BAB II

LANDASAN TEORI

Dalam pembuatan laporan Tugas Akhir ini digunakan sejumlah teori yang diperlukan umtuk membantu memecahkan permasalahan yang ada dan juga untuk menunjang kegiatan yang dilakukan. Beberapa landasan teori yang dikemukakan tersebut meliputi konsep dasar dan definisi yang berkaitan dengan perangkat lunak yang digunakan serta faktor-faktor pendukung dalam melaksanakan perancangannya.

2.1. Jaringan Komputer

Jaringan Komputer[1] adalah sebuah kumpulan Komputer, Printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar informasi dan data dan bersama-sama menggunakan *Hardware* atau *Software* yang terhubung dengan jaringan.

Setiap komputer atau *Peripheral* yang terhubung dengan jaringan disebut *Node*. Sebuah jaringan komputer dapat memiliki lebih dari satu atau puluhan, ribuan atau bahkan jutaan node. Istilah *link* digunakan untuk merepresentasikan kabel, peralatan terminal atau media lain yang dapat menghubungkan komunikasi antar *node*.

[1] Irawan, Budhi, Jaringan Komputer, Graha Ilmu, Yogyakarta 2005

Komputer-komputer yang terhubung dalam suatu jaringan dapat berkomunikasi satu sama lain menggunakan sejumlah prosedur aturan yang disebut dengan protokol. Protokol tersebut mendefinisikan *syntax* (susunan, format, pola *bit* dan *bytes*) dan *semantic* (pengendalian sistem dan konteks informasi atau pengertian dari pola deretan *bit* dan *byte*) dari pertukaran data serta sejumlah karakteristik lain seperti laju data dan pewaktuan (*Timing*).

Ciri-ciri jaringan komputer :

a. Berbagi perangkat keras (*Hardware*).
b. Berbagi perangkat lunak (*Software*).
c. Berbagi saluran komunikasi (*Internet*).
d. Berbagi data dengan mudah.
e. Memudahkan komunikasi antar pemakai jaringan.

2.1.1Jenis-Jenis Jaringan Komputer

Jaringan Komputer dapat dibedakan berdasarkan cakupan geografisnya. Ada empat kategori utama jaringan komputer[2] yaitu :

2.1.1.1***Local Area Network*** **(LAN)**

Local Area Network (LAN) biasa digunakan untuk menghubungkan komputer dalam sebuah gedung atau kampus yang berjarak sampai beberapa kilometer. Dalam LAN dimungkinkan penggunaan secara bersama suatu sumber daya (*resouce*, misalnya *Printer, File Server*) dan saling bertukar informasi.

[2] Irawan Budhi, Jaringan Komputer, Graha Ilmu, Yogyakarta 2005, hal 5

LAN pada intinya adalah sistem pengkabelan (*Cabling System*). Dengan menginstalasi *Private Cabling System* (yang dengannya perangkat komputer dapat dikoneksikan) dengan memakai *Protocol*, Sehingga komunikasi antar seluruh stasiun (*Workstation*) menjadi mungkin.

Karena LAN sangat berhubungan dengan sistem pengkabelan, yang tentu saja akan berkaitan dengan tipe kabel dan topologi sistem. Jenis kabel akan mempengaruhi kinerja sedangkan topologi akan berperan pada biaya yang harus dikeluarkan secara total dan menyangkut pada keuletan jaringan.

2.1.1.2 ***Metropolitan Area Network*** **(MAN)**

Metropolitan Area Network (MAN) pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (Swasta) atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.

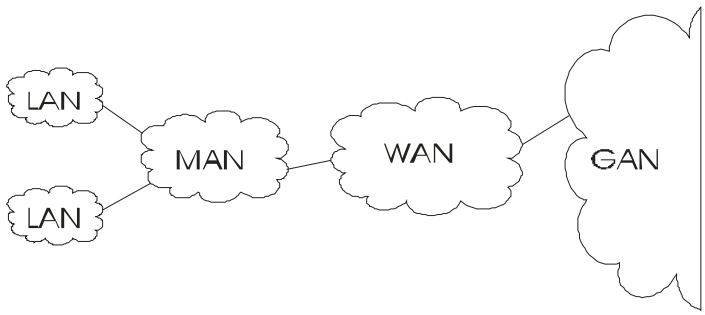
2.1.1.3 ***Wide Area Network*** **(WAN)**

Wide Area Network (WAN), Jangkauannya mencakup daerah geografis yang luas, seringkali mencakup sebuah negara bahkan benua. WAN terdiri dari kumpulan mesin-mesin yang bertujuan untuk menjalankan program-program (aplikasi) pemakai. Dalam WAN biaya untuk peralatan transmisi sangat tinggi dan biasanya dimiliki dan dioperasikan sebagai suatu jaringan publik.

2.1.1.4 ***Global Area Network*** **(GAN)**

GAN (*Global Area Network*) merupakan suatu jaringan yang menghubungkan negara-negara di seluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps sampai dengan 100 Gbps dan cakupannya mencapai jarak ribuan kilometer, contoh yang sangat baik dari GAN adalah *Internet*.

LAN (*Local Area Network*), MAN (*Metro Area Network*), WAN (*Wide Area Networ*), dan GAN (*Global Area Network*) dapat berinteraksi satu sama lain seperti yang ditunjukan pada gambar dibawah ini.



Gambar 2.1 **Interaksi antara LAN, MAN, WAN, dan GAN**

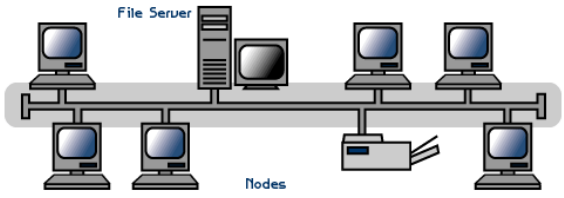
2.1.2Topologi Fisik Jaringan

Hal yang terpenting berkaitan dengan jaringan yaitu arsitektur jaringan komputer yang dapat dibedakan menjadi arsitektur fisik dan arsitektur *logic*. Arsitektur fisik berkaitan dengan susunan fisik sebuah jaringan komputer, biasanya disebut dengan topologi jaringan. Sedangkan arsitektur *logic* berkaitan dengan logika hubungan masing – masing komputer dalam jaringan.

Secara umum topologi jaringan[3] terdiri dari topologo *star, bus*, dan *ring,* dimana ketiganya memiliki karakteristik tersendiri satu dengan yang lainnya.

2.1.2.1 Topologi Bus

Pada topologi bus seluruh komputer dalam sebuah jaringan terhubung pada sebuah bus atau jalur komunikasi data (kabel). Komputer – komputer tersebut berkomunikasi dengan cara mengirimkan dan mengambil data di sepanjang bus tersebut. Pada topologi bus biasanya digunakan kabel koaksial. Seluruh jaringan biasanya merupakan satu saluran kabel yang kedua ujungnya diterminasi dengan beban 50 ohm (dummy load). Instalasi topologi bus merupakan instalasi paling sederhana dan umumnya membutuhkan biaya paling murah dibandingkan jenis–jenis topologi lainnya. Topologi ini banyak digunakan pada jaringan LAN yang kecil lima sampai dengan sepuluh komputer saja.

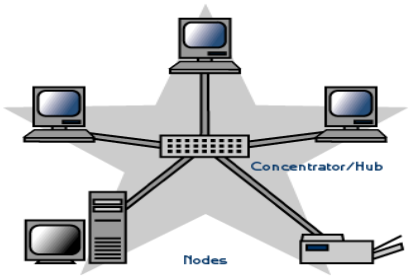


Gambar 2.2. ***Topologi bus Garis Lurus / Linear***

2.1.2.2 Topologi *Ring*

[3] Irawan Budhi, Jaringan Komputer, Graha Ilmu, Yogyakarta 2005, hal 9

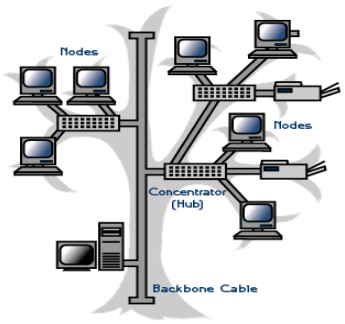
Sesuai dengan namanya ring atau cincin, maka pada topologi ini seluruh komputer dalam jaringan terhubung pada sebuah jalur data yang menghubungkan dan yang membentuk lingkaran tertutup sedemikian rupa sehingga menyerupai sebuah cincin atau ring. Secara umum layout topologi ring juga relatif sederhana. Keuntungan topologi ring sinyal akan mengalir dalam satu arah sehingga dapat menghindari terjadinya tabrakan (*collision*) paket. Karena itu pergerakan paket data dapat lebih cepat dan c*ollision detection* lebih sederhana.



Gambar 2.3. ***Topologi Jaringan Bintang/Star***

2.1.2.3 **Topologi *Tree - Pohon***

Sebuah topolgi model ini merupakan perpaduan antara topologi Garis Lurus dan Bintang, Yang mana terdiri dari kelompok-kelompok dari workstation konfigurasi Bintang yang terkoneksi ke kabel utama yang menggunakan topologi Garis Lurus/Linear. Topologi ini memungkinkan untuk pengembangan jaringan ynag telah ada, dan memungkinkan sebuah perusahaan mengkonfigurasi jaringan sesuai dengan kebutuhannya.



Gambar 2.4. ***Topologi Jaringan Pohon***

2.2 Komunikasi Data

Pada dasarnya komunikasi data[4] merupakan proses pengiriman informasi diantara dua titik menggunakan kode biner melewati saluran transmisi dan peralatan switching, bisa antara komputer dan komputer, komputer dengan terminal, atau komputer dengan peralatan, atau peratalan dengan peralatan.

Komponen-komponen dasar sistem komunikasi data antara lain :

1. ***Sumber (pemancar atau pengirim)***

   Yaitu pengirim atau pemancar informasi data. Karena pembahasan berkisar pada sistem komputer maka pemancar adalah sistem komputer. Komunikasi data dapat juga berlangsung dua arah sehingga pemancar juga dapat berfungsi sebagai penerima.

2. ***Media transmisi***

[4] Diktat Kuliah Komunikasi Data, 2005. hal 6

Yaitu saluran tempat informasi tersebut disalurkan ketempat tujuan. Media Yang dipergunakan dapat berupa : kabel, udara, cahaya, dan sebagainya.

3. ***Penerima***

Yaitu alat yang menerima informasi yang dikirimkan.



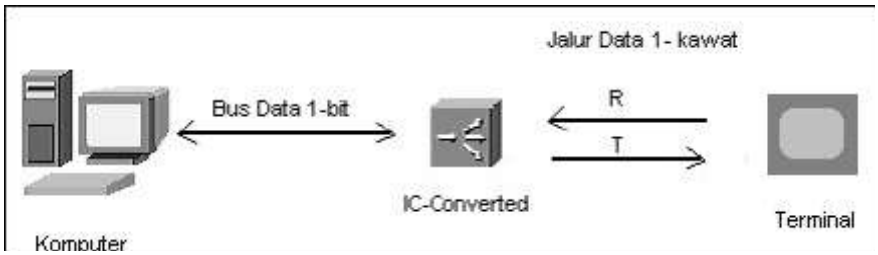
Gambar 2.5 **Komponen Dasar Sistem Komunikasi**

2.2.1Mode Transmisi

Jenis transmisi sinyal data atau informasi dalam suatu media komunikasi dapat dikelompokkan menjadi 2 mode, yaitu transmisi serial dan transmisi parallel.

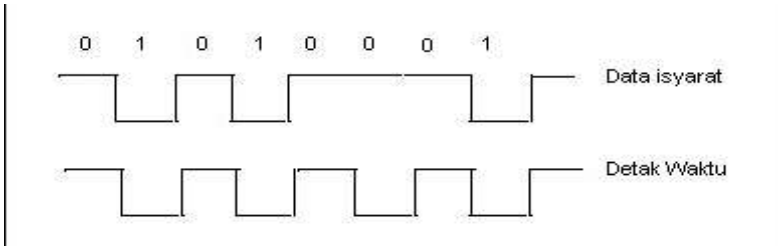
1. Serial

Pada Pengiriman seri, Data pararel internal diteruskan ke pengubah *pararel-serial* (IC Converted), bit-bit dikirimkan secara berurutan (tidak serempak) dan kecepatan pemindahan data lebih rendah dan mode transmisi pararel. Pengiriman dimulai dari LSB (Least Significant Bit) dan diakhiri MSB (Most Significant Bit). Penerima harus memecah isyarat data yang sama pada waktu yang tepat sebelum membentuk kembali karakter yang diterima.



Gambar 2.6 **Mode Transmisi Serial**

Agar data yang diterima itu benar maka selang waktu yang digunakan oleh pengirim dan penerima harus sama. Untuk keperluan tersebut mka pengirim dan penerima harus menambahkan "detak" (*Time Pulse*).



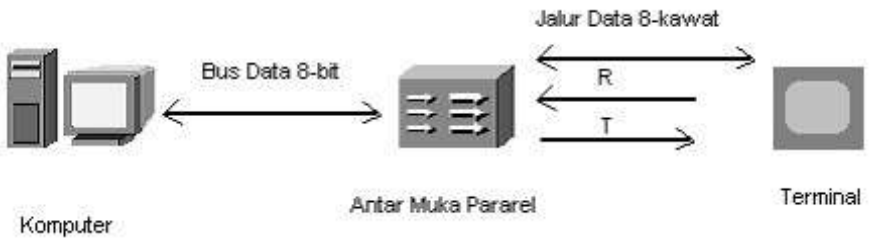
Gambar 2.7 **Detak (Time Pulse)**

2. Paralel

Pada transmisi paralel, suatu konektor yang terdiri dari 7 atau 8 bit (ASCII) ditransmisikan secara serentak setiap saat, seperti yang ditunjukkan pada gambar 2.8. Misalnya, bila digunakan kode ASCII, maka dibutuhkan sebanyak delapan jalur untuk mentransmisikan sekaligus 8 bit untuk satu karakter kode ASCII.

Pengiriman dengan model paralel ini memiliki kecepatan yang tinggi, karena setiap saat yang ditransmisikan secara paralel adalah bit-bit yang memiliki

satu karakter. Komunikasi paralel biasanya digunakan untuk komunikasi jarak dekat, biasanya transmisi jenis ini digunakan untuk mentransmisikan sinyal didalam komputer atau antara komputer ke printer.



Gambar2.8 **Model Transmisi Paralel**

2.3 Sistem Terdistribusi

Sistem Terdistribusi[5] terdiri dari dua kata, yaitu "Sistem" dan "Terdistribusi". Sistem merupakan sekumpulan elemen yang saling berhubungan satu dengan yang lainnya dan membentuk satu kesatuan untuk menyelesaikan satu tujuan yang spesifik atau menjalankan seperangkan fungsi. Adapun terdistribusi berasal dari kata "distribusi" yang merupakan lawan kata "sentralisasi", yang artinya penyebaran, sirkulasi, penyerahan, pembagian menjadi bagian-bagian kecil.

Berawal dari pengertian kata-kata pembentuknya, Sistem Terdistribusi dapat didefinisikan sebagai suatu kesatuan dari elemen-elemen yang saling

[5] Budi Sutedjo Dharma Oetoma dan Esther Wibowo,
Konsep & Aplikasi Client Server dan Sistem Terdistribusi, hal. 41

berinteraksi secara sistematis dan teratur untuk mendistribusikan data, informasi, proses, objek dan layanan dari dan kepada pengguna yang terkait di dalamnya.

Adapun infrastruktur utama dari aplikasi Sistem Terdistribusi ini, meliputi antara lain :

1. Jaringan komputer baik dalam skala lokal (LAN), metropolitan (MAN), skala luas (WAN) maupun skala global (Internet).
2. Beragam perangkat keras dan lunak, serta pengguna yang berada dan saling terkait dalam sistem jaringan yang membentuknya.

Untuk mengembangkan suatu Sistem terdistribusi, perlu diperhatikan beberapa aspek yang merupakan suatu tantangan bagi para pengembang Sistem Terdistribusi, yaitu sebagai berikut :

1. Keanekaragaman (*heterogeneity*)

   Sitem Terdistribusi mampu mendukung berbagai jenis sistem operasi, perangkat keras, dan perangkat lunak.
2. Keterbukaan (*openness*)

   Pengembangan Sistem Terdistribusi yang dilakukan dengan menambahkan komponen-komponen baru dapat dilakukan oleh programer yang berbeda-beda.
3. Keamanan (*security*)

   Sistem Terdistribusi harus dapat menyediakan keamanan yang memadai bagi sumber daya yang digunakan bersama dan pesan yang diantar dalam system.
4. Skalabilitas (*scalability*)

Ukuran Sistem Terdistribusi dapat diubah dan tetap dapat berjalan dengan baik. Perubahan dapat dilakukan dari segi jumlah pengguna maupun dari segi kekuatan perangkat keras komputer-komputer dalam Sistem Terdistribusi itu sendiri.

5. Penanganan masalah (*error-handling*)

   Kerusakan yang terjadi pada satu komputer dalam Sistem Terdistribusi tidak mempengaruhi kinerja sistem secara keseluruhan.

6. Kebersamaan (*concurrency*)

   Apabila terjadi permintaan layanan secara bersama, Sistem Terdistribusi tidak menjadi kacau.

7. Penyembunyian (*transparency*)

   Dalam beberapa buku bahasa Indonesia, istilah tersebut juga sering disebut transparansi, walaupun sebenarnya kurang tepat. Penyembunyian membuat beberapa aspek distribusi tidak tampak oleh pengguna.

2.3.1Arsitektur Sistem terdistribusi

Arsitektur didefinisikan sebagai suatu rancangan untuk penyusunan dan operasi komponen-komponen suatu sistem, di mana rancangan tersebut mengidentifikasi komponen beserta fungsi masing-masing komponen, konektivitas antarkomponen dan mendeskripsikan pemetaan fungsionalitas ke dalam komponen. Dengan demikian, arsitektur Sistem Terdistribusi diharapkan mampu menyediakan gambaran fisik dan logikal dari sistem dan mampu

menyediakan spesifikasi dari komponen-komponen system beserta hubungan antarkomponen dalam sistem.

Adapun beberapa jenis arsitektur Sistem Terdistribusi sebagai berikut :

1. ***Client Server***

   Client menghubungi server untuk mendapatkan data, yang kemudian memformat dan menampilkan pada pengguna. Arsitektur ini terdiri dari 3 jenis, yaitu arsitektur *2-tier,* dan *n-tier.*

2. ***Tightly coupled* (*clustered*)**

   Mesin-mesin terintegrasi yang menjalankan proses yang sama secara bersamaan dengan membagi tugas ke dalam beberapa bagian yang dijalankan masing-masing mesin. Apabila proses telah selesai, hasil pengerjaan masing-masing mesin digabungkan mejadi satu.

3. ***Peer-to-peer***

   Arsitektur di mana tidak ada mesin yang menyediakan layanan atau mengelola sumber daya jaringan sehingga segala tanggung jawab dibagikan diantara seluruh mesin, dikenal dengan istilah *peer*.

2.3.2Client Server

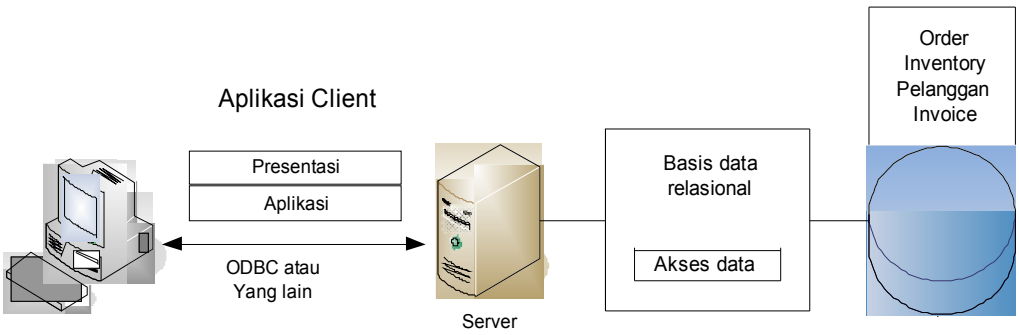
Client/server[6] adalah *sistem* yang memisahkan antara tugas-tugas komputasi antara proses-proses *client* dan *server*. Dengan *sistem client/server*, kekuatan pemrosesan bisa didistribusikan ke banyak mesin *client* dan mesin *server* yang terpisah secara fisik.

[6] Roger S.Pressman, software engineering, fifth edition. Hal 747-767

Sistem bermodel *client/server* juga bisa memanfaatkan *browser web* untuk meringankan kerja *server*. Tugas menampilkan informasi dan menyediakan tampilan pengguna (*user interface*) tidak perlu dilakukan secara langsung oleh *server*, namun diserahkan sepenuhnya kepada *browser web*.

Web di *Internet*, sebenarnya, secara mudah dapat dipandang mirip dengan konsep jaringan dengan *dumb* terminal; pengguna *browser web* mengirim permintaan ke *server web*, lalu menerima informasi dari *server* berupa dokumen statis yang oleh *browser* hanya perlu ditampilkan saja ke layar.

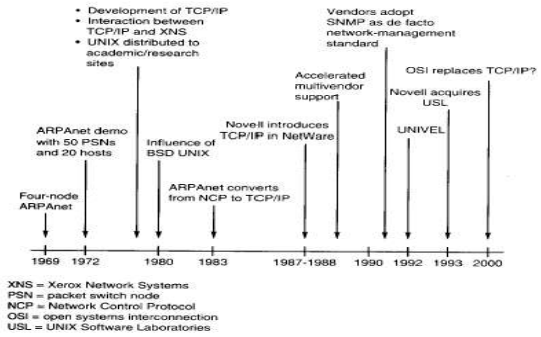
Pada kenyataannya, mesin-mesin yang menjalankan *browser* pada umumnya adalah komputer-komputer moderen, mesin-mesin *client* seharusnya bisa mengerjakan lebih banyak perkerjaan lagi. Tidak seperti *sistem* berbasis *client/server* "tradisional", kenyataannya bahwa penggunaan *browser web* nyaris tidak membutuhkan perawatan yang mahal, *upgrade* berkala dan konfigurasi yang rumit, hal ini membuat *web* (dan *Internet* pada umumnya) menjadi tempat yang menarik untuk mengembangkan *sistem* berbasis *client/server*.



Gambar2.9 **Sistem Client Server**

2.4 Protokol TCP / IP

Gambaran yang umum tentang TCP/IP adalah gabungan dari dua protokol komunikasi yang dipakai untuk kominikasi data, TCP kepanjangan dari *transmission-control-protocol* dan IP perpajangan dari *internet-protocol*. Kedua protokol tersebut dipakai untuk menyatakan sekelompok protokol yang memiliki kaitan dengan protokol TCP dan IP seperti *user-datagram-protocol* (UDP), *file transfer protocol* (FTP), *terminal-emulation-protocol* (TELNET), dll.



Gambar 2.10 ***Evolusi Protokol TCP/IP***

TCP/IP merupakan hasil riset dan pengembangan protokol pada percobaan jaringan *packet-switched* yang ditemukan oleh *defense-advanced-research-project- agency* (DARPA*)* dengan nama ARPAnet pada tahun 1970*,* yang kemudian secara umum disebut protokol TCP/IP. Rangkaian protokol ini berisi beberapa protokol penting dan kemudian menjadi standart internet yang dikeluarkan oleh i*nternetactivities-board* (IAB). IAB adalah badan yang memberikan perhatian terhadap perkembangan protokol yang berhubungan dengan TCP/IP dan internet. Pada gambar 2.10 diatas menunjukkan peristiwa penting yang berkaitan dengan perkembangan protokol TCP/IP, awalnya dimulai

dengan percobaan 4-node ARPAnet pada tahung 1969 dan dilanjutkan dengan demo pada tahun 1972. Pada periode 1978 samapai 1980 terjadi perkembangan yang besar karena interaksi antara peneliti TCP/IP, peneliti *xerox-network-system* (XNS) dan pemakai yang bekerja di fasilitas Xerox di Palo Alto menghasilkan XNS's RIP *(routing-information-protocol)* yang kemudian digunakan pada Unix BSD. Karena pada saat itu Unix BSD sedang populer maka RIP digunakan secara luas pada jaringan TCP/IP, akibat banyaknya kalangan akademis, fasilitas penelitian, dll yang menggunakan Unix BSD. Saat ini Internet yang menggantikan ARPAnet tidak hanya terdiri dari satu jaringan tetapi merupakan komglomesari dari berbagai macam jaringan, tetapi protokol yang paling banyak dipakai adalah TCP/IP, walaupun ada beberapa jaringan menggunakan protokol berbeda, milanya BITNET dan CREN yang menggunakan protokol IBM's SNA.

2.4.1 TCP/IP Layer Model

Dalam menanggapi model layering dari ISO maka TCP/IP membuat model layering dengan empat tingkatan. Konsep layering menurut TCP/IP dibagi dalam empat tingkatan, yaitu :

.1 Lapisan Aplikasi

Pada level tertinggi ini user menggunakan aplikasi program untuk mengirim atau menerima data lewat internet.

.2 Lapisan Transport

Tugas utama dari lapisan ini adalah membuat komunikasi dari suatu aplikasi program yang lain pada alamat tujuan. Hal ini biasa disebut *end-to-end.*

Selain mengatur aliran informasi, lapisan ini akan membuat reliable transport, sehingga ada kepastian akan data yang dikirimkan, adapun tiap data akan disebut sebagai paket.

.3 Lapisan Internet

Lapisan ini bertugas untuk mengirim data dari suatu mesin ke alamat yang dituju, adapun paket dari transport layer akan diberi tambahan informasi sehingga paket ini akan disebut datagram. Lapisan ini juga bertugas untuk menerima datagram, mengeceknya dan meneruskannya sampai ke tingkat aplikasi.

.4 Lapisan Network Interface

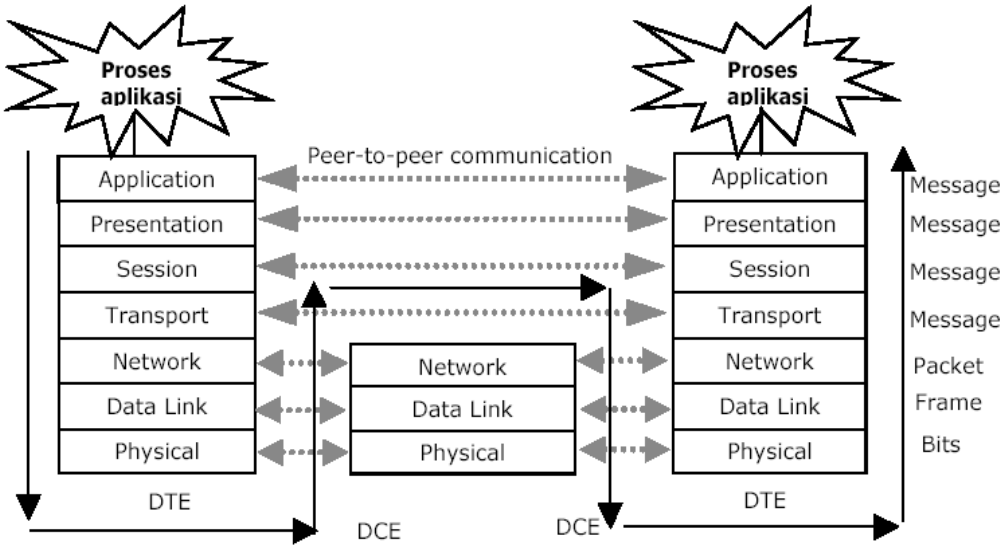
Lapisan ini bertugas untuk merespon datagram dari internet layer dan mengirimkannya melalui jaringan.

Network interface ini berisi driver yang digunakan sebagai protkol pada pengiriman data. Data pada level ini disebut frame yaitu datagram yang diberi tambahan informasi.

2.5 Model OSI

Pada tahun 1978 *international-organization-of-standards* (ISO) menyusun standart OSI yang dapat dipakai untuk mengembangkan sistem terbuka dan

sebagai referensi dengan sistem komunikasi data yang berbeda. Sistem jaringan yang dirancang menurut kerangka kerja dan spesifikasi OSI maka akan memiliki metode komunikasi yang saling kompatibel.



Gambar 2.11 ***Model Referensi OSI***

Model OSI memili 7 layer yang bekerja dari layer teratas menuju kebawah bawah sesuai urutan : aplication, presentation, session, transport, network, data-link, physical. Ketuju layer tersebut disusun berdasarkan lima prinsip yang harus diikuti untuk menentukan layer dalam komunikasi, yaitu :

- Layer dibuat jika ketika diperlukan pemisahan level yang secara teori diperlukan.
- Masing-masing layer memiliki fungsi yang jelas.
- Setiap fungsi dari masing-masing layer telah ditentukan agar sesuai dengan standart protokol secara internasional.

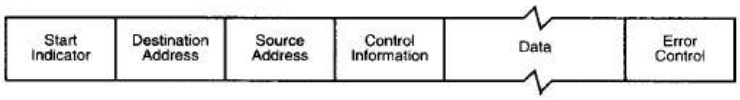
- Batas kedua layer telah ditentukan untuk mengurangi informasi menerobos antarmuka layer.
- Setiap layer ditentukan dengan jelas fungsinya, tetapi jumlah layer sebaiknya sekecil mungkin untuk menghindari arsitektur yang luas.

2.5.1 Physical Layer

Physical layer mendefinisikan karakteristik *mekanik*, *elektrik*, *fungsional*, dan *prosedural* untuk *mengaktifkan*, *mempertahankan/memelihara* (maintain), serta *memutuskan* (deactivate) koneksi untuk mentransmisikan deretan bit melalui suatu saluran fisik.

2.5.2 Data Link Layer

Data link layer memiliki fungsi untuk mewujudkan suatu transfer data yang andal melalui saluran fisik. Layer ini memetakan unit data yang bersal dari *network layer* menjadi frame data yang dapat ditransmisikan. Untuk melaksanakan fungsi-fungsi di atas, *data link layer* menyediakan proses sinkronisasi, *error control* dan *flow control*.



Gambar 2.12 ***Bentuk Frame Data-link-layer***

2.5.3 Network Layer

Fungsi *yang* dijalankan oleh *network layer* adalah menentukan alamat host dan alamat lingkungan tempat host berada. Selain alamat host dikenal juga dengan alamat hardware.

2.5.4 Transport Layer

Transport layer melaksanakan pengendalian end-to-end (station-to-station) terhadap data yang ditransmisikan serta melakukan optimasi terhadap penggunaan sumber daya jaringan.

2.5.5 Session Layer

Session adalah suatu koneksi antara dua station yang memungkinkan mereka berkomunikasi. Misalnya, untuk melaksanakan proses transfer file dengan setiap station, suatu prosesor utama melakukan beberapa *session* yang berlangsung secara bersamaan dengan terminal-terminal *remote* yang terhubung dengannya.

2.5.6 Presentation Layer

Presentation layer bertugas memberikan informasi cara mengatasi perbedaan *syntax* kepada entitas aplikasi-aplikasi yang sedang berkomunikasi. Untuk melaksanakan fungsi ini, layer 6 melakukan proses transformasi data (kompresi dan enkripsi), pembentukan (format) data, serta pemilihan *syntax*.

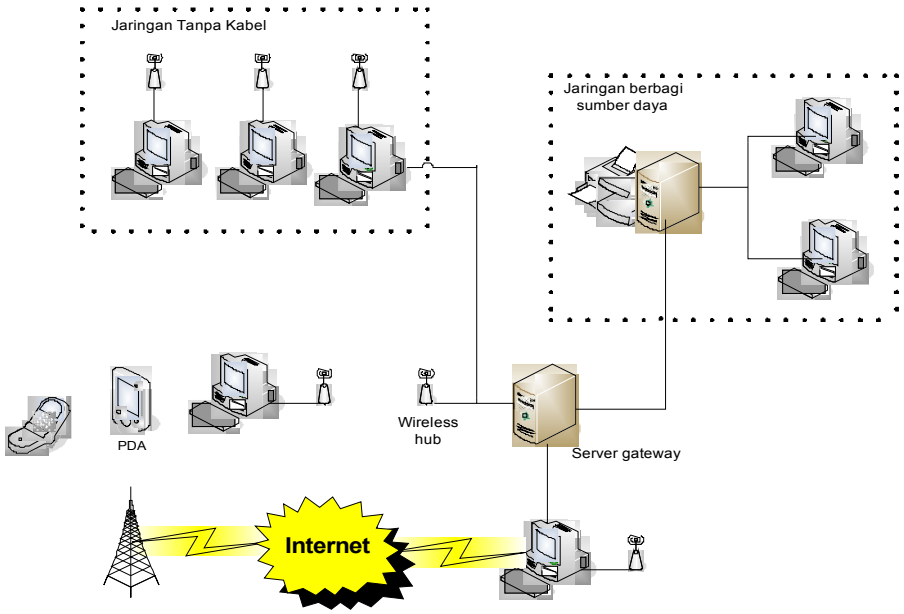
2.5.7 Application Layer

Application layer memungkinkan suatu proses aplikasi mengakses lingkungan OSI. Layer ini dapat dianalogikan sebagai ”jalan” yang menghubungkan proses proses aplikasi yang menggunakan OSI untuk saling mempertukarkan informasi. Semua layanan yang disediakan dapat diakses secara langsung oleh proses aplikasi.

2.6 Internet

Internet[7] merupakan sarana yang sangat efisien dan efektif untuk melakukan pertukaran informasi jarak jauh, maupun dalam lingkungan *internal* suatu organisasi. Adapun *internet* menyediakan fasilitas *HTTP* [8]yaitu fasilitas yang bekerja di atas TCP (Transmission Control Protocol) dimana dapat menjamin sampainya data di tujuan dalam urutan yang benar. Bila suatu kesalahan terjadi selama proses pengiriman, pihak pengirim akan mendapat pemberitahuan bahwa telah terjadi ketidakberesan. Setiap halaman *web* yang dikirim akan melibatkan satu proses penyambungan antara *client* dan *server*, baru kemudian datanya ditransfer. Setelah data selesai ditransfer, koneksi antara *server* dan *client* akan diputus.

[7] Onno W. Purbo, TCP/IP - Standar, Desain Dan Implementasi, (Jakarta : PT Elex Media Komputindo,1998), hlm. 1-18.
[8] Ibid, hlm. 356-378



Gambar 2.13 **Jaringan Internet**

2.7 Macromedia Flash MX Profesional 2004

Flash MX Professional 2004 di rancang untuk memudahkan perancang *web* dan pembuat aplikasi. Flash MX Professional 2004 terdiri dari semua *features* yang terdapat pada Flash MX 2004, ditambah dengan beberapa *tools* yang sangat membantu. Ini dapat membantu *project-management tools* dalam mengoptimalisasikan alur kerja antara anggota-anggota dari sebuah team *web* yang di dalamnya terdapat para *designer* dan *developer*.

Macromedia Flash merupakan program aplikasi *authoring tool* profesional yang digunakan untuk membuat animasi *vektor* dan *bitmap* untuk pengembangan situs *web* yang interaktif dan dinamis. Selain itu aplikasi ini juga dapat digunakan untuk membuat animasi logo, *movie*, *game*, pembuatan navigasi pada situs *web*,

banner, tombol animasi, menu interaktif, *form* interaktif, *e-card*, apikasi interaktif, *screen saver* dan pembuatan keseluruhan isi *situs web* atau pembuatan aplikasi-aplikasi *web* lainnya.

2.7.1 Terminologi-Terminologi Macromedia Flash MX Profesional

Di dalam pembuatan aplikasi menggunakan Flash MX Profesional, terdapat beberapa istilah dan terminologi-terminologi yang digunakan.

1. Artwork

Artwork merupakan objek yang digunakan oleh Flash MX Profesional dalam membuat aplikasi. *Artwork* dapat berupa objek *vektor, image bitmap*, objek teks, objek *video*, *file* suara dan objek-objek lainnya yang didukung oleh Flash MX Profesional. *Artwork* dapat dibuat dengan menggunakan Flash MX Profesional, atau di-*import* dari aplikasi lain. Atribut-atribut *artwork* tersebut dapat dimodifikasi dengan *tool-tool* yang tersedia dalam Flash MX Profesional.

2. Simbol

Simbol adalah sebuah elemen seperti objek grafik, objek tombol, klip video, *file* suara atau *font*, yang digunakan berulang kali dalam sebuah dokumen Flash MX Profesional. Simbol yang dibuat akan diletakkan ke dalam file *library*. Penggunaan simbol sangat berguna untuk pembuatan setiap elemen yang tampil berulang kali dalam sebuah dokumen.

3. Instance

Instance adalah sebuah kembaran simbol yang diletakkan pada *stage* (*form area*). Properti-properti dari sebuah *instance* dapat dimodifikasi tanpa

mempengaruhi simbol utama. Sedangkan apabila sebuah simbol utama diedit, maka dapat mengubah seluruh *instance* yang telah ada.

4. Komponen

Komponen adalah klip-klip *movie* pendek dengan parameter-parameter yang telah didefinisikan untuk membantu dalam pembuatan dan pengembangan *movie* dan aplikasi Flash yang kaya dan interaktif. Komponen dapat digunakan untuk berbagai fungsi, termasuk pembuatan elemen-elemen antarmuka, interaksi antara *client* dan *server,* serta objek-objek *audio-video*.

5. Aset

Aset *adalah* berbagai macam elemen yang digunakan untuk membuat sebuah *movie*. Yang termasuk kategori aset adalah semua objek yang ada pada *stage* dan simbol, *instance*, klip suara, dan file-file yang dapat di-*impor* lainnya.

6. Animasi

Animasi adalah sebuah objek atau beberapa objek yang tampil bergerak melintasi stage atau berubah bentuk, berubah ukuran, berubah warna, berubah keburaman, berubah putaran, dan berubah properti-properti lainnya.

7. Movie

Movie adalah serangkaian animasi yang dibuat berdasarkan suatu alur cerita. Di dalam sebuah *movie* terdiri atas beberapa *scene* yang memiliki sebuah *timeline*. Sedangkan sebuah *scene* terdiri atas beberapa *frame*. *Movie* dapat terdiri atas grafik *vektor, bitmap, teks, animasi, video*, suara, dan aplikasi-aplikasi untuk situs.

8. Movie Interaktif

Movie interaktif adalah *movie-movie* nonlinear yang dapat berinteraksi dengan pemirsa melalui pengaturan interaktifitas, baik menggunakan mouse atau keyboard, untuk beralih ke bagian-bagian yang berbeda dari sebuah *movie*, atau berinteraksi dengan aplikasi-aplikasi lainnya. Interaktif *movie* dibuat dengan menambahkan baris- baris perintah *ActionScript*, sehingga pengguna dapat menggerakkan objek-objek secara langsung, memasukkan data atau informasi kedalam *form* isian, dan melakukan operasi-operasi lainnya.

2.7.2 Macromedia Flash Communication Server MX

Macromedia *Flash Communication Server MX*, adalah suatu *platform* baru yang digunakan untuk komunikasi pribadi yang interaktif, dan terintegrasi dengan *Macromedia Flash MX* sehingga menghasilkan suatu multimedia yang diantaranya *live audio*, *video* dan data pada suatu jaringan, juga menyediakan sebuah sistem guna menyimpan data rekaman *audio, video,* dan data objek bersama, juga memungkinkan pembagian pengiriman data yang sinkron ke berbagai *client* secara *real-time*. Teknologi ini menyatukan aplikasi dan komunikasi, menyediakan *audio, video*, dan data dengan Macromedia Flash Player.

Flash Communication Server MX menyertakan sejumlah *feature* yang mendukung dan memungkinkan terjadinya suatu generasi baru dalam dunia komunikasi. Secara khusus dapat kita lihat dibawah ini :

- Menyediakan suatu fasilitas yang efisien dan berkecepatan tinggi dalam pengeksekusian kode, *content* dan komunikasi.

- Menintegrasikan *content*, komunikasi, dan tampilan antar muka kepada masyarakat luas.
- Menyediakan model objek yang berguna dan dapat diperluas untuk berinteraksi.
- Memungkinkan terjadinya pengembangan aplikasi yang cepat dengan adanya komponen-komponen pada Flash Communication *Server* MX.
- Memungkinkan penggunaan *web* dan data *services* yang disediakan oleh aplikasi *server*.
- Dapat menghubungkan dan memutuskan koneksi antar pengguna.
- Memungkinkan penyebaran dan pengembangan yang mudah pada berbagai *platform*.

2.7.3 Flash Player

Flash Player merupakan sebuah aplikasi yang digunakan untuk menjalankan aplikasi atau *file-file* yang dibuat oleh aplikasi Flash. Dengan menggunakan flash player, aplikasi yang dibuat dengan flash dapat dijalankan dengan baik dan menghasilkan output yang tepat.

2.7.4 Action Script

ActionScript adalah bahasa pemrograman yang digunakan flash untuk mengontrol objek, membuat navigasi, dan elemen interaktif lainnya. Pada Flash MX Profesional, pengembangan *ActionScript* telah sampai pada dukungannya terhadap *object oriented programming*, selain itu featur-featur baru juga telah

ditambahkan kedalam *ActionScript*, sehingga mampu untuk menangani aplikasi dinamis yang rumit sekalipun.

2.7.5 Flash Remoting

Flash Remoting MX adalah aplikasi *server* yang menjembatani antara jaringan komunikasi dengan aplikasi Flash dan *remote services*. *Remote services* terdiri dari beberapa aplikasi teknologi *server*, seperti JavaBean, Macromedia ColdFusion *component or page*, ASP.NET *page*, atau *web* service. *Service functions* memberikan fasilitas untuk mengubah *ActionScript* ke dalam aplikasi Flash *movie*. Ketika kita membandingkan ke teknik-teknik yang lain untuk menghubungkan aplikasi Flash ke penyedia data *eksternal*, seperti fungsi-fungsi HTTP misalnya mengambil URL dan load *Variables* dan fungsi-fungsi XML lainnya seperti XML SOCKET, Flash Remoting MX menyediakan keuntungan berikut :

- ***Ease Of Use***

  Flash Remoting MX menawarkan konversi jenis data otomatis dari kode *remote* asli, seperti Java, CFML, dan C#, ke *Actionscript* dan begitu pula sebaliknya. Flash Remoting MX juga secara otomatis melaksanakan *logging*, *debugging*, dan pengintegrasian keamanan.

- ***Performance***

  Flash Remoting MX juga berguna sebagai media penyampaian pesan antara aplikasi Flash dan jasa *remote* menggunakan *Action Message*

Format (AMF). AMF adalah suatu *format biner* yang terdapat pada *Simple Object Access* Protocol (SOAP) *format*.

- ***Extensibility***

  Flash Remoting MX dirancang sedemikian rupa untuk mengintegrasikan ke dalam suatu bentuk pola desain aplikasi guna membangun suatu aplikasi Flash yang baik

2.8 XML

XML[9] kependekan dari *eXtensible Markup Language*, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C (*World Wide Web Consortium*) pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan *markup language* yang tidak kalah hebatnya dengan SGML.

Seperti halnya HTML, XML juga menggunakan *elemen* yang ditandai dengan *tag* pembuka (diawali dengan '<' dan diakhiri dengan '>'), *tag* penutup (diawali dengan '</ 'diakhiri '>') dan atribut elemen (parameter yang dinyatakan

[9] Henky Priatna,kiat praktis menjadi webmaster profesional (Jakarta : PT Elex Media Komputindo,2004), hal, 123-129

dalam *tag* pembuka misal <form name=”isidata”>). Hanya bedanya, HTML medefinisikan dari awal *tag* dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan *tag* dan atribut sesuai kehendak kita.

Perbedaan utama XML dengan HTML adalah, XML didesain untuk mendeskripsikan data dan berfokus pada data yang dibawa, sedangkan HTML didesain untuk memunculkan data dan berfokus pada bagaimana tampilan dari data. Sehingga XML lebih berorientasi pada bagaimana mendeskripsikan informasi, sedangkan HTML berorientasi pada bagaimana menampilkan informasi.

Contoh dokumen XML :

<pesan>

<dari>MIS Manager</dari>

<buat>HRD Manager</buat>

<buat>Bagian rekrut</buat>

<buat>Computer Suport team</buat>

<subyek>Permohonan Tenaga kerja baru</subyek>

<isi>Mohon diberikan tenaga kerja baru untuk mengisi lowongan di Departemen MIS</isi>

</pesan>

pada contoh diatas <pesan>, <dari> <buat>,dan <isi> bukanlah *tag* standard yang telah di tetapkan dalam XML. *Tag-tag* itu dapat dibuat sendiri

sesuai keinginan. Sampai di sini XML tidak melakukan apapun. Yang ada hanyalah informasi yang dikemas dengan *tag-tag* XML. Dibutuhkan *software* lain untuk untuk mengirim, menerima atau menampilkan informasi di dalamnya. Banyak pakar berpendapat bahwa XML adalah standar *format* data masa depan, yang bisa mendukung dan menyatukan semua *sistem* operasi komputer.

Beberapa keunggulan XML antara lain :

1. XML bersifat *free.*
2. XML bisa digunakan dalam berbagai *platform* (*multi platform*) atau *sistem* operasi.
3. XML mudah dimengerti, karena *tag-tag*nya bisa disesuaikan dengan bahasa sendiri.
4. XML dapat dibuat dengan *editor* teks biasa seperti Notepad.

2.9 Dreamweaver MX

Macromedia Dreamweaver merupakan *software* yang dikenal sebagai *software web authoring tool*, yaitu *software* untuk desain dan *layout* halaman *web*. Dreamweaver MX merupakan versi terbaru Dreamweaver yang memiliki kemampuan bukan hanya sebagai *software* untuk *desain web* saja tetapi juga untuk editing kode serta pembuatan aplikasi *web* dengan menggunakan berbagai bahasa pemrograman *web*, antara lain JSP, PHP, ASP, dan ColdFusion.

2.10 PHP: Hypertext Preprocessor

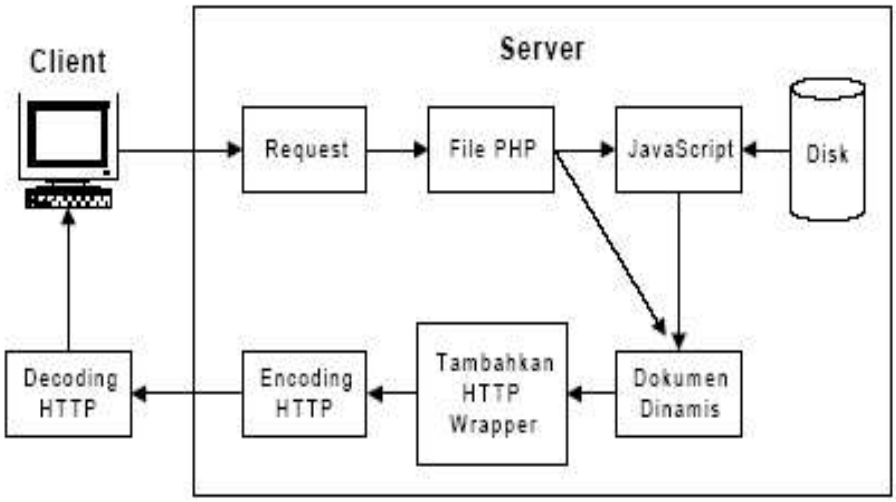
PHP[10] (PHP : *Hypertext Preprocessor*) adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi *server*. Sebagian besar perintahnya berasal dari bahasa C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi *web* menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan *server*. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI (*Common Gateway Interface*).

Dengan PHP, *browser web* mengacu secara langsung ke *file* yang dituju, yang lalu dibaca oleh *server* sebagaimana *file* HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser web*, *server web* memeriksa isi *file* dan menentukan apakah ada kode di dalam *file* tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya dimasukkan ke dalam dokumen yang sama. *Server web* bekerja secara langsung terhadap file yang bersangkutan dan seluruh kode dieksekusi di *server*. PHP memiliki tiga kegunaan utama diantaranya adalah :

1. *Server-side scripting*

   Seperti telah dijelaskan di atas, ini merupakan kegunaan utama dari PHP. Beberapa komponen yang dibutuhkannya antara lain adalah *parser* PHP (baik berupa CGI maupun modul *server*), sebuah *server web*, dan sebuah *browser web*.

[10] http://www.php.net/docs.php.



Gambar 2.14 **Prinsip Kerja PHP sebagai *Server-side scripting*.**

2. *Command line scripitng*

   Script PHP dapat dijalankan tanpa adanya *server* dan *browser*, hanya dibutuhkan *parser* PHP. Biasanya PHP digunakan untuk *script* yang secara teratur diekseskusi menggunakan *cron* (pada Unix/Linux) atau *Task Scheduler* (pada Windows). Selain itu juga dapat digunakan dalam pekerjaan pemrosean teks sederhana.

3. *Client-side* GUI (*Graphical User Interface*) *applications*

   PHP dapat digunakan untuk membuat aplikasi GUI pada sisi *client* dengan memanfatkan PHP-GTK. Dengan PHP-GTK dapat dikembangkan aplikasi *cross-platform* yang memiliki *featur-featur* seperti halnya PHP yang dijalankan pada sisi *server*.

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu:

a. *Script* (kode program) terintegrasi dengan *file* HTML.

b. Tidak ada proses *compiling* dan *linking*.

c. Berorientasi obyek (*object oriented*).

d. Sintaksis pemrogramannya mudah dipelajari, sangat menyerupai C dan Perl.

e. *Integrasi* yang sangat luas ke berbagai *server database*. Menulis *web* yang terhubung ke *database* menjadi sangat sederhana. *Database* yang didukung oleh PHP: Oracle, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, Informix, dBase, UNIX dbm.

f. PHP tidak terbatas untuk hanya menghasilkan keluaran HTML, PHP juga bisa digunakan untuk menghasilkan *images* (mengunakan GDLib dan JpGraph), animasi Flash (menggunakan libswf dan Ming) atau bahkan *file* PDF yang dinamis (menggunakan pdfLib, PHP PDF).

2.11 JavaScript

Javascript[11] adalah *client side programming language*, yang berfungsi untuk menjalankan perintah-perintah yang bisa diproses di komputer *client* sehinngga mengurangi beban pada *server* dan jaringan, sehingga akan lebih cepat bagi *client* dalam pemrosesan data. Tetapi penggunaan javascript secara berlebihan atau tidak optimal akan membuat waktu *download* menjadi lama. Berbeda dengan *server side*, *script* pada *client side* dikirim beserta *file* html lalu diproses oleh *web browser* pada komputer *client*. Agar *browser* dapat mengenali

[11] Henky Priatna,kiat praktis menjadi webmaster profesional (Jakarta : PT Elex Media Komputindo,2004),hal, 147-148.

dan memproses javascript maka diperlukan modul tambahan, yaitu JVM (*Java Virtual Machine*).

2.12 Real-Time Messaging Protocol (RTMP)

Real Time Messaging Protocol adalah protokol komunikasi yang dikembangkan sebagai koneksi *real time messaging* dua arah antara *clients* dan *Server*, yang membantu dalam pengiriman data, *audio, visual* ke dan dari klien. Para pengguna lainnya pun dapat menghubungkan kepada *server* yang sama sehingga dapat menerima pesan, mendapatkan data yang terbaru, *audio*, maupun *visual.*

2.13 Apache

Apache[12] merupakan *web server* yang paling banyak dipergunakan di *Internet*, berdasarkan survei yang dilakukan oleh Netcraft pada bulan Januari 2004, yaitu sebesar 67,38%. Disana Program ini pertama kali didesain untuk *sistem* operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya Apache mengeluarkan programnya yang dapat dijalankan di Windows

Saat ini Apache dipergunakan secara luas, hal ini disebabkan karena programnya yang gratis dengan kinerja relatif stabil. Dalam pengembangannya pun mempergunakan *sistem open source*, yakni tiap orang dibuka kesempatan seluas-luasnya untuk dapat memberikan kontribusi dalam mengembangkan program.

[12] http://www.apache.org.

Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya. Beberapa dukungan Apache :

1. Kontrol Akses

   Kontrol ini dapat dijalankan berdasarkan nama *host* atau nomor IP.

2. CGI (*Common Gateway Interface*), yang paling terkenal untuk digunakan adalah perl (*Practical Extraction and Report Language*), didukung oleh Apache dengan menempatkannya sebagai modul (mod_perl).

3. PHP (*PHP Hypertext Processor*), program dengan metode semacam CGI, yang memproses teks dan bekerja di *server*. Apache mendukung PHP dengan menempatkannya sebagai salah satu modulnya (mod_php). Hal ini membuat kinerja PHP menjadi lebih baik.

2.14 UML (UNIFIED MODELLING LANGUAGE)

UML[13] adalah sebuah bahasa untuk menentukan *visual*isasi, konstruksi, dan mendokumentasikan *artifacts* dari *sistem* software untuk memodelkan bisnis, dan *sistem software* lainnya.

Dengan UML, pemodelan dari suatu *sistem* dapat lebih mudah di mengerti karena pemodelannya berorientasi objek (*object oriented*). Model yang diberikan juga siap dipakai untuk memberikan arah ke pengkodean suatu *sistem*, dan bukan dimaksudkan untuk menjadi suatu bahasa pemrograman *visual*. Bahasa pemodelannya bebas dari berbagai bahasa pemrograman dan proses rekayasa,

[13] The UML User Guide, Booch-Jacobson-Rumbaugh.hal 468

sehingga dapat menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, *sistem* operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET, dan ActionScript. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural seperti dalam Visual Basic atau C.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan *sistem* yang besar dan kompleks. Metoda penelitian pengembangan *sistem* ini memiliki tahapan-tahapan sebagai berikut :

1. **Bussiness Process (Planning)**

   Tahapan pertama ini ditujukan untuk memperoleh data mengenai permasalahan yang dihadapi, *sistem* lama yang digunakan dan *sistem* baru yang dirancang. Selain itu juga untuk menentukan proses dan aktivitas apa yang mungkin terjadi pada saat perancangan *sistem*.

2. **Use Case Diagram**

   Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah *sistem*. Yang ditekankan adalah “apa” yang diperbuat *sistem*, dan bukan “bagaimana”. *Use case* dapat merepresentasikan sebuah interaksi antara aktor dengan *sistem*. *Use case diagram* sangat membantu dalam

menyusun *requirement* sebuah *sistem*, dalam mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada *sistem*.

3. Class Diagram

Class akan menggambarkan keadaan (*atribut/properti*) dari *sistem*, sekaligus menawarkan layanan untuk memanipulasi keadaan *sistem* tersebut (metoda/fungsi).*Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain pada *sistem* yang akan dibangun.

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

4. Statechart Diagram

Statechart *diagram* akan menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada *sistem* sebagai akibat dari *stimuli* yang diterima.

5. Activity Diagram

Activity diagrams menggambarkan berbagai alur aktivitas dalam *sistem* yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana *sistem* berakhir. *Activity diagram* juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

6. Sequence Diagram

Sequence *diagram* akan menggambarkan interaksi antar objek di dalam dan di sekitar *sistem* (termasuk pengguna, *display*, dan sebagainya). *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output.* Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

7. Collaboration Diagram

Collaboration *diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

8. Component Diagram

Component *diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*).

Komponen piranti lunak ini adalah modul berisi *code (scripting code)*, PHP dan *Actionscript* versi 2. Komponen-komponen ini terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

9. Deployment/physical diagram

Deployment*/physical diagram* akan menggambarkan detail dari komponen yang di-*deploy* dalam infrastruktur *sistem*, di mana komponen akan terletak (pada mesin, *server* atau piranti keras yang lain), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.