

MODUL I DASAR-DASAR HTML

1. TUJUAN

- Mahasiswa dapat mengenal tag dasar HTML dan memahami kegunaannya.
- Mahasiswa dapat membuat halaman web mengimplementasikan tag dasar HTML.

2. TEORI DASAR

Dalam praktikum kali ini, Anda akan mempelajari penggunaan dari beberapa tag dasar yang hampir akan selalu digunakan dalam setiap praktikum Anda nantinya. Perhatikan daftar tag di bawah ini, serta kegunaan utamanya dalam sebuah halaman HTML.

Tag HTML dasar

Start Tag	Kegunaan
<html>	Mendefinisikan sebuah dokumen html
<body>	Mendefinisikan isi/badan suatu dokumen
<h1>-<h6>	Mendefinisikan heading ke 1 s/d heading ke 6
<p>	Mendefinisikan sebuah paragraf
 	Menyisipkan sebuah line break
<hr>	Mendefinisikan sebuah garis horisontal
<!-->	Mendefinisikan komentar dalam kode sumber HTML

Setelah itu, Anda akan mempelajari penggunaan dari beberapa tag yang digunakan dalam melakukan pemformatan tampilan halaman web supaya lebih sesuai dengan keinginan atau agar lebih rapi dan terstruktur. Perhatikan beberapa daftar tag di bawah ini, serta kegunaan utamanya dalam sebuah halaman HTML. Beberapa diantaranya telah dihilangkan dari daftar tag HTML standar, namun masih memungkinkan untuk ditampilkan pada beberapa browser tertentu atau browser lama.

Tag untuk pemformatan teks dasar

Start Tag	Kegunaan
	Mendefinisikan teks tebal
<big>	Mendefinisikan teks yang lebih besar
	Mendefinisikan teks yang dimiringkan
<i>	Mendefinisikan teks yang dimiringkan
<small>	Mendefinisikan teks yang lebih kecil
	Mendefinisikan teks tebal
<sub>	Defines subscripted text
<sup>	Mendefinisikan teks yang posisinya lebih ke atas
<ins>	Mendefinisikan teks yang bergaris bawah
	Mendefinisikan teks yang hendak dihapus

Tag untuk tampilan 'hasil keluaran komputer'

Start Tag	Kegunaan
<code>	Mendefinisikan teks kode komputer

<kbd>	Mendefinisikan teks dari keyboard
<samp>	Mendefinisikan teks contoh kode komputer
<tt>	Mendefinisikan teks model teletype
<var>	Mendefinisikan sebuah variabel
<pre>	Mendefinisikan teks preformatted

Tag untuk Citation, Quotation, dan Definisi

Start Tag	Kegunaan
<abbr>	Mendefinisikan suatu kependekan
<acronym>	Mendefinisikan suatu singkatan
<address>	Mendefinisikan suatu elemen alamat
<bdo>	Mendefinisikan arah teks
<blockquote>	Mendefinisikan quotation panjang
<q>	Mendefinisikan quotation pendek
<cite>	Mendefinisikan suatu kutipan
<dfn>	Mendefinisikan suatu istilah

Contoh tag untuk Entitas karakter

Tampilan	Deskripsi	Nama Entitas	Nomor Entitas
	spasi kosong	 	
<	kurang dari	<	<
>	lebih dari	>	>
&	Dan (ampersand)	&	&
"	tanda petik/kutip (quotation mark)	"	"
'	koma atas (apostrophe)		'
¢	Cent	¢	¢
£	Pound	£	£
¥	Yen	¥	¥
§	Section	§	§
©	Copyright	©	©
®	registered trademark	®	®
×	Multiplication	×	×
÷	Division	÷	÷

Ketika Anda menulis suatu dokumen, mungkin terkadang Anda menuliskan kalimat dalam bentuk poin-poin yang menerangkan tentang suatu hal. Dalam suatu halaman web, terkadang Anda juga ingin melakukan hal yang sama. Terdapat tag khusus untuk penulisan dalam format poin-poin, perhatikan daftar di bawah ini.

Tag untuk list

Start Tag	Kegunaan
	Mendefinisikan sebuah list ordered
	Mendefinisikan sebuah list unordered
	Mendefinisikan sebuah item dalam list
<dl>	Mendefinisikan sebuah list definisi
<dt>	Mendefinisikan sebuah istilah list definisi

<dd>	Mendefinisikan sebuah keterangan list definisi
------	--

3. LATIHAN

Dalam modul ini Anda akan berlatih untuk menggunakan berbagai tag dasar untuk pemrograman web menggunakan HTML. Ketikkan setiap kelompok tag dalam sebuah file, simpan dalam folder Anda sendiri yang telah ditentukan, kemudian jalankan di web browser yang telah terinstall. Ketikkan setiap kode sumber apa adanya terlebih dahulu (termasuk spasi kosong). Perhatikan hasil tampilannya, kemudian belajar untuk memodifikasi setiap kelompok tag untuk mengetahui maksud dari setiap baris tag HTML yang Anda ketikkan. Jangan lupa untuk menyimpan setiap modifikasi Anda sebelum menjalankan ulang di dalam browser.

Silakan mencoba juga tag yang masuk dalam kategori “dihilangkan”, untuk melihat seperti apa tampilan yang dapat dimunculkan. Untuk entitas karakter, silakan coba nomor entitas selain yang telah ada pada daftar di atas, untuk melihat entitas karakter apa saja yang dapat digunakan dalam sebuah halaman web.

1. Background color dan heading

```
<html>
<body>
<body bgcolor="yellow">
<p>
Perhatikan bahwa halaman ini seharusnya berwarna kuning.
</p>
<h1>Ini adalah heading 1</h1>
<h2>Ini adalah heading 2</h2>
<h3>Ini adalah heading 3</h3>
<h4>Ini adalah heading 4</h4>
<h5>Ini adalah heading 5</h5>
<h6>Ini adalah heading 6</h6>
<p>Gunakan tag heading hanya untuk membuat heading saja. Jangan
menggunakan tag tersebut hanya untuk membuat tampilan huruf tebal.
Gunakan tag lain untuk keperluan itu. </p>
<h1 align="center">Ini adalah heading 1</h1>
<p>Heading di atas telah diposisikan untuk berada di tengah halaman
ini. </p>
</body>
</html>
```

2. Garis horisontal dan komentar tersembunyi

```
<html>
<body>
<p>Tag hr mendefinisikan sebuah garis horisontal, default-nya adalah
rata tengah:</p>
<hr color="green" >
<p align="left">Terdapat paragraf disini</p>
<hr width="80%" size="10" align="left">
<p align="right">Terdapat paragraf disini</p>
<hr width="400" size="6" align="right" color="red">
<center>
<p>Terdapat paragraf disini</p>
</center>
<!-- Ini adalah komentar yang tidak akan ditampilkan di layar browser
-->
</body>
</html>
```

3. List unordered dan ordered

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

```

<h4>Sebentuk list unordered:</h4>
<ul>
<li>Kopi</li>
<li>Teh</li>
<li>Susu</li>
</ul>
<br>
<h4>Sebentuk list ordered:</h4>
<ol>
<li>Kopi</li>
<li>Teh</li>
<li>Susu</li>
</ol>
</body>
</html>

```

4. Jenis-jenis list ordered

```

<html>
<body>
<h4>List bernomor:</h4>
<ol>
<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ol>
<h4>List dengan huruf:</h4>
<ol type="A">
<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ol>
<h4>List dengan huruf kecil:</h4>
<ol type="a">
<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ol>
<h4>List dengan angka romawi:</h4>
<ol type="I">
<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ol>
<h4>List dengan angka romawi kecil:</h4>
<ol type="i">
<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ol>
</body>
</html>

```

5. Jenis-jenis list unoredered

```

<html>
<body>
<h4>List dengan bullet berbentuk disc:</h4>
<ul type="disc">

```

```

<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ul>
<h4>List dengan bullet berbentuk lingkaran:</h4>
<ul type="circle">
<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ul>
<h4>List dengan bullet berbentuk kotak:</h4>
<ul type="square">
<li>Apel</li>
<li>Pisang</li>
<li>Lemon</li>
<li>Jeruk</li>
</ul>
</body>
</html>

```

6. List bersarang dan list definisi

```

<html>
<body>
<h4>Sebuah list bersarang:</h4>
<ul>
<li>Kopi</li>
<li>Teh
<ul>
<li>Teh hitam</li>
<li>Teh Hijau</li>
</ul>
</li>
<li>Susu</li>
</ul>
<br>
<h4>Contoh list definisi:</h4>
<dl>
<dt>Kopi</dt>
<dd>Minuman panas hitam</dd>
<dt>Susu</dt>
<dd>Minuman dingin putih</dd>
</dl>
</body>
</html>

```

MODUL II HTML LEBIH LANJUT(LINK, FRAME, DAN TABEL)

1. TUJUAN

- Mahasiswa dapat mengenal tag HTML untuk membuat link, frame dan tabel.
- Mahasiswa dapat membuat desain halaman web mengimplementasikan tag link,frame dan tabel.

2. TEORI DASAR

Situs web biasanya tidak hanya terdiri atas 1 halaman saja (*homepage*, halaman awal/ pembuka). Bila suatu situs terdiri atas lebih dari 1 halaman, situs tersebut pasti menggunakan hyperlink. Perhatikan bahwa tag untuk link memang hanya 1, tetapi memiliki beberapa atribut tambahan agar link tersebut memberikan fungsi yang berbeda. Untuk keperluan modifikasi tampilan agar link tidak hanya terlihat sebagai link standar, nantinya akan disampaikan dalam materi khusus CSS. Tampilan link juga dapat diubah sedikit lewat modifikasi tag `<body>` dalam suatu file HTML.

Start Tag	Kegunaan
<code><a></code>	Mendefinisikan sebuah anchor

Atribut Target	Kegunaan
<code>target="_blank"</code>	Memuat dokumen baru ke dalam window baru yang kosong
<code>target="_self"</code>	Memuat dokumen baru ke dalam window yang sama dengan anchor (default)
<code>target="_parent"</code>	Memuat dokumen baru ke dalam parent frame (bila menggunakan frame)
<code>target="_top"</code>	Memuat dokumen baru ke dalam keseluruhan window browser (cara yang umum digunakan untuk keluar dari frame)

Sebuah halaman web yang ditampilkan dalam suatu window browser dapat dipecah dalam beberapa tampilan yang berbeda. Masing-masing bagian tampilan dapat berisi sebuah tampilan dari file HTML, sehingga dalam satu window browser dapat dibuka beberapa file HTML sekaligus. Untuk dapat menampilkan beberapa file HTML dalam satu window browser, dibutuhkan tag untuk membuat window menjadi beberapa bingkai (frame). Di bawah ini adalah tag untuk membentuk frame pada window browser.

Start Tag	Kegunaan
<code><frameset></code>	Mendefinisikan sebuah himpunan frame
<code><frame></code>	Mendefinisikan sebuah sub window (sebuah frame)
<code><noframes></code>	Mendefinisikan sebuah bagian noframe untuk browser yang tidak dapat menangani frame
<code><iframe></code>	Mendefinisikan sebuah inline sub window (frame)

Dalam suatu halaman web, pemformatan saja tidaklah cukup untuk membentuk tampilan web yang rapi dan enak dipandang. Supaya tampilannya lebih terstruktur, Anda dapat menggunakan tabel. Perhatikan tag-tag di bawah ini.

Start Tag	Kegunaan
-----------	----------

<table>	Mendefinisikan sebuah tabel
<th>	Mendefinisikan sebuah header tabel
<tr>	Mendefinisikan suatu barisan dalam tabel
<td>	Mendefinisikan suatu sel dalam table
<caption>	Mendefinisikan sebuah caption untuk tabel
<colgroup>	Mendefinisikan sekelompok kolom dalam tabel
<col>	Mendefinisikan nilai atribut untuk satu atau lebih kolom dalam sebuah tabel
<thead>	Mendefinisikan suatu head tabel
<tbody>	Mendefinisikan suatu badan tabel
<tfoot>	Mendefinisikan suatu footer tabel

3. LATIHAN

Silakan mencoba kode-kode contoh pada modul ini, kemudian silakan mencoba variasi penggunaannya. Gunakan alamat tujuan yang berbeda dari contoh yang ada. Bila tidak terdapat koneksi ke Internet, gunakan alamat lokal saja terlebih dahulu. Alamat lokal juga mempercepat akses Anda, sehingga hasilnya lebih cepat tampil di layar monitor. Bila Anda ingin menggunakan halaman tujuan secara lokal, sebaiknya Anda membuat file HTML tambahan dengan nama file yang disesuaikan agar dapat dipanggil. Isi file HTML tambahan terserah Anda, yang penting bisa digunakan untuk menunjukkan bahwa tag Anda bekerja seperti seharusnya.

Untuk mencoba kode HTML tentang frame, silakan menyiapkan file-file yang diperlukan sesuai dengan nama-nama file yang telah disebutkan dalam contoh kode dalam modul ini. Simpan di tempat yang sesuai dengan penunjukan frame Anda, disarankan disimpan dalam folder yang sama tempat Anda menyimpan file definisi frame. Sebagai awalan, sebaiknya Anda menggunakan browser IE, sebelum mencobakannya ke Navigator atau Opera (bila ada).

Latihan menggunakan tabel mengharuskan Anda mencoba bentuk-bentuk yang ada dalam contoh, kemudian silakan mencoba-coba sendiri variasi contoh dengan menambahkan atau mengurangi sel dan isinya. Variasikan juga parameternya, agar Anda tahu setiap bagian penentu tampilan tabel.

1. Link biasa, link gambar, link ke window baru, dan link internal

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Berita Bola</title>
<style type="text/css">
<!--
.style1 {color: #FF0000}
-->
</style>
</head>

<body bgcolor="#CCCCCC">
<h1 align="justify" class="style1">Fulham Redam Hamburg Tanpa
Gol</h1>
<p align="justify">Hamburg gagal memetik kemenangan di semi-final
<em>leg</em> pertama Liga Europa. Tampil di hadapan pendukungnya

```

sendiri di HSH Nordbank Arena, Jumat (23/4) dinihari, Hamburg harus puas bermain imbang tanpa gol melawan Fulham.

Tim tuan rumah sebenarnya berusaha tampil agresif sejak awal pertandingan. Di dua menit awal, penyelesaian yang buruk dari Ruud van Nistelrooy membuat gawang Fulham yang dikawal Mark Schwarzer masih bisa selamat.

Beberapa menit berselang, giliran gelandang Piort Trochowski yang juga gagal memaksimalkan kesempatan. Bola hasil tendangan pemain tim nasional Jerman ini masih terlalu melebar dari gawang Fulham.

Sementara The Cottagers mendapatkan kesempatan pertamanya untuk mencetak gol setelah waktu berjalan hampir 15 menit. Ketika itu umpan panjang Bobby Zamora juga gagal diselesaikan dengan baik oleh Zoltan Gea.

Lalu di lima menit jelang turun minum gelandang Fulham Joanthan Pitroipa membuka harapan publik tuan rumah. Pergerakannya yang cepat ke lini pertahanan Fulham itu kemudian diselesaikan dengan sebuah tendangan. Tapi ancaman tersebut masih bisa diantisipasi oleh Swarzer dengan baik.

Di babak kedua, Hamburg kembali lagi berusaha untuk tampil lebih dominan dalam menyerang. Kerjasama Paolo Guerrero dan Nistelrooy membuka kesempatan pertama bagi Fulham. Tapi ruang gerak yang terbatas membuat tendangan Guerrero masih melambung di atas sasaran.

Sebaliknya Fulham juga tidak tinggal diam untuk mencuri kesempatan. Salah satu peluang yang tercipta di babak ini melalui pergerakan Gera yang bergerak dari sayap kiri.

Sementara kesempatan paling besar di babak ini terjadi di menit ke-70. Nistelrooy memberikan umpan ke arah Pitroipa. Gelandang asal Burkina Faso ini kemudian menyambutnya dengan tandukan kepala. Sayangnya, bola yang mengarah ke pojok atas gawang Fulham masih di tepis oleh Schwarzer.

Dengan hasil imbang ini membuat langkah Fulham untuk menembus ke partai puncak Liga Europa kian terbuka ketika akan bermain di hadapan publiknya sendiri pekan depan. </p>

<p align="justify">Lihat Foto Gambar

</p>

</body>

</html>

2. Link ke Gambar

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Untitled Document</title>
</head>
```

```
<body bgcolor="#CCCCCC">
```

```
<p>Ini Berita Fotonya :</p>
```

```
<p></p>
```

```
<p>Sumber : www.goal.com</p>
```



```
<p><a href="link.html">Back to homepage</a></p>
</body>
</html>
```

3. Membuat tabel

```
<html>
<body>
<p>
Setiap tabel dimulai dengan tag table.
Setiap baris tabel dimulai dengan tag tr.
Setiap data dalam tabel dimulai dengan tag td.
</p>
<h4>Satu baris satu kolom dengan border normal:</h4>
<table border="1">
<tr>
<td>100</td>
</tr>
</table>
<h4>Dua baris dan tiga kolom dengan border sangat tebal:</h4>
<table border="15">
<tr>
<td>100</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>400</td>
<td>500</td>
<td>600</td>
</tr>
</table>
<h4>Tabel ini tidak memiliki border:</h4>
<table border="0">
<tr>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>300</td>
<td>400</td>
</tr>
</table>
</body>
</html>
```

4. Merger Cell

```
<html>
<head>
<title>Using Table</title>
</head>

<body>
<table bgcolor="CCCCFF" width="62%" border="1"
cellpadding="0">
<tr>
<td colspan="3" align="center">Quarter 1</td>
<td colspan="3" align="center">Quarter 2</td>
</tr>
<tr align="center">
<td>Jan</td>
```

```

        <td>Feb</td>
        <td>Mar</td>
        <td>Apr</td>
        <td>May</td>
        <td>Jun</td>
    </tr>
    <tr>
        <td>100</td>
        <td>5000</td>
        <td>200</td>
        <td>1500</td>
        <td>2500</td>
        <td>1750</td>
    </tr>
    <tr>
        <td>290</td>
        <td>5050</td>
        <td>2300</td>
        <td>100</td>
        <td>270</td>
        <td>300</td>
    </tr>
</table>
</body>
</html>

```

5. RowSpan

```

<html>
<head>
<title>Rowspan</title>

</head>

<body>
<table bgcolor="lavender" width="75%" border="1"
cellpadding="0">
    <tr>
        <td></td>
        <td></td>
        <td>South</td>
        <td>North</td>
    </tr>
    <tr>
        <td rowspan="3">Quarter 1</td>
        <td>Jan</td>
        <td>1000</td>
        <td>12000</td>
    </tr>
    <tr>
        <td>Feb</td>
        <td>12500</td>
        <td>1345</td>
    </tr>
    <tr>
        <td>Mar</td>
        <td>78090</td>
        <td>71080</td>
    </tr>
</table>
</body>

```

```
</html>
```

MODUL III INTERAKTIVITAS DASAR(FORM DAN GAMBAR)

1. TUJUAN

- Mahasiswa dapat mengenal tag HTML untuk membuat membuat form inputan dari user dan memasukan gambar mengaturnya.
- Mahasiswa dapat membuat halaman web dengan mengimplementasikan tag form dan gambar pada HTML.

2. TEORI DASAR

Bila Anda suatu ketika ingin memasukkan informasi ke suatu situs web, maka anda memerlukan fasilitas khusus untuk keperluan tersebut. Dalam HTML Anda dapat membuat tampilan halaman untuk menerima masukan dari pengunjung situs, kemudian menyimpannya di situs Anda atau langsung dimunculkan kembali sebagai suatu halaman. Disinilah perlunya menggunakan form. Perhatikan tag di bawah ini.

Start Tag	Kegunaan
<form>	Mendefinisikan sebuah form untuk input dari pengunjung
<input>	Mendefinisikan sebuah field input
<textarea>	Mendefinisikan sebuah text-area (sebuah kontrol masukan teks jenis multi-line)
<label>	Mendefinisikan sebuah label ke kontrol
<fieldset>	Mendefinisikan sebuah fieldset
<legend>	Mendefinisikan suatu caption untuk sebuah fieldset
<select>	Mendefinisikan sebuah list yang dapat dipilih (drop-down box)
<optgroup>	Mendefinisikan sebuah kelompok pilihan
<option>	Mendefinisikan sebuah pilihan dalam drop-down box
<button>	Mendefinisikan sebuah tombol

Anda tentu saja tidak ingin membuat halaman web yang hanya berisi teks saja. Agar halaman web Anda lebih menarik, gunakan unsur grafis berupa gambar. Anda dapat memanipulasi gambar dalam suatu halaman web sehingga menghasilkan berbagai efek. Tag di bawah ini digunakan untuk keperluan tersebut.

Start Tag	Kegunaan
	Mendefinisikan sebuah gambar
<map>	Mendefinisikan sebuah image map (peta gambar)
<area>	Mendefinisikan sebuah area di dalam suatu image map

3. LATIHAN

Disini Anda akan berlatih membuat form dengan segala obyeknya. Tetapi hanya sekedar form untuk mengumpulkan data secara interaktif, tanpa memproses data tersebut. Di bagian ini belum akan dibicarakan tentang pemrosesan data, oleh karena itu silakan mencoba-coba setiap obyek dalam form saja, dengan berbagai cara penempatannya yang memungkinkan.

Untuk dapat melihat gambar dalam halaman dokumen Anda, silakan mempersiapkan terlebih dahulu gambar-gambar yang akan digunakan. Untuk gambar bergerak, gunakan saja format animasi GIF. Untuk gambar diam, gunakan format sesuai yang diperlihatkan dalam kode sumber contoh.

1. Field teks dan password

```

<html>
<body>
<form>
Nama depan:
<input type="text" name="namadepan">
<br>
Nama belakang:
<input type="text" name="namabelakang">
<br>
Password:
<input type="password" name="password">
<input name="btnKirim" type="submit" id="btnKirim" value="Kirim">
<input name="btnCancel" type="reset" id="btnCancel" value="Cancel">
</form>
<p>
Perhatikan bahwa ketika Anda mengetikkan karakter-karakter dalam
suatu field password, browser akan menampilkan asterisk atau bullet
saja, bukannya karakter yang diketikkan.
</p>
</body>
</html>

```

2. Checkbox dan tombol radio

```

<html>
<body>
<form>
Saya memiliki sebuah sepeda:
<input type="checkbox" name="Sepeda">
<br>
Saya memiliki sebuah mobil:
<input type="checkbox" name="Mobil" value="ON">
<br>
Jenis kelamin Anda?
<br>
Pria:
<input type="radio" name="Sex" value="pria">
<br>
Wanita:
<input type="radio" name="Sex" value="wanita" checked>
</form>
<p>
Ketika seorang pengguna memberikan klik mouse pada sebuah tombol
radio, tombol tersebut terlihat diberi tanda cek, dan semua tombol
lain yang berada dalam nama yang sama akan kehilangan tanda cek-nya
</p>
</body>
</html>

```

3. Kotak drop down dengan nilai default

```

<html>
<body>
<form>
<select name="mobil">
<option value="volvo">Volvo
<option value="peugeot">Peugeot
<option value="fiat" selected>Fiat
<option value="jaguar">Jaguar
</select>

```


MODUL IV CASE STYLE SHEET

1. TUJUAN

- Mahasiswa dapat mengenal tag CSS dan kegunaannya.
- Mahasiswa dapat membuat halaman web dengan mengimplementasikan tag CSS.

2. TEORI DASAR

PENGERTIAN CSS

CSS = Cascading Style Sheets (Bahasa lembar Gaya). CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / markup language. Jika kita berbicara dalam konteks web, bisa di artikan secara bebas sebagai : CSS merupakan bahasa yang digunakan untuk mengatur tampilan / desain suatu halaman HTML.

Beberapa hal yang dapat dilakukan dengan CSS.

- Perancangan desain text dapat dilakukan dengan mendefinisikan fonts (huruf) , colors (warna), margins (ukuran), latar belakang (background), ukuran font (font sizes) dan lain-lain. Elemen-elemen seperti colors (warna) , fonts (huruf), sizes (ukuran) dan spacing (jarak) disebut juga “styles”.
- Cascading Style Sheets juga bisa berarti meletakkan styles yang berbeda pada layers (lapisan) yang berbeda.

Ada 3 cara untuk memasang CSS pada dokumen HTML yaitu:

- External Style Sheet (file CSS berbeda dari file HTML),
- Internal Style Sheet (Kode CSS dipasang di dalam tag head HTML)
- Inline Style Sheet (Kode CSS langsung dipasang di tag HTML, tidak direkomendasikan).

SATUAN DALAM CSS

1. Statik

- * in -- satuan inchi
- * cm -- satuan centimeter
- * mm -- satuan milimeter
- * pt -- satuan point (1point = 1/72 inchi)
- * pc -- satuan pica (1pica = 12 point)
- * px -- satuan pixel (satu titik gambar terkecil dalam layar monitor)

2. Relatif

- * % -- satuan persen
- * em -- atau ems (1em = ukuran font yang tengah ada dalam elemen)
- * ex -- 1ex = x-height suatu font (x-height biasanya setengah ukuran font)

MENULIS CSS

Dalam hal ini ada beberapa istilah dan kosep dasar dari css. Tapi sebelum menuju pembahasan lebih lanjut, akan lebih baik kita harus membiasakan dengan bagaimana suatu style harus ditulis. Berikut adalah contoh mudah penulisan css pada suatu halaman web:

Contoh CSS 1

```
<html>
```

```

<head>
<style Type="text/css">
  h1{font-family:tahoma}
  h2{color: blue}
  p{font-size:11pt ; font-style: italic}
</style>
</head>
<body>
<h2>NuSinau</h2>

```

<p>NuSinau adalah suatu website tempat berbagi berbagai ilmu pengetahuan.

```

</p></body>
</html>

```

Sintak CSS (Syntax CSS)

selector (properti: nilai)

```

body (color: blue)
p {font-family: tahoma}
h1 {font-weight: bold}
dan lain-lain

```

SELECTOR

Selector biasanya merupakan elemen HTML atau tag dimana berbagai properti diberi suatu nilai. Dalam sebuah selector, setiap properti mempunyai nilai masing-masing dan sejumlah properti dengan nilai-nilai terkait dapat diatur.

Seperti pada contoh diatas; body, p, dan h1 adalah suatu selector dimana warna (color), nama font (font-family) dan jenis font (font-weight) diberi nilai untuk tiap propertinya (tahoma, bold, blue, dll).

Contoh

```
h1 (font-size: 20px; text-align: left; color: blue; font-family: Arial;)
```

dapat pula ditulis seperti:

```
h1 (font-size: 20px;
text-align: left;
color: blue;
font-family: Arial;)
```

Pengelompokan Selector (Grouping of Selectors)

Seperti dapat dilihat, satu selector dapat memiliki beberapa properti. Dalam konteks yang berbeda, beberapa selector dapat mempunyai properti dengan nilai yang sama. Dengan begitu harus dilakukan pengelompokan selector.

```
h1 , body , p , ul
{
  color : blue;
}
```

Pada contoh diatas, elemen header, body, paragraph, unordered list berada pada group yang sama. Semua elemen tersebut nantinya akan berwarna biru.

PENGELOMPOKAN DALAM CSS

- Untuk berbagi style yang sama, bila beberapa elemen mempunyai nilai properti yang sama (misalnya font, ukuran, warna, dll).
- Dengan satu perintah dapat memberikan properti style pada elemen yang berbeda dengan nilai yang sama.

Modul Praktikum Pemrograman Web – STMIK Yadika BANGIL

- Untuk menghemat waktu bila terjadi penambahan properti dan juga ketika merubah properti di elemen.

PADDING, MARGIN DAN BORDER

Padding : Menentukan jarak komponen body ke border atau Ukuran jarak bagian dalam

Border : Adalah garis tepi dari komponen

Margin : Adalah Ukuran jarak bagian luar atau ukuran jarak sesudah Border

Padding

ditulis dengan CSS padding:5px 5px 5px 5px; urutan nilai angkanya adalah atas, kanan, bawah dan kiri, atau Anda bisa menggunakan

padding-left:5px; ini adalah untuk pengaturan padding bagian kiri

padding-right:5px; ini adalah untuk pengaturan padding kanan

padding-top:5px; untuk bagian atas dan

padding-bottom:5px; untuk bagian bawah, Ingat satuan px(pixels) bisa kamu ganti sesuai satuan yang lain yang sesuai

Border

Ditulis dengan CSS border:1px dotted #000000; urutan penggunaannya adalah ukuran border, style border dan warna border, atau bisa menggunakan

border-width:1px; ini adalah ketebalan border

border-style:dotted; ini adalah jenis bordernya bisa kamu ganti dengan dashed, solid, double, groove, ridge, inset, outset dan lainnya

border-color:#FFFFFF; ini adalah warna dari border.. kamu bisa menggantinya code warnanya (www.colorschemer.com/online)

Margin

Ditulis dengan CSS margin:5px 5px 5px 5px; urutannya atas, kanan, bawah dan kiri, atau bisa menggunakan seperti padding diatas

margin-left:5px;

margin-right:5px;

margin-top:5px;

margin-bottom:5px;

keterangan lainnya bisa mengikuti keterangan padding diatas

3. LATIHAN

Bagian ini mencakup penggunaan CSS untuk mengubah tampilan standar pada halaman web Anda. Tidak semua tag style digunakan disini karena jumlahnya sangat banyak, silakan Anda cari referensi lain sebagai tambahan untuk mempelajari CSS. Seperti biasa, simpan file-file Anda dalam folder Anda sendiri, kemudian cobalah langsung untuk melihat hasil setiap contoh, lalu ubahlah isinya untuk melihat variasi tampilannya yang memungkinkan. Perhatikan bahwa semua contoh disini menggunakan CSS internal, silakan Anda berlatih untuk mengubahnya menjadi CSS eksternal. Gunakan sembarang gambar yang dapat Anda temui untuk contoh yang memerlukan gambar (sesuaikan dulu dengan nama file dalam contoh).

1. Pengaturan latar belakang

```
<html>
<head>
<style type="text/css">
body
{
background: #00ff00
url("gambar.gif")
no-repeat fixed
```

```

center center
}
</style>
</head>
<body>
<p>
<b>Catatan:</b>
Properti background-attachment dan background-position tidak bisa
digunakan dalam browser Netscape 4.0.
</p>
<p>Ada teks disini</p>
<p>Tuliskan sekehendak kalian</p>
<p>Terseher apa saja</p>
<p>Yang penting ada tulisannya</p>
</body>
</html>

```

2. Penanganan font

```

<html>
<head>
<style type="text/css">
p.normal {font-weight: normal}
p.tebal {font-weight: bold}
p.teballl {font-weight: 900}
p.lengkap
{
font: italic small-caps 900 12px arial
}
</style>
</head>
<body>
<p class="normal">
Perhatikan paragraf ini</p>
<p class="tebal">
Bandingkan dengan paragraf diatasnya</p>
<p class="teballl">
Bandingkan yang ini juga</p>
<p class="lengkap">
Paragraf dengan semua atribut font dimasukkan</p>
</body>
</html>

```

3. Pembuatan border

```

<html>
<head>
<style type="text/css">
p.borderous
{
border-style: double;
border-width: 5px 10px 1px medium;
border-color: #ff0000
#00ff00 #0000ff
rgb(250,0,255)
}
</style>
</head>
<body>
<p class="borderous">
<b>Catatan:</b> Properti "border-color" dan "border-width" tidak
dikenali oleh browser Internet Explorer bila digunakan secara
langsung. Gunakan properti "border-style" untuk menyiapkan border
dalam Internet Explorer.<br>

```

```

Internet Explorer 5.5 mendukung semua border style. Internet
Explorer 4.0 - 5.0 tidak mendukung nilai "dotted" dan "dashed" </p>
</body>
</html>

```

4. Pengaturan margin

```

<html>
<head>
<style type="text/css">p.margin {margin: 2cm 4cm 3cm 80px}
</style>
</head>
<body>
<p>
Yang ini adalah paragraf biasa tanpa pengaturan margin, sepertinya
biasa saja.
</p>
<p class="margin">
Bandingkan dengan tampilan paragraf ini. Coba Anda kecilkan ukuran
window browser agar terlihat lebih jelas perbedaannya.
</p>
<p>
This is a paragraph
</p>
</body>
</html>

```

5. Mengatur *padding* pada sel tabel

```

<html>
<head>
<style type="text/css">
td {padding: 1.5cm}
td.value2 {padding: 0.5cm 2.5cm}
</style>
</head>
<body>
<table border="1">
<tr>
<td>
Ini adalah tabel dengan sel yang menggunakan padding sama pada
setiap sisinya (1,5cm).
</td>
</tr>
</table>
<br>
<table border="1">
<tr>
<td class="value2">
Bandingkan dengan tampilan tabel ini. Padding atas dan bawah
memiliki nilai 0,5cm, sedangkan padding kanan dan kiri nilainya
2,5cm.
</td>
</tr>
</table>
</body>
</html>

```

6. Menggunakan list

```

<html>
<head>
<style type="text/css">
ul
{
list-style: square outside url("gambarkecil.gif")

```

```

}
</style>
</head>
<body>
<p>Disini Anda menggunakan gambar berukuran kecil sebagai bullet
untuk list. Properti outside akan menyebabkan list menjadi lebih
rapi bila terdiri atas item yang berupa kalimat panjang dan window
browser berukuran kecil.</p>
<ul>
<li>Kopi</li>
<li>Teh</li>
<li>Coca Cola</li>
</ul>
</body>
</html>

```

7. Elemen latar depan dan latar belakang

```

<html>
<head>
<style type="text/css">
img.x1
{
position:absolute;
left:0;
top:0;
z-index:1
}
img.x2
{
position:absolute;
left:0;
top:100;
z-index:-1
}
</style>
</head>
<body>
<p><b>Catatan:</b> Netscape 4 tidak mendukung properti "z-
index".</p>
<h1>Sepertinya Heading ini terhalang oleh gambar</h1>

<p>Nilai default z-index adalah 0. Z-index 1 memiliki prioritas
lebih tinggi, sedangkan Z-index -1 memiliki prioritas lebih
rendah.</p>
<p></p>

<p>Sekarang Anda bisa membandingkan dengan tampilan di bagian bawah
ini. Apakah masih terhalang gambar?</p>
</body>
</html>

```

8. Mengubah tampilan Link

```

<html>
<head>
<style type="text/css">
a:link {color: #ff0000}
a.one:visited {color: #0000ff}
a.one:hover {color: #ffcc00}
a.two:visited {color: #0000ff}
a.two:hover {font-size: 150%}
a.three:visited {color: #0000ff}
a.three:hover {background: #66ff66}

```

```
a.four:visited {color: #0000ff}
a.four:hover {font-family: fixedsys}
a.five:visited {color: #0000ff; text-decoration: line-through}
a.five:hover {text-decoration: overline}
</style>
</head>
<body>
<p>Gerakkan pointer mouse di atas link-link di bawah ini untuk
membuat link tersebut berubah tampilannya.</p>
<p><a class="one" href="hlm1.htm">Mengubah properti color</a></p>
<p><a class="two" href="hlm2.htm"> Mengubah properti font-
size</a></p>
<p><a class="three" href="hlm3.htm"> Mengubah properti background-
color</a></p>
<p><a class="four" href="hlm4.htm"> Mengubah properti font-
family</a></p>
<p><a class="five" href="hlm5.htm"> Mengubah properti text-
decoration</a></p>
</body>
</html>
```

MODUL V

JAVASCRIPT:DASAR, VARIABEL, & FUNGSI

1. TUJUAN

- Mahasiswa dapat mengenal Dasar pemrograman Javascript.
- Mahasiswa dapat memahami variable dan mengimplementasikan fungsi dasar yang disediakan javascript.

2. TEORI DASAR

Setelah Anda berlatih dan menguasai segi tampilan dari web menggunakan HTML dan CSS, sudah saatnya Anda berlatih untuk mengenal pemrograman web yang sebenarnya, yaitu pemrograman skrip. Mulai dari tahap ini Anda akan belajar JavaScript sebagai skrip dasar yang dikenal semua browser (skrip pada sisi client). Anda tidak akan langsung melompat belajar skrip dari sisi server, karena skrip dari sisi server akan selalu memaksa Anda untuk menggunakan satu produk tertentu (ASP, CF, PHP, dan lain-lain), dan biasanya Anda akan menjadi fanatik pada satu produk saja tanpa mengetahui kemampuan produk lain. Anda juga tidak akan belajar bahasa pemrograman Java secara khusus, karena Java fokusnya adalah portabilitas, bukan pada web saja. Disamping JavaScript, bilamana waktu memungkinkan Anda nantinya juga akan berlatih menggunakan VBScript (salah satu anggota keluarga Visual Basic yang mencakup VB, VBScript, VBA/macro, ASP, VB.net, dan ASP.net).

Operator Aritmetika

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Operator Penunjukan

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Operator Perbandingan

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Operator Logika

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 x != y returns true

3. LATIHAN

Sederhana saja, agar Anda mau membiasakan diri dengan JavaScript, maka Anda harus banyak berlatih mulai dari yang paling dasar. Buatlah semua contoh di bawah ini, jalankan untuk melihat hasilnya, dan lakukan modifikasi untuk lebih mengenal dan mengetahui kegunaan skrip yang baru Anda buat.

1. Menuliskan teks

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

2. Memformat teks dengan tag HTML

```
<html>
<body>
<script type="text/javascript">
document.write("<h1>Hello World!</h1>")
</script>
</body>
</html>
```

3. JavaScript yang diletakkan pada bagian HEAD

```
<html>
<head>
<script type="text/javascript">
```

```

function message()
{
alert("This alert box was called with the onload event")
}
</script>
</head>
<body onload="message()" >
</body>
</html>

```

4. JavaScript yang diletakkan pada bagian BODY

```

<html>
<head>
</head>
<body>
<script type="text/javascript">
document.write("This message is written when the page loads")
</script>
</body>
</html>

```

5. JavaScript eksternal

```

<html>
<head>
</head>
<body>
<script src="xxx.js">
</script>
<p>
The actual script is in an external script file called "xxx.js".
</p>
</body>
</html>

```

6. Deklarasi variabel, memberi nilai, dan menampilkannya

```

<html>
<body>
<script type="text/javascript">
var name = "Hege"
document.write(name)
document.write("<h1>"+name+"</h1>")
</script>
<p>This example declares a variable, assigns a value to it, and then
displays the variable.</p>
<p>Then the variable is displayed one more time, only this time as a
heading.</p>
</body>
</html>

```

7. Fungsi

```

<html>
<head>
<script type="text/javascript">
function myfunction()
{
alert("HELLO")
}
</script>
</head>

```



```

<body>
<form>
<input type="button"
onclick="myfunction()"
value="Call function">
</form>
<p>By pressing the button, a function will be called. The function
will alert a message.</p>
</body>
</html>

```

8. Fungsi dengan argumen

```

<html>
<head>
<script type="text/javascript">
function myfunction(txt)
{
alert(txt)
}
</script>
</head>
<body>
<form>
<input type="button"
onclick="myfunction('Hello')"
value="Call function">
</form>
<p>By pressing the button, a function with an argument will be
called. The function will alert
this argument.</p>
</body>
</html>

```

9. Fungsi dengan argumen (lagi)

```

<html>
<head>
<script type="text/javascript">
function myfunction(txt)
{
alert(txt)
}
</script>
</head>
<body>
<form>
<input type="button"
onclick="myfunction('Good Morning!')"
value="In the Morning">
<input type="button"
onclick="myfunction('Good Evening!')"
value="In the Evening">
</form>
<p>
When you click on one of the buttons, a function will be called. The
function will alert
the argument that is passed to it.
</p>
</body>
</html>

```

10. Fungsi yang mengembalikan suatu nilai

```
<html>
<head>
<script type="text/javascript">
function myFunction()
{
return ("Hello, have a nice day!")
}
</script>
</head>
<body>
<script type="text/javascript">
document.write(myFunction())
</script>
<p>The script in the body section calls a function.</p>
<p>The function returns a text.</p>
</body>
</html>
```

11. Fungsi dengan argumen yang mengembalikan suatu nilai

```
<html>
<head>
<script type="text/javascript">
function total(numberA,numberB)
{
return numberA + numberB
}
</script>
</head>
<body>
<script type="text/javascript">
document.write(total(2,3))
</script>
<p>The script in the body section calls a function with two
arguments, 2 and 3.</p>
<p>The function returns the sum of these two arguments.</p>
</body>
</html>
```

MODUL VI JAVASCRIPT: PERULANGAN DAN PENGKONDISIAN

1. TUJUAN

- Mahasiswa dapat mengenal memahami konsep pengkondisian dan perulangan dengan menggunakan javascript.
- Mahasiswa dapat membuat pengkondisian dan perulangan dengan menggunakan javascript.

2. TEORI DASAR

Bagian berikutnya yang masih merupakan dasar dari JavaScript adalah percabangan dan putaran. Dapat dikatakan bahwa percabangan dan putaran merupakan salah satu inti metode dalam semua bahasa pemrograman yang ada di dunia, karena dengan percabangan dan putaran akan dihasilkan sebuah program yang dinamis, dan bukan program yang linear serta bersifat statik. Karena JavaScript merupakan salah satu cara dalam melakukan pemrograman web di sisi client, maka JavaScript juga memiliki kemampuan ini.

Untuk menangani kondisi pada program kita membutuhkan logika yang menyeleksi nilai dari sebuah kondisi pada program tersebut. Kita bisa mengecek satu atau dua kondisi yang terjadi pada sebuah variabel dengan logika IF atau bisa juga mengecek banyak kondisi pada variabel dengan logika SWITCH.

Logika IF

Biasanya digunakan sebagai syarat untuk proses program. Misalnya sebuah program berjalan jika syarat telah terpenuhi atau bahkan sebuah program dihentikan jika sebuah variabel telah mencapai nilai tertentu.

Pada logika IF, suatu perintah tersebut akan dijalankan hanya jika kondisi pada logika IF bernilai benar atau dengan kata lain kondisi telah terpenuhi/tercapai.

Cara Penulisan Logika IF Pada Javascript :

```
if (kondisi) {
    //sekelompok statemen program yang dijalankan jika kondisi bernilai benar
}
```

Note: Penulisan if harus menggunakan huruf kecil, jika tidak akan mengakibatkan error pada javascript.

Logika IF ... ELSE

Prinsipnya sama dengan logika IF namun ditambahkan penanganan jika kondisi yang dicek bernilai salah. Jadi dengan logika IF...ELSE kita bisa menangani saat kondisi bernilai benar dan saat kondisi bernilai salah.

Cara Penulisan Logika IF...ELSE Pada Javascript :

```
if (kondisi) {
    //sekelompok statemen program yang dijalankan jika kondisi bernilai benar
}
else {
    //sekelompok statemen program yang dijalankan jika kondisi bernilai salah
}
```

Note: Penulisan if dan else harus menggunakan huruf kecil, jika tidak akan mengakibatkan error pada javascript.

Logika SWITCH

Digunakan untuk menangani banyak kondisi yang jika digunakan IF...ELSE IF akan terlalu banyak dan membingungkan.

Cara penulisan logika SWITCH pada javascript :

```
switch(expression) {
  case a:
    statemen blok kode jika expresion bernilai a
    break;
  case b:
    statemen blok kode jika expresion bernilai b
    break;
  ....//bisa diisi sampai banyak case...
  default:
    default statemen blok kode
}
```

Pada javascript disediakan alat perulangan untuk mempermudah penanganan script yang berulang dengan FOR dan WHILE. Perulangan FOR akan melakukan perulangan dengan awalan dan akhiran tertentu serta proses yang bisa menentukan akhiran perulangan FOR. Sedangkan perulangan WHILE hanya akan mengulang proses sampai syarat yang diminta WHILE terpenuhi.

Perulangan FOR

Ini cara yang banyak dan umum dipakai untuk membuat perulangan proses. Dengan perulangan FOR kita bisa menentukan awal dan akhir syarat perulangan serta membuat proses yang menentukan akhir perulangan FOR.

Cara Penulisan Perulangan FOR Pada Javascript :

```
for (statement 1; statement 2; statement 3) {
  ...//kode yang akan dijalankan
}
```

Perulangan WHILE

Dengan logika WHILE maka perulangan akan berjalan selama kondisi dalam WHILE bernilai benar (true). Jadi agar perulangan berhenti dan tidak mengacaukan browser maka didalam statemen kode yang diulang harus ditulis juga perintah yang akan menghentikan perulangan atau membuat kondisi menjadi salah.

Cara Penulisan Perulangan WHILE Pada Javascript :

```
while (kondisi) {
  ...//kode yang akan dijalankan jika kondisi bernilai benar
}
```

Perulangan DO...WHILE

Prinsipnya sama dengan while, yaitu proses mengerjakan kode program lalu mengulang proses tersebut selama sebuah kondisi bernilai benar. Perulangan berhenti jika kondisi tidak terpenuhi lagi.

Cara Penulisan Perulangan DO...WHILE Pada Javascript :

```
do {
  ...//kode yang akan dijalankan sekali dan diulang selama kondisi bernilai benar
}while (kondisi)
```

3. LATIHAN

Disini Anda akan berlatih melakukan membentuk percabangan (pemilihan berdasar kondisi) dan membuat putaran dalam skrip. Jalankan contoh-contoh di bawah ini,

kemudian perhatikan baik-baik dasar penggunaan dari setiap elemen untuk percabangan dan perulangan yang disertakan. Cobalah untuk mengembangkannya sendiri dalam bentuk modifikasi.

1. pengecekan sederhana dengan if

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time < 10)
{
document.write("<b>Good morning</b>")
}
</script>
<p>
This example demonstrates the If statement.
</p>
<p>
If the time on your browser is less than 10, you will get a "Good
morning" greeting.
</p>
</body>
</html>
```

2. pengecekan sederhana dengan if..else

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time < 10)
{
document.write("<b>Good morning</b>")
}
else
{
document.write("<b>Good day</b>")
}
</script>
<p>
This example demonstrates the If...Else statement.
</p>
<p>
If the time on your browser is less than 10, you will get a "Good
morning" greeting.
Otherwise you will get a "Good day" greeting.
</p>
</body>
</html>
```

4. pengecekan sederhana dengan switch case

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
theDay=d.getDay()
switch (theDay)
{
```

```

    case 5:
        document.write("Finally Friday")
    break
    case 6:
        document.write("Super Saturday")
    break
    case 0:
        document.write("Sleepy Sunday")
    break
    default:
        document.write("I'm really looking forward to this
weekend!")
}
</script>
<p>This example demonstrates the switch statement.</p>
<p>You will receive a different greeting based on what day it
is.</p>
<p>Note that Sunday=0, Monday=1, Tuesday=2, etc.</p>
</body>
</html>

```

5. perulangan dengan for

```

<html>
<body>
<script type="text/javascript">
for (i = 0; i <= 5; i++)
{
document.write("The number is " + i)
document.write("<br>")
}
</script>
<p>Explanation:</p>
<p>The for loop sets <b>i</b> equal to 0.</p>
<p>As long as <b>i</b> is less than , or equal to, 5, the loop will
continue to run.</p>
<p><b>i</b> will increase by 1 each time the loop runs.</p>
</body>
</html>

```

7. perulangan dengan while

```

<html>
<body>
<script type="text/javascript">
i = 0
while (i <= 5)
{
document.write("The number is " + i)
document.write("<br>")
i++
}
</script>
<p>Explanation:</p>
<p><b>i</b> equal to 0.</p>
<p>While <b>i</b> is less than , or equal to, 5, the loop will
continue to run.</p>
<p><b>i</b> will increase by 1 each time the loop runs.</p>
</body>
</html>

```

8. perulangan dengan do..while

```
<!DOCTYPE html>
<html>
<body>
<p>Silahkan klik tombol untuk mengulang tampilan angka 0-9</p>
<button onclick="showAngka()">Tampilkan</button>
<p id="view"></p>
<script>
function showAngka() {
    var text = "";
    var i = 0;
    do {
        text += "<br>Angka = " + i;
        i++;
    } while (i < 10)
    document.getElementById("view").innerHTML = text;
}
</script>
</body>
</html>
```

MODUL VII JAVASCRIPT: OBYEK STRING, DATE, & ARRAY

1. TUJUAN

- Mahasiswa dapat mengenal memahami konsep objek string, fungsi tanggal dan array pada javascript.
- Mahasiswa dapat membuat halaman web dengan menggunakan string, fungsi tanggal dan array pada javascript.

2. TEORI DASAR

Setelah Anda terbiasa dengan dasar-dasar JavaScript beserta penggunaan variabel, termasuk mulai mengenal pemrograman skrip fungsi, maka selanjutnya Anda dapat mulai berlatih menggunakan berbagai obyek yang tersedia dalam JavaScript. Seperti telah disampaikan dalam perkuliahan, JavaScript menyediakan banyak obyek *built-in* yang dapat langsung diaplikasikan oleh pengguna. Dengan beragam modifikasi atribut serta nilai yang dapat diberikan pada suatu obyek, maka Anda akan mendapatkan beragam hasil pula.

Properti obyek string

Properties	Explanation
Length	Returns the number of characters in a string

Metode obyek string

Methods	Explanation
anchor()	Returns a string as an anchor
big()	Returns a string in big text
blink()	Returns a string blinking
bold()	Returns a string in bold
charAt()	Returns the character at a specified position
charCodeAt()	Returns the Unicode of the character at a specified position
concat()	Returns two concatenated strings
fixed()	Returns a string as teletype
fontcolor()	Returns a string in a specified color
fontsize()	Returns a string in a specified size
fromCharCode()	Returns the character value of a Unicode
indexOf()	Returns the position of the first occurrence of a specified string inside another string. Returns -1 if it never occurs
italics()	Returns a string in italic
lastIndexOf()	Returns the position of the first occurrence of a specified string inside another string. Returns -1 if it never occurs. Note: This method starts from the right and moves left!
link()	Returns a string as a hyperlink

match()	Similar to indexOf and lastIndexOf, but this method returns the specified string, or "null", instead of a numeric value
replace()	Replaces some specified characters with some new specified characters
search()	Returns an integer if the string contains some specified characters, if not it returns -1
slice()	Returns a string containing a specified character index
small()	Returns a string as small text
split()	Replaces a string with a comma
strike()	Returns a string strikethrough
sub()	Returns a string as subscript
substr()	Returns the specified characters. 14,7 returns 7 characters, from the 14th character (starts at 0)
substring()	Returns the specified characters. 7,14 returns all characters from the 7th up to but not including the 14th (starts at 0)
sup()	Returns a string as superscript
toLowerCase()	Converts a string to lower case
toUpperCase()	Converts a string to upper case

Metode obyek date

Methods	Explanation
Date()	Returns a Date object
getDate()	Returns the date of a Date object (from 1-31)
getDay()	Returns the day of a Date object (from 0-6. 0=Sunday, 1=Monday, etc.)
getMonth()	Returns the month of a Date object (from 0-11. 0=January, 1=February, etc.)
getFullYear()	Returns the year of a Date object (four digits)
getYear()	Returns the year of a Date object (from 0-99). Use getFullYear instead !!
getHours()	Returns the hour of a Date object (from 0-23)
getMinutes()	Returns the minute of a Date object (from 0-59)
getSeconds()	Returns the second of a Date object (from 0-59)
getMilliseconds()	Returns the millisecond of a Date object (from 0-999)
getTime()	Returns the number of milliseconds since midnight 1/1-1970
getTimezoneOff	Returns the time difference between the

set()	user's computer and GMT
getUTCDate()	Returns the date of a Date object in universal (UTC) time
getUTCDay()	Returns the day of a Date object in universal time
getUTCMonth()	Returns the month of a Date object in universal time
getUTCFullYear())	Returns the four-digit year of a Date object in universal time
getUTCHours()	Returns the hour of a Date object in universal time
getUTCMinutes()	Returns the minutes of a Date object in universal time
getUTCSeconds ()	Returns the seconds of a Date object in universal time
getUTCMilliseconds())	Returns the milliseconds of a Date object in universal time
parse()	Returns a string date value that holds the number of milliseconds since January 01 1970 00:00:00
setDate()	Sets the date of the month in the Date object (from 1-31)
setFullYear()	Sets the year in the Date object (four digits)
setHours()	Sets the hour in the Date object (from 0-23)
setMilliseconds())	Sets the millisecond in the Date object (from 0-999)
setMinutes())	Set the minute in the Date object (from 0-59)
setMonth())	Sets the month in the Date object (from 0-11. 0=January, 1=February)
setSeconds())	Sets the second in the Date object (from 0-59)
setTime())	Sets the milliseconds after 1/1-1970
setYear())	Sets the year in the Date object (00-99)
setUTCDate())	Sets the date in the Date object, in universal time (from 1-31)
setUTCDay())	Sets the day in the Date object, in universal time (from 0-6. Sunday=0, Monday=1, etc.)
setUTCMonth())	Sets the month in the Date object, in universal time (from 0-11. 0=January, 1=February)
setUTCFullYear()	Sets the year in the Date object, in universal time (four digits)
setUTCHour())	Sets the hour in the Date object, in universal time (from 0-23)
setUTCMinutes()	Sets the minutes in the Date object, in universal time (from 0-59)
setUTCSeconds ()	Sets the seconds in the Date object, in

()	universal time (from 0-59)
setUTCMilliseconds()	Sets the milliseconds in the Date object, in universal time (from 0-999)
toGMTString()	Converts the Date object to a string, set to GMT time zone
toLocaleString()	Converts the Date object to a string, set to the current time zone
toString()	Converts the Date object to a string

Metode obyek array

Methods	Explanation
length	Returns the number of elements in an array. This property is assigned a value when an array is created
concat()	Returns an array concatenated of two arrays
join()	Returns a string of all the elements of an array concatenated together
reverse()	Returns the array reversed
slice()	Returns a specified part of the array
sort()	Returns a sorted array

3. LATIHAN

Seperti dalam pertemuan sebelumnya, gunakan contoh-contoh di bawah ini agar Anda dapat lebih mengenal akrab obyek-obyek dalam JavaScript dan cara memanfaatkannya. Perhatikan bahwa Anda benar-benar harus menelaah setiap baris skrip agar Anda tahu proses kerja setiap skrip yang Anda buat.

1. Menghitung karakter suatu string

```
<html>
<body>
<script type="text/javascript">
var str="W3CSchools is great!"
document.write("</p>" + str + "</p>")
document.write(str.length)
</script>
</body>
</html>
```

2. Menguji apakah string berisi karakter tertentu, dengan indexOf()

```
<html>
<body>
<script type="text/javascript">
var str="W3CSchools is great!"
var pos=str.indexOf("School")
if (pos>=0)
{
document.write("School found at position: ")
document.write(pos + "<br />")
}
else
{
document.write("School not found!")
}
```

```

}
</script>
<p>This example tests if a string contains a specified word. If the
word is found it returns the position of the first character of the
word in the original string. Note: The first position in the string
is 0!</p>
</body>
</html>

```

3. Menguji apakah string berisikan karakter tertentu, dengan match()

```

<html>
<body>
<script type="text/javascript">
var str = "W3CSchools is great!"
document.write(str.match("great"))
</script>
<p>This example tests if a string contains a specified word. If the
word is found it returns the word.</p>
</body>
</html>

```

4. Memunculkan bagian string tertentu

```

<html>
<body>
<script type="text/javascript">
var str="W3CSchools is great!"
document.write(str.substr(2,6))
document.write("<br /><br />")
document.write(str.substring(2,6))
</script>
<p>
The substr() method returns a specified part of a string. If you
specify (2,6) the returned string will be from the second character
(start at 0) and 6 long.
</p>
<p>
The substring() method also returns a specified part of a string. If
you specify (2,6) it returns all characters from the second
character (start at 0) and up to, but not including, the sixth
character.
</p>
</body>
</html>

```

5. Mengubah menjadi huruf besar atau kecil

```

<html>
<body>
<script type="text/javascript">
var str=("Hello JavaScripters!")
document.write(str.toLowerCase())
document.write("<br>")
document.write(str.toUpperCase())
</script>
</body>
</html>

```

6. Membuat array berisi nama-nama

```

<html>
<body>

```

```

<script type="text/javascript">
var famname = new Array(6)
famname[0] = "Jan Egil"
famname[1] = "Tove"
famname[2] = "Hege"
famname[3] = "Stale"
famname[4] = "Kai Jim"
famname[5] = "Borge"
for (i=0; i<6; i++)
{
document.write(famname[i] + "<br>")
}
</script>
</body>
</html>

```

7. Menghitung elemen yang berada dalam array

```

<html>
<body>
<script type="text/javascript">
var famname = new Array("Jan Egil", "Tove", "Hege", "Stale", "Kai
Jim", "Borge")
for (i=0; i<famname.length; i++)
{
document.write(famname[i] + "<br>")
}
</script>
</body>
</html>

```

8. Tanggal hari ini

```

<html>
<body>
<script type="text/javascript">
var d = new Date()
document.write(d.getDate())
document.write(".")
document.write(d.getMonth() + 1)
document.write(".")
document.write(d.getFullYear())
</script>
</body>
</html>

```

9. Jam saat ini

```

<html>
<body>
<script type="text/javascript">
var d = new Date()
document.write(d.getHours())
document.write(".")
document.write(d.getMinutes())
document.write(".")
document.write(d.getSeconds())
</script>
</body>
</html>

```

10. Mengeset tanggal

```

<html>
<body>
<script type="text/javascript">
var d = new Date()
d.setFullYear("1990")
document.write(d)
</script>
</body>
</html>

```

11. Melihat waktu UTC

```

<html>
<body>
<script type="text/javascript">
var d = new Date()
document.write(d.getUTCHours())
document.write(".")
document.write(d.getUTCMinutes())
document.write(".")
document.write(d.getUTCSeconds())
</script>
</body>
</html>

```

12. Memunculkan nama-nama hari

```

<html>
<body>
<script type="text/javascript">
var d=new Date()
var weekday=new
Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","S
aturday")
document.write("Today is " + weekday[d.getDay()])
</script>
</body>
</html>

```

13. Memunculkan tanggal lengkap

```

<html>
<body>
<script type="text/javascript">
var d=new Date()
var weekday=new
Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","S
aturday")
var monthname=new
Array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","N
ov","Dec")
document.write(weekday[d.getDay()] + " ")
document.write(d.getDate() + ". ")
document.write(monthname[d.getMonth()] + " ")
document.write(d.getFullYear())
</script>
</body>
</html>

```

14. Membuat jam

```

<html>
<head>

```

```
<script type="text/javascript">
var timer = null
function stop()
{
clearTimeout(timer)
}
function start()
{
var time = new Date()
var hours = time.getHours()
var minutes = time.getMinutes()
minutes=((minutes < 10) ? "0" : "") + minutes
var seconds = time.getSeconds()
seconds=((seconds < 10) ? "0" : "") + seconds
var clock = hours + ":" + minutes + ":" + seconds
document.forms[0].display.value = clock
timer = setTimeout("start()",1000)
}
</script>
</head>
<body onload="start()" onunload="stop()">
<form>
<input type="text" name="display" size="20">
</form>
</body>
</html>
```