BAB II

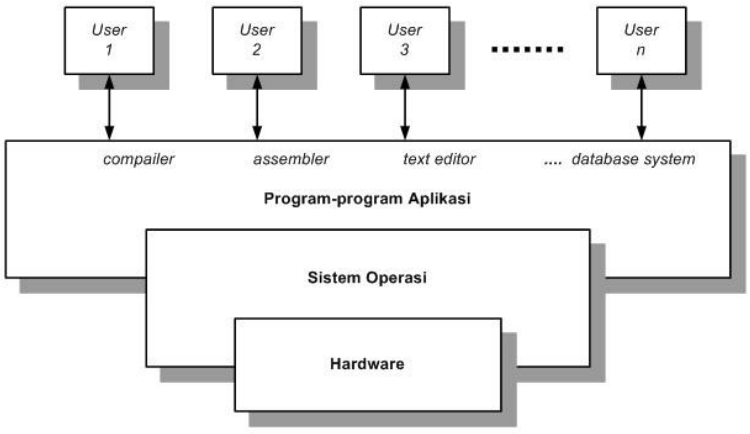
LANDASAN TEORI

Pada bab ini akan dibahas mengenai studi literatur dan penunjang yang akan digunakan dalam pengembangan dan optimalisasi kernel linux Ubuntu 9.10 LTSP.

2.1 Konsep Sistem Operasi

Sistem operasi merupakan program yang ditulis untuk mengendalikan dan mengkoordinasikan kegiatan dari sistem komputer serta menyediakan program ke pengguna dengan antarmuka yang lebih sederhana dari pada pengguna langsung mengakses hardware atau sumber daya sistem secara langsung.

Sistem operasi adalah bagian yang sangat penting bagi semua sistem komputer, secara umum sistem komputer terbagi atas hardware, sistem operasi, program aplikasi, dan *user.* [3]

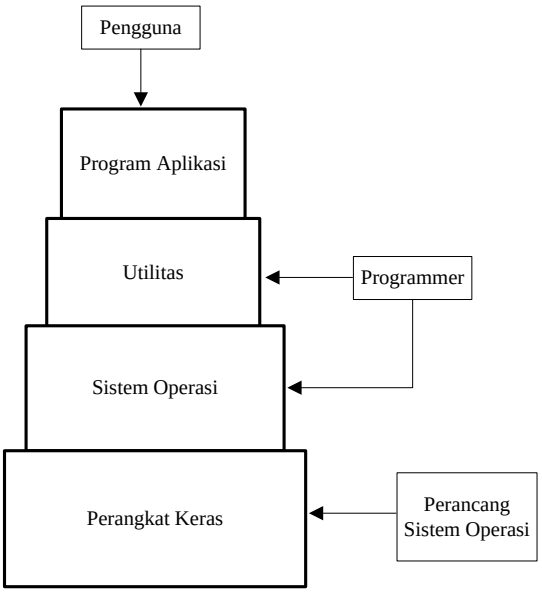


Gambar 2.1 Komponen-komponen Sistem Komputer

Sistem operasi mempunyai peranan yang sangat penting dalam sistem berbasis komputer. *Modern Operating system*, ada dua tugas utama sistem operasi. Kedua tugas tersebut yaitu sistem operasi sebagai *extended machine* dan

sistem operasi sebagai *resource manager*. Sistem operasi sebagai *extended machine* artinya sistem operasi bertugas untuk menyembunyikan kerumitan hardware dan menyediakan antarmuka yang lebih mudah dipakai 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 lainya.

Sistem operasi dilihat dari metode pengembangannya sendiri bisa dibagi menjadi dua. Pertama adalah sistem operasi dengan metode pengembangan tertutup seperti Windows dimana tidak bisa melihat dan mengubah *source code* dari sistem operasi tersebut dan yang kedua adalah sistem operasi *open source* dimana pengguna bisa memakai dan melihat kode penyusun dari sistem operasi tersebut. Contoh sistem operasi *open source* yaitu Linux, Minix dan FreeBSD. Namun, sistem operasi *free* yang paling populer untuk saat ini adalah Linux. Linux bisa berjalan diatas arsitektur prosesor yang berbeda beda, dari super komputer, *server*, komputer pribadi, *handled device* sampai *embedded* sistem. Untuk memperjelas tentang konsep sistem operasi, berikut adalah diagram yang menunjukan posisi sistem operasi pada sistem berbasis komputer. [2][3][5]

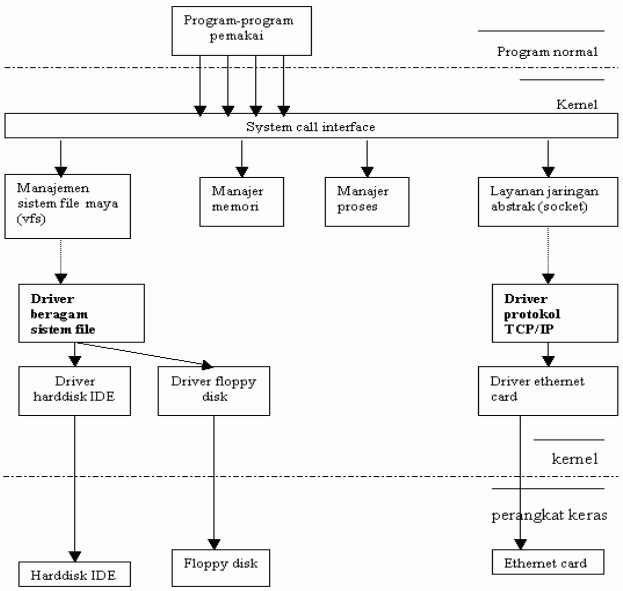


Gambar 2.2 Posisi Sistem Operasi dalam Sistem Komputer

2.2 Linux

GNU/Linux adalah sistem operasi yang dibuat oleh Linus Benedict Torvalds dan disebarkan secara bebas di internet dimana orang lain bisa mengembangkan dan menggunakan untuk keperluannya sendiri. Namun, perlu dijelaskan bahwa GNU/Linux disini bisa bermakna ganda. Pertama, GNU/Linux berarti kernel linux. Pengertian kedua berarti sebuah sistem yang didalamnya sudah terdapat kernel, *shell* dan program pendukung lain yang siap di distribusikan dan dipakai. GNU/Linux dalam penelitian ini mengacu pada pengertian yang kedua. GNU/Linux adalah sistem operasi yang bebas dipakai, didistribusikan dan dikembangkan kembali. Oleh karena itu, GNU/Linux mempunyai banyak varian yang lebih dikenal dengan istilah distro.

Sistem operasi GNU/Linux terdiri atas kernel linux (inti), program sistem, dan beberapa program aplikasi. Kernel linux merupakan inti dari sistem operasi. Program sistem dan semua program-program lainnya yang berjalan di atas kernel disebut *user mode*. Perbedaan antara program sistem dan program aplikasi adalah program sistem dibutuhkan agar suatu sistem operasi dapat berjalan sedangkan program aplikasi adalah program yang dibutuhkan untuk menjalankan suatu aplikasi tertentu.



Gambar 2.3 Arsitektur Sistem Operasi Linux

Sistem Linux terdiri atas tiga badan kode utama, dengan isi pada barisnya merupakan implementasi UNIX paling tradisional :

1. Kernel

   Kernel Linux adalah potongan orisinil dari perangkat lunak yang dibuat dari serpihan oleh komunitas Linux. Sedangkan sistem Linux merupakan gabungan dari komponen-komponen. Sistem Linux basic adalah lingkungan standar untuk aplikasi dan program *user*.

2. System libraries

   System libraries mendefinisikan *set* standar dari fungsi untuk melewatkan aplikasi agar dapat berinteraksi dengan kernel. Implementasi dari fungsi-fungsi ini sedikit banyak ada pada fungsionalitas sistem operasi yang tidak membutuhkan hak keseluruhan atas kode kernel. *System libraries* menyediakan banyak tipe dari fungsionalitas. Pada level paling sederhana, *system libraries* mengijinkan aplikasi untuk membuat permintaan *kernel-system-service*. *System libraries* juga menjaga dan mengoleksi argument *system call* dan jika diperlukan mengatur argumen-argumen tersebut ke dalam suatu bentuk khusus untuk melakukan *system call.*

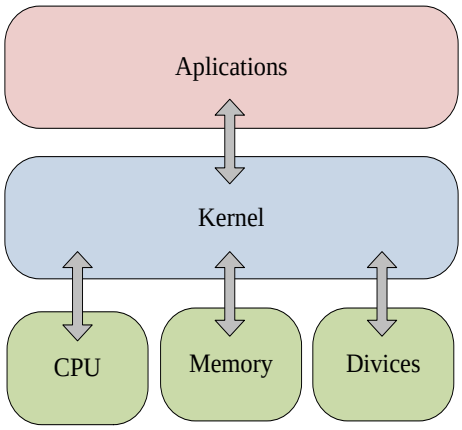
3. System utilities

   System utilities adalah program yang menunjukkan tugas manajemen yang individual dan terspesialisasi. Beberapa *system utilities* dapat dilibatkan hanya sekali saja untuk menginisialisasi dan mengatur beberapa aspek dari sistem secara permanen, memegang tugas seperti merespon pada koneksi jaringan yang masuk.

   Sistem Linux termasuk di dalamnya bermacam-macam ***user-mode*** program, baik *system utilities* maupun *user utilities*. Pada *system utilities* terdapat seluruh program yang dibutuhkan untuk menginisialisasi sistem.

2.2.1 Kernel

Kernel adalah bagian dari sistem operasi yang menjembatani antara *hardware* dan *software*. Kernel adalah jantung dari sistem operasi. Tugas utama dari kernel adalah menyediakan antar muka kepada *user* dan membuat lapisan pemisah antara *hardware, software* aplikasi dan *user*. Kernel juga menyediakan berbagai dukungan hardware, filesystem dan jaringan serta beberapa fasilitas dasar lainnya. Kernel linux adalah kernel yang dipakai oleh distribusi GNU/Linux. Sampai saat ini kernel linux sudah mencapai versi 2.6.xx.



Gambar 2.4 Struktur Kernel Sistem Operasi

Karena akses terhadap perangkat keras terbatas, sedangkan ada lebih dari satu program yang harus dilayani dalam waktu bersamaan, maka kernel juga bertugas untuk mengatur kapan dan berapa lama suatu program dapat menggunakan satu bagian perangkat keras tersebut. Hal tersebut dinamakan *multiplexing*. Akses kepada perangkat keras secara langsung merupakan masalah yang kompleks, oleh karena itu kernel biasanya mengimplementasikan sekumpulan abstraksi *hardware*. Abstraksi-abstraksi tersebut merupakan sebuah cara untuk menyembunyikan kompleksitas, dan memungkinkan akses kepada perangkat keras menjadi lebih mudah dan seragam. Sehingga abstraksi pada akhirnya memudahkan pekerjaan *programmer*.

Untuk menjalankan sebuah komputer kita tidak harus menggunakan kernel sistem operasi. Sebuah program dapat saja langsung di *load* dan dijalankan diatas mesin komputer, yaitu bilamana pembuat program ingin melakukan pekerjaannya tanpa bantuan abstraksi perangkat keras atau bantuan sistem operasi.

Sebuah kernel sistem operasi tidak harus ada dan dibutuhkan untuk menjalankan sebuah komputer. Program dapat langsung dijalankan secara langsung di dalam sebuah mesin (contohnya CMOS *setup*) sehingga para pembuat program tersebut membuat program tanpa adanya dukungan dari sistem operasi atau *hardware abstraction*.

Pengembang sistem operasi mengembangkan *kernel* sistem operasi yang pada akhirnya terbagi menjadi 4 bagian yang secara desain berbeda yaitu Monolithic kernel, Microkernel, Hybrid (modifikasi dari microkernel), dan Exokernel.

2.2.1.1 Monolithic Kernel

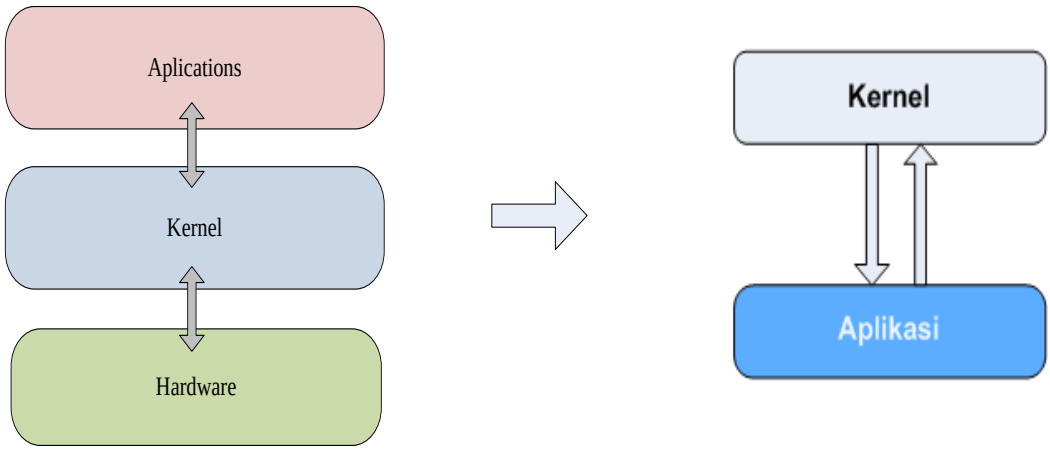
Kernel yang menyediakan abstraksi perangkat keras yang kaya dan tangguh. Kernel monolitik mengintegrasikan banyak fungsi di dalam kernel dan menyediakan lapisan abstraksi perangkat keras secara penuh terhadap perangkat keras yang berada dibawah sistem operasi.

Pendekatan kernel monolitik didefinisikan sebagai sebuah antar muka virtual yang berada pada tingkat tinggi perangkat keras, dengan sekumpulan primitif atau *system call* untuk mengimplementasikan layanan-layanan sistem operasi, seperti halnya manajemen proses, konkurensi, dan manajemen memori pada modul-modul kernel yang berjalan di dalam *mode supervisor*.

Meskipun jika setiap modul memiliki layanan operasi-operasi tersebut terpisah dari modul utama, integrasi kode yang terjadi di dalam monolitik kernel sangatlah kuat, dan karena semua modul berjalam dalam *address space* yang sama, sebuah *bug* dalam salah satu modul dapat merusak keseluruhan sistem. Akan tetapi, ketika implementasi dilakukan secara benar, integrasi komponen

internal yang sangat kuat tersebut justru mengijinkan fitur-fitur yang dimiliki oleh sistem yang berada dibawahnya dieksploitasi secara efektif, sehingga membuat sistem operasi dengan kernel monolitik sangatlah efisien, meskipun sangat sulit membuatnya.

Pada sistem operasi modern yang menggunakan kernel monolitik, seperti halnya GNU/Linux, FreeBSD, Open Solaris, dan Microsft Windows, dapat memuat modul-modul yang dapat dieksekusi pada saat kernel tersebut dijalankan sehingga mengijinkan ekstensi terhadap kemampuan kernel sesuai kebutuhan, dan tentu saja dapat membantu menjaga agar kode yang berjalan pada ruangan kernel seminim mungkin.



Gambar 2.5 Diagram Monolilhitic Kernel

Monolitik kernel linux dapat dibuat sangat kecil tidak hanya karena kemampuannya untuk memuat modul secara dinamis tetapi juga karena kemudahan kustomisasi. Bahkan, ada beberapa versi yang cukup kecil yang cocok dengan jumlah besar utilitas dan program lain pada satu floppy *disk* dan masih menyediakan sistem operasi yang berfungsi penuh. Kemampuan untuk miniaturize dengan kernel juga mengakibatkan pertumbuhan yang cepat dalam penggunaan GNU/Linux pada *embedded system* (yaitu, sirkuit komputer dibangun menjadi produk lain). Selain itu, ukuran kernel yang dikompilasi hanya sebagian kecil dari *source code* aslinya. Sehingga kernel linux yang terkompilasi

mempunyai kemampuan untuk memuat modul-modul secara dinamis pada saat *runtime.*

2.2.1.2 Modul Kernel Linux

Modul kernel linux adalah bagian dari kernel linux yang dapat dikompilasi, dipanggil dan dihapus secara terpisah dari bagian kernel lainnya saat dibutuhkan. Modul kernel dapat menambah fungsionalitas kernel tanpa perlu me-*reboot* sistem. Secara teori tidak ada yang dapat membatasi apa yang dapat dilakukan oleh modul kernel. Kernel modul dapat mengimplementasikan antara lain *device driver*, sistem berkas, dan protokol jaringan.

Kernel linux modul juga memberikan keuntungan lain yaitu membuat sistem Linux dapat dinyalakan dengan kernel standar yang minimal, tanpa tambahan *device driver* yang ikut dipanggil. *Device driver* yang dibutuhkan dapat dipanggil kemudian secara eksplisit maupun secara otomatis saat dibutuhkan. Terdapat tiga komponen untuk menunjang modul *kernel* Linux. Ketiga komponen tersebut adalah manajemen modul kernel linux, *registrasi driver*, dan mekanisme resolusi konflik. Berikut akan dibahas ketiga komponen pendukung tersebut.

2.2.1.3 Manajemen Modul Kernel Linux

Manajemen modul akan mengatur pemanggilan modul ke dalam memori dan berkomunikasi dengan bagian lainnya dari kernel. Memanggil sebuah modul tidak hanya memasukkan isi binarinya ke dalam memori kernel, namun juga harus dipastikan bahwa setiap rujukan yang dibuat oleh modul ke simbol kernel atau pun titik masukan diperbaharui untuk menunjuk ke lokasi yang benar di alamat kernel. Linux membuat tabel simbol internal di kernel. Tabel ini tidak memuat semua simbol yang didefinisikan di kernel saat kompilasi, namun simbol-simbol tersebut harus diekspor secara eksplisit oleh kernel. Semua hal ini diperlukan untuk penanganan rujukan yang dilakukan oleh modul terhadap simbol-simbol.

2.2.1.4 Registrasi Driver

Untuk membuat modul kernel yang baru dipanggil berfungsi, bagian dari kernel yang lain harus mengetahui keberadaan dan fungsi baru tersebut. Kernel membuat table dinamis yang berisi semua *driver* yang telah diketahuinya dan menyediakan serangkaian routines untuk menambah dan menghapus *driver* dari table tersebut. Routines ini yang bertanggung jawab untuk mendaftarkan fungsi modul baru tersebut.

Hal-hal yang masuk ke dalam tabel registrasi adalah:

1. *Device driver*
2. Sistem berkas
3. Protokol jaringan
4. Format binary

2.2.1.5 Resolusi Konflik

Keanekaragaman konfigurasi perangkat keras komputer serta *driver* yang mungkin terdapat pada sebuah computer pribadi telah menjadi suatu masalah tersendiri. Masalah pengaturan konfigurasi perangkat keras tersebut menjadi semakin kompleks akibat dukungan terhadap *device driver* yang modular, karena *device* yang aktif pada suatu saat bervariasi.

GNU/Linux menyediakan sebuah mekanisme penyelesaian masalah untuk membantu *arbitrasi* akses terhadap perangkat keras tertentu. Tujuan mekanisme tersebut adalah untuk mencegah modul konflik akses terhadap suatu perangkat keras, mencegah *autoprobes* menganggu keberadaan *driver* yang telah ada, menyelesaikan konflik diantara sejumlah *driver* yang berusaha mengakses perangkat keras yang sama.

Kernel membuat daftar alokasi sumber daya perangkat keras. Ketika suatu *driver* hendak mengakses sumber daya melalui *I/O port*, jalur *interrupt*, atau kanal

DMA maka *driver* tersebut diharapkan mereservasi sumber daya tersebut pada basis data kernel linux terlebih dahulu. Jika reservasinya ditolak akibat ketidaktersediaan sumber daya yang diminta, maka modul harus memutuskan apa yang hendak dilakukan selanjutnya. Jika tidak dapat melanjutkan, maka modul tersebut dapat dihapus.

2.2.2 Shell

Shell dalam komputer adalah salah satu jenis program bawaan sistem operasi yang menyediakan komunikasi langsung antara pengguna dan sistem operasi. *Shell* adalah lapisan diatas kernel dimana tugasnya menyediakan antarmuka kepada *user* untuk menajalankan perintah-perintah yang dikenali oleh sistem operasi. *Shell* yang paling banyak dipakai saat ini adalah Bash (*Bourne again shell*). Bash adalah *shell* yang mempunyai banyak kelebihan. Diantara kelebihan yang paling Nampak adalah *autocomplete* dan *command history*. Disisi lain, bash menyediakan bahasa pemrograman *shell* yang sering disebut dengan *shell script*.

2.3 Filesystem

Filesystem adalah metode dan struktur data pada sebuah sistem operasi yang mengatur berkas pada sebuah disk atau partisi. GNU/Linux merupakan sistem operasi yang masih satu keluarga dengan UNIX dimana keduanya mempunyai struktur direktori yang sudah baku.

2.3.1 Filesystem Hirarchy Standard (FHS)

Filesystem Hierarchy Standard adalah suatu peta tentang penempatan file dan direktori pada sistem operasi yang mirip UNIX, merupakan pedoman direktori standard dalam membuat sebuah distribusi Linux yang oprasional. Dalam FHS semua *file* dan direktori di GNU/Linux dan UNIX mirip sebuah pohon terbaik dimana direktori paling atas disebut dengan direktori *root* (/).

Struktur paling dasar yang harus bisa melakukan *boot*, perbaikan, atau mengembalikan sistem dalam keadaan semula. Berikut adalah daftar direktori dalam sistem *root*:

1. `/bin`

   Berisi program perintah esensisal yang dibutuhkan *user*.

2. `/boot`

   Berisi program biner yang dibutuhkan untuk *booting*.

3. `/dev`

   Berisi semua *file device* baik blok maupun karakter.

4. `/etc`

   Berisi *file* konfigurasi.

5. `/home`

   Direktori untuk setiap *user* yang terdaftar dalm sistem.

6. `/lib`

   Berisi pustaka bersama.

7. `/media`

   Point pengaitan pada media yang dapat dibongkar pasang.

8. `/mnt`

   Tempat pengaitan Sistem *file* sementara/*temporary*.

9. `/opt`

   Berisis paket aplikasi tambahan (*add-on*), biasanya berupa aplikasi *propietary*.

10. `/root`

    Direktori untuk *user root*.

11. `/sbin`

    Berisi program biner esensial yang dibutuhkan untuk menjalankan dan memperbaiki sistem.

12. `/srv`

    Direktori yang berisi data untuk semua layanan sistem yang bersangkutan.

13. `/tmp`

    Tempat menyimpan *file* sistem sementara.

14. `/usr`

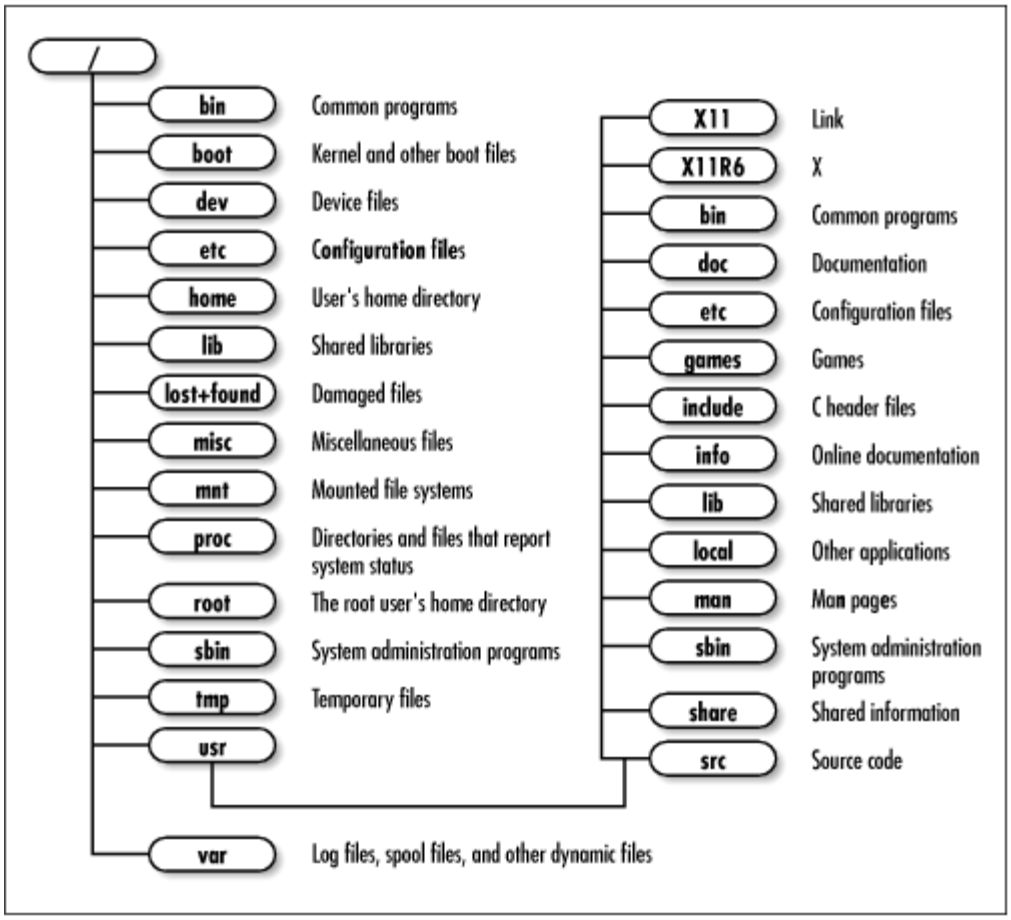
    Direktori terbesar kedua setelah *root* (/), berisi *file* sistem lengkap sebagai *file* sistem dasar.

15. `/var`

    Berisi *file-file* data yang sering berubah, misalnya *file* log, file cache dan beberapa file data sementara lainnya.

16. `/proc`

    Direktori `/proc` berisi *file-file* proses pada saat Linux berjalan, dalam direktori ini juga terdapat *file-file* yang memperlihatkan konfigurasi komputer, informasi BIOS, partisi hardisk , CPU dan juga memori.



Gambar 2.6 Struktur *Filesystem Hierarchy Standard*

2.4 Distribusi

Distribusi atau lebih sering disingkat dengan distro adalah istilah yang dipakai untuk orang atau perusahaan yang mengemas kernel Linux dan paket-paket *open source* lainnya menjadi satu yang kemudian didistribusikan kembali dengan tujuan agar orang lain bisa dengan mudah memakai Linux. Distro biasanya menyediakan kemudahan dalam instalasi, konfigurasi dan penambahan *software* atau *hardware* kedalam sistem Linux. Masing-masing distro menggunakan cara yang berbeda dalam metode instalasi dan administrasi sistemnya. Saat ini distribusi GNU/Linux sudah sangat banyak. Distro yang cukup

terkenal adalah Redhat, Slackware, Debian, dan Ubuntu. Umumnya, distro lain diturunkan dari empat distro besar ini.

2.4.1 Distro Linux

Distro adalah bundel dari kernel linux, beserta sistem dasar linux, program instalasi, *tools basic*, dan program-program lain yang bermanfaat sesuai dengan tujuan pembuatan distro. Menurut fungsi dan target penggunanya, distro dibagi menjadi dua, yaitu:

1. Distro Umum

Distro umum adalah distro yang dipakai untuk keperluan umum. Distro jenis ini umumnya disebarkan dalam beberapa keping CD/DVD dengan paket software yang sangat lengkap. Distro ini bias dipakai untuk keperluan desktop, server ataupun untuk keperluan pengembangan software.

2. Distro Khusus

Distro khusus adalah distro yang ditujukan untuk keperluan khusus seperti untuk keamanan jaringan, pemrograman atau untuk keperluan spesifik lainnya. Umumnya distro-distro ini berukuran kecil dan bias diinstall dimedia floppy *disk*, USB *flashdrive* atau didistribusikan dalam *mode Live* CD/DVD. Contoh distro jenis ini adalah damnsmall Linux, puppy Linux dan openSuSE.

2.4.2 Distro Ubuntu Linux

Ubuntu adalah salah satu distribusi Linux yang berbasiskan pada Debian GNU/Linux. Proyek Ubuntu disponsori oleh Canonical Ltd (Perusahaan milik Mark Shuttleworth). Nama Ubuntu diambil dari nama sebuah konsep ideologi di Afrika Selatan. ”Ubuntu” berasal dari bahasa kuno Afrika, yang berarti “Rasa perikemanusian terhadap sesame manusia”. Ubuntu juga bisa berarti “Aku adalah aku karena keberadaan kita semua”. Tujuan dari distribusi Linux Ubuntu adalah membawa semangat yang terkandung didalam Ubuntu ke dalam dunia perangkat lunak.

Ubuntu adalah sistem operasi lengkap berbasis GNU/Linux, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional. Perihal kebebasan inilah yang membuat Ubuntu berbeda dari perangkat lunak berpemilik (*proprietary*).

2.4.3 Live CD/DVD Linux

Live CD/DVD Linux adalah istilah yang dipakai untuk sistem yang bisa dijalankan lewat media CD/DVD ROM. *Live* CD/DVD umumnya dipakai untuk menjalankan Linux tanpa perlu melakukan instalasi kedalam harddisk. Tujuan dibuatnya *Live* CD/DVD bermacam-macam. Umumnya *Live* CD/DVD dibuat untuk mencoba Linux tanpa harus menginstalnya. Dengan begitu, para pengguna bisa mencoba dulu tanpa harus takut data atau sistem operasi yang sudah terinstal di komputernya rusak.

2.4.4 Cara Kerja Live CD/DVD Linux

Karena sistem operasi GNU/Linux tersimpan di CD/DVD, semua proses mulai dari *boot* sampai *shutdown* dilakukan dari CD/DVD tersebut. Ketika komputer *boot* dari *Live* CD/DVD, skrip yang terdapat di *Live* CD/DVD akan memuat *image* kernel (*vmlinuz*). Selanjutnya, sejumlah *ramdisk* dibuat dalam RAM (*Random Acces Memory*) pada komputer. Tujuanya sebagai tempat penyimpanan data sementara yang dibutuhkan oleh *Live* CD/DVD untuk *boot*. *Image rootdisk* (*initrd.gz*) kemudian dibongkar kedalamnya dan di *mount* sebagai sistem *file root*. Sementara direktori-direktori besar (seperti /bin, /etc, /lib, /opt, dan lainya) akan di *mount* langsung dari CD/DVD *drive.* Sehingga beban kerja komputer akan berkurang karena tidak semua data disimpan di memori.

2.5 Tool Yang Digunakan dalam Konfigurasi dan Kompilasi Kernel

Berikut ini adalah *tool-tool* yang digunakan untuk membangun sistem.

1. Kernel-package

Paket ini pada konfigurasi kernel digunakan sebagai paket kernel yang akan menjadi pedoman saat akan melakukan pengkonfigurasian kernel *source*, sebelum kernel *source* akan di kompilasi sesuai dengan kebutuhan.

2. `libncurses5-dev` dan `libcurses5`

   Paket `libncurses5-dev` ini digunakan agar saat melakukan kompilasi pada kernel *source*, dapat menampilkan modus grafik (GUI) dari kernel *source* yang akan dikompilasi sehingga memudahkan kita dalam melakukan kompilasi kernel.

3. `fakeroot`

   Paket `fakeroot` ini digunakan sebagai paket yang akan berfungsi untuk menyamarkan *user* yang kita gunakan menjadi seolah-olah menjadi *user root.*

4. `wget`

   `wget` adalah aplikasi download manager yang ada pada sistem operasi GNU/Linux, dan biasanya aplikasi ini telah menjadi *default* dari linux, sehingga kita tidak perlu menginstall aplikasi ini setelah melakukan instalasi sistem operasi linux.

5. `bzip2`

   Paket aplikasi ini digunakan untuk mengekstrak atau mengurai *file* terkompresi berbentuk *file tar.bz*.

6. `build-essential, crash kesec-tools, makedunfile, kernel-wedge, git-core, libelf-dev, asciidoc, binutils-dev` dan `crash kesec-tools.`

   Paket ini adalah paket pendukung untuk membuat paket debian.

2.6 Teknologi Terminal Server

Terminal server dalam bahasa sederhana sering disebut dengan teknologi kloning, sebuah PC *Server* yang besar diakses oleh banyak PC *workstation* melalui jaringan LAN agar merasakan kecepatan yang hampir sama dengan PC *Server* yang besar tersebut. Pada sisi ekstrem kita dapat menggunakan PC tanpa harddisk (*diskless*) sebagai *workstation*. Kita dapat menggunakan mesin 486 dengan RAM 16 Mbyte, PC jenis ini sangat cocok untuk digunakan sebagai *client workstation,* tanpa perlu dipasangi harddisk. [4]

2.7 Linux Terminal Server Project

Linux Terminal Server Project (LTSP), adalah suatu program aplikasi *thin-client* (*diskless*) yang mensupport *server* berbasis GNU/Linux, LTSP ini sangat mudah diakses dan merupakan solusi efektif hemat biaya, yang memberdayakan instansi pendidikan, bisnis, dan organisasi di seluruh dunia untuk memudahkan penginstalan dan mengembangkan *desktop workstation*, dengan semakin pesatnya pertumbuhan distribusi linux, banyak distribusi yang menambahkan LTSP dalam paket instalasinya. Konsep komputer yang akan dijalankan dalam LTSP adalah secara *thin-client*. *Thin-client* adalah sebuah sistem komputer yang dapat dijalankan tanpa menggunakan *disk storage*, dimana komputer dalam proses *booting* akan mengambil sistem operasi dari *server* dengan terlebih dahulu mengubah *setting boot priority BIOS* komputer ke setting *network* ataupun dapat menggunakan *floppy disk*, *compact disk* (CD), dan *flashdisk* yang telah diisi dengan file *image boot*ROM. [4]

2.8 Konsep Dasar Jaringan Komputer

Jaringan komputer adalah sekelompok komputer otonom (*stand alone*) yang saling berhubungan antara satu dengan yang lainnya menggunakan protokol

komunikasi melalu media komunikasi sehingga dapat saling berbagi informasi, data program aplikasi dan perangkat keras seperti *printer*. Selain itu jaringan komputer bisa diartikan sebagai kumpulan sejumlah *terminal* komunikasi yang berada diberbagai lokasi yang terdiri dari lebih satu komputer yang saling berhubungan.

Tiap komputer, printer atau perangkat yang terhubung dengan jaringan disebut *node*. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan *node*. Sebuah jaringan biasanya terdiri dari 2 atau lebih komputer yang saling berhubungan diantara satu dengan yang lainnya, yang saling berbagi sumber daya.

2.8.1 Jaringan Komputer Berdasarkan Fungsinya

Berdasarkan fungsinya pada dasarnya setiap jaringan komputer ada yang berfungsi sebagai *client* dan juga *server*. Tetapi ada jaringan yang memiliki komputer yang khusus didedikasikan sebagai *server* sedangkan yang lain sebagai *client*. Ada juga yang tidak memiliki komputer yang khusus berfungsi sebagai *server* saja, serta berdasarkan konsep hubungan terdiri dari jaringan *Client Server* dan jaringan *Peer to Peer*.

2.8.1.1 Client Server

Pada jaringan *client server* terdapat sebuah komputer yang berfungsi sebagai *server* sedangkan komputer-komputer yang lain berfungsi sebagai *client.* Sesuai namanya maka komputer *server* berfungsi dan bertugas untuk melayani seluruh komputer yang terdapat dalam jaringan tersebut.

Pada sebuah komputer dimungkinkan untuk digunakannya lebih dari komputer *server*, bahkan dengan kemampuan dan fasilitas yang berbeda. Sedangkan komputer-komputer *client* sesuai dengan namanya menerima pelayanan dari komputer *server*. Komputer-komputer ini disebut juga dengan *workstation,* yaitu komputer dimana pengguna jaringan dapat mengakses dan memanfaatkan pelayanan yang diberikan oleh komputer *server*.

2.8.1.2 Peer to Peer

Pada jaringan *peer to peer* setiap komputer yang terhubung dalam jaringan dapat berkomunikasi dengan komputer-komputer lain secara langsung tanpa melalui komputer perantara. Pada jaringan tipe ini sumber daya komputer terbagi pada seluruh komputer yang terhubung dalam jaringan tersebut, baik sumber daya yang berupa perangkat keras maupun perangkat lunak dan datanya.

Sebuah komputer yang terhubung dalam sebuah jaringan *peer to peer* pada prinsipnya mampu untuk bekerja sendiri sebagai sebuah komputer *standalone* (berdiri sendiri). Membangun sebuah jaringan seperti ini pengguna bisa menggunakan komputer-komputer yang memiliki kemampuan yang setara karena keamanan dalam jaringan tersebut diatur dan di kontrol oleh masing-masing komputer dalam jaringan tersebut.

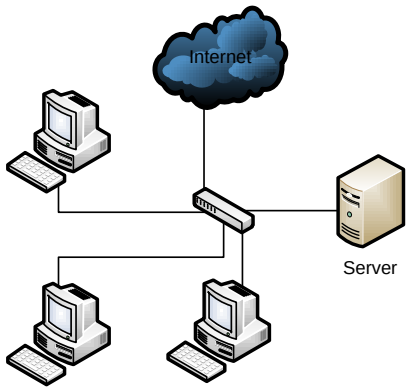
2.8.2 Berdasarkan Cakupan Geografis

Jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya, ada 3 kategori utama jaringan komputer, diantaranya :

1. LAN (*Local Area Network*)
2. MAN (*Metropolitan Area Network*)
3. WAN (*Wide Area Network*)

2.8.2.1 LAN (*Local Area Network*)

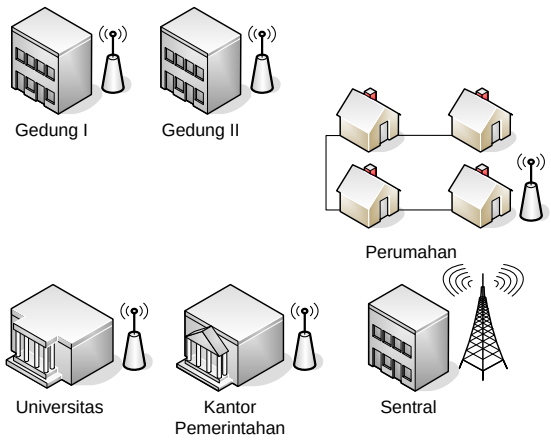
Local Area Network adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan seperti sebuah kantor pada sebuah gedung. LAN seringkali menggunakan teknologi transmisi kabel tunggal. LAN tradisional beroperasi pada kecepatan mulai 10 sampai 100 Mbps (Mega bit/detik) dengan delay rendah (puluhan mikro *second*) dan mempunyai faktor kesalahan yang kecil. LAN-LAN modern dapat beroperasi pada kecepatan yang lebih tinggi, sampai ratusan megabit/detik.



Gambar 2.7 *Local Area Network*

2.8.2.2 MAN (*Metropolitan Area Network*)

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km sampai beberapa ratusan km, suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.

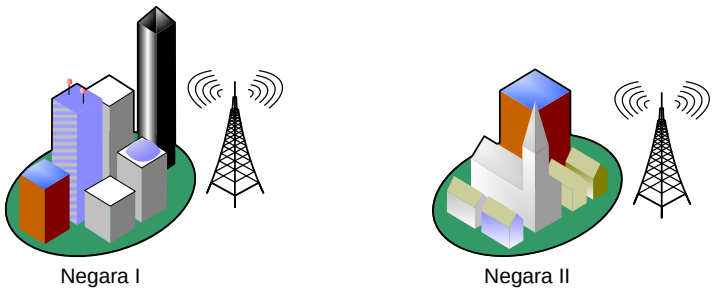


Gambar 2.8 MAN (*Metropolitan Area Network*) pada Sebuah Kota

2.8.2.3 WAN (*Wide Area Network*)

Wide Area Network adalah jaringan yang ruang lingkupnya sudah menggunakan sarana satelit, wireless, ataupun kabel *fiber optic*. WAN memiliki jangkuan yang lebih luas hingga wilayah otoritas negara lain.

WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota lain dalam satu negara. Cakupan WAN bias meliputi 100 km sampai 1.000 km, dan kecepatan antar kota bias bervariasi antara 1,5 Mbps sampai 2,4 Gbps. Dalam WAN, biaya untuk peralatan transmisi sangat tinggi, dan biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan publik. Para pelaku bisnis dapat menyewa sistem transmisi tersebut untuk menghubungkan kantor-kantor cabang yang dimilikinya.



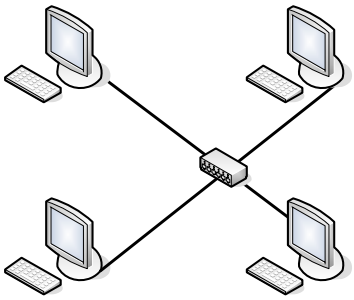
Gambar 2.9 WAN (*Wide Area Network*) antar Negara

2.8.3 Berdasarkan Topologi Jaringan Komputer

Topologi atau arsitektur jaringan merupakan pola hubungan antar terminal dalam suatu sistem jaringan komputer. Topologi ini akan mempengaruhi tingkat efektifitas kinerja jaringan. Ada beberapa jenis topologi yang dapat diimplementasikan dalam jaringan. Namun, bentuk topologi yang utama adalah Topologi Bus, Topologi Ring dan Topologi Star.

2.8.3.1 Topologi Star

Seperti namanya topologi ini berbentuk seperti bintang, masing-masing komputer dalam jaringan terhubung dengan pusat (*central*). Terminal pusat tersebut bertindak sebagai pengatur dan pengendali semua komunikasi data. Terminal inilah yang menyediakan jalur komunikasi khusus pada komputer yang akan berkomunikasi, yang berupa hub. Hub merupakan alat yang menyediakan lokasi terpusat, di mana semua kabel UTP terpasang. Berikut adalah skema dari topologi star.



Gambar 2.10 Topologi *Star*

Kelebihan dari Topologi Star adalah :

1. Kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut dan *station* yang terpaut.
2. Tingkat keamanan termasuk tinggi.
3. Tahan terhadap lalu lintas jaringan yang sibuk.
4. Penambahan dan pengurangan station dapat dilakukan dengan mudah.

Sedangkan kelemahan dari Topologi Star adalah :

1. Memiliki satu titik kesalahan, terletak pada hub. Jika hub pusat mengalami kegagalan, maka seluruh jaringan akan gagal untuk beroperasi.
2. Membutuhkan lebih banyak kabel karena semua kabel jaringan harus ditarik ke satu *central point*, jadi lebih banyak membutuhkan lebih banyak kabel daripada topologi jaringan yang lain.

3. Jumlah terminal terbatas, tergantung dari port yang ada pada hub.
4. Lalulintas data yang padat dapat menyebabkan jaringan bekerja lebih lambat.

2.9 Protokol Jaringan Komputer

Protokol adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras.

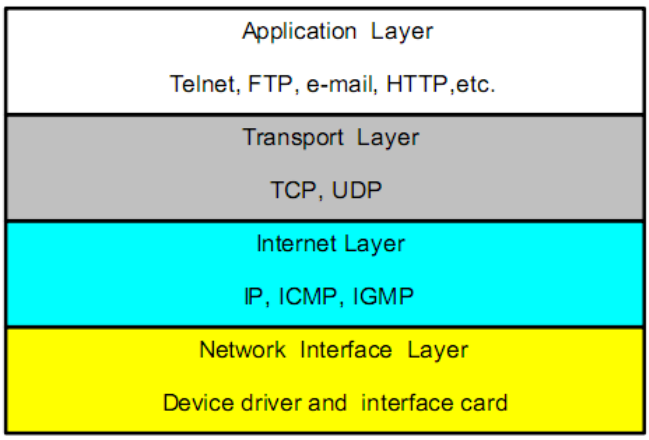
Komunikasi data merupakan proses mengirimkan data dari satu komputer ke komputer yang lain. Untuk dapat mengirim data, pada komputer harus ditambahkan alat khusus yang di kenal dengan *network interface*. Dalam proses pengiriman data ini terdapat beberapa masalah yang harus dipecahkan. Pertama, data harus dapat dikirim ke komputer yang tepat, sesuai dengan tujuannya. Namun hal ini akan menjadi rumit jika komputer tujuan tidak berada dalam jaringan lokal. Untuk memecahkan masalah *transfer* di atas maka diciptakan sebuah solusi khusus berupa aturan-aturan yang dapat menangani semua masalah komunikasi data, keseluruhan aturan ini harus bekerja sama satu dengan yang lainnya. Sekumpulan aturan untuk mengatur proses pengiriman data ini disebut sebagai protokol komunikasi data.

2.9.1 Arsitektur Protokol TCP/IP

TCP/IP adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada *wide area network* (WAN). TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Berkat protokol ini, tugas masiing-masing protokol menjadi jelas dan sederhana dan menjadi protokol komunikasi yang

fleksibel. TCP/IP dibangun sebelum OSI. TCP/IP terdiri dari 5 layer (Physical, Data Link, Network, Transport, Application).

1. Layer Application : Layer ini melayani remote terminal dan melakukan pemindahan file ke sistem lain sehingga mendukung aplikasi jaringan. Contoh : ftp, smtp, http
2. Layer Transport: Layer ini bertanggung jawab untuk mengadakan komunikasi antar dua host/komputer. Contoh : tcp, udp
3. Layer Network: Layer ini merutekan datagram dari sumber ke tujuan. Contoh : ip, routing protocols
4. Layer Datalink: Layer ini mentransfer data antar elemen dalam jaringan. Contoh : ppp
5. Layer Physical : Layer ini pengiriman bit-bit pada medium transmisi. Contohnya : Ethernet, SLIP dan PPP



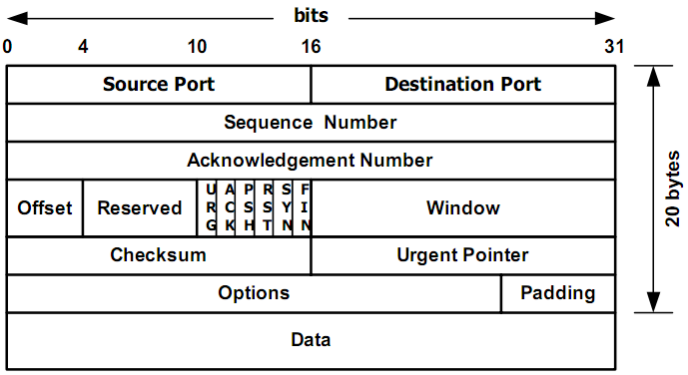
Gambar 2.11 Layer pada TCP/IP

2.9.1.1 TCP (Transmission Control Protocol)

TCP (Transmission Control Protocol), merupakan protokol yang terletak di layer transport. Protokol ini menyediakan *service* yang dikenal sebagai

connection oriented, reliable, byte stream service. *Connection Oriented* berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan hubungan (handshake) terlebih dulu. *Reliable* berarti mendeteksi kesalahan paket dan *retransmisi*. Byte Stream *Service* berarti paket dikirimkan dan sampai ke tujuan secara berurutan. Pembentukan hubungan (handshake) dilakukan dalam pembukaan hubungan TCP antara sebuah *client* dan *server*.

Keandalan TCP dalam mengirimkan data didukung oleh mekanisme yang disebut *Positive Acknowledgement with Re-transmission* (PAR). Data yang dikirim dari layer aplikasi akan dipecah-pecah dalam bagian-bagian yang lebih kecil dan diberi nomor urut (*sequence number*) sebelum dikirimkan ke leyer berikutnya. Unit data yang sudah dipecah-pecah yang disebut dengan segmen (*segment*). TCP selalu meminta konfirmasi setiap kali mengirimkan data, apakah data tersebut sampai pada komputer tujuan dan rusak. Jika data berhasil sampai mencapai tujuan, TCP akan mengirimkan data berikutnya secara berurutan. Jika tidak berhasil, maka TCP akan melakukan pengiriman ulang urutan data yang hilang atau rusak. Dalam kenyataannya TCP mengunakan sebuah *acknowledgement* (ACK) sebagai suatu pemberitahuan antara komputer pengirim dan penerima. Format segmen TCP diperlihatkan pada gambar 2.17.



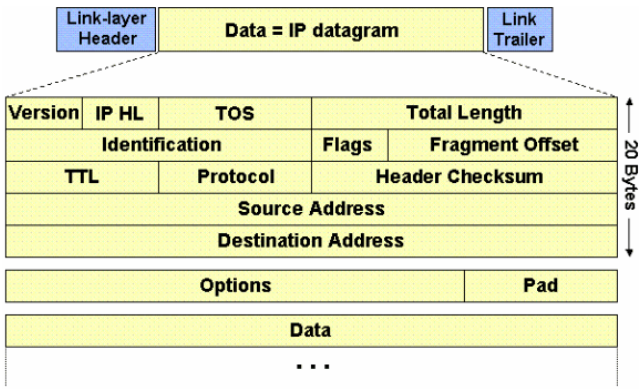
Gambar 2.12 Format *header* TCP

2.9.1.2 IP (Internet Protocol)

IP (*Internet Protocol*) merupakan protokol yang paling penting yang berada pada layer *Internet* TCP/IP. Semua protokol TCP/IP yang berasal dari layer atasnya mengirimkan data melalui protokol IP ini dan IP yang digunakan adalah IPv4. Seluruh data harus dilewatkan, diolah oleh protokol IP dan dikirimkan sebagai datagram IP untuk sampai ke sisi penerima. Dalam pengiriman data, protokol IP ini bersifat *unreliable, connectionless,* dan *datagram delivery service.*

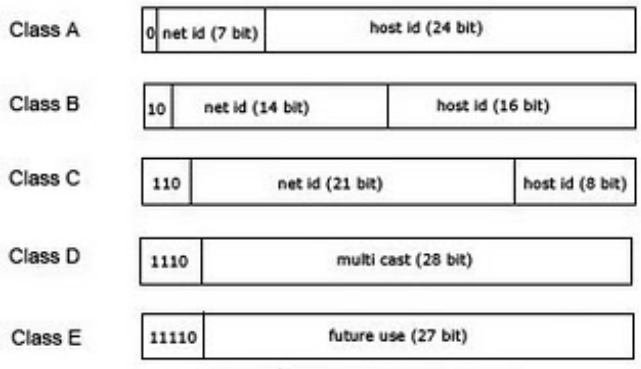
Unreliable berarti protokol IP tidak menjamin datagram yang dikirim pasti sampai ke tujuan. Protokol IP hanya melakukan cara terbaik untuk menyampaikan datagram yang dikirimkan ke tujuan.

Datagram delivery service setiap datagram yang dikirim tidak tergantung pada datagram lainya. Dengan demikian kedatangan datagram pun bisa menjadi tidak berurutan. Metode ini dipakai untuk menjamin sampainya datagram ke tujuannya walaupun salah satu jalur menuju ke tujuan mengalami masalah.



Gambar 2.13 Format datagram IP

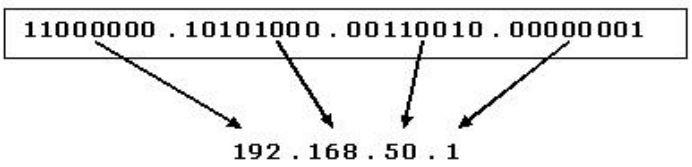
Pada bagian header dari protokol IP terdapat bagian pengalamatan sumber kirim dan tujuan masing-masing sebesar 32-bit. Pengalamatan IP (IP *Addressing*) adalah bagian terpenting dalam jaringan TCP/IP. Format IP *Address* yang dinyatakan dalam bilangan 32-bit dimana setiap 8-bitnya dipisahkan oleh tanda titik. Untuk memudahkan distribusinya, IP *Address* dibagi dalam kelas-kelas .



Gambar 2.14 Format Pembagian Kelas IP *Address*

Tabel 2.1 Pembagian IP *Address* dalam Format Bilangan Desimal

Bit inisial a	Range a	Jumlah Kelas	Kelas	Bagian Network	Bagian Host	Penggunaan
0..	1 – 126	126	A	A	b, c, d	untuk jaringan besar
10..	128 – 191	16,384	B	a, b	c, d	untuk jaringan menegah
110..	192 – 223	2,097,152	C	a, b, c	d	untuk jaringan kecil
1110..	224 – 247	-	D	a, b, c	d	cadangan : IP Multicasting
1111..	248 - 255	-	E	a, b, c	d	Cadangan Eksperimen



Gambar 2.15 Notasi Desimal Bertitik dari IP *Address*

2.10 Server

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan atau *network operating*

system. *Server* juga menjalankan perangkat lunak *administratif* yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (*printer*), dan memberikan akses kepada *workstation* anggota jaringan.

Server biasanya terhubung dengan *client* dengan kabel UTP dan sebuah *Network Card*. Kartu jaringan ini biasanya berupa kartu PCI atau ISA. Fungsi *server* sangat banyak, misalnya untuk situs internet, ilmu pengetahuan, atau sekedar penyimpanan data. Namun yang paling umum adalah untuk mengkoneksikan komputer *client* ke Internet.

2.10.1 DHCP Server

DHCP *(Dynamic Host Configuration Protocol)* adalah protokol yang berbasis arsitektur *client/server* yang dipakai untuk memudahkan pengalokasian alamat IP dalam satu jaringan. Sebuah jaringan lokal yang tidak menggunakan DHCP harus memberikan alamat IP kepada semua komputer secara manual. Jika DHCP dipasang di jaringan lokal, maka semua komputer yang tersambung di jaringan akan mendapatkan alamat IP secara otomatis dari server DHCP. Selain alamat IP, banyak parameter jaringan yang dapat diberikan oleh DHCP, seperti *default gateway* dan DNS *server*. DHCP didefinisikan dalam RFC 2131 dan RFC 2132 yang dipublikasikan oleh *Internet Engineering Task Force*. DHCP merupakan ekstensi dari protocol *Bootstrap Protocol (BOOTP).*

2.10.2 TFTP (*Trivial File Transfer Protocol*)

Trivial File Transfer Protocol adalah sebuah protocol perpindahan berkas yang sangat sederhana. TFTP memiliki fungsionalitas dasar dari protocol *File Transfer Protocol* (FTP). Karena protokol ini sangatlah sederhana, maka implementasi protokol ini dalam komputer yang memiliki memori yang kecil sangatlah mudah. Hal ini memang pertimbangan yang sangat penting pada saat itu. Akhirnya, TFTP pun digunakan untuk melakukan *booting* komputer seperti halnya *router* jaringan komputer yang tidak memiliki perangkat penyimpanan

data. Protokol ini kini masih digunakan untuk mentransfer berkas-berkas kecil antar *host* di dalam sebuah jaringan, seperti halnya ketika terminal jarak jauh *X Window System* atau *thin client* lainnya melakukan proses *booting* dari sebuah *host* jaringan atau *server.*

2.10.3 NFS (*Network File System*)

Network File System adalah sebuah kumpulan protokol yang digunakan untuk mengakses beberapa sistem berkas melalui jaringan. Spesifikasi NFS definisikan dalam RFC 1094, dan saat ini telah mencapai versi 3 yang didefinisikan dalam RFC 1813. NFS didesain sedemikian rupa untuk mengizinkan pengeksporan sistem berkas terhadap jaringan yang *heterogen* (yang terdiri dari sistem-sistem operasi yang berbeda dan platform yang juga berbeda).

NFS diimplementasikan sebagai sebuah sistem *client/server* yang menggunakan perangkat lunak NFS *server* dan NFS *client* yang berjalan diatas *workstation*. NFS *Server* akan menggunakan protokol NFS untuk mengekspor sistem berkas yang dimilikinya kepada klien NFS agar dapat dibaca oleh *client*, seolah-olah sistem berkas *remote* tersebut merupakan sistem berkas yang dimiliki oleh *client* secara lokal.

2.10.4 XDMCP (*X Display Manager Control Protocol*)

X Window System protocol tampilan dan jaringan yang dikembangkan oleh MIT. Sistem X dibangun dengan fungsi jaringan dan dengan kemampuan untuk menjalankan sesi grafis pada komputer yang di remote. Didalamnya *X Display Manager* digunakan untuk memulai sebuah session dari system lokal atau dari komputer lain. Permintaan dan permulaan session ditangani oleh XDMCP (*X Display Manager Control Protocol*) dan protokol jaringannya. XDMCP menyediakan sebuah cara untuk menjalankan X−*Terminal* untuk PC dan menggunakan X *Server* untuk menyediakan sebuah antarmuka *client server* antara perangkat keras tampilan (mouse, keyboard, dan video) juga desktop dan juga menyediakan baik infrastruktur *windowing* dan sebuah antarmuka aplikasi

standar. X−*Terminal* dapat ditampilan dengan sebuah jendela (tampilan) tunggal atau banyak jendela, berdasar kepada pengaturan dan kemampuan perangkat lunak *system* X.