantauan sistem, pembuatan profil, keterbatasan sumber daya proses dan pengelolaan proses yang berjalan. Ini mengimplementasikan banyak fungsi yang ditawarkan oleh *command line tools* seperti: *ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap*. Saat ini mendukung Linux, Windows, OSX, Sun Solaris, FreeBSD, OpenBSD dan NetBSD, arsitektur 32-bit dan 64-bit, dengan versi Python dari 2.6 hingga 3.6 (pengguna Python 2.4 dan 2.5 dapat menggunakan versi 2.1.3). PyPy juga dikenal bekerja dalam sistem ini.

2.4 Hardware

Hardware atau perangkat keras yaitu komponen fisik dari sistem komputer yang mengandung sebuah papan sirkuit, *ICs*, atau komponen elektronik lainnya. Selain yang mengandung komponen elektronik, seluruh perangkat fisik yang dapat disentuh juga merupakan bagian dari perangkat keras komputer. Tanpa perangkat keras, komputer tidak akan pernah ada, serta perangkat lunak tidak bisa digunakan. [10]

Sistem komputer, bekerja atas adanya *hardware* serta *software*. Secara konsep, sistem komputer yang sederhana terdiri dari *CPU, memory*, serta *I/O devices* dan semua saling terhubung dan dapat berkomunikasi melalui sistem *bus*.



Gambar 2.4.1 Sistem Komputer

2.4.1 Central Prosesing Unit (CPU)

Sistem Komputer memiliki bagian fungsional utama yaitu *Central Prosesing Unit* atau dikenal dengan *CPU* yang merupakan otak dari sebuah komputer. Sistem Kerja yang dikerjakan oleh *CPU* antara lain yaitu membaca, mengkodekan dan mengeksekusi intruksi program, mengirim data dari dan ke memori, serta dari dan ke bagian input/output, serta merespon interupsi dari luar, menyediakan *clock* dan sinyal kontrol kepada sistem.

Terdapat dua bagian utama dari *CPU* yaitu kontrol unit (*control unit*) dan unit logika aritmatika (*arithmathic logic unit* atau disingkat *ALU*). *CPU* juga mempunyai *register* yang merupakan unit penyimpanan sementara.

2.4.1.1 CONTROL UNIT

Control Unit merupakan bagian dari prosesor yang memiliki fungsi diantaranya sebagai berikut :

1. Mengendalikan serta mengatur *I/O devices*
2. Mengambil instruksi-instruksi dari memori utama.

3. Mengambil data dari memori utama bila diperlukan oleh proses.
4. Mengirim instruksi ke *ALU* bila proses yang akan dilakukan terkait perhitungan aritmatik atau perbandingan logika
5. Menyimpan hasil proses ke memori utama.

2.4.1.2 ALU

ALU memiliki tugas utama sebagai pusat perhitungan aritmatik pada sistem computer. Operasi aritmatik yang dilakukan menggunakan dasar pertambahan, operasi lain pun seperti pengurangan, perkalian dan pembagian dilakukan dengan dasar penjumlahan, oleh karena itu sirkuit elektronik pada *ALU* disebut *adder*.

ALU memiliki tugas lain yaitu melakukan keputusan dari operasi logika sesuai dengan instruksi program. Berbagai operasi logika digunakan untuk melakukan perbandingan dua buah elemen logika.

2.4.1.3 Register

Fungsi utama dari register yaitu sebagai tempat penyimpanan temporer untuk instruksi atau data. Register melakukan fungsi menerima, menahan, dan mengirim instruksi atau data sesuai dari arahan Control Unit. Register memiliki beberapa peran diantaranya Accumulator yang berperan mengumpulkan hasil dari komputasi. Address Register, yang memantau kemana instruksi atau data diberikan, setiap lokasi penyimpanan di memory diidentifikasi dengan sebuah alamat. Storage Register, berfungsi sebagai tempat penyimpanan sementara dari data atau instruksi yang diambil atau akan dikirim ke memory

2.4.2 Memory

Bagian dari komputer yang berfungsi menyimpan data dan instruksi untuk diproses. Memory menyimpan instruksi program atau data hanya selama program membutuhkannya. Primary storage, primary memory, main storage, internal storage, main memory, dan RAM (Random Access

Memory) merupakan istilah lain dari memory. Data atau instruksi yang disimpan di dalam memory hanya bisa dilakukan saat program dalam kondisi running. Hal tersebut karena memory bekerja hanya saat computer dalam kondisi menyala. Saat lebih dari satu program dijalankan dalam waktu yang bersamaan, suatu program tidak bisa secara eksklusif mengklaim keseluruhan memory, artinya memory saling berbagi tempat untuk program yang berjalan.

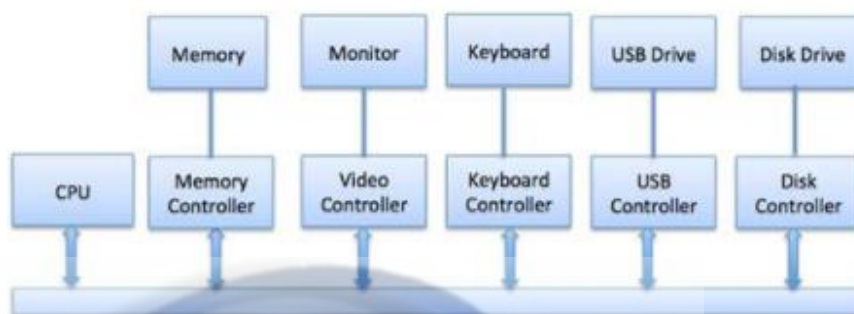
2.4.3 I/O System

Melakukan manajemen beberapa perangkat *I/O* seperti *mouse*, *keyboards*, *monitor*, dan lain sebagainya merupakan salah satu fungsi penting dari sistem operasi. *I/O system* dibutuhkan untuk mengambil *request* dari aplikasi *I/O* kemudian dikirimkan kepada perangkat fisik setelah itu menerima kembali respon apapun dari perangkat untuk dikirim ke aplikasi. Ada dua kategori *I/O devices* yaitu *block devices* dan *character evices*. Sebuah *block device* adalah dimana *driver* melakukan komunikasi dengan mengirim sekumpulan blok-blok data contohnya seperti *hard disks*, kamera *usb*, dan lain sebagainya. Sedangkan *character devices* adalah dimana *driver* melakukan komunikasi dengan mengirim dan menerima karakter-karakter *single (bytes, octets)* contohnya seperti *serial ports*, *paraller ports*, *sound cards*, dan lain sebagainya. [11]

2.4.4 Device Controllers

Perangkat-perangkat *I/O* atau *I/O devices* biasanya mengandung komponen – komponen mekanika dan komponen – komponen elektronika dimana komponen elektronik biasa disebut dengan *device controller*. Sebuah *device controller* bekerja layaknya *interface* diantara *device* dan *device driver*. *Device driver* merupakan modul-modul perangkat lunak yang bisa dimasukkan kepada sistem operasi guna menangani *device* tertentu. Sistem operasi mendapatkan bantuan dari *device drivers* untuk menangani seluruh perangkat *I/O*.

Selalu ada sebuah *device controller* dan sebuah *device driver* untuk setiap *device* dalam berkomunikasi dengan sistem operasi. Sebuah *device controller* memungkinkan untuk menangani banyak *devices*. Tiap *device* yang terhubung dengan komputer dihubungkan ke sebuah *device controller*.



Gambar 2.4.2 *Device Controller*

2.5 Cloud Computing

Cloud Computing esensinya merupakan tahap berikutnya dari inovasi dan adopsi *platform* untuk *computing*, *networking*, dan teknologi *storage* yang didesain untuk memberikan waktu yang cepat kepada pasar serta penghematan modal bagi penyedia layanan. [12]

Sedangkan menurut [13] *Cloud Computing* merupakan evolusi dari berbagai jenis teknologi yang telah datang bersama-sama untuk mengubah metode atau teknik suatu organisasi dalam membangun Infrastruktur IT.

Definisi *Cloud Computing* sendiri menurut *National Institute of Standards and Technology (NIST)* adalah model untuk menyediakan kemudahan akses bersama dalam berbagai sumber daya (*networks*, *servers*, *storage*, *applications*, dan *services*). [12]

NIST mendefinisikan esensinya 5 karakteristik dari *Cloud Computing* : [12]

1. Akses jaringan yang luas.
2. *On-demand self-service* (sesuai kebutuhan).
3. *Resource pooling*.
4. Layanan yang terukur.
5. Elastisitasnya cepat.

2.6 Virtualisasi

Pembuatan suatu bentuk atau versi virtual dari sesuatu yang bersifat fisik biasanya disebut dengan virtualisasi. Virtualisasi adalah teknologi inti dibalik infrastruktur cloud computing [2]. Teknologi ini memungkinkan pengalokasian resources atau sumber daya IT secara dinamis.

Virtualisasi melakukan isolasi terhadap satu sumber daya komputasi dengan sumber daya komputasi lainnya. Setiap elemen dalam suatu sumber daya komputasi dapat bekerja sendiri – sendiri dalam waktu yang bersamaan, hal ini dikarenakan di dalam logic stack sudah dipisahkan dalam bentuk layer-layer. Biasanya sebuah mesin virtual berisi sistem operasi beserta aplikasi yang kemudian dioperasikan secara independen dari sistem operasi yang terdapat langsung pada server fisik seolah – olah berada pada discrete hardware-nya sendiri. Beberapa mesin virtual bisa dijalankan pada sebuah server fisik, hal ini bertujuan untuk mengurangi server fisik.

Implementasi virtualisasi adalah sebagai berikut :

1. Virtualisasi presentasi, hal ini memungkinkan user untuk melakukan remote baik itu mengakses aplikasi dan sistem operasi pada sebuah host yang jauh. Biasanya digunakan pegawai suatu perusahaan dalam mengakses data perusahaan bila sedang tidak ada di depan lokasi host tempat data tersebut disimpan. User yang melakukan remote bisa memanipulasi data, mendapatkan hak akses ke dalam aplikasi, dan menggunakan sumber daya lain yang mungkin tersedia.
2. Virtualisasi aplikasi, memungkinkan provisioning dengan cepat serta membantu menghilangkan konflik yang berpotensi terjadi, dilakukan dengan cara mengisolasi aplikasi yang berjalan pada sistem operasi yang sama. Aplikasi biasanya melakukan update ke registry, namun dengan virtualisasi aplikasi, sebuah aplikasi dapat melakukan update ke virtual registry. Hal

ini membuat sistem mendapatkan kebutuhan aplikasi tersebut tanpa mengganggu aplikasi lainnya.

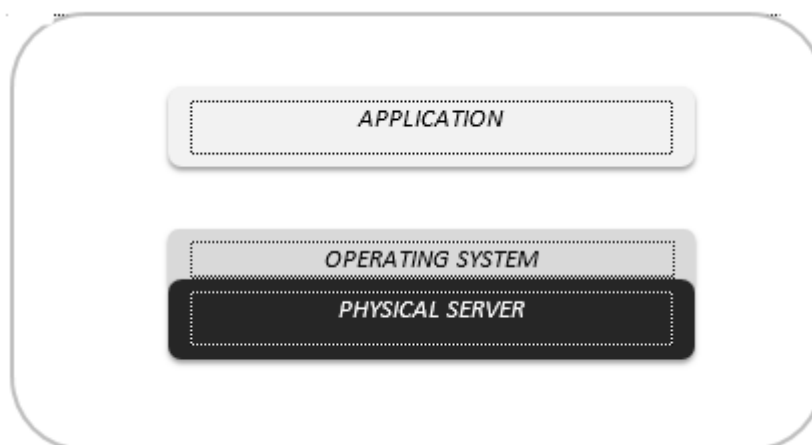
3. Virtualisasi desktop, memungkinkan setiap end user menentukan akses melalui sebuah remote graphics protocol serta dapat melakukan host desktop pada mesin virtual dalam data center. Aplikasi – aplikasi yang berkaitan dapat berjalan secara simultan pada desktop user dan memungkinkan pembuatan sistem operasi environment yang dipisah dengan desktop user.
4. Virtualisasi server, jenis virtualisasi ini bekerja dengan cara memisahkan sistem operasi yang secara logical diisolasi dari host server-nya. Virtualisasi server akan meningkatkan penggunaan resources (Hardware, utilities, space).

2.6.1 Virtualisasi Server

Ada beberapa cara dalam membangun virtualisasi *server* yaitu *Traditional Server (bare metal)*, *virtual machines*, dan *containers* [14]. Dibawah ini akan dijelaskan masing – masing dari ketiga cara dalam membangun virtualisasi *server*.

2.6.1.1 Traditional Server Setup (Bare Metal)

Traditional Server Setup merupakan solusi standar pembangunan server dalam beberapa tahun kebelakang. Sebelum *virtual machine* dan *container* mulai digunakan, metode *bare metal* ini lebih sering diimplementasikan pada banyak perusahaan – perusahaan. Aplikasi langsung dijalankan diatas sistem operasi yang di-*install* langsung diatas *server* fisik merupakan ciri utama dari *Traditional Server Setup*.



Gambar 2.6.1 Arsitektur Traditional Server

Kelebihan Traditional Server Setup (*Bare Metal*)

Ada beberapa keuntungan dari metode pembangunan *server* secara *bare metal*, diantaranya :

1. Pemanfaatan *resources* , dengan menggunakan metode *bare metal* maka akan membuat aplikasi yang ada dapat memanfaatkan lebih banyak jumlah sumber daya yang ada. Hal ini dikarenakan metode ini dalam menjalankan aplikasi lebih dekat dengan fisik *server* yang memiliki sumber daya tanpa adanya perantara seperti *hypervisor* dalam metode *virtual machine*.
2. *Isolation*, aplikasi dan data yang ada dapat terisolasi secara fisik dari aplikasi yang lain. Sebagai perbandingan, beberapa *container* dan *virtual machine* saling berbagi *environments*.

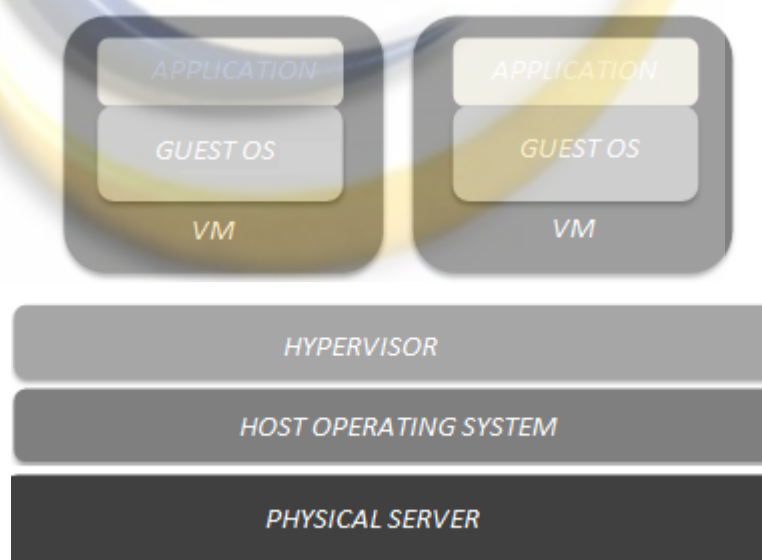
Kekurangan Traditional Server Setup (*Bare Metal*)

1. *Deployment time* yang sangat lambat, saat melakukan beberapa perubahan maka akan membutuhkan kehadiran fisik yang baru.
2. Mahal, merupakan masalah utama dari metode *bare metal* ini. *Server* fisik yang memiliki *resources* yang handal memiliki harga yang cukup mahal.

3. Sumber daya yang terbuang, hal yang sangat langka bila sebuah aplikasi membutuhkan seluruh *resources* yang disediakan oleh *server* fisik.
4. Sulit untuk dikembangkan, membeli *server* tambahan, melakukan *setting*, dan melakukan konfigurasi sangatlah memakan waktu.
5. Sulit dalam migrasi, memindahkan satu *server* fisik ke *server* fisik yang lain merupakan hal yang cukup sulit.
6. Konfigurasi yang kompleks, banyak *tools* yang harus disiapkan dalam melakukan konfigurasi *server* untuk sebuah aplikasi yang spesifik.

2.6.1.2 Virtual Machine (VM)

Metode *VM* mampu menjalankan beberapa *Guest* sistem operasi beserta aplikasi – aplikasinya dalam satu *server* fisik. *VM* dijalankan diatas *hypervisor* yang bertugas melakukan manajemen terhadap *VM*, membagi *resources* dan *network interface*.



Gambar 2.6.2 Arsitektur *Virtual Machine*

Kelebihan Virtual Machine (VM)

VM memiliki lebih banyak kelebihan bila dibandingkan dengan metode *bare metal*, kelebihanannya antara lain :

1. Penggunaan *resources* yang baik, hal ini dapat dilihat dari pembagian beberapa VM dari satu *server* fisik. *Resources* bisa dibagikan berdasarkan prioritas oleh *hypervisor* dan juga VM dapat menggunakan siklus yang sedang tidak terpakai oleh VM lainnya.
2. Mudah dikembangkan, aplikasi tambahan yang dibutuhkan bisa terus ditambahkan dengan cara menambah VM dengan catatan bahwa *resources* yang ada masih tersisa.
3. Mudah di *backup* dan migrasi, beberapa vendor VM telah menyediakan mekanisme untuk melakukan *backup*, *restore*, dan migrasi dengan lebih mudah.
4. Biaya yang lebih efisien, tentunya dibandingkan dengan harus membeli banyak *server* fisik maka biaya yang dikeluarkan bila menggunakan VM relatif lebih sedikit.
5. Fleksibilitas, arsitektur VM yang berbeda bisa berjalan secara bersama-sama.

Kekurangan Virtual Machine (VM)

1. Permasalahan dalam alokasi *resources*, masih sangat sulit untuk menjamin *resources* untuk spesifik VM saat semua *resources* dalam keadaan dibagikan.
2. *Vendor lockin*, maksudnya adalah disaat menggunakan salah satu solusi virtualisasi dari suatu *vendor* sangat sulit untuk pindah ke *vendor* yang lain.
3. Konfigurasi yang kompleks, hampir sama dengan metode *bare metal*, konfigurasi VM untuk beberapa aplikasi masih dinilai sangat kompleks.

2.6.1.3 Virtualisasi Berbasis Container (Container – based)

Penggunaan kernel dari sistem operasi milik *host* untuk menjalankan beberapa *guest instance (containers)* yang terisolasi merupakan kunci utama perbedaan virtualisasi *container – based* dengan metode virtualisasi lainnya. *Container – based* dinilai jauh lebih ringan daripada beberapa metode sebelumnya karena menggunakan kernel dari sistem operasi milik *host* serta tidak mengandalkan *hypervisor* untuk *me-manage resources*. *Container – based* juga memungkinkan membangun *container* yang terisolasi yang mana setiap *container* memiliki *root filesystem*, proses, *memory*, *devices*, dan *network interface*-nya sendiri – sendiri.



Gambar 2.6.3 Arsitektur *Container – Based*

Kelebihan Container – Based

1. *Isolation*, setiap aplikasi berjalan pada *container instance* nya sendiri.
2. *Lightweight* (ringan), *container – based* membutuhkan *resources* yang lebih sedikit dibandingkan dengan *bare metal* serta *VM*. *Container – based* juga tidak mengandalkan *hypervisor* tapi langsung menggunakan sistem operasi *kernel* milik *host*.

3. *Resources* yang efektif, hal ini dikarenakan *containers* menggunakan sistem operasi milik *host* sehingga tidak perlu sistem operasi *guest*.
4. Mudah dalam migrasi, hal ini karena *container* merupakan *instance* yang diisolasi dan tidak bergantung pada sistem operasi *guest*.
5. Keamanan, *container – based* memungkinkan dalam memiliki kewanamanan yang baik karena memiliki aplikasi – aplikasi yang masing – masing terisolasi.
6. *Overhead* rendah, hal ini dikarenakan *container – based* menggunakan *kernel* yang telah dibagi.
7. *Mirror production* dan *development environment*, tiap *container* yang berisi aplikasi di atasnya, dapat di *shipped* dengan mudah dari *local machine* kedalam *hosting environment*.
8. Komunitas, bagi *container – based* sendiri memiliki banyak komunitas untuk saling *share* dan membantu satu sama lain, salah satunya adalah komunitas *Docker*.

Kekurangan Container – Based

1. Arsitektur terbatas pada arsitektur yang sama, hal ini dapat dilihat dari *containers* yang saling berbagi kernel dengan sistem operasi *host*.
2. Aplikasi yang membutuhkan *resources* yang besar kurang cocok dengan metode *container – based* ini.

2.7 Dokcer

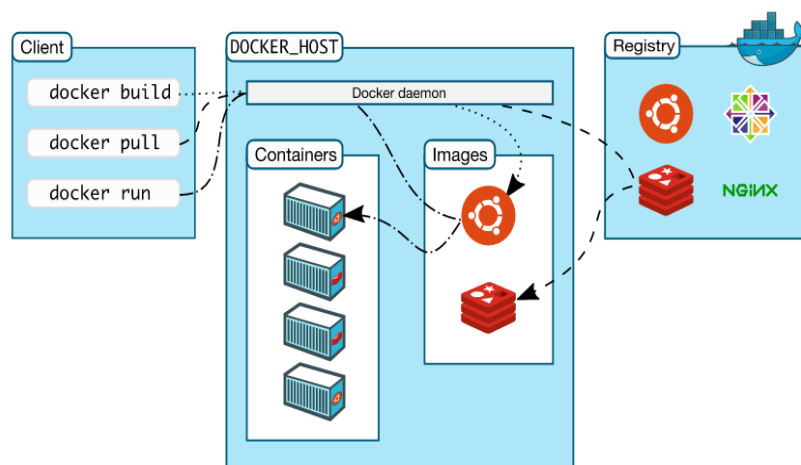
Docker merupakan project open-source yang memungkinkan para sysadmin maupun developer agar bisa membangun, mengemas serta menjalankan aplikasi dimana pun dalam sebuah container [6]. Docker adalah sebuah mesin open-source yang mengotomatisasi penyebaran aplikasi ke dalam containers. Hal itu ditulis oleh team di dalam Docker, Inc

dan dirilis oleh mereka dibawah lisensi Apache 2.0 [15]. Docker dapat berfungsi sebagai virtualisasi sebuah sistem operasi atau sebuah server atau database server ataupun sebuah web server. Dengan menggunakan docker diharapkan akan membatu developer dalam mengembangkan aplikasi sesuai dengan spesifikasi server.



Gambar 2.7.1 Logo *Docker* [6]

Docker menggunakan arsitektur client dan server, dimana Docker client akan mengirim request kepada Docker daemon untuk membangun, mendistribusikan, serta menjalankan container Docker. Docker client dan Docker daemon dapat berjalan dalam sistem yang sama, keduanya berkomunikasi via socket menggunakan RESTful API.



Gambar 2.7.2 Arsitektur *Docker* [6]

Docker terdiri dari beberapa komponen pembentuk, diantaranya yaitu *Docker Images*, *Docker containers*, serta *Registries*.

2.7.1 Docker Images

Dasar template untuk *docker container* dinamakan sebagai *Docker Images*. Sebuah *image* berisi sistem operasi dan juga aplikasi yang telah diinstall ke dalam *image*, *Containers* juga dijalankan melalui *images*.

Images merupakan format berlapis menggunakan union file system yang dibangun langkah demi langkah menggunakan beberapa instruksi, seperti :

1. *Add a file*
2. *Run a command*
3. *Open a port*

Images bisa dijadikan “source code” untuk *containers* yang mana sangat portabel dan bisa dilakukan shared, stored serta updated.

2.7.2 Docker Container

Containers dijalankan diatas *images* dan juga dapat mengandung satu atau lebih proses yang berjalan. *Docker containers* berbentuk sebuah *image* yang dapat dikemas, dibaca, dan ditulis. *Layer* baru di atas *base*

image akan terbentuk disaat ada perubahan yang disimpan dalam *containers*.

Dunia pengiriman barang melalui kontainer atau *Shipping container* merupakan konsep yang menginspirasi Docker dalam membangun sistemnya. Bila dianalogikan dalam istilah dunia *shipping container*, setiap *container* mengandung sebuah *software image* hal ini seperti kargo, kemudian *containers* tersebut dapat diberikan instruksi untuk di lakukan *created, started, stopped, restarted dan destroyed*.

2.7.3 Docker Registry

Docker Hub merupakan *registry* publik milik Docker, dimana merupakan repositori distribusi kumpulan docker image yang terpusat.

2.7.4 Versi Docker

Docker tersedia dalam dua edisi yaitu *Community Edition (CE)* dan *Enterprise Edition (EE)*. Docker *Community Edition (CE)* sangat ideal untuk pengembang dengan tim kecil untuk memulai dengan Docker dan melakukan eksperimen dengan aplikasi berbasis container.

Docker *Enterprise Edition (EE)* didisain untuk pengembang *enterprise* dan tim IT yang melakukan *build, ship*, serta menjalankan aplikasi bisnis kritikal sebagai produknya.

Beberapa perbedaan antara Docker *CE* dan Docker *EE* dirangkum dalam tabel berikut :

Tabel 2.7.1 Perbedaan Docker *CE* dan Docker *EE*

Kapabilitas	<i>Community Edition</i>	<i>Enterprise Edition Basic</i>	<i>Enterprise Edition Standard</i>	<i>Enterprise Edition Advanced</i>
Mesin <i>container</i> dan <i>built in orchestration, networking, security</i>	✓	✓	✓	✓
Infrastruktur tersertifikasi, <i>plugins</i> dan <i>ISV containers</i>		✓	✓	✓
<i>Image management</i>			✓	✓
<i>Container app management</i>			✓	✓
<i>Image security scanning</i>				✓