

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/316664126>

Modul Pemrograman Web. Dimulai dari penggunaan HTML sampai Javascript

Chapter · February 2014

CITATIONS

0

READS

146,594

2 authors, including:



Bernad Tagger

Stmik Amikom Yogyakarta

4 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

WEB PROGRAMMING

MODUL PRAKTIKUM

**Modul yang dibuat sebagai
acuan dalam kegiatan perkuliahan
praktikum Mata Kuliah Pemrograman
Web di STMIK Amikom Yogyakarta**

Disusun Oleh

**Bernadhed, M.Kom
Mulia Sulistiyono, M.Kom**

BAB 1. PENGENALAN HTML

Tujuan

1. Mahasiswa mengetahui dasar-dasar HTML
2. Mahasiswa mengetahui HTML Editor
3. Mahasiswa mampu membuat dokumen HTML dan melakukan formatting terhadap dokumen tersebut.
4. Mahasiswa mampu melakukan text formatting
5. Mahasiswa mampu melakukan listing pada dokumen HTML
6. Mahasiswa mampu membuat listing bersarang pada dokumen HTML

Overview

Sebuah aplikasi web dibangun menggunakan tag HTML. Pada sesi pertemuan ini akan dipelajari bagaimana membuat dokumen HTML sederhana, memformat tampilan pada dokumen HTML dan membuat list yang biasa kita gunakan untuk menampilkan daftar dengan urutan tertentu. Untuk praktikum ini akan menggunakan notepad sebagai HTML editornya.

1.1. Format Penulisan HTML

Untuk membuat suatu halaman HTML standar digunakan tag <html>, <head>, <body> beserta tag penutupnya masing-masing. Susunan tag – tag tersebut membentuk dokumen HTML sebagai berikut :

```
1 <html>
2   <head>
3     ... informasi tentang dokumen HTML
4   </head>
5
6   <body>
7     ... informasi yang akan ditampilkan di web browser
8   </body>
9 </html>
10
```

Gambar 1.1 Format Penulisan HTML

1.2. Memulai Membuat Dokumen dengan Format HTML

Langkah – langkah untuk memulai membuat dokumen HTML sebagai berikut :

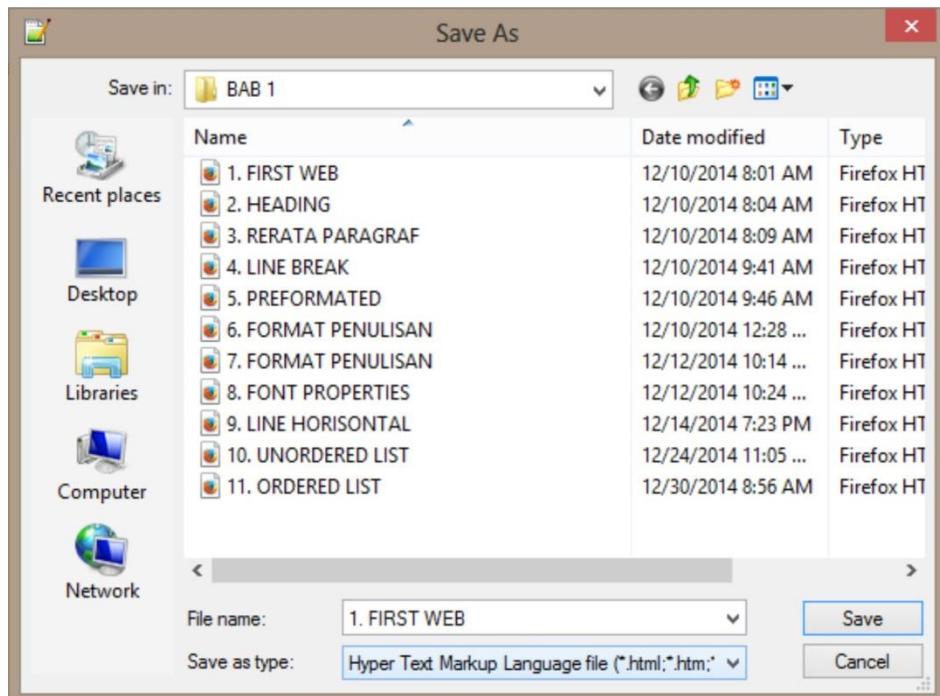
1. Buka notepad++ sebagai editor dokumen HTML.
2. Buat skrip sederhana seperti di bawah ini di notepad++ :

```
1  <html>
2   <head>
3     <title> web pertama mantap </title>
4   </head>
5   <body>
6
7     ini adalah web pertama saya dan sukses bisa!!!
8
9   </body>
10 </html>
```

Gambar 1.2 Format Penulisan HTML

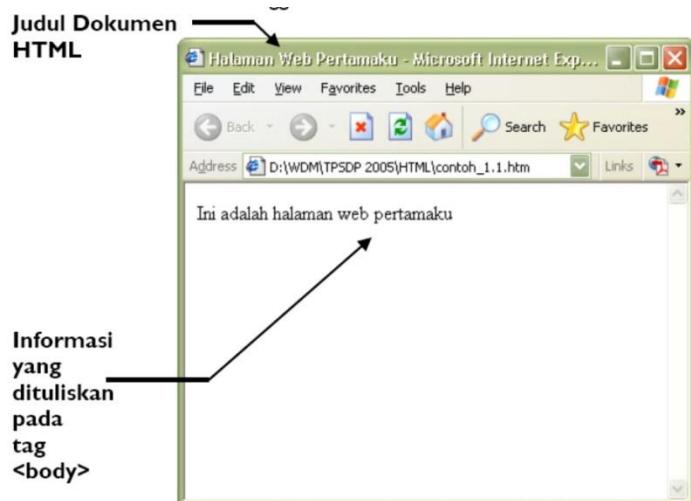
3. Simpan dokumen yang telah dibuat dengan menggunakan ekstensi .htm atau .html

Path directory :/nama_folder/xxx.html



Gambar 1.3 Format Penyimpanan HTML

4. Untuk melihat hasil dari dokumen HTML yang telah dibuat, buka dokumen tersebut menggunakan web browser. Dengan klick Run – Launch in IE.



Gambar 1. 4. Tampilan Hasil Eksekusi Contoh

1.3. Tag Dasar dalam HTML

1.3.1 Heading

Heading merupakan sekumpulan kata yang menjadi judul atau sub judul dalam dokumen HTML, dengan berbagai ukuran yang berbeda.

Pada dasarnya untuk menyatakan suatu Heading, digunakan tag `<Hx>` dimana `x` adalah level 1 sampai 6. Tag Heading atau `<Hx>` adalah tag berpasangan, yaitu tag yang mempunyai tag awal `<Hx>` dan tag penutup `</Hx>`

Format : `<H1>.....</H1> sampai <H6>.....</H6>`

Contoh :

```
<HTML>
<BODY>
<H1>Heading 1 adalah 24 Point</H1>
<br>
<H2>Heading 2 adalah 18 Point</H2>
<br>
<H3>Heading 3 adalah 14 Point</H3>
<br>
<H4>Heading 4 adalah 12 Point</H4>
<br>
<H5>Heading 5 adalah 10 Point</H5>
<br>
<H6>Heading 6 adalah 7 Point</H6>
<br>
```

1.3.2 Paragraf

a. Melakukan Formating Paragraf

Jika kita akan membuat sebuah website, dapat dipastikan kita juga kan menyajikan informasi-informasi yang diperlukan. HTML juga mendefinisikan elemen khusus, untuk mendefinisikan teks dengan makna khusus.

HTML menggunakan unsur-unsur seperti **** dan **<i>** untuk memformat output, seperti teks tebal atau miring.

Elemen format yang dirancang untuk menampilkan jenis khusus dari teks:

- teks tebal : menggunakan tag ****
- teks penting : menggunakan tag ****
- teks miring : menggunakan tag **<i>**
- teks menekankan : menggunakan tag ****
- teks menandai : menggunakan tag **<mark>**
- teks kecil : menggunakan tag **<small>**
- teks menghapus : menggunakan tag ****
- teks memasukkan : menggunakan tag **<ins>**
- subscript / pangkat dibawah : menggunakan tag **<sub>**
- superscripts / pangkat di atas : menggunakan tag **<sup>**

b. Menggunakan Line Break

Jika kita biasa menggunakan tombol Enter untuk ganti baris, maka hal itu tidak berlaku dalam penulisan dokumen HTML. Seberapa banyak kita memencet tombol Enter, maka tidak kan dianggap 1 pun.

Dalam HTML tag Line Break dapat digunakan untuk membuat baris baru layaknya fungsi tombol Enter pada *keyboard*. Line break menggunakan tag **
**

c. Preformatted Text

Jarak antar kata suatu teks atau kalimat di dalam dokumen web adalah sebesar satu spasi. Agar browser web menampilkan sesuai dengan dokumen web tersebut di dalam editor teks, maka teks yang bersangkutan harus di beri tanda tag **<pre>** di awalnya dan **</pre>** di akhir teks. Dengan tag **<pre>**, yang mempunyai kepanjangan "**preformatted**" teks akan di tampilkan dalam browser dengan

ukuran font dengan lebar - fix. Tag ini juga akan menjaga spasi, baris baru, dan tap sesuai dengan aslinya pada saat di tampilkan dalam browser.

e. Mengolah Properti Dari Font

Saat proses koding, ada kalanya kita harus melakukan pewarnaan, atau bahkan menentukan jenis huruf dari font yang kita pakai dengan alasan tertentu, mungkin karena info penting agar lebih bisa diperhatikan dll. Kita dapat menggunakan tag `style="property:value"` untuk mengubah style property dari font. Gunakan atribut style untuk elemen styling HTML

- **background-color** untuk warna latar belakang
- **color** untuk warna teks
- **font-family** untuk font teks
- **font-size** untuk ukuran teks
- **text-align** untuk perataan teks

f. Membuat garis Horizontal

Garis horizontal (horizontal line) digunakan untuk memisahkan halaman web secara visual. Tag yang digunakan adalah `<hr>`. Beberapa atributnya adalah *ALIGN*, *SIZE*, dan *WIDTH*. Tag hr seperti halnya tag br (`
`) adalah tag yang berdiri sendiri. Jadi tidak perlu ditutup dengan tag penutup.

1.3.3 List

List merupakan bentuk umum yang biasa kita gunakan untuk menampilkan daftar dengan urutan tertentu. Dalam HTML, dikenal 3 (tiga) istilah list : ordered list, unordered list dan definition list.

a. List Tanpa Urutan (Unordered Lists)

Unordered list adalah metode mengurutkan daftar dengan menggunakan simbol atau special character pada HTML. nama lain dari unordered list sendiri adalah bullet list. Bullet atau onordered list ini umumnya digunakan apabila kita ingin membuat suatu daftar yang terlepas dari suatu urutan-urutan tertentu. Untuk list jenis ini, pembuatannya sangat mirip dengan Ordered list, hanya saja ada perbedaan pada jenis synthax yang digunakan dimana unordered list akan menggunakan ``.

b. List Berurut (Ordered Lists)

Ordered list adalah suatu metode mengurutkan daftar dengan menggunakan angka, abjad dan angka romawi pada HTML. Dalam ordered list ini akan ada dua jenis elemen yang dibutuhkan yaitu **** yang merupakan singkatan ordered list dan **** yang merupakan singkatan list item

c. Definition List

Definition list merupakan jenis daftar yang umumnya digunakan untuk mendefinisikan sesuatu. *Definition list* sendiri sebenarnya sudah terlepas dari ketiga daftar diatas, dalam artian mempunyai jenis opening dan closing tag yang benar-benar berbeda dari tiga jenis daftar diatas. Definition list akan menggunakan tiga jenis tag : Diawali dengan **<dl>** yang merupakan singkatan definition list; **<dt>** yang merupakan singkatan definition term (digunakan untuk kata yang akan didefinisikan); **<dd>** definition description yang merupakan penjelasan dari suatu istilah (**<dt>**).

LATIHAN

1. Menggunakan Heading.

Simpan dengan nama **latihan1.html**

```
2. HEADING.html
1 <HTML>
2 <HEAD>
3   <TITLE>
4     MENGGUNAKAN HEADING
5   </TITLE>
6 <BODY>
7   DOKUMEN HTML INI MENGGUNAKAN HEADING
8     <H1> HEADING 1</H1>
9     <H2 ALIGN ="RIGHT"> HEADING 2 </H2>
10    <H3 ALIGN ="CENTER"> HEADING 3 </H3>
11    <H4> HEADING 4 </H4>
12    <H5> HEADING 5 </H5>
13    <H6> HEADING 6 </H6>
14    <h3 ALIGN ="LEFT"> HEADING 8 </H3>
15  </BODY>
16 </HEAD>
17 </HTML>
```

Gambar 1.5. Menggunakan Heading

2. Memformat Paragraf.

Simpan dengan nama **Latihan2.html**

```
<html>
<head>
  <title> Melakukan Formatting Paragraf </title>
<body>
  <p> Paragraf pertama </p>
  <p> Paragraf pertama ini menggunakan rata kiri </p>
  <p></p>
  <p align="center"> paragraf kedua </p>
  <p align="center"> paragraf kedua ini menggunakan rata tengah </p>
  <p></p>
  <p align="right"> paragraf ketiga </p>
  <p align="right"> paragraf ketiga ini menggunakan rata kanan </p>
</body>
</html>
```

Gambar 1.6. Skrip contoh format paragraph

3. Penggunaan Line Break dalam HTML.

Simpan dengan nama **Latihan3.html**

```
1  <html>
2    <head>
3      <title> Penggunaan Line Break </title>
4    </head>
5    <body>
6      Menggunakan Line Break <br><br>
7      Dengan menggunakan line break, kita dapat memindahkan kalimat<br>
8      setelah elemen tersebut satu baris dibawahnya.<br>
9      seperti menggunakan ENTER pada word processing.
10   </body>
11 </html>
```

Gambar 1.7. Contoh penggunaan Line Break

4. Penggunaan Pre dalam HTML.

Simpan dengan nama **Latihan4.html**

```
1  <html>
2    <head>
3      <title> Penggunaan Preformarted </title>
4    </head>
5    <body>
6      <pre>
7        Pengetikan ini dicoba menggunakan Preformarted Text
8
9        Dimana format penulisan ditampilkan dalam web sesua dengan format yang dilakukan
10       Seperti contoh melakukan satu tab seperti ini.
11     </pre>
12   </body>
13 </html>
```

Gambar 1.8. Contoh skrip penggunaan preformatted

5. Memformat Bentuk Tulisan dalam HTML (penebalan, penulisan miring dan garis bawah pada font).

Simpan dengan nama **Latihan5.html**

```
1  □<html>
2  □  <head>
3  |    <title> Menggunakan Teknik Format Penulisan </title>
4  □  </head>
5  □  <body>
6  |    <!-- membuat tulisan menjadi Tebal -->
7  |    <b> Tulisan ini dicetak TEBAL </b>
8  |    <p>
9  |        <!-- membuat tulisan menjadi Miring -->
10 |        <i> Tulisan ini dicetak MIRING </i>
11 |    <p>
12 |        <!-- membuat tulisan menjadi bergaris bawah -->
13 |        <u> Tulisan ini dicetak BERGARIS BAWAH</u>
14 |    </body>
15 </html>
```

Gambar 1.9. Contoh skrip format penulisan

6. Penggunaan Superscript dan Subscript dalam HTML.

Simpan dengan nama **Latihan6.html**

```
1  □<html>
2  □  <head>
3  |    <title> Melakukan Format Penulisan </title>
4  □  </head>
5  □  <body>
6  |    <!-- Membuat Tulisan menjadi TEBAL -->
7  |    <b> Tulisan ini telah dicetak TEBAL </b>
8  |    <p>
9  |        <!-- Membuat Tulisan tertulis MIRING -->
10 |        <i> Tulisan ini telah dicetak MIRING </i>
11 |    <p>
12 |        <!-- Membuat Tulisan bergaris bawah -->
13 |        <u> Tulisan ini dicetak BERGARIS BAWAH </u>
14 |    <p>
15 |        <!-- Membuat Format Superscript -->
16 |        Dalam matematika, x pangkat 2 ditulisa dengan X <sup>2</sup>
17 |    <p>
18 |        <!-- Membuat Format Subscript -->
19 |        Rumus kimia untuk Oksigen adalah O <sub>2</sub>
20 |    </body>
21 </html>
```

Gambar 1.10. Contoh skrip format penulisan

7. Mengolah property text.

Simpan dengan nama **Latihan7.html**

```
1 <html>
2   <head>
3     <title>
4       Pengolahan FONT dengan Properties
5     </title>
6   </head>
7   <body>
8     <font size = "1" face=tahoma color =#ff0000> tulisan ini bertulisan dengan ukuran 1, dengan font Tahoma dan berwarna HIJAU
9   </font>
10  <br>
11  <font size = "4" face=arial color=green > tulisan ini bertuliskan dengan ukuran 4, dengan font Arial dan berwarna HIJAU
12  </font>
13 </body>
14 </html>
```

Gambar 1.11. Contoh skrip property text

8. Menambahkan Garis Horisontal pada HTML.

Simpan dengan nama **Latihan8.html**

```
1 <html>
2   <head>
3     <title>
4       Menambahkan Horizontal Rules Pada Website
5     </title>
6   </head>
7   <body>
8     Menambahkan garis mendatar pada halaman html
9     <hr size= "15">
10    <hr width="100" align= "left" >
11    <hr color="blue">
12
13    <hr size = "20" width = "300" noshade>
14  </body>
15 </html>
```

Gambar 1.12. Contoh skrip untuk penambahan garis horizontal

9. Penggunaan Unordered List pada HTML

Simpan dengan nama **Latihan9.html**

```
1 <html>
2   <head>
3     <title> Penggunaan Listing </title>
4   </head>
5   <body>
6     <h3> Menggunakan Unordered List </h3>
7     <hr>
8     <h4> Ordered List Dengan Type Default </h4>
9     <ul>
10       <li> Agung Nugroho </li>
11       <li> Mulya Sulistyono </li>
12       <li> Bernadhed </li>
13     </ul>
14
15     <h4> Unordered List Dengan Type Circle </h4>
16     <ul type = "circle">
17       <li> Agung Nugroho </li>
18       <li> Mulya Sulistyono </li>
19       <li> Bernadhed </li>
20     </ul>
21
22     <h4> Unordered List Dengan Type Disc </h4>
23     <ul type ="disc">
24       <li> Agung Nugroho </li>
25       <li> Mulya Sulistyono </li>
26       <li> Bernadhed </li>
27     </ul>
28
29     <h4> Unordered List Dengan Type Square </h4>
30     <ul type ="square">
31       <li> Agung Nugroho </li>
32       <li> Mulya Sulistyono </li>
33       <li> Bernadhed </li>
34     </ul>
35   </body>
36 </html>
```

Gambar 1.13. Contoh skrip untuk penggunaan Unordered List

10. Penggunaan Ordered List pada HTML

Simpan dengan nama **Latihan10.html**

```
1 <html>
2   <head>
3     <title> Menggunakan Ordered List </title>
4   </head>
5   <body>
6     <h3> Menggunakan Ordered List </h3>
7     <hr>
8     <h4> Ordered List Menggunakan Type Default </h4>
9     <ol>
10       <li> Agung Nugroho </li>
11       <li> Bernadhed </li>
12       <li> Mulya Sulistyono </li>
13     </ol>
14
15     <h4> Ordered List Menggunakan Type a </h4>
16     <ol type = "a">
17       <li> Agung Nugroho </li>
18       <li> Bernadhed </li>
19       <li> Mulya Sulistyono </li>
20     </ol>
21
22     <h4> Ordered List Menggunakan Type I </h4>
23     <ol type = "I">
24       <li> Agung Nugroho </li>
25       <li> Bernadhed </li>
26       <li> Mulya Sulistyono </li>
27     </ol>
28
29     <h4> Ordered List Menggunakan Type Atribut Start </h4>
30     <ol start = "14">
31       <li> Agung Nugroho </li>
32       <li> Bernadhed </li>
33       <li> Mulya Sulistyono </li>
34     </ol>
35   </body>
36 </html>
```

Gambar 1.14. Contoh skrip untuk penggunaan Ordered List

Studi Kasus

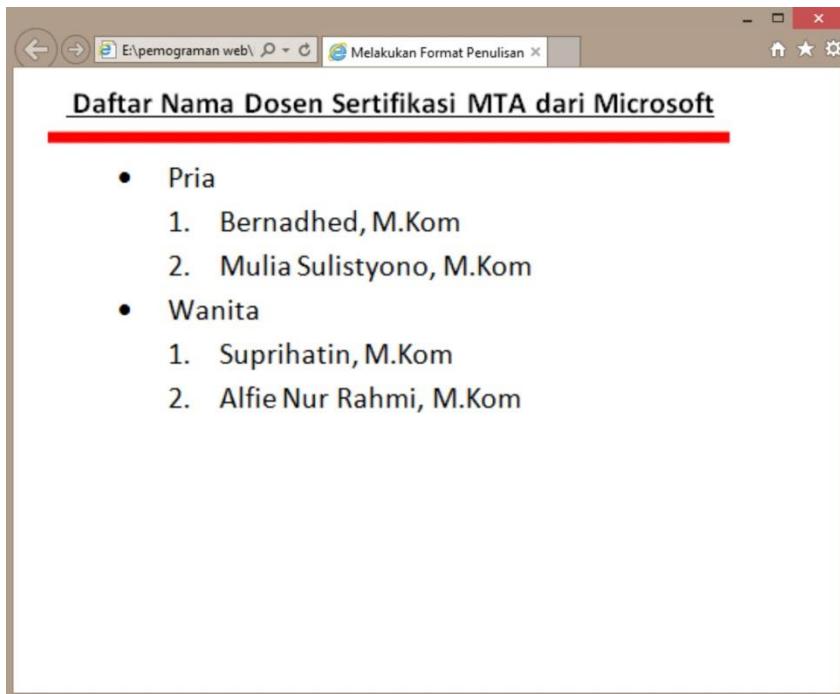
1. Buatlah Skrip HTML yang mampu menampilkan format tulisan seperti dibawah ini

H₂O

X² + 4X + 4

Linux adalah sistem operasi yang bersifat *open source*.

2. Buatlah Skrip HTML yang mampu menampilkan informasi seperti dibawah ini



Bab 2. Link dan Gambar

Tujuan

1. Mahasiswa mampu membuat link antar dokumen HTML
2. Mahasiswa dapat membuat link ke bagian tertentu dari dokumen yang sama
3. Mahasiswa mampu membuat dokumen HTML dan memasukkan gambar ke halaman HTML
4. Mahasiswa mampu membuat link dari gambar

Overview

Kemudahan dalam aplikasi berbasis web salah satunya adalah bisa menghubungkan satu dokumen dengan dokumen lainnya, baik dalam satu server aplikasi web maupun dengan server aplikasi web yang berbeda di seluruh dunia maya.

HTML menyediakan hypertext link yang merupakan daerah teks (ataupun gambar) yang bisa link (menyambungkan) ke dokumen HTML yang lain. Seiring berkembangnya teknologi internet, bukan hanya teks dan gambar yang bisa link ke dokumen tertentu, file multimedia pun sudah dapat dibuat link ke dokumen yang diinginkan.

2.1. Link

Kekuatan utama dokumen HTML terletak pada **hypertext link** atau **hyperlink** atau lebih singkat lagi disebut **link**. Dengan hyperlink ini kita bisa membuka dokumen HTML lain atau langsung menuju ke bagian tertentu sebuah dokumen HTML. Hyperlink ini dapat diletakkan pada teks tertentu ataupun pada sebuah image, bila diletakkan pada teks maka teks tersebut (secara default) akan digaris-bawahi dan warnanya menjadi berbeda. Sintaksis penulisan :

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

Beberapa atribut yang sering ditambahkan atau sering digunakan dalam penggunaan sintak diatas adalah seperti dibawah ini:

Tabel. 2.1. Atribut dalam Link

Atribut	Kegunaan
Href	Menunjukkan url yang dituju
Name	Memberikan nama pada bagian tertentu pada dokumen
Target	Menunjukkan target ditampilkannya link
Title	Menunjukkan title dari link ketika cursor digerakkan di sekitar area hyperlink

2.1.2 Membuat Link Ke Dalam Dokumen

Seringkali halaman web adalah halaman yang panjang dan selalu membukanya dari posisi awal akan sangat melelahkan bagi pengguna. Untuk itu akan lebih memudahkan bila kita langsung merujuk pada bagian tertentu sebuah halaman web. Caranya dengan menyisipkan Tag Anchor <a> dengan atribut "name" di bagian dokumen yang akan dituju.

Untuk perintah standartnya :

```
<a name="nama_section"> </a>
```

Untuk menggunakan link yang akan merujuk ke bagian dokumen itu, ditambahkan '#nama_section" menjadi :

```
<a href="sebuah_halaman.html#nama_section">langsung ke  
nama section</a>
```

2.2. Gambar

Untuk menyisipkan gambar kita dapat memanfaatkan tag . Format file gambar yang bisa ditampilkan bisa bermacam-macam, misalnya jp, jpeg, pcx, gif, psd, dan sebagainya. Pada umumnya, perancang web hanya menggunakan kombinasi dari tiga format file gambar saja yaitu jpeg, gif dan psd. Sintaksis yang digunakan secara standar adalah

```
<IMG src="url_file" width="img_width" height="img_height"  
vspace="10" hspace="10" alt="alt_teks">
```

2.2.1 Membuat obyek gambar menjadi sebuah hyperlink

Apabila kita mengakses sebuah website, seperti facebook atau search engine google terdapat logo dari dua website tersebut. Logo tersebut merupakan gambar dimana kita sering dapat melakukan klick terhadap gambar tersebut dan merujuk ke sebuah link tertentu. Berikut contoh sintaks penggunaan obyek gambar menjadi hyperlink :

```
<img src='url gambar' alt='nama alternatif gambar' width='tinggi' height='lebar'>
```

- Tag `` adalah tag HTML untuk mendefinisikan sebuah gambar.
- Atribut `src` adalah sumber/url dari gambar.
- Atribut `alt` adalah untuk memberikan nama alternatif gambar. Apabila gambar tidak bisa ditampilkan, maka yang muncul adalah nama alternatifnya.
- Atribut `height` dan `width` adalah untuk mengatur ukuran gambar.

contoh:

```
<img src='sepeda.jpg' alt='Click Me!' width='100' height='100'>
```

LATIHAN

1. Membuat Link Antar Dokumen

Dalam Contoh ini, dapat dilihat dari script di bawah ini, sebelumnya siapkan halaman html dengan nama **Magister Teknik Informatika.html**, **Sistem Informasi.html**, **Teknik Informatika.html** (disimpan di direktori anda masing-masing) dengan script masing-masing seperti dibawah ini:

```
Link Amikom .html | Magister Teknik Informatika.html | Sistem Informasi .html | Teknik Informatika .html
1 <html>
2   <head>
3     <title> Magister Teknik Informatika </title>
4   </head>
5   <body>
6     <h4> Magister Teknik Informatika </h4>
7   </body>
8 </html>
```



```
Link Amikom .html | Magister Teknik Informatika.html | Sistem Informasi .html | Teknik Informatika .html
1 <html>
2   <head>
3     <title> Sistem Informatika </title>
4   </head>
5   <body>
6     <h4> Sistem Informatika </h4>
7   </body>
8 </html>
```



```
Link Amikom .html | Magister Teknik Informatika.html | Sistem Informasi .html | Teknik Informatika .html
1 <html>
2   <head>
3     <title> Teknik Informatika </title>
4   </head>
5   <body>
6     <h4> Teknik Informatika </h4>
7   </body>
8 </html>
```

Gambar 2.1. Skrip untuk direksional link dari HTML utama

Setelah kita mengetikkan skrip diatas dan melakukan penyimpanan, selanjutnya kita akan membuat halaman HTML baru yang nanti ada link untuk merujuk ke tiap halaman HTML diatas. Simpan dengan nama **latihan11.html**.

```
1 <html>
2   <head>
3     <title> STMIK Amikom Yogyakarta </title>
4   </head>
5   <body>
6     <a href = "http://www.amikom.ac.id">
7       </a>
8
9     <center>
10      <h3> STMIK Amikom </h3>
11      <h5> Jln Ringroad Utara Depok Sleman Yogyakarta </h5>
12      <hr>
13      [ <a href="Teknik Informatika .html"> Teknik Informatika </a> ]
14      [ <a href="Sistem Informasi .html" target="blank"> Sistem Informatika </a> ]
15      [ <a href="E:/pemrograman web/PW/BAB 2/LINK DOKUMEN/Magister Teknik Informatika.html"> Magister Teknik Informatika </a> ]
16      <br>
17      <br>
18      [ <a href="http://www.bernadhtagger.com"> BAAK </a> ]
19      [ <a href="http://www.budayamenjamahgaya.com"> Dikti </a> ]
20    </center>
21  </body>
22</html>
```

Gambar 2.2. Skrip untuk Halaman utama

2. Membuat Link ke dalam Dokumen

Simpan dengan nama **latihan12.html**.

```
1 <html>
2   <head>
3     <title> Membuat Link Ke Bagian Dalam Dokumen </title>
4   </head>
5   <body>
6     <h4> Membuat Link Ke Bagian Dalam Dokumen </h4>
7     <a name="Bab1"> </a>
8     <b> Bab 1 </b>
9     <br>
10    Pada bab ini akan dijelaskan tutorial membuat dokumen HTML. <br>
11    Mulai dari pemahaman, pembuatan dokumen HTML sederhana sampai membuat Website. <br>
12    Sebagai pendahuluan , coba Klik link ini untuk
13    <a href="#sejarah"> menuju ke bagian lain </a> pada dokumen ini.
14    <p>&nbsp;</p>
15    <p>&nbsp;</p>
16    <p>&nbsp;</p>
17    <p>&nbsp;</p>
18    <p>&nbsp;</p>
19    <p>&nbsp;</p>
20    <p>&nbsp;</p>
21    <p>&nbsp;</p>
22    <p>&nbsp;</p>
23    <p>&nbsp;</p>
24    <p>&nbsp;</p>
25    <p>&nbsp;</p>
26    <p>&nbsp;</p>
27    <p>&nbsp;</p>
28    <p>&nbsp;</p>
29    <p>&nbsp;</p>
30    <p>&nbsp;</p>
31    <a name="sejarah"> </a>
32    <b> 1.1 Sekilas Sejarah Internet </b> <br>
33    Sub Bab ini merupakan bagian dari bagian bab yang dituju ketika link pada Bab1 di klik. <br>
34    Untuk kembali ke Bab1, klik <a href="#Bab1"> disini </a>
35
36</body>
</html>
```

Gambar 2.4. Skrip untuk merujuk dalam dokumen

3. Menyisipkan gambar pada HTML.

Simpan dengan nama **latihan13.html**.

```
1 <html>
2   <head>
3     <title> Menambahkan Image </title>
4   </head>
5   <body>
6     <h4> Menambahkan Image Ke Dalam Dokumen HTML </h4>
7     
8     <br>
9     <h4> Mengatur Border Pada Image </h4>
10    
11    <br>
12    <h4> Alternating Text </h4>
13    Pada Image di bawah ini terdapat Alternating Text.
14    Silahkan gerakan mouse melewati Image <br>
15    
16  </body>
17 </html>
```

Gambar 2.5. Skrip untuk menyisipkan gambar didalam HTML

4. Membuat gambar menjadi hyperlink.

```
1 <html>
2   <head>
3     <title> Halaman Awal </title>
4   </head>
5   <body>
6     <h4> Selamat Datang di Personal Web Saya. </h4>
7     Untuk Melanjutkan silahkan klik gambar dibawah ini
8     <br>
9     <a href="4. a Gambar Sebagai Link Sub .html">
10    
11    </a>
12    <br>
13    <br>
14    <br>
15    <br>
16    Selamat Menjelajah
17  </body>
18 </html>
```

Gambar 2.7. Skrip untuk menyisipkan sebuah link dalam gambar

Jika diperhatikan, skrip diatas terdapat sebuah sintak yang mengarah sebuah halaman html tertentu. Oleh karenanya, buatlah sesuai nama halaman html diatas agar bisa dirujuk oleh halaman tersebut. Simpan dengan nama **gambar sebagai link sub.html**.

```
1 <html>
2   <head>
3     <title> Bios </title>
4   </head>
5   <body>
6     <a href= "http://www.bernadtagger.com">
7       
8     </a>
9     Saya adalah seorang tenaga pendidik di sebuah Kampus Swasta Jogja
10   </body>
11 </html>
12
```

Gambar 2.8. Skrip untuk menyisipkan sebuah link tautan

Jika pengetikan skriptidak terjadi kesalahan, maka obyek gambar diatas jika diklik maka akan merujuk ke halaman html yang telah kita buat setelahnya.

Bab 3. Form

Tujuan

1. Mahasiswa mampu membuat form pada dokumen HTML
2. Mahasiswa memahami cara penggunaan dari fasilitas Form pada halaman dokumen HTML
3. Mahasiswa mampu menginputkan field sesuai dengan kegunaan

Overview

Salah satu fasilitas yang disediakan oleh dokumen HTML memungkinkan kita untuk melakukan “interaksi lebih” terhadap halaman dokumen tersebut. Semisal, kemampuan dokumen HTML mampu menyediakan fasilitas yang dapat menerima masukkan atau isian data dari user. Data isian user ini nantinya akan dapat diproses lebih lanjut menjadi informasi yang dibutuhkan oleh user maupun oleh pemilik situs tersebut. Teknik pengisian ini dilakukan menggunakan elemen Form.

Form dalam HTML adalah suatu bagian yang berfungsi sebagai input atau masukan dari pengguna yang kemudian akan diproses atau diolah untuk dapat digunakan sesuai dengan kebutuhan. Contohnya proses pengiriman data, browse, hapus, penyuntingan data dan lain sebagainya.

3.1 Menu Form

Form dalam HTML adalah suatu bagian yang berfungsi sebagai input atau masukan dari pengguna yang kemudian akan diproses atau diolah untuk dapat digunakan sesuai dengan kebutuhan. Contohnya proses pengiriman data, browse, hapus, penyuntingan data dan lain sebagainya. Cara penulisan form dalam html adalah.

```
<form action="_url_" method="get|post" enctype="">  
    ... elemen - elemen yang ditambahkan dalam form ...  
</form>
```

```
<form name="nama_form" method="post/get" action="url">...isi form...</form>
```

Pada penulisan form "nama form" adalah nama form yang sedang digunakan, method adalah metode atau cara yang digunakan untuk menyimpan data ke server sedangkan action adalah alamat atau bisa juga file lain yang digunakan untuk memproses input dari form kedalam server.

3.2 Jenis-jenis media input dalam Form

Pada bagian form tersedia kontrol-kontrol input yang digunakan untuk masing-masing keperluan dalam memasukan data input. Dan kontrol form terdiri dari :

3.2.1 Kontrol Berupa Text

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 isi textarea

Gambar 3.1 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:



Gambar 3.2 Contoh Penggunaan Elemen Input

Perhatikan juga bahwa kita menggunakan atribut `type` pada elemen ini untuk menentukan data-data yang akan diisi. 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 diisi pada atribut `type` yaitu:

- | | |
|-------------------|--------------|
| 1. color | 9. time |
| 2. datetime-local | 10. datetime |
| 3. number | 11. month |
| 4. tel | 12. search |
| 5. week | 13. url |
| 6. date | 14. password |
| 7. email | 15. text |
| 8. range | 16. file |

3.2.2 Kontrol Berupa Radio Button

Radio button digunakan untuk menyatakan pilihan yang bersifat tunggal, umumnya pada web digunakan untuk kelengkapan pertanyaan seperti pada pendaftaran online, biasanya pertanyaan itu adalah pilihan untuk menyatakan jenis kelamin, agama, ataupun sebagainya. Radio Button ini akan sering kita temui saat kita berselancar di dunia maya yang terkait dengan proses input data seperti pendaftaran dll.

Radio button dibuat dalam HTML dengan menggunakan elemen `input`, dengan atribut `type` bernilai : code`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.



Gambar 3.3 Contoh Penggunaan Elemen Radio Button

3.2.3 Kontrol Berupa Checkbox

Berbeda dengan radio button yang hanya memungkinkan user memilih satu pilihan, pada input type checked box Anda dapat memilih satu atau beberapa pilihan, atau tidak memilih sama sekali. Pilihan ini biasanya dipakai untuk memasukkan data yang sifatnya opsional. Anda dapat menggunakan “Checkbox” pada atribut. Input checkbox adalah sebuah masukan untuk memilih sesuatu opsi/ pilihan. Jika dipilih maka pengguna akan men-*check* (mencentang) dan jika tidak memilih akan dibiarkan saja.

Elemen ini sama seperti radio button, dibuat dengan elemen input yang nilai atribut type-nya berisikan checkbox.

```
<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.



Gambar 3.4 Contoh Penggunaan Elemen Checkbox

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**.

3.2.4 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:



Gambar 3.5 Contoh Penggunaan Elemen Dropdown

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>
```



Gambar 3.6 Contoh Penggunaan Elemen Dropdown dengan Pilihan Banyak

3.2.5 Button

Pada saat kita melakukan browsing di dunia maya, dan mengakses sebuah website. Sering kita menemui tombol-tombol fungsi seperti reset, submit, send dan lain sebagainya. Dan sering pula kita sering menemui gambar yang dibuat fungsi seperti tombol.

3.2.5.1 Submit

Tombol ini berfungsi untuk memanggil url yang sudah didefinisikan di atribut *action*. Pembuatan tombol submit juga dilakukan dengan menggunakan elemen `input`, yang atribut `type`-nya diisi dengan nilai `submit`, seperti berikut:

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

Perhatikan 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.

Masukkan Nilai

Gambar 3.7 Contoh Penggunaan Tombol Submit

3.2.5.2 Reset

Tombol ini berfungsi untuk mengembalikan form ke kondisi awal (mengosongkan nilai semua elemen yang ada pada form). Pembuatan tombol dilakukan sama persis seperti pada tombol submit, dengan perbedaan nilai pada atribut `type`, yang diisi dengan `reset` pada tombol penghapusan ini.

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

3.2.6 Organisasi Elemen Form

Mengetahui bagaimana membuat elemen-elemen masukan / input 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 diisi 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.

3.2.6.1 Label

Elemen label memberikan fasilitas untuk menambahkan teks pada elemen masukan form. Elemen ini diisi dengan deskripsi dari elemen masukan yang ingin ditambahkan, dan harus memiliki atribut for. Atribut for pada elemen label 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 di mana isi dari atribut for pada label adalah sama dengan isi atribut id pada elemen input.

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

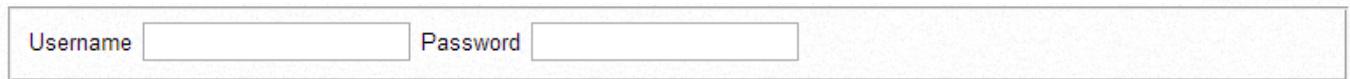
3.2.6.2 Fieldset

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:



gambar 3.8 Contoh Penggunaan Fieldset

3.2.6.2 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:



Gambar 3.9 Contoh Penggunaan Legend

dan tentunya tampilan dari legend dapat diubah dengan menggunakan CSS.

3.2.7 Validasi Masukan pada Form

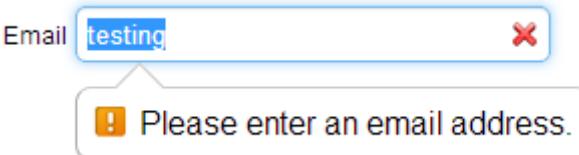
Seringkali dalam pengembangan form pada aplikasi, terdapat elemen-elemen yang tidak wajib diisi oleh pengguna dan elemen-elemen yang wajib diisi oleh pengguna. Aplikasi kemudian akan melakukan pengecekan terhadap nilai masukan dari pengguna, apakah nilai yang wajib ada sudah diisi atau belum. Jika nilai 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:

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

2. 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.



Gambar 3.10 Contoh Pesan Kesalahan Pengisian Email pada HTML

LATIHAN

1. Penggunaan input text dalam sebuah form.

Simpan dengan nama **latihan14.html**

```
1 <html>
2   <head>
3     <title> Menyisipkan Elemen Berupa Text </title>
4   </head>
5   <body>
6     <font type=arial black font size="6" > Data Diri </font>
7     <br> <br>
8     <font ty=arial size = "5" >
9       nama : <input type = "text" name = "nama">
10      <br>
11      <br>
12      nim : <input type = "text" name = "nim" value="" maxlength="9" size="9">
13      </font>
14    </body>
15 </html>
```

Gambar 3.11. Contoh skrip untuk penggunaan input text dalam sebuah form

2. Penggunaan Input Text berupa Password.

Simpan dengan nama **latihan15.html**

```
1 <html>
2   <head>
3     <title> Input Password </title>
4   </head>
5   <body>
6
7     Password : <input type ="password" name="nama" size="11" maxlength="9">
8
9   </body>
10 </html>
```

Gambar 3.12. Contoh skrip untuk penggunaan input text dalam sebuah password menu

3. Penggunaan Input text berupa radio button

Simpan dengan nama **latihan16.html**

```
1  <html>
2   <head>
3     <title> Penggunaan Radio Button </title>
4   </head>
5   <body>
6     Jenis Kelamin :
7     <br>
8     <input type="radio" name="jenis_kelamin" value="L"> Laki-laki
9     <br>
10    <input type="radio" name="jenis_kelamin" value="P"> Perempuan
11    <br>
12
13  </body>
14 </html>
```

Gambar 3.13. Contoh skrip untuk penggunaan input text berupa radio button

4. Penggunaan checkbox dalam HTML.

Simpan dengan nama **latihan17.html**

```
1  <html>
2   <head>
3     <title> Penggunaan Check Box </title>
4   </head>
5   <body>
6     Sebutkan Hobi Anda, boleh lebih dari satu :
7     <br>
8     <input type="checkbox" name="hobi_1" value="Menonton"> Menonton
9     <br>
10    <input type="checkbox" name="hobi_2" value="Game"> Bermain Game
11    <br>
12    <input type="checkbox" name="hobi_3" value="Shopping"> Belanja
13    <br>
14    <input type="checkbox" name="hobi_4" value="Mancing"> Memancing
15    <br>
16    <input type="checkbox" name="hobi_5" value="Olahraga"> Olahraga
17    <br>
18  </body>
19 </html>
```

Gambar 3.14. contoh skrip untuk penggunaan input text berupa checkbox

5. Penggunaan submit, reset, dan image sebagai button.

Simpan dengan nama **latihan18.html**

```
1 <html>
2   <head>
3     <title> Konfigurasi Button </title>
4   </head>
5   <body>
6     <form>
7
8       <input type="submit" value="Kirim" name="Submit">
9       <input type="reset" value="Ulangi" name="Reset">
10      <input type="button" value="Batal" name="Cancel">
11      <input type="image" name="Gambar" img src="research.png" width="194" height="90">
12
13    </form>
14  </body>
15 </html>
```

Gambar 3.15. Contoh skrip penggunaan input berupa button

6. Penggunaan Text area dalam Form.

Simpan dengan nama **latihan19.html**

```
1 <html>
2   <head>
3     <title> Membuat Text Area </title>
4   </head>
5   <body>
6     Deskripsi Diri:
7     <br>
8     <textarea name="deskripsi" cols="45" rows="5" align="center"> Tuliskan deskripsi singkat tentang anda.
9     </textarea>
10
11 </body>
</html>
```

Gambar 3.16. Contoh skrip penggunaan input berupa text area

7. Penggunaan Select Dropdown dalam Form.

Simpan dengan nama **latihan20.html**

```
1  <html>
2   <head>
3     <title> Menggunakan Menu Dropdown </title>
4   </head>
5   <body>
6     
7     <br>
8     Silahkan pilih orientasi minat :
9     <br>
10    <select name="Orientasi Minat">
11      <option value="dg"> Design Grafis </option>
12      <option value="pj"> Programmer Java </option>
13      <option value="sa"> Sistem Analis </option>
14      <option value="ds"> Dagelan Saja </option>
15    </select>
16    <br>
17    <br>
18    <select name="Orientasi Kemampuan" Multiple>
19      <option value="dg"> Design Grafis </option>
20      <option value="pj"> Programmer Java </option>
21      <option value="sa"> Sistem Analis </option>
22      <option value="aj"> Ahli Jaringan </option>
23      <option value="it"> Trainer IT </option>
24      <option value="ds"> Dagelan Saja </option>
25    </select>
26    </body>
27  </html>
```

Gambar 3.17. Contoh skrip penggunaan Select Dropdown

Bab 4. Layout Web Page

Tujuan

1. Mahasiswa mampu membuat dan menformat tabel pada halaman dokumen HTML
2. Mahasiswa dapat memahami dalam pembuatan frame pada dokumen HTML
3. Mahasiswa mampu memahami penggunaan frame tersebut
4. Mahasiswa mampu memahami pembagian tampilan layar ke dalam beberapa pembagian, yang setiap bagian dihubungkan dengan dokumen HTML yang terpisah
5. Mahasiswa mampu mengatur properti frame border
6. Mahasiswa mampu menentukan target frame cells dari hypertext links

Overview

Saat kita membuat sebuah halaman website, dapat dibuat menggunakan tabel dan frame. Hal ini meang tidak ada perbedaan didalamnya yang begitu signifikan, tapi saat pada proses pengaksesan halaman web mungkin akan sangat berpengaruh terhadap respons time yang diperlukan.

Tabel merupakan cara untuk menampilkan informasi dalam halaman web dengan bentuk kolom dan baris. Hampir semua web yang mempunyai kualitas dan profesional, dirancang dengan menggunakan tabel. Layaknya sebuah spreadsheet yang memiliki sel dan berisi angka-angka, tabel dalam web juga mempunyai sel yang berisi link, gambar dan text.

Frame akan dapat membagi sebuah HTML menjadi beberapa dokumen HTML. Dimana masing-masing bagian terhubung dengan dokumen HTML yang terpisah.

4.1 Tabel

4.1.1 Membuat Sebuah Tabel dan Judul Tabel

Tabel digunakan untuk menyajikan data dalam bentuk kolom dan baris. Umumnya setiap kolom menunjukkan data yang sejenis, dan setiap baris yang terdiri atas kolom-kolom menunjukkan kelompok data dalam satu kesatuan.

Sebuah tabel mempunyai judul, tempat anda menjelaskan kolom/baris yang dilibatkan, baris untuk informasi dan sel untuk setiap itemnya. Pada tabel berikut, kolom pertama berisi informasi header, setiap baris menjelaskan sebuah tag tabel HTML dan setiap sel berisi sebuah pasangan tag atau penjelasan dari fungsi tag. Berikut ini disajikan elemen-elemen yang sering digunakan dalam pembuatan tabel dalam HTML.

Tabel 4.1 Elemen-elemen dalam pembuatan tabel HTML

Elemen	Penjelasan
<TABLE>...</TABLE>	Mendefenisikan sebuah tabel dalam HTML jika atribut border dituliskan, maka browser akan menampilkan tabel dengan border.
<CAPTION>...</CAPTION>	Mendefinisikan tulisan untuk judul tabel. Posisi default dari judul adalah ditengah pada bagian paling atas tabel. Atribut align="bottom" dapat digunakan untuk menempatkan judul pada bagian bawah tabel. Catatan: judul dapat diberi tag apa saja
<TR>...</TR>	Menspesifikasi sebuah baris tabel dalam tabel. Anda dapat mendefinisikan atribut untuk seluruh baris: align(left, center, right) dan/atau valign (top, middle, bottom).
<TH>...</TH>	Mendefinisikan sel header tabel. Secara default teks dalam sel ini ditebalkan dan ditampilkan di tengah.
<TD>...</TD>	Mendefinisikan sebuah sel data tabel. Secara default teks dalam sel ini ditampilkan rata kiri, dan ditengah secara vertikal. Sel data table dapat berisi atribut untuk mendefinisikan karakteristik dari sel dan isinya.

4.1.2 Melakukan format Baris dan Kolom

Setiap tabel pasti akan terdiri dari atas baris-baris yang diwakili dengan sebuah tag <tr> dan juga mempunyai kolom-kolom yang menggunakan sebuah tag <td>. Setiap baris dan kolom pada tabel memiliki atribut nya masing-masing.

4.1.3 Pembahasan Tabel Lebih Lanjut

Pada pembahasan selanjutnya, kita akan belajar menyisipkan sebuah gambar kedalam sebuah susunan tabel. Tabel di dalam sebuah website dapat berisi hyperlink, gambar, gambar yang menunjuk ke sebuah hyperlink dan teks dengan huruf yang berwarna.

4.2 Frame

Framing dalam HTML dapat kita pergunakan untuk membuat tampilan halaman HTML yang terbagi-bagi menjadi beberapa dokumen HTML, dimana didalam setiap bagian halaman HTML yang salah satu atau beberapa bagian berganti-ganti isinya sedangan bagian lain tetap sehingga dapat melakukan efisiensi terhadap bandwidth koneksi internet dan mempercepat proses download page secara keseluruhan.

Berikut adalah sintaks umum yang sering digunakan dalam membuat frame

```
<HTML>
<HEAD>
</HEAD>
<FRAMESET BORDER="#" { [ROWS | COLS] } = { #,[#, [...] ] }>
    <FRAME SRC="url" NAME="FrameName">
</FRAMESET>
</HTML>
```

Tabel 4.1 Atribut yang digunakan dalam frame

Atribut	Fungsi
Frameset cols	Membuat frame vertikal dengan lebar kolom tertentu
Frameset rows	Membuat frame horisontal dengan tinggi baris tertentu
Frame src	Memasukkan dokumen HTML ke dalam frame
Noframe	Memasukkan body text untuk browser yang tidak menampilkan frame

LATIHAN

1. Penggunaan table dalam HTML.

Simpan dengan nama **latihan21.html**

```
1  <html>
2   <head>
3     <title> Pembuatan Tabel </title>
4   </head>
5   <body>
6     <font face="tahoma" size="2" color="blue">
7       <table border="0.8">
8         <caption align="top"> <b> <u> Daftar Alamat </u> </b> </caption>
9         <tr bgcolor=blue align="center">
10          <td width="40"> No. </td>
11          <td width="150"> Nama </td>
12          <td width="200"> Alamat </td>
13        </tr>
14        <tr bgcolor=skyblue>
15          <td align="center"> 1. </td>
16          <td> Muiva Sulistyono </td>
17          <td> Sleman </td>
18        </tr>
19        <tr bgcolor=orange>
20          <td align="center"> 2. </td>
21          <td> Hendra Kurniawan </td>
22          <td> Solo </td>
23        </tr>
24        <tr bgcolor=skyblue>
25          <td align="center"> 3. </td>
26          <td> Agung Nugroho </td>
27          <td> Borobudur </td>
28        </tr>
29        <tr bgcolor=orange>
30          <td align="center"> 4. </td>
31          <td> Agus Fatkhurahman </td>
32          <td> Magelang </td>
33        </tr>
34      </table>
35    </body>
36  </html>
```

Gambar 4.1. Skrip Penggunaan table dalam HTML

2. Penggunaan formating baris dan kolom dalam skrip HTML.

Simpan dengan nama **latihan22.html**

```
1 <html>
2   <head>
3     <title> Mengatur Baris dan Kolom </title>
4   </head>
5   <body>
6     <table border="3" cellspacing="3" cellpadding="5">
7       <caption> Daftar Order Barang </caption>
8       <thead>
9         <th> No </th>
10        <th> Barang </th>
11        <th> Jenis </th>
12        <th> Harga </th>
13        <th> Jumlah </th>
14        <th> Total Harga </th>
15      </thead>
16      <tbody>
17        <tr>
18          <td align="center" rowspan="2">1</td>
19          <td> Pocari Sweat </td>
20          <td> Minuman </td>
21          <td> Rp. 4.500 </td>
22          <td align="right" rowspan="2">5</td>
23          <td> Rp. 22.500 </td>
24        </tr>
25        <tr>
26          <td align="center" rowspan="2">2</td>
27          <td> Nutrisari Jeruk </td>
28          <td> Rp. 4.500 </td>
29          <td align="right" rowspan="2">3</td>
30          <td> Rp. 7.350 </td>
31        </tr>
32        <tr>
33          <td align="center" rowspan="3">3</td>
34          <td> Pilus Garuda </td>
35          <td> Cemilan </td>
36          <td> Rp. 9.800 </td>
37        </tr>
38        <tr>
39          <td align="center" rowspan="3">4</td>
40          <td> Tanggo Coklat </td>
41          <td> Rp. 6.450 </td>
42          <td align="right" rowspan="3">2</td>
43          <td> Rp. 6.450 </td>
44        </tr>
45        <tr>
46          <td align="center" rowspan="3">5</td>
47          <td> Makanan Ringan </td>
48          <td> Rp. 6.450 </td>
49          <td align="right" rowspan="3">2</td>
50          <td> Rp. 12.900 </td>
51        </tr>
52      </tbody>
53    </table>
54  </body>
55</html>
```

Gambar 4.2. Skrip contoh untuk melakukan formating baris dan kolom dalam table

3. Penyisipan gambar dan text dalam Tabel. Simpan dengan nama **latihan23.html**

```
1 <html>
2   <head>
3     <title> Sisipan Gambar Dalam Tabel </title>
4   </head>
5   <body>
6     <table border="2">
7       <caption> Dokument Pemotretan </caption>
8       <thead>
9         <tr>
10          <th> 1. </th>
11          <th> 2. </th>
12          <th> 3. </th>
13        </tr>
14        <tr align="center">
15          <td> Bike-Mount </td>
16          <td> Bike-Modelling </td>
17          <td> Bike-Consent </td>
18        </tr>
19        <tr align="center">
20          <td>  </td>
21          <td>  </td>
22          <td> <a href="2. Format Baris dan Kolom.html">  </a> </td>
23        </tr>
24      </tbody>
25    </table>
26  </body>
27</html>
```

Gambar 4.3. Skrip contoh untuk penyisipan gambar dan text

4. Contoh Penggunaan menu menggunakan Frame HTML

Sebelum membuat sebuah frame, kita buat beberapa dokumen web yang nanti akan ditampilkan ke dalam tiap frame. Misal kita akan membuat tiga halaman yang akan ditampilkan yang terdiri dari halaman **header.html**, **pascasarjana.html**, **menuutama.html**, **home.html** dan **menu.html**. sebuah tips jika akan membuat sebuah website dengan sub subnya, akan lebih efektif lagi jika kita membuat sub-sub nya terlebih dahulu. Misal membuat header, menu-menu baru dibuat frame utama berisi sub-sub diatas.

a. Header

Simpan dengan nama **header.html**

```
1 <html>
2   <head>
3     <title> STMIK Anikom Yogyakarta </title>
4   </head>
5   <body>
6     <center>
7       <font face="tahoma">
8         <h3> STMIK Anikom Yogyakarta </h3>
9         <h5> Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta
10        Telp: (0274) 884201 - 207 Fax: (0274) 884208 Kodepos: 55283 </h5>
11        
12        <hr>
13      </font>
14    </center>
15  </body>
16 </html>
```

Gambar 4.4. Skrip contoh untuk pembuatan header

b. Menu Utama

Selanjutnya adalah kita membuat menu utama. Dimana menu utama ini adalah menu paling depan sebagai “menu penyambut” ketika user mengakses halaman website yang kita buat. Berikut ini adalah skrip yang dibuat. Simpan dengan nama **menuutama.html**

```
1 <html>
2   <head>
3     <title> Sejarah Amikom </title>
4   </head>
5   <body>
6     <h3> STMIK Amikom </h3>
7     <pre>
8       <font face="tahoma">
9         Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta (selanjutnya disebut STMIK AMIKOM YOGYAKARTA) merupakan salah satu perguruan tinggi swasta yang berkedudukan di Provinsi DIY Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta. Beberapa jurusan yang terdapat di STMIK Amikom adalah sebagai berikut:
10
11       <ol>
12         <li> S2. Magister Teknik Informatika </li>
13         <li> S1. Teknik Informatika
14         <li> S1. Sistem Informatika </li>
15         <li> D3. Teknik Informatika </li>
16         <li> D3. Manajemen Informatika </li>
17       </ol>
18
19     </body>
20   </html>
```

Gambar 4.5. Skrip contoh untuk pembuatan menu utama

c. Sub Menu Pasca Sarjana

Pada langkah selanjutnya adalah membuat sub menu **PascaSarjana.html**. Mari kita ketikkan skrip dibawah ini kedalam aplikasi editor.

```
1 <html>
2   <head>
3     <title> Magister Teknik Informatika Amikom </title>
4   </head>
5   <body>
6     <img src ="mti logo.jpg" width="168" height="131" alt="STMIK Amikom" align="left">
7     <h3> Magister Teknik Informatika Amikom </h3>
8     <hr>
9     <pre>
10
11
12
13
14   Program Magister Teknik Informatika Program Pascasarjana STMIK AMIKOM Yogyakarta diarahkan pada
15   hasil lulusan yang berbudi luhur, berjiwa entrepreneur dan memiliki kualifikasi sebagai berikut:
16   1. Kemampuan merancang dan mengelola projek-projek perangkat lunak
17   2. Kemampuan menguasai aspek teknis dan manajemen dari pemanfaatan teknologi informasi
18   3. Kemampuan menganalisa dan menghadapi dampak dari teknologi informasi
19   4. Kemampuan mempunyai daya inovatif dalam menghasilkan produk/jasa teknologi informasi
20   5. Kemampuan merancang, mengembangkan, dan mengimplementasikan projek dalam bidang media digital
21
22     <table width="100%" border="1" cellpadding="1" cellspacing="1">
23       <tr bgcolor="#e2e2e2">
24         <td>Tahapan
25         <td>Mata Kuliah Reguler</td>
26       </tr>
27       <tr>
28         <td rowspan="7">Semester 1</td>
29         <td>Analysa and System Design</td>
30       </tr>
31       <tr><td>Design and Networking Management</td></tr>
32       <tr><td>Software Engineering</td></tr>
33       <tr><td>Information System</td></tr>
34       <tr><td>Database Management System</td></tr>
35       <tr><td>Praktikum Aplikasi Internet</td></tr>
36       <tr><td>Praktikum Dasar Jaringan Komputer</td></tr>
37
38     </table>
39   </body>
  </html>
```

Gambar 4.6. Skrip contoh untuk pembuatan sub menu “pasca sarjana”

d. Menu Pilihan Margin Kiri

Langkah selanjutnya adalah membuat bagian dimana untuk memberikan akses kepada pengguna dengan menyajikan kumpulan link-link dari sub menu yang ada. Contoh skrinya adalah sebagai berikut. Simpan dengan nama **menu.html**

```
1 <html>
2   <head>
3     <title> STMIK Amikom Yogyakarta </title>
4   </head>
5   <body>
6     <center>
7       [ <a href="menu_utama .html" target="main"> Home </a> ]<br>
8       [ <a href="Pasca .html" target="main"> MTI ]<br>
9       [ <a href="#"> S1 Teknik Informatika </a> ]<br>
10      [ <a href="#"> S1 Sistem Informasi </a> ]
11    </center>
12  </body>
13 </html>
```

Gambar 4.7. Skrip contoh untuk pembuatan sub menu pilihan

e. Frame

Setelah kita telah selesai membuat sub menu dari website tersebut, selanjutnya kita akan membuat frame. Agar dapat memilah satu halaman website menjadi beberapa bagian yang berisi tiap halaman tersebut. Berikut ini adalah contoh skrip yang akan kita buat. Simpan dengan nama **home.html**

```
1 <html>
2   <head>
3     <title>Menggunakan Frame</title>
4   </head>
5   <frameset rows=200,*>
6     <frame src="Header .html" scrolling="no" name="atas">
7     <frameset cols=250,*>
8       <frame src="menu .html" name="kiri">
9       <frame src="menu" name="main">
10    </frameset>
11
12    <noframes>
13      <body>
14        </body>
15    </noframes>
16  </html>
```

Gambar 4.8. Skrip contoh untuk pembuatan frame

Bab 5. Cascading Style Sheet (CSS)

Tujuan

1. Mahasiswa Memahami penggunaan CSS dalam HTML
2. Mahasiswa Mampu mengaplikasikan dan membuat CSS

Overview

Cascading Style Sheets (CSS) adalah suatu *bahasa stylesheet* yang digunakan untuk mengatur tampilan sebuah dokumen yang ditulis dalam *bahasa markup*. CSS diperkenalkan untuk pengembangan website kurang lebih pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi style yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan parent-child pada setiap style.

Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Spesifikasi CSS diatur oleh *World Wide Web Consortium* (W3C). CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh *browser basis-suara* atau *pembaca layar*), dan juga alat pembaca braille. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

Setelah CSS distandarisasikan, Internet Explorer dan Netscape melepas browser terbaru mereka yang telah sesuai atau paling tidak hampir mendekati dengan standar 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.

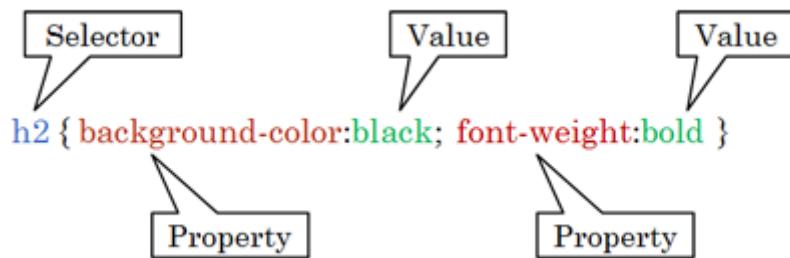
Pengertian CSS

- CSS adalah singkatan dari Cascading Style Sheets. Berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil di halaman web.
- 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.
- CSS terdiri dari style sheet yang memberitahukan browser bagaimana suatu dokumen akan disajikan.
- Fitur-fitur baru pada halaman web lama dapat ditambahkan dengan bantuan style sheet.
- Saat menggunakan CSS, Anda tidak perlu menulis font, color atau size pada setiap paragraf, atau pada setiap dokumen. Setelah Anda membuat sebuah style sheet, Anda dapat menyimpan kode tersebut sekali saja dan dapat kembali menggunakannya bila diperlukan.

Keuntungan Menggunakan CSS

- CSS memberikan keseragaman pada halaman web.
- Dengan CSS dapat menghemat banyak waktu dan pekerjaan berulang. Saat menggunakan CSS, perubahan tidak perlu dilakukan dalam setiap halaman web. Anda hanya perlu membuat perubahan dalam style sheet.
- CSS memungkinkan Anda untuk memuat halaman web Anda dengan mudah.
- Layers (Lapisan), seperti item pop-up, dapat digunakan dalam dokumen. \
- CSS membantu Anda memelihara halaman web Anda dengan mudah dan efektif.

5.1 Pengertian Selector, Property dan Value pada CSS



Gambar 5.1 Selector, Property dan Value pada CSS

5.1.1 Selector

Karena kode CSS digunakan untuk mengubah/memanipulasi tampilan dari tag HTML, CSS membutuhkan suatu cara untuk '*mengaitkan*' atau **menghubungkan** kode CSS dengan tag HTML yang sesuai. Hal inilah yang dimaksud dengan **Selector** dalam CSS.

Sesuai dengan namanya, selector digunakan untuk mencari bagian web yang ingin dimanipulasi atau yang ingin di-style. Misalnya : "*cari seluruh tag <p>*", atau "*cari seluruh tag HTML yang memiliki atribut class="warning"*" atau "*cari seluruh link yang ada di dalam tag <p>*".

Selector paling dasar dari CSS adalah tag dari HTML itu sendiri, misalnya: tag *p*, *i*, *h1*, *li*, dll. Selector didalam CSS dapat menjadi kompleks tergantung kebutuhannya.

5.1.2 Property

Property CSS adalah *jenis style*, atau elemen apa yang akan diubah dari sebuah tag HTML. CSS memiliki puluhan **property** yang dapat digunakan agar menampilkan hasil akhir yang kita inginkan. Hampir semua **property** dalam CSS dapat dipakai untuk seluruh **selector**.

Jika selector digunakan misalnya untuk "*mencari seluruh tag <p>*", maka property adalah "*efek apa yang ingin dimanipulasi dari tag p tersebut*", seperti ukuran text, warna text, jenis fontnya, dll.

5.1.3 Value

Value CSS adalah nilai dari **property**. Misalkan untuk property **background-color** yang digunakan untuk mengubah warna latar belakang dari sebuah selector, **value** atau nilainya dapat berupa **red**, **blue**, **black**, atau **white**.

5.2 Penempatan Sebuah CSS

Ada tiga cara untuk menempatkan sebuah CSS ke dalam sebuah halaman web, ketiga cara ini dapat digunakan untuk memformat halaman web dengan style yang diingkan.

5.2.1 Inline Style Sheet

CSS dalam posisi *inline style sheet* dituliskan menjadi satu dengan halaman web yang akan diatur *style*-nya dan menjadi bagian dari *body*. Penulisan *style* dilakukan dengan cara menambahkan atribut title pada elemen (tag) HTML yang akan diatur *style*-nya. Oleh karena itu, untuk mengimplementasikan CSS pada halaman web, semua tag harus diformat secara independen.

Dengan menggunakan model penempatan *inline style sheet*, jika ada sebuah tag HTML yang digunakan berulang kali dalam sebuah halaman web, pembuat website dapat mengimplementasikan *style* yang berbeda pada tag tersebut. Ketika menggunakan *inline style sheet* untuk memanipulasi halaman web, pembuat web hanya dapat menggunakan satu *property* saja pada tag HTML yang akan dimanipulasi.

5.2.2 Embedded Style Sheet

Sama halnya dengan CSS dalam posisi *inline style sheet*, penulisan CSS dalam posisi *embedded style sheet* juga menjadi satu dengan halaman web yang akan diatur *style*-nya, hanya saja, posisi CSS menjadi bagian dari header (berada diantara tag *<head>*) dengan menambahkan tag *<style type="text/css">*.

Dengan menggunakan model penempatan CSS sebagai *embedded style sheet*, pembuat web cukup satu kali mendefinisikan *style* yang akan dikenakan pada tag – tag yang berada dalam halaman web. Jika ada sebuah tag yang digunakan secara berulang, secara otomatis akan mempunya *style* yang sama, berbeda dengan model *inline style sheet* yang mengharuskan pembuat web menentukan *style* pada tag –

tag yang digunakan berulang kali dan memungkinkan untuk menentukan *style* yang berbeda pada tag tersebut.

5.2.3 Linked Style Sheet

Berbeda dengan 2 (dua) model penempatan CSS sebelumnya, menggunakan linked style sheet berarti harus menyediakan sebuah file CSS khusus berisi berbagai format *style* yang terpisah dari halaman web. File ini nantinya akan dipanggil oleh halaman web yang membutuhkan pengaturan style.

Dengan menggunakan model ini, *style* akan terpusat pada sebuah file, sehingga jika ada pengubahan *style*, pembuat website tidak perlu merubah di semua halaman web yang dibuat, tetapi cukup dengan merubah *style* yang didefinisikan pada file CSS.

Dengan menggunakan model ini, *style* akan terpusat pada sebuah file, sehingga jika ada pengubahan *style*, pembuat website tidak perlu merubah di semua halaman web yang dibuat, tetapi cukup dengan merubah *style* yang didefinisikan pada file CSS.

5.3 Penggunaan CSS Pada Halaman Web

Kita dapat menggunakan salah satu dari ketiga penempatan CSS yang ada, namun tidak menutup kemungkinan juga kita dapat menggunakan dua atau ketiganya dalam sebuah website jika memang diperlukan.

5.3.1 Pengolahan Font

Salah satu tag HTML yang biasa digunakan untuk memanipulasi font adalah paragraf (<p>). Beberapa tag yang lain juga memungkinkan untuk dimanipulasi atau ditambahkan CSS untuk memanipulasi font jika pada tag HTML tersebut akan berisi tulisan / *text*, seperti : <th><td>, <a>, , <h1>...<h6>, dan tag – tag lain yang memungkinkan berisi text.

Bentuk manipulasi font yang dimungkinkan dapat dilihat pada tabel di bawah ini :

Tabel 5.1 Font Properties

Nama Property	Value	Contoh Penggunaan
Font-family	Nama font	P { font-family: arial, Helvetica; }
Font-size	Nilai integer dengan pilihan satuan : em pt px %	P { font-size: 24em; }
Font-style	Oblique italic normal	P { font-style: italic; }
Font-variant	Normal small-caps	P { font-variant: small-caps; }
Font-weight	Normal bold bolder 100-900	P { font-weight: bold; }

5.3.2 Memanipulasi Color dan Background

Bentuk lain dari memanipulasi halaman web adalah dengan menentukan warna pada tulisan, menambahkan warna background, atau dengan menambahkan gambar sebagai background.

Beberapa property yang bisa digunakan untuk memanipulasi beberapa hal di atas, ditampilkan pada tabel di bawah ini.

Tabel 5.1 Properties Warna dan Gambar Latar

Nama Property	Value	Contoh Penggunaan
Background-color	Rangkaian 6 digit heksadesimal atau nama warna	Body { Background-color : green; }
Background-Image	url dimana image disimpan	Body { Background-image : url(gambar1.jpg); }
Background-attachment	Fixed scroll inherit	Body { Background-attachment : scroll; }
Background-Repeat	Inherit repeat no-repeat repeat-x repeat-y	Body { Background-repeat : repeat; Background-image : url(image/gambar1.jpg); }
Background-position	Bottom center left right top inherit	Body { Background-repeat : no-repeat; Background-image : url(image/gambar1.jpg); Background-position : bottom center }
Color	Rangkaian 6 digit	H1 { color : red; }

5.4 Mengenal Jenis-jenis Selector Dasar CSS

Selector adalah sebuah pola (**pattern**) yang digunakan untuk ‘*mencari*’ suatu tag di dalam HTML. Analogi untuk *selector*, misalnya: *mencari semua tag p*, atau *mencari seluruh tag h1 yang memiliki atribut class=judul*.

5.4.1 Universal Selector

Universal selector hanya ada 1 di dalam CSS, yaitu tanda bintang “*”. Selector ini bertujuan untuk ‘*mencari*’ semua tag yang ada.

Contoh Universal Selector CSS:

```
* {  
    color: blue;  
    background-color: white;}  
}
```

Kode CSS diatas bermaksud untuk membuat seluruh tag HTML berwarna biru, dan background berwarna putih.

5.4.2 Element Type Selector

Element Type Selector atau Tag Selector adalah istilah untuk selector yang nilainya merupakan tag HTML itu sendiri. Setiap tag HTML bisa digunakan sebagai selector, dan seluruh tag tersebut akan *ditangkap* oleh selector ini.

Contoh Element Type Selector CSS:

```
h1 {  
    text-decoration: underline;  
}  
p {  
    font-size:14px;  
}
```

Contoh kode CSS diatas akan membuat semua tag <h1> akan bergaris bawah, dan seluruh tag <p>akan berukuran 14pixel.

Efek dari element type selector adalah dari awal tag, sampai akhir tag. Jika didalam tag <p> terdapat tag <i>, maka tag tersebut juga akan berukuran 14 pixel, sampai ditemui tag penutup </p>.

5.4.3 Class Selector

Class Selector merupakan salah satu selector yang paling umum dan paling sering digunakan. *Class Selector* akan ‘mencari’ seluruh tag yang memiliki atribut *class* dengan *nilai* yang sesuai.

Untuk penggunaan Class Selector, kita harus memiliki tag HTML yang mempunyai atribut *class*. Contohnya:

```
<p class="paragraf_pertama"> Ini adalah sebuah paragraf pertama</p>
<h1 class="judul">Judul Artikel</h1>
<h2 class="judul penting berwarna">Sub Judul Artikel<h2>
```

Perhatikan bahwa untuk semua tag diatas, kita menambahkan atribut *class* dengan nilainya adalah nama dari kelas itu sendiri. Sebuah nama class dapat dimiliki oleh lebih dari 1 tag, dan dalam sebuah tag dapat memiliki lebih dari 1 class.

Contohnya dalam baris terakhir pada contoh diatas, tag h2 memiliki atribut *class="judul penting berwarna"*. Tag ini terdiri dari 3 class, yaitu *judul*, *penting*, dan *class berwarna*.

Sedangkan untuk kode CSS Class Selector adalah sebagai berikut:

```
.paragraf_pertama {
    color: red;
}

.judul {
    font-size: 20px;
}

.penting {
    color: red;
    font-size: 1em;
}
```

Untuk menggunakan class selector, di dalam CSS kita menggunakan tanda titik sebelum nama dari class.

Untuk contoh kita, seluruh class yang memiliki nilai “*paragraf_pertama*”, warna text akan menjadi merah. Dan seluruh class *judul* akan memiliki font 20 pixel.

5.4.4 ID Selector

ID Selector bersama-sama dengan class selector merupakan selector paling umum dan juga sering dipakai (walau tidak sesering *class selector*). Penggunaan *ID selector* hampir sama dengan class selector, dengan perbedaan jika pada Class Selector kita menggunakan atribut class untuk tag HTML, untuk ID selector, kita menggunakan atribut id.

Contoh penggunaan atribut id pada tag HTML

```
<p id="paragraf_pembuka"> Ini adalah sebuah paragraf pembuka</p>
```

```
<h1 id="judul_utama">Judul Artikel</h1>
```

```
<h2 id="sub_judul">Sub Judul Artikel</h2>
```

Atribut id selain untuk selector CSS, juga berperan sebagai kode unik untuk masing-masing tag (terutama dipakai untuk kode JavaScript). Karena hal tersebut, id yang digunakan harus unik dan tidak boleh sama. Dengan kata lain, id hanya bisa digunakan satu kali dalam sebuah halaman web dan tidak boleh sama.

Contoh penggunaan id selector kode CSS Class Selector adalah sebagai berikut:

```
#paragraf_pembuka {  
    color: red;  
}  
  
#judul_utama {  
    font-size: 20px  
}
```

Di dalam kode CSS, kita menggunakan tanda pagar "#" sebagai penanda bahwa kita mencari tag yang memiliki id tersebut.

5.4.5 Attribute Selector

Selector ini sedikit lebih *advanced* dibandingkan dengan selector-selector sebelumnya. Atribut Selector ini digunakan untuk mencari seluruh tag yang memiliki atribut yang dituliskan.

Contoh penggunaan Attribute Selector kode CSS adalah sebagai berikut:

```
[href] {  
    font-size: 20px;  
}
```

```
[type="submit"] {  
    width:30px;  
}
```

Seperti yang dapat dilihat dari contoh diatas, setiap atribut selector harus berada diantara tanda kurung siku "[" dan "]". [href] akan cocok dengan seluruh tag yang memiliki atribut href, apapun nilai dari href (href biasanya terdapat pada tag <a>). Untuk contoh [type="submit"] akan cocok dengan tag yang memiliki *atribut type* dengan nilai *submit*, yang dalam hal ini adalah tombol *submit* dalam *form*. Walaupun memiliki kemampuan mencari tag yang sangat spesifik, namun atribut selector ini tidak terlalu sering digunakan.

LATIHAN

1. Penggunaan Inline Style Sheet

Simpan dengan nama **latihan24.html**

```
1 <html>
2   <head>
3     <title>CSS untuk memanipulasi font</title>
4   </head>
5   <body>
6     <h3 style="font-family:'tahoma', Courier ">Sejarah AMIKOM</h3>
7     <p style="font-family:Verdana, Arial, Helvetica, sans-serif">
8       Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta,
9       selanjutnya disebut STMIK AMIKOM YOGYAKARTA merupakan salah satu
10      perguruan tinggi swasta yang berkedudukan di Provinsi DIY
11      Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta.
12    </p>
13
14    <p style="font-size:16px">
15      AMIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik
16      Informatika. Program studi ini masing-masing dikelola oleh seorang
17      Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perangkat Dosen,
18      dan Staff Administrasi.
19
20
21    <p style="font-style:italic">
22      Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
23      tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Penyelenggaraan
24      Program-Program Studi dan Pendirian Sekolah Tinggi Manajemen Informatika dan Komputer
25      STMIK AMIKOM Yogyakarta di DIY (Perubahan bentuk dari AMIKOM) yang diselenggarakan
26      oleh Yayasan AMIKOM Yogyakarta di DIY.
27    </p>
28  </body>
29
```

Gambar 5.2 Contoh Skrip CSS Inline Style Sheet

2. Penggunaan Embedded Style Sheet. Simpan dengan nama **latihan25.html**

```
1 <html>
2   <head>
3     <title>CSS untuk memanipulasi font</title>
4     <style type="text/css">
5       p{font-family:Verdana, Arial, Helvetica, sans-serif}
6       td{font-weight:bold font-family: calibri}
7       a {font-size:50px}
8   </head>
9   <body>
10    <h2>Sejarah Amikom</h2>
11    <p>Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta,  

12      selanjutnya disebut STMIK AMIKOM YOGYAKARTA) merupakan salah satu  

13      perguruan tinggi swasta yang berkedudukan di Provinsi DIY  

14      Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta.</p>
15    <table>
16      <tr>
17        <td>
18          AMIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik  

19          Informatika. Program studi ini masing-masing dikelola oleh seorang  

20          Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perangkat Dosen,  

21          dan Staff Administrasi.
22      </td>
23    </tr>
24
25    <tr>
26      <td>
27        Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia  

28        tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Penyelenggaraan  

29        Program-Studi dan Pendirian Sekolah Tinggi Manajemen Informatika dan Komputer  

30        STMIK) AMIKOM Yogyakarta di DIY (Perubahan bentuk dari AMIKOM) yang diselenggarakan  

31        oleh Yayasan AMIKOM Yogyakarta di DIY.
32      </td>
33    </tr>
34  </table>
35  <br>
36  Untuk mencoba menggunakan aplikasi Online Test, silahkan klik link di bawah ini :<br>
37  <a href="http://www.bernadtagger.com">Online Test</a>
38 </body>
</html>
```

Gambar 5.3 Contoh Skrip CSS Embedded Style Sheet

3. Penggunaan Linked Style Sheet. Simpan File Pertama dengan nama **style definition.css**:

```
1  /* Dokumen CSS */
2
3  tulisan_miring {
4      font-family:Arial, Helvetica, sans-serif;
5      font-size:12px;
6      font-style:italic;
7      }
8  .cetak_tebal {
9      font-family:Verdana, Arial, Helvetica, sans-serif;
10     font-size:15px;
11     font-weight:bold;
12     color:skyblue;
13     }
14
15 #14pt {
16     font-size:14pt;
17     }
18
19 p {
20     font-family:Verdana, Arial, Helvetica, sans-serif
21     }
22
23 td {
24     font-weight:bold
25     }
26
27 a {
28     font-size:24px
29     }
```

Gambar 5.4 Contoh Skrip Linked Style Sheet

Kemudian buat file kedua dengan nama **latihan26.html**

```
1 <html>
2   <head>
3     <title>CSS untuk memanipulasi font</title>
4     <link href="Style Definition.css" rel="stylesheet" type="text/css">
5   </head>
6   <body>
7     <h1>Sekolah Amikom</h1>
8     <table>
9       <tr>
10         <td>
11           <tulisan_miring>
12             Sekolah Tinggi Manajemen Informatika dan Komputer ANIKOM Yogyakarta,  

13             selanjutnya disebut STMIK ANIKOM YOGYAKARTA merupakan salah satu  

14             perguruan tinggi swasta yang berkedudukan di Provinsi DIY  

15             Pabupaten Sleman di bawah naungan Yayasan ANIKOM Yogyakarta.  

16             </tulisan_miring><br><br>;
17         </td>
18       </tr>
19       <tr>
20         <td class="cetak_tebal">
21           ANIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Jurik  

22           Informatika. Program studi ini wajib-wajib ditempuh oleh seorang  

23           Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perangkat Dosen,  

24           dan Staff Administrasi. <br>
25         </td>
26
27       <tr>
28         <td id="14pt">
29           Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia  

30           tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Pengelenggaran  

31           Program-Program Studi dan Pendidikan Sekolah Tinggi Manajemen Informatika dan Komputer  

32           STMIK ANIKOM Yogyakarta di DIY (Perbaikan berdasar dari ANIKOM yang dilenggarakan  

33           oleh Yayasan ANIKOM Yogyakarta di DIY).
34
35         </td>
36       </tr>
37
38     </table>
39   </body>
40 </html>
```

Gambar 5.5 Contoh Skrip Linked Style Sheet

4. Pemberian property multi kepada selector. Simpan dengan nama **latihan27.html**

```
2 <head>
3   <title>CSS untuk memanipulasi font</title>
4   <style type="text/css">
5     td{font-family:Arial, Helvetica, sans-serif;font-size:12px; font-style:italic}
6     a{font-family:Verdana, Arial, Helvetica, sans-serif;font-size:14px;font-weight:bold}
7   </style>
8 </head>
9 <h3>Sejarah Amikom </h3>
10 <table>
11   <tr>
12     <td>
13       Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta selanjutnya disebut STMIK AMIKOM YOGYAKARTA merupakan salah satu perguruan tinggi swasta yang berkedudukan di Provinsi DIY Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta.<br>&ampnbsp
14     </td>
15   </tr>
16   <tr>
17     <td>
18       AMIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik Informatika. Program studi ini masing-masing dikelola oleh seorang Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perangkat Dosen, dan Staff Administrasi. <br>&ampnbsp
19     </td>
20   </tr>
21   <tr>
22     <td>
23       Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
24       tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Penyelenggaraan
25       Program-Studi dan Pendirian Sekolah Tinggi Manajemen Informatika dan Komputer
26       STMIK AMIKOM Yogyakarta di DIY (Perubahan bentuk dari AMIKOM yang diselenggarakan
27       oleh Yayasan AMIKOM Yogyakarta di DIY. <br>&ampnbsp
28     </td>
29   </tr>
30   <tr>
31     <td>
32       Untuk kembali ke berita sebelumnya,
33       <a href="2. CSS Embedded Style Sheet .html">
34         silahkan klik disini << bagian ini telah mendapat style CSS pada "a" dan multiple dari "td"
35       </a>
36     </td>
37   </tr>
38 </table>
39 </html>
```

Gambar 5.6 Contoh Skrip dalam pemberian property multi kepada selector

5. Manipulasi font dengan menggunakan selector bebas, class, dan ID. Simpan dengan nama latihan28.html

```
1 <html>
2   <head>
3     <title>
4       CSS untuk memanipulasi font
5     </title>
6
7     <style type="text/css">
8       tulisan_miring {
9         font-family:Arial, Helvetica, sans-serif;
10        font-size:12px;
11        font-style:italic;
12      }
13      .cetak_tebal {
14        font-family:Verdana, Arial, Helvetica, sans-serif;
15        font-size:15px;
16        font-weight:bold;
17        color:skyblue;
18      }
19      #14pt {
20        font-size:14pt;
21      }
22    </style>
23
24   </head>
25   <body>
26     <h3>Sejarah Amikom </h3>
27     <table>
28       <tr>
29         <td>
30           <tulisan_miring>
31             Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta,
32             selanjutnya disebut STMIK AMIKOM YOGYAKARTA merupakan salah satu
33             perguruan tinggi swasta yang berkedudukan di Provinsi DIY
34             Kabupaten Sleman di bawah naungan Yayasan AMIKOM Yogyakarta.
35           </tulisan_miring><br>&ampnbsp
36         </td>
37       </tr>
38     </table>
39   </body>
40 </html>
```

```
36           </td>
37       </tr>
38
39   <tr>
40
41     <td class="cetak_tebal">
42         AMIKOM Yogyakarta memiliki Program Studi Manajemen Informatika dan Teknik
43         Informatika. Program studi ini masing-masing dikelola oleh seorang
44         Ketua Jurusan dan Sekretaris Jurusan yang didukung oleh Perangkat Dosen,
45         dan Staff Administrasi. <br>
46
47   </tr>
48
49   <tr>
50     <td id="14pt">
51         Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia
52         tanggal 24 April 2002, Nomor. 75/D/O/2002 tentang Pemberian Ijin Penyelenggaraan
53         Program-Studi dan Pendirian Sekolah Tinggi Manajemen Informatika dan Komputer
54         STMIK) AMIKOM Yogyakarta di DIY (Perubahan bentuk dari AMIKOM) yang diselenggarakan
55         oleh Yayasan AMIKOM Yogyakarta di DIY.
56
57   </td>
58 </tr>
59 </table>
60 </body>
</html>
```

Gambar 5.7 Contoh Hasil Skrip penggunaan multi kepada selector bebas, class dan ID

Bab 7. Pengenalan Javascript

Tujuan :

1. Mahasiswa mampu menjelaskan konsep dasar bahasa pemrograman JavaScript
2. Mahasiswa mampu menjelaskan cara kerja JavaScript pada sistem web
3. Mahasiswa mengetahui dan menjelaskan konsep pemrograman dinamis pada JavaScript
4. Mampu mengetahui perintah-perintah dasar JavaScript.

Overview

Javascript merupakan bagian dari 3 teknologi penting yang harus dikuasai programmer web, yakni **HTML** untuk konten (isi dari website), **CSS** untuk tampilan (presentation), dan **JavaScript** untuk interaksi (behavior). Hampir seluruh website modern saat ini menggunakan *JavaScript* untuk membuat berbagai aplikasi yang dapat berinteraksi dengan user, seperti *validasi form HTML*, games, kalkulator, fitur *chatting*, dll.

Dalam mempelajari *JavaScript*, sebaiknya telah menguasai dasar-dasar **HTML**, dan bisa membuat halaman web sederhana menggunakan **HTML**. Pengetahuan tentang **CSS** dan **PHP** juga akan membantu, walaupun tidak diharuskan.

7.1 Pengertian JavaScript

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat interaksi atau menambah fitur web dinamis ke dalam sebuah web.

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.

7.2 Fungsi JavaScript Dalam Pemograman 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*.

7.3 Cara Menjalankan Kode Program JavaScript

Cara penulisan *JavaScript* mirip dengan penulisan bahasa pemrograman web lainnya seperti **PHP** dan **CSS**, yakni dengan menyisipkan kode *JavaScript* di dalam **HTML**.

7.4 Cara Memasukkan (input) kode JavaScript ke dalam HTML

JavaScript termasuk jenis bahasa *script*, yang digunakan di dalam file **HTML**. Untuk menginput, atau memasukkan kode **JavaScript** ke dalam **HTML**, *JavaScript* menyediakan 4 alternatif, yaitu :

7.4.1 Menggunakan tag <script> (internal JavaScript)

Cara pertama untuk menginput kode *JavaScript* ke dalam halaman **HTML** adalah dengan menggunakan **tag <script>** secara *internal*. **Internal** disini berarti bahwa kode *JavaScript* ditulis pada halaman yang sama dengan **HTML**, atau di dalam satu file **HTML**.

Cara ini merupakan cara yang paling sering digunakan, jika kode *JavaScript* tidak begitu panjang, dan hanya digunakan di 1 halaman saja. Kode *JavaScript* yang akan diinput diletakkan diantara *tag pembuka <script>* dan *tag penutup </script>*

7.4.2 Menggunakan tag <script scr=""> (external JavaScript)

Cara atau metode kedua untuk menginput kode **JavaScript** ke dalam halaman **HTML** adalah dengan memindahkan kode *JavaScript* ke dalam sebuah file terpisah, lalu ‘*memanggilnya*’ dari **HTML**. Cara ini sangat disarankan karena akan memberikan banyak keuntungan dan fleksibilitas dalam membuat program *JavaScript*.

Sebuah file **JavaScript** disimpan dalam ekstensi **.js**, seperti: **kode.js**, **register.js**, atau **kodeku.js**. Dari halaman **HTML**, kita memanggilnya menggunakan **tag <script>** dengan *atribut src*. Atribut **src** berisi alamat dari file *javascript* tersebut

7.4.3 Menggunakan Event Handler (Inline JavaScript)

Cara ketiga untuk menjalankan JavaScript adalah dengan memanggilnya menggunakan **Event Handler** dari dalam *tag HTML*. secara sederhananya, **event handler** adalah pemanggilan kode javascript ketika 'sesuatu' terjadi dalam **tag HTML**.

Sesuatu disini maksudnya ketika sebuah element dalam HTML di *klik*, di *klik kanan*, di arahkan *mouse*, dan lain-lain. *Event handler* di dalam **JavaScript** ditulis dengan penambahan kata **on**. Sehingga jika sebuah tombol di-klik, maka disebut sebagai **onclick**, jika mouse berada diatas element disebut sebagai **onmouseover**, dan lain-lain.

Event yang disediakan JavaScript terbagi menjadi 2 (dua) jenis, yaitu event sistem dan event pengguna. Event sistem adalah event yang selalu ada dan disediakan oleh JavaScript serta sangat terkait dengan sistem dimana halaman web ditampilkan. Sedangkan event pengguna adalah event yang hanya tersedia relatif terhadap konteks elemen halaman web apa yang diacu oleh pengguna. Berikut dibawah ini adalah event system

Tabel 7.1 Event System dalam JavaScript

Event	Keterangan
onError	Event ini dibangkitkan ketika terjadi kesalahan. Event ini dimiliki oleh objek Window dan Image
onLoad	Event ini dibangkitkan ketika suatu objek selesai ditampilkan pada halaman web. Objek yang dimaksud pada keterangan ini adalah objek Window, Frame dan Image
onUnload	Event ini dibangkitkan ketika suatu objek telah dikeluarkan dari ruang memori, atau telah selesai digunakan. Objek ini adalah objek Window, Frame dan Image

Tabel 7.2 List event pengguna beserta keterangannya

Event	Keterangan
onAbort	Event yang dimiliki oleh objek image ini akan dibangkitkan ketika sebuah gambar dihentikan proses pemunculannya yang dimungkinkan akibat penekanan tombol stop pada browser
onBlur	Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element window, frame, select, text, dan textarea) kehilangan focusnya
onChange	Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element select, text dan textarea) telah diubah nilainya sebelum element tersebut kehilangan focusnya

onClick	Event ini dibangkitkan ketika terjadi aksi klik (dilakukan oleh pengguna) terhadap element, dimana element yang dimaksud adalah semua element dari form yang dapat diklik seperti element button (tombol). Aksi klik yang dimaksud adalah adanya penekanan tombol klik kir mouse terhadap suatu element
onContextMenu	Event ini terbangkitkan ketika terjadi aksi klik kanan dari tombol mouse terhadap suatu element
onDblClick	Hampir sama dengan event onClick, namun akan dibangkitkan ketika terjadi aksi klik ganda terhadap suatu element
onFocus	Event ini adalah kebalikan dari event onBlur, dimana akan dibangkitkan ketika suatu element dikenai focus. Element-element yang dimaksud adalah element window, frame, select, text dan textarea
onHelp	Event ini dibangkitkan ketika terjadi penekanan terhadap tombol F1, yang umumnya digunakan berbagai aplikasi sebagai tombol shortcut untuk menampilkan fasilitas bantuan (help) . Anda dapat menggunakan event ini untuk mengalihkan fitur help milik browser menjadi fitur help yang Anda buat sendiri untuk halaman web Anda.
onKeydown	Event ini dibangkitkan ketika terjadi penekanan ke bawah tombol keyboard
onKeypress	Event ini dibangkitkan ketika terjadi aksi penekanan tombol keyboard. Aksi penekanan disini adalah sebuah aksi lengkap penekanan tombol keyboard, mulai dari memencet tombol keyboard tersebut hingga melepaskannya
onkeyup	Event ini dibangkitkan ketika tombol keyboard yang awalnya ditekan lalu dilepas, ketika tombol dilepas event ini bangkit .
onmousedown	Event ini dibangkitkan, ketika tombol mouse ditekan ke bawah
onmousemove	Event ini dibangkitkan ketika mouse digerakkan (panah mouse berubah posisinya)
onmouseout	Event ini dibangkitkan ketika panah mouse keluar dari daerah lingkupan suatu element, yaitu element hyperlink dan element area.
onmouseover	Event ini dibangkitkan ketika panah mouse berada diatas element , yaitu element hyperlink dan element area .
onReset	Event yang dimiliki secara khusus oleh objek form ini akan dibangkitkan ketika terjadi penekanan terhadap tombol reset yang dimiliki form yang bersangkutan
onResize	Event ini dibangkitkan ketika jendela browser diubah ukurannya, baik diperbesar maupun diperkecil
onSelect	Event ini dibangkitkan ketika terjadi pemilihan teks pada lement text dan textarea dengan cara memberinya highlight (diblok)
onStop	Event ini dibangkitkan ketika pengguna melakukan penekanan terhadap tombol stop dari browsernya
onSubmit	Event ini dibangkitkan setelah terjadi penekanan tombol submit yang dimiliki sebuah form

7.4.4 Menggunakan URL (`href:="javascript:"`)

Cara terakhir (dan juga paling jarang digunakan saat ini) adalah dengan menyisipkan JavaScript ke dalam alamat **href** dari tag *HTML*. Cara ini disebut juga dengan **protocol javascript**. Disebut demikian, karena kita mengganti alamat link dari yang biasa menggunakan protocol **http://** menjadi **javascript**.

Dari ke-4 cara menginput kode **JavaScript**, memisahkan kode *JavaScript* kedalam sebuah file tersendiri (menggunakan **metode <script src="">**) adalah yang paling disarankan. Beberapa keuntungan menggunakan metoda `<script src>` bila dibandingkan dengan moteda cara memasukkan **JavaScript** lainnya adalah:

- Menyederhanakan halaman *HTML* dengan memindahkan seluruh kode **JavaScript**, sehingga halaman **HTML** hanya berisi konten saja.
- Sebuah file **JavaScript external** bisa digunakan untuk beberapa halaman *HTML*, sehingga jika diperlukan perubahan, kita hanya perlu mengedit sebuah file daripada mengubah secara satu persatu halaman *HTML* tempat *JavaScript* ditulis secara internal.
- Jika file *JavaScript* external digunakan oleh beberapa halaman, file tersebut hanya perlu *download* oleh web browser pada saat pertama kali saja. Pada saat loading halaman lainnya, web browser cukup mengambilnya dari *browser cache*, sehingga mempercepat loading halaman.

PENUGASAN

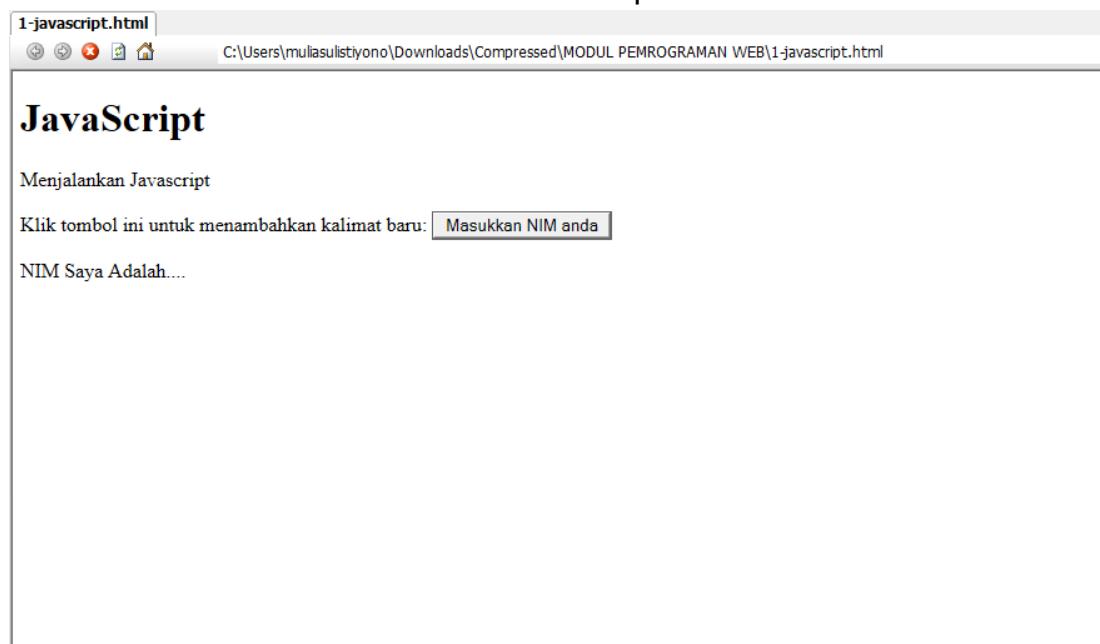
1. Menjalankan JavaScript

Silahkan buka aplikasi text editor, lalu ketikkan kode berikut:

simpan dengan nama **JS_1.html**

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Menjalankan JavaScript</title>
5
6 <script type="text/javascript">
7 function tambah_kalimat()
8 {
9     var a=document.getElementById("div_kalimat");
10    a.innerHTML+="<p>NIM Saya Adalah....</p>";
11 }
12 </script>
13
14 </head>
15
16 <body>
17 <h1>JavaScript</h1>
18 <p> Menjalankan Javascript </p>
19 Klik tombol ini untuk menambahkan kalimat baru:
20
21 <button id="tambah" onclick="tambah_kalimat()">Masukkan NIM anda</button>
22
23 <div id="div_kalimat"></div>
24 </body>
25 </html>
```

Gambar 7.1 Kode JavaScript di dalam HTML



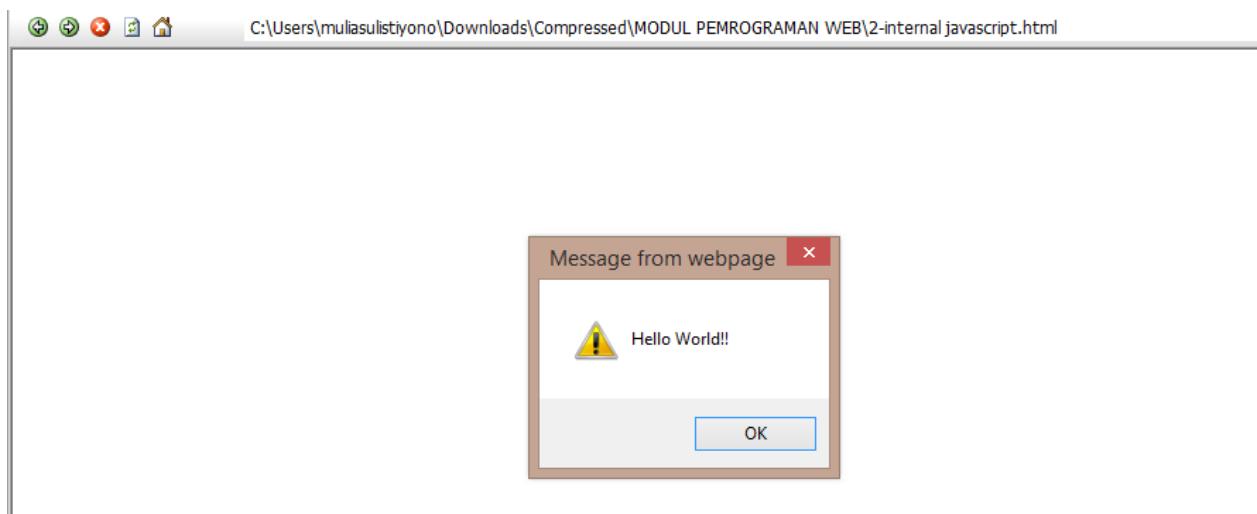
Gambar 7.2 Hasil Running Script dari kode JavaScript dalam HTML

2. Internal Javascript

simpan dengan nama **JS_2.html**

```
1 <!DOCTYPE html>
2 <html>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4   <head>
5     <title>Internal JavaScript </title>
6
7   <script>
8     alert("Hello World!!");
9   </script>
10
11 </head>
12
13 <body>
14   <h1>Cara Memasukkan JavaScript menggunakan tag script / internal JavaScript </h1>
15   <p>Mencoba Internal JavaScript</p>
16   <p>Masukkan NAMA_NIM</p>
17 </body>
18 </html>
```

Gambar 7.3 Kode Program HTML menggunakan internal JavaScript



Gambar 7.4 Hasil Running Script menggunakan internal JavaScript

3. External JavaScript

Buat Script terlebih JavaScript terlebih dahulu dan simpan dengan nama

kode_javascript.js

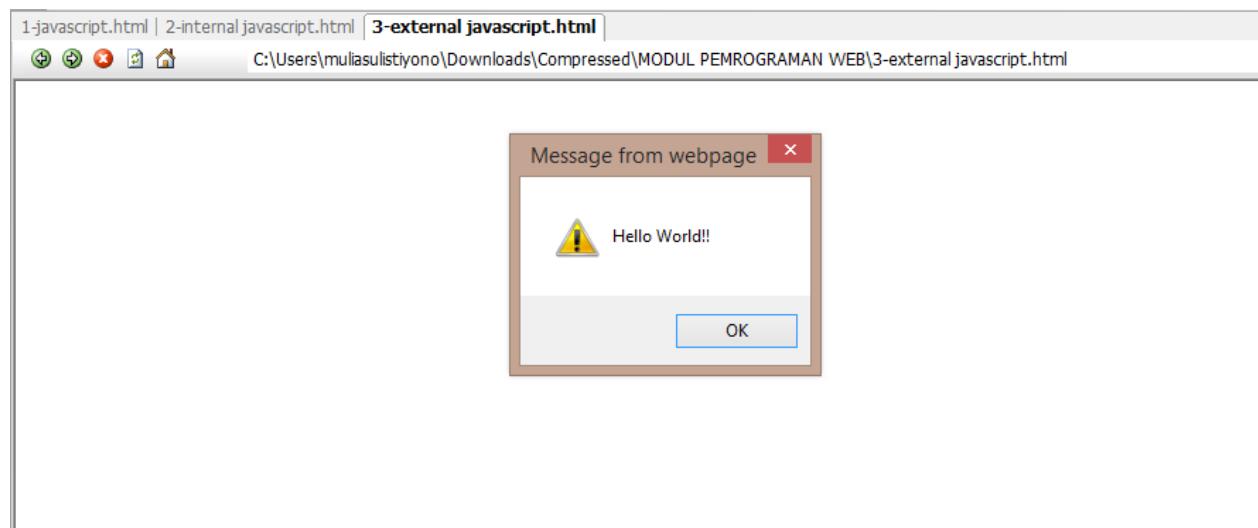
```
1 alert("Hello World!!")
```

Kemudian buat script html seperti gambar di bawah ini

Simpan dengan nama **JS_3.html**

```
1 <!DOCTYPE html>
2 <html>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4 <head>
5 <title>External JavaScript</title>
6
7 <script src="kode_javascript.js"></script>
8
9 </head>
10
11 <body>
12 <h1>Cara Memasukkan JavaScript menggunakan tag script src= / external JavaScript</h1>
13 <p>Mencoba External JavaScript</p>
14 <p>Masukkan NAMA_NIM</p>
15 </body>
16 </html>
```

Gambar 7.5 Kode Program HTML menggunakan eksternal JavaScript



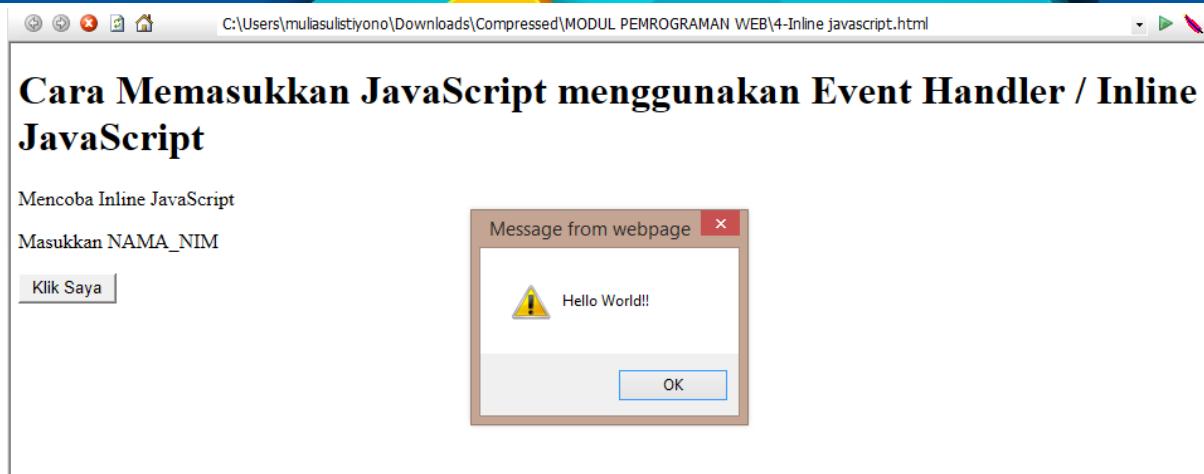
Gambar 7.6 Hasil Running Script menggunakan eksternal JavaScript

4. Inline JavaScript

Simpan dengan nama **JS_4.html**

```
1 <!DOCTYPE html>
2 <html>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4 <head>
5 <title>Inline JavaScript</title>
6 </head>
7
8 <body>
9 <h1>Cara Memasukkan JavaScript menggunakan Event Handler / Inline JavaScript</h1>
10 <p>Mencoba Inline JavaScript</p>
11 <p>Masukkan NAMA_NIM</p>
12
13 <button onclick="alert('Hello World!!!')>Klik Saya
14
15 </body>
16 </html>
```

Gambar 7.7 Kode Program HTML menggunakan inline JavaScript



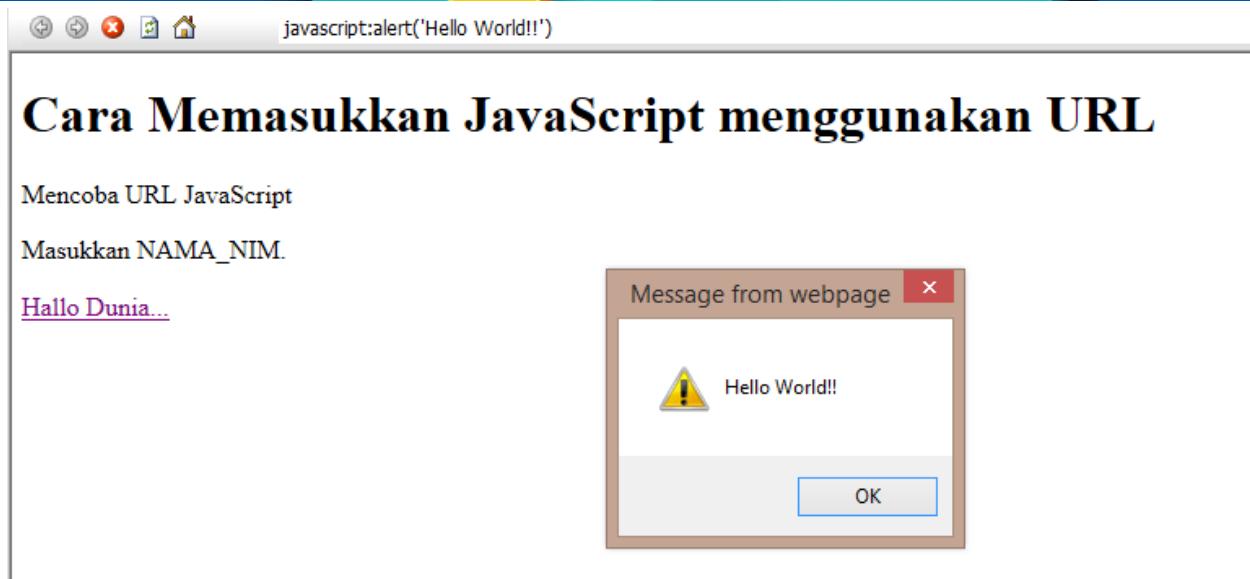
Gambar 7.8 Hasil running Script menggunakan inline JavaScript

5. JavaScript Menggunakan URL

Simpan dengan nama **JS_5.html**

```
1  <!DOCTYPE html>
2  <html>
3  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4  <head>
5  <title>URL JavaScript</title>
6  </head>
7
8  <body>
9  <h1>Cara Memasukkan JavaScript menggunakan URL</h1>
10 <p>Mencoba URL JavaScript</p>
11 <p>Masukkan NAMA_NIM.</p>
12
13 <a href="javascript:alert('Hello World!!!')">Hallo Dunia...</a>
14
15 </body>
16 </html>
```

Gambar 7.9 Kode Program HTML menggunakan *URL* JavaScript



Gambar 7.10 Hasil Running Script menggunakan *URL* JavaScript

Bab 8. Tipe Data dan Operasi Dasar JavaScript

Tujuan :

1. Mahasiswa mampu menjelaskan tipe data bahasa pemrograman JavaScript
2. Mahasiswa mampu menjelaskan variable yang digunakan dalam JavaScript
3. Mampu mengetahui perintah-perintah dasar JavaScript.

Overview

Pada pertemuan ini kita akan membahas jenis tipe data yang dapat diproses dalam JavaScript. Variabel yang digunakan untuk menyimpan data atau informasi yang akan diproses, serta operasi dasar JavaScript yang digunakan untuk melakukan pemrosesan data.

8.1 Aturan Dasar Penulisan Kode Program JavaScript

8.1.1 Perbedaan Penulisan Huruf (Case Sensitivity)

Di dalam **JavaScript**, penulisan huruf besar dan huruf kecil dibedakan, atau dalam istilah pemograman bersifat **Case Sensitif**. Hal ini berarti penulisan *variabel*, *keyword*, maupun nama *fungsi* di dalam JavaScript harus konsisten. Variabel **nama**, **Nama**, dan **NAMA** merupakan 3 variabel berbeda. Sedangkan untuk penulisan *keyword* **while**, harus ditulis dengan '**while**', bukan '**While**' atau '**WHILE**'.

Namun karena **HTML** sendiri tidak bersifat *case-sensitif*, kadang hal ini bisa mendatangkan permasalahan. Contohnya, jika menggunakan **event handler**, di dalam HTML kadang bisa ditulis: **onclick**, **onClick**, atau **OnClick**. Ketiga penulisan ini dibolehkan di dalam **HTML**. Akan tetapi untuk menghindari permasalahan, sebaiknya kita membuat kesepakatan untuk menggunakan huruf kecil untuk semua penulisan *keyword* dan *variabel* di dalam JavaScript.

8.1.2 Cara Penulisan Komentar dalam JavaScript

JavaScript mendukung 2 jenis **cara penulisan komentar**, yakni menggunakan karakter **//** untuk komentar dalam 1 baris, dan karakter pembuka

komentar /* dan penutup */ untuk komentar yang mencakup beberapa baris.

Gambar 8.1 merupakan contoh penulisan komentar dalam javascript.

```
1 <script>
2   // ini adalah komentar dalam 1 baris
3   /* Baris ini juga merupakan komentar */ //ini juga komentar
4
5   /* Komentar ini
6     mencakup beberapa
7     baris
8   */
9
10 /*
11  * Beberapa programmer
12  * menambahkan tanda bintang
13  * agar penulisan komentar
14  * lebih rapi
15  */
16 </script>
```

Gambar 8.1 Contoh Penulisan komentar dalam JavaScript

8.2 Variabel dalam JavaScript

Dalam bahasa pemograman, **variabel** adalah ‘penampung’ sebuah nilai. Tergantung dengan ‘nilai’ dari **variabel** tersebut, sebuah *variabel* di dalam JavaScript dapat bertipe Angka (**Number**), **String**,**Boolean**, atau yang lainnya.

8.2.1 Aturan Penamaan Variabel JavaScript

- Karakter pertama harus diawali dengan **huruf**, **underscore** (_) atau **tanda dollar** (\$)
- Karakter kedua dan seterusnya bisa ditambahkan dengan **huruf**, **angka**, **underscore** (_) atau **tanda dollar** (\$).

8.2.2 Cara Membuat Variabel JavaScript

Cara membuat variabel di dalam **JavaScript** di bedakan menjadi 2, yakni dengan menggunakan *keyword* **var**, dan **tanpa var**.

contoh penulisan menggunakan kata kunci **var** :

```
1 <script>
2   var a;
3   var b, c, d;
4   var e; var f;
5   var g=12;
6   var h="kuliah pemrograman web menggunakan javascript";
7 </script>
```

Gambar 8.2 contoh penulisan variable menggunakan kata kunci var

contoh penulisan tanpa menggunakan kata kunci **var** :

```
1 <script>
2   a;
3   b=2;
4   c=" kuliah pemrograman web menggunakan javascript";
5 </script>
```

Gambar 8.3 contoh penulisan variable tanpa menggunakan kata kunci var

Pembuatan variabel tanpa menggunakan *keyword* **var** memang lebih cepat, akan tetapi tidak disarankan.

8.3 Operator dalam JavaScript

Operator pada JavaScript terbagi menjadi enam, yaitu :

- Aritmatika
- Pemberian nilai (Assign)
- Pemanipulasi bit (bitwise)
- Pembanding
- Logika
- String

8.3.1 Operator Aritmatika

Tabel 8.1 Operator Aritmatika

Operator	Tunggal/Biner	Keterangan
+	Biner	Penjumlahan
-	Biner	Pengurangan
*	Biner	Perkalian
/	Biner	Pembagian
%	Biner	Modulus
-	Tunggal	Negasi
++	Tunggal	Penambahan dengan satu
--	Tunggal	Pengurangan dengan satu

Sama seperti pelajaran matematika yang kita pelajari, di dalam **JavaScript** operator *perkalian* dan *pembagian* lebih kuat dari pada

operator *penambahan* dan *pengurangan*. Perintah operasi $1 + 3 * 5$ akan proses oleh JavaScript menjadi $1 + (3 * 5)$. Operasi **modulus** (sisa hasil bagi) juga sama kuat dengan operasi *perkalian* dan *pembagian*.

Jika operasi matematika yang akan diproses memiliki lebih dari 1 operator yang sama kuat dalam sebuah perhitungan, **JavaScript** akan memprioritaskan urutan perhitungan dari kiri ke kanan. Operasi $3 * 4 / 5 * 5$ akan dikerjakan menjadi: $((3 * 4) / 5) * 5$.

Untuk membuat **prioritas** suatu operasi menjadi lebih tinggi, kita bisa menambahkan **tanda kurung** didalam program. Penggunaan tanda kurung juga akan mempermudah pembacaan program. Walaupun kita bisa menebak bagaimana operasi $5 + 3 * 2$ akan diproses, namun akan lebih mudah di baca jika operasi tersebut tetap dibuat dengan menambahkan tanda kurung menjadi $5 + (3 * 2)$.

8.3.2 Operator Pemberian Nilai (Assigment)

Operator assigment di gunakan untuk memberi nilai pada suatu varibel dengan tanda sama dengan ($=$), penulisan operator assigment memudahkan kita mempersingkat script.

Tabel 8.2 Operator Assigment

Operator	keterangan	Contoh	Ekuivalen
$=$	Sama dengan	$X=Y$	
$+=$	Ditambah dengan	$X+=Y$	$X=X+Y$
$-=$	Dikurangi dengan	$X-=Y$	$X=X-Y$
$*=$	Dikali dengan	$X*=Y$	$X=X*Y$
$/=$	Dibagi dengan	$X/=Y$	$X=X/Y$
$\%=$	Modulus dengan	$X\%=Y$	$X=X\%Y$
$\&=$	Bit AND dengan	$X\&=Y$	$X=X\&Y$
$ =$	Bit OR	$X =Y$	$X=X Y$

8.3.3 Operator Manipulasi Bit

Operasi ini berhubungan dengan pemanipulasi bit pada operan bertipe bilangan bulat.

Tabel 8.3 Operator Manipulasi Bit

Operator	Keterangan
&	Bit AND
	Bit OR
^	Bit XOR
~	Bit NOT
<<	Geser ke kiri
>>	Geser ke Kanan
>>>	Geser ke kanan dengan diisi nol

Contoh :

```
var A = 12; // A = 1100b
```

```
var B = 10; // B = 1010b
```

```
var C = A & B
```

maka akan dihasilkan bilangan seperti berikut :

1100b

1010b AND

1000b

```
var A = 12;
```

```
var C = A<< 2
```

```
var D = A >> 1
```

maka variabel C akan bernilai 48(0011 0000b)

variabel D akan bernilai 6 (0110b)

8.3.4 Operator Pembanding

Digunakan untuk membandingkan dua buah operan. Operan yang dikenal operator ini dapat bertipe string, numerik, maupun ekspresi lain.

Tabel 8.4 Operator Pembanding

Operator	Keterangan
==	Sama dengan
!=	Tidak sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

8.3.5 Operator Logika

Digunakan untuk mengoperasikan operan yang bertipe boolean.

Tabel 8.5 Operator Logika

Operator	Keterangan
&&	Operator logika AND
	Operator Logika OR
!	Operator logika NOT

Contoh :

```
var A = true;  
var B = false;  
var C = A && B; //menghasilkan false  
var D = A || B ; // false  
var E = !A; //false
```

8.3.2 Operator String

Selain operator pembanding, operator string pada JavaScript juga mengenal satu operator lagi yang bernama PENGGABUNGAN. Operator ini digunakan untuk menggabungkan beberapa string menjadi sebuah string yang lebih panjang.

Contoh :

```
nama = "Java" + "Script";  
akan menghasilkan "JavaScript" pada variabel nama
```

8.4 Jenis dan Tipe Data Dalam JavaScript

Tipe data dalam javascript diantaranya adalah tipe data **angka**, tipe data text (**string**), dan tipe data **boolean**. Selain itu dalam JavaScript ada **tipe data objek**. Contoh tipe data **objek** adalah tipe data tanggal (**date**), **array**, dan **fungsi**.

8.4.1 Tipe Data Angka

JavaScript tidak membedakan tipe data angka (**number**) antara angka bulat dengan angka desimal, atau tidak membedakan antara bilangan integer dengan float. Seluruh tipe data angka di dalam JavaScript disimpan dalam bentuk desimal (**float**).

Jangkauan tipe data angka di dalam **JavaScript** merujuk kepada aturan standar *IEEE 754* yang mencakup angka paling kecil $\pm 5 \times 10^{-324}$ dan angka paling tinggi $\pm 1.7976931348623157 \times 10^{308}$.

Angka di dalam **JavaScript** bisa dengan tepat merepresentasikan nilai antara -2^{53} sampai dengan 2^{53} , atau dari -9007199254740992 sampai 9007199254740992 .

Karena di dalam **JavaScript** sebuah variabel tidak perlu di deklarasikan akan bertipe apa, maka jika sebuah variabel diberikan nilai angka, maka variabel tersebut telah menjadi variabel dengan tipe angka. Berikut adalah contohnya:

```
1 <script>
2 var a=15; // deklarasi dengan var
3 b=7; // deklarasi tanpa keyword var
4 </script>
```

Gambar 8.4 contoh penulisan tipe data angka

8.4.2 Tipe Data String

Tipe data String di dalam **JavaScript** adalah tipe data yang terdiri dari kumpulan karakter yang berurutan. Atau di dalam penggunaan sehari-hari string adalah tipe data yang menampung nilai **text** atau kalimat.

Untuk membuat sebuah tipe data **string**, kita hanya tinggal menambahkan tanda kutip (bahasa inggris: '**quotes**') pada awal dan akhir dari text. **JavaScript** mendukung penggunaan *tanda kutip satu* (') manupun *tanda*

kutip ganda ("). Didalam sumber bahasa inggris sering disebut sebagai **single quote** dan **double quote**.

```
1 <script>
2 var nama = "mulia";
3 var kampus = 'STMIK AMIKOM Yogyakarta';
4 var pesan = 'Nama anda"';
5 var pesan2 = "Hari ini hari jum'at";
6 </script>
```

Gambar 8.5 contoh penulisan tipe data string

8.4.3 Tipe Data Boolean

Tipe data **Boolean** adalah tipe data yang hanya mempunyai dua nilai, yakni benar (**True**) atau salah (**False**). Tipe data boolean sering digunakan untuk membuat alur logika program. Struktur logika seperti **if**, **else**, **while**, dan **do while**, membutuhkan nilai **boolean** sebagai 'pengontrol' alur program.

Tipe data **boolean** juga merupakan hasil yang didapat dari *operasi perbandingan*. Misalkan apakah variabel **a** sama dengan **b**, atau apakah **a** lebih besar dari **b**.

Untuk membuat tipe data **boolean** di dalam **JavaScript**, kita cukup memberikan nilai **true**, atau **false** ke dalam sebuah **variabel**. Berikut adalah contoh pembuatan tipe data **boolean** di dalam **JavaScript**:

```
1 <script>
2 var a = true;
3 var benar = true;
4 var salah = false;
5 </script>
```

Gambar 8.6 cara membuat tipe data boolean

8.4.3 Tipe Data array

Array adalah tipe data yang digunakan untuk menyimpan berbagai nilai dalam satu nama, dibedakan melalui nilai indexnya. Array sangat populer di berbagai bahasa *programming* karena penggunaannya dapat menyederhanakan penggunaan *variable* pada *coding*.

Array pada JavaScript tentunya memiliki implikasi yang sama, namun penggunaannya tetap perlu memahami konsep dan penggunaan yang tepat. Array diperlukan sebagai object ketika digunakan di dalam JavaScript.

8.4.3.1 Membuat / Mendeklarasikan Object Array

Sebuah object array dideklarasikan di JavaScript dengan beberapa cara berikut :

1. Dengan penggunaan keyword **new Array()**, dengan contoh sebagai berikut :

```
var objArr = new Array();
```

2. Dengan penggunaan keyword **new Array(jumlahindex)**, dengan contoh sebagai berikut :

```
var objArr = new Array(3);
```

3. Dengan penggunaan keyword **new Array(item1, item2, ...)**, dimana antara satu item dengan item lainnya tidak harus memiliki tipe data yang sama.

Contoh deklarasinya adalah sebagai berikut :

```
var objArr = new Array("Sumatera", "Jawa", 100);
```

4. Dengan penggunaan literal **[item1, item2, item3, ...]**, dimana antara satu item dengan item lainnya tidak harus memiliki tipe data yang sama.

Contoh deklarasinya adalah sebagai berikut :

```
var objArr = [10, "Sarah", "Melianni", true];
```

8.4.3.2 Mengakses Array

Untuk memberikan dan membaca nilai pada objek array cukup gampang, yaitu dengan menggunakan nilai index - sebagai penunjuk posisi pada objek array. Index selalu dimulai dari angka **0**, kemudian diikuti angka **1, 2** dan seterusnya.

Penulisan objek array agar dapat diakses adalah dengan notasi penambahan angka di dalam kurung siku setelah nama objeknya dalam bentuk berikut ini :

namaobjek[NOMOR_INDEX]

Berikut adalah beberapa pemberian nilai pada array :

- `objArr[0] = 20;`
- `objArr[4] = "Budi";`

Dan berikut adalah contoh pembacaan objek array :

- `document.write(objArr[1]);`
- `var hasil = objArr[5] + objArr[6];`

8.4.3.3 Method dan Property pada Object Array

Berikut adalah beberapa method yang dapat digunakan pada variable object array :

- concat() : Menggabungkan 2 atau lebih array.
- indexOf() : Mencari suatu nilai pada array dan mengembalikan posisi indexnya.
- join() : Menggabungkan semua elemen pada array menjadi suatu teks.
- lastIndexOf() : Mencari suatu nilai pada array, dimulai dari posisi akhir dan kemudian mengembalikan posisi indexnya.
- pop() : Mengembalikan nilai elemen terakhir, dan kemudian menghilangkannya dari array.
- push() : Menambahkan nilai baru pada array, dan mengembalikan ukuran array.
- reverse() : Membalikkan posisi-posisi nilai pada array.
- shift() : Menghilangkan nilai pertama dari array, dan kemudian mengembalikan nilai tersebut.
- slice() : Mengembalikan sebagian dari array.
- sort() : Mengurutkan nilai-nilai pada array.
- splice() : Menambahkan/ menghilangkan nilai pada array.
- toString() : Melakukan dan mengembalikan konversi nilai-nilai array sebagai teks.
- unshift() : Menambahkan nilai-nilai baru pada awal array dan mengembalikan ukuran array.
- valueOf() : Mengembalikan nilai primitif dari suatu array.

Dan berikut adalah beberapa property yang dapat digunakan pada variable object array

- length : Mengembalikan ukuran array, yakni jumlah nilai / elemen yang terdapat pada array tersebut.
- constructor : Mengembalikan fungsi yang membuat prototype Array.
- prototype : Menambahkan property dan method pada objek Array.

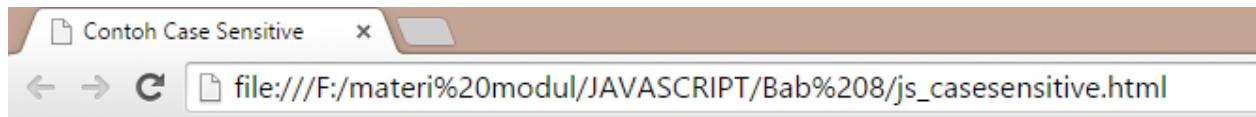
LATIHAN

1. Penulisan Case Sensitive Javascript

simpan dengan nama **js_6.html**

```
1<html>
2<head>
3<title> Contoh Case Sensitive</title>
4</head>
5<body>
6<h1> Contoh Case Sensitive</h1>
7<script type="text/javascript" language="javascript">
8    document.write("Contoh ini menunjukkan sensitive JavaScript")
9    DOCument.write("Contoh ini menunjukkan sensitive JavaScript")
10</script>
11</body>
12</html>
```

Gambar 8.7 Contoh Script penulisan case sensitive JavaScript



Contoh Case Sensitive

Contoh ini menunjukkan sensitive JavaScript

Gambar 8.8 Hasil Running Script penulisan case sensitive JavaScript

2. Penggunaan tipe data angka dan operasi matematis di dalam JavaScript

Simpan dengan nama **js_7.html**

```
1  <!DOCTYPE html>
2  <html>
3  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4  <head><title>Operasi Aritmatika</title></head>
5  <body>
6  <p><script language="JavaScript">
7  <!--
8  document.writeln("<pre>");
9  document.writeln("<h1>Operasi Aritmatik</h1>");
10 var A = 100;
11 var B = 200;
12 var C = 300;
13 var D = 400;
14 var E = A + B;
15 var F = B * C;
16 var G = A - B;
17 document.writeln('100 + 200 = ' + E);
18 //E = B + C;
19 document.writeln('200 * 300 = ' + F);
20 //F = B * C;
21 document.writeln('300 - 400 = ' + G);
22 document.writeln("<pre>");
23 //-->
24 </script></p>
25 </body>
26 </html>
```

Gambar 8.9 contoh penulisan tipe data angka dan operasi matematis di dalam JavaScript



Operasi Aritmatik

```
100 + 200 = 300
200 * 300 = 60000
300 - 400 = -100
```

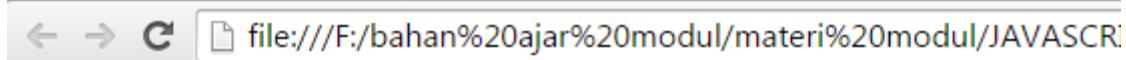
Gambar 8.10 Hasil running script tipe data angka dan operasi matematis di javascript

3. Penggunaan String di dalam JavaScript

simpan dengan nama **js_8.html**

```
1  <!DOCTYPE html>
2  <html>
3  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4  <head><title>Menambahkan String</title></head>
5  <body>
6  <p><script language="JavaScript">
7  <!--
8  document.writeln("<pre>");
9  document.writeln("<h1>Menambahkan String</h1>");
10 var A = 'mulia ';
11 var B = 'sulistiyono';
12 var C = "bernad ";
13 var D = "bernadhed";
14 var E = A + B;
15 var F = C + D;
16 document.writeln('mulia + sulistiyono = ' + E);
17 document.writeln('bernad + bernadhed = ' + F);
18 document.writeln("<pre>");
19 //-->
20 </script></p>
21 </body>
22 </html>
```

Gambar 8.11 penggunaan string dalam javascript



Menambahkan String

```
mulia + sulistiyono = mulia sulistiyono
bernad + bernadhed = bernad bernadhed
```

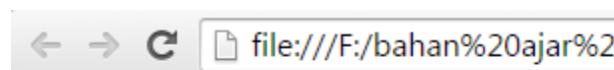
Gambar 8.12 Hasil running script tipe data string di javascript

4. Penggunaan Boolean di dalam JavaScript

simpan dengan nama **js_9.html**

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <p>Menampilkan nilai Boolean(10 > 9):</p>
6
7  <button onclick="myFunction()">Cobalah</button>
8
9  <p id="demo"></p>
10
11 <script>
12 function myFunction() {
13     document.getElementById("demo").innerHTML = Boolean(10 > 9);
14 }
15 </script>
16
17 </body>
18 </html>
```

Gambar 8.13 Penggunaan Boolean di dalam javascript



Menampilkan nilai Boolean(10 > 9):

Cobalah

true

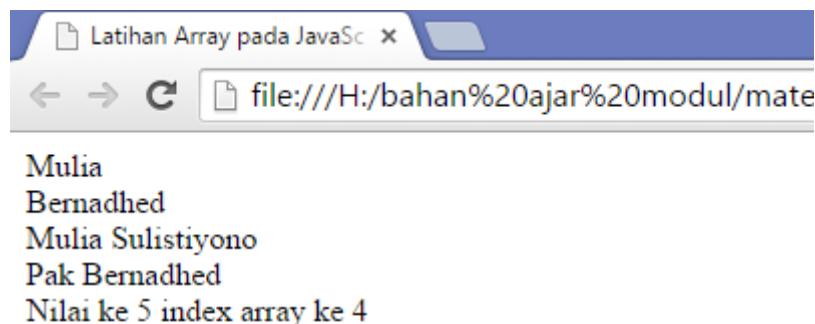
Gambar 8.14 Hasil running script tipe data boolean di javascript

5. Penggunaan Array pada Javascript

simpan dengan nama **js_10.html**

```
1<html>
2<head>
3<title>Latihan Array pada JavaScript</title>
4<script type="text/javascript">
5var array1 = ["Mulia", "Bernadhed", "Mulia Sulistiyo", "Pak Bernadhed"] ;
6document.writeln(array1[0]); document.writeln("<br>");
7document.writeln(array1[1]); document.writeln("<br>");
8document.writeln(array1[2]); document.writeln("<br>");
9document.writeln(array1[3]); document.writeln("<br>");
10array1[4] = "Nilai ke 5 index array ke 4" ;
11document.writeln(array1[4]); document.writeln("<br>");
12</script>
13<noscript>
14Teks ini muncul jika <i>browser</i> tidak dapat menangani JavaScript
15</noscript>
16</head>
17<body>
18</body>
19</html>
```

Gambar 8.15 Penggunaan Array di dalam javascript



Gambar 8.16 Hasil running script tipe data Array di javascript

Bab 9. Percabangan dan Perulangan

Tujuan :

1. Mahasiswa mampu menjelaskan konsep percabangan dan perulangan dalam bahasa pemrograman JavaScript
2. Mahasiswa mampu menjelaskan cara kerja percabangan dan perulangan JavaScript
3. Mahasiswa mengetahui dan menjelaskan konsep pemrograman dinamis pada JavaScript

Overview

Untuk membuat suatu halaman yang dinamis dan interaktif, perancang halaman Web membutuhkan perintah-perintah yang dapat mengatur aliran dari informasi. Berdasarkan hasil komputasi yang telah dilakukan, JavaScript akan membuat keputusan jalur mana yang akan dieksekusi.

9.1 Percabangan

Pada dasarnya dalam JavaScript terdapat dua macam pernyataan percabangan yaitu if..else dan switch

9.1.1 If..Else

Pernyataan ini digunakan untuk menguji sebuah kondisi dan kemudian mengeksekusi pernyataan tertentu bila kondisi tersebut terpenuhi, dan mengeksekusi pernyataan lain bila kondisi tersebut tidak terpenuhi.

```
if (kondisi)
{
    //pernyataan1 dieksekusi
    //bila kondisi terpenuhi
}
else
{
    //pernyataan2 dieksekusi
    //bila kondisi tidak terpenuhi
}
```

kondisi adalah ekspresi JavaScript yang mana hasil evaluasinya memiliki nilai Boolean true atau false. Untuk kasus yang melibatkan lebih banyak kondisi, maka kita dapat meletakkan pernyataan if lain setelah else

```
if (kondisi1)
{
    //pernyataan1 dieksekusi
    //bila kondisi1 terpenuhi
}
else if (kondisi2)
{
    //pernyataan2 dieksekusi
    //bila kodisi1 tidak terpenuhi
}
else
{
    //pernyataan3 dieksekusi
    //bila kodisi2 tidak terpenuhi
}
```

9.1.2 Percabangan Majemuk

Percabangan majemuk adalah suatu percabangan yang dapat melibatkan lebih dari 1 kondisi di dalam percabangannya. Biasanya percabangan seperti ini menggunakan operator tambahan seperti AND, OR dan sebagainya.

9.1.3 Switch

Selain menggunakan if..else, percabangan juga dapat ditangani dengan perintah switch. Dengan kata lain pernyataan switch digunakan untuk menyederhanakan pernyataan if..else yang terlalu banyak.

9.2 Perulangan

Untuk mengulang kejadian beberapa kali maka kita membutuhkan proses perulangan. Pada JavaScript dikenal beberapa metode/cara perulangan.

9.2.1 Perulangan For

Digunakan untuk mengeksekusi pernyataan-pernyataan beberapa kali. Perulangan For paling sering dipakai, jika anda sudah tahu akhir dari perulangan tersebut. . Perintah for mengulang suatu loop sampai kondisi menghasilkan evaluasi true atau loop keluar dengan perintah break. Contoh :

```
for (nilai awal;kondisi;penambahan)
{
    ulang pernyataan ini;
}
```

Contoh dalam program :

```
For (x=1;x<=10;x++)  
{  
    document.writeln("Belajar JavaScript Yuuuuu..");  
}
```

9.2.2 Perulangan While

Perulangan lain yang dapat digunakan adalah dengan menggunakan perintah While.

Perintah while digunakan untuk perulangan yang tidak diketahui berapa kali proses perulangannya. Perintah while terus mengulangi loop selama kondisi memiliki nilai true. Syntax untuk perintah while adalah sebagai berikut :

```
while (kondisi)  
{  
    ulang pernyataan ini;  
}
```

9.2.3 Perulangan Do While

Perulangan ini hampir sama seperti while, digunakan apabila kita belum tahu berapa kali perulangan harus dilakukan. Bedanya pernyataan do..while pengujiannya dilakukan di akhir pernyataan.

```
Do  
{  
    //pernyataan1 dieksekusi  
}  
while (kondisi);
```

LATIHAN

1. Program Percabangan menggunakan if..else

Simpan dengan nama **js_11.html**

```
1 <html>
2 <head><title>Percabangan IF-ELSE</title></head>
3 <body>
4 <script language="JavaScript">
5 <!--
6 function tanyabilangan()
7 {
8     var bil = parseFloat(document.fform.bilangan.value);
9     var jenis = " ";
10    if(isNaN(bil))
11    {
12        alert("Anda Belum memasukkan Bilangan");
13    }
14    else
15    {
16        if (bil > 0)
17        {
18            jenis = " Adalah bilangan Positif";
19        }
20        else if (bil < 0)
21        {
22            jenis = " Adalah bilangan Negatif";
23        }
24        else
25        {
26            jenis = " Adalah Nol";
27        }
28        alert (bil+" "+jenis);
29    }
30 }
//--></SCRIPT>
32
33
34 <form name ="fform">
35 <h2><break>Bilangan Negatif, Nol, Atau Positif ???</h2>
36 Masukkan Bilangan :<input type="text" size="11" name="bilangan">
37 <p>
38 <input type="button" value="Tanya" onclick="tanyabilangan()">
39 <input type="reset" value="Ulang"> </p>
40 </form>
41 </body>
42 </html>
```

Gambar 9.1 Contoh Program Percabangan menggunakan if..else



Gambar 9.2 Hasil Running Script Program Percabangan menggunakan if..else

2. program percabangan majemuk

simpan dengan nama **js_12.html**

```
1 <html>
2 <head><title>Percabangan IF-ELSE majemuk</title></head>
3 <body>
4 <script language="JavaScript">
5 <!--
6 function hitungip()
7 {
8     var quis = parseFloat(document.fform.iquis.value);
9     var tugas = parseFloat(document.fform.itugas.value);
10    var uts = parseFloat(document.fform.iuts.value);
11    var uas = parseFloat(document.fform.iuas.value);
12    var ip = " ";
13    var ket = " ";
14    var na = (0.10*quis)+(0.20*tugas)+(0.30*uts)+(0.40*uas);
15    if ((na > 80) && (na <=100))
16    {
17        ip = "A";
18        ket = "Lulus dengan Sangat Baik";
19    }
20    else if ((na > 68) && (na <=80))
21    {
22        ip = "B";
23        ket = "Lulus dengan Baik";
24    }
25    else if ((na > 55) && (na <=68))
26    {
27        ip = "C";
28        ket = "Lulus dengan Cukup";
29    }
30    else if ((na > 38) && (na <=55))
31    {
```

```
31 ip ="D";
32 ket="Lulus dengan Kurang";
33 }
34 else
35 {
36 ip ="E";
37 ket="Tidak Lulus";
38 }
39 document.form.oip.value=ip;
40 document.form.oket.value=ket;
41 // gunakan untuk mengecek alert (ip""+na);
42 }
43 //--></SCRIPT>
44 <form name ="fform">
45 <table border="1" width="100%" align="center" >
46   <tr>
47     <td width="100%" colspan="4"><h2 align="center">Menghitung Indeks Prestasi</h2></td>
48   </tr>
49   <tr>
50     <td width="25%">Quis (10%) :<input type="text" size="10" name="iquis"> </td>
51     <td width="25%">Tugas (20%) :<input type="text" size="10" name="itugas"> </td>
52     <td width="25%">UTS (30%) :<input type="text" size="10" name="iuts"> </td>
53     <td width="25%">UAS (40%) :<input type="text" size="10" name="iuas"> </td>
54   </tr>
55   <tr>
56     <td width="100%" colspan="4">
57       <p Align="center">
58         <input type="button" value="Hitung" onclick="hitungip()">
59         <input type="reset" value="Ulang"> </p></td>
60       </tr>
61     <tr>
62       <td width="100%" colspan="4" align="center">
63         Indeks Prestasi :<input type="text" size="5" name="oip">
64         Keterangan :<input type="text" size="25" name="oket"></td>
65     </tr>
66   </table>
67 </form>
68 </body>
69 </html>
```

Gambar 9.3 Contoh Program Percabangan majemuk



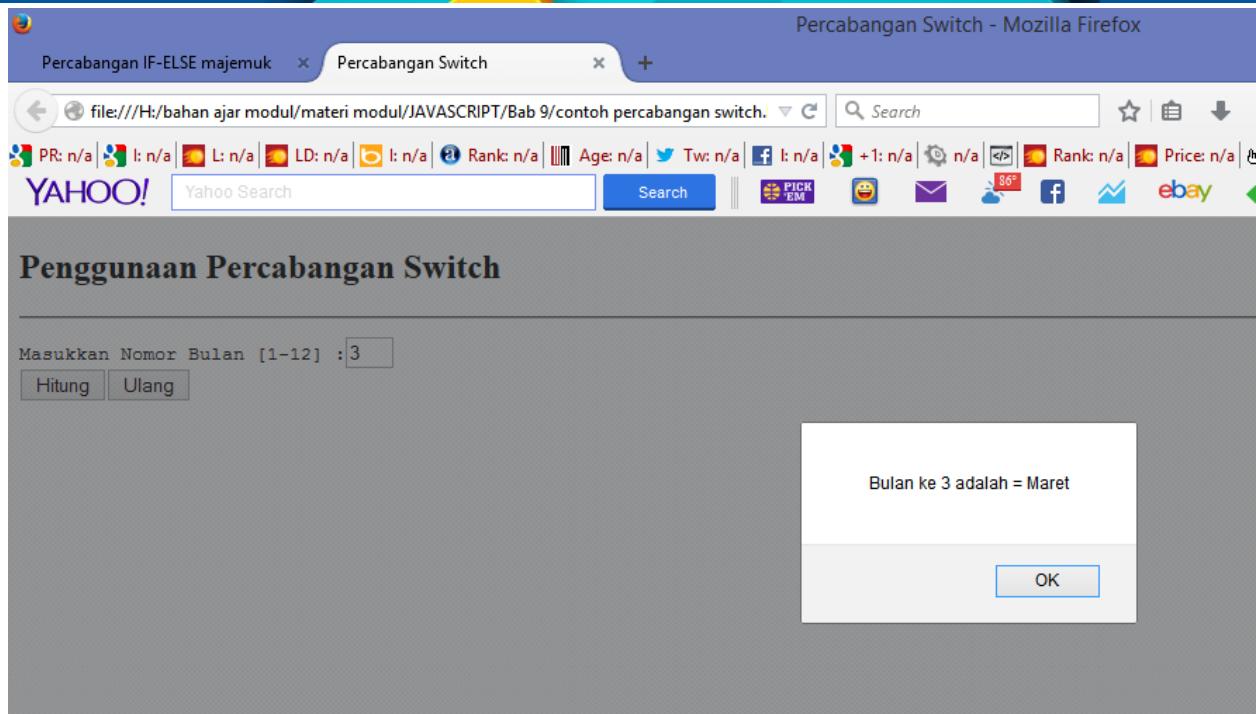
Gambar 9.4 Hasil Running Script Program Percabangan majemuk

3. program percabangan switch dengan javascript

simpan dengan nama **js_13.html**

```
1<html>
2 <head><title>Percabangan Switch</title></head>
3 <body>
4 <script language="JavaScript">
5 <!--
6 function tanyabulan()
7 {
8     var bulan = parseFloat(document.fform.ibulan.value);
9     var namabulan="";
10    switch (bulan)
11    {
12        case 1 : namabulan="Bulan ke 1 adalah = Januari";break;
13        case 2 : namabulan="Bulan ke 2 adalah = Februari";break;
14        case 3 : namabulan="Bulan ke 3 adalah = Maret";break;
15        case 4 : namabulan="Bulan ke 4 adalah = April";break;
16        case 5 : namabulan="Bulan ke 5 adalah = Mei";break;
17        case 6 : namabulan="Bulan ke 6 adalah = Juni";break;
18        case 7 : namabulan="Bulan ke 7 adalah = Juli";break;
19        case 8 : namabulan="Bulan ke 8 adalah = Agustus";break;
20        case 9 : namabulan="Bulan ke 9 adalah = September";break;
21        case 10 : namabulan="Bulan ke 10 adalah = Oktober";break;
22        case 11 : namabulan="Bulan ke 11 adalah = November";break;
23        case 12 : namabulan="Bulan ke 12 adalah = Desember";break;
24        default : namabulan="Anda salah mengisi";
25    }
26    alert(namabulan);
27 }
28 </script>
29 <form name ="fform">
30 <h2>Penggunaan Percabangan Switch</h2><hr>
31 <pre>
32 Masukkan Nomor Bulan [1-12] :<input type="text" size="2" name="ibulan">
33 <input type="button" value="Hitung" onclick="tanyabulan()"><input type="reset" value="Ulang">
34 </pre>
35 </form>
36 </body>
37 </html>
```

Gambar 9.5 Program Percabangan menggunakan switch



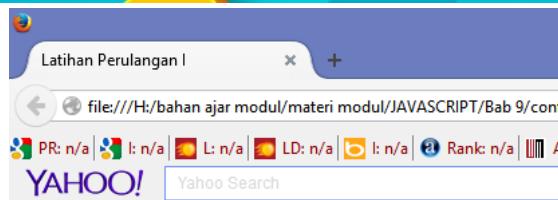
Gambar 9.6 Hasil Runnig Script Program Percabangan menggunakan switch

4. program perulangan menggunakan for

simpan dengan nama **js_14.html**

```
1<html>
2 <head><title>Latihan Perulangan I</title></head>
3 <script language="JavaScript">
4 <!-- menyembunyikan script dan browser non-JS
5 document.writeln("<H2>Latihan Perulangan For</H2>");
6 document.writeln("-----");
7 for (i=1; i<=10; i++)
8 {
9   sq=i*i;
10  document.writeln("<PRE>");
11  document.write("Angka:" + i + " -----> Kuadrat: " + sq + "<BR>");
12  document.writeln("</PRE>");
13 }
14 document.writeln("-----");
15 // akhir dari penyembunyian -->
16 </script>
17 </head>
18 <body>
19 </body>
20 </html>
```

Gambar 9.7 Program Percabangan menggunakan for



Latihan Perulangan For

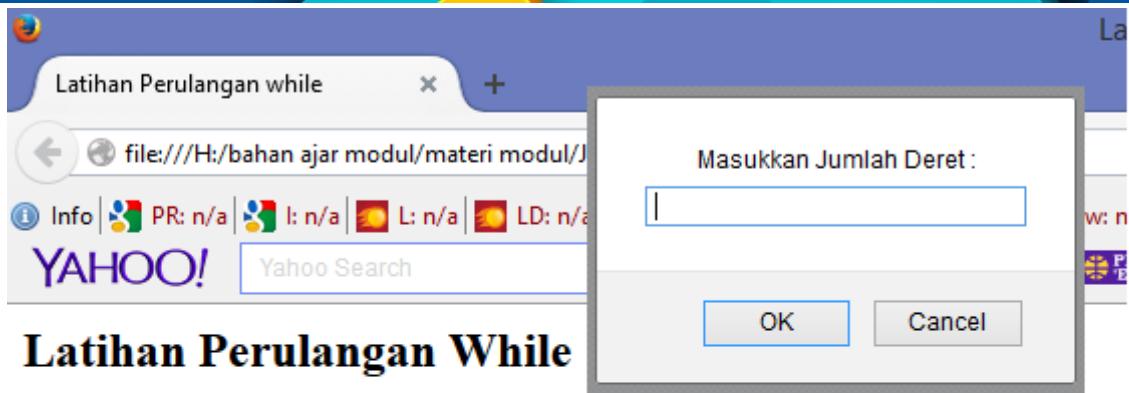
Gambar 9.8 Hasil Runnig Script Program Perulangan menggunakan for

5. program perulangan menggunakan while

simpan dengan nama **js_15.html**

```
1<html>
2<head>
3<title>Latihan Perulangan while</title>
4</head>
5<script language="JavaScript">
6<!-- menyembunyikan script dan browser non-JS
7var deret = prompt('Masukkan Jumlah Deret :','');
8document.writeln("&lt;h2&gt;Latihan Perulangan While&lt;/h2&gt;");
9document.writeln("-----");
10document.writeln("&lt;break&gt;");
11var jml = 0.0;
12var angka = 1;
13while (angka &lt;= deret)
14{
15    jml= jml+angka;
16    angka++;
17}
18document.writeln("&lt;break&gt;");
19document.writeln("Jumlah Deret dari 1 sampai "+deret+" adalah = "+jml+"");
20document.writeln("&lt;break&gt;");
21document.writeln("-----");
22// akhir dari penyembunyian --&gt;
23&lt;/script&gt;
24&lt;/head&gt;
25&lt;/html&gt;</pre>
```

Gambar 9.11 Program Perulangan menggunakan while



----- Jumlah Deret dari 1 sampai 50 adalah = 1275 -----

Gambar 9.12 Hasil Runnig Script Program Perulangan menggunakan while

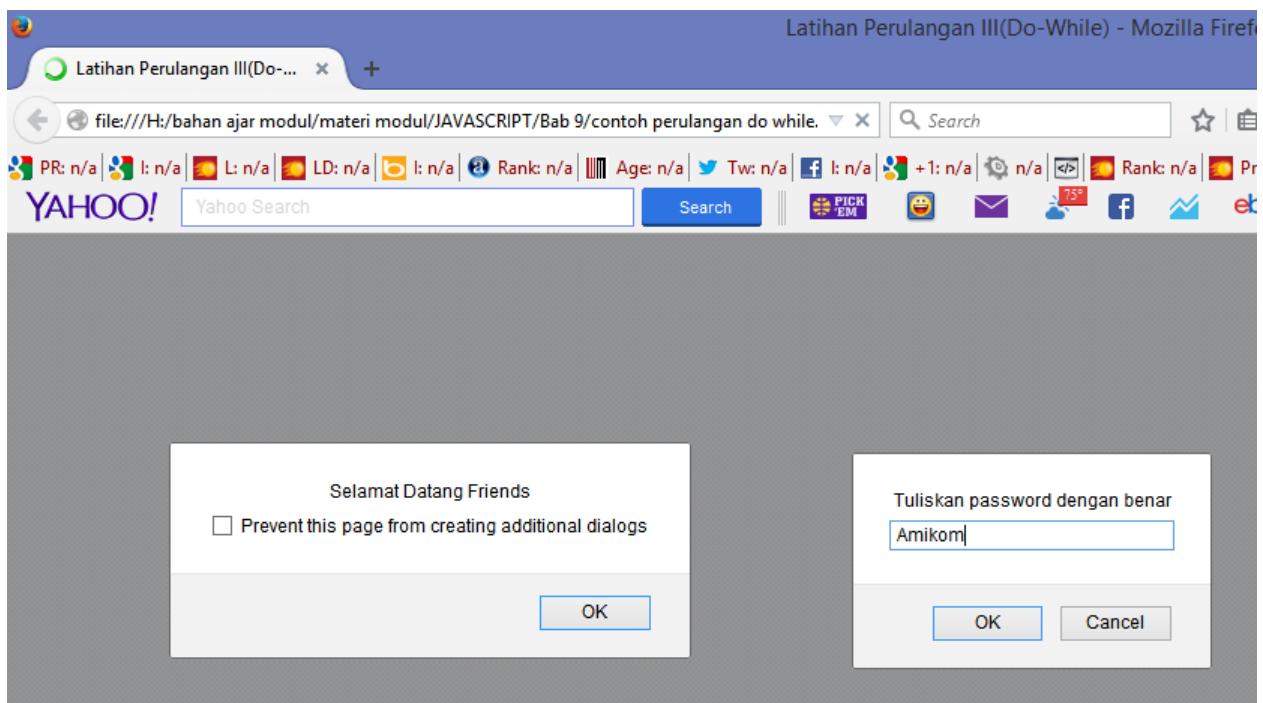
6. program perulangan menggunakan do while

simpan dengan nama **js_16.html**

```
1<html>
2 <head><title>Latihan Perulangan III(Do-While)</title></head>
3 <body>
4 <script language="JavaScript"><!-- menyembunyikan script dan browser nonfunction pass()
5 &lt;
6 var coba = 1;
7 do
8 {
9     p = prompt("Tuliskan password dengan benar","");
10    if (p=="Amikom")
11    {
12        alert("Selamat Datang Friends");
13        window.open("contoh perulangan for.html");
14        break;
15    }
16    else
17    {
18        alert("Password Salah !!! Ulangi lagi.");
19    }
20    if (coba==3)
21    {
22        alert("maaf, kesempatan anda hanya 3 kali");
23        history.go(-1);
24    }
25    coba=coba+1;</pre>
```

```
26    }
27    while (coba<=3)
28    }
29 // akhir dari penyembunyian -->
30 </script>
31 <form method="post">
32 <input type="button" value="Password" onclick="pass () ">
33 </form>
34 </body>
35 </html>
```

Gambar 9.11 Program Perulangan menggunakan do while



Gambar 9.12 Hasil Running Script Program Perulangan menggunakan do while

STUDI KASUS

Bab 9. Pengenalan PHP

Tujuan :

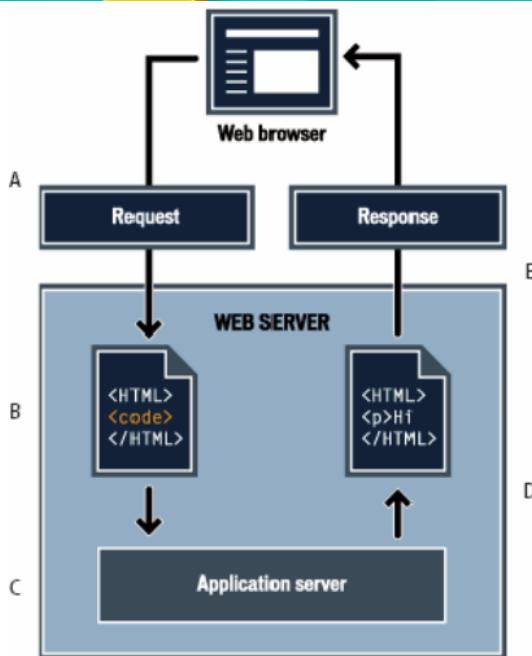
1. Mahasiswa mampu mengetahui konsep dasar server side programming menggunakan bahasa pemrograman PHP
2. Mahasiswa mampu mengetahui contoh penerapan PHP
3. Mahasiswa mengetahui dan mampu menjelaskan cara kerja PHP.

Overview

PHP (*PHP : Hypertext Preprocessor*) adalah sebuah bahasa *scripting* yang paling popular saat ini. Berfungsi untuk membuat website dengan konten dinamis maupun aplikasi web lainnya. Berbeda dengan HTML yang hanya menampilkan konten statis, PHP dapat berinteraksi dengan database, file, folder bahkan dengan lingkungan sistem operasi. PHP termasuk kedalam golongan perangkat lunak interpreter. Yakni menerjemahkan baris demi baris program (tanpa mengcompilennya) kemudian mengeksekusinya. PHP awal mula dikembangkan oleh Rasmus Lerdorf sebagai pengembangan dari Perl untuk menangani pengolahan form pada website pribadinya. Oleh karenanya, saat itu PHP disebut juga sebagai akronim dari *Personal Home Page*. Saat ini lebih dari 20 juta website menggunakan PHP *backend* aplikasi web.

10.1 Skema Aplikasi Web

Aplikasi web dinamis berbasiskan pada konsep arsitektur *client - server*, dimana server menangani apa yang dimininta oleh *client*, kemudian mengirimkan kembali hasilnya dalam bentuk HTML.



Gambar 10.1 Skema Aplikasi Web

Skema aplikasi web pada gambar diatas dapat dijelaskan sebagai berikut:

- a. Client (web browser) meminta sebuah halaman pada sebuah website.
- b. Web server mencari halaman yang diminta, dan setelah ditemukan, memberikan halaman tersebut ke application server (PHP Interpreter).
- c. Application server mem-parse (membaca) dan memproses halaman tersebut.
- d. Application server mengirimkan halaman yang telah diproses kembali ke web server.
- e. Web server mengirimkan kembali halaman yang telah diproses kepada web browser dalam bentuk HTML.

10.2 Dasar PHP

10.2.1 Aturan Penulisan

Sebagaimana umumnya bahasa pemrograman, PHP mempunyai aturan-aturan tertentu dalam penulisan skripnya. Diantaranya sebagai berikut:

- a. Program disimpan dengan ekstensi .php, .php3, .php4, atau .php5.
- b. Semua skrip harus diapit oleh tanda:
 - <?php dan ?>
 - <script language="php"> dan </script>
 - <? dan ?>

- <% dan %>
- c. Setiap perintah harus diakhiri dengan tanda titik koma (;).

Berbeda dengan bahasa banyak bahasa pemrograman web pada umumnya, skrip program PHP dapat langsung ditanamkan / disisipkan didalam HTML. Jika skrip PHP ditemukan dalam sebuah dokumen web, PHP Interpreter akan memproses skrip PHP dan hanya akan mengirimkan hasil berupa HTML ke browser client.

Dalam sebuah program, terdapat dua hal penting: *command* (perintah) dan *values* (nilai-nilai). *Command* memiliki tugas spesifik yang telah ditentukan. Sedangkan *values* adalah informasi yang berkaitan dengan tugas yang dilakukan oleh *command*. Dari contoh-contoh sebelumnya, kita telah banyak menggunakan *literal*. Yaitu *value* yang telah diterapkan secara eksplisit didalam program pada sebuah *command*. Sebagai contoh baris program berikut:

```
echo "Hello World!";
```

Pernyataan diatas melakukan *command echo* dan menggunakan *value literal* yang diberikan setelahnya, yaitu "Hello World!".

10.2.2 Mencetak String

Dalam PHP, string adalah rangkaian karakter, seperti "Once upon a time" atau "Doremifasolasido" bahkan "111211211". Ketika PHP membaca dan memproses sebuah

kode program dan mengirimkan output ke web browser, setiap data yang ditampilkan akan dianggap sebagai sebuah string. Terdapat dua fungsi yang umum digunakan untuk mencetak sebuah string. Yaitu print dan echo. Untuk mencetak sebuah string digunakan sepasang karakter *single quoted* (') atau *double quoted* (") yang mengapit string tersebut.

10.2.3 Karakter Escape

Karakter escape adalah bagian dari syntax bahasa pemrograman. Biasa dimulai dengan karakter backslash (\) dan diikuti oleh metakarakter tertentu yang disebut sebagai *escape sequences*.

Escape sequences memiliki 2 fungsi :

- a. Mengkodekan entitas syntax, yaitu perintah untuk menampilkan data khusus yang tidak dapat langsung diwakili oleh alfabet.
- b. Pengutip karakter, yaitu untuk mewakili karakter yang tidak dapat diketik dalam konteks saat ini, atau akan memiliki interpretasi yang tidak diinginkan.

Escape sequences yang bekerja pada *double quoted* (") terdapat dalam table 10.1 sebagai berikut:

Table 10.1 *Escape Sequences* pada *double quoted*

Escape sequences	Representasi Karakter
\n	Ganti baris
\r	Carriage return
\t	Tab
\\"	Backslash
\\$	Tanda dolar
\\"	Double quote
\{	Left brace
\}	Right brace
\[Left bracket
\]	Right bracket
\0 atau \777	Octal value
\x0 atau \xff	Hexa value

Terdapat dua *escape sequences* yang dapat bekerja pada *single quoted* adalah '\, yang menempatkan tanda kutip tunggal dalam string yang dikutip tunggal, dan \\, yang menempatkan sebuah backslash dalam string *single quoted*.

10.2.4 Menambah Komentar

Selain digunakan untuk memberikan keterangan mengenai kode program kita, komentar juga memberikan informasi kepada orang-orang yang ingin memahami kode program kita. Baris yang diberi komentar akan diabaikan (tidak diproses) oleh PHP.

PHP menyediakan beberapa cara untuk mendefinisikan sebuah komentar. Dimana cara penulisan ini diinspirasi oleh bahasa program yang telah ada terlebih dahulu. Seperti C, C++, and the Unix shell.

a. Unix Shell Style

Ketika PHP Interpreter menemukan tanda *hash* (#) pada kode program, segala sesuatu dari tanda *hash* ke akhir baris atau akhir dari bagian kode PHP akan dianggap sebagai komentar.

b. C++ Style

Ketika PHP Interpreter menemukan tanda *double slash* (//) pada kode program, segala sesuatu dari tanda *double slash* ke akhir baris atau akhir dari bagian kode PHP akan dianggap sebagai komentar. Hasil dari komentar C++ style ini akan sama dengan Unix Shell style.

c. C Language Style

Penulisan komentar menggunakan Unix Shell style dan C++ style sangat berguna untuk membuat komentar singkat. PHP mendukung komentar blok, yang diinspirasi oleh bahasa C. Ketika PHP menemukan sebuah garis miring diikuti dengan tanda bintang (/*), segala sesuatu sampai ditemukannya tanda bintang diikuti garis miring, akan dianggap sebagai sebuah komentar. Ini berarti C Language Style mendukung komentar dengan banyak baris sekaligus.

10.3 Variabel

Variable adalah sebuah pengenal yang digunakan untuk menyimpan suatu nilai yang nilai tersebut dapat berubah-ubah selama dijalankannya program.

10.3.1 Deklarasi Variabel

Untuk mendefinisikan sebuah variabel, diawali dengan karakter dolar (\$) dan diikuti dengan **nama pengenal**. Adapun aturan-aturan yang berkaitan dengan variabel adalah sebagai berikut:

- Terdiri dari huruf, angka dan karakter garis bawah (_) atau kombinasi antara ketiganya.
- Tidak boleh mengandung spasi dan terdiri dari minimal satu karakter.
- Dimulai dengan huruf atau garis bawah, tidak boleh diawali dengan angka.
- Bersifat *case sensitive*, yaitu huruf besar dan kecil dibedakan.

Seperti yang telah kita ketahui, sebuah variabel akan bernilai valid jika telah memenuhi persyaratan yang telah ditentukan diatas. Namun, untuk memudahkan memahami alur program kita, cobalah untuk mengikuti cara penulisan variabel seperti di bawah ini:

- a. Nama variabel mencerminkan isinya. Sebagai contoh, jangan gunakan variable \$x untuk menyimpan total harga, melainkan gunakan \$total_harga. Jika suatu saat kita ingin memperbaiki kode program beberapa bulan setelah kita pertama kali menulis program itu, kita bingung dengan variabel bernama \$x karena ia tidak mencerminkan apapun dari isinya. Berbeda dengan nama variabel \$total_harga yang dengan jelas mencerminkan isinya.
- b. Sebisa mungkin variabel harus singkat. Mendefinisikan variabel terlalu panjang kadang-kadang terlalu merepotkan. Cobalah untuk mempersingkat, namun tetap masuk akal.

10.3.2 Penugasan Variabel

Terdapat dua cara dalam penugasan variabel: dari skrip itu sendiri atau ditentukan oleh PHP Interpreter. Saat ini kita hanya fokus ada penugasan yang dilakukan dari skrip. Seperti yang telah kita pelajari, penugasan terjadi ketika sebuah nilai diberikan pada sebuah variabel. Perhatikan dua pernyataan sebagai berikut:

```
$int_tiga = 3;  
3 = $int_tiga;
```

Aturan penugasan variabel dapat dimulai dari kanan ke kiri, yang dapat diterjemahkan sebagai berikut. Untuk pernyataan pertama, dapat dibaca "bilangan 3 ditugaskan ke variabel \$int_tiga". Kemudian kita bandingkan dengan pernyataan ke dua: "variabel \$int_tiga ditugaskan ke bilangan 3". Pernyataan tidak masuk akal dan tidak akan bekerja, dan jika menerapkan pernyataan tersebut pada program, PHP akan berhenti dengan pesan error.

Tidak seperti bahasa pemrograman lainnya, tidak diperlukan deklarasi tipe data untuk sebuah variabel. Setiap variabel terbentuk dalam tipe data variant, yang artinya ia dapat menampung jenis data apapun.

Meskipun variabel pada PHP bersifat variant, kita dapat mengetahui tipe data yang dikandung pada sebuah variabel menggunakan fungsi `gettype()`.

10.3.3 Type Casting

Type casting adalah mekanisme pengubahan / konversi sebuah variabel kepada tipe data tertentu. Sebagai contoh, kadangkala kita hanya ingin menggunakan bagian integer dari sebuah bilangan pecahan. Untuk melakukan konversi, gunakan format cara berikut:

(tipe) \$variabel

Type casting dapat diterapkan pada sebuah variabel dengan tipe data sebagai berikut:

- (int) atau (integer)
- (real), (double) atau (float)
- (string)
- (array)
- (object)

10.4 Konstanta

Konstanta adalah sebuah tetapan nilai dalam sebuah program. Nilai konstanta tidak dapat diubah selama program dijalankan dan jika dilakukan hanya akan menyebabkan *error*. Deklarasi konstanta dapat dilakukan dengan cara sebagai berikut:

```
define("CONTOH_KONSTANTA", 5);
```

Perhatikan bahwa dalam pernyataan diatas, `CONTOH_KONSTANTA` tidak diawali dengan karakter dolar (\$), karena karakter dolar hanya digunakan untuk sebuah variabel.

Sebagaimana variabel, konstanta juga yang memiliki aturan yang harus diikuti. Aturan berikut memastikan kita untuk mendeklarasikan konstanta dengan benar:

- a. Tidak boleh didahului oleh karakter dolar.
- b. Nama konstanta harus dimulai dengan huruf atau garis bawah (_).
- c. Konstanta tidak dapat dideklarasikan ulang dan nilainya tetap.

Sebagaimana variabel, dibutuhkan aturan-aturan khusus untuk memudahkan memahami alur program kita, cobalah untuk mengikuti cara penulisan konstanta seperti dibawah ini:

- a. Untuk kenyamanan, se bisa mungkin gunakan nama konstanta yang singkat. Karena pada umumnya konstanta digunakan untuk pilihan pengaturan.
- b. Selalu gunakan huruf besar. Ini membantu kita membedakan konstanta dengan pernyataan lain.
- c. Pisahkan setiap suku kata dalam konstanta dengan garis bawah (_). Garis bawah diperlukan untuk membantu kita membedakan satu konstanta dengan yang lainnya.

10.5 Operator

Operator digunakan untuk memanipulasi nilai sebuah variabel. Sedangkan nilai yang dimodifikasi oleh *operator* disebut *operand*. Sedangkan kombinasi dari keduanya disebut *ekspresi*. Dalam pemrograman operator dapat dikelompokkan menjadi:

- Operator Aritmatika
- Operator Penugasan
- Operator Perbandingan
- Operator Logika
- Lain-lain

10.5.1 Operator Aritmatika

Operator aritmatika digunakan untuk melakukan perhitungan matematika.

Table 10.2 Operator Aritmatika

Operator	Nama Operasi	Contoh
+ (Unary)	Positif	+5
- (Unary)	Negatif	-5
+	Penjumlahan	5 + 5
-	Pengurangan	5 - 5
*	Perkalian	5 * 5
/	Pembagian	5 / 5
%	Hasil Bagi	5 % 5

10.5.2 Operator Relasional

Merupakan operator yang digunakan untuk membandingkan nilai dari dua operand. Hasil perbandingannya dinyatakan dalam nilai boolean. TRUE untuk pernyataan bernilai benar, dan FALSE untuk operator bernilai salah.

Table 10.3 Operator Relasional

Operator	Keterangan
==	Memeriksa apakah operand kiri bernilai sama dengan operand kanan
>	Memeriksa apakah operand kiri lebih besar dari operand kanan
<	Memeriksa apakah operand kiri lebih kecil dari operand kanan
>=	Memeriksa apakah operand kiri lebih besar atau sama dengan operand kiri
<=	Memeriksa apakah operand kiri lebih kecil atau sama dengan operand kiri
!=	Memeriksa apakah operand kiri tidak bernilai sama dengan operand kanan

10.5.3 Operator Logika

Operator logika digunakan untuk membandingkan dua atau lebih nilai variabel yang bertipe boolean. Hasil yang didapat dari penggunaan operator logika adalah boolean.

Terdapat beberapa jenis operator logika, diantaranya sebagai berikut:

- Logika AND (Disjungsi)
- Logika OR (Konjungsi)
- Logika XOR

- Logika NOT (Negasi)

10.5.3.1 Logika AND (Konjungsi)

Logika AND akan bernilai TRUE jika pernyataan-pernyataan yang menjadi prasyaratnya bernilai TRUE. Dalam PHP, operator AND disimbolkan dengan tanda **&&**. Dibawah ini adalah table 10.4 kebenaran untuk logika AND.

Table 10.4 logika AND

P	Q	P ^ Q
T	T	T
T	F	F
F	T	F
F	F	F

10.5.3.2 Logika OR (Disjungsi)

Logika OR akan bernilai TRUE jika salah satu pernyataan yang menjadi prasyarat bernilai TRUE. Dalam PHP, operator OR disimbolkan dengan tanda **||**. Dibawah ini adalah table 10.5 kebenaran untuk logika OR.

Table 10.5 logika OR

P	Q	P v Q
T	T	T
T	F	T
F	T	T
F	F	F

10.5.3.3 Logika XOR (Exclusive OR)

Logika XOR akan bernilai TRUE jika pernyataan yang menjadi prasyaratnya mempunyai nilai kebenaran yang berbeda. Dalam PHP, operator XOR disimbolkan dengan tanda **XOR**. Dibawah ini adalah table 10.6 kebenaran untuk logika XOR.

Table 10.6 logika XOR

P	Q	P v Q
T	T	F
T	F	T
F	T	T
F	F	F

10.5.3.4 Logika NOT (Negasi)

Logika NOT adalah nilai kebalikan dari pernyataan sebenarnya. Dalam PHP, operator NOT disimbolkan dengan tanda seru (!). Dibawah ini adalah tabel 10.7 kebenaran untuk logika NOT.

Table 10.7 logika NOT

P	$\neg P$
T	F
F	T

10.5.4 Operator Penugasan

Operator penugasan digunakan untuk memberikan nilai pada sebuah variabel.

Contoh:

```
$nama = "MuliaBernad";
```

Pada contoh diatas, operator penugasan "=" digunakan untuk memasukkan string "MuliaBernad" kedalam variabel \$nama. Selain operator penugasan "=", juga dikenal beberapa operator penugasan lain. Diataranya adalah sebagai berikut:

- Operator +=

Digunakan untuk menambah variabel dengan bilangan tertentu. Contoh:

`$a += 5;` akan sama artinya dengan `$a = $a + 5;`

- Operator -=

Digunakan untuk mengurangi variabel dengan bilangan tertentu . Contoh:

`$a -= 5;` akan sama artinya dengan `$a = $a - 5;`

- Operator *=

Digunakan untuk mengalikan variabel dengan bilangan tertentu. Contoh:

`$a *= 5;` akan sama artinya dengan `$a = $a * 5;`

- Operator /=

Digunakan untuk membagi variabel dengan dengan bilangan. Contoh:

`$a /= 5;` akan sama artinya dengan `$a = $a / 5;`

- Operator %=

Digunakan untuk mencari sisa hasil bagi dari sebuah variabel dengan bilangan tertentu. Contoh:

`$a %= 5;` akan sama artinya dengan `$a = $a % 5;`

- Operator &=

Digunakan untuk melakukan operasi logika AND pada sebuah variabel.

Contoh:

`$a &= 5;` akan sama artinya dengan `$a = $a & true;`

- Operator |=

Digunakan untuk melakukan operasi logika OR pada sebuah variabel. Contoh:

`$a |= 5;` akan sama artinya dengan `$a = $a | false;`

- Operator ^=

Digunakan untuk melakukan operasi logika XOR pada sebuah variabel.

Contoh:

`$a ^= 5;` akan sama artinya dengan `$a = $a ^ false;`

- Operator .=

Digunakan untuk menambahkan string pada sebuah variabel. Contoh:

`$a .= "Mulia Sulistiyono";` akan sama artinya dengan `$a = $a . "Mulia Sulistiyono";` dan `$a .= "Bernadhed";` akan sama artinya dengan `$a = $a . "Bernadhed";`

- Operator ++

Digunakan untuk menambahkan nilai satu (increment) pada sebuah variabel. Contoh:

`$a++;` akan sama artinya dengan `$a = $a + 1;`

- Operator –

Digunakan untuk mengurangi nilai satu (decrement) pada sebuah variabel.

Contoh:

`$a--;` akan sama artinya dengan `$a = $a - 1;`

10.5.5 Operator Penggabungan String

Pada bahasa pemrograman PHP, string digabungkan dengan karakter . (titik).

10.5.6 Operator Percabangan

Pada umumnya, operator percabangan dilakukan dengan struktur if-else, seperti pada contoh berikut:

```
if ($user == "mulia") {
```

```
echo "Selamat datang Mulia.";  
} else {  
    echo "Username salah.";  
}
```

Hal yang sama dapat dilakukan dengan cara :

```
echo ($user == "mulia" ? "Selamat datang Mulia." : "Username salah.");
```

Kondisi

Pernyataan 1

Pernyataan 2

Struktur diatas dapat dijelaskan sebagai berikut:

1. Pertama kali, PHP akan memeriksa apakah kondisi yang menjadi prasyarat bernilai benar. Yaitu apakah variabel \$user bernilai "mulia".
2. Jika benar, pernyataan 1 yang dipilih, dan pernyataan 2 akan dipilih jika kondisi prasyarat bernilai salah.
3. Kemudian program akan menampilkan (echo) pernyataan yang telah dipilih tadi.

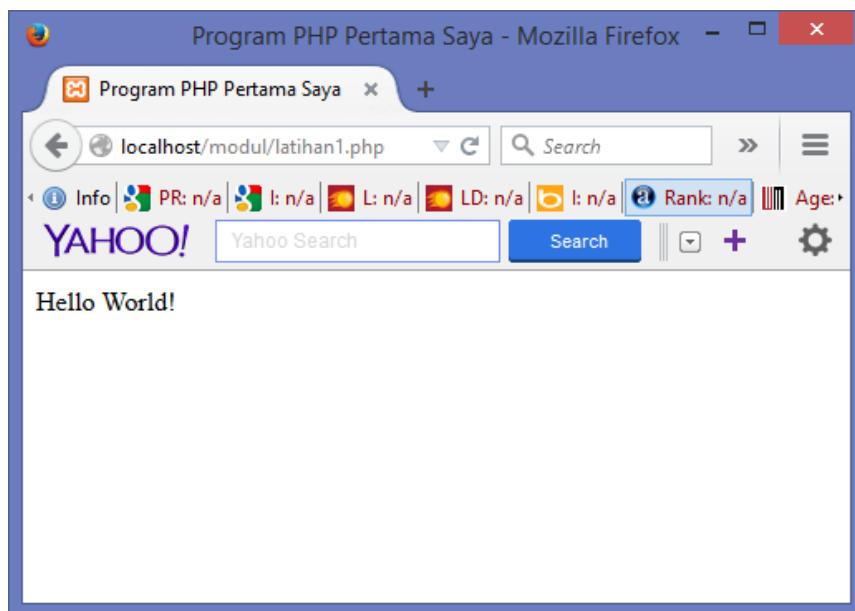
LATIHAN

1. Contoh program PHP

Simpan dengan nama **Latihan1.php**

```
1<html>
2  <head>
3    <title>Program PHP Pertama Saya</title>
4  </head>
5  <body>
6    <?php echo "Hello World!"; ?>
7  </body>
8</html>
```

Gambar 10.2 Contoh Program PHP



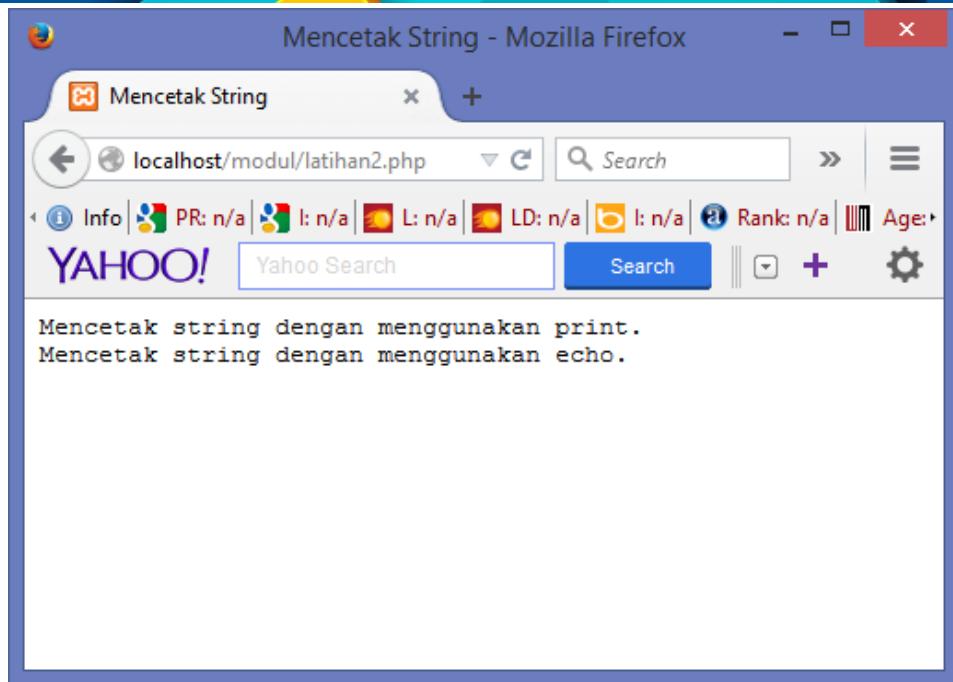
Gambar 10.3 Hasil Running Script Program PHP

2. Program PHP untuk mencetak string

Simpan dengan nama **Latihan2.php**

```
1<html>
2  <head>
3    <title>Mencetak String</title>
4  </head>
5  <body>
6    <pre><?php
7      print "Mencetak string dengan menggunakan print.\n";
8      echo "Mencetak string dengan menggunakan echo.";
9    ?>
10   </pre>
11  </body>
12</html>
```

Gambar 10.4 Script PHP untuk mencetak String



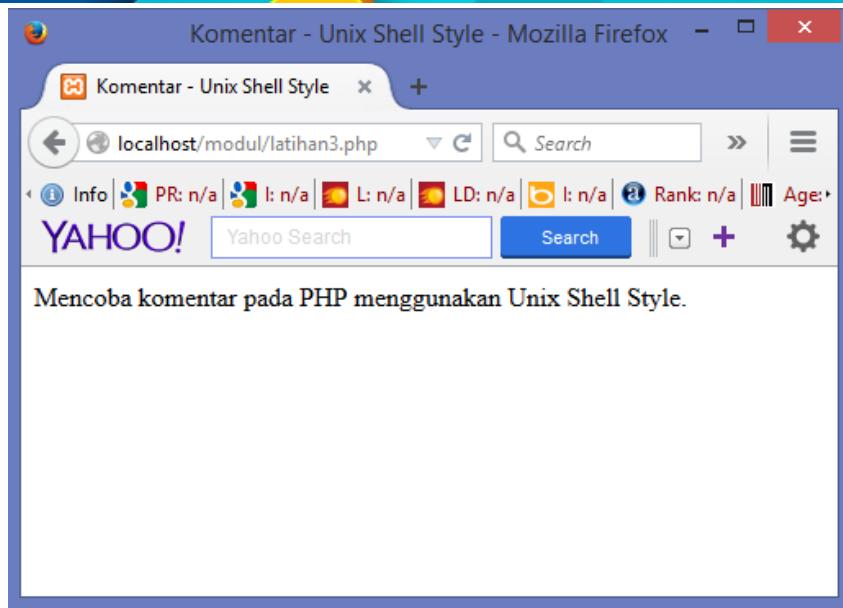
Gambar 10.5 Hasil Running Script PHP untuk mencetak String

3. Komentar PHP dengan Unix Shell Style

Simpan dengan nama **Latihan3.php**

```
1 <html>
2   <head>
3     <title>Komentar - Unix Shell Style</title>
4   </head>
5   <body>
6     <?php
7       # ini adalah skrip latihan komentar
8       # menggunakan hash ( # )
9       echo "Mencoba komentar pada PHP menggunakan Unix Shell Style.";
10      ?>
11    </body>
12  </html>
```

Gambar 10.6 Script PHP komentar menggunakan Unix Shell Style



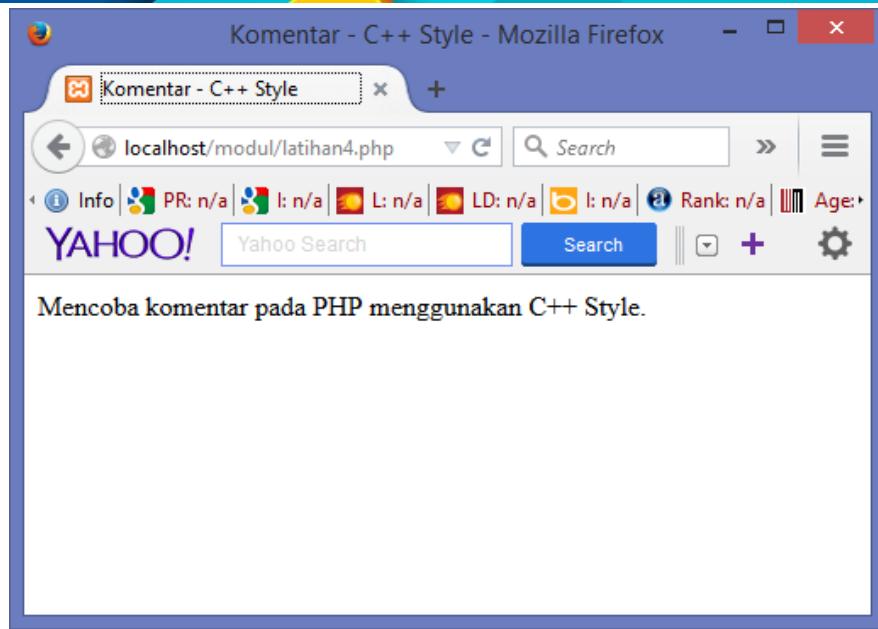
Gambar 10.7 Hasil Running Script PHP Script PHP komentar menggunakan Unix Shell Style

4. Komentar PHP dengan C++ Style

Simpan dengan nama **Latihan 4.php**

```
1<html>
2  <head>
3    <title>Komentar - C++ Style</title>
4  </head>
5  <body>
6    <?php
7      // ini adalah skrip latihan komentar
8      // menggunakan double slash ( // )
9      echo "Mencoba komentar pada PHP menggunakan C++ Style.";
10     ?>
11   </body>
12 </html>
```

Gambar 10.8 Script PHP komentar menggunakan C++ Style



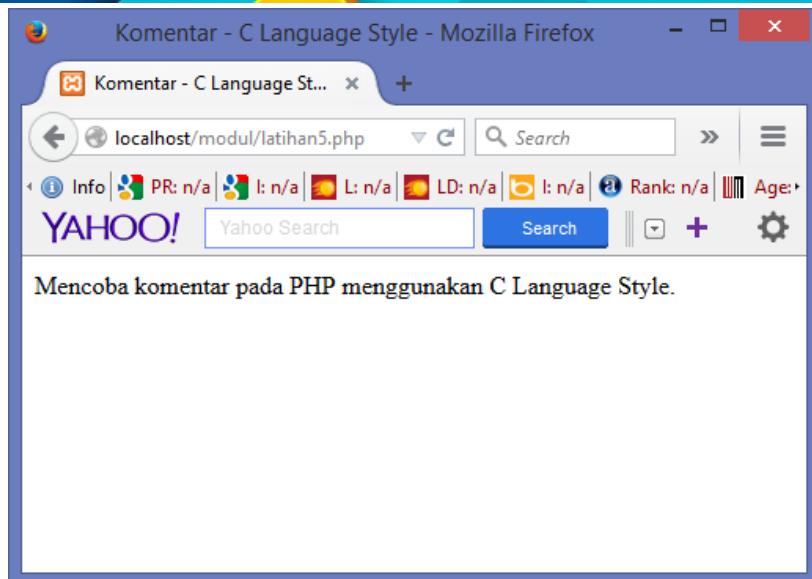
Gambar 10.9 Hasil Running Script PHP komentar menggunakan C++ Style

5. Komentar PHP dengan C *Language Style*

Simpan dengan nama **Latihan5.php**

```
1<html>
2  <head>
3      <title>Komentar - C Language Style</title>
4  </head>
5  <body>
6      <?php
7      /*
8          ini adalah skrip latihan komentar
9          menggunakan C style
10         bisa banyak baris hanya dengan 1 buah komentar saja
11     */
12     echo "Mencoba komentar pada PHP menggunakan C Language Style.";
13     ?>
14  </body>
15</html>
```

Gambar 10.10 Script PHP komentar menggunakan C Language Style



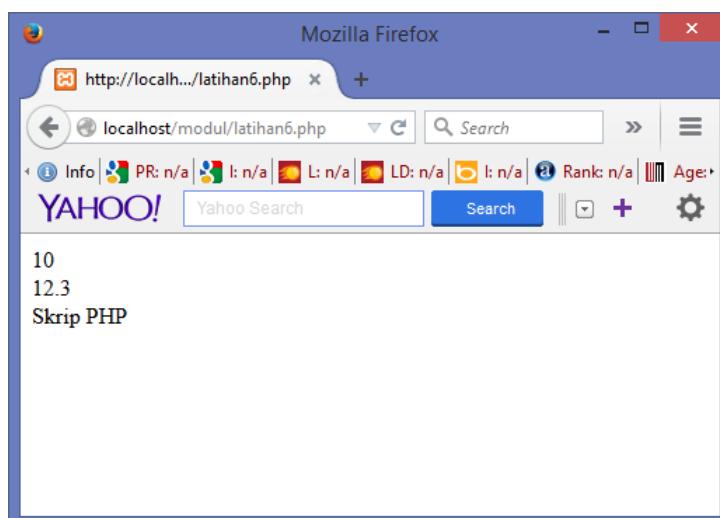
Gambar 10.11 Hasil Running Script PHP komentar menggunakan C Language Style

6. Contoh penugasa variable pada PHP

Simpan dengan nama **Latihan6.php**

```
1 <?php
2 $info = 10; // variabel $info menampung bilangan bulat
3 echo $info;
4 echo "<br />";
5 $info = 12.3; // variabel $info menampung bilangan pecahan
6 echo $info;
7 echo "<br />";
8 $info = "Skrip PHP"; // variabel $info menampung string
9 echo $info;
10 ?>
```

Gambar 10.12 Script PHP penugasan variable



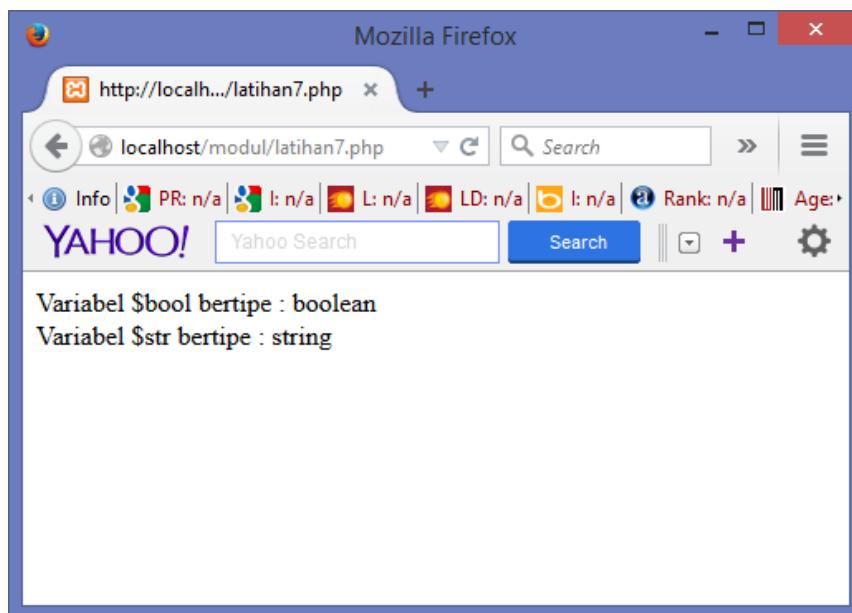
Gambar 10.13 Hasil Running Script PHP penugasan variabel

7. Program PHP variable dengan tipe data

Simpan dengan nama **Latihan7.php**

```
1 <?php
2 $bool = true; // sebuah boolean
3 $str = "foo"; // sebuah string
4 $int = 12; // sebuah integer
5 echo "Variabel \$bool bertipe : ";
6 echo gettype($bool);
7 echo "<br />";
8 echo "Variabel \$str bertipe : ";
9 echo gettype($str);
10 ?>
```

Gambar 10.14 Script PHP variable dengan tipe data



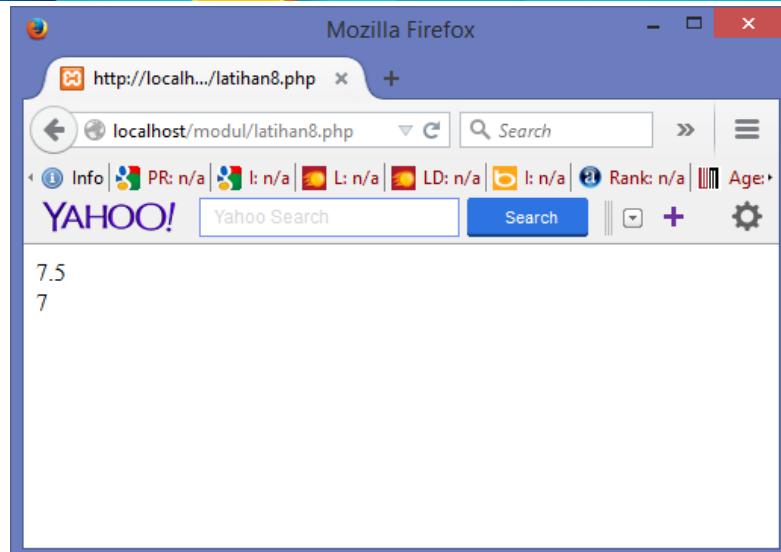
Gambar 10.15 Hasil Running Script PHP variable dengan tipe data

8. Program konversi tipe data dengan PHP

Simpan dengan nama **Latihan8.php**

```
1 <?php
2 $my_float = 7.5; // memberikan nilai pecahan ke variabel $my_float
3 echo $my_float; // tampilkan data yang terkandung dalam $my_float
4 echo "<br />";
5 echo (int) $my_float; // hasil type casting dalam bentuk bilangan integer
6 ?>
```

Gambar 10.16 Script PHP konversi tipe data



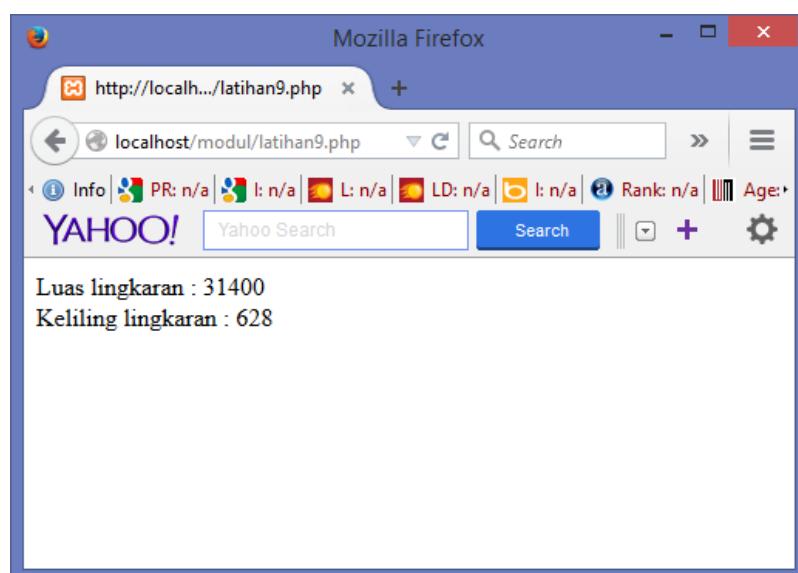
Gambar 10.17 Hasil Running Script PHP konversi tipe data

9. Program PHP menghitung luas dan keliling lingkaran

Simpan dengan nama **Latihan9.php**

```
1 <?php
2 define("PI", 3.14);
3 $radius = 100;
4 $luas_lingkaran = PI * ($radius * $radius);
5 $keliling_lingkaran = 2 * PI * $radius;
6 echo "Luas lingkaran : ".$luas_lingkaran;
7 echo "<br />";
8 echo "Keliling lingkaran : ".$keliling_lingkaran;
9 ?>
```

Gambar 10.18 Script PHP menghitung luas dan keliling lingkaran



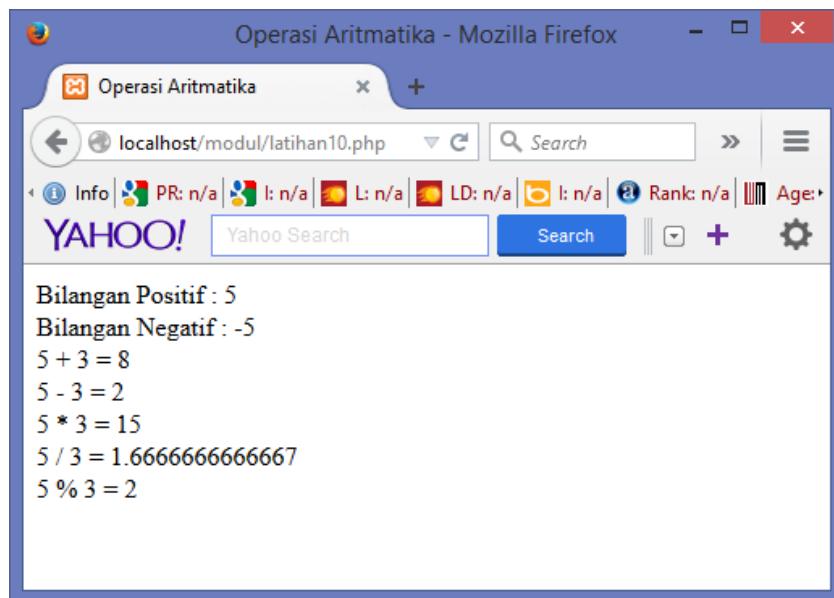
Gambar 10.19 Hasil Running Script PHP menghitung luas dan keliling lingkaran

10. Program PHP untuk operasi aritmatika

Simpan dengan nama **Latihan10.php**

```
1 <html>
2   <head>
3     <title>Operasi Aritmatika</title>
4   </head>
5   <body>
6     Bilangan Positif : <?= +5 ?>
7     <br />
8     Bilangan Negatif : <?= -5 ?>
9     <br />
10    5 + 3 = <?= 5 + 3 ?>
11    <br />
12    5 - 3 = <?= 5 - 3 ?>
13    <br />
14    5 * 3 = <?= 5 * 3 ?>
15    <br />
16    5 / 3 = <?= 5 / 3 ?>
17    <br />
18    5 % 3 = <?= 5 % 3 ?>
19  </body>
20 </html>
```

Gambar 10.20 Script PHP contoh operator aritmatika



Gambar 10.21 Hasil Running Script PHP contoh operator aritmatika

Perhatikan pada contoh kode program diatas. Pernyataan **<?= 5 + 3 ?>** secara singkat dapat menggantikan pernyataan lengkap yang lebih panjang:

```
$bil_1 = 5;
$bil_2 = 3;
$hasil = $bil_1 + $bil_2;
echo $hasil;
```

11. Program PHP dengan operator Penugasan

Simpan dengan nama **Latihan11.php**

```
1 <html>
2   <head>
3     <title>Operator Penugasan</title>
4   </head>
5   <body>
6     <?php
7       $a = 10; // mula mula nilai variabel $a = 10
8
9       echo "1. Penugasan +=";
10      echo "<br />";
11
12      $a += 8;
13      echo "Variabel \$a saat ini bernilai : ".\$a;
14      ?>
15      <br />
16      <br />
17      <?php
18      $a = 10; // nilai variabel $a dikembalikan menjadi 10
19
20      echo "2. Penugasan ++";
21      echo "<br />";
22
23      $a++;
24      echo "Variabel \$a saat ini bernilai : ".\$a;
25      ?>
26    </body>
27  </html>
```

Gambar 10.21 Script PHP contoh operator penugasan



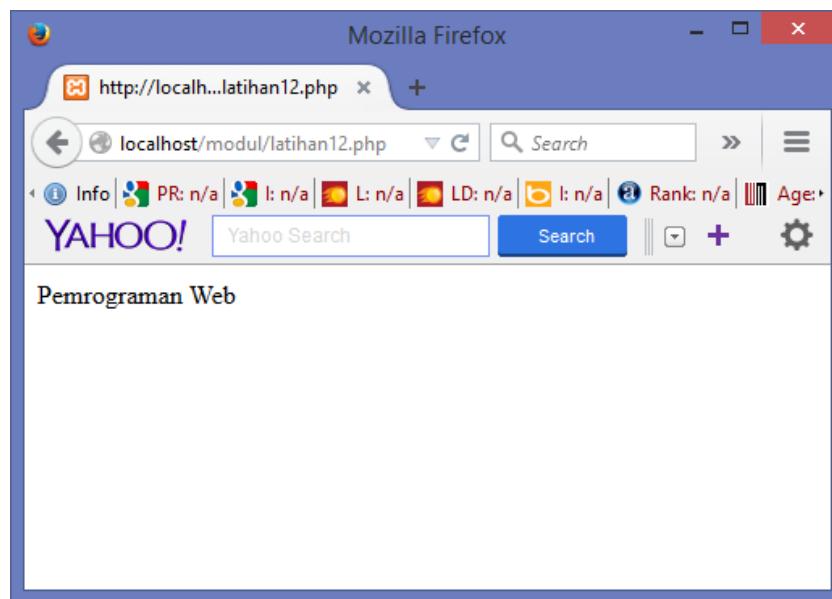
Gambar 10.22 Hasil Running Script PHP contoh operator penugasan

12. Program PHP untuk operator penggabungan string

Simpan dengan nama **Latihan12.php**

```
1 <?php
2 $string_1 = "Pemrograman";
3 $string_2 = "Web";
4 echo $string_1." ".$string_2;
5 ?>
```

Gambar 10.23 Script PHP contoh operator penggabungan string



Gambar 10.24 Hasil Running Script PHP contoh operator penggabungan string

Menyetujui

Dosen Rumpun Pemrograman Web

M. Rudyanto Arief, M.T.

NIK. 190302098