



MODUL PRAKTIKUM WEB PROGRAMING TTK

OLEH :

**DILSON, S.Kom., M.Kom
NIDN. 1001067301**

**SEKOLAH TINGGI TEKNOLOGI PAYAKUMBUH
PROGRAM STUDI D.3 TEKNIK KOMPUTER
2015**

PERISTILAHAN/ *GLOSSARY*

Animation	: Animasi dari beberapa gambar diam dibuat seperti hidup tanpa ada patah-patah dalam pergerakannya.
Banner	: Merupakan kepala atau bagian atas dari sebuah web site.
Download	: Proses pengambilan file atau mengcopy file.
Effect	: Tindakan yang dikerjakan untuk menghaluskan atau memperindah animasi pada obyek.
Event	: Tindakan yang dilakukan setelah adanya action.
Export	: Merubah jenis format ke bentuk yang lain.
Format	: Bentuk ekstensi dari jenis file.
Frame	: Bagian dari Timeline.
Grouping	: Mengumpulkan beberapa obyek untuk dijadikan satu.
HTML	: Hypertext Markup Language, bahasa penanda hipertext
Insert	: Memasukkan jenis file kedalam bidang kerja SWiSHmax.
Installation manual	: Petunjuk Instalasi
Keyframe	: Pada dasarnya hampir sama dengan frame. Hanya saja untuk Key frame lebih menunjukkan untuk satu frame.
License Agreement	: Persetujuan lisensi penggunaan suatu software tertentu
Movie	: File yang dibuat dalam SWiSHmax.
Object	: Benda yang sedang dikerjakan dalam bidang kerja SWiSHmax.
Operand	: data yang dioperasikan atau dimanipulasi.
Operator	: simbol/tanda yang digunakan untuk melakukan operasi-operasi matematis atau operasi string.
Player	: Untuk memainkan animasi yang telah dibuat.
Preview	: Melihat hasil yang telah dikerjakan.
Scane	: Satu movie merupakan satu Scane.
Search Engine	: Mesin bantu pencarian data.
Server	: Pelayan, Komputer induk yang bertugas untuk melayani komputer-komputer klien.
Software	: Perangkat lunak, program yang berjalan di komputer.
Source Code	: Kode asli suatu program
Time Line	: Garis waktu atau pewaktuan yang terdapat dalam SwiSHmax.
Web design	: Pembuatan/desain halaman-halaman web.

PENDAHULUAN

TUJUAN

- 1) Mengerti fitur-fitur *software bahasa pemrograman* dan fungsinya untuk mengelola halaman web.
- 2) Mampu membuat struktur dokumen dengan tag HTML.
- 3) Mahasiswa dapat mengenali dan memahami fungsi obyek-obyek web dan tag-tag HTML dalam pembuatan halaman-halaman web yang meliputi:
 - Memformat dokumen dan menambahkan objek
 - Membuat tabel dan frame
 - Membuat dan membangkitkan *Link*
 - Merancang halaman web dinamis menggunakan Css
 - Merancang menu navigasi menggunakan Css
- 4) Mahasiswa dapat mengenali dan memahami fitur-fitur pengelolaan file halaman web.
- 5) Mahasiswa dapat melakukan kombinasi tag-tag HTML dengan obyek-obyek lain untuk membangun halaman web yang disimpan dalam format HTML.
- 6) Mahasiswa mampu menerapkan Script Javascript pada Element Form

HTML (*HYPertext MARKUP LANGUAGE*)

HTML atau *Hypertext Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web. Sebenarnya, dokumen HTML hanyalah sebuah dokumen teks biasa dan disebut sebagai *Markup Language* yakni bahasa yang mengandung tanda (*tag*) tertentu yang digunakan untuk mengatur format tampilan suatu dokumen.

HTTP atau *Hypertext Transfer Protokol* merupakan protokol yang digunakan untuk mentransfer data atau document yang berformat HTML dari web server ke browser (Internet Explorer, Netscape Navigator, NeoPlanet, dll). Dengan HTTP inilah yang memungkinkan Anda menjelajah internet dan melihat halaman web.

Untuk membuat dokumen HTML hanya dibutuhkan sebuah aplikasi teks editor biasa. Namun sekarang ini di pasaran terdapat banyak sekali HTML authoring (software yang digunakan untuk membuat atau mendesain halaman web). Macromedia Dreamweaver, Adobe GoLive, MS FrontPage sekedar contohnya. Tetapi tanpa mengetahui dasar-dasarnya Anda tidak akan memperoleh hasil yang maksimal. Mengapa? Karena walaupun software-software tersebut semakin menawarkan kemudahan dalam membuat halaman web, tetapi biasanya seseorang masih perlu untuk mengedit halaman web tersebut secara manual. Terutama untuk halaman web yang sangat kompleks.

Dalam tutorial ini Anda akan kami ajak untuk mengetahui dasar-dasar dari desain web. Yang dibutuhkan hanya sebuah word processor. Anda bisa menggunakan Notepad, WordPad, MS Word atau yang lainnya. Tapi yang paling mudah adalah menggunakan Notepad. Setelah anda memahami betul semua tag-tag dasar html, diakhir nanti kita akan belajar membuat halaman web dengan menggunakan web tool.

Struktur Dasar Html

Dalam penggunaannya, sebagian besar kode HTML harus terletak di antara tag kontainer. Yaitu diawali dengan `<namatag>` dan diakhiri dengan `</namatag>` (terdapat tanda `"/`). Struktur dasar dokumen HTML berisi elemen-elemen atau tag sebagai berikut:

```
<html>  
  <head>  
    <title>Judul Form/Caption</title>  
  </head>  
  <body>  
    ISI WEB
```

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

Keterangan:

<html> .. </html> Mendefinisikan bahwa teks yang berada diantara kedua tag tersebut adalah file HTML.

<head> .. </head> Sebagai informasi page header. Di dalam tag ini kita bisa meletakkan tag-tag TITLE, BASE, ISINDEX, LINK, SCRIPT, STYLE & META.

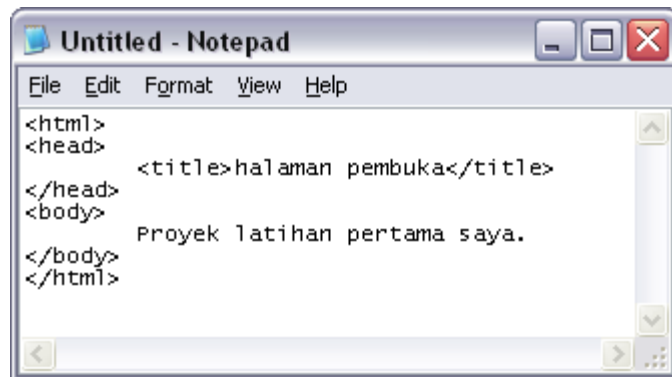
<title> .. </title> Sebagai titel atau judul halaman/form. Kalimat yang terletak di dalam tag ini akan muncul pada bagian paling atas browser Anda (pada title bar).

<body> .. </body> Mendefinisikan teks beserta formatnya yang hendak ditampilkan sebagai isi halaman web. Di dalam tag ini bisa diletakkan berbagai *page attribute* seperti *bgcolor*, *background*, *text*, *link*, *vlink*, *alink*, *leftmargin* dan *topmargin*.

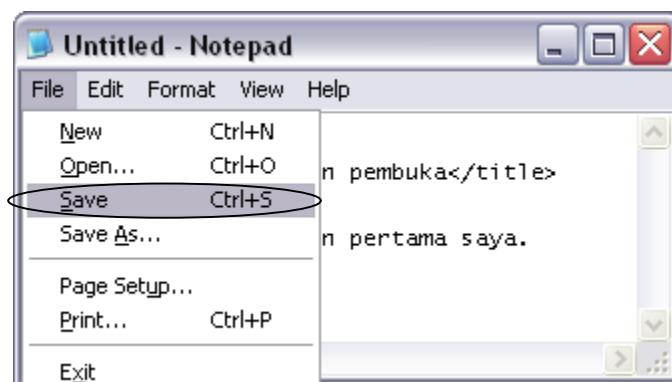
Sekarang untuk latihan pertama kita, mari kita buat dokumen HTML sederhana. Ikuti langkah-langkah berikut ini:

1. Buat direktori dengan nama latihan di drive C:. Dan selanjutnya dalam direktori ini kita akan menyimpan semua file-file latihan kita.
2. Buka *Browser*, misalnya Internet Explorer atau Netscape Navigator.
3. Buka program aplikasi teks editor misalnya dalam latihan ini adalah Notepad.
4. Mulai baris paling atas, tuliskan:

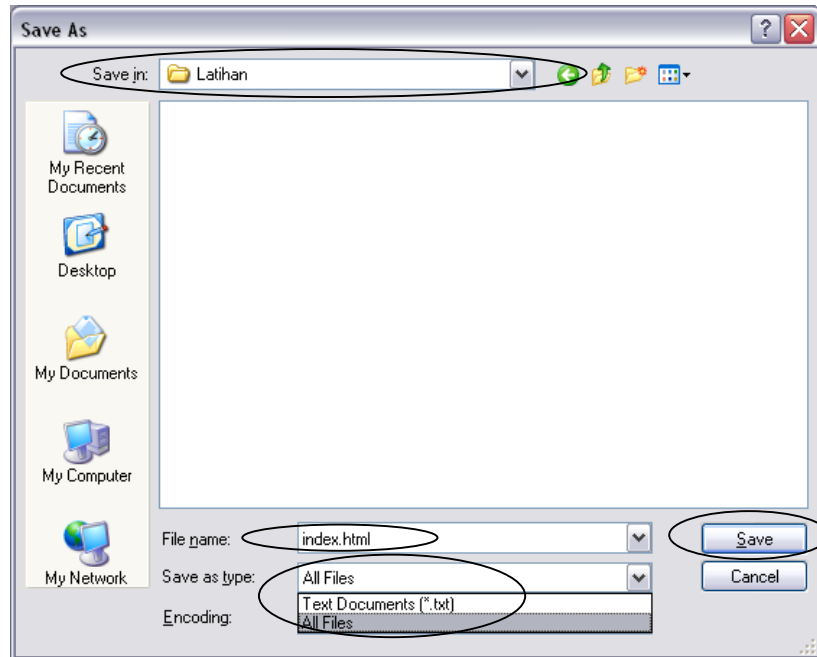
```
<html>  
  <head>  
    <title>halaman pembuka</title>  
  </head>  
  <body>  
    Proyek latihan pertama saya.  
  </body>  
</html>
```



5. Simpan file anda dengan cara klik menu File - Save:



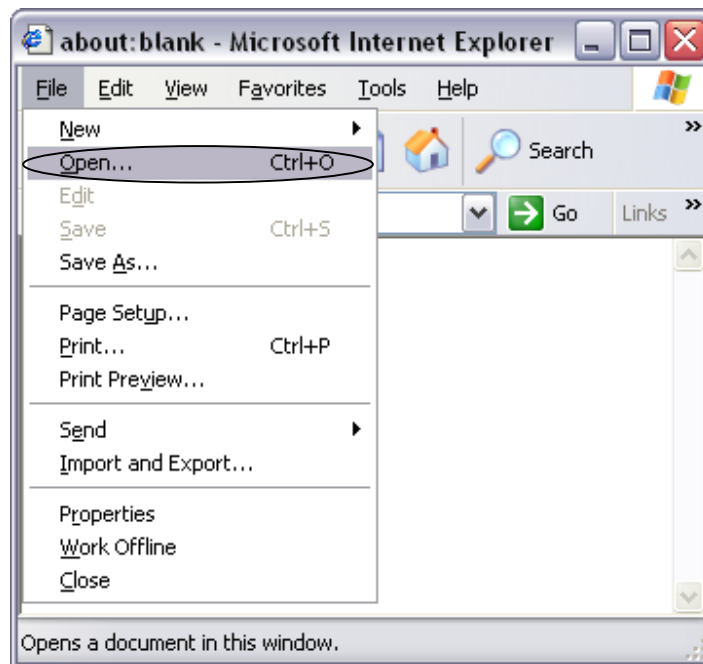
6. Selanjutnya pilih direktori latihan yang tadi kita buat.
7. Pada box File name, isikan nama filenya dengan **index.html**
8. Pada drop down list di Save as type, pilih **All Files**. Sehingga tampilannya seperti berikut:



9. Simpan proyek anda dengan meng-*klik* pada tombol **Save**

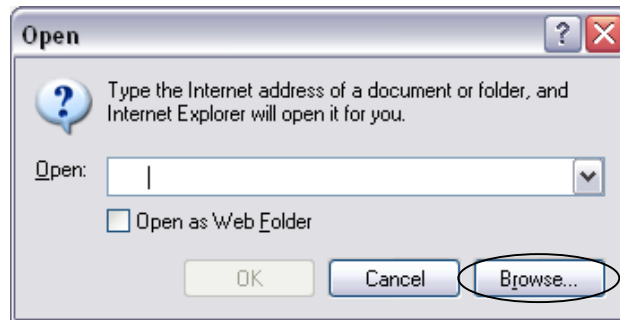
Untuk menjalankan kode-kode tersebut, silahkan buka browser seperti Internet Explorer, Netscape Navigator dan lain-lain.

1. Klik menu **File → Open**

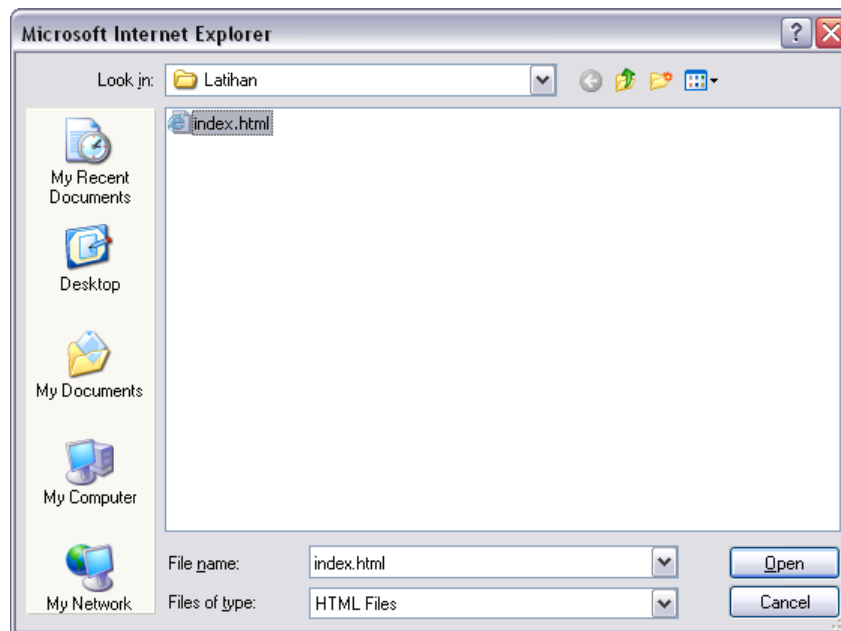


Tip: Jika anda menggunakan Netscape Navigator, klik **Open Page** kemudian **Choose File**

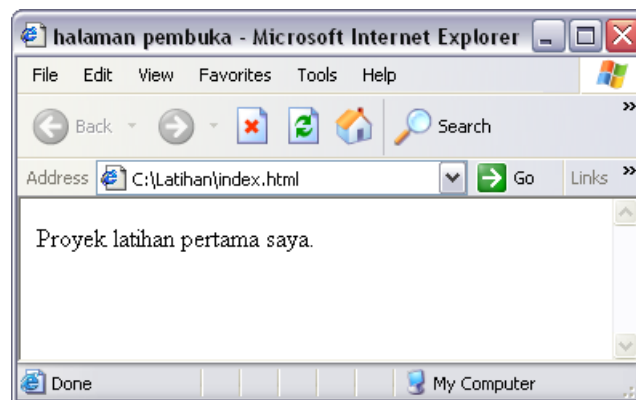
2. Setelah jendela Open terbuka, klik tombol **Browse**



3. Ketika jendela baru terbuka, pilih direktori **Latihan** dan pilih file **index.html**.



4. Klik tombol **Open** lalu tekan tombol **Ok**, halaman web yang ditampilkan dalam browser adalah seperti berikut:



Catatan:

- 📁 Semua dokumen HTML mempunyai ekstensi **.html** (atau **.htm**)
- 📁 Semua halaman Web (**homepage**) mempunyai file **index.html**. File **index.html** secara otomatis akan dipanggil ketika alamat sebuah domain atau direktori tempat menyimpan file tersebut di buka di browser. Terdapat pengecualian pada microsoft yang menggunakan Web server IIS, file yang dipanggil adalah **default.html**

Kode Warna

Pengaturan warna dalam dokumen HTML menggunakan mode kombinasi RGB (*red, green, blue*). Setiap warna ditampilkan dalam dua digit nilai heksadesimal (0, 1, 2, ..., F). Setiap bagian dua digit kode menunjukkan banyaknya intensitas dari kombinasi warna merah, hijau dan biru. Misalnya untuk warna kuning, dibuat dengan pencampuran warna sebagai berikut:

Red	Green	Blue
FF	FF	00

Berarti untuk warna kuning dapat dibuat dengan kode #ffff00. Disamping itu, pembuatan warna dapat juga dibuat dengan langsung menggunakan nama warna dalam bahasa inggris, misalnya kuning="yellow".

Berikut ini warna-warna yang dapat digunakan dalam halaman HTML.

Warna	Heksadesimal
White	#FFFFFF
Black	#000000
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Magenta	#FF00FF
Cyan	#00FFFF
Yellow	#FFFF00

Anda dapat juga membuat campuran sendiri warna-warna berdasarkan emajinasi anda. Yang penting anda mengikuti aturan diatas.

2. Pengaturan Halaman Web dan Teks

Untuk mendapatkan halaman web yang baik anda harus melakukan pengaturan terhadap halaman web dan teks-teks didalamnya seperti mengatur warna latar belakang halaman, memilih jenis dan ukuran huruf, perataan, warna teks,

menambahkan gambar dll. Tag-tag di bawah ini yang biasa digunakan dalam pengaturan halaman web dan teks:

- a. **<body>**, digunakan mendefinisikan teks beserta formatnya yang hendak ditampilkan sebagai isi halaman web. Di dalam tag ini bisa diletakkan berbagai *page attribute* seperti *bgcolor*, *background*, *text*, *link*, *vlink*, *alink*, *leftmargin* dan *topmargin*.
- 1) **Background** = digunakan untuk mengatur latar belakang dengan gambar/image.
 - 2) **Bgcolor** = digunakan untuk mengatur warna latar belakang dokumen, dengan warna putih sebagai default-nya.
 - 3) **Teks** = digunakan untuk mengatur warna teks dokumen, dengan warna hitam sebagai warna default.
 - 4) **Link** = Untuk mengatur warna link dokumen dengan warna biru sebagai warna default
 - 5) **Vlink** = Untuk mengatur warna visited link dokumen dengan default ungu
 - 6) **Alink** = digunakan untuk mengatur warna active link dokumen dengan default merah.
- b. **Heading: <Hn>..</Hn>** Digunakan untuk mengatur ukuran huruf pada header. "n" mempunyai nilai antara 1 - 6 atau antara <H1> sampai <H6>, dengan <H1> merupakan ukuran terbesar dan <H6> merupakan ukuran terkecil.

contoh1_1.html

```
<html>
<head>
    <title>::: Heading Dokumen HTML:::</title>
</head>
<body bgcolor=#ffffcc text=#003399>
    <h1>Headng Tingkat 1 </h1>
    <h2>Headng Tingkat 2 </h2>
    <h3>Headng Tingkat 3 </h3>
    <h4>Headng Tingkat 4 </h4>
    <h5>Headng Tingkat 5 </h5>
    <h6>Headng Tingkat 6 </h6>
</body>
</html>
```

Hasilnya akan terlihat seperti:



- c. **Paragraph Baru: <P>** Digunakan untuk pindah alinea atau membuat paragraf baru. Tag ini bisa diberi akhiran </P> tapi juga bisa tidak diberi. Dalam tag ini juga bisa digunakan untuk mengatur perataan kiri, tengah, kanan dan kiri-kanan, yaitu dengan attribut ALIGN. Attribut align mempunyai nilai: LEFT, RIGHT, CENTER dan JUSTIFY.
- d. **Line Break:
** Digunakan untuk pindah ke baris baru.
- e. **No Line Break: <NOBR>** Bila digunakan tag ini maka teks yang panjang tidak secara otomatis pindah baris bawahnya bila baris pertama sudah terlalu panjang.
- f. **Font ** Digunakan untuk mengatur huruf dokumen HTML. Tag FONT mempunyai attribut, yaitu: SIZE, FACE, COLOR.
 - ⇒ **SIZE:** Digunakan untuk mengatur ukuran font. Ukuran font yang digunakan berkisar antara 1 - 7 dengan 1 merupakan ukuran terkecil dan 7 merupakan ukuran terbesar.
 - ⇒ **FACE:** Digunakan untuk mengatur jenis atau nama font. Anda bisa memilih maksimal 3 jenis font yang masing-masing dipisahkan oleh koma. Bila terdapat spasi yang terletak pada nama font maka harus digunakan tanda garis bawah (_). Dalam memilih jenis font ini harus dipertimbangkan apakah font yang kita gunakan pada halaman web kita nantinya akan terdapat pada komputer pengguna yang lain (pengakses web kita). Pendeknya kita tidak usah menggunakan font-font yang bentuknya aneh-aneh, gunakan saja font

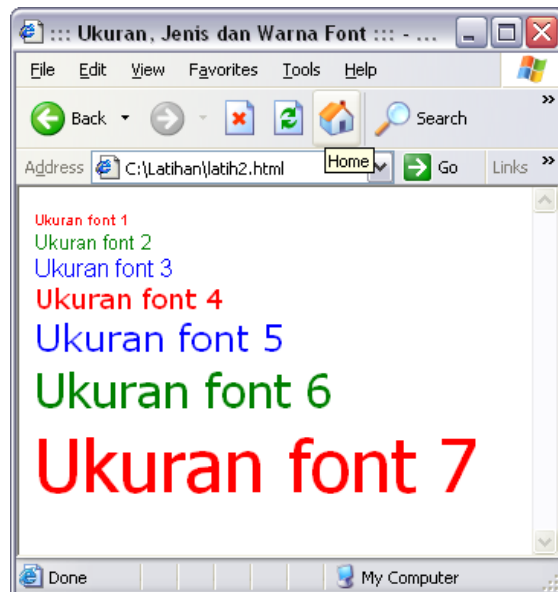
standar. Tapi bila Anda ingin menggunakan font yang sedikit "aneh" Anda bisa menggunakan graphic.

⇒ **COLOR:** Digunakan untuk mengatur warna font. Didefinisikan dengan menggunakan nilai RGB/HEX atau bisa juga langsung menggunakan nama warna (red misalnya).

contoh1_2.html

```
<html>
<head>
  <title>::: Ukuran, Jenis dan Warna Font::~:</title>
</head>
<body>
<font size=1 Face=arial color=red>Ukuran font 1</font><br>
<font size=2 Face=arial color=green>Ukuran font 2 </font><br>
<font size=3 Face=arial color=blue>Ukuran font 3 </font><br>
<font size=4 Face=verdana color=red>Ukuran font 4 </font><br>
<font size=5 Face=verdana color=blue>Ukuran font 5 </font><br>
<font size=6 Face=tahoma color=green>Ukuran font 6 </font><br>
<font size=7 Face=tahoma color=red>Ukuran font 7 </font><br>
</body>
</html>
```

Hasilnya akan terlihat



Contoh lainnya:

```
<font size=2 face="times_new_roman" color="#0066cc">
```

- g. **Base Font:** **<BASEFONT>** Digunakan untuk mendefinisikan "default text". Atribut sama dengan atribut FONT. Tag FONT akan meng*overwrite* setting pada BASEFONT.

Contoh:

```
<basefont size=2 face="arial,Helvetica" color="#ff0000">
```

Selain tag dan atribut diatas, masih terdapat lagi tag-tag yang berhubungan dengan pengaturan teks, yaitu:

Perhatian: Semua tag di bawah ini membutuhkan tap penutup.

	Bold text
<I>	<i>Italic text</i>
<U>	<u>Underscore</u>
<TT>	Typewriter
<S>	Strikeout --draws a line through the text
<PRE>	Preformatted Text <DFN> <i>Definition</i>
<BLINK>	Text berkedip (lebih baik jangan digunakan)
	Strong
<ADDRESS>	<i>Italic</i>
<CITE>	<i>Digunakan untuk quoting text</i>
<CODE>	Monospaced font (digunakan bila Anda ingin meletakkan (memperlihatkan) kode HTML pada dokumen HTML Anda)
<SAMP>	Monospaced font (digunakan bila Anda ingin meletakkan (memperlihatkan) kode HTML pada dokumen HTML Anda)
<KBD>	Monospaced font (digunakan bila Anda ingin meletakkan (memperlihatkan) kode HTML pada dokumen HTML Anda)
<BIG>	Ukuran teks akan lebih besar satu ukuran
<SMALL>	Ukuran teks akan lebih kecil satu ukuran
<SUP>	Membuat teks ^{superscript}
<SUB>	membuat teks _{sub script}
<ABBREV>	Abbreviations
<ACRONYM>	Untuk akronim
<PERSON>	Digunakan untuk indexing
<Q>	Membuat short inline quotation
<VAR>	<i>Membuat variable name, selalu dalam italics.</i>

- h. **<HR>**, digunakan untuk membuat garis horisontal. Tag ini mempunyai atribut SIZE, WIDTH, ALIGN dan NOSHADE.

Atribut **SIZE** digunakan untuk menentukan panjang garis dalam satuan pixel. Atribut **WIDTH** digunakan untuk menentukan ketebalan garis. Atribut **NOSHADE** akan menyatakan bahwa garis tersebut ditampilkan tanpa bayang-bayang 3-D.

i. **LISTS**

Daftar/list adalah merupakan kumpulan teks yang disusun sedemikian rupa sehingga item-item yang ada memiliki nomor urut atau bisa juga berupa tanda-tanda khusus/symbol.

Terdapat tiga tipe list yang dapat digunakan, yaitu :

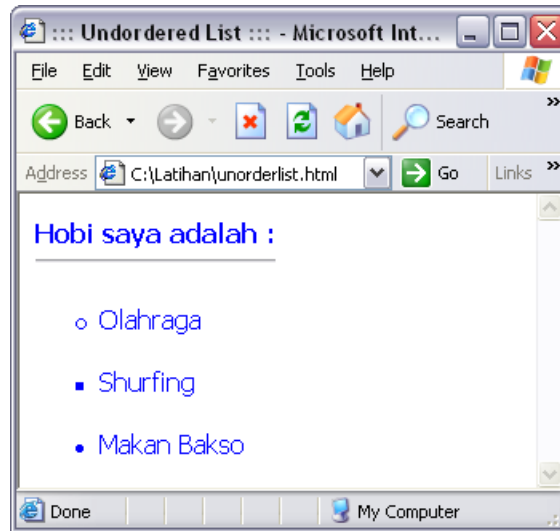
☑ **Unordered Lists: ** Untuk membuat daftar item dengan tanda bullet (tidak bernomor). List entries didefinisikan dengan tag . Pada jenis ini tidak memerlukan pengurutan data.

Bentuk item tanda pada Unordered List dapat diubah dengan menggunakan atribut **TYPE** kedalam tag dengan nilai **"circle"** untuk bentuk lingkaran tengahnya putih, **"square"** untuk bentuk kotak padat hitam dan **"disc"** bentuk lingkaran pada warna hitam.

contoh1_3.html

```
<html>
<head>
  <title>::: Undordered List:::</title>
</head>
<body>
  <b><font size=3 Face=tahoma color=blue>
  Hobi saya adalah: </b>
  <hr size=2 width=150 align=left>
  <ul type=circle>
    <li>Olahraga
  </ul>
  <ul type=square>
    <li>Shurfing
  </ul>
  <ul type=disc>
    <li>Makan Bakso
  </ul>
  </font>
</body>
</html>
```

Hasil dari kode di atas adalah:



1. , , <DL>

a.

Tag atau element HTML yang digunakan untuk membuat daftar isi

Sintak :

 Judul

 Sub Judul-1

 Sub Judul-2

Contoh :

 Daftar Hadir

 ALI

BADU

Hasilnya :

Daftar Hadir

1. ALI

2. BADU

- Untuk menentukan urutan awal List Number , kita gunakan Perintah <OL Start = [n]>, dimana [n] adalah nilai yang akan dijadikan bilangan awal

contoh :

Penomoran dimulai dari

7. Hal-hal yang dirasa perlu

8. Lain-lain

Perintah HTML nya

<OL Start = "7"> Prnomoran dimulai dari

 Hal-hal yang dirasa perlu

 Lain-lain

b.

UnOrdered List : untuk penomoran menggunakan Bullet

Contoh :

Nama Kota

- Padang
- Jakarta

HTML nya

```
<UL> nama Kota
<Li> Padang</Li>
</li>Jakarta </Li>
</UL>
```

c. <DL>

Defenition List : fungsinya untuk membuat halaman index websites

Sintak

```
<DL>
  <DT>..</DT>
  <DD>..</DD>
</DL>
```

Contoh :

Istilah Komputer

ROM

Rom adalah lokasi penyimpanan bla..bla..

Flashdisk

Flashdisk adalah media penyimpanan bla..bla..

Perintah HTMLnya :

```
<DL>Istilah Komputer
<DT><B><i>ROM</i></b></DT>
<DD> Rom adalah lokasi penyimpana bla..bla..</DD>
<DT><B><i>Flashdisk</i></b></DT>
<DD> Flashdisk adalah media penyimpanan bla..bla..</DD>
</DL>
```

Preformatted Text: <PRE>. Digunakan untuk mengatur format tampilan agar sesuai dengan aslinya.

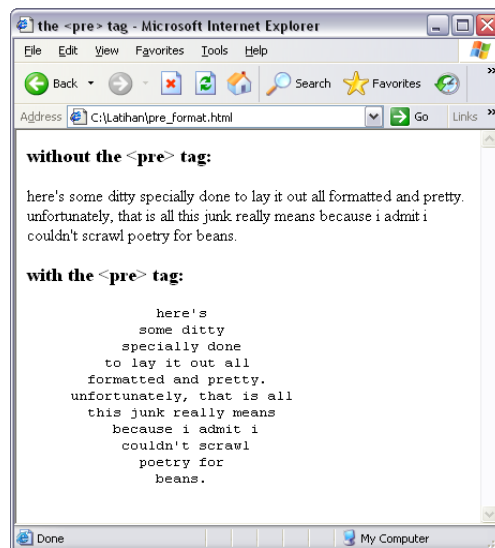
contoh1_4.html

```
<html>
<head>
<title>the &lt;pre&gt; tag</title>
</head>
<body>
<h3>without the &lt;pre&gt; tag:</h3>
  here's
  some ditty
  specially done
  to lay it out all
  formatted and pretty.
```



```
unfortunately, that is all
  this junk really means
    because i admit i
      couldn't scrawl
        poetry for
          beans.
<p><h3>with the &lt;pre> tag:</h3>
<pre>
  here's
    some ditty
      specially done
        to lay it out all
          formatted and pretty.
unfortunately, that is all
  this junk really means
    because i admit i
      couldn't scrawl
        poetry for
          beans.
</pre>
</body>
</html>
```

Hasil dari kode di atas adalah :



- ☑ **Extended Quotations: <BLOCKQUOTE>**, digunakan untuk membuat kutipan panjang, sehingga hasilnya menjorok ke dalam.

contoh1_5.html

```
<html>
<head> <title>::: Blockquote:::</title> </head>
```

```
<body>  
<h3>2. Pengaturan Teks</h3>  
<blockquote>  
Untuk mendapatkan halaman web yang baik Anda harus melakukan  
pengaturan terhadap teks seperti memilih jenis dan ukuran huruf, perataan,  
dll.  
</blockquote>  
</body>  
</html>
```

Hasilnya tampak seperti bnerikut ini:



3. Hypertext Link

Digunakan untuk membuat link/penghubung antara suatu halaman dengan halaman lain, ke URL lain, juga digunakan dalam satu halaman untuk berpindah ke sub judul yang lain atau ke lokasi tertentu dalam satu halaman yang sama.

Sintak tag link adalah sebagai berikut:

```
<a href=url_tujuan>nama_link</a>
```

url_tujuan bernilai lokasi atau nama file yang akan dituju. Sedangkan *nama_link* adalah nilai yang akan ditampilkan di browser sebagai teks link atau tombol penghubung. Tulisan yang terletak antara **<A>** dan **** akan terdapat garis bawah dan atribut warnanya dapat diatur dalam tag **<body>**. Adapun macam-macam link adalah sebagai berikut:

Link ke Dokumen Lain

Untuk membuat link/penghubung dengan target ke dokumen/halaman yang berbeda, anda harus membuat dokumen yang dituju/target dan disimpan dalam direktori tertentu, baik pada direktori yang sama atau berbeda dengan dokumen yang aktif sekarang. Untuk mencoba membuat link, silahkan buat script berikut dan beri nama dengan *link_dokumen_lain.html*.

```
<html>
```

```
<head>
  <title>::: Latihan Membuat Link Ke Dokumen Lain::~ </title>
</head>
<body bgcolor=#ffffcc>
<font face=arial size=6 color=maroon>
<b>.::: STT PAYAKUMBUH .:.. <br>
<font face=arial size=4 color=blue>
<i>Success By Discipline</i>
<font face=arial size=1 color=maroon>
<hr>
| <a href=link_dokumen_lain.html>H o m e</a>
| <a href=proli.html>Program Keahlian</a>
| <a href=ekstra.html>Ekstra Kurikuler</a> |
<hr><p><font face=verdana size=4 color=purple>
Selamat datang di web kami ...
</body>
</html>
```

Buat lagi file berikut dan beri nama dengan ***proli.html***

```
<html>
<head>
  <title>::: Link Dokumen Lain::~ </title>
</head>
<body bgcolor=#ffffcc>
<font face=arial size=6 color=maroon>
<b>.::: STT PAYAKUMBUH .:.. <br>
<font face=arial size=4 color=blue>
<i>Success By Discipline</i>
<font face=arial size=1 color=maroon>
<hr>
| <a href=link_dokumen_lain.html>H o m e</a>
| <a href=proli.html>Program Keahlian</a>
| <a href=ekstra.html>Ekstra Kurikuler</a> |
<hr><p><font face=verdana size=4 color=blue>
Program Keahlian:<br>
<font size=1>
<ul type=circle><li>Rekayasa Perangkat Lunak
  <li>Teknik Elektronika Industri
  <li>Teknik Pembangkit Tenaga Listrik
  <li>Teknik Mekanik Otomotif
  <li>Teknik Bodi Otomotif
  <li>Teknik Pemesinan
  <li>Teknik Las</ul>
</body>
</html>
```

Link ke bagian tertentu dalam dokumen yang sama

Untuk membuat link ke bagian tertentu dalam dokumen yang sama, harus diberikan nama **anchor**. Perbedaan dengan link yang berbeda dokumen adalah tujuan/target link, jika dalam dokumen yang sama yang dituju adalah nama anchor tetapi kalau dokumen yang beda yang dituju adalah nama file/dokumen yang bersangkutan.

Untuk memberi nama anchor, atribut yang dipakai adalah **NAME** pada tag anchor <A> dan untuk membuat link cukup memberikan tanda # setelah nama file dalam URL. Misal:

```
<a href="#php"> Bab 1 </a>
```

atau dapat ditulis lengkap:

```
<a href="belajar_php.html#php"> Bab 1 </a>
```

dan untuk nama anchornya:

```
<a name="php">Ada apa dengan PHP?</a>
```

Link ke alamat URL atau Website

Untuk membuat link ke alamat URL adalah dengan menambahkan:

http://nama_URL.

Contoh:

```
<a href="http://www.dikti.org">www.dikti.org</a>
```

```
<a href="http://www.sttppyk-mlg.sch.id">www.sttppyk-mlg.sch.id </a>
```

Link ke Alamat Email

Link email adalah membuat link pada teks, jika teks tersebut diklik maka akan menampilkan program pengiriman email yang terdapat pada komputer yang dipakai untuk mengakses data atau teks tersebut secara otomatis.

Untuk membuat link ke alamat email hanya menambahkan atribut **mailto:alamat_email** ke dalam tag <A HREF>

Misalnya:

```
<A HREF="mailto:webmaster@sttppyk-mlg.sch.id">Kirim email</a>
```

Link File yang akan didownload

Link file dalam hal ini adalah apabila diklik pada teks tersebut maka proses selanjutnya adalah mendownload file yang tercantum pada dokumen tersebut. Format penulisannya adalah

Misal:

```
<a href="antivirus.zip">Download anti virus</a>
```

```
<a href="latih.doc">Download latihan Html</a>
```

`Download MYSQL`

4. Menyisipkan Gambar/Images

Digunakan untuk menampilkan image atau animasi gif pada halaman web Anda. Tag yang digunakan adalah
Bentuk penulisan :

```
<Image SRC="Lokasi Gambar" ALT="Keterangan Gambar" Width="Lebar"  
Height="Tinggi">
```

Contoh :

```
<Image SRC="D:/Images/Gambar.jpg" ALT="About Me" Width="80" Height="60">
```

Attribut yang dimiliki tag `` adalah: `alt`, `align`=(center, left, right), `hspace`, `vspace`, `border`, `width` dan `height`. Jika attribut ukuran gambar tidak dituliskan, maka gambar ditampilkan sesuai dengan ukuran aslinya. Untuk mengatur ukuran gambar, menggunakan attribut `width` dan `height`. Attribut `align` digunakan untuk perataan posisi gambar. Attribut `border`, digunakan untuk memberi bingkai pada gambar. Sedangkan `alt`, digunakan untuk memberi keterangan pada gambar jika mouse berada di atasnya. Untuk mengatur letak gambar dapat menggunakan attribut `hspace` dan `vspace`.

contoh1_6.html

```
<html>  
<head>  
  <title>::: menampilkan images:::</title>  
</head>  
<body>  
<b><font size=2 Face=tahoma color=black>  
  
  
</font>  
</body>  
</html>
```

Tampilan dari contoh diatas adalah sebagai berikut:

5. Layout Halaman Web dengan Tabel (Table)

Table merupakan cara untuk menampilkan informasi dalam halaman web dengan bentuk kolom dan baris. Pada dasarnya Tabel terdiri dari beberapa Baris (Rows) Dan Baris terdiri dari beberapa kolom (COLUMNS)

Membuat Tabel

Terdapat tiga tag atau elemen utama yang digunakan dalam pembuatan table, yaitu: <TABLE>, <TR>, dan <TD>. Yang perlu diingat adalah bahwa tag <TR> dan <TD> harus terletak di antara tag <TABLE> dan </TABLE> .

<TABLE>

Digunakan untuk mendefinisikan pembuatan tabel. Memiliki atribut:

- **align** - perataan: rata kiri (left), tengah (center) atau kanan (right).
- **valign** – mengatur bentuk perataan secara vertikal
- **bgcolor** – mengatur warna latar belakang (background) dari tabel.
- **background** – menentukan gambar yang digunakan sebagai background tabel
- **color** – Untuk mengatur warna suatu sel dalam tabel
- **border** – menentukan ukuran border tabel (dalam pixel).
- **rowspan** – menggabungkan beberapa baris
- **colspan** – menggabungkan beberapa kolom
- **cellpadding** - jarak antara isi cell dengan batas cell (dalam pixel).
- **cellspacing** – mengatur spasi/jarak antar cell (dalam pixel).
- **width** – menentukan lebar tabel dalam pixel atau percent.
- **height** – Menentukan tinggi tabel

<TR>

Tag ini digunakan untuk membuat baris baru (pada tabel). Terdiri dari atribut:

- **align** - perataan: rata kiri (left), tengah (center) atau kanan (right).
- **bgcolor** - warna latar belakang dari baris.
- **valign** - perataan vertikal: top, middle atau bottom.

<TD>

Tag ini digunakan untuk membuat kolom baru pada tabel. Atributnya adalah:

- **align** – untuk menentukan perataan kolom
- **background** – untuk menentukan image yang digunakan sebagai latar belakang dari kolom.
- **bgcolor** – untuk menentukan warna latar belakang
- **colspan** - lihat gambar contoh
- **height** – untuk mengatur ukuran tinggi cell dalam pixels.

- **nowrap** – untuk membuat supaya isi dari kolom tetap berada pada satu baris.
- **rowspan** - lihat gambar contoh
- **valign** – untuk mengatur perataan vertikal: top, middle atau bottom.
- **width** – untuk menentukan lebar kolom dalam pixel atau persen.

contoh1_7.html

```
<html>
<head>
  <title>::: Pembuatan Table::~:</title>
</head>
<body>
<font face=arial size=2 color=maroon>
<table>
  <tr>
    <td>Ini contoh tabel sederhana tanpa border</td>
  </tr>
</table>
<p>
<table border=1>
  <tr>
    <td>Ini contoh tabel sederhana dengan border</td>
  </tr>
</table>
<p>
<table border=1>
  <tr>
    <td>No.</td>
    <td>Nama</td>
    <td>Alamat</td>
  </tr>
  <tr>
    <td>1.</td>
    <td>Jam 'Iyatul Khoir</td>
    <td>Payakumbuh</td>
  </tr>
  <tr>
    <td>2.</td>
    <td>Ari Mulyaningsih</td>
    <td>Padang </td>
  </tr>
</table>
</body>
</html>
```

Hasilnya adalah sebagai berikut:

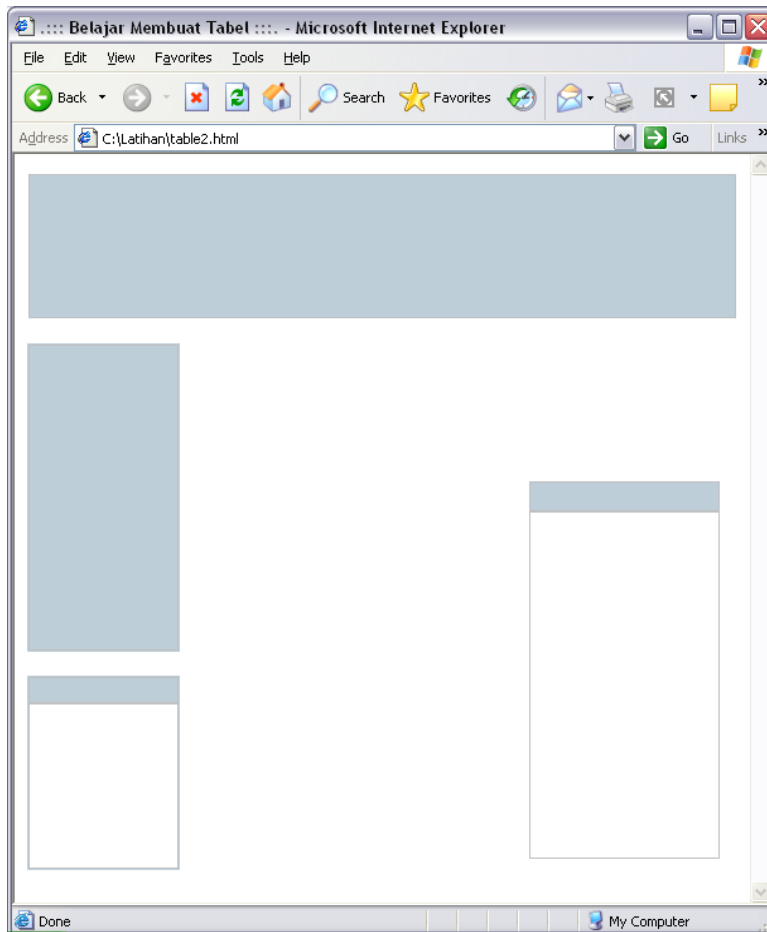
contoh1_8.html

```
<html>
<head>
  <title>...: Belajar Membuat Tabel...</title>
</head>
<body>
<table border="1" cellpadding="0" cellspacing="0" bordercolor="#ffffff"
  width="100%" height="115">
  <tr>
    <td width="100%" height="115" bordercolor="#C0C0C0"
      bgcolor="#BDCED9">&nbsp;</td>
  </tr>
</table>
<font size="1">&nbsp;</font>
<table border="0" cellpadding="0" cellspacing="0" bordercolor="#111111"
  width="100%" height="431">
  <tr>
    <td width="23%" height="307" valign="top">
      <table border="1" cellpadding="0" cellspacing="0"
        bordercolor="#BDCED9" width="94%" height="245">
        <tr>
          <td width="100%" height="244"
            bordercolor="#C0C0C0" bgcolor="#BDCED9">&nbsp;</td>
        </tr>
      </table>
      <font size="1">&nbsp;</font>
      <table border="1" cellpadding="0" cellspacing="0"
        bordercolor="#BDCED9" width="94%" height="152">
        <tr>
          <td width="100%" height="19" bgcolor="#BDCED9"
            bordercolor="#C0C0C0">&nbsp;</td>
        </tr>
        <tr>
          <td width="100%" height="132" bordercolor="#C0C0C0">
            &nbsp;</td>
        </tr>
      </table>
    </td>
    <td width="77%" height="307" valign="top">
      <table border="1" cellpadding="0" cellspacing="0" bordercolor="#ffffff"
        width="97%" height="411">
        <tr>
          <td width="64%" height="109">&nbsp;</td>
          <td width="36%" height="109">&nbsp;</td>
        </tr>
        <tr>
          <td width="64%" height="301" rowspan="2">&nbsp;</td>
          <td width="36%" height="24" bgcolor="#BDCED9"
            bordercolor="#C0C0C0">
            <font size="1">&nbsp;</font></td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```



```
<tr>
  <td width="36%" height="277" bordercolor="#C0C0C0">&nbsp;
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

Hasilnya adalah:



Menambahkan Judul Tabel

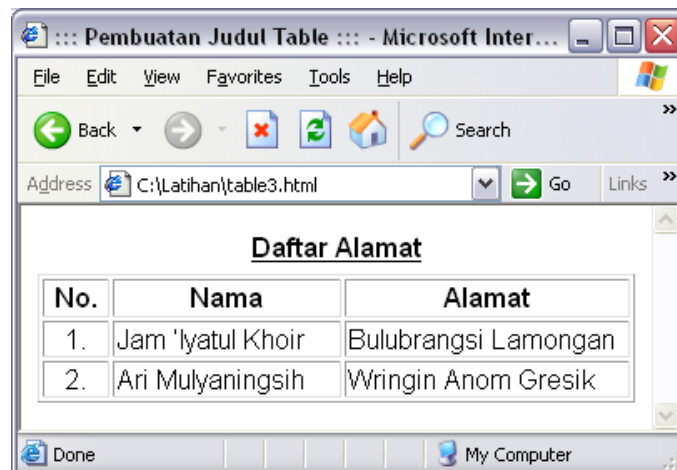
Judul dalam tabel dibedakan menjadi tiga macam, yaitu judul tabel, judul kolom table dan judul baris tabel. Judul tabel atau biasa disebut **CAPTION** terletak dibagian luar tabel yang bisa berada dibawah atau diatas tabel. Secara default *caption* diletakkan dibagian atas suatu tabel, tetapi juga dapat diletakkan dibawah suatu tabel dengan menambahkan atribut **ALIGN=BOTTOM** pada elemen *caption* tersebut.

Judul kolom atau judul baris dibuat dengan elemen yang sama yaitu elemen **TABLE HEADER <TH>**. Judul kolom terletak pada sel disebelah kiri atau kolom pertama suatu tabel, sedangkan judul baris terletak pada baris pertama suatu tabel. Judul baris atau judul kolom akan tercetak tebal.

contoh1_9.html

```
<html>
<head>
  <title>::: Pembuatan Judul Table::~:</title>
</head>
<body>
<font face=arial size=2 color=maroon>
<table border=1>
  <caption align=top><b><u>Daftar Alamat</u></b></caption>
  <tr>
    <th width=40>No.</th>
    <th width=150>Nama</th>
    <th width=200>Alamat</th>
  </tr>
  <tr>
    <td align=center>1.</td>
    <td>Jam 'Iyatul Khoir</td>
    <td>Bulubrangsi Lamongan</td>
  </tr>
  <tr>
    <td align=center>2.</td>
    <td>Ari Mulyaningsih</td>
    <td>Wringin Anom Gresik</td>
  </tr>
</table>
</body>
</html>
```

Hasilnya akan tampak sebagai berikut:



contoh1_10.html

```
<html>
<head>
  <title>::: Pembuatan Judul Table::~:</title>
</head>
<body>
<font face=arial size=2 color=maroon>
<table border=1>
  <caption align=bottom><b><u>Daftar Alamat</u></b></caption>
  <tr>
    <th width=60 align=left>No.</th>
    <td width=150>1.</td>
    <td width=200>2.</td>
  </tr>
  <tr>
    <th align=left>Nama</th>
    <td>Jam 'Iyatul Khoir</td>
    <td>Ari Mulyaningsih</td>
  </tr>
  <tr>
    <th align=left>Alamat</th>
    <td>Bulubrangsi Lamongan</td>
    <td>Wringin Anom Gresik</td>
  </tr>
</table>
</body>
</html>
```

Hasilnya akan tampak seperti berikut:



Mengatur Lebar dan Tinggi Suatu Tabel

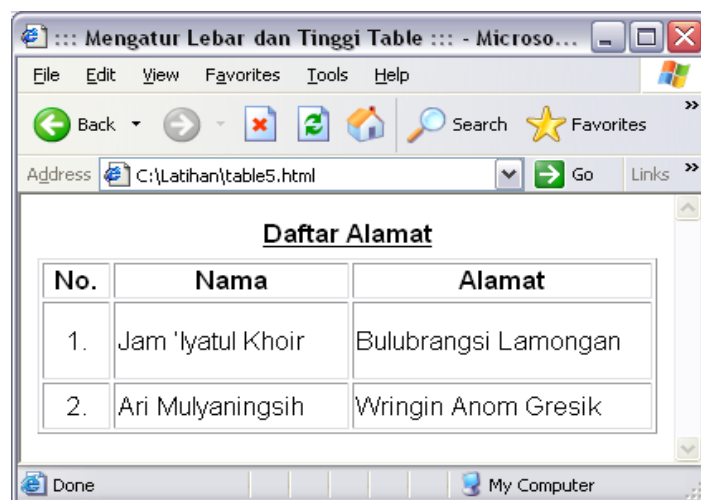
Lebar tabel diatur dengan menggunakan atribut **WIDTH** dan untuk mengatur tinggi table dengan atribut **HEIGHT**. Jika atribut WIDTH dan HEIGHT digunakan dalam elemen TABLE nilainya menunjukkan lebar dan tinggi tabel terhadap BROWSER sedangkan jika digunakan pada elemen TH dan TD, nilainya merupakan lebar dan

tinggi dari suatu tabel. Nilai atribut menggunakan ukuran % (max 100%) dan ukuran pixel.

contoh1_11.html

```
<html>
<head>
<title>::: Mengatur Lebar dan Tinggi Table::: </title>
</head>
<body>
<font face=arial size=2 color=maroon>
<table border=1 width=100%>
  <caption align=top><b><u>Daftar Alamat</u></b></caption>
  <tr>
    <th width=40>No.</th>
    <th width=150>Nama</th>
    <th width=200>Alamat</th>
  </tr>
  <tr>
    <td align=center>1.</td>
    <td height=50>Jam 'Iyatul Khoir</td>
    <td height=50>Bulubrangsi Lamongan</td>
  </tr>
  <tr>
    <td align=center>2.</td>
    <td height=30>Ari Mulyaningsih</td>
    <td height=30>Wringin Anom Gresik</td>
  </tr>
</table>
</body>
</html>
```

Hasilnya tampak sebagai berikut:



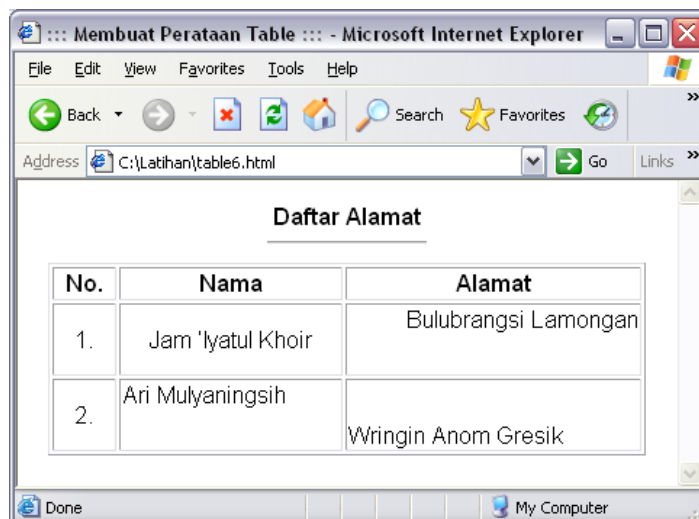
Perataan dalam Tabel

Perataan dalam tabel dikenal dua macam, yaitu perataan secara horizontal dengan atribut ALIGN dan perataan vertikal dengan atribut VALIGN, serta untuk membuat suatu tabel posisinya menjadi ditengah-tengah layar browser.

contoh1_12.html

```
<html>
<head>
  <title>::: Membuat Perataan Table:::</title>
</head>
<body>
<font face=arial size=2 color=maroon>
<table border=1 align=center>
  <caption align=top>
    <b>Daftar Alamat</b>
    <hr width=110>
  </caption>
  <tr>
    <th width=40>No.</th>
    <th width=150>Nama</th>
    <th width=200>Alamat</th>
  </tr>
  <tr>
    <td align=center>1.</td>
    <td align=center valign=middle height=50>Jam 'Iyatul Khoir</td>
    <td align=right valign=top height=50>Bulubrangsi Lamongan</td>
  </tr>
  <tr>
    <td align=center>2.</td>
    <td align=left valign=baseline height=50>Ari Mulyaningsih</td>
    <td align=left valign=bottom height=50>Wringin Anom Gresik</td>
  </tr>
</table>
</body>
</html>
```

Hasilnya seperti gambar berikut:



Membuat Warna Pada Tabel

Didalam pewarnaan tabel kita dapat mengatur warna latar belakang dan teks serta warna bordernya. Untuk menentukan warna latar belakang pada suatu tabel, gunakan atribut BGCOLOR sedangkan untuk warna border dapat menggunakan beberapa atribut sbb:

Atribut	Fungsi
BORDERCOLOR	Mengubah warna keseluruhan border
BORDERCOLORLIGHT	Mengubah warna border bagian atas dan kiri
BORDERCOLORDARK	Mengubah warna border bagian bawah dan kanan

Penggabungan Baris/Kolom

Untuk menggabungkan baris/kolom (*merge*) digunakan atribut COLSPAN dan ROWSPAN. COLSPAN digunakan untuk menggabungkan beberapa kolom menjadi 1 sedangkan ROWSPAN menggabungkan beberapa baris menjadi 1.

Cellpadding Dan Cellspacing

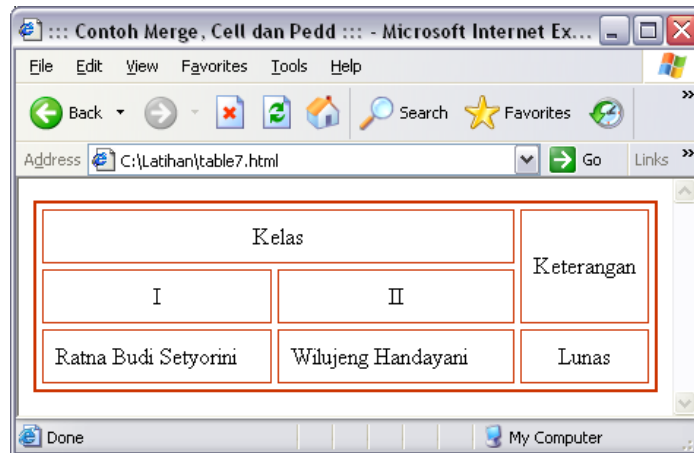
CELLPADDING digunakan untuk mengatur spasi antara border dengan tulisan, sedangkan CELLSPACING digunakan untuk mengatur spasi antar dua buah sel.

contoh1_13.html

```
<html>
<head>
  <title>::: Contoh Merge, Cell dan Pedd::~:/title>
</head>
<body>
<table border=2 align=center bordercolor=#EFFBFC bgcolor=#cc3300 cellspacing=4
cellpadding=8 width=100%>
<font face=arial size=2 color=white>
  <tr>
    <td bgcolor=#DFF9F9 colspan=2 align=center>Kelas</td>
    <td bgcolor=#DFF9F9 align=center rowspan=2>Keterangan</td>
  </tr>
  <tr>
    <td width=200 bgcolor=#DFF9F9 align=center>I</td>
    <td width=200 bgcolor=#DFF9F9 align=center>II</td>
  </tr>
  <tr>
    <td>Ratna Budi Setyorini</td>
    <td>Wilujeng Handayani</td>
    <td align=center>Lunas</td>
  </tr>
</table>
</body>
</html>
```

</html>

Hasilnya adalah sebagai berikut:



KESIMPULAN :

<TABEL> Teridir dari beberapa <TR> atau baris
 <TR>atau baris terdiri dari beberapa <TD> atau kolom

Bentuk penulisan :

```
<TABLE BORDER="0|1" WIDTH="n%|npx">
  <TR> <TD> </TD> </TR>
</TABLE>
```

Contoh :

	Kolom (C)		
Baris (B)	B1,C1	B1,C2	B1,C3
	B2,C1	B2,C2	B2,C3
	B3,C1	B3,C2	B3,C3

Contoh Kasus :

MENGGABUNG (MERGE) KOLOM 2 & KOLOM 3 PADA BARIS KE 2,

Tabel 2

	Kolom (C)		
Baris (B)	B1,C1	B1,C2	B1,C3
	B2,C1	B2,K2	
	B3,C1	B3,C2	B3,C3

Dimana : B1=Baris ke 1, C1=Kolom ke1 dst

Untuk menjawab Soal Tabel2

```
<html><head></head><title>Advance HTML</title>
```

```
<body>
```

```
<Table border="1">
```

```
<TR><TD>B1,C1</TD><TD>B1,C2</TD><TD>B1,C3</TD></TR>
```

```
<TR><TD>B2,C1</TD><TD>B2,C2</TD><TD>B2,C3</TD></TR>  
<TR><TD>B3,C1</TD><TD COLSPAN="2">B3,C2</TD></TR>  
</Table>
```

6. Membuat Frames

Frame HTML dapat digunakan untuk membuat tampilan halaman HTML yang terbagi-bagi menjadi beberapa dokumen HTML, dimana setiap bagian merupakan satu halaman HTML terpisah. Sehingga tampilan halaman HTML yang salah satu atau beberapa bagian berganti-ganti isinya sedangkan bagian lain tetap sehingga dapat menghemat bandwidth internet dan mempercepat proses download secara keseluruhan.

Format:

```
<html>  
<head>  
</head>  
<frameset>  
    <frame src>  
</frameset>  
</html>
```

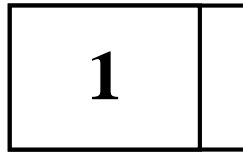
Aturan penggunaan atribut ROWS dan COLS pada frameset adalah:

```
<frameset rows="tinggi_baris,tinggi_baris,...">  
<frameset cols="lebar_kolom, lebar_kolom,...">
```

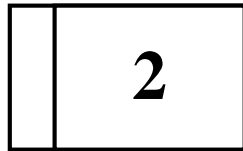
Atribut-atribut yang digunakan dalam FRAME adalah:

Atribut	Fungsi
FRAMESET COLS	Membuat frame vertikal dengan lebar kolom tertentu
FRAMESET ROWS	Membuat frame horizontal dengan tinggi baris tertentu
FRAME SRC	Memasukkan dokumen HTML ke dalam FRAME
NOFRAME	Memasukkan body teks untuk browser yang tidak dapat menampilkan frame

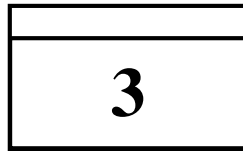
Model-model frame dan contoh pembuatannya:



1
<FRAMESET cols="*,140">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAME SRC="menu.htm"
NAME="Frame2">
</FRAMESET>



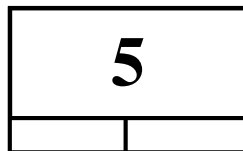
2
<FRAMESET cols="100,*">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAME SRC="menu.htm"
NAME="Frame2">
</FRAMESET>



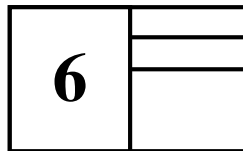
3
<FRAMESET rows="100,*">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAME SRC="menu.htm"
NAME="Frame2">
</FRAMESET>



4
<FRAMESET rows="*,60">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAME SRC="menu.htm"
NAME="Frame2">
</FRAMESET>

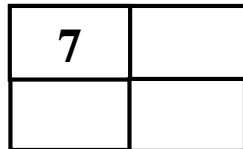


5
<FRAMESET rows="*,60">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAMESET cols="50%,50%">
<FRAME SRC="menu.htm"
NAME="Frame2">
<FRAME SRC="menu2.htm"
NAME="Frame3">
</FRAMESET>
</FRAMESET>



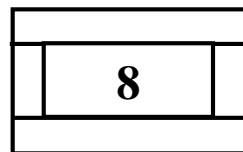
6

```
<FRAMESET cols="*,50%">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAMESET rows="15%,15%,70%">
<FRAME SRC="menu.htm"
NAME="Frame2">
<FRAME SRC="menu2.htm"
NAME="Frame3">
<FRAME SRC="menu3.htm"
NAME="Frame4">
</FRAMESET>
</FRAMESET>
```



7

```
<FRAMESET cols="50%,50%">
<FRAMESET rows="50%,50%">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAME SRC="homepage2.htm"
NAME="Frame2">
</FRAMESET>
<FRAMESET rows="50%,50%">
<FRAME SRC="menu.htm"
NAME="Frame3">
<FRAME SRC="menu2.htm"
NAME="Frame4">
</FRAMESET>
</FRAMESET>
```



8

```
<FRAMESET rows="15%,70%,15%">
<FRAME SRC="homepage.htm"
NAME="Frame1">
<FRAMESET cols="15%,70%,15%">
<FRAME SRC="menu.htm"
NAME="Frame2">
<FRAME SRC="menu2.htm"
NAME="Frame3">
<FRAME SRC="menu3.htm"
NAME="Frame4">
</FRAMESET>
<FRAME SRC="homepage.htm"
NAME="BIG">
</FRAMESET>
```

Latihan Menggunakan FRAME

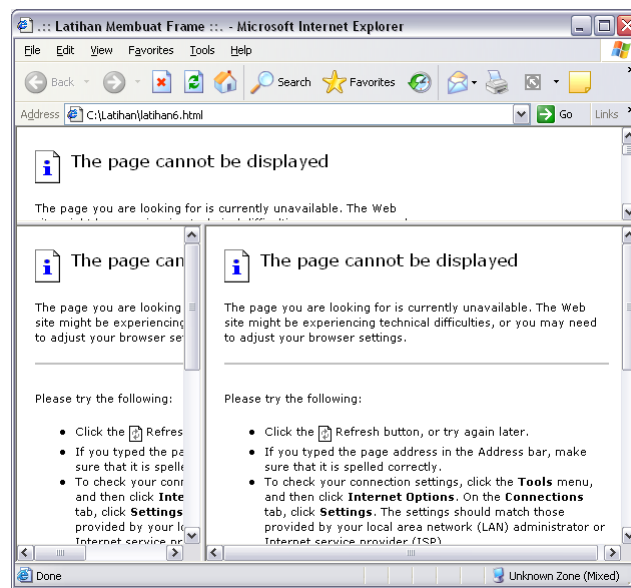
Buat file baru dengan nama latihan6.html, header.html, kiri.html, kanan.html, bab1.html dan bab2.html dengan isi masing-masing sebagai berikut:

File latihan6.html

```
<html>
<head>
<title>.:: Latihan Membuat Frame.:.</title>
</head>
<frameset rows=20%,* framespacing="0" border="0" frameborder="0">
<frame name=atas src=header.html scrolling="no" noresize>
<frameset cols=25%,* framespacing="0" border="0" frameborder="0">
    <frame name=kiri src=kiri.html scrolling="no" noresize>
    <frame name=kanan src=kanan.html scrolling="no" noresize>
</frameset>
</frameset>
</html>
```

Pada tag `<frameset rows=20%,*>` maksudnya adalah frame yang dibuat terdiri dari dua bagian/baris dengan ukuran 20% bagian paling atas dan selebihnya (80%) adalah frame bagian bawah/baris kedua yang ditunjukkan dengan tanda '*'. `<frame name=atas src=header.html>` menunjukkan bahwa nama frame adalah *atas* dan diisi dengan dokumen *header.html*. `<frameset cols=30%,*>` mempunyai arti bahwa frame bawah dibagi lagi menjadi dua kolom dengan ukuran 30% untuk frame kiri dan sisanya untuk frame sebelah kanan.

Kalau file tersebut dijalankan hasilnya sebagai berikut:



File kanan.html

```
<html>
<head>
  <title>::: Latihan Membuat Link Ke Dokumen Lain::~:</title>
</head>
<body bgcolor=#ffffff>
<table border=1 bordercolor=bordercolor="#800080"
  cellpadding=50 cellspacing=0 width=750 height=450>
<tr>
  <td valign=top>
    <font face=verdana size=4 color=blue>
    Selamat datang di web kami ...
  </td>
</tr>
</body>
</html>
```

a. Rangkuman

Setelah kita mempelajari materi pertama ini, kita dapat menarik kesimpulan, ternyata membuat/membangun halaman web itu mudah sekali. Dengan hanya bermodalkan editor teks biasa, kita sudah dapat belajar banyak tentang tag-tag html yang digunakan untuk membuat sebuah halaman web.

Struktur HTML secara garis besar dibagi menjadi <head> dan <body>. Di bagian <body> kita dapat menuliskan semua kode html yang akan kita gunakan untuk membangun halaman web, yaitu diantaranya pengaturan teks, pengaturan image, hyperlink, pembuatan table dan pembuatan frame.

Lanjutkan ke materi berikutnya agar pengetahuan anda dalam membangun halaman web lebih banyak lagi.

b. Tugas

1. Kunjungi beberapa website (minimal 10 website). Amatilah website yang telah anda kunjungi. Yang harus anda amati adalah meliputi:
 - a. **Layout,**
 - 📁 Seberapa besar daya tarik dan unsur seni dari web yang anda amati.
 - 📁 Bagaimana model layout web tersebut, berbentuk tabel atau frame?
 - b. **Content/Isi,** Apa saja isi dari web tersebut.
Silahkan catat hasil pengamatan anda.
2. Teliti profil sebuah organisasi/lembaga (misal: KUD, Pemerintah Kota, Sekolah dan lain-lain). Catat semua hasil penelitian anda. Dari hasil penelitian anda, silahkan buat website statis yang isinya adalah profile dari yang anda teliti. Bila perlu tawarkan website buatan anda kepada organisasi/lembaga tersebut.

- Silahkan cari dan *download* tutorial berbahasa Inggris di Internet yang berhubungan dengan web programming, (misalnya PHP, ASP dan lain-lain).
- Buatlah layout seperti gambar dibawah ini menggunakan perintah HTML

SURAT PERINTAH JALAN
DENPASAR

No: / SPJ / OPR / / 200

Denpasar - Bogor

Kembali dari		Kode Bus	
Tgl. Kembali		No. Polisi / No. Produk	
J a m		- Pengemudi I /NIK/TTD	
Kelas	SE EXE *)	- Pengemudi II/NIK/TTD	
		- Kenek/NIK/TTD	

*) Coret yang tidak perlu

No.	Keterangan	DPS	KET	RM		PRB	RM		K	RM	RW		Depo
				PP	TS		SR	CRN			PG		
1	Pengemudi I												
2	Pengemudi II												
3	Kenek												
	K R U	I II K	I II K	I II K	I II K	I II K	I II K	I II K	I II K	I II K	I II K	I II K	I II K
4	Kesopanan Kru												
5	Jumlah Penumpang												
6	Jam												
7	Tanggal												
8	Nama Kontrol												
9	TTD Kontrol												

Denpasar, - - 200

(_____)
Kepala Cabang

(_____)
Satpam / Operasi

Disetujui :

(_____)
Manager Operasi

(_____)
Bag. Keuangan

c. Tes Formatif

1. Jelaskan masing-masing fungsi dari tag-tag berikut:
 - a) <h1>
 - b) <p>
 - c)

 - d) <hr>
2. Apakah fungsi dari link, form dan frame dalam pembuatan web?
3. Apa perbedaan frame dan frameset?
4. Apa perbedaan checkbox dan radio button?
5. Apakah fungsi dari tombol Submit dan Reset?
6. Buatlah halaman web seperti terlihat di gambar. Dalam gambar tersebut, jika ada penekanan tombol kirim, maka hasilnya akan dikirimkan ke e-mail anda.

The screenshot shows a Microsoft Internet Explorer window titled "Form Order ::: - Microsoft Internet Explorer". The address bar shows "F:\vedc(modul\Modul Nasional)\html\job1.html". The main content area displays a form titled "FORM ORDER BARANG". The form is divided into several sections: "Isikan Data-data Anda" with fields for "Nama Depan", "Nama Akhir", and "Umur"; "Bagaimana Menghubungi Anda?" with fields for "Alamat Rumah", "Kota", "Negara", "Kode Pos", and "Nomor Telepon"; "Credit Card" with radio buttons for "Visa" and "M/C", and a "Tanggal Berakhir" field; and "Jenis Barang" with a dropdown menu showing "Pentium IV". Below the form, there is a large, bold message "Terimakasih Atas Order Anda!" and two buttons: "Kirim" and "Hapus Isian". The status bar at the bottom shows "Done" and "My Computer".

d. Kunci Jawaban Formatif 1

1. Fungsi dari tag:
 - a. <h1> : digunakan untuk mengatur ukuran huruf pada header dengan angka 1 berarti mempunyai ukuran paling besar.
 - b. <p> : digunakan untuk berpindah alinea atau membuat paragraf baru
 - c.
 : Digunakan untuk pindah baris baru.
 - d. <hr> : digunakan untuk membuat garis horisontal
2. Link, untuk melompat dari satu dokumen ke dokumen atau tempat yang lain, dapat berupa teks maupun gambar.

Form, digunakan untuk mengumpulkan informasi dari pengunjung (berinteraksi dengan pengunjung) karena form dapat berupa model isian yang harus diisi pengunjung dan akan dibaca oleh pemilik web.

Frame, merupakan pembagi halaman.

3. Frame adalah dokumen yang terdapat di dalam frameset dengan border yang mengelilinginya. Sedangkan frameset, adalah suatu frame yang dapat menampung beberapa frame lain di dalamnya dan dapat memiliki beberapa frameset yang lainnya.
4. Checkbox dan radio button memiliki fungsi yang sama yaitu untuk menentukan pilihan dari suatu daftar pilihan. Perbedaannya terletak pada jumlah pilihan yang dapat dipilih. Pada checkbox, pengunjung halaman web dapat memilih beberapa dari daftar pilihan, namun dengan radio button, pengunjung hanya berhak menentukan satu pilihan saja.
5. Tombol submit berfungsi untuk mengirimkan data yang telah dimasukkan dalam form untuk selanjutnya diolah oleh server. Dan tombol reset berfungsi untuk mengosongkan dan atau mengembalikan ke nilai *default* data yang ada dalam form.
6. Kode untuk halaman web tersebut adalah:

```
<html>
<head>
  <title>::: Form Order::~</title>
</head>
<body>
<font face=verdana>
<h3 align=center>FORM ORDER BARANG
<form action="mailto:jack_delphi@yahoo.com" method=post>
<table border=2 cellpadding=1>
<tr>
  <td rowspan=2><font size=2>Isikan Data-data Anda</td>
  <td><input type=text name="NamaDepan" size=20></td>
  <td><input type=text name="NamaBelakang" size=20></td>
  <td><input type=text name="Usia" size=3></td>
</tr>
<tr>
  <td><font size="-2">Nama Depan</font></td>
  <td><font size="-2">Nama Akhir</font></td>
  <td><font size="-2">Umur</font></td>
</tr>
<tr>
  <td rowspan=3><font size=2>Bagaimana Menghubungi Anda?</td>
  <td colspan=4 valign=top><font size=2>Alamat Rumah:
    <textarea name="Jalan" rows=2 cols=30></textarea></td>
</tr>
<tr>
  <td colspan=2><font size=2>Kota:
    <input type=text name="kota" size=20></td>
  <td colspan=2><font size=2>Negara:
    <input type=text name="negara" size=25></td>
</tr>
</tr>
```



```
<tr>
  <td colspan=2><font size=2>Kode Pos:
    <input type=text name="kodepos" size=10></td>
  <td colspan=2><font size=2>Nomor Telepon
    <input type=text name="telp1" size=4> -
    <input type=text name="telp2" size=11></td>
</tr>

<tr>
  <td><font size=2>Credit Card
    <input type=radio name=CC value=Visa checked>Visa
    <input type=radio name=CC value=MasterCard checked>M/C</td>
  <td colspan=2 align=center>
    <input type=text name=nomorCC1 size=4>
    <input type=text name=nomorCC2 size=4>
    <input type=text name=nomorCC3 size=4>
    <input type=text name=nomorCC4 size=4></td>
  <td colspan=2 align=center><font size=2>Tanggal Berakhir:
    <input type=text name=blnakhir size=2>
    <input type=text name=thnakhir size=2></td>
</tr>

<tr>
  <td><font size=2>Jenis Barang </td>
  <td colspan=4><font size=2>
    <select multiple name=Merchandise size=1>
    <option selected>Pentium IV
    <option>Pentium III
    <option>Monitor
    <option>CD-ROM
    <option>Kamera Digital
    <option>Printer
    <option>Mouse
    <option>Scanner
    </select></td>
</tr>

<tr>
  <td align=center colspan=5>
    <h1>Terimakasih Atas Order Anda!</h1>
  </td>
</tr>
</table><p>
<center>
<input type="submit" value="Kirim">
<input type="Reset" value="Hapus Isian">
</center>
</form>
</body>
</html>
```

CASCADING STYLE SHEET (CSS)

1. Aturan CSS

A. Internal CSS

- Cara menempatkan CSS dalam HTML

Cara Deklarasi/pembuatan css dalam HTML adalah ditempatkan di antara <HEAD> dan </HEAD>

Sintak :

```
<style type = "text/css">
  Nama_selector
  {
      Nama_property1: nilai1 ;
      Nama_property2: nilai2;
      .....
      Nama_propertyn: nilai n;
  }
</style>
```

Contoh penerapan CSS dalam kehidupan nyata :

```
<style type="Text/css">
  Orang
  {
    Mata:biru;
    Tinggi:178m;
    Berat:78kg;
  }
</style>
```

2. Group dalam CSS

Untuk membuat selector secara banyak, dapat digunakan tanda koma sebagai pemisah antar selector.

Sintak :

```
Selektor1, Selektor2, Selektorn
{
  Property : value;
}
```

Contoh :

```
<style type='text/css'>
h1, h2, h3, h4, h5, h6
{
font-family: sans-serif;
color: maroon;
border-bottom: 1px solid rgb(200, 200, 200);
}
</style>
```

Penerapan dalam HTML :

```
<html>
```

```
<head>
<title>Group Dalam CSS</title>
<style type='text/css'>
h1, h2, h3, h4, h5, h6
{
font-family: sans-serif;
color: maroon;
border-bottom: 1px solid rgb(200, 200, 200);
}
</style>
</head>
<body>
<h1>Style Sheet</h1>
<h2>Rule</h2>
<h3>Selector</h3>
<h4>Declaration</h4>
<h5>Property</h5>
<h6>Value</h6>
</body>
</html>
```

3. Komentar dalam CSS

Untuk memberi komentar dalam CSS, bisa menggunakan tanda `/* komentar */`

4. Penerapan CSS dalam HTML

- a. Body
- b. Format Huruf
- c. Table

Contoh :

```
<style type='text/css'>
table {
width: auto;
background: black;
color: white;
}
</style>
```

Penerapannya dalam HTML

```
<body>
<table>
<tbody>
<tr>
<td>Pembuatan Tabel dengan CSS, lebar otomatis, latar hitam dan tulisan putih</td>
</tr>
</tbody>
</table>
</body>
```

d. Div

Contoh :

```
<style type='text/css'>
div {
width: auto;
background: black;
color: white;
}
</style>
```

Penerapannya dalam HTML

```
<body>
<div>Penggunaan DIV dengan lebar kotak otomatis, warna background=hitam, dan
tulisan = putih</div>
</body>
```

e. Color/ Penggunaan warna

- Dengan menyebutkan nama warna

Contoh :

```
div {
color: black;
background-color: red;
border: thin solid orange;
}
```

- Dengan RGB

Contoh :

```
body {
background-color: rgb(128, 128, 128);
}
```

- Dengan kode Hexa

Contoh :

```
div {
color: #000000;
background-color: #FF0000;
border: thin solid #FFA500;
}
```

f. URL (menggunakan alamat /lokasi gambar)

g. Strings

```
<style type='text/css'>
div
{
font-family: sans-serif;
}
div::before
{
content: "Halo \"Selamat Datang!\"";
background: black;
color: white;
```

```
margin-right: 25px;
}
</style>
```

Cara memanggilnya :

```
<div>Halo Juga..!</div>
```

h. Hover

Digunakan untuk membuat style pada link < A HREF="">

Contoh :

```
A:link { color : mediumblue; }
A:visited { color:magenta;}
A:hover { text-decoration:underline/none;}
A:active { color:crimson;}
```

i. **Class Selector**

Setiap pembuatan Class Selector selalu dimulai dengan tanda titik (.)

Contoh :

```
.coba {
Position : absolute;
Top:0;
Left:0;
Bottom:15px; }
```

Penerapannya dalam HTML

```
<div class='coba'> <img src=lokasi/gambar.jpg' alt='Coba' class='coba'/> </div>
```

j. **ID Selector**

Pembuatan Class Selector dimulai dengan tanda Sharp (#), lalu nama selectornya

Contoh :

```
#coba
{ left:0; }
```

Penerapannya dalam HTML

```
<div id='coba'>
<img src=lokasi/gambar.jpg' alt='Coba' class='coba'/>
</div>
```

B. External CSS

Rumus :

```
<link rel='stylesheet' href='lokasi/nama_file.css' type='text/css' />
```

PRAKTIKUM 1:

Buatlah program berikut ini, lalu simpan dengan nama Latihan_CSS.html

```
<html>
<head>
<title>Ukuran kotak DIV</title>
<style type='text/css'>
body
{
font: 1em sans-serif;
}
p
{
background: rgb(234, 234, 234);
border: 1px solid rgb(200, 200, 200);
}
p#em-ukuran
{
width: 12em;
padding: 1em;
}
p#px-ukuran
{
width: 192px;
padding: 16px;
}

div {
background: #000;
border: 1px solid rgb(128, 128, 128);
color: white;
font: 9px monospace;
margin: 15px;
text-align: center;
}
div#inches
{
width: 1in;
height: 1in;
}
div#pixels
{
width: 96px;
height: 96px;
}

p.container {
border: 1px solid rgb(29, 179, 82);
```

```
background: rgb(202, 222, 245);  
padding: 10px;  
width: 245px;  
height: 245px;  
float: left;  
margin: 10px;  
}  
p.box {  
border: 1px solid rgb(69, 199, 115);  
background: rgb(164, 201, 245);  
}  
p.tank {  
border: 1px solid rgb(107, 214, 145);  
background: rgb(124, 180, 245);  
clear: left;  
}  
p#container-1234 {  
border: 1px solid rgb(154, 232, 181);  
background: rgb(82, 157, 245);  
}  
/* ---- */  
h1 {  
margin: 5px;  
}  
del {  
color: crimson;  
}  
ins {  
color: forestgreen;  
}  
p {  
border: 1px solid rgb(200, 200, 200);  
background: rgb(234, 234, 234);  
padding: 5px;  
margin: 5px;  
}  
p.note {  
background: yellow;  
border: 1px solid gold;  
}  
span.code {  
font-family: monospace;  
padding: 0 10px;  
}  
p > span.code {  
background: yellow;  
}  
p.note > span.code {
```

```
background: lightyellow;  
}  
</style>
```

```
</head>  
<body>  
<div id='inches'>&lt;-- 1 Inci --&gt;</div>  
<div id='pixels'>&lt;-- 96 Pixel --&gt;</div>  
Penerapan Paragraph  
<p id='em-ukuran'>  
Paragraph ini lebarnya adalah 12em, dengan jarak spasi dari atas = 1em.  
</p>  
<p id='px-ukuran'>  
Lebar paragraf = 192 pixels, dengan jarak spasi dari atas=16px.  
</p>
```

```
<h1>Universal Selectors</h1>  
<p>  
Universal selectors are wildcard selectors.  
</p>  
<p>  
When a universal selector is used alone, all elements  
within a document are selected.  
</p>  
<p>  
Even form elements are selected.  
</p>  
<form method='post' action='#'>  
<fieldset>  
<legend>Feedback Form</legend>  
<table>  
<tbody>  
<tr>  
<td><label for='topic'>Topic:</label></td>  
<td>  
<input type='text' name='topic' id='topic' value='Universal Selectors' size='25' />  
</td>  
</tr>  
<tr>  
<td><label for='feedback'>Feedback:</label></td>  
<td>  
<textarea cols='55' rows='10' name='feedback' id='feedback'>  
Universal selectors have some practical applications.  
For instance, when debugging styles you can select  
all elements and apply a border to see dimensions.  
This could help you identify rogue elements causing
```


undue disorder in a document.

```
</textarea>
</td>
</tr>
</tbody>
</table>
</fieldset>
</form>
```

** Selector Langsung

```
<h1><ins>Direct Child</ins> <del>Descendant</del> Selectors</h1>
```

```
<p>
```

<ins>Direct Child</ins> Descendant selectors apply styles based on <ins>parent/child</ins> ancestral relationships.

The first <ins>direct child</ins> descendant example I present applies style to the

```
<span class='code'>&lt;span&gt;</span> element named <em>code</em>,
which is a <del>descendant</del> <ins>child</ins> of
<span class='code'>&lt;p&gt;</span> elements.
```

To do this, the selector p <ins>></ins> span.code is used.

```
</p>
```

```
<p>
```

Using CSS, styles can be applied to any number of documents. Since this is the case, there may be elements with a class name of code in several documents, but have different styles applied depending on the context it appears, which is the exact situation the inventors of the descendant <ins>child</ins> selector had in mind when it was conceived.

```
</p>
```

```
<p class='note'>
```

The note text is given different styles. To do this another

```
<del>descendant</del> <ins>direct child</ins>
```

selector is used, this time the selector is

```
<span class='code'>p.note <ins>&gt;</ins> span.code</span>
```

```
</p>
```

```
</body>
```

```
</html>
```

PRAKTIKUM 2: Rancanglah sebuah menu navigasi seperti gambar dibawah menggunakan CSS

HOME PROFIL ▾ JAMINAN KESEHATAN ▾ LAYANAN KESEHATAN



PEMBAHASAN :

A. Membuat CSS Untuk Menu Navigator

1. Bukalah Notepad baru.
2. Ketik program berikut :

```
#cssmenu,  
#cssmenu ul,  
#cssmenu ul li,  
#cssmenu ul li a,  
#cssmenu #menu-button {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    list-style: none;  
    line-height: 1;  
    display: block;  
    position: relative;  
    -webkit-box-sizing: border-box;  
    -moz-box-sizing: border-box;  
    box-sizing: border-box;  
}  
#cssmenu:after,  
#cssmenu > ul:after {  
    content: ".";  
    display: block;  
    clear: both;  
    visibility: hidden;  
    line-height: 0;  
    height: 0;  
}  
#cssmenu #menu-button {  
    display: none;  
}  
#cssmenu {  
    width: auto;  
    font-family: 'Open Sans', sans-serif;  
    line-height: 1;  
    background: #ffffff;
```

```
}
#menu-line {
  position: absolute;
  top: 0;
  left: 0;
  height: 3px;
  background: #009ae1;
  -webkit-transition: all 0.25s ease-out;
  -moz-transition: all 0.25s ease-out;
  -ms-transition: all 0.25s ease-out;
  -o-transition: all 0.25s ease-out;
  transition: all 0.25s ease-out;
}
#cssmenu > ul > li {
  float: left;
}
#cssmenu.align-center > ul {
  font-size: 0;
  text-align: center;
}
#cssmenu.align-center > ul > li {
  display: inline-block;
  float: none;
}
#cssmenu.align-center ul ul {
  text-align: left;
}
#cssmenu.align-right > ul > li {
  float: right;
}
#cssmenu.align-right ul ul {
  text-align: right;
}
#cssmenu > ul > li > a {
  padding: 20px;
  font-size: 12px;
  text-decoration: none;
  text-transform: uppercase;
  color: #000000;
  -webkit-transition: color .2s ease;
  -moz-transition: color .2s ease;
  -ms-transition: color .2s ease;
  -o-transition: color .2s ease;
  transition: color .2s ease;
}
#cssmenu > ul > li:hover > a,
#cssmenu > ul > li.active > a {
  color: #009ae1;
}
#cssmenu > ul > li.has-sub > a {
  padding-right: 25px;
}
#cssmenu > ul > li.has-sub > a::after {
  position: absolute;
  top: 21px;
  right: 10px;
}
```

```
width: 4px;
height: 4px;
border-bottom: 1px solid #000000;
border-right: 1px solid #000000;
content: "";
-webkit-transform: rotate(45deg);
-moz-transform: rotate(45deg);
-ms-transform: rotate(45deg);
-o-transform: rotate(45deg);
transform: rotate(45deg);
-webkit-transition: border-color 0.2s ease;
-moz-transition: border-color 0.2s ease;
-ms-transition: border-color 0.2s ease;
-o-transition: border-color 0.2s ease;
transition: border-color 0.2s ease;
}
#cssmenu > ul > li.has-sub:hover > a::after {
  border-color: #009aef;
}
#cssmenu ul ul {
  position: absolute;
  left: -9999px;
}
#cssmenu li:hover > ul {
  left: auto;
}
#cssmenu.align-right li:hover > ul {
  right: 0;
}
#cssmenu ul ul ul {
  margin-left: 100%;
  top: 0;
}
#cssmenu.align-right ul ul ul {
  margin-left: 0;
  margin-right: 100%;
}
#cssmenu ul ul li {
  height: 0;
  -webkit-transition: height .2s ease;
  -moz-transition: height .2s ease;
  -ms-transition: height .2s ease;
  -o-transition: height .2s ease;
  transition: height .2s ease;
}
#cssmenu ul li:hover > ul > li {
  height: 32px;
}
#cssmenu ul ul li a {
  padding: 10px 20px;
  width: 160px;
  font-size: 12px;
  background: #333333;
  text-decoration: none;
  color: #dddddd;
  -webkit-transition: color .2s ease;
```

```
-moz-transition: color .2s ease;
-ms-transition: color .2s ease;
-o-transition: color .2s ease;
transition: color .2s ease;
}
#cssmenu ul ul li:hover > a,
#cssmenu ul ul li a:hover {
  color: #ffffff;
}
#cssmenu ul ul li.has-sub > a::after {
  position: absolute;
  top: 13px;
  right: 10px;
  width: 4px;
  height: 4px;
  border-bottom: 1px solid #dddddd;
  border-right: 1px solid #dddddd;
  content: "";
  -webkit-transform: rotate(-45deg);
  -moz-transform: rotate(-45deg);
  -ms-transform: rotate(-45deg);
  -o-transform: rotate(-45deg);
  transform: rotate(-45deg);
  -webkit-transition: border-color 0.2s ease;
  -moz-transition: border-color 0.2s ease;
  -ms-transition: border-color 0.2s ease;
  -o-transition: border-color 0.2s ease;
  transition: border-color 0.2s ease;
}
#cssmenu.align-right ul ul li.has-sub > a::after {
  right: auto;
  left: 10px;
  border-bottom: 0;
  border-right: 0;
  border-top: 1px solid #dddddd;
  border-left: 1px solid #dddddd;
}
#cssmenu ul ul li.has-sub:hover > a::after {
  border-color: #ffffff;
}
@media all and (max-width: 768px), only screen and (-webkit-min-device-
pixel-ratio: 2) and (max-width: 1024px), only screen and (min--moz-
device-pixel-ratio: 2) and (max-width: 1024px), only screen and (-o-
min-device-pixel-ratio: 2/1) and (max-width: 1024px), only screen and
(min-device-pixel-ratio: 2) and (max-width: 1024px), only screen and
(min-resolution: 192dpi) and (max-width: 1024px), only screen and (min-
resolution: 2dppx) and (max-width: 1024px) {
  #cssmenu {
    width: 100%;
  }
  #cssmenu ul {
    width: 100%;
    display: none;
  }
  #cssmenu.align-center > ul,
  #cssmenu.align-right ul ul {
```

```
    text-align: left;
}
#cssmenu ul li,
#cssmenu ul ul li,
#cssmenu ul li:hover > ul > li {
    width: 100%;
    height: auto;
    border-top: 1px solid rgba(120, 120, 120, 0.15);
}
#cssmenu ul li a,
#cssmenu ul ul li a {
    width: 100%;
}
#cssmenu > ul > li,
#cssmenu.align-center > ul > li,
#cssmenu.align-right > ul > li {
    float: none;
    display: block;
}
#cssmenu ul ul li a {
    padding: 20px 20px 20px 30px;
    font-size: 12px;
    color: #000000;
    background: none;
}
#cssmenu ul ul li:hover > a,
#cssmenu ul ul li a:hover {
    color: #000000;
}
#cssmenu ul ul ul li a {
    padding-left: 40px;
}
#cssmenu ul ul,
#cssmenu ul ul ul {
    position: relative;
    left: 0;
    right: auto;
    width: 100%;
    margin: 0;
}
#cssmenu > ul > li.has-sub > a::after,
#cssmenu ul ul li.has-sub > a::after {
    display: none;
}
#menu-line {
    display: none;
}
#cssmenu #menu-button {
    display: block;
    padding: 20px;
    color: #000000;
    cursor: pointer;
    font-size: 12px;
    text-transform: uppercase;
}
#cssmenu #menu-button::after {
```

```
    content: '';
    position: absolute;
    top: 20px;
    right: 20px;
    display: block;
    width: 15px;
    height: 2px;
    background: #000000;
}
#cssmenu #menu-button::before {
    content: '';
    position: absolute;
    top: 25px;
    right: 20px;
    display: block;
    width: 15px;
    height: 3px;
    border-top: 2px solid #000000;
    border-bottom: 2px solid #000000;
}
#cssmenu .submenu-button {
    position: absolute;
    z-index: 10;
    right: 0;
    top: 0;
    display: block;
    border-left: 1px solid rgba(120, 120, 120, 0.15);
    height: 52px;
    width: 52px;
    cursor: pointer;
}
#cssmenu .submenu-button::after {
    content: '';
    position: absolute;
    top: 21px;
    left: 26px;
    display: block;
    width: 1px;
    height: 11px;
    background: #000000;
    z-index: 99;
}
#cssmenu .submenu-button::before {
    content: '';
    position: absolute;
    left: 21px;
    top: 26px;
    display: block;
    width: 11px;
    height: 1px;
    background: #000000;
    z-index: 99;
}
#cssmenu .submenu-button.submenu-opened:after {
    display: none;
}
}
```

```
}
```

3. Save All atau simpan dengan nama file = **Style.css** (pastikan keadaan All Files) pada folder Praktikum yang sama

B. Membuat Menu Navigator menggunakan Notepad, simpan dengan nama **PraktikumCSS.html**

```
<!doctype html>
<html lang=''>
<head>
  <link rel="stylesheet" href="Gaya5B.css">
  <style type="text/css">
    #cssmenu p {
      font-weight: bold;
      font-size: 14px;
      color: #FF0;
    }
  </style>
  <title>Pertemuan VII Menu Navigator</title>
</head>
<body>

<div id='cssmenu'>
<ul>
  <li><a href='#'><span>Home</span></a></li>
  <li class='active has-sub'><a href='#'><span>Profil</span></a>
    <ul>
      <li class='has-sub'><a href='#'><span>Visi dan
Misi</span></a></li>
      <li class='has-sub'><a href='#'><span>Struktur
Organisasi</span></a></li>
      <li class='has-sub'><a href='#'><span>Sejarah</span></a></li>
      <li class='has-sub'><a href='#'><span>Landasan
Hukum</span></a></li>
      <li class='has-sub'><a href='#'><span>Direksi</span></a></li>
      <li class='has-sub'><a href='#'><span>Galeri</span></a>
        <ul>
          <li><a href='#'><span>Video</span></a></li>
          <li class='last'><a href='#'><span>Foto</span></a></li>
        </ul>
      </li>
    </ul>
  </li>
  <li class='active has-sub'><a href='#'><span>Jaminan
Kesehatan</span></a>
    <ul>
      <li class='has-sub'><a href='#'><span>Peserta</span></a></li>
      <li class='has-sub'><a href='#'><span>Manfaat</span></a></li>
      <li class='has-sub'><a href='#'><span>Iuran</span></a></li>
      <li class='has-sub'><a href='#'><span>Prosedure
Pendaftaran</span></a></li>
    </ul>
  </li>

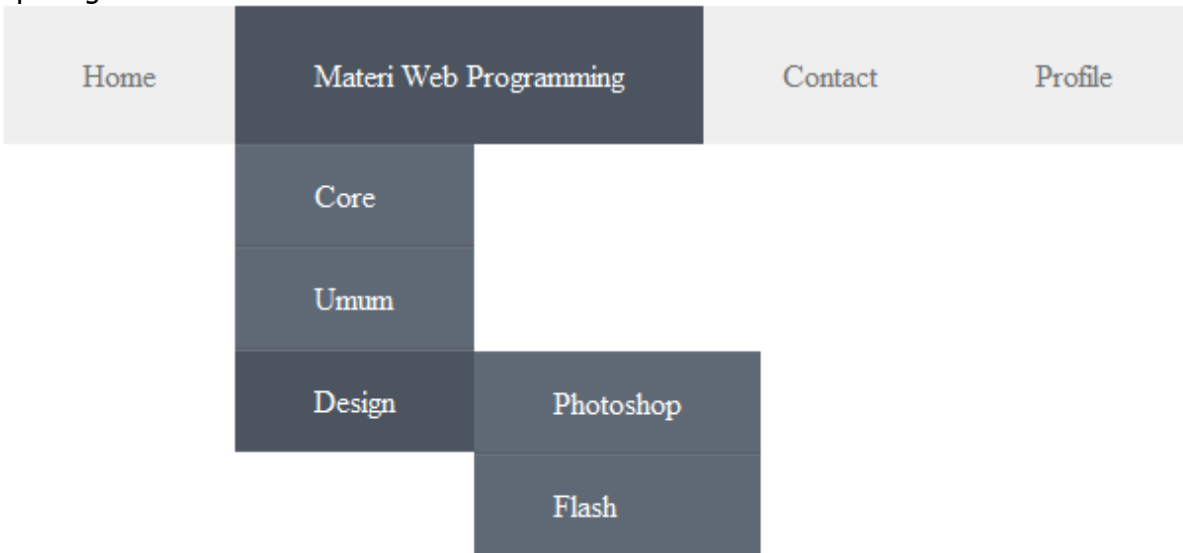
  <li class='last'><a href='#'><span>Layanan Kesehatan</span></a></li>
</ul>
```



```
</div>  
</body>  
<html>
```

Langkah terakhir Jalankan PraktikumCSS.HTML melalui Chrome, Mozilla, IE

TUGAS : Rancanglah sebuah menu navigasi menggunakan CSS dan panggil melalui file HTML, seperti gambar berikut ini :



Simpan Tugas di atas dengan nama :

- Nama file CSS → StyleTugas.css,
- nama file HTML → TugasCSS.HTML

FORM

Sebuah form merupakan bagian dari dokumen web yang dapat diisi oleh pengguna, untuk memberikan informasi tertentu dari pengguna kepada website. Sebuah form sangat penting dalam sebuah aplikasi web, terutama aplikasi web dinamis, karena form merupakan satu-satunya sarana bagi pengembang website untuk mendapatkan informasi dari pengguna.

Mengetahui bagaimana membuat form pada HTML dengan benar penting untuk memastikan tidak terdapat kesalahan pemahaman pengguna dalam menggunakan form yang disediakan. Bagian ini akan menjelaskan bagaimana membuat form dengan HTML, elemen-elemen form yang disediakan, serta bagaimana elemen-elemen tersebut digunakan. Pemrosesan form secara dinamis tidak akan dibahas, dan properti CSS untuk memperindah tampilan form akan dibahas pada bagian berikutnya.

Inisialisasi Form

Form pada HTML dibuat dengan menggunakan elemen form. Elemen form harus membungkus seluruh elemen-elemen masukan form lain untuk memastikan masukan informasi pengguna

dapat dibaca oleh aplikasi web. Pembungkusan elemen-elemen masukan dilakukan sama seperti kita membungkus elemen-elemen lain pada div:

```
<form action="#" method="post">  
....  
</form>
```

Seperti yang dapat dilihat pada kode di atas, terdapat dua atribut yang wajib dimiliki oleh elemen form. Atribut pertama ialah `action`, yang berfungsi untuk memberitahukan browser alamat pengiriman dari data-data yang diisikan pengguna di dalam form. Idealnya parameter ini diisikan dengan sebuah URL pada server yang melakukan pemrosesan data.

Atribut kedua yang wajib diisikan ialah atribut `method`. Atribut ini memberitahukan browser *bagaimana* data akan dikirimkan kepada server yang alamatnya diisikan pada `action`. Terdapat dua nilai yang dapat diisikan pada bagian ini, yaitu `get` dan `post`. Kedua nilai ini sendiri merupakan protokol HTTP yang digunakan untuk pengiriman data

Elemen Masukan Form

Terdapat banyak elemen-elemen masukan pada form, yang dapat digunakan oleh pengguna untuk mengisikan data dengan yang berbeda-beda jenisnya. Bagian ini akan membahas tiap-tiap elemen masukan form, beserta dengan cara pembuatannya.

Elemen Masukan Teks: TextField dan TextArea

Pengisian informasi dalam bentuk teks pada form HTML dapat dilakukan dengan menggunakan dua buah elemen: `textarea` dan `input`. `textarea` digunakan untuk masukan teks yang terdiri dari beberapa baris, sementara `input` digunakan untuk masukan teks yang hanya satu baris.

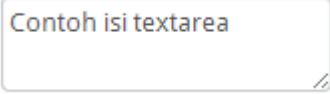
Penggunaan elemen `textarea` dapat dilakukan dengan sangat sederhana, hanya langsung memasukkan tag-nya saja:

```
<textarea>  
</textarea>
```

selain memberikan tag kosong seperti di atas, kita juga dapat mengisikan tag untuk mendapatkan nilai masukan standar:

```
<textarea>  
Contoh isi textarea
```

```
</textarea>
```



Contoh Penggunaan Elemen TextArea

Pengaturan panjang dan lebar dari `textarea` dapat dilakukan melalui CSS, dengan menggunakan properti `height` dan `width`.

Walaupun dapat mengisikan teks dengan banyak sekaligus, `textarea` tentunya tidak dapat digunakan untuk seluruh kasus pengisian data. Seringkali kita menginginkan pengguna hanya mengisikan data singkat, tanpa isi teks yang banyak. Untuk jenis masukan seperti itu, kita dapat menggunakan elemen `input`:

```
<input type="text">
```

yang akan menghasilkan elemen masukan seperti berikut:



Contoh Penggunaan Elemen Input

Perhatikan juga bahwa kita menggunakan atribut `type` pada elemen ini untuk menentukan data-data yang akan diisikan. Terdapat banyak jenis tipe data yang dapat kita isikan, dan browser akan menyesuaikan jenis masukan teks yang ada, sesuai dengan atribut `type` yang kita tentukan. Nilai-nilai yang dapat diisikan pada atribut `type` yaitu:

1. `color`
2. `datetime-local`
3. `number`
4. `tel`
5. `week`
6. `date`
7. `email`
8. `range`
9. `time`
10. `datetime`
11. `month`
12. `search`
13. `url`

14. password
15. text
16. file

Elemen Pemilihan: Radio Button, Checkbox, dan Dropdown

Selain mengisikan data berupa teks, seringkali kita juga menemukan data berupa pilihan, misalnya jenis kelamin atau hobi. Pengisian data seperti ini biasanya memberikan beberapa pilihan kepada pengguna, dan pengguna dapat memilih satu atau beberapa pilihan yang diberikan.

Jika kita ingin memberikan pilihan kepada pengguna, dan memastikan pengguna hanya memilih satu pilihan saja, kita dapat menggunakan **radio button**. Radio button dibuat dalam HTML dengan menggunakan elemen input, dengan atribut type bernilai `radio` :

```
<input type="radio" name="sex" value="pria"> Pria <br>  
<input type="radio" name="sex" value="wanita"> Wanita
```

dalam pembuatan elemen radio button, kita juga wajib menambahkan dua atribut lainnya, yaitu name dan value. Atribut name digunakan untuk memberitahukan browser bahwa radio button dengan atribut name yang sama adalah merupakan kumpulan radio button yang sama, sehingga pengguna tidak boleh memilih dua buah pilihan pada radio button tersebut.

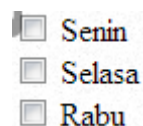
Atribut value digunakan untuk menyimpan nilai yang ingin kita kirimkan ke server.



Jika ingin memberikan pengguna kemampuan untuk memilih banyak pilihan sekaligus, kita dapat menggunakan elemen **checkbox**. Elemen ini sama seperti radio button, dibuat dengan elemen input yang nilai atribut type-nya berisikancheckbox.

```
<input type="checkbox" name="day" value="senin"> Senin <br>  
<input type="checkbox" name="day" value="selasa"> Selasa <br>  
<input type="checkbox" name="day" value="rabu"> Rabu
```

Seperti yang dapat dilihat, atribut name dan value juga wajib dimiliki oleh checkbox, dengan alasan yang sama untuk radio button.

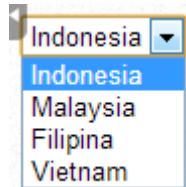


Meskipun dapat memberikan pilihan nilai kepada pengguna, radio button maupun checkbox sangat tidak ideal untuk digunakan pada pilihan yang memiliki nilai yang sangat banyak. Misalnya, pilihan negara yang ditinggali oleh pengguna dari ratusan negara yang ada di dunia akan jika ditampilkan pada radio button akan menyebabkan daftar yang dipaparkan sangat banyak dan sulit dibaca oleh pengguna. Dalam kasus seperti ini lebih baik kita menggunakan **dropdown list**.

Pembuatan elemen dropdown dilakukan dengan menggabungkan dua elemen, yaitu `select` dan `option`. Elemen `select` membungkus seluruh elemen `option` yang ada, untuk membentuk sebuah dropdown. Elemen `option` sendiri merupakan nilai dari dropdown yang diinginkan. Perhatikan kode di bawah:

```
<select name="country">
  <option value="indonesia">Indonesia</option>
  <option value="malaysia">Malaysia</option>
  <option value="filipina">Filipina</option>
  <option value="vietnam">Vietnam</option>
</select>
```

yang akan menghasilkan:



dan jika ingin memungkinkan pengguna memilih beberapa pilihan kita dapat menambahkan atribut `multiple` pada elemen `select`:

```
<select name="country" multiple>
  <option value="indonesia">Indonesia</option>
  <option value="malaysia">Malaysia</option>
  <option value="filipina">Filipina</option>
  <option value="vietnam">Vietnam</option>
</select>
```



Elemen Tersembunyi

Elemen tersembunyi memberikan kebebasan pada pengembang untuk mengirimkan data ke server tanpa menampilkan data tersebut kepada pengguna. Elemen tersembunyi biasanya berisi kode khusus untuk melacak pengguna, kunci enkripsi, atau kode-kode lainnya yang tidak penting bagi pengguna, tetapi diperulka oleh website.

Pembuatan elemen tersembunyi dilakukan dengan menggunakan elemen input, yang memiliki atribut `type` bernilai `hidden`.

```
<input type="hidden" name="csrf_token" value="a1923axclkaseruczxcna">
```

Tombol pada Form

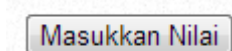
Setelah pengguna mengisikan data-data yang diminta oleh dokumen web, tentu saja pengguna harus mengirimkan data tersebut ke server. Pengiriman data dapat dilakukan pengguna melalui tombol khusus yang disediakan oleh HTML.

Tombol Pengiriman

Pembuatan tombol pengiriman juga dilakukan dengan menggunakan elemen input, yang atribut `type`-nya diisikan dengan nilai `submit`, seperti berikut:

```
<input type="submit" name="submit" value="Masukkan Form">
```

Perhatikann bahwa berbeda dengan elemen-elemen input sebelumnya, nilai dari atribut `value` diproses menjadi teks dari tombol yang ditampilkan ke pengguna. Hal ini menyebabkan kita tidak lagi perlu menambahkan teks setelah elemen form, seperti pada radio button ataupun checkbox.



Tombol Penghapus Input

Ketika sedang mengisikan form, seringkali pengguna menyadari bahwa data-data yang diisikannya mayoritas adalah data yang salah. Pada saat ini biasanya pengguna akan menghapus

seluruh isi dari form, satu per satu. Untungnya HTML telah memberikan fitur untuk menghapus seluruh isi form sekaligus, melalui tombol penghapus input.

Pembuatan tombol dilakukan sama persis seperti pada tombol pengiriman, dengan perbedaan nilai pada atribut type, yang diisikan dengan reset pada tombol penghapusan ini.

```
<input type="reset" name="reset" value="Kosongkan Form">
```

Organisasi Elemen Form

Mengetahui bagaimana membuat elemen-elemen masukan pada form masih merupakan langkah awal dalam pembuatan form HTML. Elemen-elemen masukan form jika ditampilkan tanpa informasi tambahan tidaklah berguna, karena pengguna tidak dapat mengetahui data apa yang harus diisikan ke dalam elemen-elemen tersebut.

UNTuk memberikan informasi tambahan kepada pengguna, HTML juga menyediakan elemen-elemen yang dapat digunakan sebagai penanda dari elemen masukan form. Adapun elemen-elemen yang dapat digunakan untuk tujuan tersebut yaitu label, fieldset, dan legend.

Label

Elemen label memberikan fasilitas untuk menambahkan teks pada elemen masukan form. Elemen ini diisikan dengan deskripsi dari elemen masukan yang ingin ditambahkan, dan harus memiliki atribut for. Atribut for pada elemenlabel berisikan nilai yang sama dengan atribut id pada elemen masukan form. Pengisian nilai yang sama akan mengikatkan elemen label dengan elemen masukan, sehingga pengguna dapat langsung mengisikan nilai dengan melakukan klik pada teks yang dihasilkan oleh elemen label.

Perhatikan kode berikut:

```
<label for="username">Username</label>  
<input type="text" name="username" id="username">
```

di mana isi dari atribut for pada label adalah sama dengan isi atribut id pada elemen input. Lihat hasil eksekusi kode pada [halaman berikut](#) dan coba klik teks "Username" untuk melihat efeknya. Coba juga untuk menggantikan isi atribut for atau id untuk melihat hasilnya.

Fieldset

Elemen fieldset merupakan elemen yang digunakan untuk membungkus beberapa elemen masukan form, untuk menandakan bahwa elemen-elemen tersebut merupakan elemen masukan yang berada pada satu grup yang sama, atau saling berhubungan.

Secara standar, elemen `fieldset` akan memberikan border di sekitar grup elemen-elemen di dalamnya, yang tentunya dapat diubah dengan menggunakan CSS. Berikut adalah contoh penggunaan `fieldset`:

```
<fieldset>
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <label for="password">Password</label>
  <input type="text" name="password" id="password">
</fieldset>
```

yang hasil eksekusinya adalah:



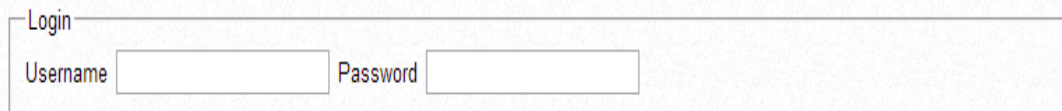
A screenshot of a web form. It features a single rectangular border (fieldset) containing two text input fields. The first field is preceded by the label 'Username' and the second by 'Password'. Both fields are empty.

Legend

Elemen `legend` digunakan untuk memberikan judul pada sebuah `fieldset`. Penggunaan elemen `legend` sangat sederhana, hanya dengan menambahkan elemen tersebut sebagai *child* pertama dari `fieldset`, seperti berikut:

```
<fieldset>
  <legend>Login</legend>
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <label for="password">Password</label>
  <input type="text" name="password" id="password">
</fieldset>
```

yang akan menghasilkan tampilan seperti berikut:



A screenshot of a web form. The form is enclosed in a rectangular border. At the top left of the border is the word 'Login'. Below it, there are two text input fields. The first field is preceded by the label 'Username' and the second by 'Password'. Both fields are empty.

dan tentunya tampilan dari `legend` dapat diubah dengan menggunakan CSS.

Validasi Masukan pada Form

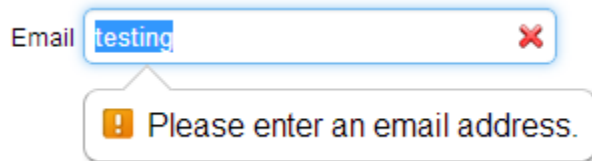
Seringkali dalam pengembangan form pada aplikasi, terdapat elemen-elemen yang tidak wajib diisi oleh pengguna dan elemen-elemen yang wajib diisi oleh pengguna. Aplikasi kemudian akan melakukan pengecekan terhadap nilai masukan dari pengguna, apakah nilai yang wajib ada sudah diisi atau belum. Jika nilai belum diisi maka aplikasi akan menolak form pengguna, dengan pesan kesalahan yang menjelaskan bahwa terdapat isian yang wajib diisi oleh pengguna. Proses pengecekan kewajiban isi atau berbagai batasan lain ini dikenal dengan nama validasi.

Validasi pada elemen-elemen form HTML dilakukan dengan beberapa cara, yaitu:

1. Atribut `required` pada elemen-elemen masukan yang wajib diisi. Jika pengguna tidak mengisi elemen maka browser secara otomatis akan memberikan peringatan kepada pengguna dan membatalkan pengiriman form. Contoh penggunaan atribut ini yaitu:

2. `<input type="text" name="username" required>`

3. Penggunaan atribut `type` yang benar, untuk memastikan browser dapat melakukan pengecekan nilai yang diisi pengguna dengan format sesuai dengan yang dispesifikasikan pada standar HTML. Misalnya, jika menggunakan `type="email"`, maka pengguna tidak dapat mengisi bukan email sebagai masukan form.



Contoh Pesan Kesalahan Pengisian Email pada HTML

Jadi, pastikan anda menggunakan atribut-atribut yang tepat sesuai dengan data yang ingin disimpan, untuk memastikan browser dapat bekerja dengan optimal dalam melakukan validasi terhadap elemen-elemen dalam form.

PRAKTIKUM : Buatlah sebuah form seperti pada gambar dibawah ini. Simpan dengan nama latihan_Form.html


FORMULIR CALON PESERTA DIDIK BARU SMK N 1 RAMBAH TP. 2014/2015				
1. IDENTITAS PESERTA DIDIK				
Nama Lengkap	: <input type="text"/>			
Jenis kelamin*)	: <input type="checkbox"/> 1) Laki-laki <input type="checkbox"/> 2) Perempuan			
NISN	: <input type="text"/> NIS : <input type="text"/>			
NIK	: <input type="text"/>			
Tempat, Tanggal Lahir	: <input type="text"/> / <input type="text"/> / <input type="text"/>			
Agama*)	: <input type="checkbox"/> 1 Islam <input type="checkbox"/> 2 Kristen Protestan <input type="checkbox"/> 3 Kristen Katholik <input type="checkbox"/> 4 Hindu <input type="checkbox"/> 5 Budha			
Alamat Tempat Tinggal	: <input type="text"/> <input type="text"/>			
Dusun	: <input type="text"/> RT <input type="text"/> RW <input type="text"/>			
Kelurahan / Desa	: <input type="text"/> Kode Pos : <input type="text"/>			
Kecamatan	: <input type="text"/>			
Kabupaten/Kota	: <input type="text"/>			
Provinsi	: <input type="text"/>			
Alat transportasi ke sekolah*)	: <input type="checkbox"/> 1) jalan kaki <input type="checkbox"/> 2) kendaraan pribadi <input type="checkbox"/> 3) kendaraan umum <input type="checkbox"/> 4) antar jemput			
Jenis Tinggal*)	: <input type="checkbox"/> 1) bersama Ortu <input type="checkbox"/> 2) wali <input type="checkbox"/> 3) kost <input type="checkbox"/> 4) panti asuhan <input type="checkbox"/> 5) lainnya			
No Telepon Rumah	: <input type="text"/> - <input type="text"/> No. HP : <input type="text"/>			
Email pribadi	: <input type="text"/>			
No Ijazah SMP/Madrasah/Sederajat	: <input type="text"/>			
No SKHUN SMP/Madrasah/Sederajat	: <input type="text"/>			
Apakah sebagai penerima KPS*)	: <input type="checkbox"/> 1) Ya <input type="checkbox"/> 2) tidak No. KPS <input type="text"/>			
DATA ORANG TUA CALON SISWA				
Nama Ayah	: <input type="text"/> Tahun Lahir <input type="text"/>			
Pekerjaan*)	: <input type="checkbox"/> 1) tidak bekerja <input type="checkbox"/> 2) petani <input type="checkbox"/> 3) PNS/TNI/Polri <input type="checkbox"/> 4) Karyawan Swasta <input type="checkbox"/> 5) Dagang <input type="checkbox"/> 6) wiraswasta <input type="checkbox"/> 7) wirausaha <input type="checkbox"/> 8) buruh <input type="checkbox"/> 9) pensiunan			
Pendidikan*)	: <input type="checkbox"/> 1) tdk sekolah <input type="checkbox"/> 2) putus SD <input type="checkbox"/> 3) SD <input type="checkbox"/> 4) SLTP <input type="checkbox"/> 5) SLTA <input type="checkbox"/> 6) D1 <input type="checkbox"/> 7) D2 <input type="checkbox"/> 8) D3 <input type="checkbox"/> 9) D4/S1 <input type="checkbox"/> 10) S2 <input type="checkbox"/> 11) S3			
Penghasilan bulanan*)	: <input type="checkbox"/> 1) kurang dari Rp1.000.000,- <input type="checkbox"/> 2) Rp1.000.000 - Rp2.000.000,- <input type="checkbox"/> 3) > Rp2.000.000,-			
Nama Ibu	: <input type="text"/> Thn. Lahir <input type="text"/>			
Pekerjaan*)	: <input type="checkbox"/> 1) tidak bekerja <input type="checkbox"/> 2) petani <input type="checkbox"/> 3) PNS/TNI/Polri <input type="checkbox"/> 4) Karyawan Swasta <input type="checkbox"/> 5) Dagang <input type="checkbox"/> 6) wiraswasta <input type="checkbox"/> 7) wirausaha <input type="checkbox"/> 8) buruh <input type="checkbox"/> 9) pensiunan			
Pendidikan*)	: <input type="checkbox"/> 1) tdk sekolah <input type="checkbox"/> 2) putus SD <input type="checkbox"/> 3) SD <input type="checkbox"/> 4) SLTP <input type="checkbox"/> 5) SLTA <input type="checkbox"/> 6) D1 <input type="checkbox"/> 7) D2 <input type="checkbox"/> 8) D3 <input type="checkbox"/> 9) D4/S1 <input type="checkbox"/> 10) S2 <input type="checkbox"/> 11) S3			
Penghasilan bulanan*)	: <input type="checkbox"/> 1) kurang dari Rp1.000.000,- <input type="checkbox"/> 2) Rp1.000.000 - Rp2.000.000,- <input type="checkbox"/> 3) > Rp2.000.000,-			
DATA WALI				
Nama Wali	: <input type="text"/> Tahun Lahir <input type="text"/>			
Pekerjaan*)	: <input type="checkbox"/> 1) tidak bekerja <input type="checkbox"/> 2) petani <input type="checkbox"/> 3) PNS/TNI/Polri <input type="checkbox"/> 4) Karyawan Swasta <input type="checkbox"/> 5) Dagang <input type="checkbox"/> 6) wiraswasta <input type="checkbox"/> 7) wirausaha <input type="checkbox"/> 8) buruh <input type="checkbox"/> 9) pensiunan			
Pendidikan**)	: <input type="checkbox"/> 1) tdk sekolah <input type="checkbox"/> 2) putus SD <input type="checkbox"/> 3) SD <input type="checkbox"/> 4) SLTP <input type="checkbox"/> 5) SLTA <input type="checkbox"/> 6) D1 <input type="checkbox"/> 7) D2 <input type="checkbox"/> 8) D3 <input type="checkbox"/> 9) D4/S1 <input type="checkbox"/> 10) S2 <input type="checkbox"/> 11) S3			
Penghasilan bulanan*)	: <input type="checkbox"/> 1) kurang dari Rp1.000.000,- <input type="checkbox"/> 2) Rp1.000.000 - Rp2.000.000,- <input type="checkbox"/> 3) > Rp2.000.000,-			
DATA PERIODIK				
Tinggi Badan	: <input type="text"/> cm Berat Badan : <input type="text"/> Kg			
Jarak tempat tinggal ke sekolah*)	: <input type="checkbox"/> 1) Kurang dari 1 Km <input type="checkbox"/> 2) Lebih dari 1 Km, sebutkan <input type="text"/> Km			
Waktu tempuh berangkat ke sekolah*)	: <input type="checkbox"/> 1) Kurang dari 30 menit <input type="checkbox"/> 2) 30 - 60 menit <input type="checkbox"/> Jika lebih dari 60 menit sebutkan <input type="text"/>			
Jumlah saudara kandung	: <input type="text"/> orang			
CATATAN PRESTASI				
Jenis Prestasi	Tingkat	Nama Prestasi	Tahun	Penyelenggara
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<p>Jenis : 1) Sains, 2) Seni, 3) Olah raga, 4) lain-lain Tingkat : 1) Sekolah, 2) Kecamatan, 3) Kab/Kota, 4) Provinsi, 5) Nasional, 6) Internasional</p> <p style="text-align: right;">Pasir Pengaraian,2014</p> <p style="text-align: center;">Orang Tua Calon Peserta Didik, Calon Peserta Didik,</p> <p style="text-align: center;">..... nama jelas & tanda tangan ctt : 1 *) Tulis dengan angka nama jelas & tanda tangan</p>				

Ketentuan Latihan :

Jenis Kelamin dan Agama menggunakan Radiobutton pada Form, Selain Jenis Kelamin dan Agama Yang ada Opsi Pilihan gunakan Select/Menu List pada Form. Untuk Tek isian (contoh : nama, tinggi dll) gunakan Textfield.

TUGAS : Buatlah sebuah layout website seperti gambar dibawah ini menggunakan element-element form.

www.euromanagement.co.id



Euro Management
"Gateway [to study] in Europe"

Kantor Pusat (Headquarters)
Gebung BPPT L 14b Floor
Jl. M.H. Thamrin No. 8
Jakarta 10340, INDONESIA
Phone : +62 21 390-38786, 707-79761, 709-44971, 792-75953
Fax : +62 21 396-38705
Email : euro@tnc-toppt.com

Member of Business Technology Center
Pusat Inovasi Teknologi (PIT) BPPT dan BTK-ACI-2003

**Formulir Pendaftaran
Program Persiapan Studi**

Program* : Jerman

Prancis

Austria

Tingkat* : S1

S2

S3

A. DATA SISWA

1. Nama Lengkap (Sesuai KTP) :
2. Tempat/Tanggal Lahir :
3. Alamat :

Kota : Kode Pos:

4. Nomor Telpon Rumah :
5. Nomor Handphone :
6. E - Mail :
7. Agama :
8. Ijazah terakhir yang dimiliki* : SMA SMK MAN Universitas
9. Asal Sekolah / Universitas* : Sekolah
- Universitas
- IPA IPS
11. Kemampuan Bahasa Jerman/ Prancis* : Tidak Ada Dasar Menengah Mahir
11. Perguruan Tinggi yang diminati* : Universitas (TU/TH/Universität)
- Fachhochschule (FH)
12. Jurusan yang diminati : 1.
2.
3.

B. DATA ORANG TUA

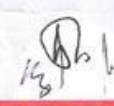
1. Nama Ayah :
2. Nama Ibu :
3. Pekerjaan Ayah :
4. Pekerjaan Ibu :
5. Alamat :

Kota : Kode Pos:

6. Nomor Telpon Rumah :
7. Nomor Handphone : Ayah : Ibu:
8. E - Mail : Ayah : Ibu:

* pilih salah satu dengan tanda

Jakarta,200.....
Tanda tangan calon siswa

(.....) 

Management Consulting • Education Counseling • Language & Academic Preparation • Cultural Orientation

Keterangan :

Elemen Form Yang Digunakan :

- a. Select/Menu List : Agama (Islam, Katolik, Protestan, Hindu, Budha), Ijazah Terakhir, Jurusan, Kemampuan Bahasa Jerman/Perancis,
 - b. Radiobutton : Perguruan Tinggi Yang Diminati sebanyak 3 buah.
- Selain Itu Gunakan Textfield. Gunakan kreatifitas Sdr/i untuk merancang form di atas.

TUGAS 2 : Rancanglah layout website seperti gambar di bawah ini menggunakan elemen-elemen form

REGISTER | Dapatkan id Playmojo untuk memainkan semua game Playmojo

User ID * **User ID tersedia**

Email * **Email tersedia**
* E-mail otentikasi yang diperlukan untuk menemukan password

Gender * Male Female

Password *


Confirm Password *

TelePhone -

Propinsi *

Question

Answer *



If you can't read the numbers in the above image, reload the page to generate a new one.

saya setuju dan telah membaca [ketentuan dan peraturan playmojo.com](#)

Ketentuan Soal :

- Gender menggunakan elemen from Radiobutton
- Propinsi dan Question menggunakan elemen from Select/List Menu
- Untuk chapcha silahkan cari di internet, bagaimana pembuatan chapcha (gambar yang bertulisan 355382)

JAVASCRIPT

A. SEJARAH TENTANG JAVASCRIPT

Pengertian JavaScript

JavaScript adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. Client Side Programming Language adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada *web browser* seperti Google Chrome dan Mozilla Firefox.

Jenis bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman *Server Side* seperti PHP, dimana untuk *server side* seluruh kode program dijalankan di sisi server.

Untuk menjalankan JavaScript, kita hanya membutuhkan aplikasi *text editor*, dan *web browser*. JavaScript memiliki fitur: *high-level programming language*, *client-side*, *loosely typed*, dan berorientasi objek.

Fungsi JavaScript Dalam Pemrograman Web

JavaScript pada awal perkembangannya berfungsi untuk membuat interaksi antara user dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di *web server*. Sebelum *javascript*, setiap interaksi dari user harus diproses oleh *web server*.

Bayangkan ketika kita mengisi *form registrasi* untuk pendaftaran sebuah situs web, lalu men-klik tombol *submit*, menunggu sekitar 20 detik untuk website memproses isian form tersebut, dan mendapati halaman yang menyatakan bahwa terdapat kolom form yang masih belum diisi.

Untuk keperluan seperti inilah JavaScript dikembangkan. Pemrosesan untuk mengecek apakah seluruh form telah terisi atau tidak, bisa dipindahkan dari *web server* ke dalam *web browser*.

Dalam perkembangan selanjutnya, *JavaScript* tidak hanya berguna untuk *validasi form*, namun untuk berbagai keperluan yang lebih modern. Berbagai animasi untuk mempercantik halaman web, fitur chatting, efek-efek modern, games, semuanya bisa dibuat menggunakan *JavaScript*.

Akan tetapi karena sifatnya yang dijalankan di sisi client yakni di dalam web browser yang digunakan oleh pengunjung situs, user sepenuhnya dapat mengontrol eksekusi JavaScript. Hampir semua web browser menyediakan fasilitas untuk mematikan JavaScript, atau bahkan mengubah kode JavaScript yang ada. Sehingga kita tidak bisa bergantung sepenuhnya kepada JavaScript.

Perkembangan JavaScript Saat Ini

Dalam perkembangannya, JavaScript mengalami permasalahan yang sama seperti kode pemrograman web yang bersifat *client side* seperti CSS, yakni bergantung kepada implementasi web browser.

Kode *JavaScript* yang kita buat, bisa saja tidak bekerja di *Internet Explorer*, karena web browser tersebut tidak mendukungnya. Sehingga programmer harus bekerja extra untuk membuat kode program agar bisa “mengakali” dukungan dari web browser.

Karena hal tersebut, JavaScript pada awalnya termasuk bahasa pemrograman yang rumit, karena harus membuat beberapa kode program untuk berbagai web browser.

Namun, beberapa tahun belakangan ini, JavaScript kembali bersinar berkat kemudahan yang ditawarkan oleh komunitas programmer yang membuat library JavaScript seperti jQuery. Library ini memudahkan kita membuat program JavaScript untuk semua web browser, dan membuat fitur-fitur canggih yang sebelumnya membutuhkan ribuan baris kode program menjadi sederhana.

Kedepannya, JavaScript akan tetap menjadi kebutuhan programmer, apalagi untuk situs saat ini yang mengharuskan punya banyak fitur modern sebagai standar.

B. ATURAN PENULISAN JAVASCRIPT

Penulisan Script Javascript :

a. Pada bagian head

Script ini akan dieksekusi ketika dipanggil (biasanya berbentuk function) atau dipanggil berdasarkan trigger pada event tertentu. Peletakan script di head akan menjamin skript di-load terlebih dahulu sebelum digunakan (dipanggil)

```
<html>
<head>
<script type="teks/javascript">
...
</script>
</head>
</html>
```

b. Pada Body

Script ini dieksekusi ketika halaman di-load sampai di bagian <body>. Ketika menempatkan script pada bagian <body> berarti antara isi dan JavaScript dijadikan satu bagian.

```
<html>
<head>
</head>
<body>
<script type="teks/javascript">
...
</script>
</body>
</html>
```

c. Pada lokasi lain

Kode JavaScript juga bisa diletakkan di file tersendiri yang berekstensi .js (singkatan dari JavaScript). Untuk memanggil kode JavaScript yang terdapat di file sendiri, di bagian awal <head> harus ditentukan dahulu nama file .js yang dimaksud menggunakan contoh kode seperti berikut:

```
<script type="teks/javascript" src="/alamat.js">
</script>
```

Penggunaan atribut `type="text/javascript"` digunakan untuk membedakan *javascript* dengan bahasa script lain seperti VBScript yang ditulis sebagai `type="text/vbscript"`.

Namun karena VBScript sudah jarang digunakan, hampir semua web browser modern menjadikan JavaScript sebagai bahasa *default*, sehingga kita tidak perlu menulis `type="text/javascript"`.

Penulisan script Javascript biasanya menggunakan perintah berikut ini :

```
<script language="text/javascript">
//kode javascript diletakkan disini
</script>
```

Atau boleh

```
<script>
//kode javascript diletakkan disini
</script>
```

2. Aturan dan tata cara mendeklarasikan Variabel

- Diawali dengan **var [nama_variabel];**
- CASE Sensitive (huruf besar/kecil dibedakan)
- Tidak boleh menggunakan spasi (sebaiknya gunakan underscore) untuk menghubungkan variabel yang terdiri dari 2 suku kata.
- Harus diawali dengan huruf/karakter (bukan angka atau operator matematika)
-

3. Aturan dan cara menampilkan text, variabel dan tag HTML

a. Menggunakan alert()

```
<script>
window.onload = function()
{
    var hasil;
    hasil = 211+79;
    alert(hasil);
}
</script>
```

b. Menggunakan document.write()

Contoh :

```
<script>
window.onload = function()
{
    document.write("Hello STTP");
}
</script>
```

c. Menggunakan console.log

```
<script>
window.onload = function()
{
    var hasil;
    hasil = 211+79;
    console.log(hasil);
}
</script>
```

d. Menggunakan Prompt

```
<SCRIPT LANGUAGE = "JavaScript">
<!--
var nama = prompt("Isikan nama Anda?","Biodata");
document.write("Halo, " + nama);
//-->
```

```
</SCRIPT>
<html>
<head>
<title>Dasar JavaScript</title>
<script>
window.onload = function()
{
    var hasil;
    hasil = 1+3+5+7+9;
    document.getElementById("tampil_disini").innerHTML=hasil;
}
</script>
</head>
<body>
<h1>Dasar JavaScript</h1>
<div id="tampil_disini">
</div> </body> </html>
```

e. Menggunakan fungsi

```
<HTML><HEAD><TITLE>Contoh Program Javascript</TITLE></HEAD>
<SCRIPT language="Javascript">
function pesan(){
    alert ("memanggil javascript lewat body onload")
}
</SCRIPT>
<BODY onload=pesan() >
</BODY>
</HTML>
```

C. OPERATOR

1. Pengenalan Variabel dalam Javascript

a) Cara mendefinisikan variable

Var nmlengkap="ichon";

Atau

Nmlengkap="ichon";

JS :

Document.write ("Nama Lengkap : " + nmlengkap);

2. Type Data Dalam Javascript

a) Numbers (Angka)

Contoh :

1

3.14

0001

6.02e23

b) Booleans

True or False

c) String

Contoh :

```
Var nama= "ABCDE";  
Var alamat ="FGHIJKLM"; dll
```

d) Objects

Contoh :

```
o.x = 1;  
o.y = 2;  
o.total = o.x + o.y;
```

e) Operator

```
+, -, *, /, ^  
&& = AND  
|| = OR
```

4. Pengenalan Event

1. onfocus="" Form field gets focus (validation)
2. onblur="" Form field loses focus (validation)
3. onchange="" Content of a field changes (validation)
4. onselect="" Text is selected
5. onmouseover="" Mouse moves over a link (animated buttons)
6. onmouseout="" Mouse moves out of a link (animated ...)
7. onclick="" Mouse clicks an object
8. onload="" Page is finished loading (initial actions, info,)
9. onSubmit="" Submit button is clicked (validation etc.)

D. STRUKTUR KENDALI DENGAN JAVASCRIPT

Dalam bahasa Javascript ada 3 model kendali logika yang digunakan, yaitu :

1. 1 Kondisi

Sintak yang digunakan

```
if (keadaan1 [operator perbandingan ] keadaan2)  
{  
    Pernyataan yang dilaksanakan;  
}
```

Contoh :

```
Var a,b;  
A=5;  
B=2;  
If ( A>B) { alert("Benar"); }
```

2. 2 Kondisi

Digunakan untuk mengkondisikan 2 pernyataan yang harus dijalankan /dipenuhi salah satunya

Sintak :

```
if (keadaan1 [operator perbandingan ] keadaan2)  
{  
    Pernyataan1 yang dilaksanakan;  
}
```

Else

```
{  
    Pernyataan2 yang dilaksanakan;  
}
```

Contoh :

Diketahui sebuah robot diinginkan bisa mendeteksi 1 keadaan yang harus dijalankan

Jika didepan ketemu jalan buntu, maka robot bisa belok kiri/kanan.

Jawab :

```
Var robot,belok_kiri,belok_kanan;  
Belok_kiri=True;  
Belok_kanan=True;  
If(robot == "ketemu jalan buntu")  
{  
    Belok_kiri;  
}  
Else { belok_kanan; }
```

3. 3 atau lebih kondisi yang dipakai

a. Menggunakan if..elseif..else..end if

Sintak :

```
If (kondisi1){ pernyataan_1; } else if (kondisi2) { pernyataan_2} else if (kondisi_n)  
{ pernyataan_n; } else { pernyataan_lainnya; }
```

Contoh :

Masih contoh yang sama (soal no.2) :

Dikembangkan logikanya :

jika dikanan juga jalan buntu, maka robot hanya bisa belok kiri/ atau mundur.

Jika robot belok kiri dalam waktu 5 detik, robot tersebut joget Sakitnya tu disini.

Jawab :

```
Var robot,belok_kiri,belok_kanan,mundur;  
Belok_kiri=True;  
Belok_kanan=True;  
Mundur=true;  
  
If(robot == "ketemu jalan buntu")  
{  
    Belok_kiri;  
}  
else if (robot == "jalan 5 detik")  
{  
    Robot="Joget";  
}  
Else  
{  
    Robot="Diam";  
}
```

b. Menggunakan switch... case ..

Sintak :

```
Switch (kondisi)  
{  
    case [ke-1] :
```

```
        Pernyataan_1;  
        Break;  
    case [ke-2]:  
        Pernyataan_2;  
        Break;  
    case [ke-n]:  
        Pernyataan_n;  
        Break;  
    default:  
        Break;  
}
```

Contoh kasus :

Sebuah perusahaan ingin membuat sebuah perangkat cerdas dengan kondisi sebagai berikut :

- Jika ditekan tombol A, maka Lampu Taman Berkedip
- Jika ditekan tombol B, maka Pagar terkunci otomatis
- Jika ditekan tombol C, maka Garasi Motor terbuka
- Jika ditekan tombol S, maka lampu rumah mati semua
- Jika ditekan tombol A bersamaan tombol B, maka Layar CCTV ruang tamu dan ruang depan ON

Jawab :

```
Var tombol, aksi ;  
If (tombol=="A")  
{  
    Aksi = "Lampu Taman Berkedip";  
}  
else If (tombol=="B")  
{  
    Aksi = "Pagar Terkunci Otomatis";  
}  
else If (tombol=="C")  
{  
    Aksi = "Garasi Motor terbuka";  
}  
else If (tombol=="S")  
{  
    Aksi = "lampu rumah mati semua";  
}  
else If (tombol=="A" && tombol=="B")  
{  
    Aksi = "Layar CCTV ruang tamu dan ruang depan ON";  
}  
Else  
{  
    Aksi="Tidak ada";  
}  
}
```

TUGAS :

CARILAH 3 BUAH KASUS YANG MENYANGKUT DENGAN SISTEM PAKAR / TEKNOLOGI KOMPUTER.
DIMINTA :

- A. BUATKAN ALGORITHMMA MASING-MASING KASUS
- B. TERAPKAN KASUS TERSEBUT KEDALAM JAVASCRIPT.
- C. CETAK OUTPUT HASIL DAN SCRIPT PROGRAM DENGAN KERTAS A4

LOOPING (PERULANGAN)

Dalam bahasa JavaScript terdapat dua struktur loop, yaitu :

1. Perintah for dan perintah while.

Untuk control pada struktur loop, JavaScript menyediakan perintah break dan continue.

Syntax :

```
for ([var_awal;] [kondisi;] [var_awal_tambah_atau_kurang])
{
    pernyataan
}
```

Dimana :

- Var_awal = variabel yang digunakan untuk melakukan perulangan
- Kondisi = operator perbandingan yang digunakan (> <, >=, <=)
- Var_awal+/- = variabel perulangan dibaca secara menaik atau menurun.

Contoh :

```
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
  <!--
    for (i=1; i<=10; i++)
      {
        bil=i*i
        document.write("Angka ke: " + i + " Adalah : " + bil + "<BR>")
      }
  //-->
</SCRIPT>
</HEAD><BODY></BODY></HTML>
```

Jika diterapkan kedalam data Array, maka fungsi for ...harus diikuti dengan perintah length, seperti contoh berikut :

Contoh :

```
< HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
  <!--
    Array_huruf= ['A', 'B', 'C'];
    for (i = 0; i < Array_huruf.length; i++) {
      console.log(Array_huruf [i]);
    }
  //-->
</SCRIPT>
</HEAD><BODY></BODY></HTML>
```

Jika data lebih besar, maka sebaiknya kita gunakan perintah berikut untuk mengakses array.

```
< HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
  <!--
```

```
Array_huruf= ['A', 'B', 'C'];  
for (i = 0; i < Array_huruf.length;i<j; i++) {  
  console.log(Array_huruf [i]);  
}  
  //-->  
</SCRIPT>  
</HEAD><BODY></BODY></HTML>
```

2. Perintah while

Syntax :

```
while (kondisi)  
{  
  pernyataan  
}
```

Contoh:

```
<HTML><HEAD>  
<SCRIPT LANGUAGE="JavaScript">  
  <!--  
    while (i<=10)  
    {  
      bil=i*i  
      document.write("Angka ke: " + i + " Adalah : " + bil + "<BR>")  
    }  
  
  //-->  
  
</SCRIPT>  
</HEAD><BODY></BODY></HTML>
```

3. Perintah do...while

Syntax :

```
Do  
{  
  Pernyataan;  
}  
While (kondisi)
```

Contoh :

```
var i = 10;  
do  
{  
  i++;  
  console.log(i);  
}  
while (i < 10)
```

4. Perintah for ...in

Tidak berbeda dengan FOR Loop sebelumnya, perintah ini biasanya dipakai untuk menyebutkan properti dari suatu Objek satu persatu.

PRAKTIKUM : simpan dengan nama file Latihan_Perulangan.html

```
<HTML><HEAD>  
<SCRIPT LANGUAGE="JavaScript">  
<!--
```

```
nama = {
  firstName: 'Tito',
  middleName: 'Pandu',
  lastName: 'Brahmanto'
};
for (i in nama) {
  console.log(i + ' : ' + nama[i]);
}

//-->
</SCRIPT>
</HEAD><BODY></BODY></HTML>
<HTML><HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
function cetak(n)
{
  for(var i = 1; i <= n; i++){
    document.write(i+" ");
  }
}
function CetakAngka(M)
{
  var j = 1;
  while(j <= M)
  {
    cetak(j);
    document.write("<br />");
    j++;
  }
}

function CetakAngka2(M)
{
  var tengah = Math.round(M / 2);
  var j = 1;
  while(j <= M){
    if(j < tengah){
      var dicetak = j;
    }
    else {
      dicetak = ditengah;
      ditengah--;
    }
    cetak(dicetak);
    document.write("<br />");
    j++;
  }
}
// buatkan sebuah fungsi baru dari latihan dibawah
function perulangan()
{
  // jawaban latihan
}
</script>
```

```
        <div class='data'>
<script>
    CetakAngka(5);
</script>

</div>
        <div class='data'>
<script>
    CetakAngka2(7)
</script>
</div>

<style>
.data
{
    border: 1px solid green;
    margin:15px;
    padding:6px 0px 6px 12px;
    font-size:0.7em;
    color:#0000aa;
    background: #fff0aa;
    width:100px;
    float: left;
}
.data:hover
{
    border: 2px solid blue;
    font-size:0.8em;
    color:#0099ff;
    background: #f1ffaa;
}
</style>
</HEAD>
<BODY on load ="perulangan()"></body></html>
```

TUGAS 1 :

Pada praktikum di atas terdapat sebuah fungsi untuk menampilkan perintah-perintah perulangan, yaitu :

```
function perulangan()
{
} .
```

PERTANYAAN : lengkapi fungsi perulangan() tersebut dengan soal-soal dibawah ini :

```
function perulangan_1()
{
// didalam fungsi perulangan () ini buatlah program javascript untuk menampilkan :
a. Bilangan sbb :
    1    =>   1
    2    =>   3
    3    =>   5
    4    =>   7
    5    =>   9
```

b. Bilangan sbb :

11111
22222
33333
44444
55555
1
12
123

c. Bilangan sbb :

12345
1234
123
12
1
0 0 0 0 0
0 0 0 0
0 0 0
0 0
0

// gunakan perintah document.write → untuk menampilkannya

}

TUGAS 2 : Buatlah sebuah program javascript untuk menampilkan perulangan pada soal dibawah ini :

```
function perulangan_2()
```

```
{
```

```
// didalam fungsi ini buatlah program javascript perulangan untuk menampilkan :
```

a. Tampilkan Bilangan sbb :

Jika 1, maka 4
Jika 2, maka 7
Jika 3, maka 12
Jika 4, maka 19
Jika 5, maka 28
Jika 6, maka 39
Jika 7, maka 53
Jika 8, maka 67

Kata kunci : *self, times, add, three*

b. Bilangan sbb :


```
55555
4444
333
22
333
4444
55555
```

c. Bilangan sbb :

```
0 0 0 0
0 0 0
0 0
0 0 0
0 0 0 0
0 0 0 0 0
X X X X
X X X
X X
X
}
```

TUGAS 3: Buatlah sebuah program javascript untuk menampilkan perulangan pada soal dibawah ini :

```
function perulangan_3()
{
    // didalam fungsi ini buatlah program javascript perulangan untuk menampilkan :
    a. Bilangan deret sbb :
        1 , 2, 2, 4, 3, 6, 4, 8, 5, 10, 6, 12
        2, 4, 8, 16,32,64
    b. Bilangan sbb :

        55555
        4444
        333
        22
        333
        4444
        55555
    c. Bilangan sbb :
        12345
        1234
        123
        12
        1
    // gunakan perintah document.write → untuk menampilkannya
}
```

Document Object Model (DOM)

Document Object Model (DOM) merupakan sebuah ketentuan yang dikembangkan oleh W3C untuk berinteraksi dengan objek-objek yang ada di dalam HTML, XML, maupun XHTML. DOM bersifat *cross-platform* dan *language-independent*, yang artinya DOM dapat digunakan dengan bahasa pemrograman apapun, dalam sistem operasi manapun. Standar DOM dikembangkan untuk berinteraksi dengan elemen-elemen dokumen HTML dan XML, mulai dari pembuatan elemen baru sampai dengan manipulasi dan penghapusan elemen.

a. STRUKTUR ELEMEN DOM

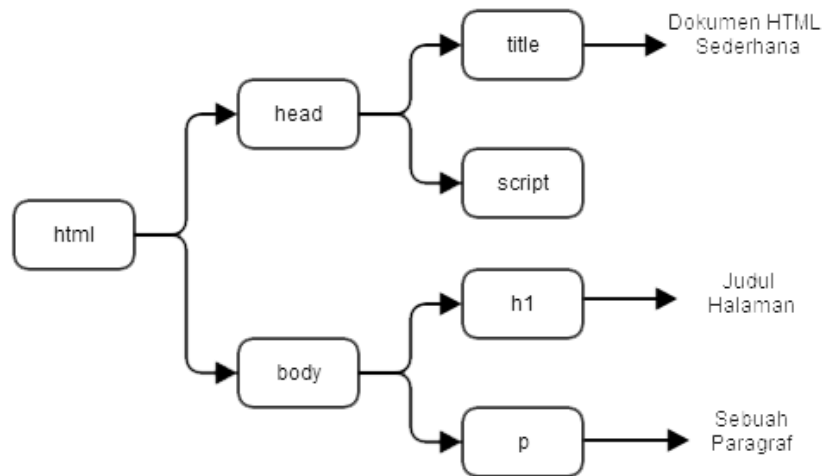
Sebuah dokumen HTML direpresentasikan oleh DOM dalam bentuk sebuah struktur hirarkis pohon.

Contoh dokumen HTML sederhana seperti berikut:

```
<html>
<head>
<title>HTML Sederhana</title>
<script type="text/javascript" src="scripts/scripts.js">
</script>
</head>
  <body>
    <h1>Judul</h1>
    <p>paragraf</p>
  </body>
</html>
```

Pada contoh kode di atas, kita dapat melihat bagaimana di dalam tag `html` terdapat banyak elemen-elemen lain, dan elemen lain dapat mengandung elemen lain lagi di dalamnya. Elemen `html`, yang menjadi penampung seluruh elemen lain dianggap sebagai akar (*root*) dari pohon, dan setiap elemen di dalamnya sebagai *leaf* (daun).

Representasikan kode HTML tersebut sebagai pohon sesuai dapat dilihat pada gambar berikut:



HTML direpresentasikan Sebagai Pohon

Manipulasi DOM

Sebelum mulai melakukan manipulasi terhadap elemen-elemen DOM, tentunya kita harus terlebih dahulu mampu memilih mana elemen yang akan kita ubah propertinya. Terdapat beberapa fungsi yang dapat kita gunakan untuk mengakses elemen DOM, yaitu:

1. **document.getElementById**, digunakan untuk mengakses elemen DOM berdasarkan nilai properti id pada elemen. Hanya akan mengembalikan satu elemen DOM karena idealnya nilai properti id bersifat unik dalam satu dokumen. Jika terdapat lebih dari satu elemen dengan id yang sama, maka elemen yang pertama ditemukan akan dikembalikan.
2. **document.getElementsByTagName**, digunakan untuk memanggil seluruh elemen DOM dengan nilai properti class yang diberikan. Mengembalikan objek mirip array, yaitu HTMLCollection dari seluruh elemen DOM yang cocok.
3. **document.getElementsByName**, fungsinya untuk mengambil elemen berdasarkan nilai properti name.
4. **document.getElementsByTagName**, memilih elemen-elemen dengan tag HTML tertentu.
5. **document.getElementsByTagNameNS**, sama seperti **getElementsByTagName**, hanya saja fungsi ini melakukan penyaringan tambahan berdasarkan namespace pada dokumen XHTML (element ini jarang digunakan).
6. **document.querySelector**, fungsinya untuk mencari elemen DOM pertama yang sesuai dengan aturan selector CSS yang diberikan ke fungsi.
7. **document.querySelectorAll**, sama seperti **querySelector**, tetapi mengembalikan semua elemen yang memenuhi aturan (bukan hanya elemen pertama). Fungsi ini mengembalikan NodeList, bukan HTMLCollection.

```
<html lang="id">
<head>
<title>Akses Elemen DOM</title>
</head>
<body>
<div id="container">
    <h1 class="heading">Judul Pertama</h1>
    <p>Paragraf Pertama</p>
    <p>Paragraf Kedua</p>
    <p>Paragraf Ketiga</p>
    <h2 class="heading">Judul Kedua</h2>

    <form action="post" id="comments">
    <input type="text" name="comment-title" id="comment-title"><br>
    <textarea name="comment-box" id="comment-box" class="box">
    </textarea><br>
    <input type="reset" class="button">
    <input type="submit" class="button submit">
    </form>
    </div>
    <script type="text/javascript" src="script.js"></script>
</body>
</html>
```

Kita dapat mengakses salah satu elemen dalam dokumen, contohnya elemen `input` untuk judul komentar (id: "comment-title") menggunakan fungsi `getElementById`:

```
document.getElementById("comment-title")
```

Pemanggilan fungsi di atas akan memberikan kita elemen HTML yang benar, dibungkus dalam objek `Element`. Objek `Element` digunakan untuk merepresentasikan sebuah elemen DOM dalam dokumen HTML, dan memiliki beberapa *method* dan properti standar. *Method* dan properti standar ini tidak akan kita bahas semuanya langsung, tetapi sedikit demi sedikit, sesuai dengan kebutuhan. Jika ingin melihat daftar lengkap dari properti dan *method*, silahkan kunjungi dokumentasi yang relevan pada MDN (Mozilla Developer Network).

Karena `getElementById` (dan fungsi-fungsi sejenis lain) mengembalikan objek, maka kita dapat menyimpan hasil eksekusi fungsi ke dalam variabel, layaknya objek-objek lain:

```
var ct = document.getElementById("comment-title");

ct.getAttribute("type"); // mengembalikan "text"
ct.getAttribute("name"); // mengembalikan "comment-title"
```

Pada contoh di atas, kita memanggil *method* milik `Element`, yaitu `getAttribute` yang berguna untuk mengambil nilai dari atribut yang dimiliki elemen. Sebagai objek khusus untuk representasi DOM, `Element` juga ditampilkan secara khusus pada *developer tools* yang ada dalam *browser*:

```
> var ct = document.getElementById("comment-title")
undefined
> ct
<input type="text" name="comment-title" id="comment-title">
> ct.getAttribute("type")
"text"
> ct.getAttribute("name")
"comment-title"
> var test = "string"
undefined
> test
"string"
```

Tampilan **Element** dalam Chrome Developer Tools

Dari gambar di atas, dapat dilihat bagaimana variabel **test** yang berisi string ditampilkan hanya sebagai string, sementara **ct** ditampilkan dalam bentuk *tag* HTML. Kita dapat memanfaatkan tampilan ini ketika *debugging* untuk memastikan bahwa elemen yang kita ambil sudah benar.

Perhatikan juga bagaimana kita memanggil fungsi **getAttribute** terhadap **ct**, bukan **document**. Objek **document** pada DOM merepresentasikan keseluruhan dokumen HTML kita, sementara **ct** menandai elemen spesifik. Karena kita ingin mengambil atribut hanya pada elemen spesifik, maka kita memanggil **getAttribute** hanya pada **ct**. Pastikan anda mengerti perbedaan kedua objek ini agar dapat mengerti bagian-bagian berikutnya dengan mudah.

Selanjutnya, untuk mengambil elemen DOM berdasarkan nama **class**, kita dapat menggunakan fungsi **getElementsByClassName**:

```
var headings = document.getElementsByClassName("heading");
```

```
headings[0]; // elemen h1 pada dokumen
headings[1]; // elemen h2 pada dokumen
```

Fungsi **getElementsByClassName** akan mengembalikan **HTMLCollection**, yang sifatnya sekilas mirip dengan sebuah *array*. Perbedaan utama adalah bahwa **HTMLCollection** memiliki tambahan dua buah *method*, yaitu **item** dan **namedItem**. Seperti yang dapat ditebak, fungsi kedua *method* tersebut adalah untuk mengambil elemen yang tepat, dari indeks (**item**) maupun nama (**namedItem**).

```
var buttons = document.getElementsByClassName("button");
buttons.item(0); // sama dengan button[0]
button.namedItem("btn-submit"); // button submit (cek atribut "name")
```

Sama seperti **Element**, **HTMLCollection** juga ditampilkan secara khusus pada *developer tools* dalam *browser*.

```
> var headings = document.getElementsByClassName("heading");  
undefined  
> headings  
[<h1 class="heading">Judul Pertama</h1>, <h2 class="heading">Judul Kedua</h2>]  
> headings[0]  
<h1 class="heading">Judul Pertama</h1>  
> headings[1]  
<h2 class="heading">Judul Kedua</h2>  
> headings[0].innerHTML  
"Judul Pertama"
```

Tampilan `HTMLCollection` dalam Chrome Developer Tools

Fungsi-fungsi selanjutnya, yaitu `getElementsByName` dan `getElementsByTagName` memiliki cara penggunaan yang sama dengan `getElementsByClassName`:

```
var paragraphs = document.getElementsByTagName("p");  
paragraphs.length; // mengembalikan 3  
paragraphs.item(0); // mengembalikan paragraf pertama  
var ct = document.getElementsByName("comment-title");  
ct[0]; // mengembalikan kotak judul komentar
```

Selain semua fungsi di atas, kita juga dapat menggunakan aturan *selector* CSS untuk mengambil elemen DOM, melalui fungsi `querySelector` dan `querySelectorAll`. `querySelector` mengambil hanya satu elemen pertama yang memenuhi syarat *selector* CSS, sementara `querySelectorAll` mengambil semua elemen yang memenuhi aturan.

```
// mengambil semua elemen dengan class "heading"  
// dan tag p  
var els = document.querySelectorAll(".heading, p");  
  
els.length; // mengembalikan 5  
els.item(0); // elemen h1.heading  
  
// mengembalikan hanya h1.heading saja, karena  
// elemen tersebut merupakan elemen pertama yang  
// memenuhi syarat selector CSS yang dikirimkan  
document.querySelector(".heading");
```

Ingat juga bahwa `querySelectorAll` mengembalikan `NodeList`, bukan `HTMLCollection`. Perbedaan utama antara kedua objek ini adalah bahwa `NodeList` dapat menampung objek selain dari `Element`, dan `NodeList` tidak memiliki `method` `namedItem`. Terdapat beberapa perbedaan lain secara internal, tetapi tidak akan kita bahas lebih dalam karena tidak terlalu banyak perbedaan yang tampak pada penggunaan umum.

Membuat Elemen DOM

Seperti yang dijelaskan di bagian awal, DOM memiliki dua jenis elemen, yaitu elemen `node` dan teks. Karenanya, pembuatan elemen baru untuk DOM juga dilakukan dengan dua cara, yaitu `document.createElement` untuk pembuatan `node`, dan `document.createTextNode` untuk pembuatan teks.

```
var paragraf = document.createElement("p");  
var teks = document.createTextNode("Teks yang dibuat secara dinamis");
```

Sampai titik ini, `paragraf` masih merupakan sebuah elemen paragraf yang tidak memiliki isi apa-apa. Kita dapat memasukkan `teks` ke dalam `paragraf` dengan menggunakan `method` `appendChild`:

```
paragraf.appendChild(teks);
```

Sekarang kita telah memiliki sebuah elemen `p` berisi teks "Teks yang dibuat secara dinamis". Setelah memiliki elemen baru, tentunya kita harus memasukkan elemen tersebut ke dalam dokumen HTML sebelum elemen baru dapat dilihat oleh pengguna. Untuk memasukkannya, kita masih dapat menggunakan `appendChild`, pada bagian dokumen yang kita inginkan. Misalnya kita ingin memasukkan teks di dalam `div` dengan id `container`:

```
var container = document.getElementById("container");  
container.appendChild(paragraf);
```

Setelah menambahkan elemen paragraf seperti pada kode di atas, kita dapat melihat bagaimana sebuah teks paragraf baru bertambah pada **akhir** `div`:



Halaman Sebelum dan Setelah Penambahan Elemen

Selain menambahkan elemen pada akhir seperti ini, kita juga dapat menambahkan elemen **di depan** sebuah elemen lainnya melalui fungsi `insertBefore`. Fungsi `insertBefore` menerima dua parameter: parameter pertama berupa elemen yang ingin ditambahkan, sementara parameter kedua adalah tempat di mana parameter pertama diletakkan. Agar tidak bingung mari kita langsung lihat contohnya. Misalkan `paragraf` ingin diletakkan sebelum “Judul Pertama” sehingga hasil yang diinginkan adalah sebagai berikut:



Paragraf Dimasukkan Sebelum “Judul Pertama”

Maka kita dapat menggunakan `insertBefore` seperti ini:

```
var judulPertama = document.querySelector("h1.heading");
container.insertBefore(paragraf, judulPertama);
```

Cara penambahan elemen baru ke dalam DOM memang cukup sederhana. Cara penambahan DOM dengan fungsi `appendChild` maupun `insertBefore` secara otomatis akan membuat *browser* menyusun ulang elemen-elemen yang ada dalam dokumen HTML, untuk memastikan dokumen ditampilkan dengan benar. Proses penyusunan ulang ini kita kenal dengan istilah *reflow*.

Setiap kali kita memanggil `appendChild` ataupun `insertBefore`, *browser* akan melakukan *reflow*. Pada penambahan satu atau dua elemen hal ini tidak terlalu berpengaruh, tetapi jika kita menambahkan banyak elemen, performa dari *browser* akan menjadi lambat. Pergeseran banyak elemen berkali-kali juga tentunya akan membuat pengguna tidak nyaman menggunakan halaman web kita.

```
var link;

for (var i = 0; i < 10; i++) {
  link = document.createElement('a');
  link.innerHTML = 'test';
  container.appendChild(link);
}
9
```

Solusi dari permasalahan ini yaitu dengan menggunakan `DocumentFragment`. `DocumentFragment` merupakan sebuah objek yang menampung sepotong dokumen. Objek ini dibuat dengan tujuan agar kita dapat membangun elemen DOM kompleks yang memiliki banyak komponen. Setelah membangun elemen dengan banyak komponen ke dalam `DocumentFragment` kita kemudian dapat menambahkan semua elemen tersebut ke dalam dokumen HTML sekaligus. Dengan cara ini *browser* hanya akan melakukan satu kali *reflow*, sehingga tidak ada penalti performa.

`DocumentFragment` dapat dibuat melalui fungsi `createDocumentFragment` seperti berikut:

```
var link, fragment = document.createDocumentFragment();

for (var i = 0; i < 10; i++) {
  link = document.createElement('a');
  link.innerHTML = 'test';
  fragment.appendChild(link);
}
```

```
}  
  
// reflow hanya terjadi di sini  
element.appendChild(fragment);
```

Perhatikan bagaimana `createDocumentFragment` dipanggil melalui objek `document`. Hal ini dikarenakan `DocumentFragment` yang dibuat oleh `createDocumentFragment` masih kosong, sehingga bisa dikatakan fungsi ini memiliki aturan main yang sama dengan `createElement`, tetapi menghasilkan objek yang berbeda.

Menghapus Elemen DOM

Penghapusan elemen DOM dilakukan dengan sangat sederhana, hanya melalui satu fungsi saja: `removeChild`. Prosesnya sederhana: pilih elemen yang akan dihapus, kemudian panggil `removeChild` pada penampung elemen yang akan dihapus tersebut:

```
var judulKedua = document.querySelector("h2.heading");  
container.removeChild(judulKedua);
```

Berbeda dengan penambahan elemen, tidak terdapat kasus khusus untuk penghapusan banyak elemen DOM. Pada prakteknya, kita juga sangat jarang membuang banyak elemen DOM sekaligus.

Modifikasi Atribut DOM

Inti dari pengembangan dokumen HTML yang interaktif adalah perubahan konten maupun atribut dari DOM. Misalnya jika ingin membuat sebuah elemen *text area* bertambah besar ketika pengguna menekan `Enter`, kita dapat menambahkan nilai atribut `row` pada *text area*. Kebanyakan interaksi dan dinamisme dokumen HTML dilakukan dengan cara seperti ini. Pertanyaannya tentunya adalah: bagaimana kita mengubah atribut-atribut yang ada pada DOM?

Pada bagian sebelumnya kita telah melihat fungsi `getAttribute` yang mengambil nilai atribut DOM. Untuk mengganti atau menambahkan nilai atribut DOM, kita dapat menggunakan fungsi kakaknya, yaitu `setAttribute`. Fungsi `setAttribute` menerima dua parameter, yaitu atribut yang ingin ditambahkan sebagai parameter pertama, dan isi dari atribut tersebut sebagai parameter kedua. Langsung saja, berikut contoh penggunaan fungsi ini:

```
var textbox = document.getElementById("comment-title");  
  
textbox.setAttribute("size", 40);
```

```
// textbox menjadi:  
// <input type="text" name="comment-title" id="comment-title" size="40">  
textbox;
```

Kita juga dapat menggunakan `setAttribute` untuk atribut yang sudah ada. Jika digunakan pada atribut yang sudah ada `setAttribute` akan memperbaharui nilai atribut pada parameter pertama:

```
textbox.setAttribute("size", 30);  
  
// textbox menjadi:  
// <input type="text" name="comment-title" id="comment-title" size="30">  
textbox;
```

Selain dapat menambahkan atribut baru, DOM juga memungkinkan kita untuk menghapus atribut yang telah ada. Fungsi untuk melakukan penghapusan atribut yaitu `removeAttribute`. Fungsi `removeAttribute` hanya menerima satu parameter, yaitu atribut yang ingin dihapus. Berikut adalah cara penggunaan fungsi ini:

```
textbox.removeAttribute("size");  
  
// textbox menjadi:  
// <input type="text" name="comment-title" id="comment-title">  
textbox;
```

Penambahan, perubahan, dan penghapusan atribut merupakan cara yang sederhana dan efektif untuk menambahkan sedikit efek dinamis pada dokumen HTML. Tetapi tentunya metode ini tidak memberikan kita terlalu banyak kebebasan untuk membangun dokumen HTML yang dinamis. Ingat kembali, *tag* HTML beserta atributnya dirancang hanya sebagai kode untuk merepresentasikan makna semantik dokumen. Tampilan dari dokumen idealnya kita perindah melalui CSS. Karenanya, jika ingin mengembangkan dokumen interaktif yang juga cantik (dengan berbagai animasi dan perubahan tampilan) kita perlu mengubah juga CSS yang mempengaruhi DOM. Kita akan melihat cara untuk melakukan hal ini pada bagian berikutnya.

Mengganti Tampilan Elemen DOM

Jika pada bagian sebelumnya kita telah melihat bagaimana mengganti atribut elemen DOM, maka pada bagian ini kita akan melihat cara menggantikan atribut CSS pada elemen DOM. Salah satu cara yang paling sederhana untuk mengganti properti CSS adalah dengan memanfaatkan atribut `style` pada sebuah elemen. Karena atribut `style` akan menerapkan properti CSS sesuai dengan nilai yang diberikan, kita dapat langsung menggantikan nilainya melalui `setAttribute`:

```
var container = document.getElementById("container");  
container.setAttribute("style", "background: #D33");
```

Cara penggantian kode CSS seperti ini, meskipun mudah dilakukan dan memberikan hasil yang benar, sangat tidak disarankan untuk kita gunakan. Bayangkan jika kita ingin menambahkan satu lagi properti CSS setelah menambahkan properti dengan cara di atas. Mau tidak mau, kita harus mengambil nilai atribut `style` kembali, dan kemudian mengubah isi nilai tersebut, dan mengirimkannya kembali melalui `setAttribute` seperti ini:

```
var cssLama = cont.getAttribute("style");  
var cssBaru = cssLama + "; color: white;";  
cont.setAttribute("style", cssBaru);
```

Dan sekarang bayangkan lagi jika kita ingin mengganti properti CSS `background` menggunakan `setAttribute`. Modifikasi nilai CSS seperti ini tentunya tidak optimal, dan akan sangat menyulitkan kita di masa depan. Untungnya, DOM menyediakan cara yang lebih mudah untuk menggantikan nilai CSS yang melekat pada sebuah elemen, yaitu melalui properti `style`.

Properti `style` dapat dipanggil pada semua elemen DOM yang juga merupakan elemen HTML. `style` termasuk sebagai salah satu properti dari antarmuka (*interface*) `HTMLElement`. Ketika kita menggunakan `style`, *browser* akan mengimplementasikan nilai CSS yang diberikan secara *inline*, yang artinya hasil akhir yang didapatkan akan sama dengan mengubah atribut `style` secara langsung.

Berikut adalah contoh penggunaan atribut `style` untuk mengubah properti CSS dari sebuah elemen:

```
var container = document.getElementById("container");  
container.style.background = "#D33";  
container.style.color = "white";  
// jika ingin mengubah nilai background lagi  
container.style.background = "#D37";
```

Dan karena sifat dinamis dari Javascript, kita dapat menyimpan properti `style` dari sebuah elemen ke dalam variabel, dan kemudian mengubah nilainya di sana:

```
var CSSContainer = container.style;  
CSSContainer.padding = "1em";
```

Pengambilan nilai dari CSS yang ada juga dapat dilakukan dengan mudah:

```
1 var warnaTeks = CSSContainer.color;
```

Salah satu kekurangan dari pengambilan nilai properti CSS melalui `style` seperti ini adalah bahwa kita hanya dapat mengambil nilai properti CSS satu per satu. Selain itu, kita juga tidak dapat mengambil nilai properti CSS yang diterapkan pada *pseudo-element* seperti `:hover`.

Untuk menutupi kedua kekurangan tersebut, kita dapat menggunakan fungsi `window.getComputedStyle`. Fungsi ini akan mengembalikan seluruh properti CSS yang ada pada sebuah elemen sekaligus. `window.getComputedStyle` menerima dua parameter, yaitu elemen yang ingin diambil kode CSS-nya sebagai parameter pertama, dan parameter kedua yang berisipseudo-element yang akan diterapkan pada elemen di parameter pertama. Parameter kedua tidak harus diisikan, dan jika tidak diisikan akan memberikan seluruh CSS pada elemen di parameter pertama saja.

Fungsi `window.getComputedStyle` akan mengembalikan sebuah objek bertipe `CSSStyleDeclaration`, yang menampung seluruh properti CSS yang diterapkan pada sebuah elemen, beserta dengan nilai dari masing-masing propertinya. Objek `CSSStyleDeclaration` sendiri memiliki banyak *method* dan properti, yang detailnya dapat dibaca pada dokumentasi yang relevan. *Method* yang akan kita gunakan pada contoh nanti hanya satu, yaitu `getPropertyValue`, yang sesuai namanya mengambil nilai dari properti CSS yang relevan.

Langsung saja, berikut contoh penggunaan dari `window.getComputedStyle`:

```
var container = document.getElementById("container");
var propertiCSS = window.getComputedStyle(container, null);
// Mengembalikan rgb(255, 255, 255)
var warnaTeks = propertiCSS.getPropertyValue("color");
```

Pada contoh kode di atas, `propertiCSS` merupakan objek `CSSStyleDeclaration` yang menampung seluruh properti CSS yang ada, dan diterapkan pada `container`. Dengan mengambil seluruh properti CSS yang ada sekaligus, kita dapat melakukan perhitungan kompleks terhadap nilai-nilai properti yang ada tanpa perlu terlalu banyak kode berulang. Misalnya kita dapat menghitung dan menyamakan jarak margin per elemen yang ada dalam halaman tanpa perlu memanggil `style` berkali-kali.

Mengganti Tampilan DOM dengan Kelas CSS

Sejauh ini kita telah melihat bagaimana kita dapat mengakses dan mengganti properti CSS pada elemen DOM dengan menggunakan `style` maupun `getComputedStyle`. Walaupun sangat berguna, kedua metode awal ini memiliki satu kekurangan yang cukup buruk, yaitu bahwa cara seperti ini mencampur adukkan bagian logis kode kita (javascript) dengan bagian penampilan (CSS).

Ingat kembali bahwa tujuan dari pemisahan HTML-CSS-Javascript adalah untuk mendapatkan kerapian kode dengan membuat masing-masing komponen memiliki tujuan khusus. Pergantian tampilan secara langsung melalui javascript seperti ini akan mengakibatkan kode kita tercampur aduk antara kode logis dan penampilan. Belum lagi perubahan nilai pada `style` yang akan selalu menghasilkan *reflow*. Kode seperti ini:

```
element.style.fontWeight = 'bold';  
element.style.textDecoration = 'none';  
element.style.color = '#000';
```

Akan memaksa *browser* menjalankan **tiga kali *reflow***! Oke, sekarang kita telah melihat masalahnya. Bagaimana dengan solusi dari masalah ini?

Salah satu solusi terbaik pada kasus ini adalah dengan melakukan perubahan terhadap atribut `class` pada elemen secara langsung. Ide utamanya adalah bahwa kita menyimpan seluruh kode tampilan dalam sebuah kelas CSS, dan jika ingin mengubah tampilan elemen DOM kita dapat mengubah kelas CSS-nya sesuai dengan penampilan yang kita inginkan.

Bayangkan kombinasi HTML dan CSS berikut (kita tidak memisahkan kode CSS ke dalam *file* sendiri di sini untuk menyederhanakan contoh):

```
<!DOCTYPE HTML>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Class List CSS</title>  
  
  <style type="text/css">  
    .teksWah {  
      color: #D55;  
      font-size: 2em;  
      font-style: italic;  
    }  
  
    .teksWew {  
      color: #55D;  
      font-size: 1.5em;  
    }  
  </style>  
</head>  
<body>  
  <h1 id="teks">Sebuah teks sederhana</h1>  
</body>  
</html>
```

Kita dapat melihat bahwa terdapat dua buah kelas CSS pada kode di atas, yaitu `teksWah` dan `teksWew`. Masing-masing dari kelas ini memiliki properti CSS yang berbeda-beda,

dan kita akan mencoba mengubah tampilan pada `h1#text` dengan menambahkan dan mengurangi kelas CSS.

Pertama-tama, pastinya kita harus mengambil elemen yang relevan terlebih dahulu:

```
var teks = document.getElementById("teks");
```

Selanjutnya, kita dapat mengakses seluruh `class` yang ada pada suatu elemen dengan menggunakan properti `classList`. Karena elemen kita belum memiliki kelas, maka kita akan menyimpan `classList` ke dalam sebuah variabel terlebih dahulu.

```
var cssTeks = teks.classList;
```

Kita kemudian dapat menambahkan atau menghapus `class` CSS dari `teks` dengan menggunakan `method` `add` dan `remove`, yang cukup jelas kegunaannya dari namanya.

```
// teks menjadi memiliki class teksWew  
// tampilan otomatis berubah  
cssTeks.add("teksWew");
```

```
// kelas teksWew dihilangkan dari teks  
// tampilan kembali ke awal  
cssTeks.remove("teksWew");
```

Jika ingin membalikkan keaktifan `class` (jika tidak ada jadikan ada; jika ada jadikan tidak ada) kita dapat menggunakan `method` `toggle`:

```
cssTeks.toggle("teksWah"); // kelas teksWah ditambahkan (karena belum ada)  
cssTeks.toggle("teksWah"); // kelas teksWah dihilangkan (karena sudah ada)
```

Terakhir, kita juga dapat mengecek keberadaan sebuah `class` dalam elemen DOM dengan menggunakan `method` `contains` milik `classList`:

```
cssTeks.contains("teksWew"); // mengembalikan false  
cssTeks.add("teksWew");  
cssTeks.contains("teksWew"); // mengembalikan true
```

Dengan memanfaatkan fitur-fitur yang disediakan oleh `classList` ini, kita dapat mengubah tampilan dari sebuah elemen DOM dengan mudah dan rapi, serta hanya menjalankan satu kali `reflow`. Idealnya kita harus selalu mencoba untuk menggunakan cara ini dibandingkan dengan `style` maupun `getComputedStyle`.

Sampai di sini kita telah melihat bagaimana kita dapat melakukan perubahan terhadap tampilan elemen DOM melalui kode program. Tentunya hal ini tidak terlalu berguna, karena yang sebuah dokumen HTML interaktif disebut “interaktif” adalah karena adanya intervensi pengguna. Mungkin

kita ingin teks berubah warna ketika pengguna menekan tombol tertentu. Atau mungkin gambar membesar ketika di-klik. Untuk mencapai efek seperti itu kita harus menggunakan teknik perubahan tampilan yang baru saja kita pelajari berbarengan dengan *event* DOM.

Kita akan melakukan pembahasan mengenai *event* DOM pada bab berikutnya. Sebelum masuk ke *event* DOM, kita akan melihat penelusuran DOM terlebih dahulu, karena tidak terlalu berguna jika kita hanya mampu mengganti tampilan satu dari elemen DOM saja. Lebih baik jika kita dapat menelusuri seluruh elemen DOM, dan mengganti tampilannya ketika diperlukan.

Penelusuran Elemen DOM

Ketika membuat dokumen HTML dinamis, seringkali kita perlu menambahkan atau menghapus elemen-elemen DOM yang tidak kita ketahui dengan pasti. Misalnya, bisa saja kita ingin menghapus semua baris dari sebuah tabel yang memiliki nilai tertentu pada kolom pertamanya. Untuk melakukan hal-hal seperti ini kita perlu menelusuri elemen DOM yang ada, selangkah demi selangkah. Pada contoh penghapusan baris tabel tadi misalnya, kita dapat mengambil tabel (misalnya berdasarkan id) dan kemudian menelusuri isi dari tabel tersebut.

Langsung saja, andai kata kita memiliki tabel seperti berikut:

```
<table id="tabel-contoh">
  <thead>
    <tr>
      <th>Judul 1</th>
      <th>Judul 2</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Isi 1</td>
      <td>Isi 2</td>
    </tr>
    <tr>
      <td>Isi 3</td>
      <td>Isi 4</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Total 1</td>
      <td>Total 2</td>
    </tr>
  </tfoot>
</table>
```


Untuk melakukan penelusuran elemen-elemen dalam tabel di atas, pertama-tama tentunya kita harus mengambil tabelnya terlebih dahulu:

```
var tabel = document.getElementById("tabel-contoh");
```

Setelah mendapatkan elemen DOM dari tabel, kita kemudian dapat mengambil seluruh elemen DOM yang beradasetingkat di bawah tabel dengan memanggil properti `childNodes`:

```
// mengembalikan  
// [#text, <thead>...</thead>, #text,  
// <tbody>...</tbody>, #text, <tfoot>...</tfoot>, #text]  
tabel.childNodes;
```

Perhatikan bagaimana selain mendapatkan `thead`, `tbody`, dan `tfoot` kita juga diberikan tiga buah `#text`. Objek `#text` ini merupakan konten teks yang juga dianggap bagian dari DOM, dan berisi spasi dan baris baru yang kita berikan untuk merapikan kode HTML. Jadi misalnya jika kita menuliskan HTML tanpa spasi (sering dijumpai pada HTML hasil kompilasi):

```
<table id="tabel-contoh"><thead><tr><th>Judul 1</th><th>Judul 2</th>  
</tr></thead><tbody><tr><td>Isi 1</td><td>Isi 2</td></tr><tr><td>Isi  
3</td><td>Isi 4</td></tr></tbody><tfoot><tr><td>Total 1</td><td>Tota  
l 2</td></tr></tfoot></table>
```

maka kita tidak akan melihat `#text` ketika memanggil `childNodes`.

```
// mengembalikan [<thead>...</thead>, <tbody>...</tbody>, <tfoot>...</tfoot>]  
tabel.childNodes;
```

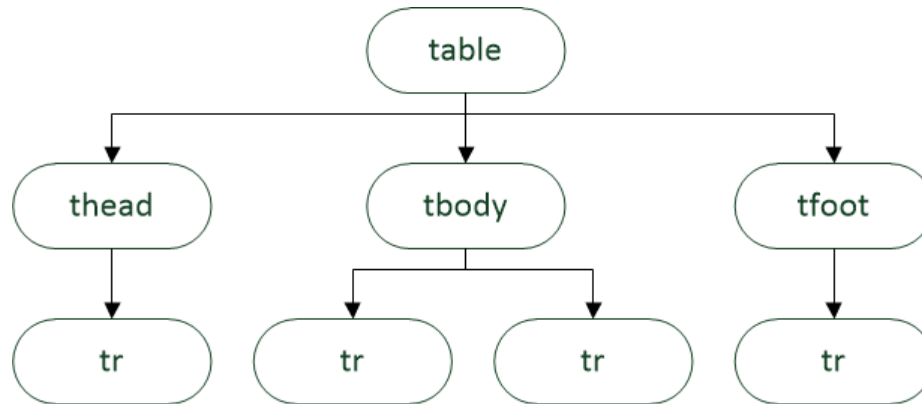
Karena `childNodes` mengembalikan sebuah *array*, maka kita dapat mengambil “anak” dari tabel yang kita inginkan seperti kita mengambil elemen *array*:

```
var anakTabel = table.childNodes;  
var badanTabel = anakTabel[1]; // badanTabel berisi <tbody>
```

Kita juga dapat mengakses `thead` dan `tfoot` melalui `tbody` dengan memanggil properti `nextSibling` dan `previousSibling`:

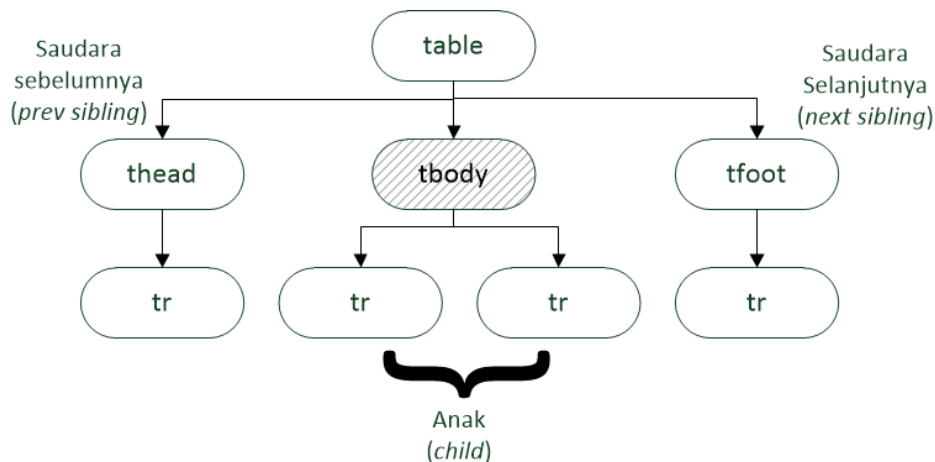
```
var kepalaTabel = badanTabel.previousSibling;  
var kakiTabel = badanTabel.nextSibling;
```

Penggunaan istilah “anak” (*child*) dan “saudara” (*sibling*) ini berasal dari penampilan tabel yang mirip dengan pohon silsilah keluarga:



Penampilan Pohon

Penampilan pohon yang mirip seperti silsilah ini memungkinkan kita melihat cabang dari pohon sebagai individu di dalam keluarga. Misalnya jika kita melihat silsilah dari **tbody** pada gambar sebelumnya, maka kita dapat melihat kenapa **kepalaTabel** adalah saudara dari **badanTabel**:



Pohon Tabel dengan Keterangan (Pusat: tbody)

Dari gambar di atas, dapat dilihat juga bahwa secara otomatis, keseluruhan tabel menjadi “orang tua” (*parent*) dari **badanTabel**. Berdasarkan nama properti sebelumnya, kita dapat menebak bahwa untuk mengakses seluruh tabel dari **badanTabel** kita dapat memanggil properti **parentNode**:

```
// mengembalikan <table id="contoh-tabel">..</table>
badanTabel.parentNode;
```

Dengan pengetahuan yang kita miliki sekarang, kita dapat mengimplementasikan sebuah fungsi yang menelusuri seluruh elemen DOM secara rekursif seperti berikut:

```
var TelusuriDOM = function telusuri(node) {
  node = node.firstChild;

  while (node) {
    telusuri(node);
    node = node.nextSibling;
  }
}
```

```
    }  
};
```

Fungsi di atas akan melakukan penelusuran ke anak dari *node* yang aktif sekarang, dan kemudian ke seluruh anak dari anak tersebut, serta anak dari saudara-saudaranya. Tetapi penelusuran saja tentunya tidak terlalu berguna. Algoritma yang digunakan pada fungsi ini adalah algoritma yang paling sederhana untuk penelusuran pohon secara rekursif.

Mari kita tambahkan fitur untuk memanggil sebuah fungsi (yang diberikan pengguna) untuk dijalankan pada setiap *node*:

```
var TelusuriDOM = function telusuri(node, fungsi_aksi) {  
    fungsi_aksi(node);  
  
    node = node.firstChild;  
  
    while (node) {  
        telusuri(node);  
        node = node.nextSibling;  
    }  
};
```

Atau mungkin lebih lagi, yaitu jika kita hanya ingin menjalankan **fungsi_aksi** pada saat tertentu saja, di mana syarat eksekusinya ditentukan oleh pengguna (melalui fungsi lagi tentunya):

```
var TelusuriDOM = function telusuri(node, fungsi_cek, fungsi_aksi) {  
    if (fungsi_cek(node)) {  
        fungsi_aksi(node);  
    }  
  
    node = node.firstChild;  
  
    while (node) {  
        telusuri(node, fungsi_cek, fungsi_aksi);  
        node = node.nextSibling;  
    }  
};
```

Kita kemudian dapat memberikan **fungsi_cek** dan **fungsi_aksi** kepada **TelusuriDOM**. Contohnya, jika kita ingin menghapus baris-baris dalam tabel yang memiliki kolom berisi "Isi 4". **fungsi_cek** sudah pasti harus diisikan dengan pengecekan nilai "Isi 4" dalam setiap *node*, sementara **fungsi_aksi** akan melakukan penghapusan baris. Berikut detail implementasinya:

```
// Merupakan fungsi_cek  
// Memastikan bahwa node merupakan sebuah <td> (melalui nodeName)  
// dan isi dari node adalah "Isi 4"  
var CekIsiKolom = function (node) {  
    return node.innerHTML === "Isi 4" && node.nodeName === "TD";  
};
```

```
// Merupakan fungsi_aksi
// Karena fungsi ini hanya dijalankan ketika CekIsiKolom benar,
// maka kita dapat mengasumsikan node selalu adalah <td>.
var HapusBaris = function (node) {
    var baris = node.parentNode; // orang tua <td> selalu adalah <tr>
    var badan = baris.parentNode; // (logikanya) idem

    badan.removeChild(baris);
};

TelusuriDOM(tabel, CekIsiKolom, HapusBaris);
```

PRAKTIKUM 1 :

```
<html>
<head>
<title>Untitled Document</title>
<script language="javascript">
<!--
function kode_pesanan_change()
{
    var ambil = document.form1.xkode_pesanan.value;
    if (ambil.substring(0,2)=="SP" )
        {
            document.form1.xjenis_kopi.value="SUPER";
            document.form1.xharga_perkilo.value =65000;
        }
    else if (ambil.substring(0,2)=="ST" )
        {
            document.form1.xjenis_kopi.value="STANDAR";
            document.form1.xharga_perkilo.value =42500;
        }

        // PERINTAH MENGAMBIL 3 DIGIT DITENGAH
    if (ambil.slice(3,-4)=="MXC")
        {
            document.form1.xnegara_tujuan.value="MEXICO";
            document.form1.xbea_cukai.value=400;
            document.form1.xbonus.value="MAKAN MIESO TOWER PAYAKUMBUH";
        }
    else if (ambil.slice(3,-4)=="KWT")
        {
            document.form1.xnegara_tujuan.value="KUWAIT";
            document.form1.xbea_cukai.value=475;
            document.form1.xbonus.value="MAKAN MIESO TOWER PAYAKUMBUH";
        }
    else if (ambil.slice(3,-4)=="QTR")
        {
            document.form1.xnegara_tujuan.value="QATAR";
            document.form1.xbea_cukai.value=625;
            document.form1.xbonus.value="VACATION TO BALI";
        }
    else if (ambil.slice(3,-4) == "KLM")
        {
            document.form1.xnegara_tujuan.value="KALIMANTAN";
```

```
        document.form1.xbea_cukai.value=0;
        document.form1.xbonus.value="MAKAN MIESO TOWER PAYAKUMBUH";
    }
    else if (ambil.slice(3,-4) == "IRJ")
    {
        document.form1.xnegara_tujuan.value="IRIAN JAYA";
        document.form1.xbea_cukai.value=0;
        document.form1.xbonus.value="MAKAN MIESO TOWER PAYAKUMBUH";
    }
}
// FUNGSI KE - 2
function banyak_pesanan_change()
{
    document.form1.xjumlah_harga.value =
parseFloat(document.form1.xharga_perkilo.value) *
parseFloat(document.form1.xbanyak_pesanan.value);

    if (document.form1.xjumlah_harga.value <= 25000000)
    {
        document.form1.xpotongan.value = "15";
    }
    else if (document.form1.xjumlah_harga.value >= 47000000)
    {
        document.form1.xpotongan.value = "125";
    }
    document.form1.xppn.value = 0.1 * document.form1.xjumlah_harga.value;
    document.form1.xtotal_biaya.value =
parseFloat(document.form1.xjumlah_harga.value)+
parseFloat(document.form1.xppn.value);
}
//-->
</script>
</head>

<body>
<form name="form1" method="post" action="">
    <table width="450" border="0" align="center" cellpadding="0"
cellspacing="0">
        <tr>
            <td colspan="3"><div align="center">ENTRY EXPORT IMPORT &quot;KOPI
LUWAK&quot; </div></td>
        </tr>
        <tr>
            <td colspan="3"><hr></td>
        </tr>
        <tr>
            <td width="224">KODE PEMESANAN </td>
            <td width="16">:</td>
            <td width="202"><input name="xkode_pesanan" type="text" id="xkode_pesanan"
onChange="kode_pesanan_change()"></td>
        </tr>
        <tr>
            <td>JENIS KOPI</td>
            <td>:</td>
            <td><input name="xjenis_kopi" type="text" id="xjenis_kopi"></td>
        </tr>
    </table>
</body>
```

```
<tr>
  <td>HARGA PERKILO </td>
  <td>:</td>
  <td><input name="xharga_perkilo" type="text" id="xharga_perkilo"></td>
</tr>
<tr>
  <td>BANYAK PESAN </td>
  <td>:</td>
  <td><input name="xbanyak_pesan" type="text" id="xbanyak_pesan"
onChange="banyak_pesan_change()"></td>
</tr>
<tr><td>NEGARA TUJUAN </td>    <td>:</td>
  <td><input name="xnegara_tujuan" type="text" id="xnegara_tujuan"></td>
</tr>
<tr>
  <td>BEA CUKAI ($ US) </td>
  <td>:</td>
  <td><input name="xbea_cukai" type="text" id="xbea_cukai"></td>
</tr>
<tr>
  <td>JUMLAH HARGA </td>
  <td>:</td>
  <td><input name="xjumlah_harga" type="text" id="xjumlah_harga"></td>
</tr>
<tr>
  <td>POTONGAN BEA CUKAI </td>
  <td>:</td>
  <td><input name="xpotongan" type="text" id="xpotongan"></td>
</tr>
<tr>
  <td>PPn (10%) </td>
  <td>:</td>
  <td><input name="xppn" type="text" id="xppn"></td>
</tr>
<tr>
  <td>BONUS</td>
  <td>:</td>
  <td><input name="xbonus" type="text" id="xbonus"></td>
</tr>
<tr>
  <td>TOTAL BIAYA </td>
  <td>:</td>
  <td><input name="xtotal_biaya" type="text" id="xtotal_biaya"></td>
</tr>
<tr>
  <td colspan="3"><hr></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td><input type="submit" name="Submit" value="TAMPILKAN"></td>
</tr>
</table>
</form>
</body>
</html>
```

PENERAPAN DOM PADA ELEMENT FORM HTML

a. <Form>

Tag <Form> mendefinisikan sebuah elemen form HTML

Contoh :

```
<form method="POST/GET" Action ="Link_File">  
    Elemen form  
</form>
```

Catatan :

GET :

data akan tampil dialamat web browser, jika dikirim ke server
(informasi yang dikirim dapat dilihat oleh user lain)

POST : (untuk informasi yang sensitif/rahasia)

data tidak tampil dialamat web browser, jika dikirim ke server
(informasi yang dikirim tidak dapat dilihat oleh user lain)

b. Text/TextField

Syntax

```
<input type="Text" name="nama_objek" id="id_objek">
```

Contoh :

```
<form>
```

Nama Lengkap :

```
<input type="Text" name="nm_lkp" id="nm_lkp">  
</form>
```

c. Radio

Syntax :

```
<form>
```

```
<input type="radio" name="jk" value="Laki-Laki" Checked>Laki-laki  
<br>
```

```
<input type="radio" name="jk " value=" Perempuan">Perempuan
```

```
</form>
```

d. Checkbox

Syntax

```
<form>
```

```
<input type="checkbox" name="xhobi" id="xhobi ">Renang
```

```
<input type="checkbox" name="xhobi2" id="xhobi2 ">Volley
```

```
<input type="checkbox" name="xhobi3" id="xhobi3 ">Futsal
```

```
</form>
```

e. Select (List/Menu)

Syntax :

```
<select name="cbkode" id="cbkode">
```

```
    <option value="1-1/A">1-1/A</option>
```

```
    <option value="2-1/B">2-1/B</option>
```

```
    <option value="3-2/A">3-2/A</option>
```

```
</select>
```

f. Button

Syntax :

```
<form>
```

```
<input type="Submit" value="Submit">
</form>

g. TextArea
<form>
<textarea name="keterangan" cols="60" rows="7" id="keterangan"></textarea>
</form>

h. Fieldset
<form action="nama_link.php">
<fieldset>
<legend>Informasi:</legend>
Nama :<br>
<input type="text" name="xnama " value="Nama Anda ">
<br>
<br>
<input type="submit" value="Kirim"></fieldset>
</form>
```

1. Event / Methode dalam Javascript

- **onClick** = event ini dijalankan ketika diklik sebuah element html (textfield, textArea, dll)
- **onChange** = event ini dijalankan ketika terjadi perubahan sebuah nilai pada elemen HTML
- **onError** = event ini di jalankan ketika terjadi sebuah error atau kesalahan.
- **onLoad**= Event ini di gunakan untuk menjalankan perintah ketika suatu objek selesai ditampilkan pada halaman web.
- **onSubmit** = event ini di jalanka ketika setelah menekan sebuah tombol pada form.
- **onSelect** = event ini di jalankan ketika melakukan pemilihan pada text atau text di block
- **onBlur** = Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element window, frame, select, text, dan textarea) kehilangan focusnya
- **onClick**=event ini terjadi jika terjadi klik pada tombol,hyperlink,gambar,dll.
- **onContextMenu** = event ini terjadi jika terjadi klik kanan pada sebuah element yang di beri event ini.
- **onDbClick**=event ini terjadi jika terjadi dauble klik.
- **onKeyPress**= event ini terjadi jika terjadi pengetikan keyboard.
- **onKeyDown** = event ini terjadi jika terjadi penekanan pada tombol bawah pada keyboard.
- **onMouseMove** = event in terjadi jika pointer mouse bergerak.
- **onMouseout** = event ini terjadi jika pointer keluar dari element yang di beri event ini.
- **onMouseover** = event yang terjadi jika pointer berada di atas element yang di beri event ini (kebalikan dari event onMouseout).

2. Contoh Sederhana Penerapan Event ada Elemen Form dengan Javascript

a. Pada textfield

Contoh :

Banyak	:	<input type="text" value="xbanyak"/>
Harga	:	<input type="text" value="xharga"/>
Jumlah	:	<input type="text" value="xjumlah"/>

Misalkan nama form adalah FORM1

Logika :

Jumlah = Banyak * harga

Atau

Xjumlah= xbanyak * xharga

Pembahasan :

Untuk element Textfield (event yang sering digunakan adalah onChange, onBlur, onKeyUp, dll)

Pada kasus ini kita menggunakan event onChange, dimana logikanya adalah pada saat kita mengisi nilai pada Textbox harga, maka otomatis Textbox Jumlah akan berisi nilai.

Untuk itu diperlukan sebuah fungsi yang akan dipanggil pada saat nilai pada Textbox harga berisi nilai. (kita misalkan nama fungsinya adalah nilai_jumlah).

Jawab :

```
<html><head><title>event pada Textfield</title>
<script language = "javascript">
//-- >
function nilai_jumlah() // nama fungsi
{
// tampung nama form
Var nmform = document.form1;
//proses nilai jumlah = banyak * harga
nmform.xjumlah.value = nmform.xbanyak.value * nmform.xharga.value;
}
<!--
</script></head>
<body>
<form action="">
<input type="TEXT" name="xbanyak" id="xbanyak">
<input type="TEXT" name="xharga" id=" xharga" onChange="nilai_jumlah()">
<input type="TEXT" name="xjumlah" id="xjumlah">
</form></body>
</html>
```

Keterangan :

onChange="nilai_jumlah()"

merupakan penerapan event onChange pada element HTML TextField (xharga)

b. Pada Select (list/menu)

Untuk penerapan event onClick pada element Select (list/Menu) dapat anda pelajari pada contoh kasus (bayangan soal UAS) dibahas terakhir modul ini.

PEMOTONGAN STRING DALAM JAVASCRIPT PADA DOM

Pada dasarnya pemotongan string semua bahasa pemrograman sama yaitu : pemotongan dari kiri (LEFT), kanan (RIGHT) atau tengah (MID), tetapi dalam bahasa Javascript pemotongan dari kiri, kanan dan tengah tersebut dapat diwakilkan dengan perintah *substr*.

Contoh :

- a. Pemotongan terhadap sebuah variabel yang berisi data
Misalkan sebuah variabel kita beri nama dengan variabel NAMA,
dimana data variabel NAMA tersebut = "MUHAMAD THAMRIN"

Pembahasan :

```
// 1 2 345 6789. 1 2 345
var NAMA="MUHAMAD THAMRIN"; // jumlah karakter pada
                               // variabel NAMA termasuk spasi = 15 huruf

// kita ingin menampilkan huruf MUH (3 digit dari KIRI)
// caranya :
var ambil_kiri = nama.substr(0,3); //→ hasilnya = MUH
// kita ingin menampilkan huruf AMAD (4 digit ditengah)
// caranya :
var ambil_tengah = nama.substr(3,4); //→ hasilnya = AMAD
// kita ingin menampilkan huruf RIN (3 digit dikanan)
// caranya :
var ambil_kanan = nama.substr(12,3); //→ hasilnya = RIN
```

- b. Pemotongan terhadap objek form (document object method)

Contoh :

Untuk kasus ini kita coba dengan menggunakan element Select (List/Menu)

Kode Wilayah :

KODE WILAYAH :
NAMA BENUA :
NAMA NEGARA :

Misalkan dalam sebuah Select (List/Menu) terdapat 3 data sebagai berikut :

A/01

B/02

B/03

Dimana

A,B → logika untuk mencari Nama Benua

Contoh :

Jika A, maka benua Amerika, jika B, maka Benua EROPA

01,02,03 → logika untuk mencari nama Negara

Contoh :

Jika 01, maka Negara = "Los Angeles"

Jika 02, maka Negara = "Netherland"

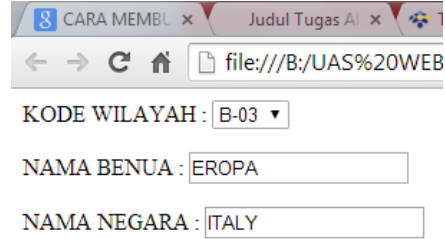
Jika 03, maka Negara= "ITALY"

Pembahasan dengan Javascript :

```
<html>
<head>
<title>Untitled Document</title>
<script language="javascript">
//-->
function logika_wilayah()
{
var nmform = document.form1;
  // logika untuk menentukan nama benua
  if(nmform.cbkode.value.substr(0,1)=="A")
  {
    nmform.xbenua.value="AMERIKA";
  }
  else
  {
    nmform.xbenua.value="EROPA";
  }
  // logika untuk menentukan nama negara
  if(nmform.cbkode.value.substr(2,2)=="01")
  {
    nmform.xnegara.value="Los Angeles";
  }
  else if(nmform.cbkode.value.substr(2,2)=="02")
  {
    nmform.xnegara.value="Netherland";
  }
  else
  {
    nmform.xnegara.value="ITALY";
  }
}
<!--
</script>
</head>
<body>
<form id="form1" name="form1" method="post" action="">
  <p>KODE WILAYAH :
    <label for="cbkode"></label>
    <select name="cbkode" id="cbkode" onChange="logika_wilayah()">
      <option value="A-01">A-01</option>
      <option value="B-02">B-02</option>
      <option value="B-03">B-03</option>
    </select>
  </p>
```

```
<p>NAMA BENUA :  
<label for="xbenua"></label>  
<input type="text" name="xbenua" id="xbenua" />  
</p>  
<p>NAMA NEGARA :  
<input type="text" name="xnegara" id="xnegara" />  
</p>  
</form>  
</body>  
</html>
```

Jika dijalankan dengan browser (tekan F12), maka hasilnya



REVIEW UAS

BAYANGAN UJIAN AKHIR SEMESTER
WEB PROGRAMMING KELAS 5A

Kode Robot	:	2-1/B
Jenis Robot	:	Robot Manipulator (Tangan)
Bahan yang dibutuhkan	:	2 bh Triplek/acrylic ukuran 20x20 cm, 2 unit Mobil bekas (dipotong as-nya), 2 unit Relay 5 volt 8 pin, Limit switch 2 unit, IC Regulator 7805 1 unit, Transistor TIP31 / ttransistor lain 2 unit, Variabel resistor 2 unit, 1 buah PCB Polos ukuran 8x5 cm, Speser/baut 4 unit, 2 buah Header pin hitam, 2 Resistor dan kapasitor, 3 rol Kabel, timah, batre / Lippo 7,4 V
Alat yang dibutuhkan	:	Solder listrik, Tang Potong, Tang Lancip, Gunting Kuku, Gerinda Kecil, Gunting, Cutter, Obeng
Software Yang Digunakan	:	KI ROBOT

Tampilan Form di atas, jika dipilih Kode Robot = 2-1/B

Penjelasan Form :

Data Input adalah Data yang ditampung dalam Select (list/Menu) pada kode robot.

Isi data Kode Robot =

a b c

1-1/A

2-1/B

3-2/A

Dimana

- Kolom *a* : untuk mendapatkan jenis robot & bahan yang digunakan
- Kolom *b* : untuk mendapatkan alat yang dibutuhkan
- Kolom *c* : untuk mendapatkan software yang digunakan

Logika untuk masing-masing kolom dapat dilihat pada tabel berikut :

Jika Kolom a →	1	2	3
Maka Jenis Robot →	Robot Mobile (Bergerak)	Robot Manipulator (Tangan)	Robot Humanoid
Maka Bahan yang digunakan →	2 buah dinamo motor 1,5 Volt, buah klip penjepit kertas kecil, buah klip penjepit kertas besar, buah baterai ukuran AA, buah dudukan baterai AA, batang lem bakar, 1 buah manik-manik untuk roda, 1 m kabel tunggal kecil/kabel jumper, 2 bh microswitch saklar power on/off	2 bh Triplek/acrylic ukuran 20x20 cm, 2 unit Mobil bekas (dipotong as-nya), 2 unit Relay 5 volt 8 pin, Limit switch 2 unit, IC Regulator 7805 1 unit, Transistor TIP31 / ttransistor lain 2 unit, Variabel resistor 2 unit, 1 buah PCB Polos ukuran 8x5 cm, Speser/baut 4 unit, 2 buah Header pin hitam, 2 Resistor dan kapasitor, 3 rol Kabel, timah, batre / Lippo 7,4 V	Semua bahan yang digunakan pada Robot Mobile dan Robot Manipulator

Jika Kolom b →	1	2
Maka alat yang digunakan →	Solder listrik, Tang Potong, Tang Lancip, Gunting Kuku, Gerinda Kecil, Gunting, Cutter, Obeng	Solder listrik, Tang Potong, Tang Lancip, Gunting Kuku, Gerinda Kecil, Gunting, Cutter, Obeng, Lem, Glue Gun dan Isinya, Electric Drill, Hand Drill, Line's Men Knife, Vise, Offset Wrech, Wire Striper

Jika Kolom c →	A	B
Maka software yang digunakan →	ROBO MIND	KI ROBOT

Berdasarkan Data Input/masukan dan Proses logika di atas, maka dapat kita buat program lengkapnya seperti script dibawah ini :

```
<html>
<head>
<title>Bayangan Soal UAS Kela 5A</title>
<script language="javascript">
//-->
function proses ()
{
    var nmform = document.form1;
    if (nmform.cbkode.value.substr(0,1) == "1")
    {
        nmform.xjenis.value="Robot Mobile (Bergerak)";
    }
}
```

```
nmform.xbahan.value="2 buah dinamo motor 1,5 Volt,buah klip penjepit kertas
kecil,buah klip penjepit kertas besar,buah baterai ukuran AA,buah dudukan baterai
AA,batang lem bakar,1 buah manik-manik untuk roda,1 m kabel tunggal kecil/kabel
jumper,2 bh microswitch saklar power on/off";

}
else if (nmform.cbkode.value.substr(0,1)=="2")
{

nmform.xjenis.value="Robot Manipulator (Tangan)";
nmform.xbahan.value="2 bh Triplek/acrylic ukuran 20x20 cm,2 unit Mobil bekas
(dipotong as-nya),2 unit Relay 5 volt 8 pin,Limit switch 2 unit,IC Regulator 7805 1
unit,Transistor TIP31 / ttransistor lain 2 unit,Variabel resistor 2 unit,1 buah PCB
Polos ukuran 8x5 cm,Speser/baut 4 unit,2 buah Header pin hitam,2 Resistor dan
kapasitor,3 rol Kabel, timah, batre / Lippo 7,4 V ";
}
else
{

nmform.xjenis.value="Robot Humanoid";
nmform.xbahan.value="2 buah dinamo motor 1,5 Volt,buah klip penjepit
kertas kecil,buah klip penjepit kertas besar,buah baterai ukuran AA,buah dudukan
baterai AA,batang lem bakar,1 buah manik-manik untuk roda,1 m kabel tunggal
kecil/kabel jumper,2 bh microswitch saklar power on/off,2 bh Triplek/acrylic ukuran
20x20 cm,2 unit Mobil bekas (dipotong as-nya),2 unit Relay 5 volt 8 pin,Limit switch
2 unit,IC Regulator 7805 1 unit,Transistor TIP31 / ttransistor lain 2 unit,Variabel
resistor 2 unit,1 buah PCB Polos ukuran 8x5 cm,Speser/baut 4 unit,2 buah Header pin
hitam,2 Resistor dan kapasitor,3 rol Kabel, timah, batre / Lippo 7,4 V ";
}
//alat yang digunakan
if (nmform.cbkode.value.substr(2,1) == "1")
{
nmform.xalat.value="Solder listrik,Tang Potong,Tang Lancip,Gunting Kuku,Gerinda
Kecil ,Gunting,Cutter,Obeng";
}
else
{
//alat yang digunakan
nmform.xalat.value="Solder listrik,Tang Potong,Tang Lancip,Gunting Kuku,Gerinda
Kecil ,Gunting,Cutter,Obeng,Lem,Glue Gun dan Isinya,Electric Drill,Hand Drill,Line's
Men Knife,Vise,Offset Wrech,Wire Striper";
}
//software yang digunakan
if (nmform.cbkode.value.substr(4,1)=="A")
{
nmform.xsoftware.value="ROBO MIND";
}
else
{
nmform.xsoftware.value="KI ROBOT";
}
}
}

<!--
</script>
</head>

<body>
<form name="form1" method="post" action="">
<table width="600" border="0" align="center" cellpadding="1" cellspacing="1">
<tr>
<td colspan="3">BAYANGAN UJIAN AKHIR SEMESTER <br>
```

```
        WEB PROGRAMMING KELAS 5A</td>
</tr>
<tr>
  <td colspan="3"><hr></td>
</tr>
<tr>
  <td width="149">Kode Robot</td>
  <td width="3">:</td>
  <td width="438"><label for="cbkode"></label>
    <select name="cbkode" id="cbkode" onChange="proses()">
      <option value="1-1/A">1-1/A</option>
      <option value="2-1/B">2-1/B</option>
      <option value="3-2/A">3-2/A</option>
    </select></td>
</tr>
<tr>
  <td>Jenis Robot</td>
  <td>:</td>
  <td><label for="xjenis"></label>
    <input name="xjenis" type="text" id="xjenis" size="50"></td>
</tr>
<tr>
  <td>Bahan yang dibutuhkan</td>
  <td>:</td>
  <td><label for="xalat"></label>
    <textarea name="xbahan" id="xbahan" cols="60" rows="10"></textarea></td>
</tr>
<tr>
  <td>Alat yang dibutuhkan</td>
  <td>:</td>
  <td><textarea name="xalat" id="xalat" cols="60" rows="10"></textarea></td>
</tr>
<tr>
  <td>Software Yang Digunakan</td>
  <td>:</td>
  <td><label for="xsoftware"></label>
    <input name="xsoftware" type="text" id="xsoftware" size="50"></td>
</tr>
<tr>
  <td colspan="3"><hr></td>
</tr>
<tr>
  <td>NPM</td>
  <td>:</td>
  <td>diisi dengan NPM</td>
</tr>
<tr>
  <td>NAMA</td>
  <td>:</td>
  <td>diisi dengan Nama Lengkap</td>
</tr>
</table>
</form>
</body>
</html>
```

** End of file **