

Web Programming

 GRAHA ILMU

Buku Web Programming I berisikan materi belajar mengenai dasar-dasar pemrograman web. Buku ini direkomendasikan bagi pemula belajar pemrograman web. Buku ini menjelaskan bagaimana belajar dasar-dasar pemrograman web dengan mudah, praktis dan cepat disertakan contoh latihan-latihan. Dan adanya latihan contoh studi kasus membuat website yang responsive. Buku ini membahas mengenai dasar-dasar bahasa pemrograman web antara lain : *HyperText Markup Language (HTML)*, *Cascading Style Sheets (CSS)*, *Hypertext Preprocessor (PHP)* dan *JavaScript*.



Ani Oktarini Sari, S.Kom, MMSI. Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Mendapat gelar Sarjana pada tahun 2010 di STMIK Nusa Mandiri Jakarta dan Magister Manajemen Sistem Informasi Peminatan Rekayasa Perangkat Lunak di Universitas Gunadarma pada tahun 2015.



Ari Abdilah, M.Kom. Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Mendapatkan gelar Sarjana pada tahun 2011 dan Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta pada tahun 2015.



Sunarti, M.Kom. Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Menyelesaikan pendidikan Sarjana tahun 2009 dan Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta pada tahun 2011.



Buku ini diterbitkan atas kerjasama dengan
Universitas Bina Sarana Informatika

ISBN: 978-623-228-221-6



Web Programming

Ani Oktarini Sari

Ari Abdilah

Sunarti

Web Programming



GRAHA ILMU

WEB PROGRAMMING

oleh Ani Oktarini Sari; Ari Abdilah; Sunarti

Hak Cipta © 2019 pada penulis

Edisi Pertama: Cetakan I ~ 2019



GRAHA ILMU

Ruko Jambusari 7A Yogyakarta 55283

Telp: 0274-889398; 0274-882262; Fax: 0274-889057;

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apa pun, secara elektronis maupun mekanis, termasuk memfotokopi, merekam, atau dengan teknik perekaman lainnya, tanpa izin tertulis dari penerbit.

ISBN: 978-623-228-221-6

Buku ini tersedia sumber elektronisnya

DATA BUKU:

Format: 17 x 24 cm; Jml. Hal.: xii + 94; Kertas Isi: HVS 70 gram; Tinta Isi: BW; Kertas Cover: Ivori 260 gram; Tinta Cover: Colour; Finishing: Perfect Binding: Laminasi Doff.



PRAKATA

Alhamdulillah, puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat dan hidayah-Nya, penulis dapat menyelesaikan buku Web Programming belajar dasar-dasar pemrograman web.

Buku Web Programing belajar mengenai dasar-dasar pemrograman web disusun untuk keperluan mahasiswa atau siapapun yang ingin belajar mengenai dasar-dasar pemrograman web. Pada buku ini berisi bagaimana dapat membuat website dan belajar dasar-dasar pemrograman web dengan mudah, praktis dan cepat disertakan contoh latihan-latihan.

Penulis menyusun materi didalam buku ini secara sistematis antara lain untuk keperluan mahasiswa atau siapapun mengenai dasar-dasar web yang digunakan pada bidang ilmu komputer. Adapun materi inti dari web programming adalah *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), *Hypertext Preprocessor* (PHP) dan *JavaScript*. Kehadiran buku ini sangat diharapkan dapat berguna bagi mahasiswa atau siapapun dalam mempelajari dasar-dasar pemrograman web.

Penulis mengucapkan terima kasih kepada tim sehingga bisa menyelesaikan penyusunan buku ini. Semoga buku ini berguna bagi para pem-baca baik mahasiswa ataupun siapapun yang bisa dijadikan bahan referensi untuk pembelajaran.

Jakarta, 2019

Tim Penulis



KATA PENGANTAR

Segala puji syukur kepada Tuhan Yang Maha Esa karena berkat rahmatNya penulisan buku ajar Web Programming mengenal dasar-dasar pemrograman web dapat terselesaikan dengan baik. Buku ini disusun untuk memenuhi kebutuhan mahasiswa atau siapapun yang akan belajar mengenai dasar-dasar pemrograman web. Pada buku Web Programming yang disajikan dalam bentuk praktikum dan diharapkan dapat membekali mahasiswa atau siapapun yang mempelajari buku ini.

Buku Web Programming ini membahas materi mengenai *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), *Hypertext Preprocessor* (PHP) dan *JavaScript*. Tujuan akhir baik mahasiswa ataupun siapapun yang mempelajari buku ini diharapkan dapat mengimplementasikan bahasa pemrograman berbasis website dengan baik.

Penulis menyadari sepenuhnya bahwa buku ini memiliki banyak kekurangan. Untuk itu penulis dengan mengharapakan adanya masukan dan kritik yang membangun dari berbagai pihak demi kesempurnaan buku ini di masa yang akan datang. Semoga buku ini bisa bermanfaat bagi para pembacanya.

Jakarta, 2019

Tim Penulis



DAFTAR ISI

PRAKATA	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
BAB 1 KONSEP DASAR WEB	1
1.1 Pengertian Website	1
1.2 Pengertian Pemrograman Web	2
1.3 Bahasa Skrip Pemrograman Web	2
1.4 Istilah-Istilah dalam Pemrograman Web	2
1.5 Struktur Navigasi	4
1.6 Text Editor	7
BAB 2 PENGENALAN HTML	9
2.1 Pengertian Hypertext Markup Language (HTML)	9
2.2 Struktur Dasar HTML	10
2.3 TAG	13
2.4 Pembuatan Tabel Menggunakan HTML	16
BAB 3 PHP	23
3.1 Pengertian PHP	23
3.2 Perbedaan HTML dengan PHP	25
3.3 Dasar-dasar PHP	26

BAB 4	OPERATOR	31
4.1	Mengenal Operator	31
4.2	Jenis-Jenis Operator	32
BAB 5	PENGENALAN FORM DAN PENGGUNAAN HTTP SERVER	37
5.1	Komponen <i>Form</i>	37
5.2	Pengolahan Data Dari Form	38
BAB 6	PERCABANGAN	47
6.1	Pernyataan Seleksi	47
BAB 7	PERULANGAN	55
7.1.	Pengertian Dasar Perulangan	55
7.2.	Pengertian Dasar Perulangan Pada PHP	56
BAB 8	JAVASCRIPT	65
8.1.	Pengertian Javascript	65
8.2.	Latihan pembuatan koding javascript sederhana	66
BAB 9	CSS	71
9.1.	Pengertian CSS	71
9.2.	Beberapa hal yang dapat dilakukan dengan CSS.	72
9.3.	Cara Pemasangan CSS Pada Dokumen HTML	72
9.4.	Penulisan CSS	74
9.5.	Properti-properti CSS	75
9.6	PADDING, MARGIN DAN BORDER	77
BAB 10	MEMBUAT DESIGN WEB RESPONSIVE MENGGUNAKAN CSS	83
10.1	Merancang Design Web Responsive menggunakan CSS	83
10.2	Merancang Halaman Index Web Responsive	88
DAFTAR PUSTAKA		93



DAFTAR GAMBAR

Gambar 1.1	Struktur Navigasi Linier	4
Gambar 1.2	Struktur Navigasi Hirarki	5
Gambar 1.3	Struktur Navigasi Non-Linier	6
Gambar 1.4	Struktur Navigasi Campuran	7
Gambar 2.1	Simpan File dengan Notepad	12
Gambar 2.2	Menjalankan Apache via Xampp Control	12
Gambar 2.3	Lihat Isi modul_web1	13
Gambar 2.4	Hasil dari contoh01.html	13
Gambar 2.5	Hasil tampilan contoh02.html	15
Gambar 2.6	Hasil tampilan LatihanTag.html	16
Gambar 2.7	tabel sebelum di gabung kolom	18
Gambar 2.8	tabel setelah di gabung kolom	18
Gambar 2.9	tabel sebelum di gabung baris	19
Gambar 2.10	tabel setelah di gabung baris	19
Gambar 2.11	Hasil contoh03.html	21
Gambar 2.12	Hasil dari tabelcell.html	22
Gambar 3.1	Hasil contoh04.php	24
Gambar 3.2	Contoh pemanggilan file html di browser	25
Gambar 3.3	Contoh pemanggilan file php di browser	25
Gambar 3.4	Hasil Contoh04.php	27
Gambar 3.5	Hasil dari contoh05.php	28
Gambar 3.6	Hasil tampilan contoh06.php	29

Gambar 4.1	Hasil Operatoraritmatika	32
Gambar 4.2	Hasil Operatorperbandingan.php	34
Gambar 4.3	Hasil operatorlogika.php	35
Gambar 4.4	Hasil Operatorstring.php	35
Gambar 4.5	Hasil dari jawaban latihan	36
Gambar 5.1	Hasil metodeget.php	38
Gambar 5.2	Hasil metodegetproses.php	39
Gambar 5.3	Hasil metodepost.php	40
Gambar 5.4	Hasil metodepostproses.php	41
Gambar 5.5	Hasil formdatadiri.php	42
Gambar 5.6	Hasil kiriman data formdatadiri	42
Gambar 5.7	Tampilan forminputmahasiswa.php	43
Gambar 5.8	Tampilmahasiswa.php	43
Gambar 6.1.	Hasil pernyataan if dan else	49
Gambar 6.2.	Hasil pernyataan if majemuk	50
Gambar 6.3	Hasil statement switch	52
Gambar 6.4	Rancangan Tampilan Input Latihan	52
Gambar 6.5	Rancangan Tampilan Output Latihan	52
Gambar 7.1	Hasil Perulangan Menggunakan For	57
Gambar 7.2	Hasil Perulangan Menggunakan While	58
Gambar 7.3	Hasil Perulangan Menggunakan Do-While	59
Gambar 7.4	Hasil Perulangan Menggunakan Do-While 2	60
Gambar 7.5	Hasil Perulangan Menggunakan Foreach Array Tanpa Key	61
Gambar 7.6	Hasil Perulangan Menggunakan Foreach Array dengan Key	62
Gambar 7.7	Hasil tampilan latihan	63
Gambar 8.1	Rancangan Tampilan Latihan di browser	69
Gambar 9.1	Susunan penulisan sintaks CSS	74
Gambar 9.2	Box Model CSS	77
Gambar 9.3	Hasil Tampilan latihan CSS	81
Gambar 10.1	Rancangan tampilan responsive yang akan dibuat	83
Gambar 10.2	Hasil tampilan halaman responsive	88

BAB I

KONSEP DASAR WEB

Deskripsi

Membahas mengenai konsep dasar pemrograman web, istilah-istilah dalam pemrograman web, menggunakan text editor, mengenal dan mengimplementasikan struktur navigasi

Tujuan Pembelajaran :

1. Mampu memahami konsep dasar Pemrograman Web
2. Mampu memahami istilah-istilah yang ada dalam pemrograman web
3. Mampu menggunakan text editor
4. Mampu mengimplementasikan Struktur Navigasi

1.1. Pengertian Website

Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara dan video atau gabungan dari semuanya yang terkoneksi oleh internet, sehingga dapat dilihat oleh seluruh siapapun yang terkoneksi jaringan internet. Jenis Kategori Website:

a. Web Statis

Merupakan website yang mempunyai halaman yang tidak berubah. Perubahan suatu halaman dilakukan secara manual dengan mengedit code yang menjadi stuktur dari website tersebut.

b. Web Dinamis

Merupakan website yang secara terstruktur diperuntukan untuk diupdate sesering mungkin. Biasanya disediakan halaman backend untuk melakukan perubahan konten dari website tersebut. Contohnya : web portal, web berita, dll.

c. **Web Interaktif**

Merupakan website yang berinteraksi antara penggunanya. Biasanya berupa forum diskusi maupun blog. Dimana adanya moderator sebagai pengatur alur diskusi.

1.2. Pengertian Pemrograman Web

Pemrograman web adalah pembuatan aplikasi program dengan bahasa skrip yang akan menghasilkan sebuah aplikasi yang diakses pada web browser.

1.3. Bahasa Skrip Pemrograman Web

Dalam mempelajari pemrograman web, perlu diketahui pemahaman beberapa bahasa skrip yang dibutuhkan dalam pembuatan 1 halaman website, yaitu :

- a. HTML
- b. PHP
- c. CSS
- d. JAVASCRIPT

1.4 Istilah-Istilah dalam Pemrograman Web

Ada beberapa istilah yang sering digunakan apabila anda mempelajari pemrograman web ini diantaranya yaitu:

1. **Internet**

Internet berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan. Sistem komputer terhubung secara global dan menggunakan TCP/IP sebagai *protocol*. Melalui internet pertukaran informasi dapat dilakukan tanpa batas asalkan kedua pihak terhubung kedalam jaringan yang sama.

2. *World Wide Web (WWW)*

WWW merupakan kumpulan *web server* diseluruh dunia yang dapat menyediakan data dan informasi untuk dapat digunakan secara massal.

3. *Website*

Website merupakan istilah yang merujuk kepada suatu nama halaman web yang dapat diakses jika terkoneksi dengan internet. Untuk mengakses sebuah website, selain terkoneksi dengan internet, anda pun harus membukanya dalam sebuah aplikasi web browser.

4. *Web Server*

Web Server merupakan perangkat lunak yang digunakan untuk menerima permintaan melalui Protokol HTTP atau HTTPS dari client, kemudian dikirimkan kembali dalam bentuk halaman-halaman web. Contoh web server : Xampp, Apache2Triad, dll.

5. *URL (Universal Resource Locator)*

URL merupakan suatu alamat yang menunjukkan sebuah halaman tertentu *internet*. Contoh URL adalah: <http://www.google.com>

6. *HTTP (Hypertext Transfer Protocol)*

HTTP merupakan bagian dari URL digunakan untuk mengidentifikasi lokasi *web*, dan digunakan dalam protokol HTML.

7. *DNS (Domain Name System)*

DNS merupakan sistem *database* terdistribusi yang tidak banyak dipengaruhi oleh bertambahnya *database*. DNS menjamin informasi *host* terbaru akan disebarkan ke jaringan bila diperlukan.

8. *IP (Internet Protocol)*

IP (*Internet Protocol*) merupakan protokol yang digunakan dalam *internet*, secara teknis bermakna suatu bentuk pengisian dan

pengalamatan data-data dan informasi yang akan dikirim melalui *internet*.

9. *Hyperlink*

Hyperlink atau disebut *link* merupakan sebuah tools yang berperan dalam mempopulerkan penggunaan *internet*, *hyperlink* dapat mereferensikan sebuah teks atau gambar ke alamat lain di *internet*.

10. *Web Browser*

Menggunakan *web browser* mudah, yang diperlukan hanyalah Anda harus memiliki alamat *web* yang akan dibuka. Alamat ini biasa disebut dengan *Uniform Resource Locator* (URL). Jika menggunakan sistem operasi *Windows* terdapat program *web browser* bawaan, yaitu *Internet Explorer*. Terdapat banyak program *alternative web browser* yang sebagian besar bersifat gratis, seperti *Firefox*, *Opera*, *Safari*, *chrome* dan seterusnya.

1.6 Struktur Navigasi

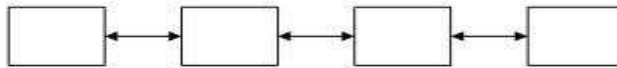
Struktur Navigasi adalah bagan hirarki dari suatu website yang menggambarkan isi dari setiap halaman dan link. Struktur Navigasi dapat dikatakan sebagai gambaran dari hubungan atau rantai kerja dari seluruh elemen yang akan digunakan dalam aplikasi.

Kriteria-kriteria yang digunakan dalam pengelompokan dari struktur navigasi adalah sebagai berikut: kebutuhan akan objek, kemudahan pemakaian, interaktif, dan kemudahan membuatnya yang berpengaruh terhadap waktu pembuatan suatu websites. Dalam penggambarannya Struktur Navigasi terbagi kedalam 4 Struktur yang berbeda yaitu: *Linier*, *Non Linier*, *Hierarchical* (Hirarki) dan *Composit* (Campuran).

Ada 4 macam bentuk dasar dari peta navigasi yang biasa digunakan dalam proses pembuatan aplikasi web, yaitu:

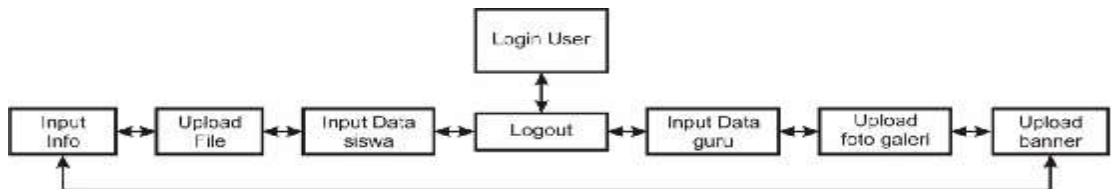
1. Struktur Navigasi Linier

Pada Struktur navigasi linier merupakan rangkaian urutan tampilan dari sebuah halaman website yang ditampilkan secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.



Gambar 1.1. Struktur Navigasi Linier

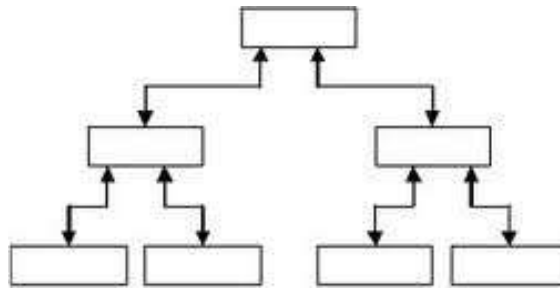
Contoh :



Gambar 1.2 Contoh Struktur Navigasi Linier

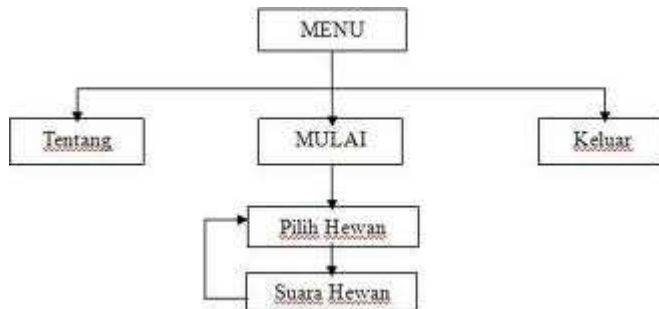
2. Struktur Navigasi Hirarki

Pada Struktur navigasi hirarki merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Jika digambarkan, maka struktur navigasi hirarki ada halaman web yang disebut sebagai Master Page (halaman utama pertama). Pada halaman utama ini mempunyai halaman percabangan yang disebut *Slave Page* (halaman pendukung). Dan jika halaman pendukung diklik, maka dia akan berubah menjadi Master Page dan kemudian memiliki percabangan halaman pendukung lainnya, dan seterusnya. Dalam struktur navigasi ini tidak diperbolehkan adanya tampilan secara linier.



Gambar 1.3. Struktur Navigasi Hirarki

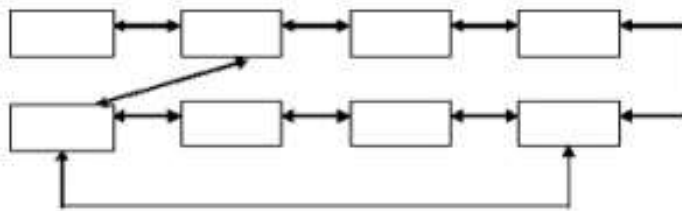
Contoh :



Gambar 1.4 Contoh Struktur Navigasi Hirarki

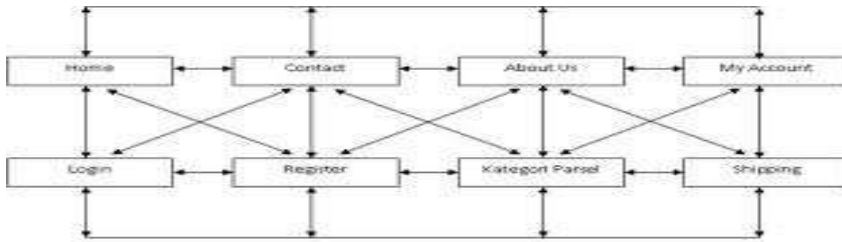
3. Struktur Navigasi Non-Linier

Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang ada pada struktur nonlinier ini berbeda dengan percabangan yang ada pada struktur hirarki, karena pada percabangan nonlinier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai tingkatan yang sama yaitu tidak ada *Master Page*(halaman utama website) dan *Slave Page*(halaman pendukung website).



Gambar 1.5. Struktur Navigasi Non-Linier

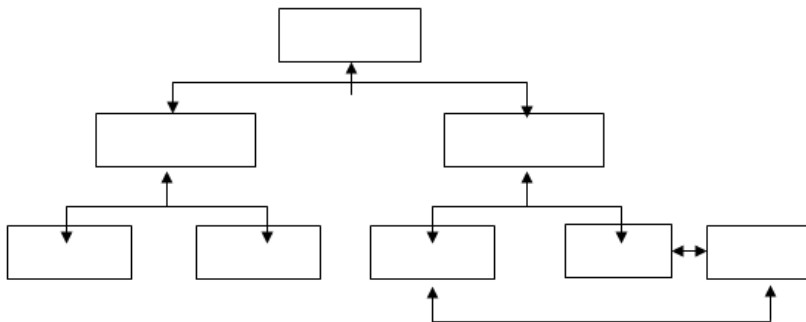
Contoh :



Gambar 1.6. Contoh Struktur Navigasi Non-Linier

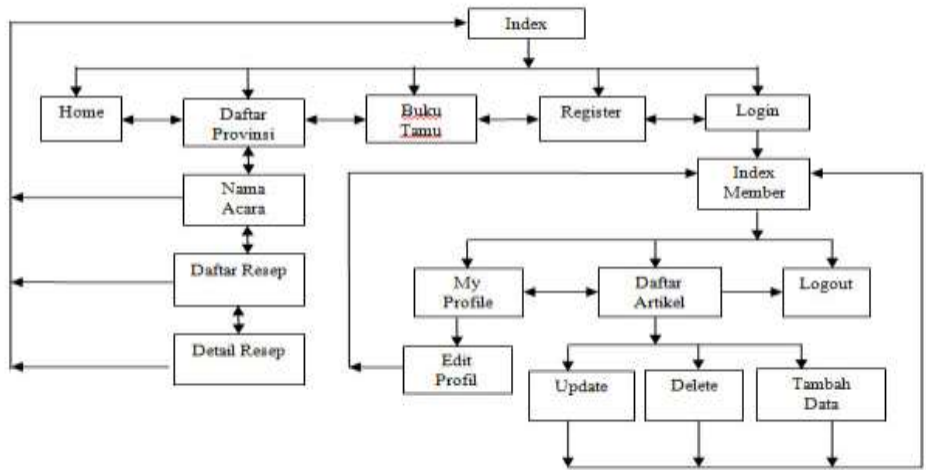
4. Struktur Navigasi Campuran

Struktur navigasi campuran disebut struktur navigasi bebas. Dimana didalamnya terdapat beberapa gabungan dari struktur navigasi lainnya. Struktur navigasi campuran ini banyak digunakan dalam membuat website, karena struktur ini dapat memberikan tingkat interaksi yang lebih tinggi. Dan keterikatan dalam halaman website dapat dibuat lebih efisien dan menarik.



Gambar 1.7. Struktur Navigasi Campuran

Contoh :



Gambar 1.8. Contoh Struktur Navigasi Campuran

1.3 Text Editor

Dalam membuat sebuah halaman web dibutuhkan text editor. Text editor yang dapat dipergunakan dalam pembelajaran pemrograman web ini dapat menggunakan pilihan sebagai berikut :

1. Notepad

Notepad adalah aplikasi text editor yang sudah ada didalam instalasi os windows anda masing-masing. Yang perlu diperhatikan jika menggunakan notepad sebagai text editor, yakni extension type file dituliskan langsung dan pemilihan save as type adalah All Files

2. Notepad++

Notepad++ adalah sebuah [penyunting teks](#) dan [penyunting kode sumber](#) yang berjalan di [sistem operasi Windows](#). Notepad++ menggunakan komponen [Scintilla](#) untuk dapat menampilkan dan menyuntingan teks dan berkas [kode sumber](#) berbagai [bahasa pemrograman](#).

3. Sublime Text

Text editor yang terbelang masih baru yang sangat mudah digunakan, penampilan simple namun enak dipandang. Sublime text adalah aplikasi berbayar tapi anda dapat mendownload versi demo-nya (meskipun versi demo tapi tidak ada batasan dalam penggunaannya).

4. Atom

Atom adalah teks editor gratis dan terbuka untuk macOS, Linux, dan Microsoft Windows dengan dukungan untuk plug-in yang ditulis dalam Node.js, dan tertanam Kontrol Git, yang dikembangkan oleh GitHub.

BAB II

Pengenalan HTML

Deskripsi :

Mampu mengenal skrip html, menggunakan ragam tag dan pendeklarasian tabel beserta atributnya, mampu menuliskan skrip dalam html

Tujuan Pembelajaran :

1. Memahami skrip html
2. Mampu menggunakan skrip html
3. Mampu menggunakan tag dalam penulisan skrip html
4. Mampu menggunakan tabel, penggunaan cell padding, dan cell span

2.1 Pengertian *Hypertext Markup Language* (HTML)

Hypertext Markup Language (HTML) adalah bahasa pemrograman yang digunakan untuk menampilkan sebuah website. HTML termasuk dalam bahasa pemrograman gratis, artinya tidak dimiliki oleh siapapun, pengembangannya dilakukan oleh banyak orang di banyak negara dan bisa dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama secara global.

Dokumen HTML adalah dokumen teks yang dapat diedit oleh editor teks apapun. Dan disimpan dengan file extension .html . Dokumen HTML punya beberapa elemen yang dikelilingi oleh tag-teks yang dimulai dengan symbol “ < ” dan berakhir dengan sebuah symbol “ > ”.

2.2. Struktur Dasar HTML

Elemen HTML dimulai dengan tag awal, yang diikuti dengan isi elemen dan tag akhir. Tag berakhir termasuk simbol / diikuti oleh tipe elemen, misalnya </HEAD>. Sebuah elemen HTML dapat bersarang di dalam elemen lainnya. Sebuah dokumen HTML standar terlihat seperti ini :

```
<html>
<head>
    <title> Judul Web </title>
</head>
<body>

</body>
</html>
```

Keterangan :

1. Tag HTML secara default dimulai dari <HTML> dan diakhiri dengan </HTML>.
2. Tag <HEAD> ... </HEAD> merupakan tag kepala sebelum badan. Tag kepala ini akan terlebih dulu dieksekusi sebelum tag badan. Di dalam tag ini berisi tag <META> dan <TITLE>. Tag <META> merupakan informasi atau header suatu dokumen HTML. Atribut yang dimiliki oleh tag ini antara lain:
 - a. HTTP_EQUIV, atribut ini berfungsi untuk menampilkan dokumen HTML secara otomatis dalam jangka waktu tertentu.
 - b. CONTENT, atribut ini berisi informasi tentang isi document HTML yang akan dipanggil.
 - c. NAME, atribut ini merupakan identifikasi dari meta itu sendiri. Tag <META> dalam suatu document HTML boleh ada maupun tidak.
3. Tag <TITLE> ... </TITLE> adalah tag judul. Sebaiknya setiap halaman web memiliki judul, dan judul tersebut dituliskan di dalam <TITLE> ... </TITLE>. Judul ini akan muncul dalam titlebar dari browser.
4. Tag <BODY> ... </BODY> adalah tag berisi content dari suatu halaman web.

Contoh penggunaan script HTML

Buat lembar baru pada Notepad, kemudian ketikkan perintah di bawah ini.

Simpan dengan nama **Contoh01.html**

```
<html>
<head>
  <title>Contoh 01 </title>
</head>
<body>
  Halo... <br>
  ini script HTML pertamaku
</body>
</html>
```

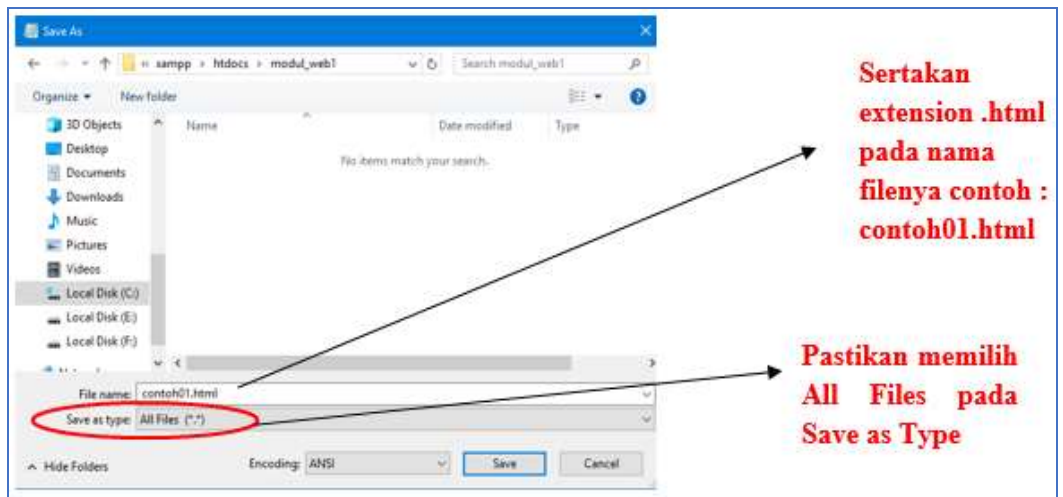
Kemudian simpan file di atas di dalam folder c:\XAMPP\htdocs\buat folder baru untuk menyimpan file di dalam folder htdocs. Simpan file dengan nama contoh01.html. Pembuatan nama file pada saat penyimpanan harus diakhiri dengan extention “.html”

Cara penyimpanan dengan Notepad, perhatikan cara berikut :

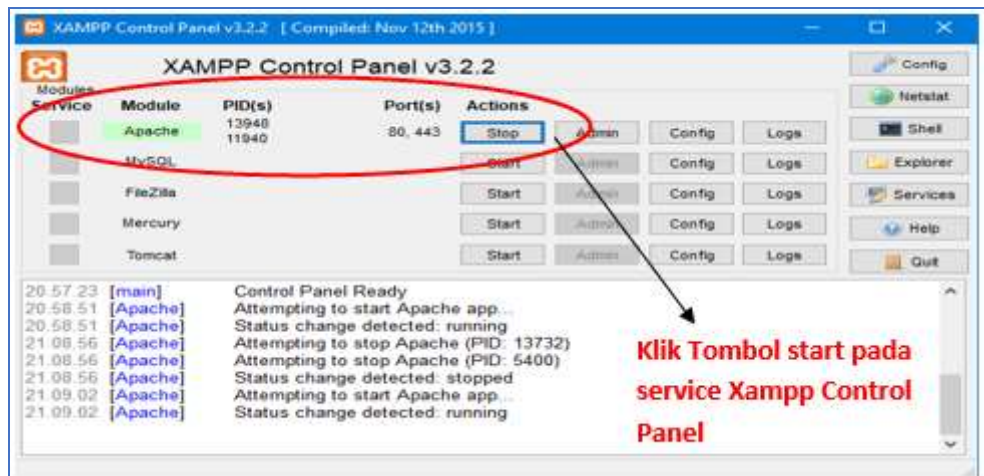
Untuk melihat hasil dari file di atas dapat menggunakan browser Mozilla, google chrome, internet explorer atau jenis browser lain. Ketikkan pada address bar “Localhost\Nama Folder Penyimpanan\”, kemudian pilih file contoh01.html

Sebelum di ketikkan alamat file tersebut, pastikan anda telah menjalankan Module Apache pada Xampp Control Panel.

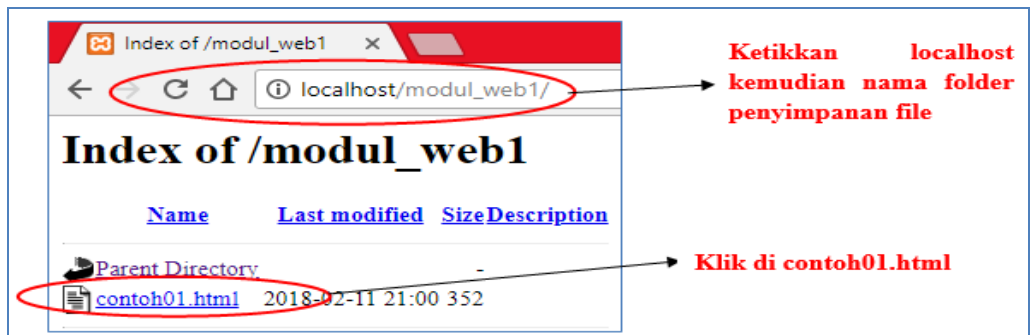
Lihat gambar di bawah ini :



Gambar 2.1 Simpan File dengan Notepad

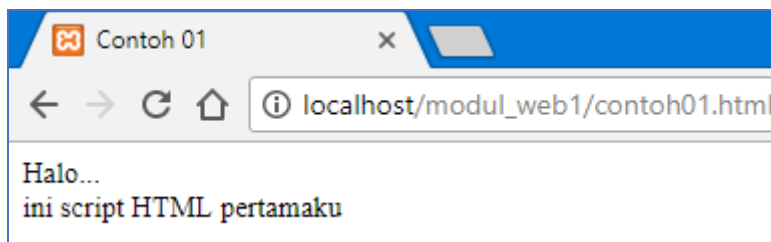


Gambar 2.2 Menjalankan Apache via Xampp Control



Gambar 2.3 Lihat Isi modul_web1

Hasilnya :



Gambar 2.4 Hasil dari contoh01.html

2.3 TAG

Kode-kode dalam HTML biasanya disebut **TAG**. Tag dalam HTML dituliskan diapit oleh tanda lebih kecil (<), tanda lebih besar (>), dan garis miring (/). Tag dituliskan berpasangan, jika tidak menggunakan tanda garis miring(/) setelah penulisan namanya, disebut sebagai tag pembuka. Namun, jika menggunakan tanda garis miring (/) sebelum nama tag, maka disebut sebagai tag penutup. Tag bersifat incasesensitiv yang dimana penulisan dengan huruf besar, huruf kecil dan campuran tidak masalah. Namun, untuk standarisasinya tag di tuliskan dalam huruf kecil.

Jenis – jenis tag dalam HTML :

Beberapa jenis tag yang dapat di pergunakan dalam penulisan skrip html, antara lain sebagai berikut :

Tag	Kegunaan
<html>	Untuk mendefinisikan sebuah dokumen HTML
<body>	Mendefinisikan body atau isi sebuah dokumen
<h1>...</h1> s/d <h6>...</h6>	Mendefinisikan heading 1 sampai 6, ukuran fontsize judul yang besar sampai yang terkecil
<p>....</p>	Mendefinisikan sebuah paragraph
 	Mendefinisikan break line / baris baru
<hr>	Mendefinisikan horizontal rule pemisah antar bagian atau paragraph
.....	Mendefinisikan pembuatan order

	list/penomoran dengan angka/huruf
.....	Mendefinisikan pembuatan unordered list/penomoran dengan bullets
.....	Mendefinisikan isi data dalam list
<i>.....</i>	Mendefinisikan format italic/huruf miring
.....	Mendefinisikan format bold/huruf tebal
<u>.....</u>	Mendefinisikan format underline/huruf bergaris bawah
_{.....}	Mendefinisikan teks subscript
^{.....}	Mendefinisikan teks superscript
.....	Mendefinisikan tampilkan gambar
<marquee>....</marquee>	Mendefinisikan tulisan bergerak
<table></table>	Mendefinisikan pembuatan tabel

Contoh script penggunaan Tag HTML

Buat lembar baru pada Notepad, kemudian ketikkan perintah di bawah ini.

Simpan dengan nama **Contoh02.html**

```
<html>
<head>
  <title>Contoh 02</title>
</head>
<body bgcolor="#00CCFF" text="#FF0000">
<p> Ini adalah contoh penggunaan formating TAG dalam HTML <br>
  masing-masing TAG memiliki atribut masing-masing<br> </p>
<font color="#000000">Ini juga termasuk contoh penggunaan formating
TAG<br></font>
<h1><marquee width="50%" bgcolor="#000099">Ini juga salah satu
penggunaan Tag</marquee> </h1>
</body>
</html>
```

Hasil Tampilan di browser



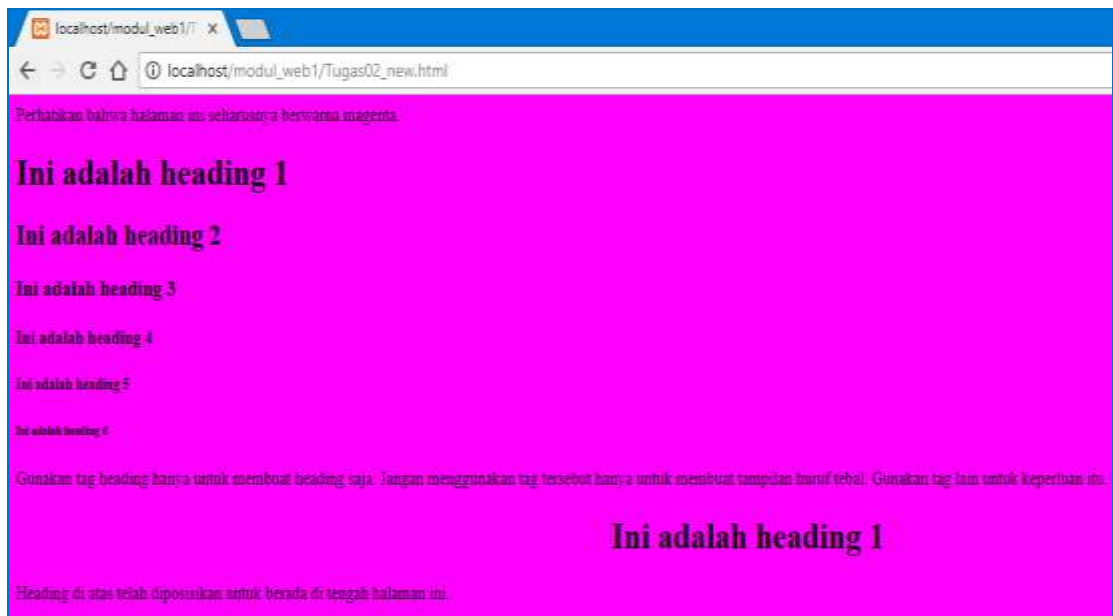
Gambar 2.5 Hasil tampilan contoh02.html

Latihan Tag

Simpan dengan nama LatihanTag.html

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

Hasil tampilan di web browser :



Gambar 2.6 Hasil tampilan LatihanTag.html

2.4 Pembuatan Tabel Menggunakan HTML

Tabel penting peranannya dalam halaman Web, selain untuk menampilkan teks atau gambar dalam format lajur dan kolom bias juga menggunakan tabel untuk membantu me-layout tampilan halaman.

Tabel merupakan sebuah kotak yang terdiri atas baris/row dan kolom.column. Untuk membuat tabel, anda menggunakan tag <table> dan menutupnya dengan tag </table>. Anda bisa juga menambahkan atribut lain di tag <table> pembuka. Misalnya menentukan warna, border, dan sebagainya.

Di dalam tag <table> ada beberapa tag lain yang perlu dipahami, yaitu :

a. Tag <tr>

Artinya tag untuk menuliskan baris biasa di tabel. TR singkatan dari Table Row.

b. Tag <td>

Artinya tag untuk menuliskan kotak di dalam baris, makanya tag <td> ada di dalam tag <tr>. TD singkatan dari Table Data.

c. Tag <th>

Artinya tag untuk menuliskan kotak biasa seperti <td>, namun untuk header tabel. TH singkatan dari Table Header.

Menggabungkan sel

Sel-sel tabel secara normal memiliki lebar dan tinggi yang sama. Jika kita ingin membuat sebuah sel memiliki lebar atau tinggi yang berbeda dari sel-sel lainnya, maka satu-satunya cara yang bisa kita lakukan adalah dengan menggabungkan beberapa sel menjadi satu. Cara ini disebut merge atau penggabungan sel.

Untuk menggabungkan sel-sel tabel ini diperlukan atribut rowspan atau colspan. Atribut rowspan digunakan untuk menggabungkan sel-sel tabel pada kolom yang sama. Atribut colspan untuk menggabungkan sel-sel tabel pada baris yang sama.

Berikut contoh penggabungan kedua jenis :

1. Secara Vertikal (*Rowspan*)

Tabel dengan kode HTML dibawah ini sel-sel kolom pertama akan digabung:

```
<table>
  <tr>
    <td>.....</td>
    <td>.....</td>
  </tr>
  <tr>
    <td>.....</td>
    <td>.....</td>
  </tr>
</table>
```


Gambar 2.7 tabel sebelum di gabung kolom

Setelah digabung maka kondisi kode HTML menjadi seperti berikut:

```
<table>
  <tr>
    <td rowspan="3">.....</td>
    <td>.....</td>
  </tr>
  <tr>
    <td>.....</td>
    <td>.....</td>
  </tr>
</table>
```


Gambar 2.8 tabel setelah di gabung kolom

2. Secara Horisontal (*Colspan*)

Tabel dengan kode HTML dibawah ini sel-sel baris pertama akan digabung:

```
<table>
  <tr>
    <td>.....
  </td>
    <td>.....
  </td>
</tr>
```


Gambar 2.9 tabel sebelum di gabung baris
 Setelah digabung maka kondisi kode HTML menjadi seperti berikut:

Gambar 2.10 tabel setelah di gabung baris

```

    <table>
      <tr>
        <td colspan=2>.....</td>
        <td>.....</td>
      </tr>
      <tr>
        <td>.....</td>
        <td>.....</td>
      </tr>
      <tr>
        <td>.....</td>
        <td>.....</td>
      </tr>
    </table>
  
```

Contoh script pembuatan tabel

Buat lembar baru pada Notepad, kemudian ketikkan perintah di bawah ini.
 Simpan dengan nama **Contoh03.html**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title> Contoh 03 - Penggunaan Tabel</title>
</head>
<body>
<h1>Tabel Data Siswa</h1>
<table border="1" bgcolor="pink">
<tr>
<th>Nim</th>
<th>Nama</th>
<th>Alamat</th>
<th>Tempat, Tanggal Lahir</th>
<th>Jurusan</th>
</tr>
<tr>
<td>12110001</td>
<td>Anita</td>
<td>Cengkareng</td>
<td>Jakarta, 20 Agustus 1990</td>
<td>Sistem Informasi</td>
</tr>
<tr>
<td>12110002</td>
<td>Aditya</td>
<td>Tangerang</td>
<td>Semarang, 01 Januari 1989</td>
<td>Sistem Informasi</td>
</tr>
<tr>
<td>12110003</td>
<td>Firman</td>
<td>Bogor</td>
<td>Jakarta, 18 September 1988</td>
<td>Sistem Informasi</td>
</tr>
</table>
</body>
</html>

```

Jika dilihat di browser, maka terlihat sebagai berikut :



Contoh 03 - Penggunaan Tabel

localhost/modul_web1/contoh03.html

Tabel Data Siswa

Nim	Nama	Alamat	Tempat, Tanggal Lahir	Jurusan
12110001	Anita	Cengkareng	Jakarta, 20 Agustus 1990	Sistem Informasi
12110002	Aditya	Tangerang	Semarang, 01 Januari 1989	Sistem Informasi
12110003	Firman	Bogor	Jakarta, 18 September 1988	Sistem Informasi

Gambar 2.11 Hasil contoh03.html

Penggunaan *Cellpadding* dan *Cellspacing*

Cellpadding merupakan atribut dari tag `<table>` digunakan untuk mengatur jarak border bagian dalam dengan isi dalam tabel. Sedangkan *cellspacing* dipergunakan untuk mengatur jarak border bagian dalam dan luar. Satuan angka yang digunakan adalah pixel. Contoh `<table cellspacing="2">` maka artinya pengaturan jarak border bagian dalam dan luar sebanyak 2 pixel.

Buat lembar baru pada Notepad, kemudian ketikkan perintah di bawah ini.

Simpan dengan nama **tabelcell.html**

```
<!DOCTYPE html>
<html>
<head>
<title>Penggunaan atribut Cellpadding dan cellspacing dalam Tabel</title>
</head>
<body>
<h3>Belajar atribut cellpadding & cellspacing dalam Tabel</h3>
<table border="1" cellspacing="0" cellpadding="0">
  <tr>
    <td>Baris 1, Kolom 1</td>
    <td>Baris 1, Kolom 2</td>
    <td>Baris 1, Kolom 3</td>
  </tr>
  <tr>
    <td>Baris 2, Kolom 1</td>
    <td>Baris 2, Kolom 2</td>
    <td>Baris 2, Kolom 3</td>
  </tr>
</table>
<br />
```

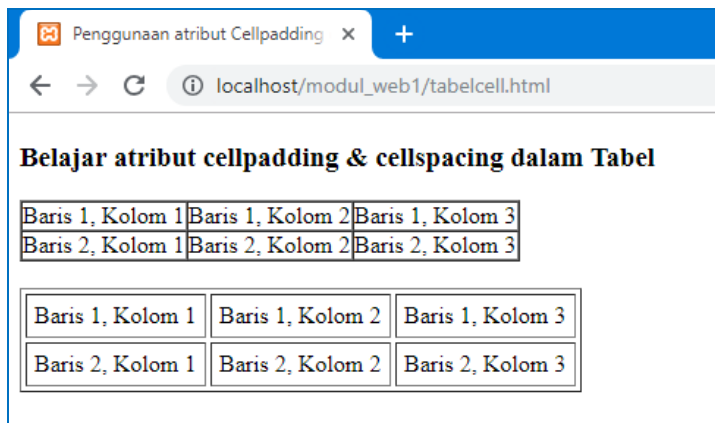


```

<table border="1" cellspacing="3" cellpadding="5">
  <tr>
    <td>Baris 1, Kolom 1</td>
    <td>Baris 1, Kolom 2</td>
    <td>Baris 1, Kolom 3</td>
  </tr>
  <tr>
    <td>Baris 2, Kolom 1</td>
    <td>Baris 2, Kolom 2</td>
    <td>Baris 2, Kolom 3</td>
  </tr>
</table>
</body>
</html>

```

Hasil di browser :



Gambar 2.12 Hasil dari tabelcell.html

BAB III

PHP

3.1 Pengertian PHP

PHP atau *PHP Hypertext Preprocessor* adalah sebuah bahasa script berbasis server (*server-side*) yang mampu mem-parsing kode php dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan website yang dinamis di sisi client (browser). Dengan menambahkan skrip PHP, anda bisa menjadikan halaman HTML menjadi lebih powerful, dinamis dan bisa dipakai sebagai aplikasi lengkap, misalnya web portal, *e-learning*, *e-library*, dll.

PHP pertama kali dikembangkan oleh seorang programmer bernama Rasmus Lerdorf pada tahun 1995. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakan PHP/FI sehingga banyak pemrograman yang tertarik untuk ikut mengembangkan PHP. Dan selanjutnya pada tahun 1997 perusahaan bernama Zend, mengembangkan interpreter PHP tersebut menjadi lebih baik.

Kode PHP diproses melalui pemrosesan dari sisi server, makanya PHP disebut skrip server-side. Sehingga kode PHP tidak bisa diberikan langsung ketika ada permintaan dari client(browser).

Kode PHP dimasukkan ke dalam kode HTML dengan cara menyelipkannya di dalam kode HTML. Untuk membedakan kode PHP dengan kode HTML, di depan kode PHP tersebut diberi tag pembuka dan diakhir kode PHP diberi tag penutup.

Dengan adanya kode PHP, sebuah halaman web bisa melakukan banyak hal yang dinamis, seperti mengakses database, membuat gambar, membaca dan menulis file, dan sebagainya. Hasil pengolahan kode PHP akan dikembalikan lagi dalam bentuk kode HTML untuk ditampilkan di browser.

Ada 4 jenis tag yang bisa digunakan untuk memasukkan kode PHP.

Jenis Tag	Tag Pembuka	Tag Penutup
Tag Standar	< ? php	? >
Tag Pendek	< ?	? >
Tag ASP	< %	% >
Tag Script	<script language = "php">	</script>

Yang dapat langsung diterapkan disemua platform adalah tag standard dan tag script. Di dalam modul ini bahasa pemrograman yang digunakan adalah PHP Versi 5 sehingga jenis tag yang harus digunakan adalah tag standar. Untuk tag lainnya perlu penyetingan di server oleh administrator server.

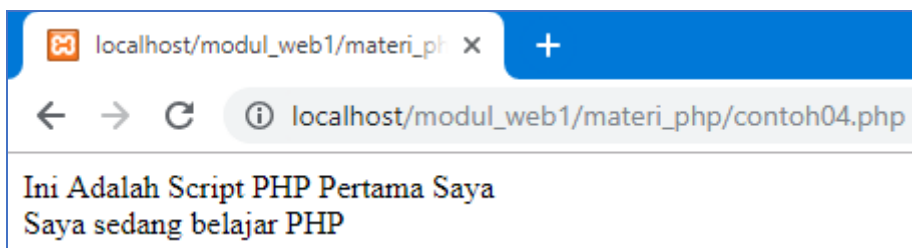
Contoh Script PHP

Buka file baru di Notepad. Kemudian ketikkan script seperti di bawah ini :

```
<?php  
echo "Ini Adalah Script PHP Pertama Saya <br>";  
echo "Saya sedang belajar PHP";  
?>
```

Simpan file dengan nama **contoh04.php**

Untuk melihat hasilnya buka browser masuk ke dalam localhost dan folder penyimpanan. Pilih file contoh04.php maka akan tampil hasilnya :

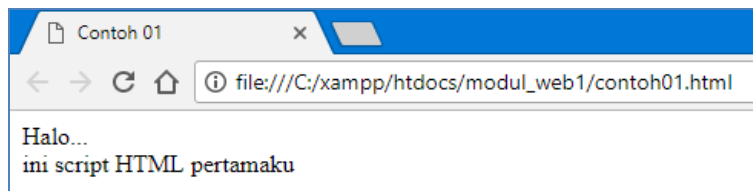


Gambar 3.1 Hasil contoh04.php

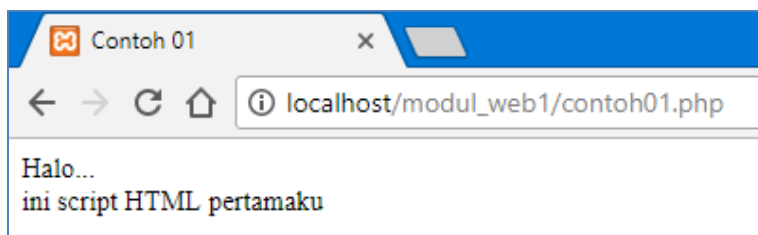
Contoh04.php merupakan contoh script php yang berdiri sendiri tanpa ada tambahan script yang lain. Perintah **echo** merupakan perintah yang digunakan untuk mencetak. Script PHP bisa juga digabung dalam tag HTML.

3.2 Perbedaan HTML dengan PHP

- HTML dapat diakses langsung tanpa melalui akses server saat ada permintaan dari client(*browser*)
- PHP harus di akses melalui server saat ada permintaan dari client(*browser*)



Gambar 3. 2 Contoh pemanggilan file html di browser



Gambar 3.3 Contoh pemanggilan file php di browser

Dari 2 gambar di atas dapatkah anda melihat perbedaannya, tanpa melihat extension nama filenya?

Ya, untuk file dengan extension html digambar 3.2, kita dapat melihat hasilnya langsung di browser, tanpa harus menjalankan akses server. Namun, untuk file dengan extension php digambar 3.3, kita harus

menjalankannya melalui akses server, yaitu localhost, dan penyimpanan filenya pun, disimpan pada htdocs yang ada di folder xampp

3.3 Dasar-dasar PHP

a. Variable

Variable merupakan sebuah istilah yang menyatakan sebuah tempat yang menampung nilai-nilai tertentu di mana nilai di dalamnya bisa diubah-ubah. Variable merupakan tempat untuk menyimpan data dalam tipe tertentu, variable bisa berupa null (belum ada isinya), angka, string, objek, array, Boolean, dan isinya bisa diubah-ubah nantinya. Variable penting karena tanpa adanya variable tidak bisa menyimpan nilai tertentu untuk diolah.

Variable ditandai dengan adanya tanda dolar (\$) yang kemudian bisa diikuti dengan angka, huruf, dan *underscore*. Namun variable tidak bisa mengandung spasi. Berikut ini contoh pendefinisian variable : \$nama, \$no_telp, \$pekerjaan. Untuk mendefinisikan variable, hanya perlu menuliskannya maka otomatis variable dikenali oleh PHP.

Contoh skrip Variable dalam PHP :

```

<html>
<head>
    <title> Contoh Script PHP</title>
</head>
<?php      //opening penulisan script php

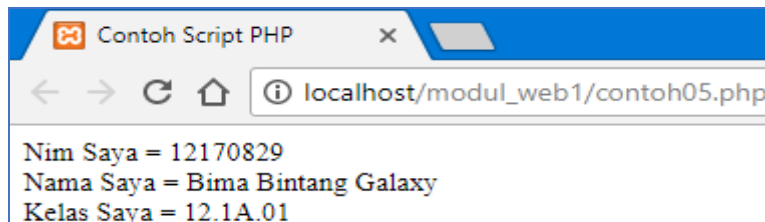
    //script dibawah ini adalah contoh deklarasi variabel
    $nim ="12170829";
    $nama ="Bima Bintang Galaxy";
    $kelas ="12.1A.01";

    //script dibawah ini digunakan sebagai output
    echo "Nim Saya = $nim<br>";
    echo "Nama Saya = $nama<br>";
    echo "Kelas Saya = $kelas<br>";

    //script dibawah ini adalah closing penulisan script php
    ?>
</body>
</html>

```

Hasil yang dihasilkan di browser :



Gambar 3.4 Hasil Contoh04.php

b. Tipe Data

Berbeda dengan bahasa pemrograman lain, variable di PHP lebih fleksibel. Kita tidak perlu mendefinisikan jenisnya ketika mendefinisikan pertama kali. Ada 6 Tipe data dasar yang dapat diakomodasi di PHP, seperti berikut ini :

Type	Contoh	Penjelasan
Integer	134	Semua angka bukan pecahan
Double	5.1234	Nilai pecahan
String	“asep”	Kumpulan karakter
Type	Contoh	Penjelasan
Boolean	False	Salah satu nilai True atau False
Object		Sebuah instance dari class
Array		Larik

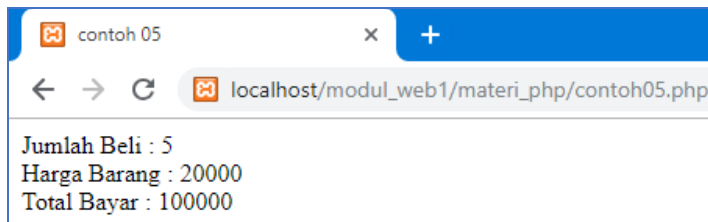
Contoh05.php:

```

<html>
<head>
    <title> contoh 05 </title>
</head>
<body>
<?php
    $jumlah = 5;
    $harga = 20000;
    $total=$harga*$jumlah;
echo "Jumlah Beli : $jumlah <br>";
echo "Harga Barang : $harga <br>";
echo "Total Bayar : $total <br>";
?>
</body>
</html>

```

Hasil Tampilan di browser :



Gambar 3.5 Hasil dari contoh05.php

c. **Konstanta**

Selain variable, sebuah program umumnya juga memungkinkan adanya konstanta. Konstanta fungsinya sama seperti variable namun nilainya statis/konstan dan tidak bisa berubah. Cara untuk mendefinisikan konstanta adalah :

```
Define ("NAMA_KONSTANTA", nilai_konstanta);
```

Setelah didefinisikan, kita dapat langsung menggunakannya dengan mengetikkan nama konstanta tersebut. Nama konstanta umumnya diketik menggunakan huruf besar.

d. **Komentar**

Program merupakan kegiatan menuliskan bahasa yang dipahami oleh mesin. Walaupun bahasa yang digunakan adalah bahasa tingkat tinggi, namun tentu masih tidak semudah dipahami oleh bahasa biasa. Untuk itu kita bisa menggunakan komentar. Berikut ini contoh pembuatan komentar di php.

```
//komentar satu baris
```

```
#ini juga komentar satu baris
```

```
/*komentar
```

```
Banyak baris
```

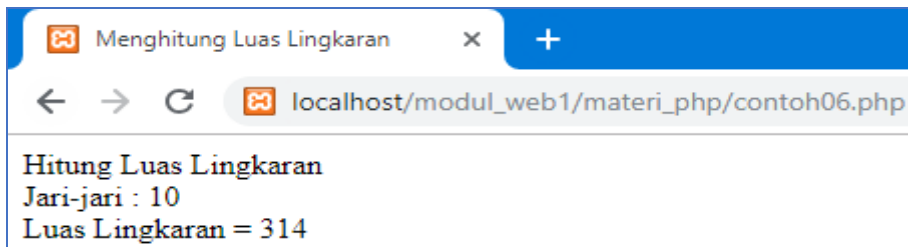
```
Kode di sini tidak Dieksekusi oleh parser */
```


Contoh penggunaan Konstanta dan Komentar : contoh06.php

```
<html>
<head>
    <title> Menghitung Luas
    Lingkaran</title>
</head>
<body>
<?php
//konstanta untuk nilai judul
define("Judul","Hitung Luas
Lingkaran");
//konstanta untuk nilai phi
define("PHI",3.14);
```

```
echo "<br> Jari-jari : $r <br>";
$luas = PHI*$r*$r;
echo "Luas Lingkaran = $luas";
?>
</body>
</html>
```

Hasil tampilan di browser :



Gambar 3.6 Hasil tampilan contoh06.php

BAB IV

OPERATOR

Deskripsi:

Membahas penggunaan jenis-jenis operator yang ada dalam bahasa pemrograman web dan dapat mengimplementasikan masing-masing operator tersebut

Tujuan Pembelajaran:

Setelah memahami pokok bahasan pada bab ini diharapkan mahasiswa mampu:

1. Mengetahui jenis-jenis operator
2. Mengimplementasikan masing-masing operator tersebut dalam bahasa pemrograman web

4.1. Mengetahui Operator

Sebuah bahasa pemrograman juga wajib untuk mampu mengolah nilai operator (Variable atau konstanta yang dioperasikan) menggunakan operator, seperti menjumlah, membagi, dan sebagainya.

Operator merupakan simbol yang berfungsi untuk melakukan aksi/operasi tertentu terhadap nilai operator yang pada umumnya dari hasil operasi tersebut menghasilkan nilai baru. Sementara operator adalah nilai yang dilibatkan dalam operasi oleh operator.

4.2. Jenis-Jenis Operator

a. Operator Aritmatika

Operator ini digunakan untuk melakukan perhitungan matematika,sebagian berikut:

Operator	Nama	Contoh	Hasil
+	Penambahan	1+4	5
-	Pengurangan	1-4	-3
/	Pembagian	1/4	0.25
*	Perkalian	1*4	4
%	Sisa Pembagian	5%2	1
++	Inkremen	X=5; X++	X=6
-	Dekremen	X=5; X--	X=4

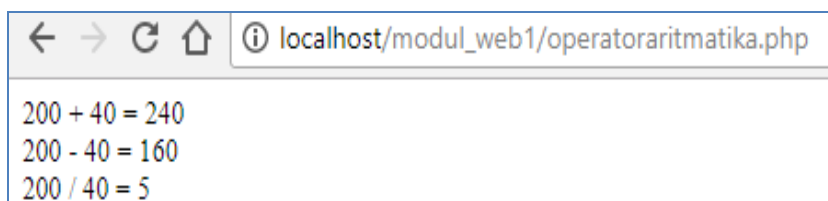
Contoh skrip :

Operatoraritmatika.php

```
<?php
$bil1=200;
$bil2=40;
$hasil=$bil1+$bil2;
echo "$bil1 + $bil2 = $hasil<br>";
$hasil=$bil1-$bil2;
echo "$bil1 - $bil2 = $hasil<br>";
$hasil=$bil1/$bil2;
echo "$bil1 / $bil2 = $hasil<br>";
?>
```

Hasil :

Hasil dari skrip diatas seperti gambar berikut:



Gambar 4.1 hasil Operatoraritmatika

b. Operator Perbandingan

Operator perbandingan digunakan untuk menghasilkan 2 nilai yang hasil akhirnya adalah nilai *Boolean true* dan *false*. Operator ini

sangat berguna dalam pemrograman karena bisa menentukan arah pemrograman. Operator perbandingan di PHP adalah :

Operator	Nama	Contoh	Hasil
==	Sama dengan	6 == 6	False
!=	Tidak sama dengan	3 != 3	False
>	Lebih besar	1 > 5	False
>=	Lebih besar atau sama dengan	3 >= 4	False
Operator	Nama	Contoh	Hasil
<	Lebih kecil	2 < 4	True
<=	Lebih kecil atau sams dengan	5 <= 4	False

Opertorperbandingan.php

```
<?php
$bil1=200;
$bil2=40;
$teks1="PHP";
$teks2="php";

$hasil=$bil1==$bil2;
echo "$bil1==$bil2 = $hasil <br>";

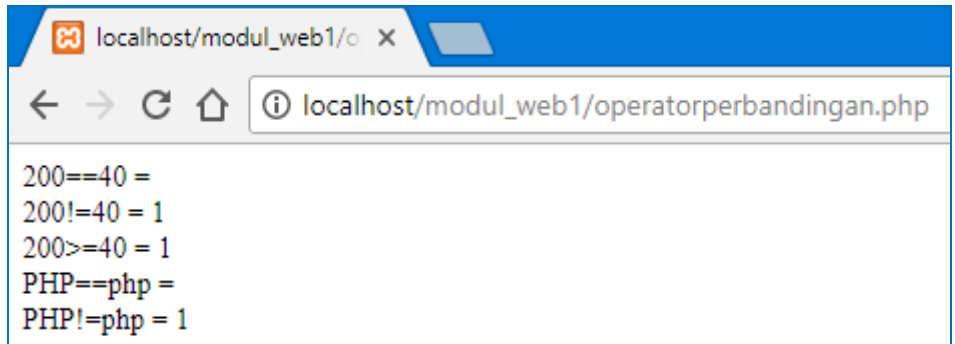
$hasil=$bil1!=$bil2;
echo "$bil1!=$bil2 = $hasil <br>";

$hasil=$bil1>=$bil2;
echo "$bil1>=$bil2 = $hasil <br>";

$hasil=$teks1==$teks2;
echo "$teks1==$teks2 = $hasil <br>";

$hasil=$teks1!= $teks2;
echo "$teks1!= $teks2 = $hasil <br>";
?>
```

Hasil tampilan di browser :



Gambar 4.2 hasil Operatorperbandingan.php

c. Operator Logika

Operator untuk menyusun kalimat ekspresi/ungkapan logika. Hasil operasi ini akan didapatkan nilai satu jika benar dan nol jika salah.

Operator	Fungsi
AND atau &&	Operasi logika AND
OR atau	Operasi logika OR
XOR	Operasi logika eksklusife OR
!	Ingkaran/negasi

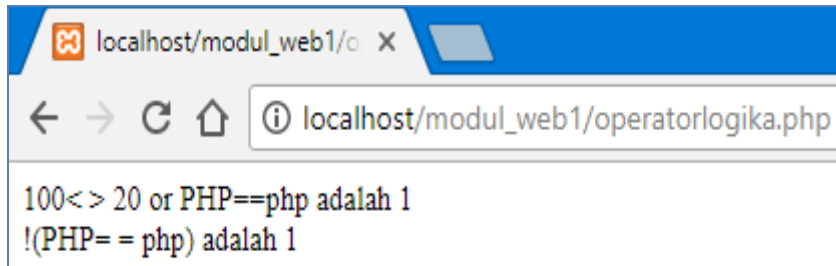
Operatorlogika.php

```
<?php
$bil1=100;
$bil2=20;
$teks1="PHP";
$teks2="php";

$hasil=($bil1<> $bil2) or ($teks1==$teks2);
echo " $bil1<> $bil2 or $teks1==$teks2 adalah $hasil <br>";

$hasil=!($teks1== $teks2);
echo "!($teks1== $teks2) adalah $hasil <br>";
?>
```

Hasil dari skrip diatas seperti gambar berikut:



Gambar 4.3 hasil operatorlogika.php

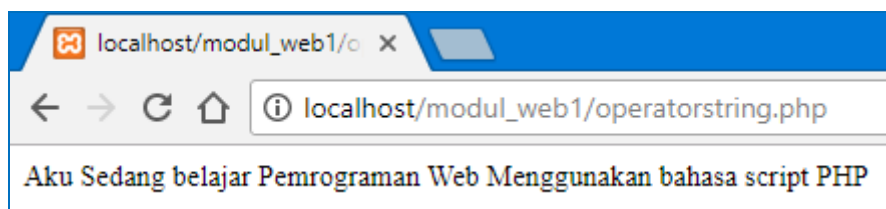
d. Operator String

Dalam PHP juga tersedia operator string, yaitu digunakan untuk operasi penggabungan teks. Adapun simbol yang digunakan yaitu berupa karakter titik (.).

Operatorstring.php

```
<?php
$teks1="Aku Sedang belajar ";
$teks2="Pemrograman Web ";
$teks3="Menggunakan bahasa script PHP ";
$hasil=$teks1 . $teks2 . $teks3;
echo "$hasil ";
?>
```

Hasil dari skrip diatas seperti gambar berikut:



Gambar 4.4 hasil Operatorstring.php

Latihan

Buat skrip menggunakan bahasa pemrograman PHP untuk menghitung volume Kubus menggunakan fungsi operator aritmatika dan operator string dengan ketentuan sebagai berikut :

Panjang sisi kubus = 15cm

Hitung volume balok dengan rumus = sisi x sisi x sisi

Buat variable teks1 yang berisi ="Belajar Menghitung" dan teks2 yang berisi ="Volume Kubus".

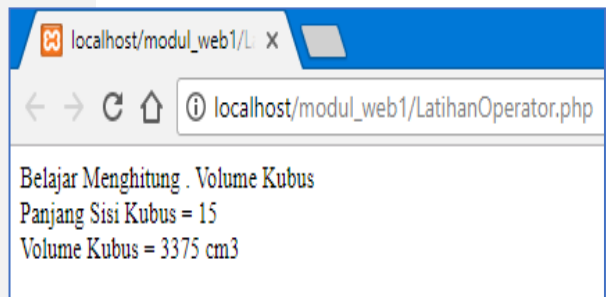
Buat perintah untuk menggabungkan nilai dari Variable teks1 dan teks2 menggunakan operator string.

Jawaban latihan:

Hasil dari skrip latihan adalah:

```
<?php
$sisi=15;
$teks1="Belajar Menghitung";
$teks2="Volume Kubus";
$volume = $sisi*$sisi*$sisi;

echo"$teks1 . $teks2<br>";
echo"Panjang Sisi Kubus =
$sisi<br>";
echo"Volume Kubus = $volume
cm3<br>";
?>
```



Gambar 4.5 hasil dari jawaban latihan

BAB V

PENGENALAN FORM DAN PENGGUNAAN HTTP SERVER

Deskripsi:

Membahas komponen form, pengolahan data dari form yang ada dalam bahasa pemrograman web, mempraktikkan penggunaan HTTP SERVER dengan metode GET dan POST.

Tujuan Pembelajaran:

Setelah memahami pokok bahasan pada bab ini diharapkan mahasiswa mampu:

1. Membuat tampilan form dalam bahasa pemrograman web
2. Menggunakan metode get dan post dalam mengirim data.

5.1 Komponen *Form*

Sebuah *website* dinamis seringkali memerlukan interaksi antara *browser client* dan *server* bisa berupa pemasukan data teks, angka, atau *upload* file untuk diproses oleh *server*. Untuk memudahkan suatu data yang dikirimkan oleh *browser client*, dibutuhkan adanya FORM HTML. Penggunaan *form* misalnya untuk pendaftaran keanggotaan, pemasukan kode kartu kredit, *login user*, transaksi perbelanjaan, dan *upload* file.

5.2. Pengolahan Data Dari Form

Form di HTML dikenal dengan adanya tag <FORM> dan ditutup dengan tag </FORM>. Di dalam tag pembuka <FORM> diikuti dengan atribut *action* dan *method*. Pada atribut *action* menerangkan ke halaman

yang akan digunakan pada proses *input*, sedangkan method digunakan untuk mengatur cara mem-parsing konten.

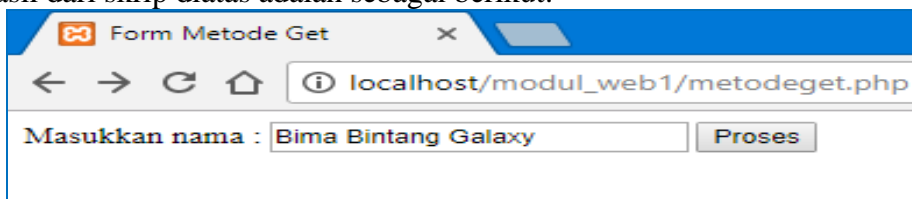
Pada halaman web akan menerima inputan dari *user* atau para pengunjung dengan menggunakan metode GET dan POST. GET akan mengirimkan data bersama dengan URL, sedangkan POST akan mengirimkannya secara terpisah. *User* mengirimkan data *input* dengan mengisi teks atau pilihan pada attribut form html.

a. Proses Form menggunakan Metode GET.

File **metodeget.php**

```
<html>
<head>
    <title> Form Metode Get</title>
</head>
<body>
<form action="metodegetproses.php" method="get">
    Masukkan nama : <input type = "text" name="nama" size="25">
<input type="submit" value="Proses">
</form>
</body>
</html>
```

Hasil dari skrip diatas adalah sebagai berikut:



Gambar 5.1 hasil metodeget.php

Buat file untuk memproses Variable yang diberikan oleh file metodeget.php, beri nama filenya : **metodegetproses.php**

```
<html>
<head>
    <title> Form Metode Get Proses</title>
</head>
<body>
Data Nama Yang Diinputkan adalah : <?php echo
$ GET["nama"]?>
```

Hasil dari skrip diatas adalah sebagai berikut:



Gambar 5.2 hasil metodegetproses.php

Penjelasan dari gambar hasil metodegetproses adalah:

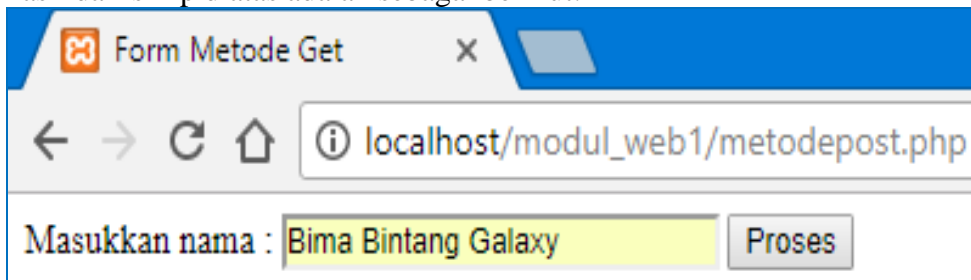
Perhatikan tulisan pada url di web browser karena menggunakan metode GET, maka data yang dikirimkan akan ditampilkan di URL

b. Proses Form menggunakan metode : POST

Untuk membuat inputan, dan beri nama file : metodepost.php

```
<html>
<head>
    <title> Form Metode Get</title>
</head>
<body>
    <form action="metodepostproses.php" method="post">
        Masukkan nama : <input type="text" name="nama" size="25">
        <input type="submit" value="Proses">
    </form>
</body>
</html>
```

Hasil dari skrip diatas adalah sebagai berikut:

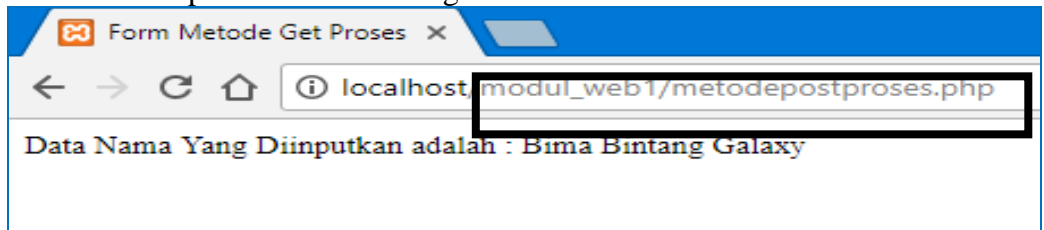


Gambar 5.3 hasil metodepost.php

Buat file untuk memproses Variable yang diberikan oleh file metodepost.php beri nama filenya : **metodepostproses.php**

```
<html>
<head>
    <title> Form Metode Get Proses</title>
</head>
<body>
    Data Nama Yang Diinputkan adalah : <?php echo $_POST["nama"];?>
</body>
</html>
```

Hasil dari skrip diatas adalah sebagai berikut:



Gambar 5.4 hasil metodepostproses.php

Penjelasan dari gambar hasil metodegetproses adalah:

Perhatikan tulisan pada url di web browser karena menggunakan metode POST, maka data yang dikirimkan akan ditampilkan di URL.

c. Contoh Pembuatan Form dengan Input, Proses, dan Output Dalam Satu File

Untuk membuat form *input* dan halaman untuk menampilkan dalam satu file, maka bisa menggunakan statement :

```
If (!Empty (nama_Variable))
```

Artinya jika Variable yang dicari tidak kosong (alias ada) maka baru ditampilkan, sementara jika tidak ada maka tidak akan ditampilkan.

Yang perlu diketahui adalah digunakannya alamat action :

```
$_server ['php_self']
```

Artinya alamat action akan mengacu ke halaman itu sendiri dan bukan di halaman yang lain. Dengan demikian, jika form di submit, maka halaman yang dipanggil tetap halaman yang sama.

Contoh formdatadiri.php

```
<html>
<head>
    <title> Masukan Data</title>
</head>
<body>
<h1> Masukan Identitas Anda</h1>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<pre>
Isikan Nama   : <input type="text" name="nama">
Isikan No Telp: <input type="text" name="notelp">
Isikan Alamat : <textarea name="alamat" rows="5" cols="40"></textarea>
<input type="submit" value="TAMPIL"><input type="reset"
value="BATAL">
</pre>
```

```
<?php
$nama = $_POST['nama'];
$telp = $_POST['notelp'];
$alamat = $_POST['alamat'];
if(!empty($nama)){
echo "Nama   : $nama <br>"; }
if(!empty($telp)){
echo "No Telp : $telp <br>"; }
if(!empty($alamat)){
echo "Alamat : $alamat <br>"; }
?>
</body>
</html>
```

Hasil dari skrip diatas adalah sebagai berikut:



The screenshot shows a web browser window with the title 'Masukan Data' and the URL 'localhost/modul_web1/FormDataDiri.php'. The page has a heading 'Masukan Identitas Anda'. Below the heading are three labels with corresponding input fields: 'Isikan Nama' with the value 'Bima Bintang Galaxy', 'Isikan No Telp' with the value '08791234567', and 'Isikan Alamat' with the value 'Jalan Anggrek No. 9 Bandung'. At the bottom left of the form are two buttons: 'TAMPIL' and 'BATAL'.

Gambar 5.5 hasil formdatadiri.php

Selanjutnya setelah ada hasil formdatadiri maka KLIK Tombol Tampil, maka akan tampil kiriman datanya.



This screenshot shows the same web form after the 'TAMPIL' button was clicked. The input fields are now empty. Below the input fields, the submitted data is displayed in a bold font: 'Nama : Bima Bintang Galaxy', 'No Telp : 08791234567', and 'Alamat : Jalan Anggrek No. 9 Bandung'. The 'TAMPIL' and 'BATAL' buttons remain at the bottom left.

Gambar 5.6 hasil kiriman data formdatadiri

Latihan

1. Buat skrip program untuk membuat form *input* disimpan dengan nama **forminputmahasiswa.php** sebagai berikut ini:

Pengelolaan Data Mahasiswa

Nama : Bima Bintang Galaxy

Alamat : Jakarta

Jenis Kelamin : ☒ Laki - Laki ☐ Perempuan

Pekerjaan : Pelajar

Hobi : ☒ Olahraga ☒ Musik ☒ Jalan-Jalan

Gambar 5.7 Tampilan forminputmahasiswa.php

2. Buat skrip program simpan dengan nama **tampilmahasiswa.php** untuk memanggil data dari form *input* dengan bentuk sebagai berikut ini:

Data Mahasiswa	
Nama	Bima Bintang Galaxy
Alamat	Jakarta
Jenis Kelamin	Laki-Laki
Pekerjaan	Pelajar
Hobi	Olahraga, Musik, Jalan-Jalan

[INPUT DATA LAGI](#)

Gambar 5.8 tampilmahasiswa.php

Jawaban latihan

1. forminputmahasiswa.php

```
<html>
<head>
    <title>Input Data Mahasiswa</title>
</head>
<body bgcolor="green">
<form action="tampilmahasiswa.php" method="post">
<b>Pengelolaan Data Mahasiswa</b>
<br>
<pre>
Nama : <input type="text" name="nama" size="25"
maxlength="50">
Alamat : <input type="text" name="alamat" size="25"
maxlength="50">
</pre>
Jenis Kelamin :
<input type="radio" name="jeniskel" value="Laki-Laki"> Laki -
Laki
<input type="radio" name="jeniskel" value="Perempuan">
Perempuan
<p>
Pekerjaan :
<select name="pekerjaan">
<option value="-Pilih-">
<option value="Pelajar">Pelajar
<option value="Karyawan">Karyawan
<option value="Wirausaha">Wirausaha
<option value="Lain-lain">Lain-lain
</select>
<p>
Hobi :
<input type="checkbox" name="hobi1"
value="Olahraga">Olahraga
<input type="checkbox" name="hobi2" value="Musik">Musik
<input type="checkbox" name="hobi3" value="Jalan-Jalan">Jalan-
Jalan
<p>
<input type="submit" value="Kirim"><input type="reset"
value="Batal">
</form>
</body>
</html>
```

2. tampilmahasiswa.php

```
<html>
<head>
    <title> Data Mahasiswa </title>
</head>
<body>

<?php
$nama=$_POST['nama'];
$alamat=$_POST['alamat'];
$jeniskel=$_POST['jeniskel'];
$pekerjaan=$_POST['pekerjaan'];
$hobi1=$_POST['hobi1'];
$hobi2=$_POST['hobi2'];
$hobi3=$_POST['hobi3'];
?>

<table border=1 bgcolor="Cyan">
<tr>
<td colspan=2 align="center"><b>Data Mahasiswa</b></td>
</tr>
<tr>
<td>Nama</td><td><?php echo $nama; ?></td>
</tr>
<tr>
<td>Alamat</td><td><?php echo $alamat; ?></td>
</tr>
<tr>
<td>Jenis Kelamin</td><td><?php echo $jeniskel; ?></td>
</tr>
<tr>
<td>Pekerjaan</td><td><?php echo $pekerjaan; ?></td>
</tr>
<tr>
<td>Hobi</td><td><?php echo $hobi1,",", $hobi2,",", $hobi3; ?></td>
</tr></table>
<a href="forminputmahasiswa.php">INPUT DATA LAGI</a>

</body>
</html>
```


BAB VI

PERCABANGAN

Deskripsi:

Membahas konsep percabangan dalam bahasa pemrograman web

Tujuan Pembelajaran:

Setelah memahami pokok bahasan pada bab ini diharapkan mahasiswa mampu:

1. Menggunakan perintah percabangan if tunggal
2. Menggunakan perintah percabangan if dan else
3. Menggunakan perintah percabangan if majemuk
4. Menggunakan perintah switch

6.1 Pernyataan Seleksi

Sebagian besar bahasa pemrograman mengandung pernyataan seleksi. Pada dasarnya pernyataan seleksi adalah suatu mekanisme yang menjelaskan apakah pernyataan akan dikerjakan atau tidak, hal ini tergantung kondisi yang dirumuskan. Dalam bahasa pemrograman PHP pernyataan seleksi diterapkan dengan menggunakan statement IF dan Switch Case.

a. Statement IF

1. If Tunggal

Statement IF merupakan statement yang penting dan pasti terdapat di semua bahasa pemrograman. Statement ini berguna untuk membuat percabangan berdasarkan kondisi tertentu yang harus dipenuhi.

Bentuk umum: if (kondisi)
 {
 statement;
 }

Prinsip kerjanya adalah perintah di atas akan dikerjakan apabila kondisi bernilai TRUE atau benar, sedangkan jika kondisi salah/FALSE maka statement di atas tidak akan dikerjakan.

2. Pernyataan If dan Else

Pernyataan ELSE merupakan bagian dari pernyataan if. Else digunakan untuk memberikan alternatif perintah apabila kondisi bernilai salah / FALSE.

Bentuk umum :

```
if ( kondisi )
{
    statement_1;
}
else
{
    statement_2;
}
```

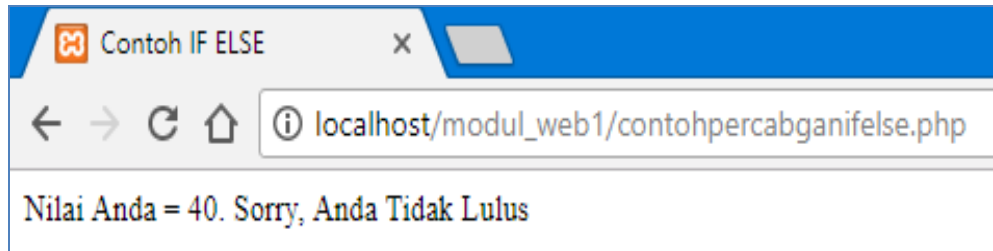
Contoh dari pernyataan if dan else adalah sebagai berikut:

```
<html>
<head>
    <title> Contoh IF ELSE</title>
</head>

<?php
    $nilai = 40;
    if ($nilai >= 60)
    {   echo "Nilai Anda = $nilai. Selamat, Anda Lulus" ; }
    else
    {   echo "Nilai Anda = $nilai. Sorry, Anda Tidak Lulus" ; }
?>

</body>
</html>
```

Hasil dari skrip diatas adalah sebagai berikut:



Gambar 6.1. hasil pernyataan if dan else

3. Pernyataan IF Majemuk

Jika pernyataan else memberikan alternatif pilihan kedua, maka untuk pernyataan ElseIf dapat digunakan untuk merumuskan banyak alternatif pilihan (lebih dari dua pilihan).

```
Bentuk umum : if ( kondisi_1)
                {
                    statement_1;
                }
                elseif ( kondisi_2)
                {
                    statement_2;
                }
                elseif ( kondisi_3)
                {
                    statement_3;
                }
                else
                {
                    statement_n;
                }
```

Contoh dari pernyataan if majemuk adalah sebagai berikut:

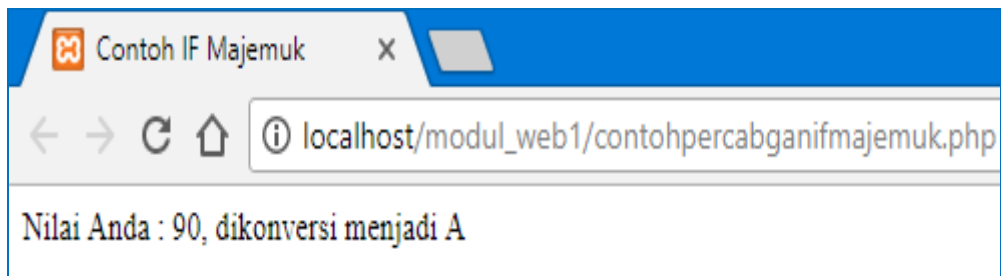
```

<html>
<head>
    <title> Contoh IF Majemuk</title>
</head>

<?php
    $nilai = 90;
    if (($nilai >= 0)&&($nilai < 50))
    { $grade = "E";}
    elseif(($nilai >= 50)&&($nilai < 60))
    { $grade = "D";}
    elseif(($nilai >= 60)&&($nilai < 75))
    { $grade = "C";}
    elseif(($nilai >= 75)&&($nilai < 85))
    { $grade = "B";}
    elseif(($nilai >= 85)&&($nilai < 100))
    { $grade = "A";}
    else
    { $grade = "Nilai anda di luar jangkauan"; }
    echo "Nilai Anda : $nilai, dikonversi menjadi
    $grade";
?>

```

Hasil dari skrip pernyataan if majemuk adalah sebagai berikut:



Gambar 6.2. hasil pernyataan if majemuk

b. Statement Switch

Statement untuk pengatur alur program berikutnya adalah switch. Salah satu keuntungan switch adalah ada bisa langsung mengevaluasi satu statement dan memerintahkan aksi dalam jumlah yang lebih banyak.

Bentuk umum :

```

switch ( nilai_ekspresi ){
    case nilai_1: statement_1; break;
    case nilai_2: statement_2; break;
    default: statement_n; }

```

Contoh dari statement Switch adalah sebagai berikut:

```
<html>
<head>
    <title> Contoh IF Majemuk</title>
</head>

<?php
    $angka = 6;
    switch($angka) {
        case 0 : $terbilang = "NOL"; break;
        case 1 : $terbilang = "SATU"; break;
        case 2 : $terbilang = "DUA"; break;
        case 3 : $terbilang = "TIGA"; break;
        case 4 : $terbilang = "EMPAT"; break;
        case 5 : $terbilang = "LIMA"; break;
        case 6 : $terbilang = "ENAM"; break;
        case 7 : $terbilang = "TUJUH"; break;
        case 8 : $terbilang = "DELAPAN"; break;
        case 9 : $terbilang = "SEMBILAN"; break;
        default : $terbilang = "Nilai di luar jangkauan"; break;
    }
    echo "Bentuk Terbilang dari angka $angka adalah $terbilang";
?>
</body>
</html>
```

Hasil dari skrip statement switch adalah sebagai berikut:



Gambar 6.3 hasil statement switch

Latihan

Buatlah skrip sehingga mendapatkan tampilan sebagai berikut:

Tampilan input

PENERIMAAN BEASISWA

Nama :

NIM :

Kelas :

Jurusan : SI ▼

Jenis Kelamin : ☐ Pria ☐ Wanita

IPK :

Gambar 6.4 Rancangan Tampilan Input Latihan

Tampilan output:

Data Penerima Beasiswa

Nama : Sandrina Ghaisani Ismail

Nim : 12035894

Kelas : Pagi

Jurusan : Sistem

Jenis kelamin : Wanita

IPK : 3,50

Menyatakan bahwa Sandrina Ghaisani Ismail Mendapat beasiswa sebesar Rp.1.000.000

[Kembali](#)

Gambar 6.5 Rancangan Tampilan Output Latihan

Ketentuan Soal :

- ✚ Jika $ipk \geq 3.0$ maka keterangannya= mendapat beasiswa sebesar Rp.1000.000 “
Selain itu tidak dapat beasiswa
- ✚ Jika klik **Kembali Ke Awal** akan kembali ke halaman input

Jawaban latihan:

1. Tampilan input

```
<html>
<head>
  <title>Info Beasiswa</title>
</head>
<body>
  <form action="beasiswa_output.php" method="post">
    <h2>PENERIMAAN BEASISWA</h2>
    <pre>
Nama                : <input type="text" name="nama">
NIM                 : <input type="text" name="nim">
Kelas              : <input type="text" name="kelas">
Jurusan             : <Select name=jurusan>
  <option value=Sistem Informasi>SI</option>
  <option value=Tehnik Informatika>TI</option>
</select>
Jenis Kelamin       : <input type="radio" value=Pria
name="jkel">Pria
                    <input type="radio" value=Wanita
name="jkel">Wanita
IPK                  : <input type="text" size=4
name="ipk">
```

2. Untuk tampilan output:

```
<html>
<head>
  <title>Beasiswa</title>
</head>
<body>
  <h2> Data Penerima
Beasiswa</h2>
  <pre>
<?php
$nama=$_POST["nama"];
$nim=$_POST["nim"];
$kelas=$_POST["kelas"];
$jurusan=$_POST["jurusan"];
```

```
if($ipk>=3.0)
{
  $ket="Mendapat beasiswa sebesar
Rp.1.000.000";
}
else
{
  $ket="Tidak dapat beasiswa"; }
echo "Nama                : $nama<br>";
echo "Nim                 : $nim<br>";
echo "Kelas              : $kelas<br>";
echo "Jurusan             : $jurusan <br>";
echo "Jenis kelamin       : $jkel <br>";
echo "IPK                  : $ipk<br>";
echo "Menyatakan bahwa $nama $ket";
?>
<br>
<a href=beasiswa_input.php>Kembali</a>
</pre>
</body></html>
```

BAB VII

PERULANGAN

Deskripsi:

Membahas pengertian dasar struktur perulangan atau dikenal juga dengan istilah loop, mempraktikkan cara penggunaan perulangan for, while, do while dan Foreach.

Tujuan Pembelajaran:

Setelah memahami pokok bahasan pada bab ini diharapkan mahasiswa mampu:

3. Memahami pengertian dasar perulangan
4. Mengetahui jenis-jenis perulangan dalam pemrograman
5. Mengimplementasikan jenis-jenis perulangan tersebut dalam bahasa pemrograman web.

7.1. Pengertian Dasar Perulangan

Perulangan atau looping. Ketika kita mempelajari bahasa pemrograman pasti akan menemui dengan yang dinamakan perulangan, terutama perulangan pada php. Hal yang tentunya wajib dikuasai dalam bahasa pemrograman salah satunya mengenai perulangan. Banyak algoritma yang tentunya mengandalkan proses perulangan atau looping guna terciptanya sarana algoritma seperti menggunakan array, searching, filtering dan banyak lagi.

7.2. Pengertian Dasar Perulangan Pada PHP

perulangan atau looping yaitu digunakan untuk mempermudah melakukan proses yang berulang-ulang sesuai dengan perintahnya. Contoh kecilnya adalah, apabila kita mengurutkan angka 01 sampai 50, maka hal yang paling mudah dalam pemrograman ialah dengan menggunakan perulangan, yaitu dengan Algoritma menambahkan nilai 1 pada setiap angka yang diulang.

Misalnya di awali dengan 1, ditambah dengan 1 maka akan memperoleh nilai 2, dan seterusnya hingga mencapai 50.

Ada beberapa jenis metode perulangan atau looping yang bisa digunakan dalam pembuatan program menggunakan bahasa PHP diantaranya yaitu:

for, while, do while dan Foreach.

a. Perulangan FOR pada PHP

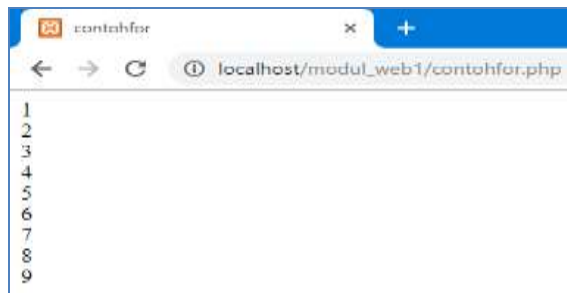
Perulangan for Biasanya Dipakai untuk mengulang suatu pernyataan sebanyak yang kita mau. Penggunaan sintak for pada bahasa pemrograman php yaitu dengan menentukan kondisi jumlah atau nilai yang ingin diulang “Variable = nilai; Variable < batas; Variable++”.

Contohfor.php

Ketikan koding berikut:

```
<html>
<head>
    <title>contohfor </title>
</head>
<body>
<?php
for($i = 1; $i <= 9; $i++)
{
    echo $i . '<br />';
}
?>
</body>
</html>
```

Hasil Dari kode perulangan for pada php diatas akan mencetak angka 1 – 9, seperti gambar dibawah ini:



Gambar 7.1 hasil Perulangan Menggunakan For

Penjelasan Koding Diatas:

`$i = 1`

Variable `$i` memiliki nilai 1

`$i <= 9`

Lalu disini kita akan melakukan batas perulangan Variable `$i` dengan menggunakan [operator perbandingan](#) pada php

`$i++`

Selanjutnya disinilah nilai `$i` akan berubah dengan menambahkan setiap angka dimulai dari 1 ditambah dengan 1 dengan menggunakan metode [post-increment](#), tetapi looping pertama tidak akan terjadi proses penambahan dan angka 1 akan tetap dicetak sebagai angka 1.

b. Perulangan While pada PHP

Perulangan while mirip dengan for, namun untuk menentukan kondisinya hanya dengan menggunakan [operator perbandingan](#) saja, misalnya `$i <= batas`.

Buat file baru didalam folder htdoc/perulangan/**while.php**

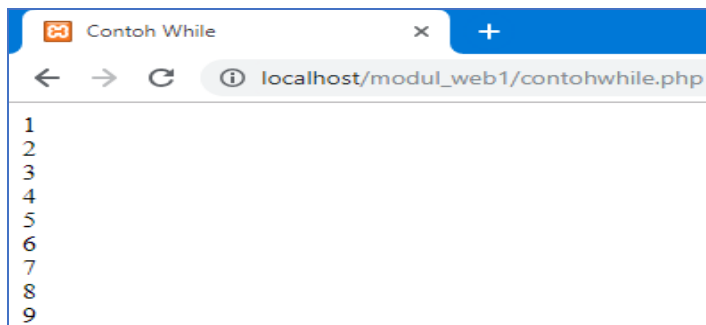
Ketikan koding berikut:

```

<html>
<head>
    <title> Contoh while</title>
</head>
<body>
<?php
$i = 1;
while($i <= 9)
{
    echo $i . '<br />';
    $i++;
}
?>
</body>
</html>

```

Hasil Dari kode perulangan for pada php diatas akan mencetak angka 1 – 9, seperti gambar dibawah ini:



Gambar 7.2 hasil Perulangan Menggunakan While

Ket: perbedaan yang mencolok dengan menggunakan for adalah kondisi tidak ditulis didalam fungsi perulangannya, dan proses penambahan angka dengan menggunakan metode [post-increment](#) ditulis di dalam proses perulangan.

c. Perulangan Do-While pada PHP

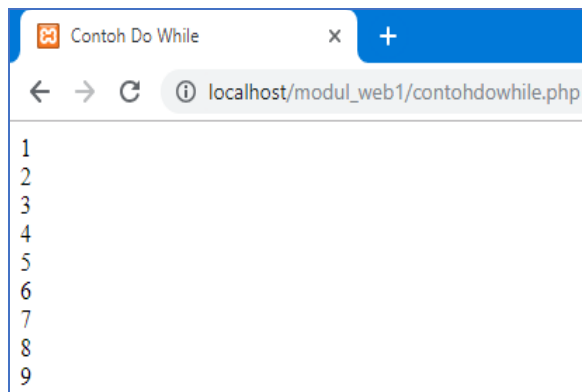
Perbedaan yang mendasar perulangan **Do-While** dengan **While** adalah dimana yang dilakukan adalah dengan mengeksekusi pernyataan terlebih dahulu, setelah itu membaca kondisi,

Buat file baru didalam folder htdoct/perulangan/**dowhile.php**

Ketikan koding berikut:

```
<html>
<head>
    <title> Contoh Do While</title>
</head>
<body>
<?php
    $i = 1;
    do{
        echo $i . '<br />';
        $i++;
    }
    while($i <= 9);
?>
</body>
</html>
```

Hasil tampilan di browser :



Gambar 7.3 hasil Perulangan Menggunakan Do-While

Sebenarnya yang menentukan perhitungan adalah posisi operator *post-increment* (\$i++), yaitu Variable akan dicetak terlebih dahulu baru ditambahkan nilai 1. Namun jika posisi *post-increment* (\$i++) berada diatas perintah cetak maka yang terjadi adalah program akan terlebih dahulu menambahkan nilai satu pada Variable \$i, misalnya:

```

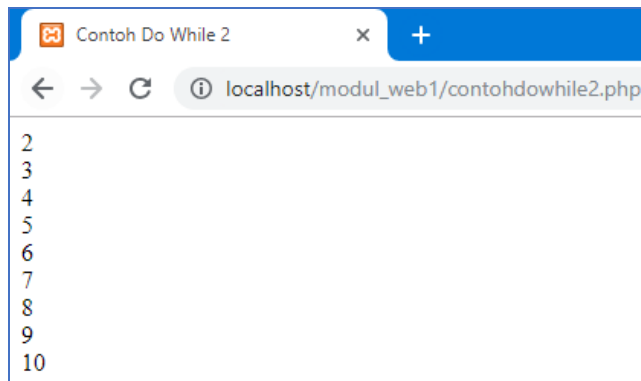
<html>
<head>
    <title> Contoh Do While 2</title>
</head>

<body>
<?php
$i = 1;
do{
    $i++;
    echo $i . '<br />';
}
while($i <= 9);

?>
</body>
</html>

```

Dari contoh perpindahan operator *post-increment* diatas maka akan tercetak seperti gambar dibawah ini:



Gambar 7.4 hasil Perulangan Menggunakan Do-While 2

Terlihat terjadi perubahan proses, yaitu nilai akan ditambahkan lebih dulu maka akan menghasilkan angka 2 – 10, berbeda dengan sebelumnya. Sehingga dapat disimpulkan bahwa hasil dari proses adalah bergantung pada posisi operator *post-increment*.

d. Perulangan *Foreach* pada PHP

Jenis perulangan *Foreach* pada PHP ini adalah yang paling berbeda dari perulangan sebelumnya, yaitu perulangan *Foreach* pada PHP ini berguna untuk memecah isi array, atau lebih sederhananya ialah untuk menyederhanakan nilai array agar dapat dibaca dengan mudah.

berikut ini adalah bentuk umum penulisannya:

```
foreach(Array as Key => Value)
{
    # Code...
}
```

Contoh 1. Perulangan *Foreach* Array Tanpa Key

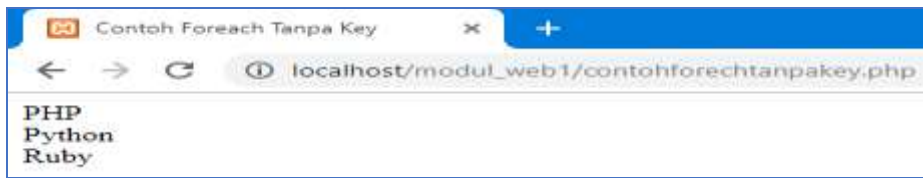
Buat file baru didalam folder htdoc/perulangan/**foreach1.php**

Ketikkan koding berikut:

```
<html>
<head>
    <title> Contoh Foreach Tanpa Key</title>
</head>

<body>
<?php
$array = array('PHP', 'Python', 'Ruby');
foreach($array as $value)
{
    echo $value . '<br />';
}
?>
</body>
</html>
```

Pada code bentuk umum penulisan pernyataan didalam fungsi *Foreach* ialah dengan menggunakan key untuk mengambil kunci array tersebut, namun dapat juga langsung mengambil nilainya saja, dan code diatas akan menghasilkan output seperti gambar dibawah ini:



Gambar 7.5 hasil Perulangan Menggunakan Foreach Array Tanpa Key

Contoh 2. Perulangan Foreach Array Dengan Key

Buat file baru didalam folder htdoct/perulangan/foreach2.php

Ketikan koding berikut:

```
<html>
<head>
    <title> Contoh Foreach Dengan Key</title>
</head>
<body>
<?php
$array = array('Java', 'PHP', 'C++');
foreach($array as $key => $value)
{
    echo $key . ' yaitu Bagian dari ' . $value . '<br />';
}
?>
</body>
</html>
```

Terlihat bahwa Variable \$key digunakan untuk menampilkan kunci dari array tersebut, berbeda dengan yang sebelumnya yaitu hanya dengan menampilkan nilainya saja. Contoh sederhana diatas maka akan menampilkan output seperti gambar dibawah ini:

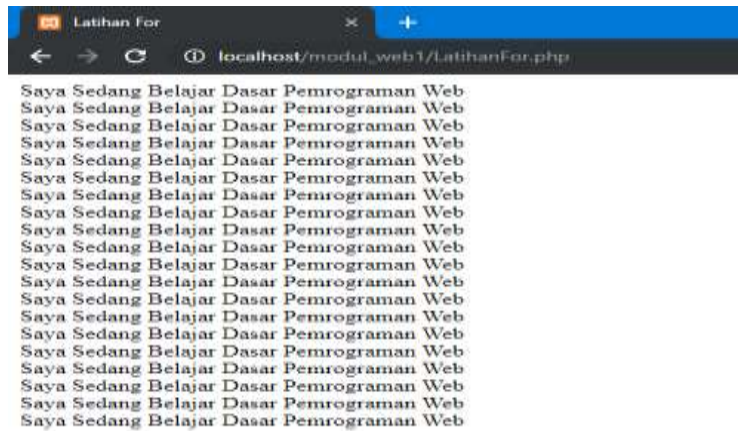


Gambar 7.6 hasil Perulangan Menggunakan Foreach Array Dengan Key

LATIHAN

1. Buatlah tampilan tulisan kata “Saya Sedang Belajar Dasar Pemrograman Web” sebanyak 20 kali secara berulang,

menggunakan script perulangan, untuk lebih jelasnya lihat tampilan dibawah ini:



Gambar 7.7 hasil tampilan latihan

JAWABAN:

```
<html>
<head>
    <title> Latihan For</title>
</head>

<body>
<?php
for($kata=1;$kata<=20;$kata++)
{
echo "Saya Sedang Belajar Dasar Pemrograman Web"."<br/>";
}
?>
</body>
</html>
```


BAB VIII

JAVASCRIPT

Deskripsi:

Membahas pengertian dasar dan penulisan script sederhana menggunakan Javascript, membahas tentang bagaimana step by step pembuatan dan penyimpanan file Javascript.

Tujuan Pembelajaran:

Setelah memahami pokok bahasan pada bab ini diharapkan mahasiswa mampu:

1. Memahami pengertian dasar javascript
2. Membuat tampilan website dengan penggunaan koding javascript sederhana

8.1. Pengertian Javascript

Javascript merupakan suatu bahasa script yang banyak digunakan dalam dunia teknologi terutama internet, bahasa ini dapat bekerja di sebagian besar web browser seperti Internet Explorer (IE), Mozilla Firefox, Netscape, opera dan web browser lainnya. bahasa javascript dapat dideskripsikan dalam bentuk fungsi (*Function*) yang ditaruh di bagian dalam tag <head> yang dibuka dengan tag <script language =” javascript”> Isi dari script javascript sama dengan konsep yang sudah dipelajari dalam materi PHP, yakni ada deklarasi Variable, penggunaan operator, percabangan, looping, dan fungsi. Di dalam java script juga sebuah komponen Alert yang digunakan untuk menampilkan kotak pesan pada browser ketika fungsinya di jalankan.

Untuk berlatih deklarasi script pada javascript, salin contoh-contoh berikut ini pada editor anda. Dan jalankan pada browser, amati tampilannya.

8.2. Latihan pembuatan koding javascript sederhana

Berikut merupakan contoh-contoh sederhana penulisan script penggunaan perintah javascript

a. Menuliskan teks = contohjs1.html

```
<html>                                </script>
<head>                                </body>
    <title> Untitled Document</title>  </html>
</head>
<html>
<body>
<script type="text/javascript">
<html>
<head>
    <title> Untitled Document</title>
</head>

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

c. JavaScript yang diletakkan pada bagian HEAD = contohjs3.html

```
<html>
<head>
    <title> Untitled Document</title>
</head>

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

d. JavaScript yang diletakkan pada bagian BODY = contohjs4.html

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

e. Fungsi = contohjs5.html

```
<html>
<head>
  <script type="text/javascript">
    function myfunction( )
    {
      alert("HELLO")
    }
  </script>
</head>
<body>
  <form>
    <input type="button" onclick="myfunction( )" value="Call function">
  </form>
  <p>By pressing the button, a function will be called. The function
  will alert a message.</p>
</body>
</html>
```

f. Fungsi dengan argumen = contohjs6.html

```
<html>
<head>
<script type="text/javascript">
function myfunction(txt)
{
alert(txt)
}
</script>
</head>
<body>
<form>
<input type="button"
onclick="myfunction('Good Morning!')"
value="Selamat Pagi">
</form>
</body>
</html>
```

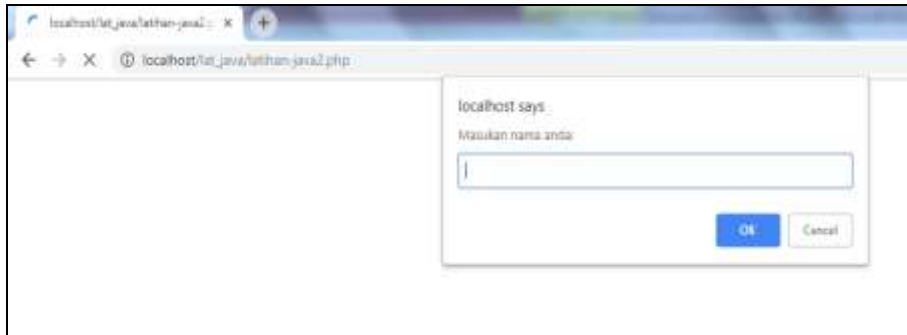
<input type="button"
onclick="myfunction('Good Evening!')"
value="Selamat Malam">
</form>
<p>
ketika di tekan salah satu tombol maka
fungsi akan di panggil dan pesan akan di
tampilkan
</p>
</body>
</html>

g. Memunculkan tanggal lengkap = contohjs7.html

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

LATIHAN

1. Buatlah Program Sederhana menggunakan perintah JavaScript
2. Proram yang akan dibuat berupa tampilan pesan seperti dibawah ini menggunakan perintah javascript



Gambar 8.1 Rancangan Tampilan Latihan di browser

JAWABAN:

1. Buka aplikasi notepad atau yang lainnya.
2. Buat dokumen baru pada aplikasi tersebut.
3. Tuliskan coding JavaScript dibawah ini.

```
var n = prompt("Masukan nama anda:");
var c = confirm("Hai "+n+"! Apakah saya tampan?");
if (c == true) {
    alert('Oh Thanks!!');
}else{
    alert('Why?!');
}
```

4. Simpan file tersebut dengan nama program.js
5. Buat dokumen baru lagi pada aplikasi anda.
6. Tuliskan coding html berikut ini, yang digunakan untuk memanggil file javascripts diatas.

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="program.js"></script>
  </head>
  <body>
```

7. Simpan file html tersebut dengan nama *index.html* harus dingat kedua file harus tersimpan dalam satu folder yang sama.
8. Jalankan file diatas dengan memanggil nama folder tempat folder diatas disimpan.

BAB IX

CSS

Deskripsi:

CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / *markup language*. apabila kita membahasnya dalam konteks web, bisa di artikan sebagai bahasa yang digunakan untuk mengatur tampilan / desain sebuah halaman HTML.

Tujuan Pembelajaran:

Setelah memahami pokok bahasan pada bab ini diharapkan mahasiswa mampu:

1. Memahami pengertian dasar tentang CSS
2. Memahami Kegunaan dari CSS
3. Mengimplementasikan penerapan CSS pada sebuah web.
4. Mempelajari padding, margin dan border pada CSS .

9.1. Pengertian CSS

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

9.2. Beberapa hal yang dapat dilakukan dengan CSS.

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

9.3. Cara Pemasangan CSS Pada Dokumen HTML

Ada 3 cara untuk memasang CSS pada dokumen HTML yaitu:

- *External Style Sheet*

Aturan CSS disimpan pada suatu file sehingga terpisah dari dokumen HTML.

Kemudian tambahkan kode pemanggilan file CSS dalam dokumen HTML.

File CSS(Misalnya disimpan dengan nama style.css) berisi :

```
p{text-align: justify;}
```

Dokumen HTML berisi :

```
<html>
<head>
    <title>CSS Secara Internal</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<p> Paragraph yang diatur CSS Secara External</p>
</body>
</html>
```

Akhiran file CSS adalah **.CSS**

- *Internal Style Sheet*

Aturan CSS ditulis pada bagian HEAD dokumen HTML menggunakan tag

`<style>`

```
<html>
<head>
    <title>CSS Secara Internal</title>
    <style type="text/css">P{text-align:justify;}
</style>
</head>
<body>
<p> Paragraph yang diatur CSS Secara Internal</p>
</body>
</html>
```


- *Inline Style Sheet*

Aturan CSS ditulis langsung pada tag HTML yang akan diatur tampilannya menggunakan atribut style:

```
<html>
<head>
  <title>CSS Secara Internal</title>
  <style type="text/css">P{text-align:justify;}
</style>
</head>
<body>
<p style ="text-align:justify;"> Paragraph yang diatur
CSS Secara Internal</p>
</body>
</html>
```

SATUAN DALAM CSS

1. Statik

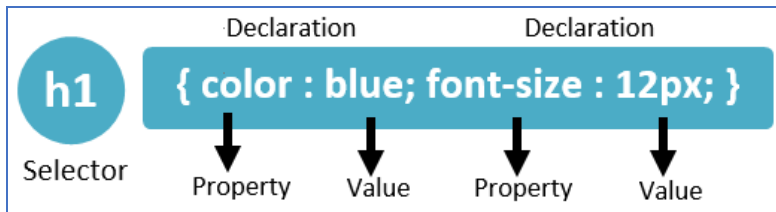
- * in → satuan inchi
- * cm → satuan centimeter
- * mm → satuan milimeter
- * pt → satuan point (1point = 1/72 inchi)
- * pc → satuan pica (1pica = 12 point)
- * px → satuan pixel (satu titik gambar terkecil dalam layar monitor)

2. Relatif

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

9.4. Penulisan CSS

Sintaks penulisan CSS sebagai berikut:



Gambar 9.1 susunan penulisan sintaks CSS

Penjelasan:

Aturan CSS terdiri 2 bagian:

1. *Selector*

Biasanya berupa tag HTML, id, class

- id menggunakan tanda # didepan nama selector
- class menggunakan tanda titik didepan nama selector

contoh :

h1 { color : blue ; } → tag html h1

#teks { color :green; } → id

.warna { color : red; } → class

2. *Declaration*

Berisi aturan-aturan css yang terdiri dari properti dan nilainya yang dipisahkan oleh tanda titik dua. Setiap aturan css harus diakhiri dengan tanda titik koma.

a. **Selector ID pada CSS**

Untuk selector id pada css ditandai dengan tanda #(pagar) contoh penulisan seperti berikut :

```
#teks
{
  Color : blue;
  Font-family: Calibri;
}
```

Penggunaanya dalam script HTML :

```
<body>
  <p id="teks">TEST
</p>
</body>
```

Yang perlu di perhatikan jika menggunakan selector id :

- 1) Sebuah elemen HTML hanya boleh memiliki 1 id
- 2) Setiap halaman hanya boleh memiliki 1 elemen dengan id tersebut
- 3) Dapat di gunakan sebagai penanda halaman untuk link
- 4) Digunakan juga untuk javascript
- 5) Sebaiknya tidak digunakan untuk css (lebih baik gunakan class)

b. Selector Class pada CSS

Untuk selector class pada css ditandai dengan tanda . (titik) contoh penulisan seperti berikut :

```
.warna
{ background-color: lightgreen; }
```

Penggunaanya dalam script HTML :

```
<body class="warna">

</body>
```

9.5. Properti-properti CSS

Properti CSS jumlahnya sangat banyak, berikut beberapa diantaranya:

Properti	Fungsi	Nilai	Contoh penulisan :
Color	Mengatur warna pada Teks	Nama warna, kode hexa warna	Color : blue; Color:#ffffff;
Background-color	Mengatur warna latar	Nama warna, kode hexa warna,rgb	Background-Color:rgb(200,0,55);
Background-image	Mengatur gambar latar	Nama file gambar	Background-image:url(header.jpeg);
Text-align	Mengatur perataan teks	Left,Center, right, justify	Text-align:left;
Font-family	mengatur jenis font	Nama-nama jenis huruf, co: arial, times new roman, georgia	Font-family:calibri;
Dll.			

Pseudo-Class

Adalah sebuah kelas semu yang dimiliki oleh elemen HTML, yang membuat kita dapat mendefinisikan style pada keadaan tertentu dari elemen tersebut. *Pseudo-class* terbagi menjadi beberapa type, sebagai berikut :

1. Yang berhubungan dengan link

a) : link

Style default pada sebuah link (a yang memiliki href)

b) : hover

Style ketika kursor mouse berada diatas sebuah link / elemen

c) : active

Style ketika sebuah link di klik (keadaan aktif)

d) : visited

Style ketika sebuah link sudah pernah di kunjungi sebelumnya (menggunakan browser yang sama)

2. Yang berhubungan dengan posisi elemen (ada pada css 3)

a) : first-child

Memilih elemen pertama dari sebuah parent (elemen pembungkusnya)

b) : *last-child*

Memilih elemen terakhir dari sebuah parent (elemen pembungkusnya)

c) : *nth-child(n)*

Memilih elemen ke (n) dari sebuah parent (elemen pembungkusnya)
n bisa berarti urutan 1,2,3,..... atau pola $(2n)$, $(3n+2)$, atau ganjil dan genap, even & odd

d) : *first-of-type*

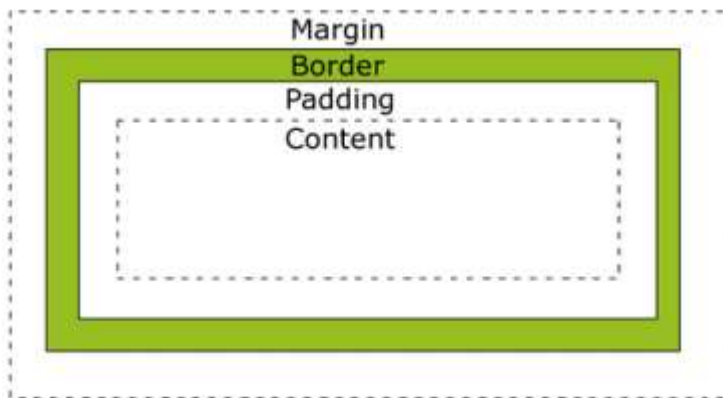
Memilih elemen pertama dari sebuah jenis / tipe tag

e) : *last-of-type*

Memilih elemen terakhir dari sebuah jenis / tipe tag

9.6 PADDING, MARGIN DAN BORDER

Dalam CSS dikenal istilah 'Box Model'. Perhatikan gambar berikut ini:



Gambar 9.2 Box Model CSS

Keterangan gambar :

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

Border : Adalah garis tepi dari komponen

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

CSS menggunakan konsep ini dalam mengatur tag-tag HTML. Pada gambar, bayangkan area 'Content' misalnya adalah sebuah paragraph. Obyek paragraph ini akan dianggap CSS memiliki area padding, border, dan margin disekitarnya. Keberadaan area-area ini berguna untuk pengaturan tata letak. Misalnya ingin diatur agar 2 buah gambar yang terletak berdampingan tidak terlalu rapat, maka kita dapat memperbesar lebar dari area margin agar jarak antara gambar lebih lebar.

a. Padding

ditulis dengan CSS `padding:5px 5px 5px 5px;` urutan nilai angkanya adalah atas, kanan, bawah dan kiri, atau bisa menggunakan penulisan lengkap seperti berikut ini :

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

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

`padding-top:5px;` ➔ untuk bagian atas dan

`padding-bottom:5px;` ➔ untuk bagian bawah, Ingat satuan px(pixels) bisa kamu ganti

sesuai satuan yang lain yang sesuai

b. Border

Ditulis dengan CSS `border:1px dotted #000000;` ➔ urutan penggunaanya adalah ukuran border, style border dan warna border, atau bisa menggunakan penulisan lengkap seperti berikut ini :

`border-width:1px;` ➔ ini adalah ketebalan border sebesar 1pixel

`border-style:dotted;` ➔ ini adalah jenis bordernya yang bisa diganti dengan tipe dashed, solid, double, groove, ridge, inset, outset dan lainnya

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

c. Margin

Ditulis dengan CSS `margin:5px 5px 5px 5px;` ➔ urutanya atas, kanan, bawah dan kiri,

atau bisa menggunakan penulisan lengkap seperti berikut ini :

`margin-left:5px;`

`margin-right:5px;`

`margin-top:5px;`

`margin-bottom:5px;`

Latihan CSS :

- Salin script berikut ini, lalu simpan di folder `lat_css` dengan

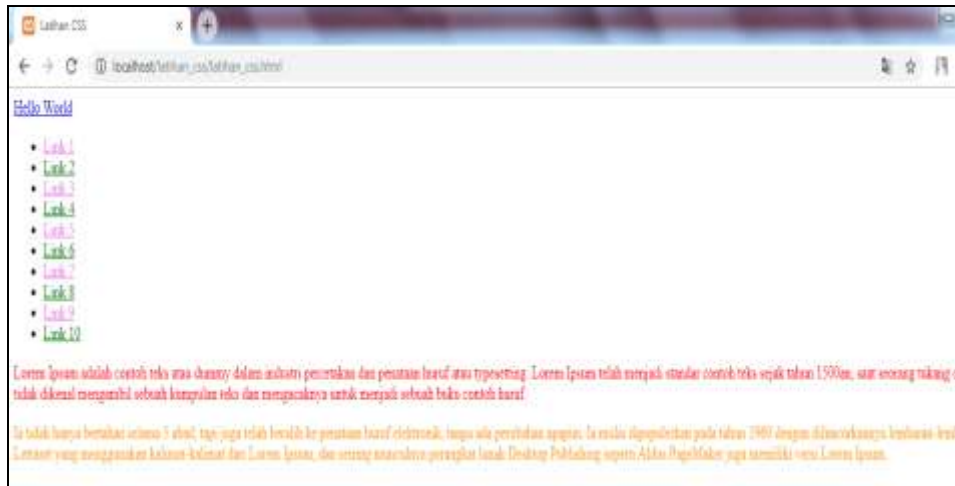
```
/* css carikan a,di dalam li yang merupakan
anak pertama*/
    li:first-child a{
        color:green;
    }
/* css carikan a,di dalam li yang merupakan
anak terakhir*/
    li:last-child a{
        color:green;
    }
/* css carikan a, di dalam li yang merupakan
anak genap*/
    li:nth-child(even) a{
        color:green;
    }
```

```
/* css carikan a, di dalam li yang merupakan
anak ganjil*/
    li:nth-child(odd) a{
        color:violet;
    }
/* css carikan p, di dalam halaman yang urutannya pertama*/
p:first-of-type {
    color: red;
}
/* css carikan p, di dalam halaman yang urutannya terakhir*/
p:last-of-type {
    color: orange;
}
```

- Salin script berikut ini, lalu simpan di folder modul_web1 dengan nama latihan_css.html

```
<!DOCTYPE html>
<html>
<head>
    <title> Latihan CSS</title>
    <link rel="stylesheet" type="text/css" href="style_css.css">
</head>
<body>
<a href="#" class="hello">Hello World</a>
<ul>
    <li><a href="#">Link 1</a></li>
    <li><a href="#">Link 2</a></li>
    <li><a href="#">Link 3</a></li>
    <li><a href="#">Link 4</a></li>
    <li><a href="#">Link 5</a></li>
    <li><a href="#">Link 6</a></li>
    <li><a href="#">Link 7</a></li>
    <li><a href="#">Link 8</a></li>
    <li><a href="#">Link 9</a></li>
    <li><a href="#">Link 10</a></li>
</ul>
<p>Lorem Ipsum adalah contoh teks atau dummy dalam industri percetakan dan penataan huruf atau typesetting. Lorem Ipsum telah menjadi standar contoh teks sejak tahun 1500an, saat seorang tukang cetak yang tidak dikenal mengambil sebuah kumpulan teks dan mengacaknya untuk menjadi sebuah buku contoh huruf. </p>
<p> Ia tidak hanya bertahan selama 5 abad, tapi juga telah beralih ke penataan huruf elektronik, tanpa ada perubahan apapun. Ia mulai dipopulerkan pada tahun 1960 dengan diluncurkannya lembaran-lembaran Letraset yang menggunakan kalimat-kalimat dari Lorem Ipsum. dan seiring munculnya perangkat lunak
```


- Hasil yang akan ditampilkan dibrowser :



Gambar 9.3 hasil Tampilan latihan CSS

BAB X

Membuat Design Web Responsive Menggunakan CSS

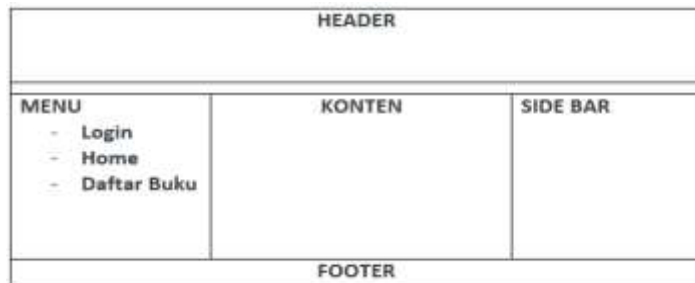
10.1 Merancang Design Web Responsive menggunakan CSS

Pada pembahasan materi kali ini kita akan membuat Design Web Responsive.

Design Web Responsive adalah sebuah metode atau pendekatan sistem web desain yang bertujuan memberikan pengalaman berselancar yang optimal dalam berbagai perangkat, baik mobile maupun komputer. Dengan metode ini. Berikut kita akan membuat form responsive. Langkah-langkah yang perlu di gunakan dalam pembuatan web ini adalah :

1. Tentukan tema website
 - Tema web perpustakaan
2. Membuat sketsa / blueprint / rancangan dari halaman web yang akan di buat

Seperti contoh berikut ini :



Gambar 10.1 rancangan tampilan responsive yang akan dibuat

3. Web responsive berarti web yang tampilannya mengikuti ukuran layar gadget yang di gunakan untuk akses web tersebut
 - a. Buat folder di xampp/htdocs dengan nama : perpustakaan
 - b. Didalam folder perpustakaan, siapkan 2 buah folder :
 - images ➔ untuk simpan gambar
 - css ➔ untuk simpan file css

- c. Buka editor, salin script css berikut : simpan di folder css dengan nama style.css

```
*{
    padding      : 0;
    margin       : 0;
    box-sizing   : border-box;
}
body{
    background    : #dedede;
    font-family   : 'Verdana',arial,sans-serif;
    font-size     : 15px;
}
p{
    margin-bottom : 20px;
    line-height   : 1.5em;
}

h3{
    margin-bottom : 20px;
    border-bottom : 1px solid #aaa;
}

a{
    text-decoration : none;
    color           : #333;
}

a:hover{
    color : #666;
}

.container{
    max-width : 1080px;
    margin    : 20px auto;
    background : #fff;
    overflow  : hidden;
    padding   : 10px;
}

.header{
    border      : 1px solid #dedede;
    padding     : 10px;
    margin      : 10px;
    background  : #9E9AFB;
}

/* main */
.left{
    width      : 250px;
    border     : 1px solid #dedede;
    padding    : 10px;
    margin     : 10px;
    float      : left;
}
```

```

        .left ul li a{
            display      :block;
            border-bottom : 1px solid #dedede;
            margin-bottom : 10px;
            padding       : 10px 5px;
            font-color    : #D3D2ED;
        }

        .left ul li a:hover{
            color:#461AF3;
        }

        .middle{
            width      : 500px;
            border     : 1px solid #dedede;
            padding    : 10px;
            margin     : 10px;
            float      : left;
        }

        .middle img{
            max-width   : 100%;
            height      : auto;
        }

        .middle a{
            font-wight:bold;
        }

        .right{
            width       : 250px;
            border      : 1px solid #dedede;
            padding     : 10px;
            margin      : 10px;
            float       : left;
        }

        .right ul{
            list-style-type : none;
        }
        .right ul li{
            display : block;
        }
        .right ul li a{
            display      :block;
            border-bottom : 1px solid #dedede;
            margin-bottom : 10px;
            padding       : 10px 5px;
        }
        .right ul li a:hover{
            color:#461AF3;
        }

        .footer{
            clear      : both;
            border     : 1px solid #dedede;
            padding    : 15px;

```

```

@media screen and (max-width:959px){
    #container{
        width    : 100%;
    }
    #left-column{
        width    : 70%;
    }
    #right-column{
        width    : 30%;
    }
    img{
        width    : 100%;
    }
}

/*MEDIA QUERIES ( Responsive )
******/
@media screen and (max-width:1080px)
{
    .container{
        width    : 100%;
    }
    .left{
        width        : 25%;
        background    : #D6CCFE;
    }
    .middle{
        width        : 68%;
        float        : right;
    }
    .right{
        clear        : both;
        padding      : 1% 4%;
        width        : auto;
        float        : none;
        background    : #D6CCFE;
    }
}

/* untuk ukuran layar 700px kebawah */
@media screen and (max-width: 780px)
{
    .header,
    .footer{
        text-align    : center;
    }
    .left {
        width        : auto;
        float        : none;
    }

    .middle {
        width        : auto;
        float        : none;
    }
}

```

10.2 Merancang Halaman Index Web Responsive

Tampilan index yang akan kita buat seperti gambar dibawah ini



Gambar 10.2 hasil tampilan halaman responsive

1. Buka file baru di aplikasi anda
2. Untuk membuat tampilan seperti diatas berikut adalah kodingan yang perlu kita ketikan.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title> Website Responsive</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <div class="container">
    <div class="header">
      <h1 align="center"> Selamat Datang DI Pepustakaan XYZ </h1>
    </div>
    <div class="main">
      <div class="left">
        <h3 align="center">Menu</h3>
        <ul>
          <li><a href="#">Login</a></li>
          <li><a href="#">Home</a></li>
          <li><a href="#">Daftar Buku</a></li>
        </ul>
      </div>
      <div class="middle">
        <h3 align="center">Berita</h3>
        <p align="center"><strong><a href="#">Membuat Design Web
Responsive</a></strong></p>
        <p></p>
        <p><a href="#">Baca Selengkapnya >> </a>

```

```

<div class="right">
  <h3 align="center">Buku Terpopuler</h3>
  <p>
    <ul>
      <li><a href="#">Web Design</a></li>
      <li><a href="#">Pemrograman</a></li>
      <li><a href="#">Database</a></li>
    </ul>
  </p>
</div>
</div>

<div class="footer">
  <p align="center">Copyright &copy; 2018 Belajar CSS Responsive </a></p>
</div>
</div>
</body>
</html>

```

```

<meta name="viewport" content="width=device-width, initial-
scale=1.0">

```

merupakan syntax yang berfungsi untuk menampilkan web sesuai dengan layar.

```

<link rel="stylesheet" href="css/main.css">

```

merupakan External Style sheet yang digunakan untuk mengload file css .

Pada Struktur HTML berikut terdapat 3 class utama yaitu header , main dan footer .

Yaitu :

```

<div class="container">
  <div class="header">digunakan untuk bagian header</div>
  <div class="main">
    <div class="left">Digunakan kolom bagian Kiri...</div>
    <div class="middle">Digunakan kolom bagian
tengah...</div>
    <div class="right">Digunakan kolom bagian kanan...</div>
  </div>
  <div class="footer">Digunakan bagian footer...</div>
</div>

```

3. Setelah selesai mengetikan kodingan diatas, simpanlah file tersebut di dalam folder perpus dengan nama index.html
4. Lalu panggil file index.html diatas di browser anda.

DAFTAR PUSTAKA

Abdulloh, R. (2018). *7 in 1 Pemrograman Web Untuk Pemula*. Jakarta: Elex Media Komputindo.

Aldeheid, A. (2015). *Website no.1 Cara Mudah Bikin Website dan Promosi ke Seo*. Yogyakarta: Penerbit Mediakom.

Hidayatulloh, P., & Kawistara, J. K. (2014). *Perancangan Web*. Bandung: Informatika.

Lee, C. (2011). *Referensi Ringkas HTML 5*. Jakarta: Elex Media Komputindo.

Sidik, B. (2017). *Pemrograman Web dengan PHP 7*. Bandung: Informatika.

Simarmata, J. (2010). *Rekayasa Web*. Yogyakarta: Penerbit Andi.

<https://www.bahasaweb.com>

<https://www.duniailkom.com>

<https://www.malasngoding.com>

TENTANG PENULIS



Ani Oktarini Sari, S.kom, MMSI

Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Mendapat gelar Sarjana pada tahun 2010 di STMIK Nusa Mandiri Jakarta dan Magister Manajemen Sistem Informasi Peminatan Rekayasa Perangkat Lunak di Universitas Gunadarma pada tahun 2015.



Ari Abdilah, M.Kom

Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Mendapatkan gelar Sarjana pada tahun 2011 dan Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta pada tahun 2015.



Sunarti, M.Kom

Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Menyelesaikan pendidikan Sarjana tahun 2009 dan Magister Ilmu Komputer di STMIK Nusa Mandiri Jakarta pada tahun 2011.

Sinopsis Buku

Buku Web Programing berisikan materi belajar mengenai dasar-dasar pemrograman web. Buku ini direkomendasikan bagi pemula belajar pemrograman web. Buku ini menjelaskan bagaimana belajar dasar-dasar web dengan mudah, praktis dan cepat disertakan contoh latihan-latihan. Dan adanya latihan contoh studi kasus membuat website yang responsive. Buku ini membahas mengenai dasar-dasar bahasa pemrograman web antara lain : *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), *Hypertext Preprocessor* (PHP) dan *JavaScript*.