

BAB II

DASAR TEORI

2.1 Gambaran Umum Merpati Nusantara Airlines

Merpati Nusantara *Airlines* merupakan sebuah Perseroan Terbatas (PT) yang awalnya merupakan anak perusahaan PT Garuda Indonesia.

2.1.1 Sejarah Berdirinya Perusahaan ¹

PT Merpati Nusantara *Airlines* (PT MNA) didirikan berdasarkan PP No.19 tahun 1962 tanggal 2 September 1962 dalam bentuk Perusahaan Negara (PN) dengan lapangan usaha, meliputi penyelenggaraan perhubungan udara di daerah dan berhubungan serba guna serta memajukan segala sesuatu yang berkaitan dengan angkutan udara.

Berdasarkan PP No.70 tahun 1971, pada tahun tersebut dialihkan bentuk menjadi persero. Selanjutnya berdasarkan Peraturan Pemerintah No.30 tahun 1978 tentang pengalihan pengusahaan Modal Negara Republik Indonesia, dalam perusahaan Perseroan (Persero) PT. MNA kepada PT. Garuda Indonesia.

Dalam status sebagai anak perusahaan PT. Garuda Indonesia mengalihkan wilayah operasi secara bertahap meliputi pengalihan jalur penerbangan, SDM dan pengalihan asset yang ditinggalkan PT. Garuda Indonesia *Airways*. Dengan adanya pengalihan ini PT. MNA maju pesat, ini terlihat dengan meningkatnya jumlah asset dan jumlah pendapatan yang diterima perusahaan.

¹ Skripsi Hariani Sartika (001100056) STIE Pariwisata API Program studi Manajemenr “Analisi Sikap Konsumen Terhadap Pelayanan Jasa Penerbangan PT Merpati Nusntara *Airlines* di Jogjakarta (41-42)

Dilain pihak akibat kebijaksanaan tersebut mengakibatkan adanya kendala yang dihadapi PT. MNA sebagai pihak yang ditentukan terhadap PT. Garuda Indonesia. PT. Garuda Indonesia merasa sebagai pihak yang menentukan, terhadap PT. MNA, sehingga PT. MNA tidak dapat menjalankan operasi perusahaan secara efisien dan efektif karena adanya campur tangan dari induk perusahaan yaitu PT. Garuda Indonesia.

Memperhatikan dan mempertimbangkan kondisi diatas, maka pada tanggal 7 Agustus 1990 dicapai kata sepakat dan ditandatangani keputusan bersama Menteri Perhubungan dan Menteri Keuangan mengenai pemisahan perusahaan PT. MNA dari anak perusahaan PT. Garuda Indonesia menjadi BUMN yang berdiri sendiri.

2.1.2 Daerah pemasaran²

Daerah pemasaran yaitu berupa jalur penerbangan yang dilayani PT. Merpati Nusantara Airlines adalah menghubungkan antar daerah di Indonesia dan penerbangan perintis.

Kota-kota yang masih dilayani oleh jalur penerbangan PT. Merpati Nusantara Airlines hingga saat ini terlihat pada tabel 2.1 berikut :

Tabel 2.1 Daftar kota yang dilayani PT. Merpati Nusantara Airlines

Pulau	Kota
Sumatra	Bandar Lampung, Bengkulu, Medan, Padang, Pekanbaru
Kepulauan Riau	Batam, Pangkal Pinang, Tanjung Pinang, Tanjung Pandan
Kalimantan	Balikpapan, Pangkalanbun, Pontianak
Sulawesi	Manado, Palu, Makassar
Irian Jaya	Biak, Fak-fak, Jayapura, Kaimana, Merauke, Nabire, Serui, Sorong, Timika, Wamena
Kepulauan Ambon	Ambon

² Time Table PT. Merpati Nusantara Airlines, 1998

NTT	Alor, Bajawa, Enden, Kupang, Labuhanbajo, Larantuka, Maumere, Ruteng, Tambulaka, Waingapu
NTB	Bima, Mataram
Jawa	Bandung, Jakarta, Semarang, Surabaya, Jogjakarta
Negara	Kota
Malaysia	Kuala Lumpur
Timur-timor	Dili

2.1.3 Sekilas tentang pemesanan tiket

Dalam memudahkan konsumen untuk mendapatkan tiket, penumpang dapat langsung memesan tiket yang dapat dilayani di bandara Adi Sucipto – Jogjakarta atau di kantor cabang yang terletak di Jl. Diponegoro no. 31 Jogjakarta yang merupakan kantor pusat cabang Jogjakarta ataupun di agen-agen yang terdaftar resmi. Perusahaan penerbangan PT Merpati Nusantara *Airlines* di Jogjakarta melengkapi informasi tentang perusahaan dengan membuka situs *internet* dengan alamat www.merpati.co.id guna memberikan informasi kepada konsumen dan melayani pemesanan tiket.

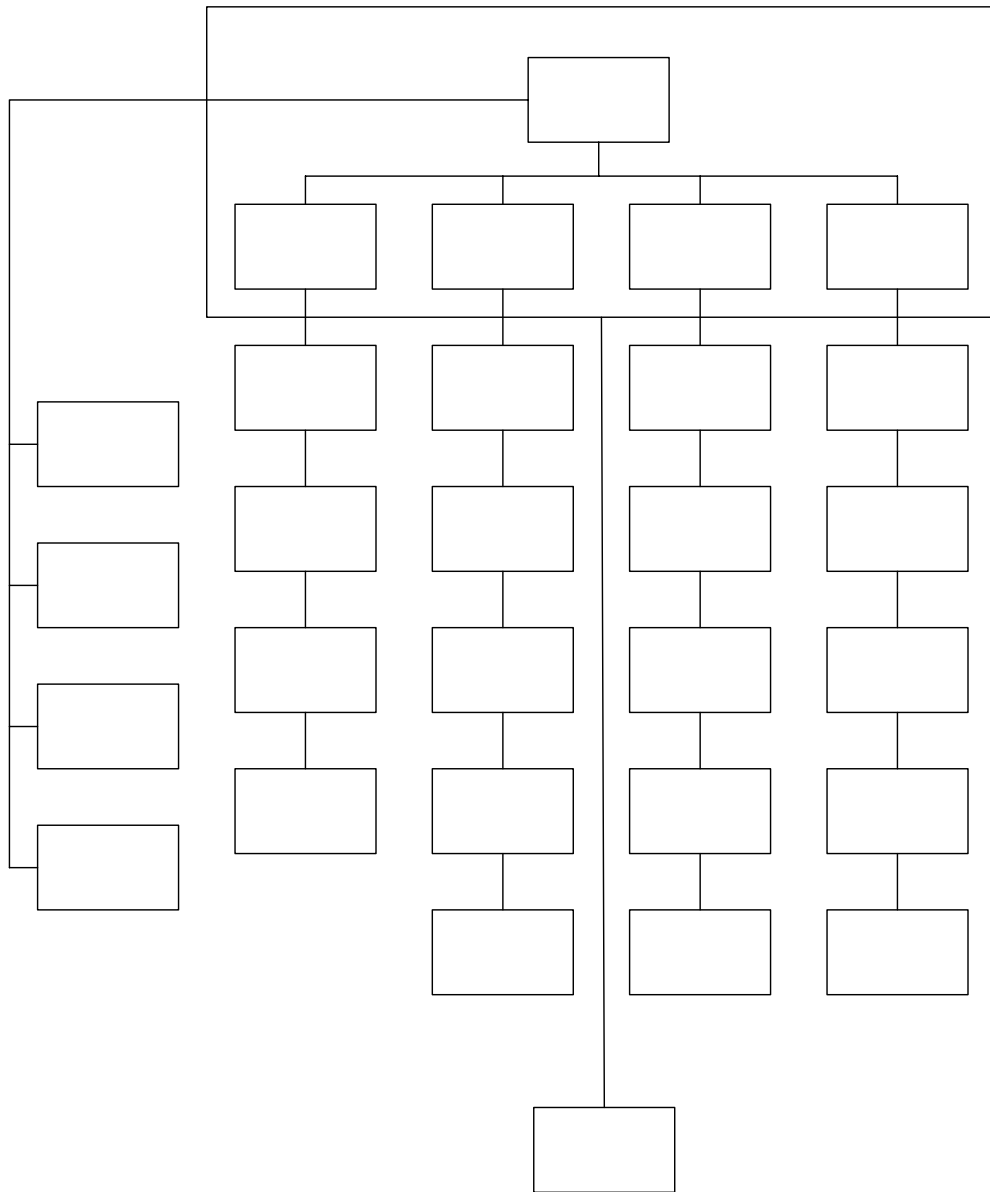
PT Merpati Nusantara *Airlines* di Jogjakarta menyediakan rute penerbangan langsung dari Jogjakarta, dengan beberapa jurusan yaitu Jogjakarta-Surabaya, Jogjakarta-Balikpapan, Jogjakarta-Tarakan dan untuk ke kota lain penumpang akan ditransit dikota yang telah menyediakan jurusan yang akan dituju. Perusahaan penerbangan PT Merpati Nusantara *Airlines* juga membagi dua kelas penerbangan yaitu premium (lebih muran 10% dari kelas eksekutif) dan kelas ekonomi (lebih murah 30% dari kelas eksekutif).

Pelayanan yang diberikan oleh Merpati Nusantara *Airlines* dapat meliputi pelayanan pemesanan tiket yang memberikan kesempatan kepada calon

penumpang atau konsumen dalam memesan tiket dan pembelian tiket maupun konfirmasi tempat duduk. Untuk konfirmasi tempat duduk dapat dilakukan 1(satu)bulan sebelumnya, kemudian untuk pemesanan tiket dapat dilakukan 3 hari sebelum hari H. Sedangkan pelayanan informasi diberikan kepada calon konsumen atau penumpang yang ingin mengetahui jadwal keberangkatan dan kedatangan pesawat serta harga tiket pesawat.

2.1.4 Struktur Organisasi

Bentuk struktur organisasi pada PT. Merpati Nusantara Airlines adalah sebagai berikut :



2.2 TCP/IP

Dalam arti yang sederhana, TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah nama keluarga protokol jaringan. Protokol adalah sekelompok aturan yang harus diikuti oleh perusahaan-perusahaan dan produk-produk *software* agar produk mereka itu bisa kompatibel satu dengan yang lainnya. Suatu protokol menentukan bagaimana suatu *software* berkomunikasi dengan *software* lainnya, juga menentukan bagaimana setiap bagian dari keseluruhan paket menentukan bagaimana perjalanan informasinya.

Internet, memang dari awal sudah dibuat dengan menggunakan TCP/IP pada tingkat protokolnya yang memungkinkan sistem apapun yang terhubung kedalamnya bisa berkomunikasi dengan sistem lain tanpa mempedulikan bagaimana sistem masing-masing bekerja.

TCP/IP sebenarnya adalah dua macam protokol berbeda. Tidak seperti di anggap kebanyakan orang, istilah TCP/IP mengacu ke seluruh keluarga protokol yang dirancang untuk mentransfer informasi sepanjang jaringan. TCP/IP dirancang untuk menjadi komponen perangkat lunak dari suatu jaringan. Semua bagian didalam keluarga TCP/IP memiliki tugas tersendiri, misalnya mengirim e-mail, mentransfer *file*, menyediakan layanan *login* jarak jauh (*remote login*) dan menangani informasi *routing* jaringan.

Protokol TCP bertanggung jawab memecah informasi kedalam beberapa paket, sedang IP bertanggung jawab mengangkut (mentransfer) paket-paket tersebut sesuai tujuan. Kemudian TCP bertugas menyatukan kembali paket-paket tersebut ke urutan yang benar.

Layanan dalam TCP/IP yang berbeda dikelompokkan menurut fungsinya. Protocol-protokol transport mengendalikan pergerakan data antara dua mesin, mencakup :

1. TCP (*Transmission Control Protocol*)

Protokol ini bersifat *connection-based*, artinya kedua mesin pengirim dan penerima tersambung dan berkomunikasi antara satu sama lain sepanjang waktu.

2. UDP (*User Datagram Protocol*)

Protokol ini bersifat *connectionless* (tanpa koneksi), artinya data dikirim tanpa mesin penerima dan pengirim saling berhubungan. Ini seperti mengirim surat lewat kantor pos, surat dikirim oleh si pengirim namun ia tidak pernah tahu apakah surat itu sampai di tujuan atau tidak.

Sementara itu adapula protokol-protokol *routing* untuk menangani pengalamatan (*addressing*) data dan menentukan jalur terbaik untuk mencapai tujuan. Protokol-protokol tersebut juga bertanggung jawab memecah informasi-informasi ukuran besar dan menyusunnya kembali pada tujuan. Protokol-protokol tersebut antara lain :

1. IP (*Internet Protocol*) menangani transmisi data yang sebenarnya.
2. ICMP (*Internet Control Message Control Protocol*) menangani informasi status untuk IP, seperti *error* (kesalahan) dan perubahan-perubahan dalam perangkat keras jaringan yang mempengaruhi *routing* (penentuan jalur).

3. RIP (*Routing Information Protocol*) dan OSPF (*Open Shortest-Path First*), yaitu satu dari berbagai protokol yang menentukan metode *routing* terbaik untuk menyampaikan data.

2.3 World Wide Web (WWW)

2.3.1 Pengenalan WWW

Menurut Sampurna (1996) *World Wide Web* atau biasa yang disebut *web* merupakan sesuatu kumpulan informasi pada beberapa server yang terhubung satu sama lainnya dalam jaringan *internet*. Informasi-informasi dalam *web* mempunyai *link-link* yang menghubungkan informasi tersebut ke informasi lain didalam jaringan *internet*, yang biasanya ditandai dengan teks berwarna biru dan digarisbawahi, atau dapat berupa bentuk *icon* maupun gambar yang dikelilingi suatu kotak. Salah satu penyebab utama pesatnya pertumbuhan *WWW* ini adalah kemudahan dalam penggunaannya.

Hampir segala macam usaha dapat ditemui dalam *web*, bahkan bukan hanya informasi yang berbau bisnis, tetapi juga informasi tentang politik, hiburan, kesehatan, pengetahuan dan berbagai macam informasi lainnya.

2.3.2 Cara Kerja WWW

Cara kerja pada *web* mencakup dua hal penting, yaitu: *software web browser* dan *software web server*. Kedua *software* ini bekerja seperti sebuah sistem *client server*. *Web browser* yang bekerja sebagai *client* memungkinkan kita dapat menginterpretasikan dan melihat informasi pada *web*, sedangkan *web server* yang bertindak sebagai *server* memungkinkan kita untuk menerima informasi yang diminta oleh *browser* (Sampurna, 1996). Jika suatu permintaan akan suatu

informasi datang, *web server* mencari *file* yang diminta tersebut kemudian mengirimkan ke *browser* yang memintanya.

2.3.3 Hypertext Transfer Protocol (HTTP)

Web merupakan terobosan baru sebagai teknologi sistem informasi yang menghubungkan data dari banyak sumber dan layanan yang beragam macamnya di *Internet*. Pengguna tinggal mengklikkan *mousenya* pada *link-link hypertext* yang ada untuk melompat ke dokumen-dokumen di berbagai lokasi di *Internet*. *Link-linknya* sendiri bisa mengacu kepada dokumen *web*, *server FTP (File Transfer Protocol)*, *e-mail* ataupun layanan-layanan lain.

Server dan *browser web* berkomunikasi satu sama lain dengan protokol yang dibuat khusus untuk ini, yaitu HTTP. HTTP bertugas menangani permintaan-permintaan (*request*) dari *browser* untuk mengambil dokumen-dokumen *web*.

HTTP bisa dianggap sebagai sistem yang bermodel *client-server*. *Browser web* sebagai *clientnya*, mengirimkan permintaan kepada *server web* untuk mengirimkan dokumen-dokumen *web* yang dikehendaki pengguna. *Server web* lalu memenuhi permintaan ini dan mengirimkannya melalui jaringan kepada *browser*. Setiap permintaan akan dilayani dan ditangani sebagai suatu koneksi terpisah yang berbeda.

Semua dokumen *web* dikirim sebagai *file* teks biasa. Sewaktu mengirimkan *request* kepada *server web*, *browser* juga mengirimkan sedikit informasi tentang dirinya, termasuk jenis-jenis *file* yang bisa dibaca olehnya. Informasi ini lalu

digunakan oleh *server web* untuk menentukan apakah dokumen yang diminta bisa dikirimkan kepada *browser* atau tidak.

HTTP bekerja di atas TCP (*Transmission Control Protocol*) yang menjamin sampainya data ditujuan dalam urutan yang benar. Bila suatu kesalahan terjadi selama proses pengiriman, pihak pengirim akan mendapatkan pemberitahuan bahwa telah terjadi ketidakberesan. Karenanya *server* dan *client* tidak harus menyediakan mekanisme untuk memeriksa kesalahan transmisi data, yang berarti mempermudah pengerjaan pemrograman. Namaun demikian, HTTP tidak memiliki apa yang disebut *session*, seperti halnya FTP, yang menjaga hubungan antara *server* dan *client* secara konsisten. Setiap halaman *web* yang dikirim akan melibatkan satu proses penyambungan antara *client* dan *server*, baru kemudian datanya ditransfer, koneksi antara *server* dan *client* diputus. Sifat ini membuat HTTP sering disebut dengan istilah *hit-and-run*.

Suatu halaman *web* seringkali berisi beberapa *file* gambar, atau beberapa *file-file* lain. HTTP memaksa *server* untuk menjalin hubungan baru setiap kali hendak mengirim satu buah *file*. Ini tidak efisien dan menguntungkan, mengingat proses hubung-putus-hubung semacam ini menyebabkan beban bagi jaringan.

Standar baru protokol HTTP, yaitu HTTP/1.1 dirancang untuk mengatasi masalah diatas. *Web* diarahkan agar mengarah ke penggunaan *persistent connection* (sambungan yang terjaga berkesinambungan) secara lebih efisien. Dalam HTTP/1.1, *server* tidak akan memutuskan hubungan dengan *client* pada akhir pentransferan dokumen. Hubungan tetap dibuka untuk melayani bila saja ada *request* lagi dalam waktu yang singkat. Hubungan baru akan diputus jika

melewati suatu batas waktu tertentu (yang bisa ditentukan oleh *administrator server*) *client* tidak mengirimkan *request* lagi.

Keuntungan lain dari *persistent connection* adalah penggunaan *pipelining*. *Pipelining* adalah proses pengiriman *request* berikutnya segera setelah *request* sebelumnya dikirimkan tanpa menunggu balasan dari *server* terlebih dahulu. *Server*nya tetap harus melayani setiap *request* secara berurutan, namun ini mengurangi waktu tunda antara setiap *request*. Hasilnya data akan lebih cepat sampai tujuan.

2.3.4 Uniform Resource Locator (URL)

URL (*Uniform Resource Locator*) adalah suatu sarana yang digunakan untuk menentukan lokasi informasi pada suatu *web* server. URL dapat diibaratkan suatu alamat, dimana alamat tersebut terdiri atas:

1. protokol yang digunakan oleh suatu *browser* untuk mengambil informasi,
2. nama komputer (*server*) dimana informasi tersebut berada,
3. jalur/path serta nama *file* dari suatu informasi.

Format umum dari URL adalah sebagai berikut:

Protokol_transfer://nama_host/path/nama_file

Contoh: <http://www.akakom.ac.id/upt/index.html>

Dinama:

1. http adalah protokol yang digunakan
2. www.akakom.ac.id adalah nama *host* atau *server* komputer dimana informasi yang dicari berada
3. upt adalah jalur/path dari informasi yang dicari
4. index.html adalah nama *file* dimana informasi tersebut berada.

2.4 *Browser*

Untuk mengakses *web* memerlukan suatu program yang disebut *web browser* atau biasa disebut dengan *browser* saja. *Browser* merupakan suatu program yang dirancang untuk mengambil informasi-informasi dari suatu *server* komputer pada jaringan *internet*. Informasi-informasi ini biasanya dikemas dalam *page-page*, dimana setiap *page* bisa mempunyai beberapa *link* yang menghubungkan *web page* tersebut ke sumber informasi lainnya. Jika suatu *link* diklik, *browser* akan melihat alamat dari tujuan *link* tersebut dan kemudian mencarinya di *web server*. Jika *Browser* menemukan alamat dari tujuan *link* tersebut, *Browser* akan menampilkan informasi yang ada, dan jika tidak menemukan *browser* akan memberikan suatu pesan yang menyatakan bahwa alamat tujuan *link* tersebut tidak dapat ditemukan (Sampurna, 1996).

2.4.1 Sekilas Tentang Pemrograman *Web (Web Programming)*

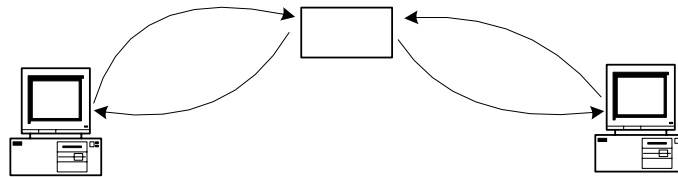
Saat ini *web* merupakan salah satu sumber informasi yang banyak dipakai. Berbagai aplikasi *Web* dibuat dengan tujuan agar pemakai dapat berinteraksi dengan mudah dan cepat, melalui dunia *internet*. Aplikasi *Web* tidak lagi terasa terbatas sebagai pemberi informasi yang statis, melainkan juga mampu memberikan informasi yang berubah secara dinamis, dengan cara melakukan koneksi terhadap *database*.

Situs/*Web* dapat dikategorikan menjadi dua yaitu *Web Statis* dan *Web Dinamis atau Internet*. *Web statis* adalah *Web* yang berisi atau menampilkan informasi-informasi yang sifatnya statis (tetap), sedangkan *Web dinamis* adalah *Web* yang menampilkan informasi serta dapat berinteraksi dengan *user* yang

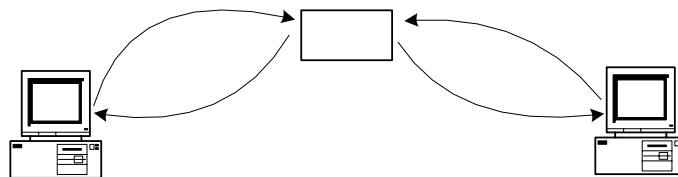
sifatnya dinamis. Sehingga untuk membuat *Web* dinamis dibutuhkan kemampuan pemrograman *Web*. Bahasa pemrograman *Web* ada dua kategori:

1. *server-side programming*
2. *client-side programming*

Pada *server-side programming*, perintah-perintah program (*script*) dijalankan di *Web Server*, kemudian hasilnya di *browser* dalam bentuk HTML biasa. Untuk lebih memperjelas perhatikan gambar 2.2 berikut:



Sedangkan *client-side programming* perintah program dijalankan di *Web Browser*, sehingga ketika diketik meminta dokumen yang menjalankan *script*, maka *script* tersebut akan di *download* dari servernya kemudian dijalankan di *browser* yang bersangkutan. Untuk lebih memperjelas perhatikan gambar 2.3 berikut:



Program *Web* yang tergolong dalam *server-side programming* seperti: CGI/PERL, ASP (*Active Server Page*), JSP (*Java Server Page*), PHP, CFM (*Cold Fusion*), dan lain-lain. Sedangkan *client-side programming* seperti: Java Script, Vb Script, HTML.

2.5 Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) merupakan standard bahasa yang di gunakan untuk menampilkan dokumen *web*, yang bisa anda lakukan dengan HTML yaitu: Mengontrol tampilan dari *web page* dan *contentnya*. Mempublikasikan dokumen secara *online* sehingga bisa di akses dari seluruh dunia. Membuat *online form* yang bisa di gunakan untuk menangani pendaftaran, transaksi secara *online*. Menambahkan objek-objek seperti *image*, *audi*, *video* dan juga *java applet* dalam dokumen HTML.

Browser merupakan *software* yang di install di mesin *client* yang berfungsi untuk menterjemahkan tag-tag HTML menjadi halaman *web*. *Browser* yang sering di gunakan biasanya *Internet Explorer*, *Netscape Navigator* dan masih banyak yang lainnya.

Program yang di gunakan untuk membuat dokumen HTML, ada banyak HTML editor yang bisa anda gunakan diantaranya: *Ms FrontPage*, *Dreamweaver*, *Notepad*.

2.5.1 Tag-tag HTML

Command HTML biasanya disebut TAG, TAG digunakan untuk menentukan tampilan dari dokumen HTML.

<BEGIN TAG> </END TAG>

Contoh: Setiap dokumen HTML diawali dan diakhiri dengan tag HTML.

<HTML>

. . .

<HTML>

Tag tidak *case sensitive*, jadi anda bisa gunakan <HTML> atau <html>

keduanya menghasilkan *output* yang sama. Bentuk dari tag HTML sebagai berikut:

<ELEMENT ATTRIBUTE = value>

Element - nama tag

Attribute - atribut dari tag

Value - nilai dari atribut.

Contoh:

<BODY BGCOLOR=lavender>

BODY merupakan element, BGCOLOR (*Background*) merupakan

atribut yang memiliki nilai *lavender*.

Untuk memulai bekerja dengan HTML anda bisa gunakan *editor Notepad*

1. Pad atau editor lainnya. Caranya:
2. Ketikkan tag-tag berikut di *notepad*.
3. Simpan dengan nama *file* one.htm atau one.html
4. Buka dengan *browser internet explorer* *file* one.htm maka *outputnya* akan di tampilkan seperti gambar di bawah ini.

2.5.2 Struktur HTML Dokumen

Dokumen HTML bisa di bagi mejadi tiga bagian utama:

HTML

Setiap dokumen HTML harus di awali dan di tutup dengan tag HTML

<HTML></HTML> tag HTML memberi tahu *browser* bahwa yang di dalam kedua tag tersebut adalah dokumen HTML.

HEAD

Bagian *header* dari dokumen HTML di apit oleh tag `<HEAD></HEAD>` di dalam bagian ini biasanya dimuat tag *TITLE* yang menampilkan judul dari halaman pada *titlenya browser*. Selain itu *Bookmark* juga menggunakan tag *TITLE* untuk memberi mark suatu *web site*. *Browser* menyimpan “*title*” sebagai *bookmark* dan juga untuk keperluan pencarian (*searching*) biasanya *title* di gunakan sebagai *keyword*. *Header* juga memuat tag *META* yang biasanya di gunakan untuk menentukan informasi tertentu mengenai dokumen HTML, anda bisa menentukan *author name*, *keywords*, dan lainnya pada tag *META*. Contoh:

```
<META name="Author" contents="Bocah Gunung"> Author dari dokumen tersebut adalah “Bocah Gunung” Atribut http-equiv dapat di gunakan untuk meletakkan nama HTTP Server atribut untuk menciptakan HTTP header. Contoh: <META http-equiv="Expires" content="Wed, 7 May 2003 20:30:40 GMT"> yang akan menciptakan HTTP header : Expires: Wed, 7 May 2003 20:30:40 GMT Sehingga jika dokumen di cached, HTTP akan megetahui kapan untuk mengupdate dokumen tersebut pada cache.
```

BODY

Dokumen *body* di gunakan untuk menampilkan text, image *link* dan semua yang akan di tampilkan pada *web page*.

```
<html>
<head>
<title>Welcome to HTML</title>
</head>
<body bgcolor="lavender">
<p>Dokumen HTML yang Pertama</p>
</body>
</html>
```


2.6 Mengenal PHP

Pada awalnya, PHP dirancang untuk diintegrasikan dengan *Web Server Apache*. Namun, belakangan ini PHP juga bekerja dengan *Web Server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*), dan *Xitami*.

Pada saat ini, PHP cukup populer sebagai piranti pemrograman *Web*, terutama di lingkungan LINUX. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada server-server yang berbasis UNIX, Windows NT, dan Machintos. Bahkan versi untuk Windows 95/98 pun tersedia.

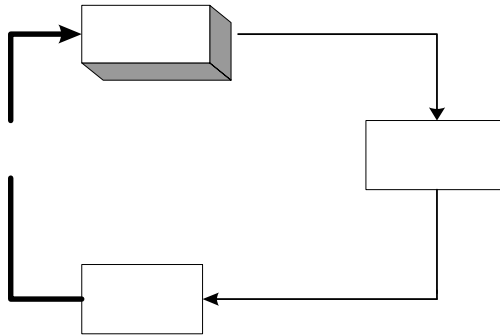
2.6.1 Skrip PHP

skrip PHP berkedudukan sebagai tag dalam bahasa HTML. Sebagaimana diketahui, HTML (*HiperText Markup Language*) adalah bahasa standart untuk membuat halaman-halaman *Web*. Kode PHP diawali dengan `<?php` dan diakhiri dengan `?>`. Pasangan kedua kode inilah yang berfungsi sebagai tag kode PHP. Berdasarkan tag inilah, pihak *server* dapat memahami kode PHP dan kemudian memprosesnya. Hasilnya dikirim ke *browser*. Kode PHP menyerupai bahasa C, walaupun tidak sepenuhnya sama.

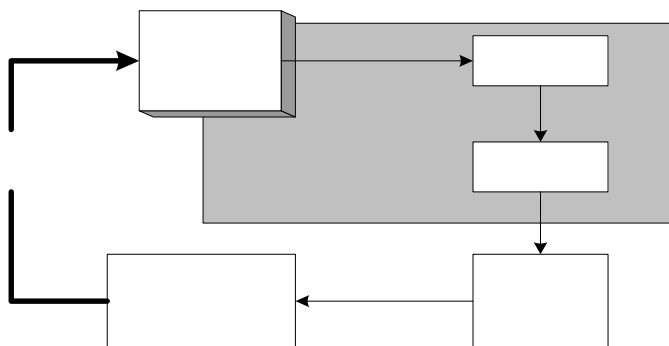
2.6.2 Konsep Kerja PHP

Metode kerja HTML diawali dengan permintaan suatu halaman *Web* oleh *browser*. Berdasarkan URL atau dikenal dengan sebutan alamat *internet*, *browser* mendapatkan alamat dari *Web Server*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *Web Server*. Informasi yang disampaikan oleh *Web Server* adalah nama *browser*, *versinya*, dan *system operasi*.

Selanjutnya, *Web Server* akan mencari berkas yang diminta dan memberikan isinya ke *browser*. *Browser* yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan menampilkan ke layar pemakai.



Tidak berbeda dengan HTML, prinsip PHP serupa dengan HTML. Hanya saja ketika berkas PHP yang diminta didapatkan oleh *Web Server*, isinya segera dikirim ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke *Web Server*. Selanjutnya, *Web Server* menyampaikan ke *Client*.



Salah satu kelebihan PHP adalah mampu berkomunikasi dengan berbagai *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari *database*, merupakan hal yang mudah untuk diimplementasikan. Itulah sebabnya sering dikatakan bahwa PHP sangat cocok untuk membangun halaman-halaman *Web* dinamis.

Pada saat ini PHP sudah dapat berkomunikasi dengan berbagai *database* meskipun dengan kelengkapan yang berbeda-beda. Beberapa diantaranya adalah: dBASE (dBASE III+, Visual dBASE, Visual FoxPro, dan semacamnya), DBM, *FilePro* (Personix, inc), Informix, Ingres, InterBase, Microsoft Access, MSQL, Oracle, ProstreSQL, Sybase.

2.6.3 Cookies

Cookies merupakan mekanisme penyimpanan sebuah variable data dengan waktu tertentu pada sisi *client (browser)*. Data pada *cookies* inilah yang pertama kali akan dikirimkan ke *server*, yang kemudian disimpan dalam *browser web*. *Cookies* ini memiliki kemampuan untuk menyimpan serta memanggil kembali data yang ada pada *hardisk client*.

Hanya saja *cookies* ini sifatnya sementara sehingga ketika *browser* ditutup, atau waktu menyimpan *cookies* tersebut sudah habis, maka *cookies* juga akan dihapus. Dengan demikian, ketika *browser* dibuka kembali atau melewati jangka waktu tertentu, nilai *cookies* akan kosong.

Sebelum *output* program dikirim ke *browser client*, *cookies* harus dideklarasikan terlebih dahulu. Untuk mendeklarasikannya dengan menggunakan fungsi *setcookie()*.

2.6.4 Session

Banyak sekali kegunaan yang diperoleh ketika anda menggunakan *session*. *Session* ini akan dimulai ketika pengunjung mulai masuk *web* site dan akan berakhir begitu pengunjung tersebut menutup *web* site. Dengan demikian, pengunjung akan mendapat variable yang terus ada selama ia melakukan kunjungan tersebut.

Dalam *session* ini, setiap pengunjung akan diberikan sebuah *id member*, yang disebut dengan *session_id*. Sama halnya dengan *cookies*, *session* akan hilang begitu anda menutup *browser*.

2.6.5 Keunggulan PHP

a. Pengaksesan *file* GIF dengan PHP

PHP tidak terbatas hanya untuk menghasilkan *output* halaman HTML. PHP juga dapat digunakan untuk menghasilkan *file* gambar GIF. Untuk dapat menampilkan *file* GIF sintak PHP harus *compile* dengan GD *Library* yang mengandung fungsi-fungsi manipulasi GIF.

b. *File Upload*

PHP mampu menerima *file Upload* dari setiap *browser* yang RFC-1867 *compliant*. Kemampuan ini memungkinkan *user* untuk mengupload *file* teks atau binary. Dengan fungsi-fungsi *authentication* dan *logic* PHP, kita memiliki kontrol untuk menentukan siapa yang boleh mengupload dan apa yang harus dikerjakan terhadap *file* setelah *file* selesai diupload

c. *HTTP Cookie*

PHP secara transparan mendukung *HTTP Cookie*. *Cookie* adalah mekanisme untuk menyimpan data pada *browser* sehingga *user* dapat dilacak atau diidentifikasi.

d. *Koneksi ke database server*

PHP dapat melakukan koneksi ke *database server* sehingga dapat melakukan manipulasi tabel.

e. *HTTP Authentication dengan PHP*

Kelebihan *HTTP Authentication* pada PHP hanya tersedia jika PHP berjalan modul *web server apache*. Pada *script PHP* dapat digunakan fungsi *header* () untuk mengirim pesan.

f. *Error Handling*

Jika menggunakan kontrol *error*, program yang dibuat akan semakin mudah. Hal ini karena adanya tampilan pesan *error* pada *browser* jika terjadi kesalahan dalam penulisan program. Kontrol *error* ini akan menunjuk baris yang *error* pada program yang dibuat sehingga dapat dengan cepat mengetahui serta membetulkannya.

2.7 Mengenal MySQL

Database telah menjadi bagian integral dalam kehidupan manusia. Sebuah pekerjaan akan terasa berat jika tidak didukung oleh *database*. Sebagai contoh, keberadaan *database* dirasakan sangat penting dalam pekerjaan di bank, universitas, dan perpustakaan.

Contoh lainnya adalah beberapa fasilitas di *internet* yang tidak mungkin berjalan tanpa adanya dukungan *database*, seperti *search engine*, *e-commerce*, dan *web site* berita. *Database* yang berjalan dan dikendalikan di komputer dinamakan *database server*.

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu, ia bersifat *free* pada pelbagai *platform* (kencuali pada *Windows*, yang bersifat *shareware* atau anda perlu membayar setelah melakukan evaluasi dan memutuskan untuk digunakan untuk keperluan produksi). Perangkat lunak MySQL sendiri bisa di-download dari <http://www.mysql.org> atau <http://www.mysql.com>.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL menyediakan program bernama *mysq* dari sisi *client*. Untuk melakukan manipulasi suatu *database*, kita perlu melakukan koneksi langsung ke *database* tersebut terlebih dahulu.

2.7.1 Keunggulan MySQL

Berikut ini adalah beberapa keunggulan MySQL:

1. Mampu menangani jutaan *user* dalam waktu yang bersamaan.
2. Mampu menampung lebih dari 50.000.000 *record*.
3. Sangat cepat mengeksekusi perintah.

4. Memiliki *user privileges system* yang mudah dan efisien.

2.7.2 Tahap membuat tabel

User privilege sistem yang dimiliki MySQL adalah pembagian hak untuk mengakses *database* dengan menggunakan *username* dan *password*. Kewenangan membuat *database*, *username*, *password* sepenuhnya berada ditangan *administrator server*.

Beberapa perusahaan *web hosting* membebaskan pelanggannya membuat *database*, *username*, *password* melalui *control panel* yang disediakan. Biasanya, saat pertamakali mendaftar pada perusahaan *web hosting*, *administrator server* akan memberikan informasi mengenai penggunaan *database*.

Paling tidak ada tiga cara untuk membuat tabel, yaitu melalui perintah baris (*command line*) dengan fasilitas telnet ke komputer *server*, mengeksekusi *file* PHP yang telah diupload di komputer *server* dari komputer *client*, dan melalui program visual seperti phpMyadmin atau MySQL-Front.

2.8 Apache

Untuk menjalankan PHP dan MySQL, anda membutuhkan *web server*. *Web server*, yang juga dikenal dengan istilah HTTPD (*HyperText Transfer Protocol Daemon*) atau *HTTP server*, adalah *service* yang bekerja untuk melayani *request* dari *HTTP client* (*Web browser*) ke komputer *server*.

PHP dan MySQL dapat bekerja sama dengan banyak *web server*. Salah satu *web server* yang dikenal dan sudah terbukti ketangguhannya serta konektivitasnya dengan PHP dan MySQL, adalah Apache.

Seperti halnya PHP dan MySQL, Apache juga dikembangkan oleh komunitas *open source* di *internet*. Saat ini Apache merupakan *web server* yang paling populer. Berdasarkan hasil penelitian *Netcraft Web Server Survey* pada tahun 2002, 63% *web site* didunia menggunakan Apache sebagai *web server*.

Untuk *menginstal* Apache, anda dapat mendownload source terbarunya yang sudah stabil dari <http://httpd.apache.org/download.cgi>.

2.9 Cara Kerja Apache, PHP, dan MySQL

Apache, PHP, dan MySQL adalah tiga sekawan yang bekerja di komputer *server* untuk melayani *request* dari komputer *client* melalui jalur HTTP. Apache bertugas sebagai penjaga gawang komputer *server*.

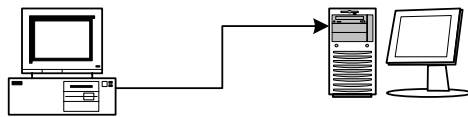
Ketika komputer *client* menyampaikan *request*, Apache memeriksa apakah *request* ini membutuhkan PHP. Jika ya, ditandai dengan *ekstensi file.php*, maka Apache akan mengirimkan *request* tersebut ke PHP.

Selanjutnya PHP akan memeriksa bagian mana dari *request* tersebut yang membutuhkan pelayanannya, yang ditandai dengan tag pembuka dan penutup PHP (`<?php...?>`). PHP akan menjawab *request* untuk melakukan sesuatu dan mengirimkannya kembali ke Apache. Selanjutnya tugas Apache adalah menyampaikan hasil jawaban PHP kepada komputer *client* melalui jalur HTTP.

Jika dalam *script* PHP ditemukan perintah untuk memasukan, memperbarui, menghapus, atau menampilkan data dari MySQL, maka PHP akan mengirimkan *request* tersebut kepada MySQL untuk ditindaklanjuti. Setelah selesai mengerjakan perintah yang diberikan kepadanya. MySQL mengirimkan kembali data ke PHP untuk disampaikan ke Apache.

Selanjutnya, Apache yang akan menghubungi komputer *client* untuk menyampaikan apakah *requestnya* dapat dipenuhi atau tidak. Jadi, selain sebagai penjaga gawang, Apache juga berperan sebagai juru bicara komputer server pada jalur HTTP.

Perhatikan bagaimana komputer *client* dan server berkomunikasi melalui jalur HTTP dalam ilustrasi berikut:



2.10 Konsep Basis Data

Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis dapat diartikan sebagai markas atau gudang, tempat berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Basis data sendiri dapat didefinisikan dalam berbagai sudut pandang, seperti:

- a. himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat digunakan kembali,
- b. kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan,

- c. kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Sistem basis data terdiri dari empat komponen, yaitu data, perangkat keras, perangkat lunak, dan *brainware*.

1. Data merupakan sekumpulan fakta yang akan diolah menjadi informasi. Data ini akan dikumpulkan pada suatu media penyimpan dalam bentuk basis data.
2. Perangkat Keras merupakan peranti komputer yang digunakan untuk menyimpan dan memproses data yang telah disimpan.
3. Perangkat Lunak merupakan program-program yang idgunakan untuk memberi perintah pada komputer untuk mengerjakan sustu pekerjaan tertentu.
4. *Brainware* merupakan orang yang berkepentingan dengan data serta hasil pengolahan data.

2.10.1 Struktur Basis Data

Setiap tabel terdiri atas baris (*row*) dan kolom (*Column*). Baris sering disebut *record* atau *tupple*, kolom sering disebut *field* atau atribut. Setiap baris memuat satu kesatuan. Jumlah baris sering disebut *cardinality* atau kardinalitas. Jumlah field sering disebut *degree* atau *arity*.

Supaya data dapat diakses dengan lebih mudah dan cepat, maka data harus diurutkan berdasarkan suatu *field* tertentu. *Field* yang dijadikan dasar pengurutan tersebut disebut *field* kunci (*key field*). Ada beberapa katagori kunci :

1. *Primary Key* (Kunci Utama), kunci utama yang dijadikan dasar pengurutan.
2. *Secondary Key* (Kunci Sekunder), kunci kedua yang dijadikan dasar pengurutan.
3. *Foreign Key* (Kunci tamu), *field* yang dijadikan dasar hubungan dengan tabel lainnya.
4. *Candidate Key* (Kunci Kandidat), *field* yang mempunyai kemungkinan untuk dijadikan *field* kunci.

Biasanya *field* yang dijadikan kunci adalah *field* yang datanya bersifat unik. (artinya tidak mungkin ada duplikasinya).

2.10.2 Normalisasi

Proses *normalisasi* merupakan proses pengelompokan elemen data menjadi tabel-tabel pada suatu basis data yang menunjukkan entitas serta relasi antar entitas tersebut. Pada umumnya, *normalisasi* dilaksanakan sampai dengan bentuk ke 3 (3NF).

1. Bentuk tidak normal yaitu proses yang dilaksanakan hanya merupakan pengumpulan data (semua data). Semua data yang ada dicatat apa adanya.
2. Bentuk normal ke-1 (1NF) yaitu bentuk yang diambil dari bentuk tidak normal kemudian dicari atribut (*field*) pada dokumen dasar tersebut. Setelah itu dibuat format tabel dan kemudian data ditempatkan pada masing-masing kolom yang telah ditetapkan.

3. Bentuk normal ke-2 (2NF) yaitu bentuk yang didasari oleh 1NF, dicari atribut yang bias mewakili suatu entitas. Atribut tersebut kemudian dijadikan kunci utama yang bersifat unik.

2.10.3 Entity Relationship

Pemodelan data merupakan proses untuk merepresentasikan pandangan pengguna terhadap data. Bagian ini merupakan bagian ini merupakan bagian yang penting dalam pengembangan aplikasi basis data. Salah satu pemodelan data adalah model E/R (*entity Relationship*).

1. *Entity* merupakan sesuatu yang bisa diidentifikasi dalam lingkungan kerja dari pengguna.
2. *Atribut* mendeskripsikan karakteristik dari entitas yang bersangkutan.
3. *Identifier* merupakan atribut dari suatu entitas yang mewakili entitas tersebut, jadi bisa dikatakan bahwa identifier merupakan kunci.
4. *Relationship* merupakan hubungan antara satu entitas dengan entitas yang lain. Suatu relasi dapat mempunyai atribut. Suatu relasi dapat terdiri atas lebih dari dua entitas.
5. Ada tiga kemungkinan relasi antar entitas :
 1. *One to One* (1 : 1)
Yaitu hubungan satu entitas lawan satu entitas
 2. *One to Many* (1 : M)
yaitu hubungan satu entitas dengan banyak entitas
 3. *Many to Many* (N : M)
Yaitu hubungan banyak entitas lawan banyak entitas

2.10.4 Diagram Arus Data (DAD)

Pada tahap analisis, diagram arus data sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut disimpan.

Diagram Arus Data (DAD) akan menjelaskan kepada pengguna bagaimana nantinya fungsi-fungsi pada sistem informasi secara logika bekerja. Diagram arus data ini digambarkan dalam bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan kegiatan dari sistem informasi berbasis komputer. Ada beberapa simbol yang digunakan DAD yaitu:

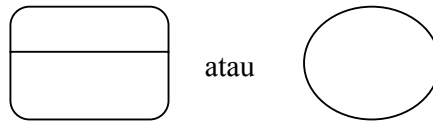
1. *External Entity* (kesatuan luar) yang disimbolkan dengan suatu notasi kotak. Kegiatan luar merupakan kesatuan dari lingkungan luar sistem yang akan memberikan *input* atau menerima *output* dari sistem. Contoh gambarnya adalah sebagai berikut.



Gambar 2.7 Simbol *External Entity*

2. *Process* (proses) disimbolkan dengan lingkaran atau persegi panjang tegak dengan setiap sudutnya tumpul. Suatu proses adalah kegiatan atau kerja yang dilakukan organisasi, mesin atau komputer.

Contoh gambarnya adalah sebagai berikut.



Gambar 2.8 Simbol *Process*

3. *Data Flow* (arus data) dilambangkan dengan garis atau panah. Arus data ini menunjukkan arus data yang dapat berupa masukan atau hasil dari proses sistem.

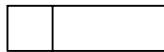
Contoh gambarnya adalah sebagai berikut.



Gambar 2.9 Simbol *Data Flow*

4. *Data Store* (penyimpanan data) disimbolkan dengan sepasang garis horizontal paralel. Simpanan data merupakan simpanan dari data yang dapat berupa *file* atau database, arsip atau catatan manual.

Contoh gambarnya adalah sebagai berikut.



Gambar 2.10 Simbol *Data Store*