BAB II
TEORI PENUNJANG

2.1 *Server*

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* berfungsi untuk menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat didalamnya. Singkatnya, *Server* adalah pengelola semua sumber daya jaringan.

Server menggunakan sistem operasi khusus yang disebut sebagai sistem operasi jaringan atau sering dikenal juga dengan sistem operasi *server*. Didalam sistem operasi *server* terdapat berbagai macam layanan yang menggunakan arsitektur *client – server.* Contoh dari layanan *server* adalah *Domain Name Server.* Untuk menunjang kinerja *server, server* memerlukan spesifikasi perangkat keras yang lumayan besar. Contoh sistem operasi *server* dari Windows adalah Windows NT 3.51, Windows NT 4.0, Windows *server* 2000, Windows *server* 2003, Windows *server* 2008, dan Windows *server* 2012. Sedangkan contoh sistem operasi *server* dari Linux adalah Fedora, RedHat Ubuntu, Suse, dan Mandriva.

2.2 Sistem Operasi

Sistem operasi adalah perangkat lunak dasar yang berfungsi sepenuhnya untuk mengendalikan sistem komputer, yang mengandung sejumlah program, dan beberapa program tergolong sebagai utilitas [1]. Sistem operasi mempunyai tiga fungsi utama yaitu manajemen proses, manajemen sumber daya, dan manajemen data.

1. Manajemen proses mencakup penyiapan, penjadwalan, dan pemantauan proses pada komputer.
2. Manajemen sumber daya berkaitan dengan pengendalian terhadap pemakaian sumber daya dalam sistem komputer yang dikakukan oleh perangkat lunak sistem maupun perangkat lunak aplikasi yang sedang

dijalankan.

3. Manajemen data berupa pengendalian terhadap data masukan dan keluaran, pada *memory* utama maupun *memory* sekunder.

2.2.1 Jenis Sistem Operasi

Sistem operasi yang banyak digunakan saat ini adalah :

1. Microsoft Windows, terdiri dari Windows XP, Windows Vista, Windows 7, dan Windows 8.
2. UNIX, yang menggunakan antarmuka sistem operasi POSIX, seperti SCO UNIX, dan GNU/Hurd.
3. Linux, adalah sistem operasi tiruan UNIX. Sistem operasi Linux antara lain Fedora, Ubuntu, Mint, Debian, Suse, dan RedHat.
4. Mac OS, adalah sistem operasi untuk komputer keluaran Apple yang biasa disebut *Mac* atau *Macintosh,* varian Mac diantaranya Mac Lion, dan Mac Snow Leopard.

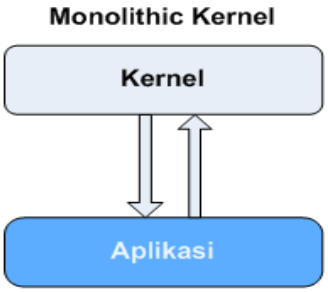
2.2.2 *Kernel*

Kernel adalah suatu perangkat lunak yang menjadi bagian utama dari sebuah sistem operasi. Tugasnya melayani berbagai program aplikasi untuk mengakses perangkat keras komputer secara aman. *Kernel* juga bertugas untuk mengatur kapan dan berapa lama suatu program dapat menggunakan satu bagian perangkat keras, hal ini dikarenakan akses terhadap perangkat keras yang terbatas, sedangkan ada lebih dari satu program yang harus dilayani dalam waktu yang bersamaan.

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 mudah dan seragam. Sehingga abstraksi pada akhirnya memudahkan pekerjaan *programer*. Berdasarkan abstraksinya *kernel* terbagi menjadi :

1. *Monolithic kernel*

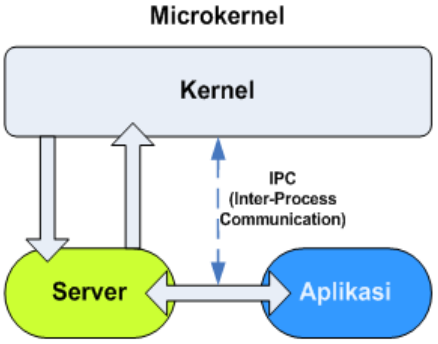
   Kernel yang menyediakan abstraksi perangkat keras.



Gambar 2.1 *Monolithic Kernel*

2. *Microkernel*

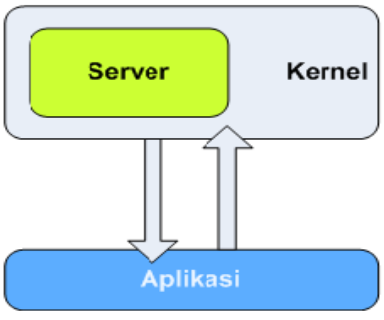
   Kernel yang menyediakan sekumpulan kecil abstraksi perangkat keras sederhana, dan menggunakan aplikasi-aplikasi yang disebut sebagai *server* untuk menyediakan fungsi-fungsi lainnya.



Gambar 2.2 *Microkernel*

3. *Hybrid kernel*

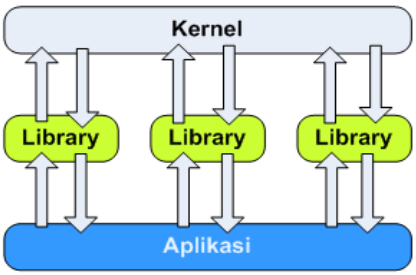
   Kernel yang mirip *microkernel*, tetapi *hybrid kernel* memasukkan beberapa kode tambahan di *kernel* agar menjadi lebih cepat.



Gambar 2.3 *Hybird kernel*

4. *Exokernel*

   Kernel yang tidak menyediakan abstraksi perangkat keras, tapi menyediakan pustaka yang berisi fungsi-fungsi akses ke perangkat keras secara langsung.



Gambar 2.4 *Exokernel*

Dari ke-empat kategori *kernel* yang disebutkan di atas, *kernel* Linux termasuk kedalam kategori *monolithic kernel*. *Kernel* Linux berbeda dengan sistem Linux, *kernel* Linux merupakan sebuah perangkat lunak orisinil yang dibuat oleh komunitas Linux, sedangkan sistem Linux yang dikenal saat ini, mengandung banyak komponen yang dibuat sendiri atau dipinjam dari proyek pengembangan lain. *Kernel* Linux pertama yang dipublikasikan adalah versi 0.01, pada tanggal 14 Maret 1991. Sistem berkas yang didukung hanya sistem berkas Minix. *Kernel* pertama dibuat berdasarkan kerangka Minix (sistem UNIX kecil yang dikembangkan oleh Andy Tanenbaum).

2.3 Linux

Linux adalah sebuah sistem operasi yang dikembangkan oleh Linus Benedict Torvalds dari Universitas Helsinki Finlandia sebagai proyek hobi mulai tahun 1991. Benedict Trovalds menulis Linux untuk sebuah *kernel* pada *processor* 80386, yang merupakan prosesor 32-bit pertama, baru pada tanggal 14 Maret 1994 versi 1.0 mulai diluncurkan. Linux merupakan sistem operasi yang didistribusikan secara *open source*, artinya kode sumber Linux diikutsertakan sehingga dapat dipelajari dan dikembangkan lagi untuk kepentingan pembelajaran. Linux dikembangkan oleh GNU (*General Public License*), Linux dapat digunakan untuk berbagai keperluan seperti jaringan dan pengembangan *software.*

2.3.1 Distribusi Linux

Distribusi Linux merupakan tampilan atau *Graphic User Interfacenya*-nya Linux. Sehingga yang tadinya *kernel* Linux hanya menggunakan bahasa C saja yang ditampilkan, namun berkat distribus*i* Linux, Linux sekarang dapat ditampilkan layaknya sistem operai ternama seperti Windows. Distribusi Linux telah mengalami pertumbuhan yang pesat dari segi popularitas, sehingga lebih populer dari versi UNIX yang menganut sistem lisensi dan berbayar (*proprietary*) maupun versi UNIX bebas lainnya yang pada awalnya menandingi dominasi Microsoft Windows.

Terdapat banyak distributor Linux untuk mempermudah memperoleh Linux, antara lain Fedora, RedHat, Ubuntu, Suse, Slackware, Debian, Mandrake, dan sebagainya. Distribusi-distribusi menggunakan *kernel* yang disetujui Linux Trovald sehingga menjamin kompatibilitas. Kerberadaan banyak distributor menyebabkan persaingan yang sehat dalam meningkatkan layanan ke pengguna. Selain itu saat ini Linux didukung perusahaan besar seperti IBM, Infomix, SyBase, Oracle, Corel, Intel, Compaq, Dell, dan sebagainya.

2.3.1.1 Linux Fedora

Distributor Linux yang digunakan untuk pembuatan Tugas Akhir ini adalah Fedora**.** Linux Fedora adalah sebuah distro Linux berbasis RPM dan YUM yang dikembangkan oleh Fedora Project yang didukung oleh komunitas pemrogram serta disponsori oleh Red Hat. Nama Fedora berasal dari karakter fedora yang digunakan di logo Red Hat. Pada rilis 1 sampai 6 distro ini bernama Fedora Core yang kemudian berubah menjadi Fedora pada rilis ke-7. Fedora dikenal di dunia Linux sebagai sebuah distro yang menjadi *pioneer* dalam penggunaan teknologi terkini.

2.3.1.2 Fedora Cambridge

Fedora Cambridge merupakan pembaharuan dan perbaikan dari versi sebelumnya Fedora 9. Pada awal *release* Fedora Cambridge didistribusikan secara gratis. Cambridge dilengkapi dengan sistem grafik untuk booting terbaru yang disebut dengan playmouth. Fedora Cambridge memiliki fitur antara lain adanya penambahan support untuk hardware seperti perbaiakan array untuk web cams, dan

penanganan printer yang lebih mudah baik untuk printer individu maupun untuk *network* printer. Fedora Cambridge disertai software-software antara lain OpenOffice.org 3.0.0.9, Firefox 3.02, GIMP 2.6, dan Empath 2.24. Paket-paket lainnya dapat ditambahkan ketika installasi sudah selesai. Cambridge dilengkapi dengan 2.6.27 kernel yang akan membuat penginstalan dan penambahan paket menjadi lebih cepat. Cambridge menyediakan DVD tambahan yang terdiri dari lima DVD repositori *everything*, dan satu CD untuk NetInstall.

2.3.2 Repositori Linux

Repositori di Linux adalah lokasi *server* tempat menyimpan koleksi-koleksi *package* (paket) yang menggunakan *packaging system* tertentu, karena tidak semua distro Linux menggunakan *packaging system* yang sama, contohnya Redhat dan distro turunannya seperti Fedora dan CentOS menggunakan packaging system bernama RPM, sedangkan Debian dan distro turunannya seperti Ubuntu dan Linux Mint menggunakan packaging system bernama *deb*.

Paket-paket pada repository itu mengandung metadata, yang isinya berupa informasi mengenai *dependancy* (ketergantungan), pengertian *dependancy* disini adalah daftar paket-paket lain yang dibutuhkan oleh paket yang akan diinstall agar paket tersebut bisa berjalan di sistem operasi Linux.

2.3.2.1 *Redhat Pack Manager*

Redhat Pack Manager (RPM) adalah sebuah sistem manajemen paket. Nama RPM Mengacu kepada dua hal yaitu format berkas paket perangkat lunak, dan sebuah alat bantu installasi perangkat lunak.

2.3.2.2 *Yellowdog Updater Modified*

*Yellowd*og *Updater Modified* (YUM) adalah aplikasi instalasi program dengan menggunakan repository program, bisa dari file, Internet, FTP atau share. Dengan cara seperti ini maka proses instalasi program di Linux akan jadi lebih mudah karena dependensi sudah carikan oleh program tersebut. Program yum dipopulerkan pada distro Linux Fedora dan RedHat.

2.4 Jaringan Komputer

Jaringan Komputer adalah suatu himpunan *interkoneksi* sejumlah komputer yang *autonomous* (mandiri). Dua buah komputer dikatakan terinterkoneksi bila keduanya dapat saling bertukar informasi. Betuk koneksinya tidak harus melalui kawat tembaga saja melainkan dapat menggunakan serat optik, gelomabang mikro, atau satelit komunikasi.[2]

2.4.1 Jenis Jaringan Komputer

Jenis- jenis jaringan komputer dapat dibedakan berdasarkan beberapa kriteria, seperti media transmisi, pola pengoperasian, dan luas area.

1. Berdasarkan media transmisi yang digunakan, jaringan komputer dapat dibagi menjadi :
   a. *Wire network* atau *wireline network*

      Wire network adalah jaringan komputer yang menggunakan kabel sebagai media penghantarnya, kabel yang umum digunakan biasanya menggunakan bahan dasar tembaga dan serat optik. Kabel tembaga biasanya digunakan pada LAN, sedangkan untuk MAN dan WAN menggunakan gabungan kabel tembaga dan serat optik.
   b. *Wireless network*

      Wireless network adalah jaringan komputer yang menggunakan media penghantar berupa gelombang mikro dan cahaya. Frekuensi yang digunakan *wireless network* biasanya 2.4 GHz dan 5.8 GHz.

2. Berdasarkan pola pengoperasian atau fungsi masing-masing komputer maka jaringan komputer dibagi menjadi :

   a. *Peer to peer*

      Peer to peer adalah jenis jaringan komputer dimana setiap komputer bisa menjadi *server* sekaligus *client.* Setiap komputer dapat memberikan atau menerima akses dari satu komputer ke komputer lainnya.
   b. *Client server*

      Client server adalah jaringan komputer yang salah satu atau lebih

komputernya difungsikan sebagai *server* untuk melayani komputer lain. Komputer yang dilayanai oleh *server* disebut *client. Client server* banyak digunakan oleh *internet* dan *intranet.*

3. Berdasarkan cakupan areanya, jaringan komputer terbagi menjadi LAN, MAN, WAN, dan internet. Untuk lebih jelasnya perhatikan tabel di bawah ini:

Tabel 2.1 Jaringan Komputer Berdasarkan Area

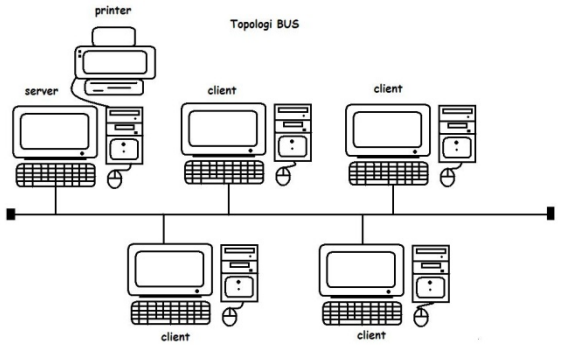
Jarak (meter)	***Network***	**Contoh Area**
10 – 1000	LAN	Gedung
10000 – 100000	MAN	Kota
100000 – 1000000	WAN	Negara
> 1000000	Internet	Antar Negara

2.4.2 Topologi Jaringan

Topologi adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Topologi ini akan mempengaruhi tingkat efektifitas kinerja jaringan. Topologi jaringan terbagi menjadi :

1. Topologi *Bus*

   Merupakan topologi yang menghubungkan semua terminal ke satu jalur komunikasi yang kedua ujungnya ditutup dengan terminator.



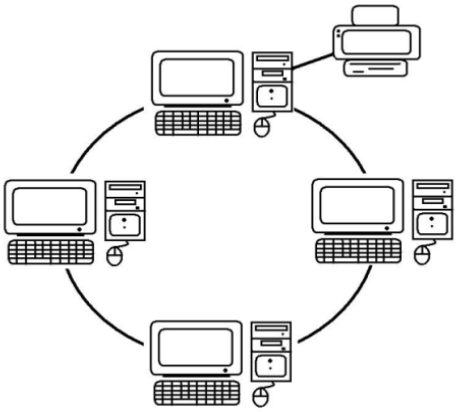
Gambar 2.5 Topologi *Bus*

Kriteria topologi bus :

a. *Linier*
b. Sederhana
c. Mudah instalasi
d. Saling ketergantungan
e. Kabel : Twisted pair, Coaxial, dan Fiber optic
f. Protocol : Enthernet

2. Topologi *Ring*

Pola dari topologi *ring* hampir sama dengan topologi *bus*, tetapi kedua terminal yang berada di ujung saling dihubungkan sehingga hubungan antar terminal berlangsung dalam suatu lingkaran tertutup.



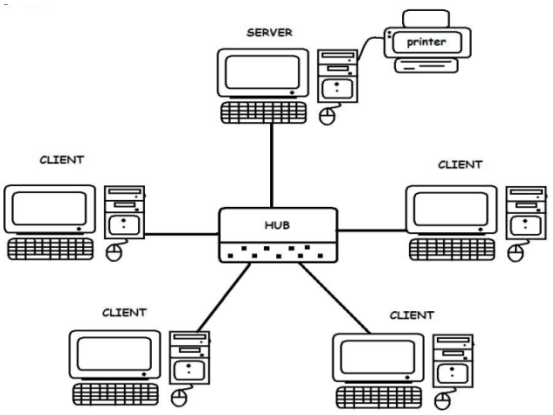
Gambar 2.6 Topologi *Ring*

3. Topologi *Star*

Pada topologi *star*, terdapat sebuah terminal pusat (hub/*switch*) yang mengatur dan mengendalikan semua kegiatan komunikasi data.

Keuntungan dan kerugiannya :

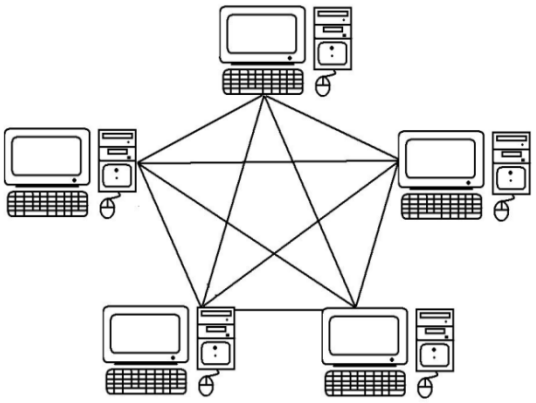
a. Menggunakan HUB/*Switch*
b. Instalasi cukup mahal
c. Kabel *: Twisted pair*
d. *Protocol* : Ethernet



Gambar 2.7 Topologi *Star*

5. Topologi *Mesh*

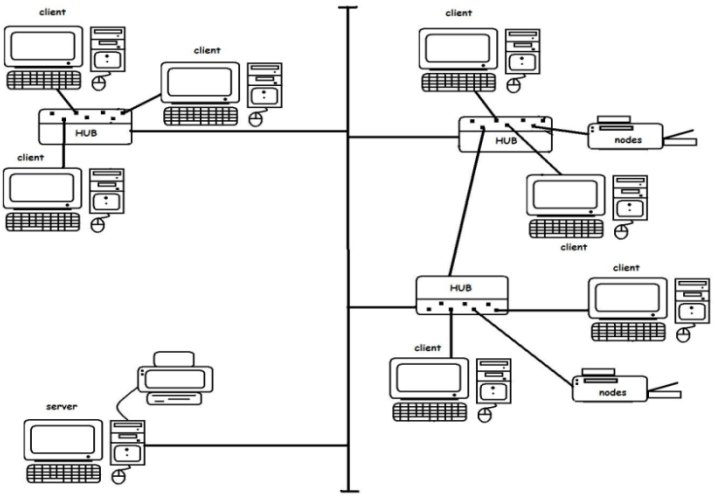
   Topologi *mesh* menghubungkan setiap komputer secara point-to-point. Artinya semua komputer akan saling terhubung satu-satu sehingg tidak dijumpai ada *link* yang putus. Topologi ini biasanya digunakan pada lokasi yang penting atau kritis, seperti instalasi nuklir.



Gambar 2.8 Topologi *Mesh*

6. Topologi *Extended Star*

   Topologi *extende*d *star* merupakan pengembangan dari topologi *star*, cara kerjanya adalah menggabungkan beberapa topologi star menjadi satu kesatuan. Alat yang digunakan untuk menghubungkan masing-masing topologi *star* adalah *hub* atau *switch.*



Gambar 2.9 Topologi *Extended Star*

2.4.3 *IP Address*

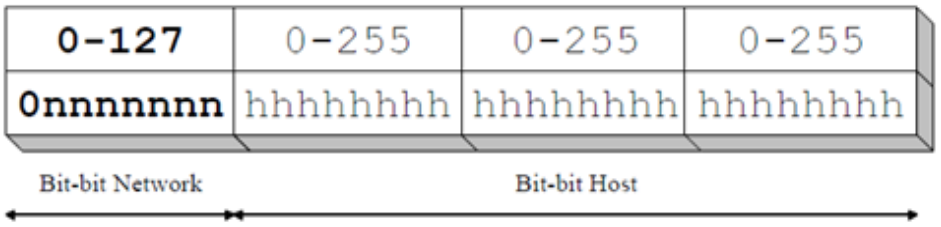
Internet Protocol (IP) merupakan protokol yang terletak pada *layer* internet, IP bersifat *connectionless* dan *unreliable*. IP merupakan inti dari protokol TCP/IP, pada *header* IP ada *field* berisi informasi internet *address* atau *IP address*. *IP address* merupakan identifikasi setiap *host* pada jaringan internet. Artinya tidak boleh ada *host* lain yang menggunakan *IP address* yang sama. IP Address dapat dipisahkan menjadi 2 bagian, yakni bagian *network* (net ID) dan bagian *host* (*host* ID). Net ID merupakan identifikasi suatu *network* dari *network* yang lain, sedangkan *host* ID merupakan identifikasi *host* dalam suatu *network*.

IP address yang sering digunakan saat ini adalah *IP address* versi 4 atau yang sering disebut dengan IPv4. IPv4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IPv4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalamati hingga 4 miliar *host* komputer atau lebih tepatnya 4.294.967.296 *host* di seluruh dunia, jumlah *host* tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4 (karena terdapat 4 oktet) sehingga nilai maksimal dari alamt IPv4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga jumlah *host* yang dapat ditampung adalah 256x256x256x256 = 4.294.967.296 *host*.

IPv4 dibagi menjadi 5 kelas, pembagian kelas-kelas ini ditujukan untuk mempermudah alokasi IPv4 namun hanya 3 kelas pertama yang bisa digunakan untuk kepentingan umum. Kelas-kelas tersebut yaitu :

1. Kelas A

   Bit pertama IPv4 kelas A adalah 0, dengan panjang net ID 8 bit dan panjang *host* ID 24 bit.

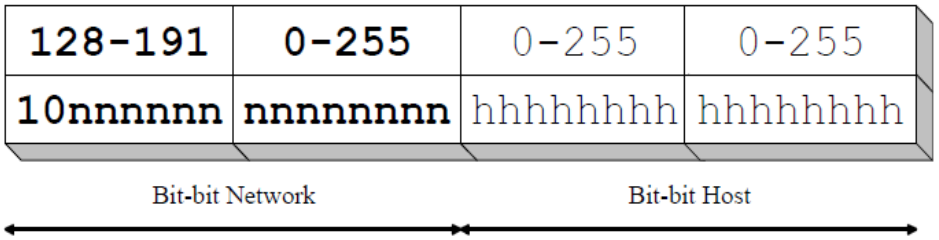


Gambar 2.10 IPv4 Kelas A

IPv4 kelas A mempunyai *range* dari 0-127, Jadi pada kelas A terdapat 128 *network* dengan tiap *network* dapat menampung sekitar 16 juta *host* (255x255x255). IPv4 kelas A digunakan untuk jaringan yang membutuhkan *host* dengan jumlah yang sangat besar.

2. Kelas B

   Dua bit pertama IPv4 kelas B biasanya diset 10, sehingga *byte* pertamanya bernilai antara 128-191. *Network* ID adalah 16 bit pertama, dan 16 bit sisanya adalah *host* ID.

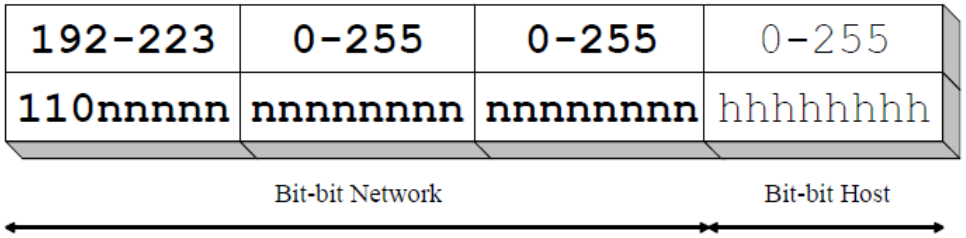


Gambar 2.11 IPv4 Kelas B

IPv4 kelas B ini mempunyai range IP dari 128.0.xxx.xxx sampai 191.255.xxx.xxx, Jadi berjumlah 65.255 *network* dengan jumlah *host* tiap *network* 255 x 255 *host* atau sekitar 65 ribu *host*.

3. Kelas C

Tiga bit pertama IPv4 kelas C biasanya diset 110, sehingga *byte* pertamanya bernilai antara 192-223.



Gambar 2.12 IPv4 Kelas C

Network ID terdiri dari 24 bit, dan 8 bit sisanya adalah *host* ID sehingga dapat terbentuk sekitar 2 juta *network* dengan masing-masing *network* memiliki 256 *host*. IPv4 kelas C sering digunakan untuk jaringan berukuran kecil seperti LAN.

IP address kelas D digunakan untuk keperluan *multicasting*, 4 bit pertama IP address kelas D selalu diset 1110 sehingga byte pertamanya berkisar antara 224-247, sedangkan bit-bit berikutnya diatur sesuai keperluan *multicast group* yang menggunakan IP address ini. Dalam *multicasting* tidak dikenal istilah *network* ID dan *host* ID. IP address kelas E tidak diperuntukkan untuk keperluan umum, 4 bit pertama IP address kelas ini diset 1111 sehingga byte pertamanya berkisar antara 248-255.

Tabel 2.2 IPv4 *Public*

Kelas	Oktet Pertama	Subnetmask	Broadcast
A	1-126	255.0.0.0	XXX.255.255.255
B	128-191	255.255.0.0	XXX.XXX.255.255
C	192-223	255.255.255.0	XXX.XXX.XXX.255
D	224-239	-	-
E	240-255	-	-

Oktet pertama digunakan untuk mempermudah pembacaan kelas yang digunakan, sedangkan yang digunakan untuk menentukan kelas IPv4 adalah subnetmask. Subnetmask adalah istilah teknologi informasi dalam bahasa Inggris

yang mengacu kepada angka biner 32 bit yang digunakan untuk membedakan *network* ID dengan *host* ID, menunjukkan letak suatu *host*, apakah berada di jaringan lokal atau jaringan luar.

Broadcast *address* IPv4 adalah alamat yang dibentuk dengan cara mengeset semua bit host menjadi 1 dalam sebuah alamat yang menggunakan kelas (classfull). Contohnya adalah, dalam NetID 192.168.100.22, alamat broadcast-nya adalah 192.168.100.255. Broadcast *address* digunakan untuk melakukan *broadcasting* ke semua host didalam jaringan. contoh ip address 192.168.100.22 subnetmask 255.255.255.0, kemudian lakukan ping ke 192.168.57.255 maka *host* yang aktif di dalam *network* akan melakukan *reply*.

2.4.4 *Domain Name*

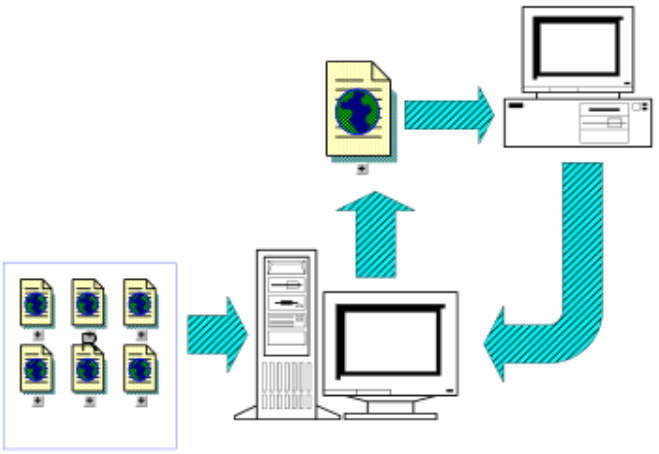
Manusia lebih mudah mengenal nama dibandingkan internet protocol *address* atau yang dikenal dengan *IP address,* itulah sebabnya selama bertahun-tahun para ahli telah mencari cara untuk menggantikan *IP address* menjadi *hostname*. Pada tahun 1981, diusulkan sebuah metode konversi *IP address* menjadi *hostname* dan sebaliknya, metode ini disebut DNS (*Domain Name System*). *Domain name* terdiri dari beberapa bagian yang dipisahkan oleh titik. Bagian terakhir menunjukan level teratas / *top level domain,* biasanya menunjukan jenis organisasi yang menggunakan *domain name* tersebut. Kemudian bagian selanjutnya menunjukan sub bagian dari organisasi tersebut, bagian awal biasanya menunjukan nama *host* atau *server*.

Top level domain adalah deretan kata dibelakang nama *domain,* seperti .com, .net, .co.id, dan lainnya. Ada dua macam / *top level domain* main, yaitu *Global Top Level Domain* (GTLD) dan *Country Code Top Level Domain* (CCTLD). Contoh GTLD adalah .com, .net, .org dan lainnya yang bersifat global, sedangkan CCTLD yang diperuntukkan untuk masing-masing negara, seperti Indonesia dengan kode ID (co.id, net.id, or.id) .

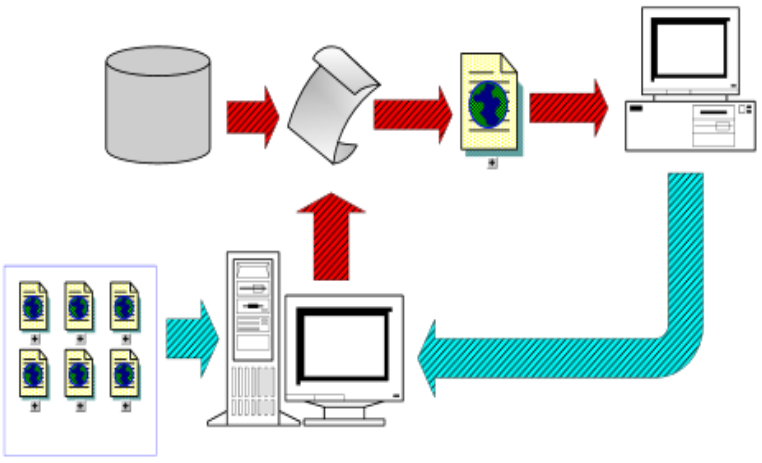
DNS *Server* di Linux biasanya dijalankan oleh program yang bernama *Named*. Program ini merupakan bagian dari paket *Bind* yang dikoordinasikan oleh Paul Vixie. Biasanya program ini terletak di */usr/sbin/named* dan dijalankan pada waktu booting dari */etc/rc.d/init.d/named start*.

2.4.5 *Web Server*

Web server merupakan sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman *web* melalui HTTP atau HTTPS dari klien yang dikenal dengan *browser web* dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman–halaman *web* yang umumnya berbentuk dokumen HTML



Gambar 2.13 *Standard Web Architecture*



Gambar 2.14 *Dynamic web architecture*

Beberapa *web server* yang banyak digunakan di internet antara lain :

1. Apache *Web Server*
2. Internet Information Services (IIS)
3. Sun Java System *Web Server*

2.4.6 Proxy *Server*

Proxy *server* adalah sebuah komputer *server* atau program komputer yang dapat bertindak sebagai komputer yang melakukan request terhadap content dari Internet atau intranet. Proxy *Server* bertindak sebagai gateway terhadap akses Internet untuk setiap komputer didalam jaringan. Proxy *server* tidak terlihat oleh komputer klien. Seorang pengguna yang berinteraksi dengan Internet melalui sebuah proxy *server* tidak akan mengetahui bahwa sebuah proxy *server* sedang menangani request yang dilakukannya.

Web server yang menerima request dari proxy *server* akan menginterpretasikan request-request tersebut seolah-olah request itu datang secara langsung dari komputer klien, bukan dari proxy *server*. Proxy *server* juga dapat digunakan untuk mengamankan jaringan pribadi yang dihubungkan ke sebuah jaringan publik (seperti halnya Internet). Proxy *server* memiliki lebih banyak fungsi daripada router yang memiliki fitur packet filtering karena memang proxy *server* beroperasi pada level yang lebih tinggi dan memiliki kontrol yang lebih menyeluruh terhadap akses jaringan. Proxy dapat dipahami sebagai pihak ketiga yang berdiri ditengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara, sedemikian sehingga pihak pertama dan pihak kedua tidak secara langsung berhubungan, akan tetapi masing-masing berhubungan dengan perantara, yaitu Proxy. Dalam komunikasi jaringan komputer dan internet, proxy *server* mempunyai 3 fungsi utama, yaitu :

a. Connection Sharing
b. Filtering
c. *Caching*

Masing masing fungsi akan dijelaskan lebih lanjut dibawah. Proxy dalam pengertiannya sebagai perantara, bekerja dalam berbagai jenis protokol komunikasi jaringan dan dapat berada pada level-level yang berbeda pada hirarki layer protokol komunikasi jaringan. Suatu perantara dapat saja bekerja pada layer Data-Link, layer *Network* dan layer Trans*port*, maupun layer Aplikasi dalam hirarki layer komunikasi jaringan menurut OSI. Namun pengertian proxy *server* sebagian besar adalah untuk menunjuk suatu *server* yang bekerja sebagai proxy pada layer Aplikasi, meskipun juga akan dibahas mengenai proxy pada level sirkuit. Jaringan

lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu gateway, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. Gateway ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut akan sulit dilakukan bila tidak ada garis batas yang jelas jaringan lokal dan internet.

Squid adalah sebuah daemon yang digunakan sebagai *proxy server* dan *web cache*. Squid memiliki banyak jenis penggunaan, mulai dari mempercepat *server web* dengan melakukan *caching* permintaan yang berulang-ulang, *caching* DNS, *caching* situs *web*, dan *caching* pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga membantu keamanan dengan cara melakukan penyaringan (*filter*) lalu lintas. Squid umumnya didesain untuk berjalan di atas sistem operasi mirip UNIX.

2.4.9 *Port*

Dalam protokol jaringan TCP/IP, sebuah *port* adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya di dalam jaringan. *Port* dapat mengidentifikasikan aplikasi dan layanan yang menggunakan koneksi di dalam jaringan TCP/IP. Sehingga, *port* juga mengidentifikasikan sebuah proses tertentu di mana sebuah server dapat memberikan sebuah layanan kepada klien atau bagaimana sebuah klien dapat mengakses sebuah layanan yang ada dalam di server. *Port* dapat dikenali dengan angka 16-bit (dua byte) yang disebut dengan *port number* dan diklasifikasikan dengan jenis protokol transport apa yang digunakan, ke dalam *Port* TCP dan *Port* UDP. Karena memiliki angka 16-bit, maka total maksimum jumlah *port* untuk setiap protokol trans*port* yang digunakan adalah 65536 buah.

Dilihat dari penomorannya, *port* UDP dan TCP dibagi menjadi tiga jenis, yakni sebagai berikut:

1. *Well-known Port*, yang pada awalnya berkisar antara 0 hingga 255 tapi kemudian diperlebar untuk mendukung antara 0 hingga 1023. *Port* number yang termasuk ke dalam well-known *port*, selalu merepresentasikan layanan jaringan yang sama, dan ditetapkan oleh Internet Assigned Number

Authority (IANA). Beberapa di antara *port-port* yang berada di dalam range Well-known *port* masih belum ditetapkan dan direservasikan untuk digunakan oleh layanan yang bakal ada pada masa depan. *Well-known port* didefinisikan dalam RFC 1060.

2. *Registered Port, Port-port* yang digunakan oleh vendor-vendor komputer atau jaringan yang berbeda untuk mendukung aplikasi dan sistem operasi yang mereka buat. Registered *port* juga diketahui dan didaftarkan oleh IANA tapi tidak dialokasikan secara permanen, sehingga vendor lainnya dapat menggunakan *port* number yang sama. Range registered *port* berkisar dari 1024 hingga 49151 dan beberapa *port* di antaranya adalah *Dynamically Assigned Port*.
3. *Dynamically Assigned Port*, merupakan *port-port* yang ditetapkan oleh sistem operasi atau aplikasi yang digunakan untuk melayani request dari pengguna sesuai dengan kebutuhan. Dynamically Assigned *Port* berkisar dari 1024 hingga 65536 dan dapat digunakan atau dilepaskan sesuai kebutuhan.

Berikut adalah beberapa *Well-known port* yang digunakan dalam jaringan komputer :

a. 20 & 21: *File Transfer Protocol* (FTP)
b. 22: *Secure Shell* (SSH)
c. 23: *Telnet remote login service*
d. 25: *Simple Mail Transfer Protocol* (SMTP)
e. 53: *Domain Name System* (DNS)
f. 80: *Hypertext Transfer Protocol* (HTTP)
g. 110: *Post Office Protocol* (POP3)
h. 119: *Network News Transfer Protocol* (NNTP)
i. 143: *Internet Message Access Protocol* (IMAP)
j. 161: *Simple Network Management Protocol* (SNMP)
k. 194: *Internet Relay Chat* (IRC)
l. 443: *HTTP Secure* (HTTPS)
m. 465: *SMTP Secure* (SMTPS)
n. 3128 & 8080 : *Proxy server*

2.5 *Interface*

Interface (antarmuka) adalah salah satu layanan yang disediakan sistem operasi sebagai sarana interaksi antara pengguna dengan sistem operasi. Antarmuka merupakan komponen sistem operasi yang bersentuhan langsung dengan pengguna. Terdapat dua jenis antarmuka, *Command Line Interface* (CLI) dan *Graphical User Interface* (GUI).

2.5.1 *Command Line Interface*

Command Line Interface (CLI) adalah tipe antarmuka dimana pengguna berinteraksi dengan sistem operasi melalui *text-terminal*. Pengguna menjalankan perintah dan program di sistem operasi tersebut dengan cara mengetikkan baris-baris tertentu. Meskipun konsepnya sama, tiap-tiap sistem operasi memiliki nama atau istilah yang berbeda untuk CLI-nya.

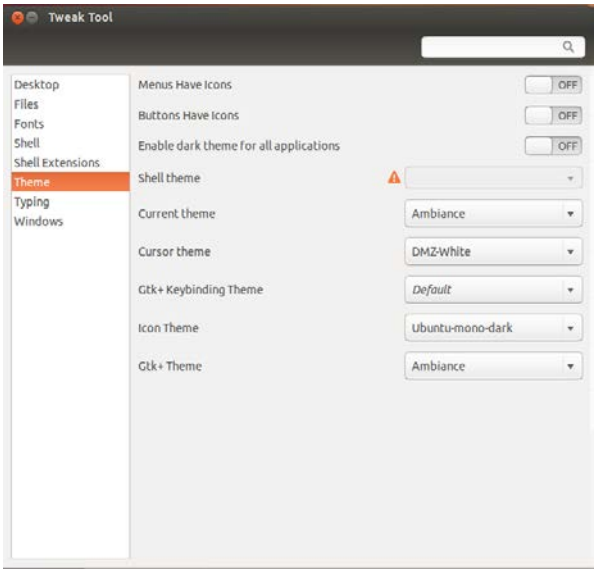
UNIX memberi nama CLI-nya sebagai bash, ash, ksh, dan lain sebagainya. Microsoft Disk Operating System (MS-DOS) memberi nama command.com atau Command Prompt. Sedangkan pada Windows Vista, Microsoft menamakannya PowerShell. Pengguna Linux mengenal CLI pada Linux sebagai *terminal*, sedangkan pada Apple namanya adalah *command shell*



Gambar 2.15 *Command Line Interface*

2.5.2 *Graphical User Interface*

Graphical User Interface (GUI) adalah suatu media *virtual* yang dapat membuat pengguna memberikan perintah tertentu pada komputer tanpa perlu mengetik perintah panjang lebar, namun menggunakan gambar yang tersedia ataupun dengan perintah secara singkat seperti misalnya mengetikan data masukan pada suatu *textbox* atau *inputbox.*



Gambar 2.16 *Graphical User Interface*

2.6 Bahasa Pemrograman

Bahasa pemrograman adalah notasi untuk menulis program yang mana merupakan spesifikasi dari komputasi dan algoritma [*Anthony Aaby, 2004*]. Bahasa *web* atau *Markup Language* (XML, HTML) tidak dimasukkan dalam kelompok bahasa pemrograman. Pola penulisan dalam bahasa tertentu disebut sintaks/*syntax*. Bahasa pemrograman pada dasarnya adalah tekstual karena walaupun ada yang menggunakan komponen visual tetapi tetap diperlukan penulisan notasi-notasi baik itu variabel, perintah-perintah dan hubungan antar komponen.

Bahasa pemrograman memiliki tingkatan yang ditentukan oleh seberapa dekatnya bahasa mengakses mesin komputer. Tingkatan tersebut adalah :

a. Bahasa Tingkat Rendah (*Low Level Language*)

   Biasa disebut bahasa mesin, perintah-perintahnya berupa bahasa mesin. Contoh : Assembly

b. Bahasa Tingkat Menengah (*Middle Level Language*)

   Bahasanya yang perintah-perintahnya sebagian mudah dimengerti manusia dan sebagian lagi merupakan bahasa mesin. Contoh : C

c. Bahasa Tingkat Tinggi (*High Level Language*)

   Bahasa yang perintah-perintahnya sepenuhnya mudah dimengerti manusia. Contoh: Pascal.

Berdasarkan orientasinya, pemrograman dapat digolongkan menjadi :

a. Pemrograman Berorientasi Prosedural (*Procedural Oriented*)
b. Pemrograman Berorientasi Fungsi (*Functional Oriented*)
c. Pemrograman Berorientasi Logika (*Logic Oriented*)
d. Pemrograman Berorientasi Objek (*Objec Oriented*)

Berdasarkan *interface* yang digunakan bahasa pemrograman dibedakan menjadi :

a. *Text Based* / Berbasis Teks

   Bahasa pemrograman yang berjalan di sistem operasi yang berbasis teks atau editornya non grafik. Misalnya : Pascal, C, Java

b. *Graphic Based* / Berbasis Grafik

   Bahasa pemrograman yang berjalan di sistem operasi yang berbasis grafik. Bahasa ini menggunakan komponen GUI (*Graphic User Interface*) yang memudahkan programmer membuat tampilan grafis. Biasanya disebut juga Pemrograman Visual. Misalnya : Delphi, Visual Basic.

2.6.1 Shell

Fungsi pemrograman shell adalah menyusun atau mengelompokkan beberapa perintah shell (internal atupun eksternal command) menjadi kumpulan perintah yang melakukan tugas tertentu sesuai tujuan penyusunnya. Kelebihan shell di linux dibanding sistem operasi lain adalah bahwa shell di linux memungkinkan kita untuk menyusun serangkaian perintah seperti halnya bahasa pemrograman (interpreter language), melakukan proses I/O, menyeleksi kondisi, looping, membuat fungsi, dsb. adalah proses - proses yang umumnya dilakukan oleh suatu bahasa pemrograman, jadi dengan shell di linux kita dapat membuat program seperti halnya bahasa pemrograman, untuk pemrograman shell pemakai unix atau linux menyebutnya sebagai script shell.

Berikut adalah perintah-perintah dasar shell baik itu internal *command* yang telah disediakan shell maupun eksternal *command* atau *utility*, seperti :

- `cd, pwd, times, alias, umask, exit, logout, fg, bg, ls, mkdir, rmdir, mv, cp, rm, clear, ...`

- Utilitas seperti `cat, cut, paste, chmod, lpr,...`
- Redirection (cara mengirim output ke file atau menerima input dari file), menggunakan operator redirect `>, >>, <, <<`, contohnya:

  `ls > data`

  hasil ls dikirim ke file data, jika file belum ada akan dibuat tetapi jika sudah ada isinya akan ditimpa.

  `ls >> data`

  hampir sama, bedanya jika file sudah ada maka isinya akan ditambah di akhir file.

  `cat < data`

  file data dijadikan input oleh perintah `cat`

Tidak seperti sistem operasi lain yang hanya menyediakan satu atau dua jenis shell, sistem operasi dari keluarga Unix misalnya Linux sampai saat ini dilengkapi oleh banyak shell dengan kumpulan perintah yang sangat banyak, sehingga memungkinkan pemakai memilih shell mana yang paling baik untuk membantu menyelesaikan pekerjaannya, atau dapat pula berpindah-pindah dari shell yang satu ke shell yang lain dengan mudah,

Beberapa shell yang ada di Linux antara lain:

- Bourne shell (sh),
- C shell (csh),
- Korn shell (ksh),
- Bourne again shell (bash).

Masing-masing shell mempunyai kelebihan dan kekurangan yang mungkin lebih didasarkan pada kebutuhan pemakai yang makin hari makin meningkat, untuk dokumentasi ini shell yang digunakan adalah bash shell dari GNU, yang merupakan pengembangan dari Bourne shell dan mengambil beberapa feature (keistimewaan) dari C shell serta Korn shell

2.6.2 Bash Shell

Bash adalah singkatan dari Bourne Again SHell, versi baru dari Bourne Shell yang dikembangkan oleh Steve Bourne. Bash ini adalah shell yang paling umum digunakan pengguna Linux.

Konsep kerja dari pemrograman bash shell hampir mirip dengan bahasa pemrograman lainnya. Pemrograman bash shell juga menggabungkan perintah-perintah untuk memilih suatu kondisi, memproses suatu I/O, looping, dan membuat fungsi-fungsi yang dapat dijalankan user. Konsep pemrograman bash shell ini akan mudah dipelajari apabila kita sudah mengetahui perintah-perintah sederhana yang ada di bash shell seperti whoami, cd, cat, dan lainnya. Dengan penguasaan perintah-perintah sederhana ini, pemrograman bash shell akan membuat pekerjaan user menjadi lebih efektif. Hal penting yang harus diketahui sebelum menggunakan pemrograman bash shell adalah konsep mengenai variabel, format syntax dan struktur di dalam pemrograman bash shell itu sendiri.

Format syntax dan struktur dalam bash shell programming adalah hal yang paling penting untuk diketahui. Kita dapat menulis script menggunakan editor seperti vim, emacs, gedit, dan dtpad. Namun vim dan emacs adalah yang paling cocok karena bisa membantu kita dalam membenarkan syntax dari script yang kita tulis jika kita mengaktifkan fiturnya. File script harus disimpan dalam bentuk .sh. Kemudian di headernya harus diberikan komentar bahwa script ini merupakan pemrograman berbasis bash shell. Contoh dari pemrograman bash di Linux adalah sebagai berikut :

```
#!/bin/bash
#kedai

lagi='y'
while  [ $lagi == 'y' ] || [ $lagi == 'Y' ];
do
   clear
   select menu in "Bakso" "Gado-Gado" "Exit";
   case $REPLY in
        1) echo -n "Banyak mangkuk =";
           read jum
           let bayar=jum*1500;
           ;;
        2) echo -n "Banyak porsi =";
           read jum
           let bayar=jum*2000;
           ;;
        3) exit 0
           ;;
        *) echo "Maaf, tidak tersedia"
           ;;
  esac
do
```

```
echo "Harga bayar = Rp. $bayar"
echo "THX"
echo
echo -n "Hitung lagi (y/t) :";
read lagi;


#untuk validasi input
    while  [ $lagi != 'y' ] && [ $lagi != 'Y' ] && [ $lagi
!= 't' ] && [ $lagi != 'T' ];
    do
       echo "Ops, isi lagi dengan (y/Y/t/Y)";
       echo -n "Hitung lagi (y/t) :";
       read lagi;
    done
```

2.6.3 JavaScript

JavaScript adalah sebuah bahasa script yang dicantumkan pada sebuah halaman *web* dan dijalankan pada *browser web*. JavaScript merupakan nama implementasi Netscape Communications Corporation untuk ECMAScript standar, suatu bahasa skrip yang didasarkan pada konsep pemrograman berbasis prototipe. Bahasa ini terutama terkenal karena penggunaannya di situs *web* (sebagai JavaScript sisi klien) dan juga digunakan untuk menyediakan akses skrip untuk *embedded* objek di aplikasi lain. Walaupun memiliki nama serupa, JavaScript hanya sedikit berhubungan dengan bahasa pemrograman Java, dengan kesamaan utamanya adalah penggunaan sintaks C. Secara semantik, JavaScript memiliki lebih banyak kesamaan dengan bahasa pemrograman Self.

Skrip JavaScript yang dimasukkan di dalam berkas HTML ataupun XHTML harus dimasukkan di antara tag <script>…</script>. Contoh skrip untuk menampilkan dialog box berisi Halo Dunia! ketika sebuah tombol diklik oleh pengguna :

```
<input type="button" value="Klik di sini"onclick="halo();">
<script type="text/javascript">
function halo() {
alert( "Halo Dunia!" );
}
</script>
```

2.6.4 HTML

HTML (*Hypertext Markup* Language) adalah bahasa pengkodean untuk menghasilkan dokumen-dokumen hypertext untuk digunakan di World Wide *Web*. Namun HTML bukanlah sebuah bahasa pemrograman. tetapi hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun.

HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language* (SGML). HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu. HTML dibuat oleh *Tim Berners-Lee* ketika masih bekerja untuk *CERN* dan dipopulerkan pertama kali oleh *browser* Mosaic. Selama awal tahun 1990, HTML mengalami perkembangan yang sangat pesat.

Setiap pengembangan HTML pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya. Namun perkembangan resmi dikeluarkan pada bulan November 1995 oleh IETF *(Internet Engineering Task Force*)**.** HTML 2.0 ini merupakan penyempurnaan dari HTML+ (1993). HTML3.0 (1995) memberikan kemampuan lebih daripada versi sebelumnya. Sebuah Usaha dari *World Wide Web Consortium's* (W3C) HTML Working Group pada tahun 1996 menghasilkan HTML 3.2. HTML versi ini secara resmi diterbitkan pada bulan Januari 1997. HTML versi terbaru adalah HTML 4.01 yang dikeluarkan secara resmi oleh W3C pada tanggal 24 April 1998. HTML merupakan perbaikan dari HTML 4.0 yang lebih dulu diterbitkan. (18 Desember
1997).

HTML harus di awali dan diakhiri dengan kode-kode html dimana menunjukkan bagaimana keluarannya pada saat di jalankan pada sebuah *web browser*. Bahasa pemrograman itu dipahami oleh semua komputer di dunia atau internet. Perintah **html** harus berada didalam tanda **<..>**, yang digunakan untuk menampilkan fungsi – fungsi tertentu misalnya fungsi link dan sebagainya.

Dinamakan ***Hypertext*** karena kita bias membaca isi dari sebuah dokumen/berita tanpa harus membacanya baris demi baris. Kita bias melompat – lompat dari satu topik ke topik lainnya. Dinamakan ***MarkUp Languange*** karena dokumen html mengandung tanda atau perintah tertentu untuk menentukan tampilan teks, gambar, dan sebagainya beserta fungsi fungsi yang diinginkan.

Seperti umumnya dokumen lain, dokumen HTML terdiri dari teks-teks dan bahkan lebih dari dari itu. Dokumen HTML juga dapat mengandung suatu gambar, suara, ataupun video. Satu hal yang membedakan dokumen HTML dengan dokumen-dokumen lainnya adalah adanya elemen-elemen HTML beserta tag-tagnya. Elemen dan tag HTML berfungsi untuk menformat atau menandai suatu bagian tertentu dari dokumen HTML dan juga untuk menentukan struktur bagian tersebut dalam dokumen. HTML.Elemen dan Tag inilah yang merupakan ciri utama dari suatu dokumen HTML. Secara garis besar, untuk menulisakn sebuah dokumen HTML dibutuhkan kerangka penulisan dengan tag-tag dasar, yaitu HTML, HEAD,TITLE dan BODY.

Ada dua komponen utama pembentuk dokumen HTML, dua komponen itu adalah elemen dan tag. Dengan adanya dua komponen ini, maka kita dapat membuat dokumen HTML dengan baik.

a. Elemen

Dalam dokumen HTML, elemen dibagi menjadi dua kategori utama, yaitu elemen-elemen <HEAD> yang memberikan informasi tentang dokumen tersebut.

```
<html>
<head>
<title>teks pada title bar web browser</title>
</head>
<body>
Berisi tentang text, gambar, atau apapun yang
tampil pada dokumen web.
</body>
</html>
```

Elemen <HEAD> ini meliputi judul dokumen atau hubungannya dengan dokumen lain. Elemen lain yaitu elemen-elemen <BODY> yang menentukan bagaimana isi suatu dokumen ditampilkan oleh *browser*. Isi elemen ini misalnya paragraf, list (daftar), tabel dan lain-lain. Di dalam elemen HTML terdiri dari tag-tag beserta teks yang ada dalam tag-tag tersebut.

b. Tag

Pada saat *web browser* menampilkan suatu *web page*, *browser* tersebut akan membaca teks-teks pada dokumen HTML, dan mencari kode khusus yang disebut tag. Tag diapit oleh tanda <>. Tag biasanya merupakan pasangan, yang disebut tag awal dan tag akhir. Tag awal dinyatakan dalam bentuk *<nama tag>* sedang tag akhir dinyatakan dalam bentuk *</nama tag>*.

HTML tidak hanya menyediakan teks saja dalam dokumennya, tetapi juga mampu menampilkan objek-objek multimedia seperti gambar, suara, video, dan bahkan telah merambah lebih jauh dengan adanya VRML serta applet-applet Java. Untuk informasi yang berupa teks, HTML telah menyediakan bermacam-macam elemen, seperti:

a. *Paragraph*

Elemen <P>…</P> digunakan untuk menandai sekumpulan teks sebagai suatu paragraph. Tag <P> untuk awal paragraf dan tag </P> digunakan untuk mengakhiri paragraf. Tag paragraf memiliki atribut yang dapat dipakai sebagai tambahan untuk pemformatan paragraph.

b. *Blockquote*

Tag <BLOCKQUOTE> digunakan untuk menuliskan suatu kutipan teks. *Browser* biasanya menampilkan kutipan teks dengan mengidentifikasikan teks tersebut atau dengan mengabaikan spasi dalam teks seperti tag paragraph. Kutipan teks dinyatakan dengan indentasi yang menjorok ke dalam dan berbeda dalam satu paragraf.

c. *Preformatted text*

Pada tag paragraf, penekanan tombol *ENTER, Tab*, Spasi tidak memberikan pengaruh pada *web page* yang dibuat. Jika akan menambahkan spasi dan *enter* pada teks dalam *web page* dengan menggunakan tag <PRE>…..</PRE>. Sehingga teks yang berada di dalam tag pre akan mengikuti sesuai dengan pengetikan yang dilakukan.

d. *Divider*

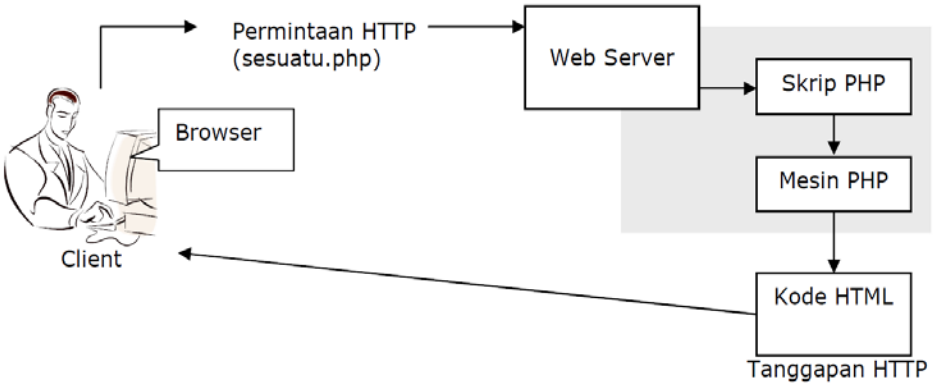
Elemen DIV digunakan untuk membagi-bagi dokumen HTML dalam suatu hierarki yang terstruktur. Teks yang dikelilingi oleh tag

<DIV>……..</DIV> akan diformat menurut nilai atribut ALIGN yang ditentukan di dalamnya. Nilai atribut ALIGN yaitu, *Left, Center* dan *Right*. Penggunaan DIV *dengan Align=”center”* dapat diganti dengan tag <CENTER>….<CENTER>. Keduanya menghasilkan hasil yang sama.

2.6.5 PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman *web* bersifat *serverside*, artinya bahasa berbentuk *script* yang disimpan dan dijalankan di komputer *server* (*WebServer*) sedang hasilnya yang dikirimkan ke komputer *client* (*WebBrowser*) dalam bentuk *script* HTML (Hypertext Mark up Language).

Konsep kerja PHP diawali dengan satu permintaan suatu halaman *web* oleh *browser*. Berdasarkan *URL (Uniform Resource Locator)* atau dikenal dengan alamat Internet, *browser* mendapat alamat dari *webserver*, mengidentifikasikan alamat yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *Web Server*. Selanjutnya *Web Server* akan mengirimkan isinya ke mesin php dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode html) ke *web server*, selanjutnya *web server* menyampaikan ke client .



Gambar 2.17 Konsep Kerja PHP

Salah satu kelebihan dari PHP adalah mampu berkomunikasi dengan berbagai data base yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari database, merupakan hal yang mudah untuk diimplementasikan.

Halaman *web* biasanya disusun dari kode-kode html yang disimpan dalam sebuah file berekstensi ***.html***. File html ini dikirimkan oleh *server* (atau file) ke *browser*, kemudian *browser* menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program php, program ini harus diterjemahkan oleh *web-server* sehingga menghasilkan kode html yang dikirim ke *browser* agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode html sehingga dapat langsung ditampilkan bersama dengan kode-kode html tersebut. Program php dapat ditambahkan dengan mengapit program tersebut di antara tanda **<?** dan **?>**. Tanda-tanda tersebut biasanya disebut tanda untuk *escaping* (kabur) dari kode html. File html yang telah dibubuhi program php harus diganti ekstensi-nya menjadi .php atau .php3.

Ketika *e-commerce* semakin berkembang, situs-situs yang statispun semakin ditinggalkan, karena dianggap sudah tidak memenuhi keinginan pasar, padahal situs tersebut harus tetap dinamis. Pada saat ini bahasa PERL dan CGI sudah jauh ketinggalan jaman sehingga sebagian besar designer *web* banyak beralih ke bahasa *server*-side scripting yang lebih dinamis seperti PHP.

Seluruh aplikasi berbasis *web* dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan system database di dalam *web*. Sistem database yang dapat didukung oleh PHP adalah :

1. Oracle
2. MySQL
3. Sybase
4. PostgreSQL

PHP dapat diinstal sebagai bagian atau modul dari apache *web server* atau sebagai CGI script yang mandiri. Banyak keuntungan yang dapat diperoleh jika menggunakan PHP sebagai modul dari apache, di antaranya adalah :

1. Tingkat keamanan yang cukup tinggi
2. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemograman *web* lainnya yang berorientasi pada *server*-side scripting.

3. Akses ke system database yang lebih fleksibel. seperti MySQL. Dalam modul ini kita akan mempelajari PHP sebagai *server*-side scripting yang menggunakan apache sebagai *webserver*.

2.6.6 CSS

CSS (*Cascading Style Sheet*) adalah suatu dokumen yang digunakan untuk melakukan pengaturan halaman *web* dan juga berfungsi untuk mempersingkat penulisan tag HTML seperti font,color,text, dan table menjadi lebih ringkas sehingga tidak terjadi pengulangan tulisan. Menggunakan CSS tidak memerlukan perangkat lunak tertentu karena merupakan *script* yang telah *embeded* dengan HTML. *Style sheet* merupakan feature yang sangat penting dalam membuat Dynamic HTML. Meskipun bukan merupakan suatu keharusan dalam membuat *web*, akan tetapi penggunaan *style sheet* merupakan kelebihan tersendiri. *Style sheet* digunakan untuk mengontrol dan memanage *style-style* yang ada, s*tyle sheet* mendeskripsikan bagaiman tampilan document HTML di layar.

Seorang *Web Developer* cukup mendefinisikan CSS untuk dapat diterapkan disemua *web pages* yang diinginkan. Sehingga proses pengaturan tampilan *web* dapat dilakukan secara otomatis. CSS dapat diimplementasikan kedalam 3 cara :

a. *Inline Style*, yaitu *style* yang menjadi atribut dalam tag HTML, atributnya adalah *style*. Cara pendefinisian *property* dan nilainya adalah sebagai berikut : `<style=”property-1: nilai; property-2: nilai ;…..property-n: nilai”>`, contoh :

```
1  <html>
2  <head>
3   <title>Contoh Bentuk Inline </title>
4   </head>
5  <body bgcolor="#FFFFFF">
6  <p id="cth1">
7   Ini adalah contoh tag P tanpadiformat menggunakan CSS </p>
8
9  <p id="cth2" style="font-size:20pt">
10  Tag P ini diformat dengan besar font 20 point </p>
11
12 <p id="cth3" style="font-size:14pt; color:red">
13  Tag P ini diformat dengan besar font 14 point, dan menggunakan warna merah </p>
14  </body>
15  </html>
```

Gambar 2.18 CSS *Inline Style*

b. *Embed Style*, yaitu *style* yang berada di dalam tag <HEAD>. *Embed style* CSS didefinisikan sebagai berikut :

`<STYLE type="text/css">…</STYLE>`, contoh:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Guide</title>
<style type="text/css">
p {
color: #FFFFFF;
font-family: arial;
background: #FF0000;
font-weight: bold;
}

p.group {
color: #FF0000; /* membuat warna text putih */
font-family: arial;
background: #FFFFFF; /* membuat background putih */
}

p#one {
color: blue;
font-family: "times new roman";
}
</style>
</head>

<body>
<p>Ini adalah paragraf 1. Ini adalah paragraf 1. Ini adalah paragraf 1.
Ini adalah paragraf 1. Ini adalah paragraf 1. </p>
<p class="group">Ini adalah paragraf 2. Ini adalah paragraf 2. Ini
adalah paragraf 2. Ini adalah paragraf 2. Ini adalah paragraf 2. </p>

<p class="group" id="one">Ini adalah paragraf 3. Ini adalah paragraf 3.
Ini adalah paragraf 3. Ini adalah paragraf 3. Ini adalah paragraf 3.

</p>
</body>
</html>
```

Gambar 2.19 CSS *Embed Style*

c. *External Style Sheet*, yaitu *style* CSS berada terpisah pada sebuah file .css, untuk memanggilnya menggunakan *script* di dalam dokumen HTML dengan cara:

`<Link rel="stylesheet"type="text/css"href="file.css" />`

```
style.css
p {
color: #FFFFFF;
font-family: arial;
background: #FF0000;
font-weight: bold;
}

p.group {
color: #FF0000; /* membuat warna text putih */
font-family: arial;
background: #FFFFFF; /* membuat background putih */
}

p#one {
color: blue;
font-family: "times new roman";
}
```

Gambar 2.20 CSS *External Style Sheet*

Halaman HTML yang sudah direlasi dengan CSS,

```
Lantas… bagaimana cara mengintegrasikan style.css tersebut ke halaman web yang kita
buat? Caranya adalah dengan memberikan perintah
<link rel="stylesheet" type="text/css" href="/style.css" />
pada bagian header halaman web.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Guide</title>
<link rel="stylesheet" type="text/css" href="/style.css" />
</head>

<body>
<p>Ini adalah paragraf 1. Ini adalah paragraf 1. Ini adalah paragraf 1.
Ini adalah paragraf 1. Ini adalah paragraf 1. </p>
<p class="group">Ini adalah paragraf 2. Ini adalah paragraf 2. Ini
adalah paragraf 2. Ini adalah paragraf 2. Ini adalah paragraf 2. </p>
<p class="group" id="one">Ini adalah paragraf 3. Ini adalah paragraf 3.
Ini adalah paragraf 3. Ini adalah paragraf 3. Ini adalah paragraf 3.

</p>
</body
</html>
```

Gambar 2.21 HTML *External Style Sheet*