

BAB II

LANDASAN TEORI

2.1 Perl

PERL (Practical Extraction and Report Language). Perl dibuat pada tahun 1986 oleh Larry Wall. Pada saat ini perl adalah salah satu *scripting language* yang sangat populer yang banyak digunakan pada system operasi linux/unix platform. Pada tugas akhir kali ini digunakan perl versi 5.004_04. Perl memiliki syntax yang mirip dengan bahasa C, selain itu perl juga banyak mengandung model-model command dari unix/linux shell script. Perl juga memiliki elemen-elemen yang sama dengan HTML. Pembentukan sebuah halaman web berikut fungsinya dapat dideklarasikan dalam sebuah program yang ditulis dengan bahasa perl ini.

2.1.1. Memulai Perl

Perl biasanya sudah terdapat pada sistem operasi linux ketika mulai melakukan proses instalasi pada komputer. Untuk memastikannya cukup mengetikkan perintah tertentu pada command prompt.

```
$ perl -v [enter]
```

Sedangkan untuk mengetahui pada directory manakah program perl tersimpan, cukup dengan memberikan perintah sebagai berikut :

```
$ which perl [enter]
```

Secara umum perl akan terdapat pada directory /usr/bin/perl. Lokasi directory ini sangat penting artinya untuk setiap program yang dibuat, tanpa pemberian tanda lokasi yang benar sistem operasi akan kesulitan untuk menentukan dimanakah

program perl berada. Lokasi atau directory program perl akan digunakan setiap memulai membuat program dengan perl.

Berikut adalah program perl sederhana yang akan menampilkan sebuah kalimat.

```
#!/usr/bin/perl

print "Red Hat Linux Unleashed, 3rd edition\n";
```

program di atas akan menampilkan kalimat yang berada diantara tanda kutip yaitu Red Hat Linux Unleashed, 3rd edition. Tanda \n digunakan untuk pindah baris.

2.1.2. Menjalankan Perl.

Sebelum menjalankan program, simpan terlebih dahulu program tersebut. Nama program tersebut, misal trivial.pl, selanjutnya rubah *file permission* dari file trivial.pl dengan perintah `chmod +X` untuk merubah file trivial.pl menjadi *executable file*.

2.1.3. Struktur Data dan Variabel Perl.

Perl tidak memiliki konsep data type seperti integer, string, char dan sebagainya. Perl memiliki beberapa macam variabel sebagai berikut :

- a. Scalar Variables, ditunjukkan dengan karakter dollar (\$) sebelum nilai variabel [*\$variables*], yang dapat diisi dengan angka atau huruf. Dapat pula menggunakan variabel yang sama baik untuk text/data string maupun data numeric. Syntax dari scalar: `$scalar = "expression";`

Berikut adalah contoh penggunaan variabel scalar :

```
#Configuration parameters
# this program's name
$programe = "mailform"
```

- b. List, list merupakan deretan dari sejumlah nilai dari variabel sacalar, huruf, angka, atau kombinasi dari keduanya.
- c. Arrays, disimbolkan dengan @arrays, berisi satu atau lebih elemen yang dapat diurut dengan menggunakan index. Sebagai contoh \$names[12] akan memberikan sejumlah 13 elemen di dalam array @names. Perlu diingat bahwa penomoran dimulai dari 0.
- d. Associative arrays, diindikasikan dengan %assoc_array, menyimpan nilai-nilai yang dapat dijadikan acuan dengan kata kunci. Sebagai contoh \$days{Feb} akan memberikan element dalam associative arrays %days yang dapat berkorespondensi langsung dengan kata “Feb” sebagai kata kunci.

2.1.4. Statement Kondisi.

Struktur IF/ELSE sangat bermanfaat dalam pemrograman untuk mengambil keputusan terhadap kemungkinan adanya lebih dari satu alternatif. Apabila kondisi bernilai benar, maka perintah-perintah yang berada dalam kelompok statement akan dieksekusi namu apabila kondisi bernilai salah maka perintah-perintah yang pada kelompok statement ke dua yang akan dieksekusi. Dalam perl struktur if/else memiliki syntax :

```

If (expression) {
    Statements;
}
if (expression) {
    # executed if true
    statements;
} ELSE {
    # executed if false
    statements;
}

```

Pada perl syntax dari kondisi IF/ELSE dapat dipersingkat menjadi **ELSIF** adalah sebagai berikut :

```

If      (Condition/Expression) {
        Statement(s) ;
        }
elsif   (Condition/expression) {
        statement(s)
        }
else    {
        statement(s)
        }

```

Tanda { } (braces) yang digunakan pada pernyataan di atas digunakan untuk menandai kelompok-kelompok expression, dan digunakan juga untuk mengakhiri statement yang ada. Condition dapat saja berupa pernyataan (statement) atau perbandingan (Comparison). Apabila sebuah statement bernilai “benar”, maka statement tersebut akan dieksekusi. Nilai benar yang dimaksudkan disini adalah :

1. angka bukan 0 (nol)
2. string yang tidak bernilai 0 (nol) atau kosong .
3. kondisi-kondisi lainnya yang membuat bernilai benar.

Contoh penggunaan kondisi if/else :

```

If      ($favorite eq "Chocolate") {
        Print " I like chocolate too.\n";
        }
elsif   ($favorite eq "spinach") {
        print "Oh, I don't like spinach.\n";
        }
else    {
        print "Your favorite food is $favorite. \n"
        }

```

2.1.5. Looping.

Perl memiliki empat tipe looping :

a. for.

For menampilkan sebuah statement (pernyataan) atau sekelompok statement pada sebuah kelompok kondisi, yang didefinisikan sebagai berikut :

```
For (start condition; end condition; increment function) {
    Statement(s)
}
```

Start condition merupakan initial expression, dimana perintah-perintah yang ada dalam kelompok atau blok statement akan dieksekusi, end condition merupakan terminated condition expression dimana merupakan kondisi perulangan (loop) akan berakhir, sedangkan increment function merupakan loop expression. Pada awal proses loop, kondisi awal harus sudah dideklarasikan terlebih dahulu. Setiap kali loop di jalankan atau dieksekusi fungsi *increment* akan berubah secara berurutan sampai kondisi nilai akhir tercapai. Dibawah adalah contoh dari loop dengan menggunakan for.

```
For ($i=1; $i<=10; $i++) {
    Print "$i\n"
}
```

b. Foreach.

Foreach menampilkan sebuah statement atau sekelompok statement dimana setiap elementnya berada dalam satu kelompok, seperti list atau array.

```

Foreach [SCALAR VARIABEL] (LIST) {
    STATEMENTS;
}

```

Setiap kali melalui loop, nilai pada variabel scalar akan dipindahkan ke elemen berikutnya dalam list, dan setelah itu statement akan dieksekusi.

Berikut adalah contoh penggunaan foreach.

```

Foreach $name (@names) {
    Print "$name \n"
}

```

c. While.

While akan mengeksekusi kelompok perintah yang berada dalam blok statement secara berulang, selama kondisi tertentu bernilai benar. While akan berhenti mengeksekusi statement apabila kondisi telah bernilai salah.

```

While (expression) {
    Statements;
    Statements;
}

```

contoh dari loop while adalah sebagai berikut :

```

While ($x<10) {
    Print "$x \n";
    $x++;
}

```

d. Until.

Until adalah kebalikan dari statement while. Selama kondisi tertentu bernilai salah, akan terus menjalankan statement sampai kondisi berubah menjadi benar, until akan berada dalam kondisi terminated (berhenti).

```

Until ($x>10) {
  Print "$x \n";
  $x++;
}

```

2.1.6. Conditional.

Conditional berfungsi untuk membandingkan nilai-nilai yang berbeda. Perl menyediakan dua paket conditional, satu untuk tipe data numeric, dan yang lainnya untuk data strings.

Perl	Usage
Eq	Equal
Ge	Greater than or equal to
Gt	Greater than
Le	Less than or equal to
Lt	Less than
Ne	Not equal
Cmp	Returns

Tabel 2.1. String comparisson

Perl	Usage
==	Equal
!=	Not Equal
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than wqual to
<=>	Return

Tabel 2.2. Numeric comparisson

2.1.7. Subrutin

Subrutin adalah bagian dari perl, yang dapat digunakan untuk mengeksekusi perintah-perintah khusus.

Pernyataan sub didefinisikan dalam sebuah blok dimulai dari tanda '{' sampai '}'. Syntax dari subrutin adalah sebagai berikut :

```
Sub NAME {
    Subroutines STATEMENTS;
}
```

Berikut adalah contoh subrutin yang akan mengeksekusi perintah pendefinisian masalah.

```
# define an error routine:
sub error {
    ($message) = @_ ;
    print (“<b>ERROR:<b>”,
        $message,
        “<p>Contact the author of the previous page for assistance\n”);
    exit(0);
}
```

2.2. CGI (COMMON GATEWAY INTERFACE)

CGI (Common Gateway Interface) adalah sebuah spesifikasi standar yang dikeluarkan oleh National Center for Supercomputing Applications (NCSA) yang digunakan sebagai media antara sebuah client dengan sebuah server pada suatu jaringan internet yang menjalankan aplikasi WWW. Seperti halnya aplikasi yang berjalan pada sebuah PC, pada web server juga dimungkinkan untuk menulis sebuah aplikasi yang dapat dijalankan pada client dan hasilnya ditampilkan pada sebuah browser. Secara umum dapat dijelaskan bahwa prinsip kerja dari CGI adalah ketika web server menerima sebuah permintaan dari client, CGI akan mencari semua nilai dari variabel yang disertakan pada dokumen HTML dan diarahkan pada standar input <STDIN> untuk proses penerimaan data. Kemudian webserver menjalankan aplikasi

yang sesuai dengan permintaan client, setelah semua proses yang diminta selesai dijalankan, maka hasil keluaran dari proses tersebut akan diarahkan pada standar output <STDOUT> untuk proses menghasilkan keluaran. Keluaran yang dihasilkan dari proses yang menggunakan antar-muka CGI adalah berbentuk dokumen HTML

Pemrograman CGI atau yang lebih dikenal dengan CGI script adalah tehnik pemrograman yang sama dilakukan pada saat menulis sebuah aplikasi pada umumnya. Hanya saja yang membedakan terletak pada proses masukan dan keluaran. CGI script menerima sebuah masukan setelah browser mengirimkan permintaan sebuah proses dimana variabel atau data yang dikirimkan disertakan dalam bentuk dokumen HTML. Begitu juga dengan proses keluaran, CGI script memberikan hasil keluaran juga dalam bentuk dokumen HTML.

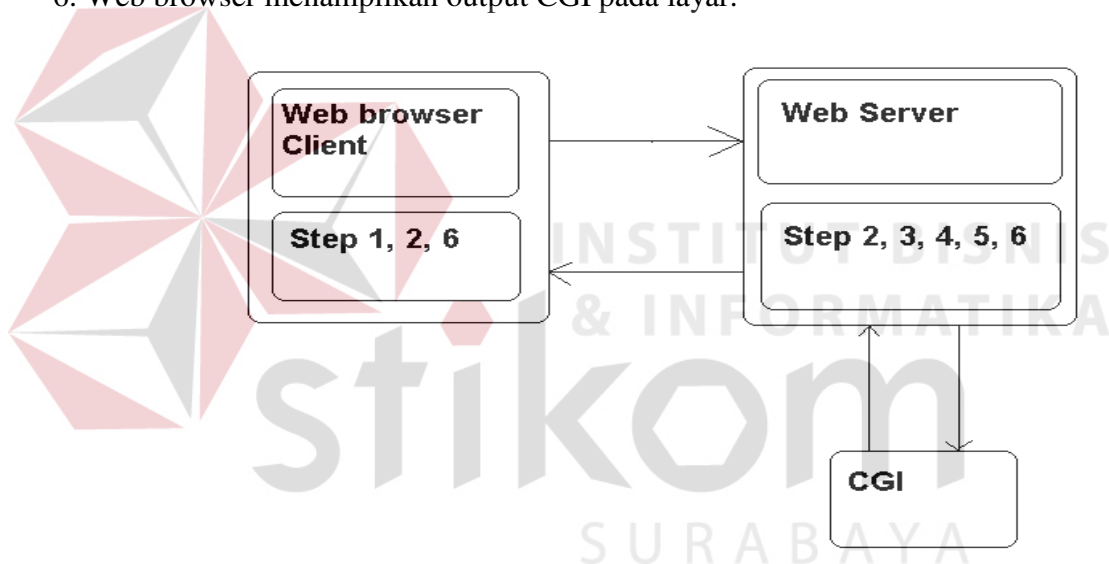
Untuk Menulis sebuah CGI script hampir semua jenis bahasa pemrograman dapat digunakan. Juga tidak tertutup kemungkinan untuk menulis CGI script dalam platform atau sistem operasi tertentu. Dengan kata lain pemilihan bahasa pemrograman hendaknya disesuaikan dengan sistem operasi dimana sebuah web server dijalankan.

Program CGI mempunyai gerbang dengan dua jalur, dimana data dapat disampaikan pada program CGI dan sebaliknya program CGI dapat menyampaikan data kembali kepada browser. Proses yang dilakukan oleh program CGI dari web browser cukup sederhana, sama seperti akses dokumen HTML biasa. Berikut urutan terjadinya suatu hubungan antara browser dan web server.

1. Program CGI dipanggil dengan melakukan suatu query atau menjalankan form.

Ketika user menekan tombol submit, browser mengirimkan data yang dimasukkan

- oleh pemakai ke server.
2. Web browser bernegosiasi dengan web server untuk menentukan hak akses dari pemakai.
 3. Web server memeriksa konfigurasi akses file yang telah ditentukan oleh *administration* server untuk memastikan apakah permintaan pemakai bisa dipenuhi.
 4. Web server memeriksa apakah program CGI tersebut ada. Jika ada, maka program tersebut dijalankan.
 5. Output yang dihasilkan oleh server diterima oleh browser. Syarat utama dari lancarnya proses input/output dari sistem CGI adalah program CGI harus menghasilkan sebuah dokumen dengan header HTML.
 6. Web browser menampilkan output CGI pada layar.



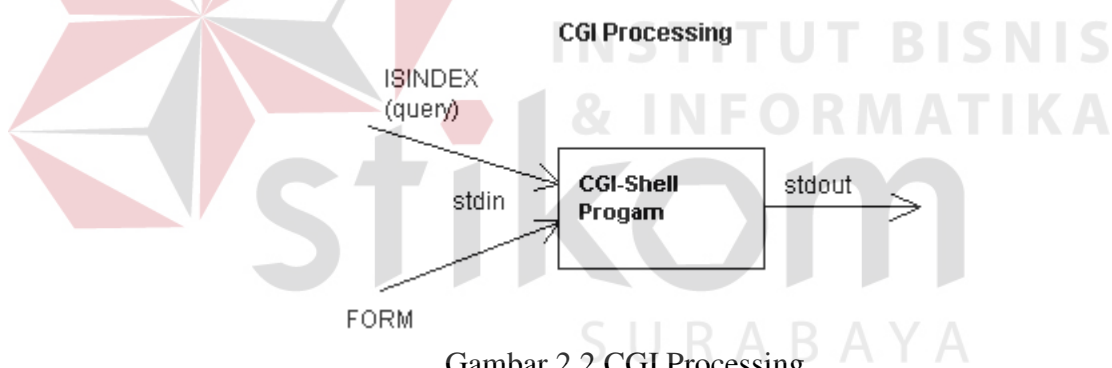
Gambar 2.1. Proses hubungan client dan server dalam CGI

Jika tidak ada data yang dikirim kembali ke client, program CGI harus memberi tanda bahwa tidak ada data yang disampaikan. Dalam hal ini hubungan

melalui protocol TCP/IP tetap jalan. Oleh karena itu, program yang dijalankan oleh web server harus selalu memberikan output berupa header dan isinya dengan mengikuti standar MIME. Bila sub proses ini tidak dijalankan maka komputer tempat menjalankan program browser akan mengalami gangguan. Hal ini disebabkan karena hubungan antara client dengan server tetap berlangsung ketika dokumen HTML hasil dari program CGI telah dibuat, seketika itu juga hubungan terputus.

Client dapat mengirimkan data atau *arguments* ke arah gateway melalui HTTP server . Program gateway yang sebelumnya merupakan suatu *program static* yang memiliki keluaran yang sama setiap kali berjalan, akan berubah menjadi sebuah *dynamic entity* yang akan merespon setiap kebutuhan user. Ada dua cara dimana client dapat mengirimkan data kepada program gateway, yaitu melalui :

1. Dengan menggunakan Environment variables (ISINDEX).
2. Sebagai standar input <STDIN> dimana data akan dikirim melalui forms kepada program.



Gambar 2.2 CGI Processing.

Environment variabel merupakan fasilitas yang selalu disertakan pada hampir semua sistem operasi, tidak terkecuali pada sebuah web server. Dengan environment variabel sebuah web server dapat mengetahui status dari sebuah permintaan yang

dikirim oleh browser yang dijalankan pada client. CGI mempunyai beberapa environment variabel standar yang terdapat pada semua web server untuk semua sistem operasi yang berbeda. Environment variabel tersebut adalah :

a. `SERVER_SOFTWARE`.

Menunjukkan nama dan versi dari software yang membuat CGI.

Contoh : `SERVER_SOFTWARE=NCSA/1.3`

b. `SERVER_NAME`.

Nama dari host, DNS, Alias, atau IP Address.

Contoh : `SERVER_NAME = que.mcp.com`

c. `GATEWAY_INTERFACE`.

Nama dan versi dari protocol CGI yang digunakan

Contoh : `GATEWAY_INTERFACE = CGI/1.1`

Variabel-variabel diatas adalah variabel yang mengandung informasi HTTP tentang server dan tidak bergantung pada client request.

d. `SERVER_PROTOCOL`.

Nama dan versi dari protocol yang digunakan dimana CGI sedang dijalankan.

Contoh : `SERVER_PROTOCOL = HTTP/1.0`

e. `SERVER_PORT`.

Port yang digunakan untuk melayani permintaan dari CGI.

Contoh : `SERVER_PORT = 80`

f. `REQUEST_METHOD`.

Metode yang digunakan oleh HTTP untuk menerima permintaan proses dari CGI.

Contoh : `REQUEST_METHOD = POST`

g. PATH_INFO.

Path tambahan yang digunakan pada saat memanggil sebuah URL.

Contoh : `http://www.innergy.com/cgi-bin/foo/whitepaper/may15`

`PATH_INFO = /whitepaper/may15`

h. PATH_TRANSLATED.

Alias yang digunakan oleh sebuah server untuk sebuah URL.

Contoh : `http://www.innergy.com/cgi-bin/foo/whitepaper/may15`, dan web server meletakkan semua dokumennya pada path `/bin/htdocs/docs`.

`PATH_TRANSLATED = /bin/htdocs/docs/whitepaper/may15`

i. SCRIPT_NAME.

Path dari sebuah script yang sedang dieksekusi.

Contoh : `/cgi-bin/foobar`

j. QUERY_STRING.

Semua karakter yang terletak di belakang tanda tanya pada sebuah URL.

Contoh : `http://www.innergy.com/cgi-bin/foo?gordon+susan`

`QUERY_STRING=Gordon+Susan`

k. REMOTE_HOST.

Host atau client yang meminta request pada web server.

Contoh : `REMOTE_HOST=slip-08.shore.net`

l. REMOTE_ADDR.

IP Address dari client yang meminta request pada web server.

Contoh : `REMOTE_ADDR=192.233.85.130`

m. AUTH_TYPE.

Metode yang dibuat untuk menentukan wewenang dari seorang pemakai.

Contoh : AUTH_TYPE=Perl.

n. REMOTE_USER.

Nama dari client yang meminta request pada web server.

Contoh : REMOTE_USER=aeinstein

o. CONTENT_TYPE.

Tipe dari Multipart Internet Mail Extensions (MIME) yang dikirimkan oleh web server.

Contoh : text/html.

p. HTTP_ACCEPT.

MIME yang dapat dikenali oleh web server.

Contoh : HTTP_ACCEPT=image/gif, image/x-bitmap, image/jpeg

q. HTTP_USER_AGENT.

Berisikan nama dari browser yang digunakan pada client.

Contoh : HTTP_USER_AGENT=Mozilla/20 (Win 16).

Variabel –variabel di atas bergantung pada kedatangan permintaan dari client.

2.2.1. Mengirim Data.

Mengirim data melalui environment variables (METHOD=GET) sangatlah baik namun memiliki banyak keterbatasan dan sebenarnya dapat menimbulkan kerusakan pada sistem. Sebuah program gateway memiliki string yang sangat panjang, yang terdiri dari URL ditambah dengan query string yang akan digunakan bersamaan dengan shell script. Hal ini akan menyebabkan *crash* karena shell script memiliki keterbatasan dalam panjang *command line* yang dimilikinya.

Untuk menghindari bahaya penggunaan teknik METHOD=GET data akan dikirim dengan menggunakan standard input ke program external. Sebuah form dengan METHOD=POST dapat digunakan untuk mengirim data menuju stdin program gateway. Data yang dikirim akan dikodekan dan setelah melewati proses dalam gateway data akan dikembalikan seperti kondisi semula. Berikut adalah potongan perl script yang dapat digunakan untuk menghindari keadaan di atas tersebut.

```
#!/usr/local/bin/perl
# name.pl
$user_input = read(STDIN, $_, $ENV{CONTENT_LENGTH});
print "Content - type : text/html\n\n";
print "You input \"\$user_input\" in the input boxes\n\n";
exit;
```

Program di atas akan menghasilkan screen output seperti biasanya kecuali pada hasil URL dimana tidak akan dicantumkan kode QUERY_STRING setelah nama program. Sebuah form yang menggunakan METHOD=POST akan membuat environment variabel sebaik stdin, dengan merubah tag form html dengan :<FORM METHOD=POST ACTION=http://www.some.box/name.pl/screen=subscribe> dan data akan dikirim pada stdin dan PATH_INFO variabel.

2.2.2. MIME pada CGI Environment

Ketika sebuah client melakukan proses request dan program cgi dieksekusi untuk memenuhi kebutuhan client, data dari sebuah form atau lebih akan ditulis pada standard output (stdout) dan akan dikirim oleh server kepada client.

Hal penting yang harus dimengerti pada komunikasi client-server-gateway. Adalah pengiriman data. Client dapat mengirim cara dengan dua cara, lewat url dan atau lewat message body (dalam form method=post).

Ada dua arti dasar dari client untuk mengirim data melalui server ke gateway program melalui URL (ISINDEX) atau melalui *message body* (melalui METHOD=POST form). Cara ini akan mengirimkan data ke gateway program melalui stdin program. Kemudian environment variabel CONTENT_LENGTH akan disiapkan sejumlah karakter yang akan dikirim.

Bagaimana server mengirim data ke gateway program. Pada saat ini server telah menerima data dari client. Ada dua cara untuk mengirim data ke gateway program.

- a. melalui stdin gateway program. Jika request_method dikirim, pertama kali server akan mengkodekan data, kemudian baru dikirim ke gateway program sebagai stdin.
- b. Melalui environment variabel server.

Gateway output. Sebuah program gateway harus memulai keluarannya dengan *proper header* yang dapat dimengerti oleh server. Ada tiga header yang dapat dikenali server.

- a. Content-type.
- b. Location : [URL]
- c. Status: [message strings]

2.3. Hyper Text Markup Language

HTML (Hyper Text Markup Language) adalah sebuah format dokumen yang hampir sama dengan dokumen teks, dimana struktur dokumen HTML mempunyai elemen-elemen seperti heads, paragraph, dan list. Format dokumen dalam bentuk HTML ini mampu untuk langsung ditampilkan dalam sebuah browser, yang

selanjutnya akan dapat dipakai sebagai sebuah lembar halaman dalam sebuah web.

Dokumen dengan format HTML adalah dokumen teks standar yang mempunyai beberapa tanda khusus di dalamnya. Tanda-tanda khusus yang dimaksud disebut TAG. Sebuah tag adalah gabungan dari beberapa karakter atau string yang diapit dengan tanda lebih kecil dan tanda lebih besar. Tag memberikan status dari sebuah teks dan bagaimana harus menampilkannya di dalam browser.

Terdapat tiga aturan yang perlu diperhatikan pada saat menulis sebuah dokumen HTML.

a. Spasi kosong diabaikan di dalam dokumen HTML.

Spasi kosong seperti tanda “ “, tanda tabulasi dan baris baru tidak dianggap sebagai suatu perintah pada saat dokumen HTML dibaca. Sehingga penempatan sebuah baris baru dengan melakukan penekanan tombol enter tidak akan memberikan hasil apa-apa. Hal ini berarti jika ingin menambahkan sebuah baris baru dapat menggunakan sebuah tag

b. Tag tidak membedakan antara penulisan huruf besar dengan huruf kecil.

Tag <Pre> dengan tag <PRE> adalah sama. Para penulis dokumen HTML biasanya menggunakan susunan huruf sama besar untuk menuliskan sebuah tag. Penggunaan penulisan dengan menggunakan susunan huruf besar dilakukan hanya untuk mempermudah dalam membedakan antara tag dengan teks atau dokumen.

c. Hampir semua tag mempunyai pasangan.

Sebuah tag biasanya mempunyai tag pembuka dan tag penutup. Sebuah tag penutup ditandai dengan tanda garis miring di depan tag tersebut. Sebuah contoh, tag <HTML> akan diakhiri dengan tag </HTML>.

Seperti halnya dokumen lainnya, dokumen HTML juga mempunyai struktur yang membedakan antara header dengan data pada dokumen tersebut. Struktur dari dokumen HTML adalah :

a. Deklarasi.

Pada bagian deklarasi selalu diawali dengan tag `<HTML>` pada awal dokumen dan ditutup dengan tag `</HTML>` di akhir dokumen. Bagian deklarasi bertujuan untuk memberitahu awal dan akhir dari dokumen HTML.

b. Header.

Bagian header di dalamnya terdapat beberapa tag yang disertakan pada bagian ini. Seperti `<TITLE>` dan tag `<BASE>`. Pada dasarnya bagian header bertujuan untuk memberikan penjelasan mengenai status, judul dan asal dari dokumen HTML.

c. Body.

Bagian body atau bagian utama dari dokumen HTML diawali dengan tag `<BODY>` dan diakhiri dengan tag `</BODY>`. Bagian body berisikan tentang informasi yang akan disampaikan kepada para pemakai fasilitas internet dan beberapa subrutin yang bertujuan memperindah tampilan yang dihasilkan.

Versi yang terbaru dari dokumen HTML mempunyai banyak tag yang memungkinkan sebuah dokumen HTML untuk diakses oleh beberapa browser untuk jenis sistem operasi yang berbeda. Berikut ini adalah beberapa tag dan penjelasannya yang sering digunakan dalam menulis dokumen HTML.

a. TITLE.

Berfungsi untuk memberikan penjelasan singkat berupa judul dari dokumen HTML.

b. BASE.

Berfungsi untuk mendefinisikan asal dari dokumen HTML yang berupa Uniform Resource Location (URL).

c. HEAD.

Berfungsi untuk memisahkan antara bagian header dari dokumen dengan bagian body pada dokumen HTML.

d. H1,H2,H3,H4,H5.

Huruf H merupakan kependekan dari Heading. Tampilan yang dihasilkan jika menggunakan tag <H> berupa judul dengan ukuran huruf yang besar.

e. HTML

Untuk memberikan keterangan status kepada browser bagian yang bukan HTML dan bagian yang merupakan dokumen HTML.

f. PRE.

Digunakan untuk memberitahu pada browser bahwa teks yang akan ditampilkan sudah diformat atau disusun sesuai dengan bentuk aslinya.

g. I.

Merupakan kependekan dari Italic. Huruf yang dihasilkan akan ditulis dengan cetak miring.

h. B.

Merupakan kependekan dari Bold atau cetak tebal. Penggunaannya hampir sama dengan tag .

i. DL.

Merupakan kependekan dari Definition List. Berisikan daftar dari item yang telah

didefinisikan.

j. DT.

Digunakan untuk mendefinisikan teks yang sudah terdapat pada deklarasi tag <DL>

k. UL.

Menandakan awal dari sebuah list atau daftar item. Penggunaan tag harus diikuti dengan penggunaan tag .

l. LI.

Digunakan setelah tag , tag berfungsi untuk mendefinisikan item dari sebuah list.

m. HR.

Merupakan kependekan dari Horizontal Rule yang berfungsi untuk menggambar sebuah garis mendatar.

n. BR.

Berfungsi untuk menyisipkan sebuah garis baru.

o. P.

Menandakan awal dari sebuah paragraf pada dokumen HTML.

p. IMG.

Digunakan untuk menyisipkan sebuah citra atau grafik yang ditampilkan dalam bentuk sebuah baris dokumen.

q. HREF.

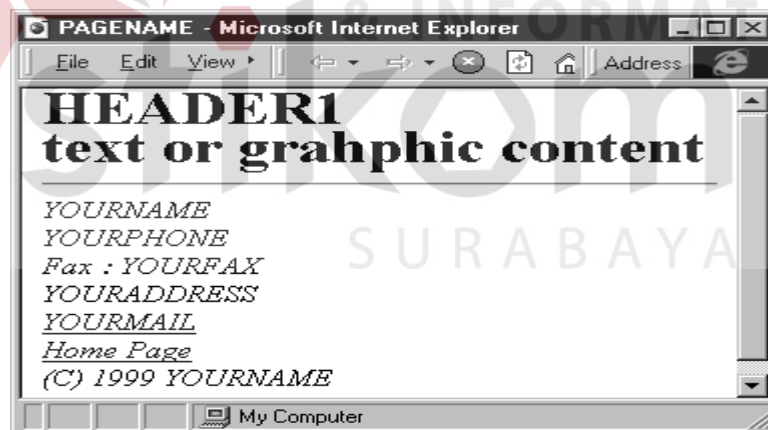
Merupakan kependekan dari Hypertext Reference. Tag HREF digunakan untuk menghubungkan sebuah dokumen HTML dengan dokumen html yang lain.

r. ISINDEX.

Dapat digunakan untuk mengirimkan informasi dalam bentuk sebuah script, sering digunakan untuk proses pencarian.

Contoh penggunaan tag HTML :

```
<HTML>
<HEAD>
<TITLE>PAGENAME</TITLE>
</HEAD>
<BODY BGCOLOR="#00FFFF">
<H1>HEADER1</H1>
text or graphic content<P>
<A HREF="page.html"><IMG SRC="photo.jpeg"></A>
<A HREF="page.html"><IMG SRC="photo.jpeg"></A><BR>
</BODY>
<HR>
<ADDRESS>
YOURNAME<BR>
YOURPHONE<BR>
Fax : YOURFAX<BR>
YOURADDRESS<BR>
<A HREF="mailto:YOURMAIL">YOURMAIL</A><BR>
<A HREF="index.html">Home Page</A><BR>
(C) 1999 YOURNAME
</ADDRESS>
</HTML>
```



Gambar 2.3. Contoh halaman HTML

Semakin bertambahnya orang menggunakan dokumen HTML untuk berbagai keperluan menjadikan HTML sebagai salah satu kebutuhan utama dalam menampilkan informasi. Untuk itu diperlukan beberapa fasilitas tambahan yang dapat menangani beberapa tugas yang tidak terdapat pada HTML versi sebelumnya. Pada HTML 3.2 terdapat beberapa fasilitas baru yang menghasilkan tampilan informasi yang lebih menarik. Fasilitas-fasilitas yang baru tersebut adalah sebagai berikut :

a. Tabel.

Tabel digunakan untuk menampilkan informasi yang hasilnya disajikan dalam susunan baris dan kolom. Tag-tag yang digunakan dalam membuat tabel adalah sebagai berikut:

a.1. TABLE.

Untuk menandakan awal dari sebuah tabel. Atribut-atribut yang digunakan pada tag <TABLE> adalah sebagai berikut :

a.1.1. HEIGHT.

Nilai yang diberikan dapat berupa angka atau prosentase. Atribut HEIGHT berfungsi untuk mengatur tinggi dari sebuah tabel.

a.1.2. WIDTH .

Seperti atribut HEIGHT, atribut WIDTH dapat mempunyai nilai berupa prosentase atau jumlah pixel.

a.1.3. BORDER

Menentukan lebar tepi batas dari sebuah tabel.

a.2. TR

Untuk menentukan jumlah baris dari sebuah tabel.

a.3. TH.

Tag <TH> digunakan untuk memberikan judul atau keterangan pada sebuah tabel. Judul tabel dapat diletakkan pada setiap baris. Atribut-atribut yang dapat mempengaruhi tampilan yang dihasilkan adalah sebagai berikut :

a.3.1. COLSPAN.

Berfungsi menggabungkan beberapa kolom untuk ditampilkan dalam satu kolom.

a.3.2. ROWSPAN.

Berfungsi menggabungkan beberapa baris untuk ditampilkan dalam satu baris.

a.4. TD

Memberikan data untuk setiap sel pada sebuah tabel. Atribut-Atribut yang dapat digunakan adalah sebagai berikut :

a.4.1. ALIGN

Atribut ALIGN digunakan untuk menampilkan data atau teks rata kanan, rata kiri atau rata tengah secara mendatar.

a.4.2. VALIGN

Atribut VALIGN hampir sama dengan atribut ALIGN, perbedaannya atribut menampilkan teks rata pada sumbu tegak.

Setelah mengetahui beberapa tag dan atribut yang dapat digunakan untuk membuat sebuah tabel, di bawah ini akan diberikan contoh dan tampilan penggunaan sebuah tabel dalam dokumen HTML.

```
<HTML>
<HEAD>
```

```

<TITLE>Example of table feature</TITLE>
</HEAD>
<BODY>
<TABLE BORDER=3>
<TR>
<TH COLSPAN=3> Time for a Nearest Neighbour Search </TH><P>
<TR><TH>Machine</TH><TH>32768 Nodes</TH><TH>65536 Nodes</TH><P>
<TR><TD>MM32K</TD><TD>2.2 msec</TD><TD>3.1 msec<BR></TH></TR>
<TR><TD>i486</TD><TD>350 msec</TD><TD>700 msec<BR></TH></TR>
<TR><TD>MIPS</TD><TD>970 msec</TD><TD>1800 msec<BR></TH></TR>
<TR><TD>Alpha</TD><TD>81 msec</TD><TD>177 msec<BR></TH></TR>
<TR><TD>Sparc</TD><TD>410 msec</TD><TD>820 msec<BR></TH></TR>
</TABLE>
</BODY>
</HTML>

```

Time for a Nearest Neighbour Search		
Machine	32768 Nodes	65536 Nodes
MM32K	2.2 msec	3.1 msec
i486	350 msec	700 msec
MIPS	970 msec	1800 msec
Alpha	81 msec	177 msec
Sparc	410 msec	820 msec

Gambar 2.4. Tampilan contoh penggunaan tabel pada dokumen HTML

b. FRAME.

Fasilitas frame pertama kali diperkenalkan oleh browser yang diproduksi oleh Netscape. Dengan Frame ini sebuah dokumen dapat ditampilkan menjadi beberapa bagian di dalam satu layar. Susunan dan tag yang digunakan untuk membuat frame adalah sebagai berikut :

b.1.FRAMESET.

Mendeklarasikan sebuah group frame baru. Frame dapat disajikan dalam beberapa kolom dalam satu layar atau beberapa baris dalam satu layar. Untuk

mendefinisikan sebuah frame dapat digunakan dengan membagi layar dalam bentuk prosentase atau jumlah pixel.

b.2.FRAME SRC.

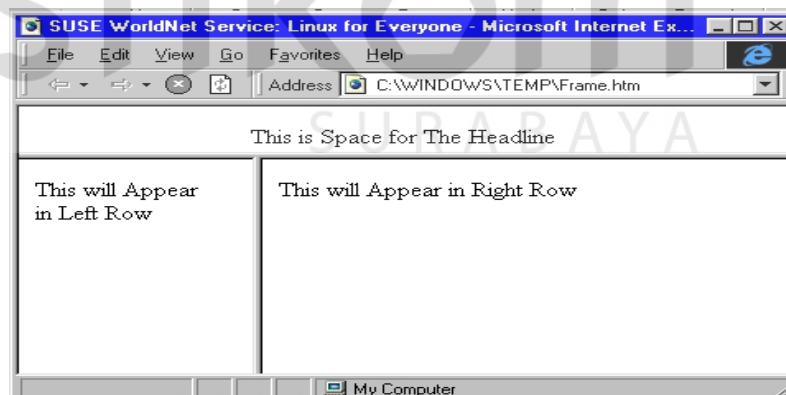
Dokumen yang akan ditampilkan pada salah satu bagian sebuah frame. Format dari dokumen yang akan ditampilkan adalah dalam bentuk dokumen HTML.

b.3.NOFRAME.

Tag <NOFRAME> digunakan untuk menampilkan informasi pada browser yang tidak mendukung fasilitas frame.

Di bawah ini akan diberikan potongan dokumen HTML yang menggunakan fasilitas frame dan tampilan yang dihasilkan.

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<title>frame2</title>
</head>
<body bgcolor="#FFFFFF">
<p>This will Appear in Right Row</p>
</body>
</html>
```



Gambar 2.5. Contoh penggunaan frame

c. FORM.

Salah satu fasilitas yang disediakan oleh HTML 3.2 untuk dapat berinteraksi dengan sebuah host atau web server adalah fasilitas form atau yang sering disebut dengan FILL-IN-FORM. Form pada HTML menyediakan berbagai macam jenis kontrol masukan seperti teks input, radio button, check box, dan command button. Terdapat beberapa tag yang digunakan untuk membuat form pada dokumen HTML.

c.1. FORM.

Tag <FORM> menentukan awal dan akhir dari sebuah form pada dokumen HTML. Atribut yang disertakan pada tag <FORM> adalah sebagai berikut :

c.1.1. ACTION.

Menentukan sebuah URL yang dijadikan sebagai rujukan jika sebuah form akan dikirimkan. Atribut ACTION harus disertakan pada setiap form dalam dokumen HTML.

c.2.2. METHOD.

Protocol HTTP menyediakan beberapa metode yang dapat digunakan untuk meminta sebuah proses mengambil data pada web server. Metode-metode yang disediakan adalah GET, PUT, POST dan HEAD.

c.2. INPUT.

Menyisipkan sebuah variabel yang akan diproses bersamaan pada saat sebuah form dikirimkan pada web server. Tag <INPUT> mempunyai nama

dan atribut. Penggunaan atribut yang berbeda akan memberikan bentuk masukan yang berbeda juga. Jenis kontrol masukan yang disediakan oleh tag <INPUT> adalah sebagai berikut :

c.2.1. CHECKBOX.

Checkbox adalah jenis kontrol masukan yang memungkinkan untuk memberikan nilai lebih dari satu pada suatu variabel.

c.2.2. RADIO.

Seperti pada checkbox , radio button merupakan sebuah pilihan yang berbentuk ganda. Akan tetapi yang membedakan antara check box dengan radio button adalah jika satu pilihan sudah dipenuhi, maka pilihan yang lain tidak dapat dipilih.

c.2.3. TEXT.

Sebuah masukan yang digunakan untuk memberikan nilai pada sebuah variabel dilakukan dengan mengetikkan karakter pada tempat yang telah disediakan.

c.2.4. HIDDEN.

Atribut HIDDEN digunakan jika suatu nilai variabel yang akan dikirimkan ke web server tidak ingin ditampilkan pada layar monitor.

c.2.5. PASSWORD.

Atribut password sama halnya dengan atribut text. Perbedaannya terletak pada tampilan yang dihasilkan. Jika atribut PASSWORD disertakan pada tag <INPUT> maka karakter yang ditampilkan

berupa tanda bintang.

c.2.6. RESET.

Digunakan untuk memberikan nilai kosong kepada semua variabel yang terdapat pada form di dalam dokumen HTML.

c.2.7. SUBMIT.

Bentuk dari attribut SUBMIT adalah sebuah tombol. Jika tombol tersebut ditekan maka sebuah form beserta dengan semua variabel yang terdapat di dalamnya akan dikirimkan ke dalam sebuah web server.

c.3. SELECT.

Bentuk masukan yang dihasilkan adalah sebuah list box. Biasanya di dalam tag <SELECT> selalu terdapat tag <OPTION> yang bertujuan untuk memberikan nilai dari item-item yang terdapat pada sebuah list box. Attribut yang digunakan untuk membuat sebuah list box adalah sebagai berikut :

c.3.1. MULTIPLE.

Menentukan banyaknya item yang akan ditampilkan dalam sebuah list box .

c.3.2. NAME.

Nama variabel dari list box. Atribut NAME harus dicantumkan.

c.3.3. SIZE.

Menentukan ukuran dari list box.

c.4. TEXTAREA.

Tag <TEXTAREA> digunakan untuk membuat sebuah daerah masukan

berupa text yang lebih dari satu baris. Untuk mendeklarasikan sebuah text area diperlukan tiga buah atribut yang harus selalu ada. Atribut-attribut yang digunakan adalah sebagai berikut :

c.4.1. COLS.

Menentukan besar kolom dari sebuah text area.

c.4.2. ROWS.

Menentukan lebar baris dari sebuah taext area.

c.4.3. NAME.

Memberikan nama dari sebuah tag <INPUT>.

Setelah mengetahui beberapa bentuk masukan dan atribut yang digunakan untuk membuat sebuah kontrol masukan, di bawah ini akan diberikan potongan program dan tampilan yang dihasilkan dari penggunaan sebuah form.

```
<HEAD>
<HTML>
<TITLE>Example of form</TITLE>
</HEAD>
<body>
<form action="/cgi-bin/sample" method="post">
<center>
<table width=7,5% cellpadding=8 border=2>
<tr><td align="center" colspan=5>
<font size+1>t</font>ravel planning service</td></tr>
<td align="left" colspan=2>
What brochure would you like? <br>
(check all that apply)
<blockquote>
<input name"vacloc" type="checkbox" value="bahamas">Bahamas<br>
<input name"vacloc" type="checkbox" value="honolulu">Honolulu<br>
<input name"vacloc" type="checkbox" value="las vegas">Las Vegas<br>
<input name"vacloc" type="checkbox" value="hoboken">Hoboken<br>
<blockquote></td>
<td valign="top" align="right" colspan=3>
<pre>
Username: <input name="username" type="text" rows=1 maxlegth=10 size"10">
```

```

Password : <input name="password" type="password" rows=1 maxlength=10
size"10">
</pre></td>
</tr>
<tr align="center"><!-- -Bottom line - ->
<td width=28%>Weeks of vacation :</td>
<td width=18%><input name="vacwks" type="radio" value="1">1</td>
<td width=18%><input name="vacwks" type="radio" value="2">2</td>
<td width=18%><input name="vacwks" type="radio" value="3">3</td>
<td width=18%><input name="vacwks" type="radio" value="4">4</td>
</tr>
</table>
<input type="submit" value="send this request">
<input type="reset" value="Clear form">
</center>
</form>
</body>
</html>

```

travel planning service

What brochure would you like?
(check all that apply)

Bahamas
 Honolulu
 Las Vegas
 Hoboken

Username:
Password :

Weeks of vacation : 1 2 3 4

send this request Clear form

Gambar 2.6. Tampilan contoh penggunaan form pada dokumen HTML.