

BAHASA PEMROGRAMAN WEB

(HTML, PHP, CSS, JavaScript, CMS, XML)

1. PENDAHULUAN

Pastilah kita pernah mengunjungi suatu situs yang sangat berkesan, misalnya dari sisi layout dan desainnya atau dari sisi interaktifnya, seperti bhinneka.com, amazon.com, ebay.com, dan lain-lain. Pastilah kita pernah mengisikan sesuatu pada sebuah situs baik itu sebuah buku tamu, polling, email dan lain-lain. Lalu kita juga dapat melihat pada halaman tersebut seperti tanggal saat ini, jam, dan jumlah pengunjung yang telah melihat situs tersebut.

Sebelum kita mempelajari bahasa – bahasa pemrograman dalam web lebih lanjut, ada baiknya kita mengerti dahulu cara kerja Internet dalam hal ini konsep client server dan peer to peer. Kita tidak usah membicarakan bagaimana data berjalan dan sampai ketujuan dengan melewati berbagai perangkat-perangkat jaringan computer karena tidak akan habis dibahas dalam satu bab.

Konsep client server adalah koneksi dan komunikasi yang dilakukan dua computer dimana satu sisi bertindak sebagai klien dan sisi lain sebagai server, server hanya melayani permintaan klien dan klien mengirimkan atau meminta suatu proses pada server. Sedangkan peer to peer adalah konsep teknologi dimana antara kedua sisi tersebut menjadi kabur, satu sisi computer tersebut bisa berupa client dengan meminta suatu layanan ke server dan sisi lain computer tersebut bisa berupa server dengan menerima proses permintaan dari client.

Web dibuat dengan suatu bahasa pengkodean HTML, agar dapat interaktif maka seorang web development membuat suatu pemrograman agar dapat interaksi antara pengunjung dan situs tersebut, ada banyak bahasa yang dapat digunakan seperti ASP, PHP, Javascript, Css, XML, CMS dan lain-lain. Contohnya pada saat kita masuk ke situs tertentu, terdapat hit counter, jam, dan bukutamu, dan halaman polling. Lalu pertanyaan dasar gimana kita membuat ini, apakah bisa diwujudkan dengan hanya menggunakan

HTML, atau butuh suatu script lain untuk mewujudkannya. Jawabannya dengan kode HTML ini bisa diwujudkan dengan sangat sederhana dan terbatas kemampuannya.

2. KONSEP DASAR PEMROGRAMAN WEB

World Wide Web ("WWW", atau singkatnya "Web") adalah suatu ruang informasi dimana sumber-sumber daya yang berguna diidentifikasi oleh pengenalan global yang disebut Uniform Resource Identifier (URI). WWW sering dianggap sama dengan Internet secara keseluruhan, walaupun sebenarnya ia hanyalah bagian dari padanya.

Hiperteks dilihat dengan sebuah program bernama browser web yang mengambil informasi (disebut "dokumen" atau "halaman web") dari server web dan menampilkannya, biasanya di sebuah monitor. Kita lalu dapat mengikuti pranala di setiap halaman untuk pindah ke dokumen lain atau bahkan mengirim informasi kembali kepada server untuk berinteraksi dengannya. Ini disebut "surfing" atau "berselancar" dalam bahasa Indonesia. Halaman web biasanya diatur dalam koleksi material yang berkaitan yang disebut "situsweb".

3. BAHASA PEMROGRAMAN WEB

3.1. HTML (HyperText Markup Language)

HTML adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser Internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML, HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web dan HTML kini merupakan standar Internet yang saat ini dikendalikan oleh World Wide Web Consortium (W3C).

Versi terakhir dari HTML adalah HTML 4.01, meskipun saat ini telah berkembang XHTML yang merupakan pengembangan dari HTML. HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai

dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

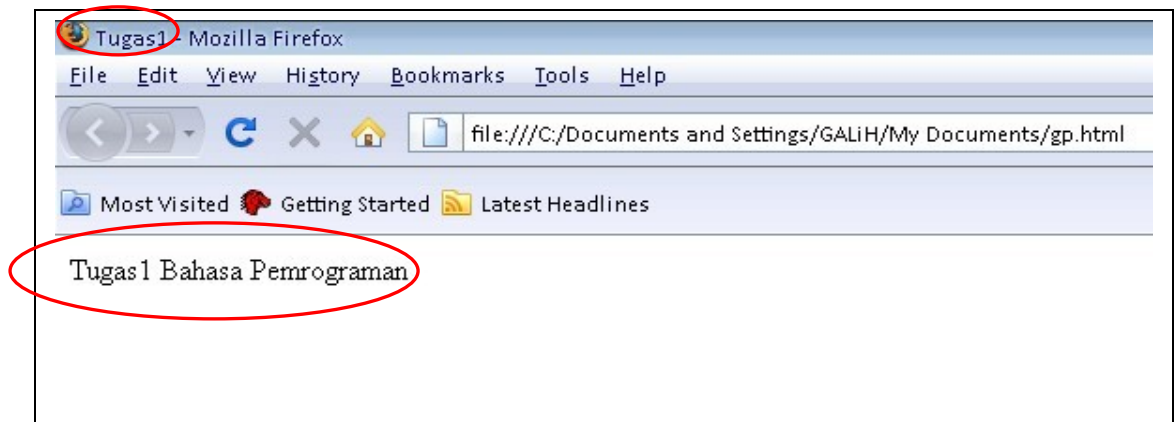
3.1.1.Struktur HTML

Secara umum dokumen web dibagi menjadi dua section (bagian), yaitu section head dan section body. Sehingga setiap dokumen HTML harus mempunyai pola dasar lengkap.

Contoh umum HTML :

```
<html>
<head>
<title>Tugas1</title>
</head>
<body> Tugas1 Bahasa Pemrograman
</body>
</html>
```

Maka hasilnya akan terlihat seperti berikut :



Keterangan :

html

merupakan tag dasar yang mendefinisikan bahwa dokumen ini adalah dokumen HTML

head

merupakan tag berikutnya setelah <html> untuk menuliskan keterangan tentang dokumen web yang akan ditampilkan

title

merupakan tag didalam head untuk memberikan judul pada caption browser web serta topik atau judul dari dokumen web yang akan ditampilkan dalam browser.

body

merupakan section utama dari web. Pada section ini semua isi dokumen yang akan ditampilkan didalam browser harus dituliskan.

3.1.2.Perintah Dasar HTML

<h1> sampai <h6> : untuk mengubah ukuran teks

<hr> membuat garis horisontal

<i> membuat teks miring

 membuat teks tebal

<u> membuat teks bergaris bawah

<center> menengahkan teks

 memasukan fungsi enter atau ganti baris kebawah

<p> untuk memisahkan paragraf

 dan <face> untuk mengubah jenis dan ukuran font

 membuat daftar list

 mendefinisikan list dalam tag

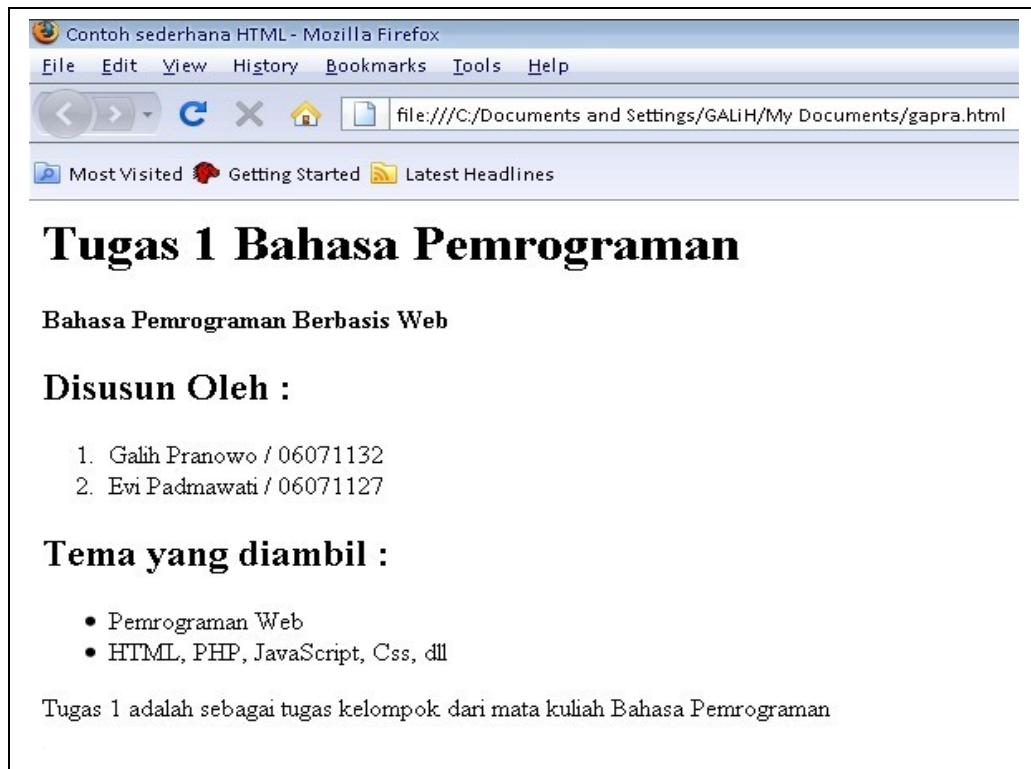
 membuat daftar list

Contoh sederhana dari perintah-perintah dasar HTML :

```
<html>
<head><title>Contoh sederhana HTML</title>
</head>

<body>
<h1 align="center">
<font face="Times New Roman">Tugas 1 Bahasa Pemrograman</h1>
</font>
<p><b>Bahasa Pemrograman Berbasis Web</p></b>
  <h2>Disusun Oleh :</h2>
    <ol>
      <li>Galih Pranowo / 06071132</li>
      <li>Evi Padmawati / 06071127</li>
    </ol>
  <h2>Tema yang diambil :</h2>
    <ul>
      <li>Pemrograman Web</li>
      <li>HTML, PHP, JavaScript, Css, dll</li>
    </ul>
  <p>Tugas 1 adalah sebagai tugas kelompok dari mata kuliah Bahasa
  Pemrograman</p>
</body>
</html>
```

Hasilnya akan terlihat seperti :



3.2.PHP (PHP: Hypertext Preprocessor)

PHP adalah sebuah script yang bersifat Server Side yang artinya semua proses akan dikerjakan disisi server dan hanya hasilnya saja yang akan dikirim ke browser. PHP pertama kali dibuat oleh Rasmus Lerdorf untuk mengetahui siapa saja yang telah mengunjungi situsnya.

Penulisan script PHP harus selalu didahului dengan tanda (<?) dan diakhiri dengan tanda (?>). Apabila tanda tersebut tidak ada maka script yang ditulis dianggap hanya sebagai HTML biasa.

Identifier dalam PHP terdiri dari fungsi, variabel, dan classes. Identifier memiliki aturan penulisan sebagai berikut :

- Harus dimulai dengan huruf atau under_score (_)
- Tidak boleh menggunakan tanda baca
- Identifier adalah case sensitive, kecuali fungsi-fungsi yang telah disediakan oleh PHP

- Variabel diawali dengan tanda dolar (\$)
- Nama fungsi yang dibuat tidak boleh sama dengan nama fungsi yang telah tersedia dalam bahasa PHP

3.2.1. Tipe Data PHP

PHP mengenal tiga maca tipe data, yaitu integer, floating point, dan string. Floating point lebih dikenal dengan double dan selalu dalam bentuk desimal. Penulisan string diawali dengan tanda petik ganda (“ ”) atau dengan petik tunggal (‘). Contoh-contoh penulisan tipe data :

Tipe Data	Contoh
Integer	\$jumlah=10; \$nilai=-5;
Double	\$skor=90.00; \$bunga=12.50;
String	\$institut="akprind"; \$jurusan="ilmu komputer";

3.2.2. Variabel

Dalam setiap bahasa pemrograman, pasti akan ditemui konsep variable. Variable adalah sebuah tempat untuk menyimpan data yang nilainya dapat berubah-ubah. Tidak seperti bahasa-bahasa pemrograman lain yang mengharuskan kita untuk mendeklarasikan variable terlebih dahulu, variabel dalam PHP tidak harus dideklarasikan sebelum variable tersebut digunakan. Untuk memberikan nilai variable digunakan tanda sama dengan (=).

Variable tempat menyimpan data di dalam PHP diawali dengan karakter \$ diikuti dengan huruf karakter pertama setelah \$, kemudia kombinasi karakter dengan angka. Tidak boleh ada spasi dan tanda baca dalam penamaannya, kecuali karakter _ (garis bawah, under score).

Contoh nama variable :

```
$nama_user
$password
```

3.2.3. Konstanta

Konstanta adalah variable yang nilainya tetap. Konstanta hanya diberi nilai pada awal program dan nilainya tidak pernah berubah selama program berjalan. PHP telah mendefinisikan beberapa konstanta, misalnya : `PHP_VERSION` yaitu konstanta yang telah memberikan informasi tentang versi PHP yang digunakan. Selain konstanta yang telah disediakan PHP kita dapat membuat konstanta sendiri dengan standar penulisannya adalah :

```
Define (“nama konstanta”, “nilai konstanta”);
```

3.2.4. Operator

Operator adalah symbol yang digunakan untuk memanipulasi data. Operator dikelompokkan dalam empat fungsi yaitu :

- a. Operator aritmatika : Operator ini merupakan operator yang berhubungan dengan fungsi matematika.
- b. Operator logika : Operator ini akan membandingkan TRUE, FALSE. Seperti bahasa C, PHP mendefinisikan FALSE dengan 0 dan TRUE dengan 1.
- c. Operator bitwise : Operator ini digunakan untuk memanipulasi bit-bit dari nilai data.
- d. Operator lain.

3.2.5. Menampilkan Data

Untuk menampilkan data ke dalam standart output dapat menggunakan perintah **print** atau **echo**. Sintaksnya adalah sebagai berikut :

```
print(data); atau print data;
```

```
echo(data; atau echo data;
```

Data disini bias data steing, numeric ataupun sebuah object. Jika data tersebut adalah data string maka harus diapit oleh tanda petik (‘) atau double petik (“).

Contoh umum dari bahasa PHP adalah sebagai berikut :

```
<html>
<head><title>Tugas 1</title>
</head>
<body>
<?php
    print "Tugas 1 Bahasa Pemrograman";
?>
<br>
<?php
$a = 10;
print "Isi dari variabel \ $a=$a";
?>
</body>
</html>
```

3.3. CSS (Cascading Style Sheets)

Cascading Style Sheets(CSS) adalah suatu teknologi yang digunakan untuk memperindah halaman website (situs), dengan CSS kita dapat dengan mudah mengubah keseluruhan warna dan tampilan yang ada disitus kita sekaligus memformat ulang situs kita.

CSS ini telah distandarkan oleh World Wide Web Consortium (W3C) untuk digunakan diweb browser.

3.3.1. KeuntunganCSS

- Dapat di-update dengan cepat dan mudah, karena kita cukup mendefinisikan sebuah style-sheet global yang berisi aturan-aturan CSS tersebut untuk diterapkan pada seluruh dokumen-dokumen HTML pada halaman situs kita.
- User yang berbeda dapat mempunyai style-sheet yang berbeda pula.
- Ukuran dan kompleksitas document code dapat diperkecil.

Sebuah style sheet terdiri dari beberapa aturan (rules). Masing-masing aturan terdiri dari satu atau lebih selektor (selector) dan sebuah blok deklarasi (declaration block). Sebuah blok deklarasi terdiri dari beberapa deklarasi yang dipisahkan oleh titik koma (;). Masing-masing deklarasi terdiri dari property, titik dua (:) dan nilai (value).

Contoh:

```
<STYLE TYPE="text/css">
  I, U { color:red }
  B { color:green; text-decoration:underline; font-family:Arial }
</STYLE>
```

3.3.2. Font Properties

Font Family

Syntax:

```
font-family: [[<family-name> | <generic-family>],]* [<family-name> | <generic-family>]
```

Possible Values:

<family-name>

- Any font family name may be used

<generic-family>

- serif(e.g., Times)
- sans-serif(e.g., Arial or Helvetica)
- cursive(e.g., Zapf-Chancery)
- monospace(e.g., Courier)

Font Style

Syntax: font-style: <value>

Possible Values: normal | italic| oblique

Font Variant

Syntax: font-variant: <value>

Possible Values: normal | small-caps

Font Weight

Syntax: font-weight: <value>

Possible Values: normal | bold| bolder| lighter | 100 | 200 | 300 | 400 | 500 | 600| 700| 800| 900

Font Size

Syntax:

font-size: <absolute-size> | <relative-size> | <length> | <percentage>

Possible Values:

- <absolute-size>
- xx-small | x-small | small | medium | large | x-large | xx-large
- <relative-size>
- larger | smaller
- <length>
- <percentage>(in relation to parent element)

Font

Syntax:

font: <value>

Possible Values: [<font-style>|| <font-variant>|| <font-weight>]? <font-size>[/<line-height>]? <font-family>

Contoh:

P { font: italic bold 12pt/14pt Times, serif }

3.3.3. Color & Background Properties

Color

Syntax: color: <color>

Nilai (value) dari color adalah sebuah keyword atau sebuah kode RGB. 16 keyword diambil dari palette Windows VGA : **aqua**, **black**, **blue**, **fuchsia**, **gray**, **green**, **lime**, **maroon**, **navy**, **olive**, **purple**, **red**, **silver**, **teal**, white, and **yellow**.

Ada empat cara dalam menuliskan warna menggunakan kode RGB:

- #rrggbb(e.g., #00cc00)
- #rgb(e.g., #0c0)
- rgb(x,x,x) dimana x adalah integer antara 0 dan 255 (e.g., rgb(0,204,0))
- rgb(y%,y%,y%) dimana y adalah nomor antara 0.0 dan 100.0 (e.g., rgb(0%,80%,0%))

Semua contoh diatas digunakan untuk menuliskan warna yang sama.

Untuk menghindari konflik dengan style sheets pengguna, properti background dan color sebaiknya ditulis bersamaan.

Background Color

Syntax: background-color: <value>

Possible Values: <color>| transparent

Background

Syntax: background: <value>

Possible Values: <background-color>|| <background-image>|| <background-repeat>|| <background-attachment>|| <background-position>

3.3.4. Text Properties

Text Alignment

Syntax: text-align: <value>

Possible Values: left | right | center | justify

3.3.5. Box Properties

Bottom Border Width

Syntax: border-bottom-width: <value>

Possible Values: thin | medium | thick | <length>

Width

Syntax: width: <value>

Possible Values: <length> | <percentage> | auto

Height

Syntax: height: <value>

Possible Values: <length> | auto

Border Style

Syntax: border-style: <value>

Possible Values: [none | dotted | dashed | solid | double | groove | ridge | inset | outset] {1,4}

3.3.6. Classification Properties

Display

Syntax: display: <value>

Possible Values: block | inline | list-item | none

Properti Display digunakan untuk mendefinisikan sebuah elemen dengan salah satu dari nilai berikut ini :

- block (a line break before and after the element)
- inline (no line break before and after the element)
- list-item (same as block except a list-item marker is added)

- none (no display)

Whitespace

Syntax: white-space: <value>

Possible Values: normal | pre | nowrap

Properti **white-space** property will determine how spaces within the element are treated. This property takes one of three values :

- normal(collapses multiple spaces into one)
- pre(does not collapse multiple spaces)
- nowrap(does not allow line wrapping without a
tag)

3.4. JavaScript

JavaScript adalah bahasa scripting yang paling populer di internet dan bekerja pada banyak browser seperti Internet Explorer, Mozilla, Firefox, Netscape, Opera. JavaScript digunakan pada Web pages untuk meningkatkan design, validate forms, detect browsers, create cookies, GUI dsb. Menggunakan sintaks C++/Java.

Apa itu JavaScript ?

- JavaScript dirancang untuk menambah interaktif HTML pages.
- JavaScript adalah bahasa scripting (bahasa scripting adalah sebuah light weight programming language).
- JavaScript terdiri dari baris-baris code executable computer.
- JavaScript biasanya embedded secara langsung pada HTML pages.
- JavaScript adalah interpreted language (artinya bahwa scripts dijalankan tanpa dikompile terlebih dahulu).
- Setiap orang dapat menggunakan JavaScript tanpa harus membeli license.

3.4.1. Keunggulan JavaScript

- JavaScript dapat bereaksi terhadap events
JavaScript dapat di-set untuk menjalankan saat terjadi sesuatu, seperti sebuah page telah selesai dipanggil atau saat seorang user meng-klik pada HTML element.
- JavaScript dapat membaca dan menulis HTML elements
JavaScript dapat membaca dan mengubah isi dari HTML element.
- JavaScript dapat digunakan untuk mem-validasi data
JavaScript dapat digunakan untuk mem-validasi form data sebelum di-submitted keserver, hal ini akan mengamankan server dar ipemrosesan extra.
- JavaScript dapat digunakan untuk mendeteksi browser pengunjung
JavaScript dapat digunakan untuk mendeteksi browser pengunjung dan memanggil page lain yang secara spesifik didesain untuk browser tersebut.
- JavaScript dapat digunakan untuk membuat cookies
JavaScript dapat digunakan untuk menyimpan dan memanggil informasi di komputer pengunjung.

3.4.2. Sintaks JavaScript

Program JavaScript dituliskan pada file HTML (.html atau.htm) menggunakan tag <SCRIPT>. Output dari program JavaScript ditampilkan secara langsung seolah-olah dimasukkan dalam HTML.

Contoh script sederhana dari javasript adalah sebagai berikut :

```

<HTML>
<HEAD>
<SCRIPT language="javascript">
<!--
// Build HTML here
<!--akhirscript -->
</SCRIPT>
</HEAD>

```

</HTML>

Contoh program JavaScript :

```
<html>
<!--COMP519 js01.html 7.09.2005 -->
<head>
  <title>JavaScript Page</title>
</head>
<body>
  <script type="text/javascript">
    // silly code to demonstrate output
    document.write("<FONT COLOR='GREEN'>
      Hello world!</FONT>");
    document.write("<p>How are <br/>" +
      "<i>you</i>?</p>");
  </script>
  <p>Here is some static text as well.
  </p>
</body>
</html>
```

- `document.write` menampilkan teks dalam page.
- Teks yang ditampilkan dapat termasuk HTML tags
- tags diterjemahkan oleh browser saat teks ditampilkan
- Seperti pada C++/Java, statement diakhiri dengan `;`
- Komentar pada Java Script sama dengan C++/Java
- `//` awal satu baris komentar
- `/*...*/` akhir dari beberapa baris komentar

3.4.3. Tipe Data dan Variabel JavaScript

JavaScript hanya mempunyai tiga tipe data primitive

String : "foo" 'howdy do' "I said 'hi'." ""

Number: 12 3.14159 1.5E6

Boolean : true false

Inisialisasi seperti pada C++/Java

```
message = "howdy";
```

```
pi = 3.14159;
```

Nama variable terdiri dari letters, digits, dan underscores : diawali dengan letter.

Nama variables adalah case-sensitive

```
<html>
<!--COMP519 js02.html 07.09.2005 -->
<head>
  <title>Data Types and Variables</title>
</head>
<body>
  <script type="text/javascript">
    varx, y;
    x= 1024;
    y=x;x = "foobar";
    document.write("<p>x = " + y + "</p>");
    document.write("<p>x = " + x + "</p>");
  </script>
</body>
</html>
```

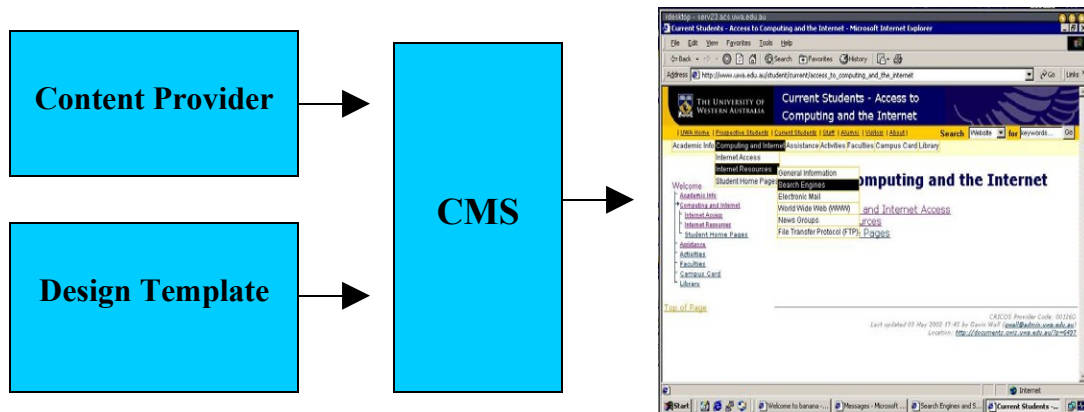
Mungkin dalam pembahasan JavaScript ini tidak selengkap mungkin, karena masih banyak sekali pembelajaran untuk mendalami JavaScript untuk kita pelajari sendiri.

3.5. CMS (Content Manajement System)

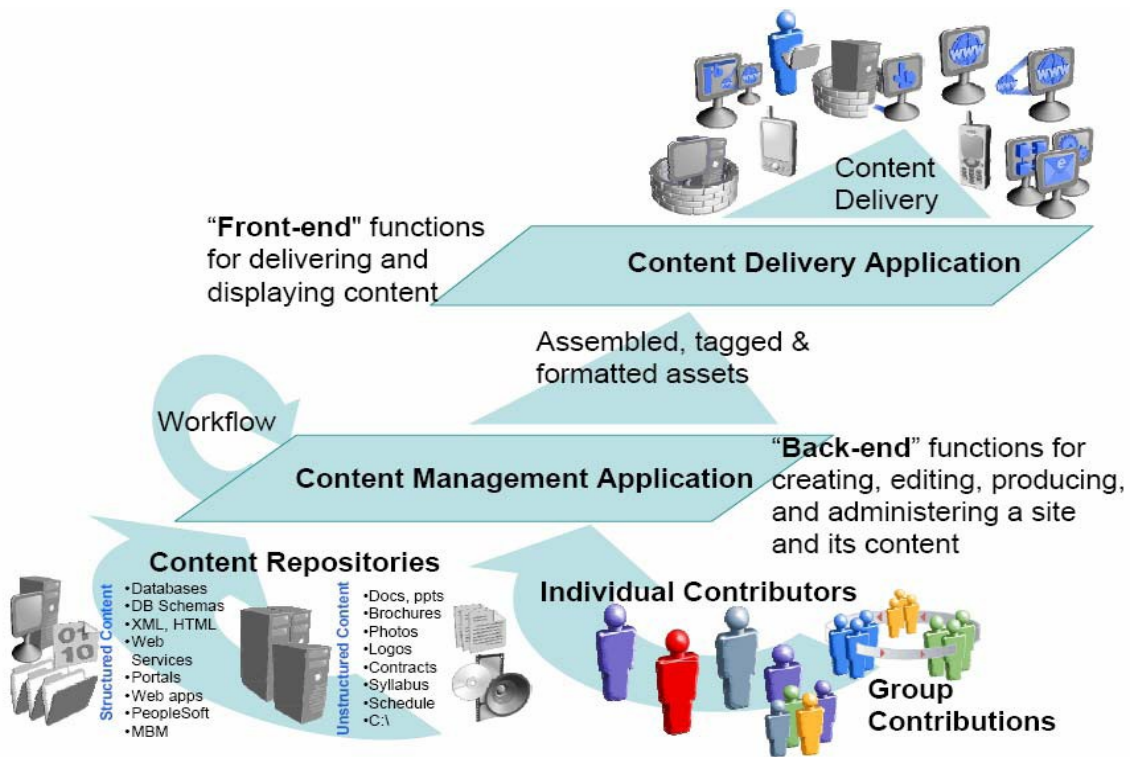
CMS adalah software yang digunakan untuk membuat, mengubah dan mempublikasikan content kedalam sebuah website. Fasilitas yang umumnya terdapat dalam CMS sangat banyak, terutama yang berkaitan dengan publikasi isi website, pengaturan halaman, pengubahan isi, pencarian dan lain-lain. Sebuah CMS, dapat berbentuk program yang sederhana, atau dapat juga merupakan suatu program kompleks yang terdiri dari berbagai modul-modul sesuai dengan fasilitas yang terdapat didalamnya.

Karena para pengelola atau pemilik website yang tidak mahir dalam menggunakan kode HTML dapat melakukan pembuatan, pengubahan dan publikasi content terhadap website-nya sendiri. CMS menyediakan framework manajemen proses yang dibutuhkan dalam pengembangan website yang menghendaki pengelolaan yang sering / dalam frekuensi yang tinggi.

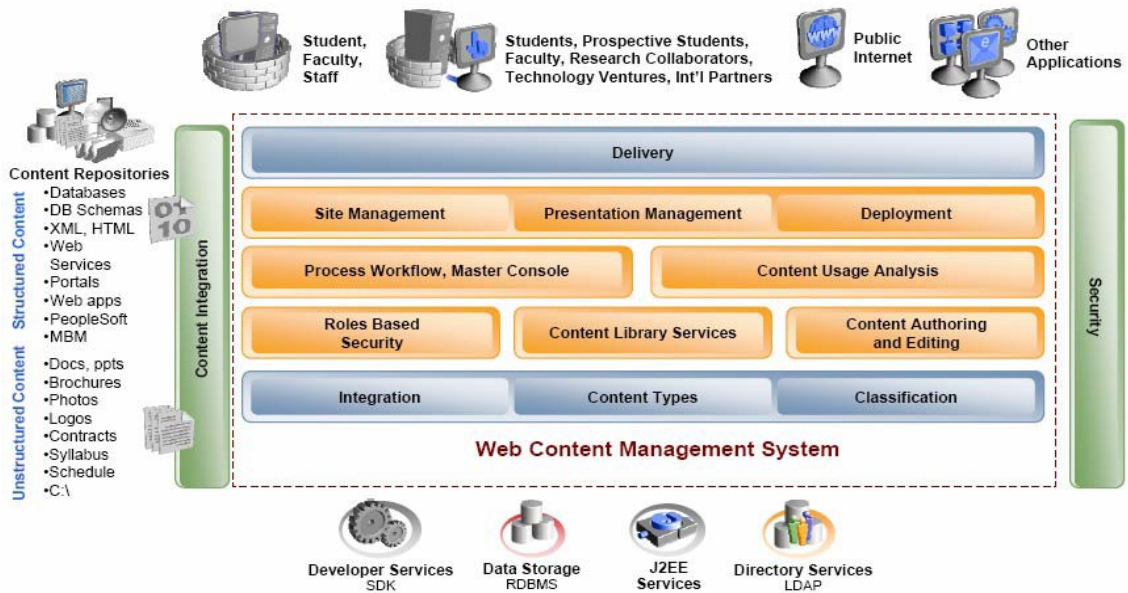
3.5.1. Content dan Design



3.5.2. CMS Basic



3.5.3. CMS Web



3.5.4. Keuntungan CMS

- Konsistensi design website dapat dijaga.
- Tidak diperlukan keahlian khusus untuk pengelolaan website.
- Content yang dikehendaki dapat dipublikasikan tanpa pengeditan oleh orang lain.
- Menghemat biaya untuk mempekerjakan web specialist.
- Notifikasi otomatis kepada pemilik website jika ada content yang sudah kadaluarsa.
- Memungkinkan kerjasama yang baik antar pengelola suatu website.
- Mengurangi kompleksitas dalam pengelolaan informasi ke website.

3.5.5. Kolaborasi Di CMS

1. Department/Division Project Manager
 - Bertanggung jawab terhadap website, pemimpin proyek, kontak personal terhadap proyek, membuat jadwal pengerjaan dll
2. Content Specialist
 - Membuat/mengedit content, reporter, interviewer, dll.
3. Technical Lead/Web Developer
 - CMS Administrator
 - Database Administrator
 - Web Developer / Designer
4. Server/CMS Administrator
 - Bertanggungjawab dalam install dan konfigurasi CMS (termasuk module), patch/update, membuat account pengguna CMS dan hak-hak-nya, memantau performance server dan CMS.
5. Database Administrator
 - Mengelola dan membuat account pengguna database, backup data, memantau performance server.
6. Web Developer / Designer

- Mengelolascripting program CMS, menambahmodule, membuatdesign template baru.

3.6. XML (Extensible Markup Language)

Extensible Markup Language (XML) adalah meta-language yang menerangkan tentang isi dari suatu document (self describing data).

Java = Portable Program

XML = Portable Data

XML tidak menerangkan tag-set atau grammar dari XML itu sendiri. Menggunakan DTD (Document Type Definition) untuk menerangkan data. XML bukan pengganti HTML.

Mengapa tidak menggunakan HTML, karena HTML tidak memberikan “informasi” tentang isi dari sebuah halaman page, tampilan tidak bisa diparsing, dan tidak bisa digunakan kembali. Terbatas hanya untuk melakukan format tampilan pada web browser.

3.6.1. Aplikasi XML

- Konfigurasi File :
 - Arsitektur J2EE
- Media untuk pertukaran data
- B2B transaksi
 - Electronics Bussines Order (ebXML)
 - Financial Exchange (IFX)
 - Messaging Exchange (SOAP)

3.6.2. Element XML

Menggunakan building block yang sama seperti HTML : Element, Attribute dan Value. Sebuah element mengandung tag pembuka dan tag penutup:

Contoh : <animal>Lion</animal>

<animal class=“mammals”>Lion</animal>,

Maka : animal adalah element, class adalah attribute dan mammals adalah nilai dari attribute.

3.6.3. Aturan XML

Dokumen XML harus “well-formed” yang artinya harus memenuhi aturan-aturan sebagai berikut :

- Harus mempunyai sebuah root element yang berisikan element-element lainnya.
- Semua elemen harus diakhiri dengan tag penutup, Contoh:

```
<picturefile="test.jpg"/>
<name>Lion</name>
```

- Element tidak boleh overlapping
- XML bersifat case sensitive
- Untuk nilai harus berada dalam kutip satu (‘) atau kutip dua (“)
- Simbol-simbol special harus dideklarasikan pada DTD

Pendeklarasian file sebagai sebuah file XML :

```
<?xml version="1.0"?>
```

Penulisan komentar pada file xml :

```
<!--Ini adalah komentar-->
```

3.6.4. DTD

DTD berfungsi untuk menjelaskan spesifikasi dan aturan terhadap element-element dan attributes yang harus dimiliki oleh sebuah xml dokument. Kumpulan dari aturan-aturan tersebut disebut dengan SCHEMA.

SCHEMA tidak wajib dimiliki, tetapi diperlukan untuk memastikan konsistensi dokumen. Mendefinisikan element pada DTD :

- <!ELEMENT animal (lion)>: element animal hanya boleh memiliki satu elemen lion.
- <!ELEMENT picture EMPTY>: element picture tidak mempunyai elemen lain.

- <!ELEMENT animal ANY>: element animal dapat berisikan element lainnya.

Mendefinisikan element yang berisikan text

```
<!ELEMENT name(#PCDATA)>
```

Mendefinisikan element yang mempunyai beberapa element

```
<!ELEMENT animal(name, weight)>
```

Berdasarkan aturan diatas maka element harus mempunyai element nama dan weight:

```
<animal>
  <name>lion</name>
  <weight>350 pounds</weight>
</animal>
```

Mendefinisikan element dengan beberapa pilihan element

```
<!ELEMENT animal ((name, weight) | (picture))>
```

Berdasarkan aturan diatas maka element animal harus mempunyai elemen name dan weight atau hanya punya element picture.

Pendefinisian Unit pada element :

```
<!ELEMENT animal (name+, weight?, picture, subspecies*)>
```

Artinya:

- name harus tampil setidaknya satu kali
- weight dapat tampil sekali atau tidak tampil sama sekali
- picture hanya harus tampil sekali
- subspecies dapat tampil beberapa kali pun atau tidak tampil sama sekali

3.6.5. Attribute

Terkadang penggunaan attribute dapat lebih berarti daripada memecah element menjadi sub-sub element. Kedua element ini berarti sama :

```
-<population animal="lion">80</population>
-<population><animal>lion</animal><quantity>80</quantity></population>
```

Attribute harus dideklarasikan juga pada DTD untuk dapat dipergunakan

–<!ELEMENT population (#PCDATA)>

<!ATTLIST population year CDATA #IMPLIED>

- Mendefinisikan Attribute wajib (Required Attribute) :

<!ELEMENT population (#PCDATA)>

<!ATTLIST population year (2000 | 2001) #REQUIRED>

Berarti: element populasi harus mempunyai attribute year yang bernilai 2000 atau 2001

<population year="2000">80</population>

- Mendefinisikan Attribute default (Default Attribute) :

<!ELEMENT population (#PCDATA)>

<!ATTLIST population year CDATA "2000">

- Mendefinisikan Attribute default (Default Attribute) :

<!ELEMENT population (#PCDATA)>

<!ATTLIST population year CDATA #FIXED "2000">

Invalid : <population year="2001">80</population>

Valid : <population year="2000">80</population>

Valid : <population>80</population>

- Mendefinisikan Unique Attribute :

<!ELEMENT animal (name)>

<!ATTLIST animal code ID #REQUIRED>

4. KESIMPULAN

Web dibuat dengan suatu bahasa pengkodean HTML, agar dapat interaktif maka seorang web development membuat suatu pemrograman agar dapat interaksi antara pengunjung dan situs tersebut, ada banyak bahasa yang dapat digunakan seperti ASP, PHP, Javascript, Css, XML, CMS dan lain-lain. Contohnya pada saat kita masuk ke situs tertentu, terdapat hit counter, jam, dan buktamu, dan halaman polling. Lalu pertanyaan dasar gimana kita membuat ini, apakah bisa diwujudkan dengan hanya menggunakan HTML, atau butuh suatu script lain untuk mewujudkannya. Jawabannya dengan kode HTML ini bisa diwujudkan dengan sangat sederhana dan terbatas kemampuannya.

DAFTAR PUSTAKA

Yahya Kurniawan. “Aplikasi Web DataBase dengan ASP”.

www.microsoft.com/net

www.ilmukomputer.com

elista.akprind.ac.id/staff/catur