

KONSEP DASAR PEMROGRAMAN WEBSITE dengan PHP

```
34 * Page. This variable should always be output last, after all other content.
35 * content.
36 * - $classes String of classes that can be used to style contextually through
37 * CSS.
38 *
39 * @see template_preprocess()
40 * @see template_preprocess_html()
41 * @see template_process()
42 */
43 ?<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
44 "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
45 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<php print $language;">
  version="XHTML+RDFa 1.0" dir="<php print $language->dir;"><php print $rdf_namespaces;">
  >>
46
47 <head profile="<php print $grddl_profile;">
48 <?php print $head;">
49 <title><?php print $head_title;"></title>
50 <?php print $styles;">
51 <?php print $scripts;">
52 </head>
53 <body class="<php print $classes;"> <?php print $attributes;">
54 <div id="skip-link">
55 <a href="#main-content" class="element-invisible element-focusable"><php print t('
  Skip to main content')> </a>
56
57 </div>
58 <?php print $page_top;">
59 <?php print $page;">
60 <?php print $page_bottom;">
61 </body>
62 </html>
```

```
4 <?php
5
6 if (isset($_GET['category/create'])) {
7   # code...56
8   include("controller/CategoryController.php");
9
10   $category=new CategoryController();
11   $category->create();
12   #else if (isset($_GET['category/store'])) {
13     # code...56
14     include("controller/CategoryController.php");
15
16     $name=$_POST['name'];
17     $status=$_POST['status'];
18
19     $category=new CategoryController();
20     $category->store($name,$status);
21   }else{
22     include("controller/DashboardController.php");
23
24     $dashboard=new DashboardController();
25     $dashboard->index();
26   }
27 }
```

BUKU AJAR KONSEP DASAR PEMROGRAMAN WEBSITE DENGAN PHP

ELGAMAR



Penerbit: CV. Multimedia Edukasi

BUKU AJAR KONSEP DASAR PEMROGRAMAN WEBSITE DENGAN PHP

Penulis:

Elgamar

Editor:

Ndari Pangesti

Desain Cover dan Tata Letak:

Tim Ahlimedia

Penerbit:

CV. Multimedia Edukasi

Jl. Ki Ageng Gribig, Gang Kaserin MU No. 36

Kota Malang 65138

Telp: +6285232777747

www.multidukasi.co.id

ISBN: 978-623-7531-97-5

Cetakan Pertama, Juli 2020

Hak cipta oleh Penulis dan Dilindungi Undang-Undang Republik Indonesia Nomor 19 Tahun 2002 Tentang Hak Cipta, Pasal 72. Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit.

KATA SAMBUTAN

Assalamualaikum.wr.wb

Bersyukur dan ribuan terima kasih kami ucapkan atas dedikasi penulis Bapak **Elgamar, S.Kom., M.Kom** yang telah rela mengorbankan waktu tenaga dan pikirannya untuk dapat menghasilkan sebuah buku ajar yang dapat memberikan manfaat kepada para pembacanya, terkhusus Mahasiswa/i Program Studi Teknik Informatika (Universitas Islam Kuantan Singingi) Telukkuantan-Riau. Kami berharap buku ini dapat menjadi buku pegangan bagi mahasiswa/i Teknik Informatika dalam memahami Konsep Dasar Pemrograman Website Dengan PHP. Semoga terus ada dosen-dosen di Prodi Teknik Informatika yang menghasilkan buku-buku dan karya-karya yang dapat memberikan manfaat bagi khalayak ramai. Terima kasih atas segala kontribusi penulis terhadap ilmu dan pengetahuan, dan semoga bermanfaat.

Wassalamualaikum.wr.wb

Telukkuantan, 07 Juli 2020

Ketua Program Studi Teknik Informatika
Universitas Islam Kuantan Singingi

KATA PENGANTAR

Assalamualaikum.wr.wb

Puji dan syukur penulis ucapkan kehadiran Allah yang Maha Kuasa, yang telah memberikan kekuatan kepada penulis untuk menyelesaikan buku ajar yang berjudul **Konsep Dasar Pemrograman Website Dengan PHP**. Dengan adanya buku ini, penulis berharap dapat memberikan manfaat dan kontribusi pada ilmu dan pengetahuan bagi para pembaca, dan khususnya Mahasiswa/i Program Studi Teknik Informatika. Penulis menyadari bahwa masih banyak terdapat kekurangan pada buku ajar ini, oleh karena itu penulis menerima segala bentuk kritikan dan masukan dari pembaca semua, sehingga penulis dapat memperbaiki segala kekurangan. Silahkan kirim kritik dan saran melalui email: elgamar.mkom@gmail.com dan Terima kasih telah membaca buku ini, semoga bermanfaat.

Wassalamualaikum.wr.wb

Telukkuantan, 12 Juli 2020

Penulis

DAFTAR ISI

Kata Sambutan	iii
Kata Pengantar	iv
Daftar Isi	v
Bab I Pendahuluan	1
Bab II Website dan Perangkat Lunak Pendukung	2
Bab III Dasar Pembangunan Website dan HTML	23
Bab IV Dasar Penulisan PHP	48
Bab V Dasar Penulisan PHP Fungsi IF, SWITCH CASE, Operator Perbandingan	54
Bab VI Dasar Penulisan PHP dengan Fungsi Perulangan (Looping)	59
Bab VII Dasar Penulisan PHP dengan Form Input	65
Bab VIII Dasar Penulisan PHP dengan Method POST dan GET	74
Lampiran	79
Daftar Pustaka	80
Profil Penulis	81

Pendahuluan

A. Ruang Lingkup Pembelajaran

Pemrograman Web merupakan mata kuliah wajib pada Program Studi Teknik Informatika yang berjumlah 3 SKS. Mata kuliah ini memberikan pemahaman tentang bahasa pemrograman web yang terdiri HTML, PHP, CSS, Javascript dan Database MySQL.

B. Tujuan Pembelajaran

Setelah mengikuti perkuliahan Pemrograman Web, mahasiswa diharapkan dapat memahami pentingnya sebuah alur atau algoritma dari sebuah bahasa pemrograman web.

C. Deskripsi

Mata kuliah ini memberikan pengetahuan dasar kepada mahasiswa tentang alur sebuah bahasa pemrograman Web, yang terdiri dari struktur HTML, PHP, CSS, Javascript serta pengorganisasian sistem menggunakan database MySQL.

Pendekatan Pembelajaran: Ekspositori dan Inkuiri

Metode : Ceramah, Latihan praktek dan Tanya Jawab

Tugas : Melatih kemampuan program dan analisa sistem

Media : OHP, LCD.

Kriteria Penilaian: Bobot penilaian kemampuan atau keberhasilan belajar didasarkan pada:

- Presensi 75% dari keseluruhan kegiatan tatap muka, berpartisipasi aktif dalam perkuliahan, diskusi, dan pengumpulan tugas.
- Tugas Individual/ Kelompok
- Ujian Tengah Semester (UTS)
- Ujian Akhir semester (UAS)

D. Rincian Materi Pembelajaran

1. Pengertian Website
2. Pengertian Web Server dan Database MySQL
3. Perancangan Website
4. Elemen Dasar dan contoh Tag HTML
5. Dasar penulisan PHP
6. Fungsi dan Atribut PHP
7. Operator Matematika
8. Fungsi IF dan Perulangan (*Loop*)
9. PHP dan *Form*

E. Capain Pembelajaran

1. Memahami dasar-dasar Web dan perangkat lunak pendukungnya.
2. Memahami dasar pembangunan web dan HTML.
3. Memahami setiap fungsi tag HTML.
4. Memahami dasar Pemrograman PHP.
5. Memahami konsep basisdata dalam pemrograman PHP
6. Memahami konsep pembangunan web dengan fungsi form input.
7. Memahami konsep dan fungsi Method POST dan GET.

Website dan Perangkat Lunak Pendukung

A. Pengertian Website

Website merupakan sebuah media yang memiliki banyak halaman yang saling terhubung (*hyperlink*), dimana website memiliki fungsi dalam memberikan informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya. Website pada saat sekarang ini umumnya telah bersifat dinamis, meskipun dahulu juga ada website yang bersifat statis, namun website statis telah jarang dan bahkan hampir tidak ada lagi ditemukan. Karakteristik utama yang dimiliki oleh website adalah halaman-halaman yang saling terhubung, dan dilengkapi dengan domain sebagai alamat (url) atau *World Wide Web* (www) dan juga hosting sebagai media yang menyimpan banyak data. Website dapat diakses menggunakan jaringan internet dengan platform yang disebut browser, seperti chrome, mozilla firefox, internet explorer (IE), opera dan sebagainya.

Website dapat dibangun dalam mode localhost, yang artinya website dapat dirancang, dibangun dan dimodifikasi tanpa menggunakan jaringan internet. Dalam pembangunan sebuah website sampai pada mode publikasi ke internet ada beberapa aplikasi yang dibutuhkan,

diantaranya adalah database (MySQL, Oracle) *etc*, Web Server Apache, PHP Editor (Macromedia, Notepad++) *etc*, dan *browser*. Website atau aplikasi berbasis web umumnya dibangun menggunakan bahasa pemrograman seperti *Hypertext Preprocessor* (PHP) dan *Active Server Pages* (ASP), yang dikombinasikan dengan *Hypertext Markup Language* (HTML), *Cascading Style Sheet* (CSS), dan Javascript. Pada tingkatan *expert* seorang programmer juga membutuhkan aplikasi tambahan lain seperti Photoshop, Coreldraw dan lain sebagainya. Setelah aplikasi tersebut di instalasi pada komputer maka barulah dilakukan proses perancangan, desain dan pengkodean (coding) terhadap website yang akan dibangun.

B. Apa itu PHP?

PHP telah menjadi bahasa pemrograman web yang digunakan secara luas untuk membuat halaman web yang dinamis. Dalam hal ini diperkuat oleh Solichin (2016) yang menyatakan bahwa PHP menjadi salah satu bahasa pemrograman yang digunakan sebagai pengembangan website. PHP awalnya mulai dikembangkan pada akhir tahun 1994 oleh Rasmus Lerdorf, namun sekarang di ambil oleh *The PHP Group*. Semula PHP merupakan singkatan dari *Personal Home Page*, namun dalam perkembangannya diubah menjadi PHP: *Hypertext Preprocessor*. PHP menjadi salah satu bahasa pemrograman yang bersifat interpreter, dalam artian membaca setiap instruksi dari sintaks (coding) dengan cara membaca satu persatu atau baris perbaris code program. Menurut Kadir (2017) ciri dari bahasa pemrograman

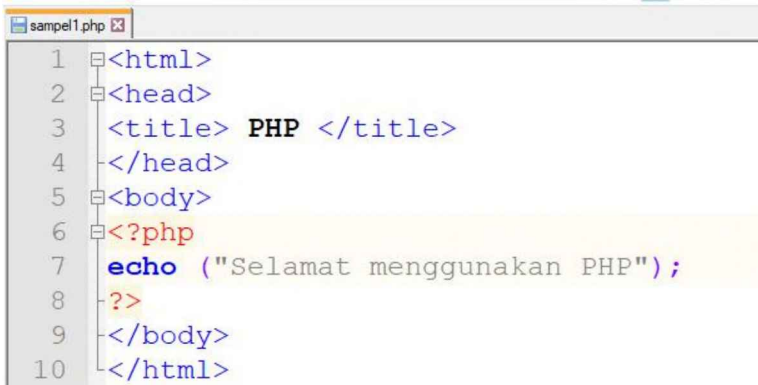
Interpreter adalah bekerja menerjemahkan instruksi pada saat program mulai di eksekusi.

Dengan kian banyaknya programmer atau peminat dalam menggunakan bahasa pemrograman PHP, hingga PHP terus berkembang dari masa ke masa yang dimanfaatkan sesuai dengan kebutuhan pengguna, dan hingga kini PHP telah masuk pada versi 7.0, berikut tabel perkembangan PHP dari masa ke masa:

Tabel 2.1. Perkembangan Bahasa Pemrograman PHP

Versi PHP	Tahun
1. PHP 1.0	1994
2. PHP 2.0	1996
3. PHP 3.0	1998
4. PHP 4.0	2000
5. PHP 5.0	2004
6. PHP 6.0	2005
7. PHP 7.0	2015

PHP juga dikenal dengan bahasa pemrograman *open source* yang berarti dapat digunakan secara gratis, PHP juga dapat disatukan dengan bahasa HTML. Prasetio (2012) menyatakan bahwa PHP merupakan bahasa pemrograman yang memiliki kemampuan dalam memisahkan kode PHP dari HTML. Lihat contoh berikut ini:



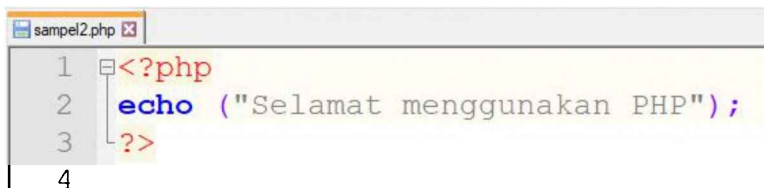
```

1 <html>
2 <head>
3   <title> PHP </title>
4 </head>
5 <body>
6 <?php
7   echo ("Selamat menggunakan PHP");
8 ?>
9 </body>
10 </html>

```

Gambar 2.1 Code PHP dalam HTML

PHP dapat juga dijalankan tanpa HTML, seperti contoh berikut:



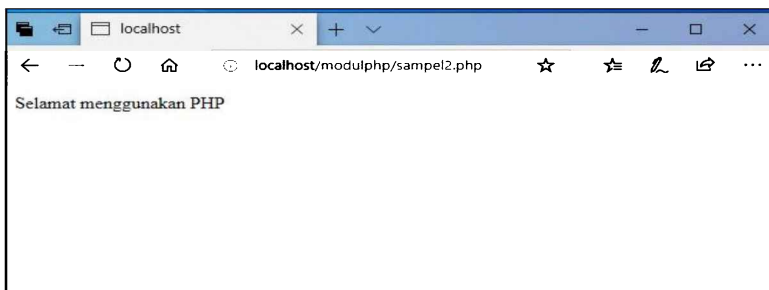
```

1 <?php
2   echo ("Selamat menggunakan PHP");
3 ?>
4

```

Gambar 2.2 Code PHP tanpa HTML

Berikut tampilan pada browser dari pengkodean PHP tanpa dan menggunakan HTML.



Gambar 2.3 Tampilan hasil Script PHP tanpa dan menggunakan HTML

C. Web Server Apache

Web Server adalah perangkat lunak yang memiliki fungsi dalam menerima permintaan klien, dimulai dari halaman web browser melalui protokol HTTP atau HTTPS, dan kemudian mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang biasanya berbentuk dokumen HTML (Marisa, 2017). Jika terjadi permintaan dari *browser*, maka kemudian *web server* akan memproses permintaan tersebut dan memberikan hasil prosesnya berupa data yang diminta kembali ke browser dan ditampilkan sesuai dengan kemampuan browser itu sendiri, berpedoman pada format *Standard General Markup Language* (SGML).

Dalam perkembangannya, ada 2 jenis web server yang umum digunakan, yaitu web server Apache dan *Internet Information Service* (IIS). Web server apache merupakan *web server* yang paling populer digunakan di internet. Apache merupakan perangkat lunak berbasis *open source* yang berarti dapat digunakan secara bebas. Menurut Raharjo & Istiyanto (2003) kelebihan perangkat lunak berbasis *open source* sebagai berikut:

1. Bebas dari biaya *licensi*.
2. Penyertaan kode sumber, sehingga isi dari perangkat lunak dapat dilihat secara jelas.
3. Diberi kebebasan dalam mengubah isi perangkat lunak.
4. *Up-to-date* dalam masalah keamanan.

Apache dilengkapi dengan program pendukung yang cukup banyak. Hal ini mampu memberikan layanan yang cukup lengkap bagi penggunaanya. Adapun dukungan apache seperti berikut:

Tabel 2.2. Dukungan Apache

Aspek Pendukung	Keterangan
1. <i>Access Control</i>	Proses kontrol berjalan berdasarkan nama <i>host</i> atau nomor IP
2. <i>Common Gateway Interface</i> (CGI)	Dukungan Apache dengan menempatkan sebagai modul (mod_perl)
3. PHP	PHP didukung apache dengan menempatkannya sebagai salah satu modulnya (mod_php)
4. <i>Server Side Includes</i> (SSI)	Halaman <i>web</i> dapat memasukkan potongan informasi yang tidak secara langsung disimpan dalam file.

Web server Internet Information System (IIS), web server keluar Microsoft yang banyak digunakan sebagai web server untuk aplikasi web dengan sistem operasi windows, dan bahasa pemrograman web ASP atau ASP.NET.

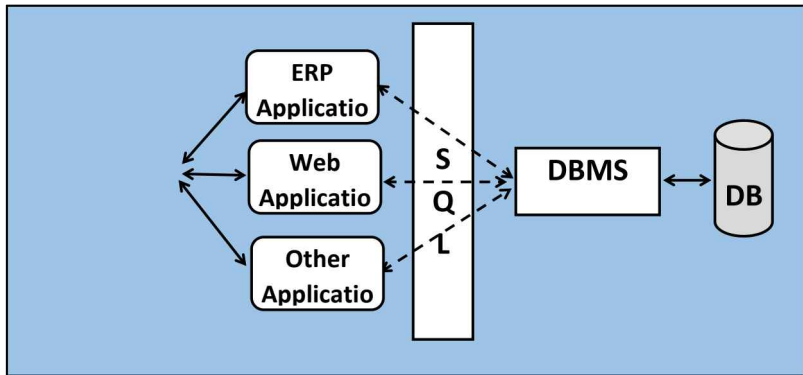
D. Database MySQL

Database merupakan media penyimpanan data yang dibuat secara sistematis dan terstruktur. Dalam pengelolaannya, *database*

memerlukan sebuah perangkat lunak yang disebut dengan *Database Management System* (DBMS). DBMS merupakan perangkat lunak yang memungkinkan pengguna untuk membuat, memelihara, mengontrol, dan mengakses *database* secara praktis dan efisien. Melalui perangkat lunak DBMS, pengguna akan lebih mudah mengelola, mengontrol dan memanipulasi data yang ada. Database dirancang agar dapat menggunakan data secara berulang-ulang sesuai kebutuhan pengguna.

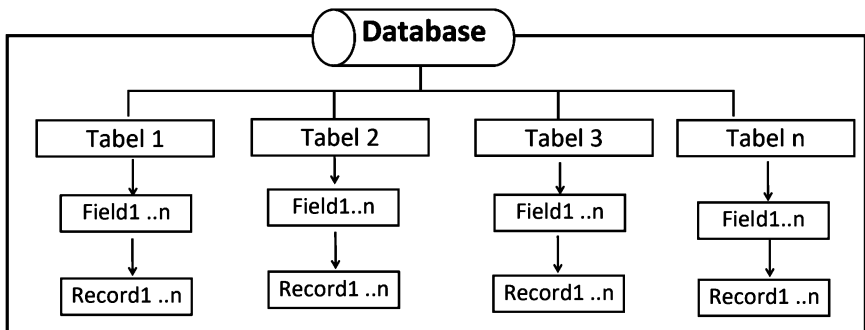
Ada banyak jenis database yang ada sekarang, paling tidak ada empat jenis yang umum digunakan oleh *user*, seperti MySQL, Microsoft SQL Server, PostgreSQL dan Oracle. Pada buku ini, penulis hanya menjelaskan database MySQL, karena MySQL menjadi database yang paling populer digunakan pada saat sekarang ini. MySQL merupakan database yang memiliki tipe data bersifat relasional, yang berarti MySQL memiliki cara dalam penyimpanan datanya berbentuk tabel-tabel yang saling terhubung (Zaki & Comunity, 2008).

Database My Structure Query Language (MySQL) berfungsi dalam mengolah *database* menggunakan bahasa *Structured Query Language* (SQL). SQL menjadi sebuah bahasa yang dipergunakan untuk mengakses dan memproses data dalam basis data relasional. Bahasa ini menjadi bahasa standar yang digunakan dalam manajemen basis data relasional.



Gambar 2.4 Skema Operasi SQL dan DBMS

Database MySQL memiliki struktur yang terdiri dari: **tabel**, **field** dan **record**.



Gambar 2.5 Struktur Database

Contoh baris perintah SQL:

//perintah membuat database:

```
CREATE DATABASE "nama_basisdata"
```

//perintah membuat tabel:

```
CREATE TABLE "nama_tabel"
(
  nama_field1 tipe_data [constraints],
  nama_field2 tipe_data,
  nama_field3 tipe_data
);
```

E. Instalasi Software Pendukung

Pada penjelasan sebelumnya, agar anda dapat bekerja dalam pembuatan website atau aplikasi berbasis web menggunakan bahasa pemrograman PHP, maka dibutuhkan beberapa perangkat lunak pendukung seperti Apache web server, MySQL *Database* yang termasuk dalam paket XAMPP dan juga PHP Editor.

1. Instalasi XAMPP

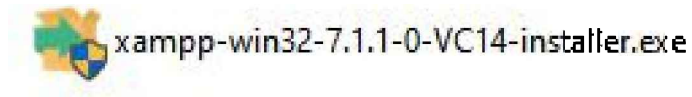
Perangkat lunak (*software*) XAMPP versi 7.1.1 ini merupakan sejenis paket aplikasi yang didalamnya terdapat beberapa perangkat lunak lain yang mendukung dalam pengolahan website atau aplikasi berbasis web. Adapun proses instalasi yang akan kita lakukan terlihat pada penjelasan berikut ini:

Silahkan unduh terlebih dahulu paket aplikasi XAMPP versi 7.1.1 di internet, kemudian klik dua kali file master XAMPP yang telah anda unduh tersebut (contoh yang penulis pakai disini adalah **xampp-win32-7.1.1-0-VC14-installer.exe**).

Versi XAMPP yang terbaru sudah masuk versi 7.3.2 dengan file size 104MB, rilis pada 20 November 2019, namun disini saya masih menggunakan versi 7.1.1.

Tabel 2.3. Link Download XAMPP

Software	Link Download
XAMPP	https://xampp.site/download/xampp-for-windows-32-bit/



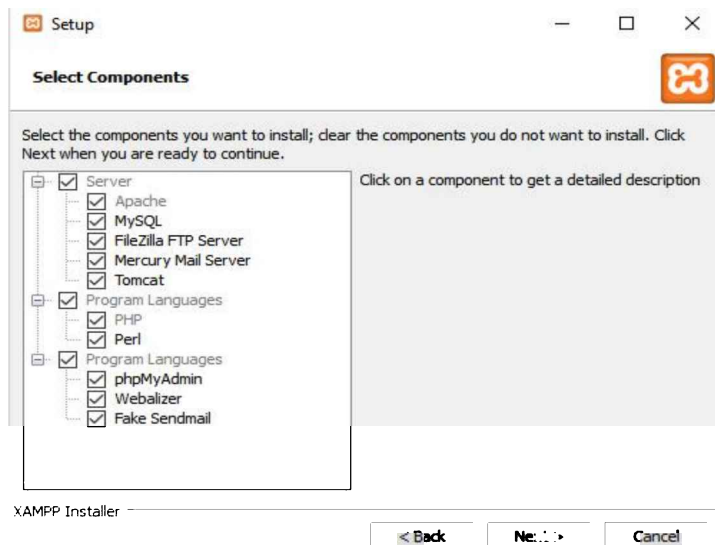
Gambar 2.6 Master Instaler XAMPP versi 7.1.1

Gambar diatas merupakan file installer dari XAMPP versi 7.1.1 untuk win32. Setelah download, anda dapat melakukan klik 2 kali pada installer tersebut, maka akan tampil halaman seperti berikut:



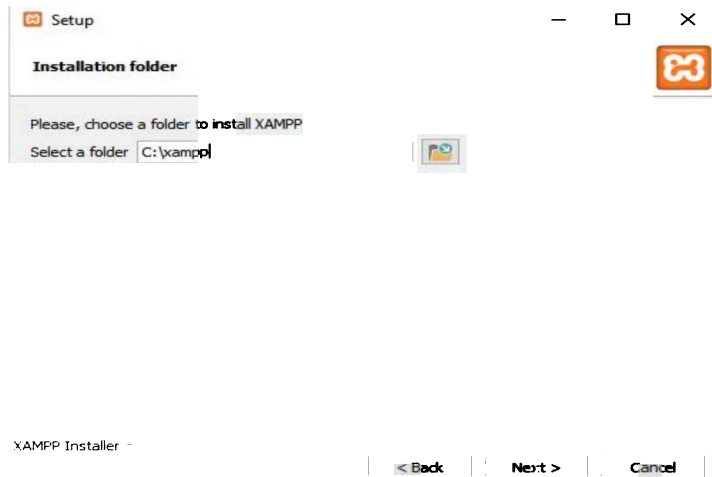
Gambar 2.7. Halaman awal Installer XAMPP

- Klik *button Next* pada halaman Setup - XAMPP yang muncul.



Gambar 2.8 Halaman *Setup Select Component*

- Klik *button Next* pada jendela *setup, select component* yang muncul. Biarkan saja pilihan pada posisi *default*, kecuali jika anda tidak menginginkan beberapa pilihan yang akan dilakukan instalasi.
- Pada jendela berikut ini, anda diminta untuk memilih pada drive atau folder mana file XAMPP akan dilakukan pemasangan, secara default anda dapat memilih pada drive **C:\xampp**. Jika sudah dapat menentukan posisi file yang akan di install, maka klik *button next*.



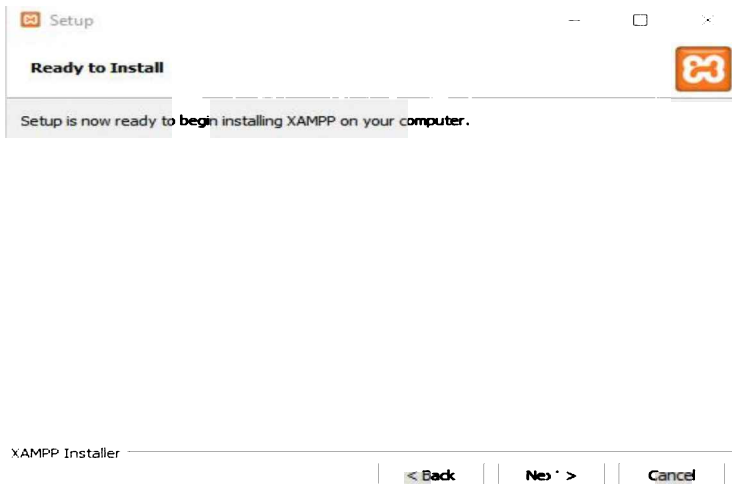
Gambar 2.9 Halaman *Instalation Folder*

- Halaman Berikut ini merupakan Bitnami for XAMPP sebagai software pendukung dari XAMPP, biarkan saja di ceklis secara default, klik *next*.



Gambar 2.10 Halaman *Bitnami for XAMPP*

- Kemudian akan muncul halaman untuk melakukan proses instalasi, klik *next*.



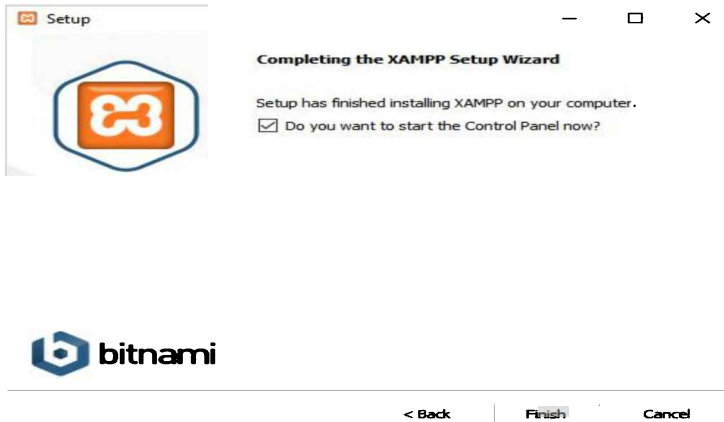
Gambar 2.11 Halaman Memulai Instalasi

- Kemudian muncul halaman proses instalasi yang sedang berjalan, dan tungguilah dalam beberapa menit.



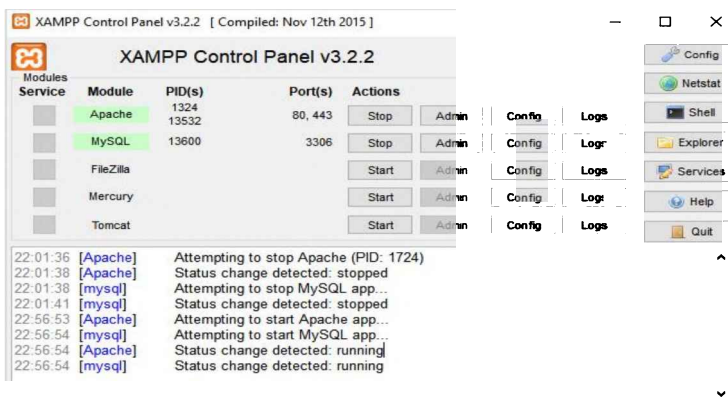
Gambar 2.12 Halaman Proses Instalasi Paket XAMPP

- Setelah halaman finish muncul, maka anda dapat mengaktifkan fungsi Apache dan MySQL pada halaman XAMPP-Control panel, klik finish.



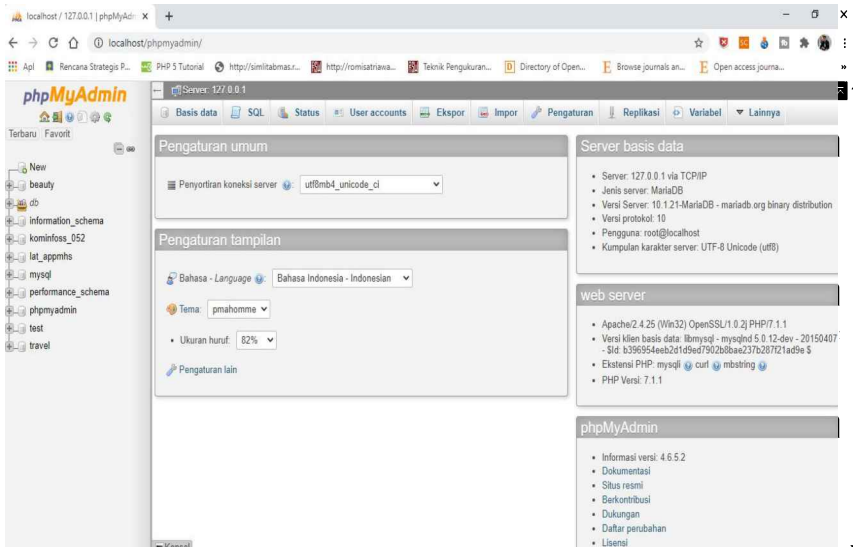
Gambar 2.13 Halaman Instalasi *Finish*

- Kemudian pada halaman XAMPP *Control-panel*, klik *start* pada *module* Apache dan MySQL. Maka dengan ini, anda telah dapat menggunakan fungsi web server apache dan MySQL.



Gambar 2.14 Halaman XAMPP *Control-panel*

Tidak sampai di sini, anda harus menguji apakah program XAMPP yang dipasang telah berjalan sebagaimana mestinya, silahkan anda buka browser misalnya chrome, dan ketik pada URL: **localhost/phpmyadmin**. Maka akan tampil halaman seperti berikut.



Gambar 2.15 Halaman phpmyadmin

Halaman ini dapat anda penggunaan untuk membuat database dan menguji website atau aplikasi web yang anda buat. Setiap kali anda bekerja dalam pembuatan web atau aplikasi berbasis web dalam PC-lokal atau bersifat **localhost**, anda diharuskan menjalankan fungsi **Apache web server** dan **MySQL database** dalam kondisi **running**. Kondisi itu dapat anda lihat dengan mengklik icon XAMPP control panel yang terdapat pada gambar 2.14.

2. Instalasi Visual Studio Code (PHP Editor)

Sebenarnya banyak sekali aplikasi PHP Editor yang ada, seperti Notepad++, Macromedia CS3, Macromedia Dreamweaver 8 dan Visual Studi Code. Namun pada prinsipnya PHP editor apapun kita dapat bekerja, tetapi disini saya menggunakan PHP editor **Visual Studi Code** versi 1.39.2. Berikut langkah-langkah dalam instalasi Visual Studio Code:

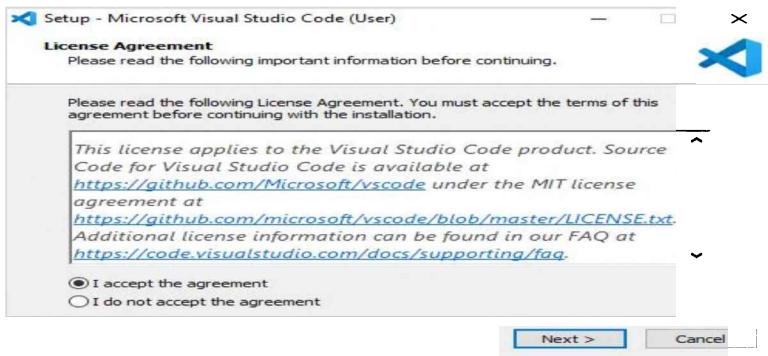
Tabel 2.4. *Link Download* Visual Studio Code

Software	Link Download
Visual Studio Code	https://code.visualstudio.com/download



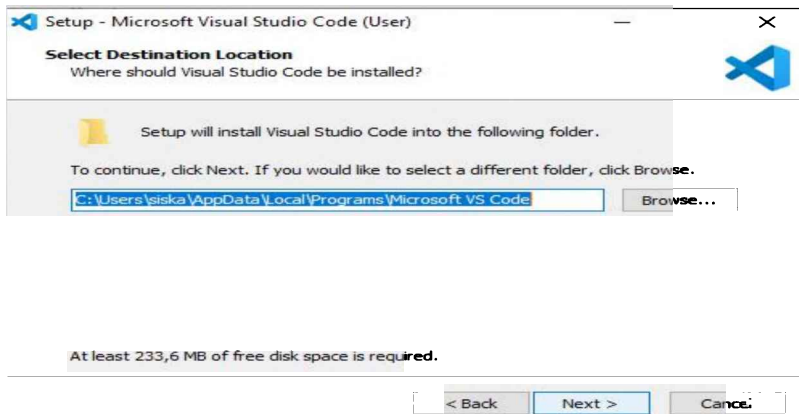
Gambar 2.16 Master installer Visual Studi Code

- Setelah melakukan klik 2 kali pada master installer VS Code, maka akan tampil halaman seperti gambar berikut ini, dan kemudian silahkan anda pilih *“I accept the agreement”* dan klik *button next*.



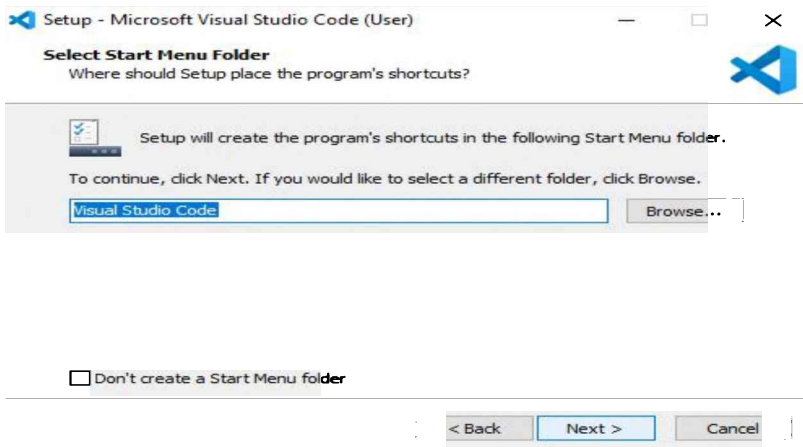
Gambar 2.17 Halaman *License Agreement*

- Kemudian pilih lokasi tempat anda akan menyimpan file instalasi VS Code, secara default akan diarahkan pada drive C:\ dan klik *next*.



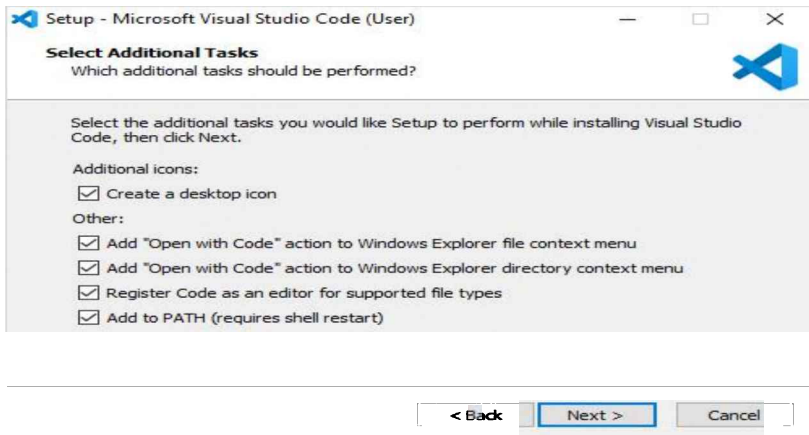
Gambar 2.18 Halaman *Location Folder for Instalation*

- Berikut ini halaman pernyataan akan membuat shortcut VS Code pada start menu windows. Disini anda bisa memilih ceklis atau tidak, jika sudah klik *next*.



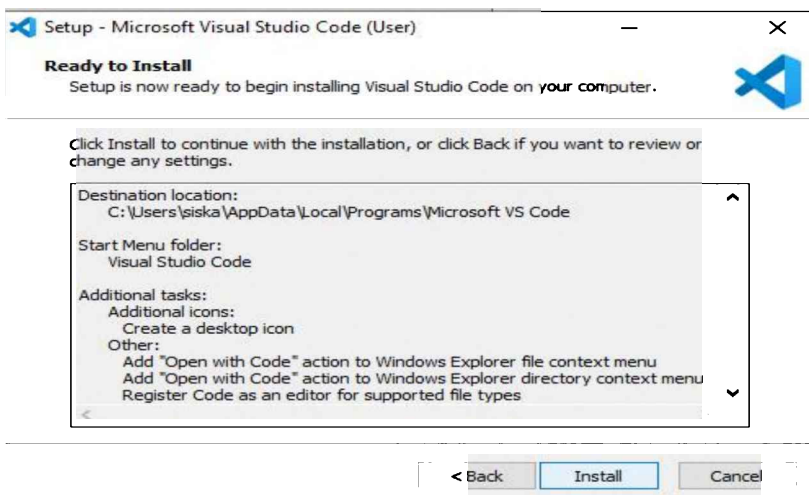
Gambar 2.19. Halaman *create shortcut* pada *start menu windows*

- Pada bagian halaman ini, biarkan saja default dan diceklis semua pilihan, kemudian klik *next*.



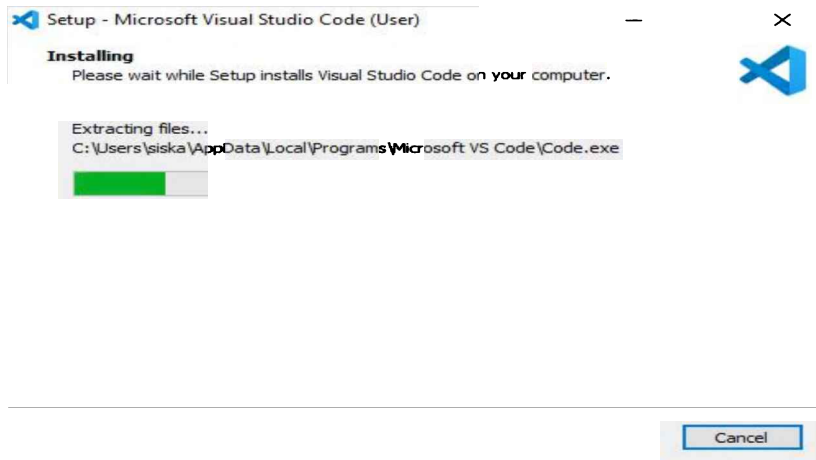
Gambar 2.20 Halaman *Additional Tasks*

- Halaman berikut merupakan halaman awal proses instalasi akan dilakukan, klik tombol install.



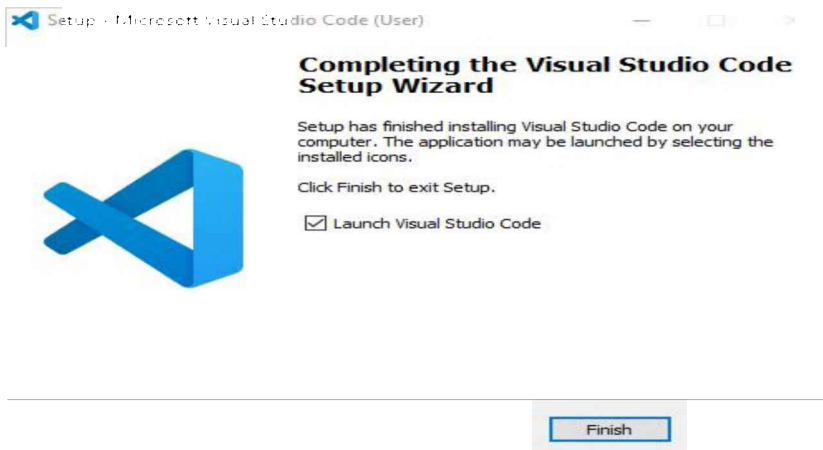
Gambar 2.21. Halaman Awal proses instalasi VC Code

- Setelah itu akan tampil halaman dari berjalannya proses instalasi, dan biarkan beberapa menit.



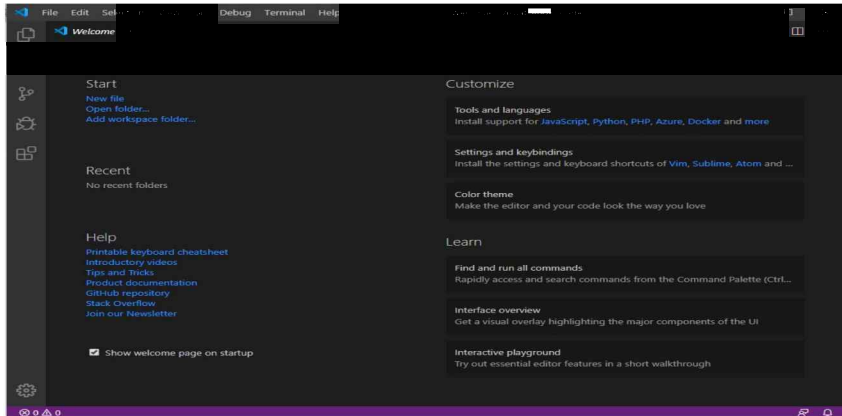
Gambar 2.21. Halaman Proses Instalasi VC Code

- Tunggulah beberapa menit sampai proses pemasangan selesai.



Gambar 2.22. Proses *Finish* pemasangan aplikasi

- Setelah proses pemasangan selesai, maka anda akan melihat menu awal dari aplikasi VS Code seperti dibawah ini, dengan aplikasi ini anda akan dimudahkan dalam pembangunan website atau aplikasi berbasis web.



Gambar 2.23 Tampilan awal Aplikasi VS Code

VS Code memiliki banyak fitur yang memudahkan anda dalam membangun website atau aplikasi berbasis web, di antaranya sebagai berikut:

1. *Cross platform*
2. *Lightweight*
3. *Powerful*
4. *Code Debugging*
5. *Source control*
6. *Integrated terminal*
7. *Other*

Dasar Pembangunan Website dan HTML

A. Perancangan Website

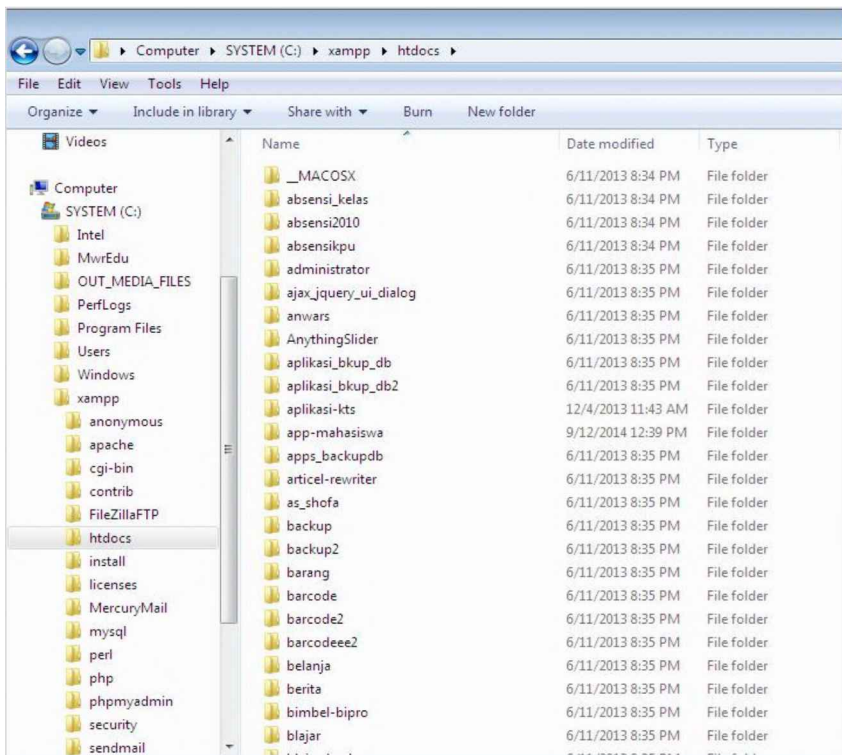
Membangun sebuah website atau aplikasi berbasis web, tidak terlepas dari proses perancangan. Perancangan pola ataupun sketsa menjadi dasar dalam pembangunan sebuah website, perancangan ini menjadi acuan yang sangat penting dalam pembangunan website atau aplikasi yang berbasis web. Secara umum, perancangan sebuah website yang baik didasari dengan pembuatan pola yang terdiri dari **header**, **navigation**, **sidebar**, **content** dan **footer**. Di bawah ini dapat anda lihat salah satu contoh dari bentuk dasar suatu website.

Header	
Navigation	
Sidebar	Content
Footer	

Gambar 3.1 Contoh Rancangan Dasar Website

B. Lokasi Penyimpanan File PHP

Pada awalnya website dan aplikasi berbasis web yang kita bangun masih berjalan pada mode *localhost* atau belum terhubung ke internet, sehingga lebih mudah melakukan perubahan-perubahan pada *script*. Maka dari itu kita perlu mengetahui direktori penyimpanan seluruh file PHP yang ada pada komputer. Perhatikan gambar berikut:

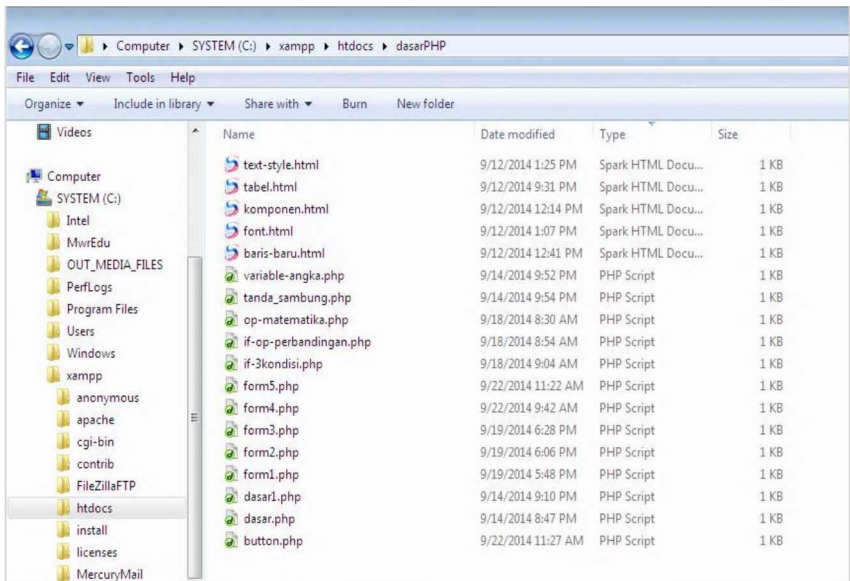


Gambar 3.2 Direktori Penyimpanan file PHP

Adapun lokasi penyimpanan file PHP tersebut adalah :

C:\xampp\htdocs\folder project\file-PHP

Di dalam folder htdocs, anda dapat membuat sebuah folder project yang akan berisi file-file seperti PHP, HTML, CSS, Java script, JPG atau file lainnya. Perhatikan gambar di bawah ini:



Gambar 3.3. Lokasi penyimpanan File PHP, HTML, CSS dan lainnya

Apabila anda akan mengakses folder project yang anda buat, maka dapat dilakukan dengan cara mengetikkan alamat pada URL browser seperti contoh berikut: localhost/**folder project**.

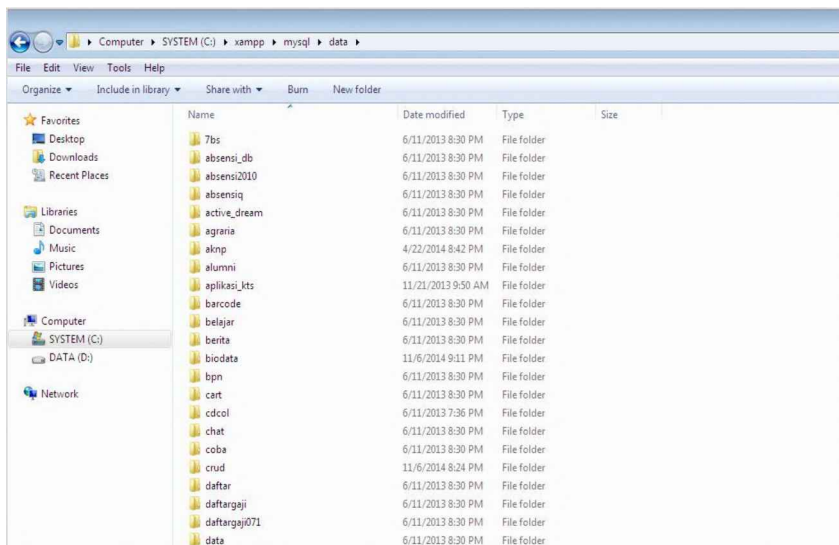
Folder project diberi nama **modulphp** maka diketik: localhost/modulphp



Gambar 3.4 Lokasi penyimpanan File PHP, HTML, CSS dan lainnya

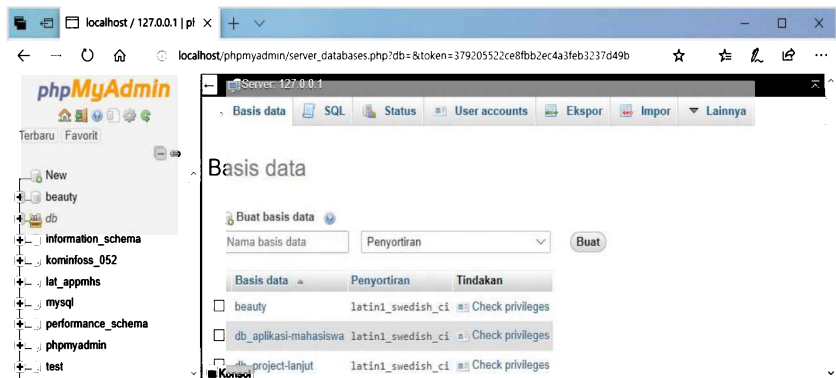
C. Lokasi Penyimpanan File *database* MySQL

File *database* MySQL terletak pada direktori C:\xampp\mysql\data, seperti gambar dibawah ini :



Gambar 3.5 File *database* MySQL

Ketika anda telah membuat *database* pada halaman phpmyadmin, akses: **localhost/phpmyadmin** maka secara otomatis akan terbentuk dengan sendirinya folde tempat menyimpan file *database* yang telah anda buat tersebut.



Gambar 3.6 Halaman phpmyadmin “Create Database”

D. Elemen dasar HTML

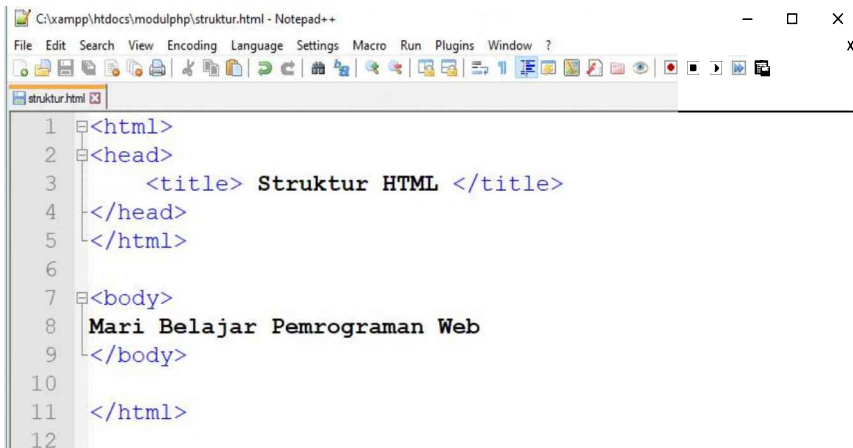
Dalam pembangunan sebuah website, tidak terlepas dari pada fungsi html sebagai elemen dasar pembangunan website, ada baiknya sedikit membahas mengenai elemen dasar html, seperti contoh berikut.

1. Struktur HTML

Tabel 3.1. Code Struktur Dasar HTML

Code Tag Pembuka	Code Tag Penutup
<html>	</html>
<head>	</head>
<title>	</title>
<body>	</body>

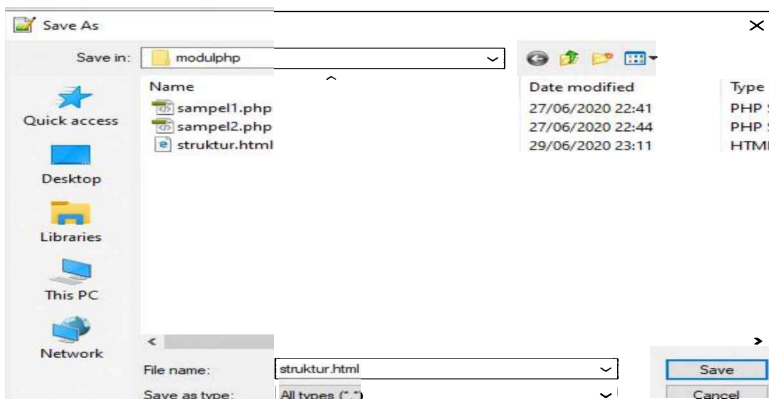
- Cobalah buat sebuah file dengan nama: **struktur.html** menggunakan notepad++ dengan contoh script seperti dibawah ini:



```
1 <html>
2 <head>
3     <title> Struktur HTML </title>
4 </head>
5 </html>
6
7 <body>
8     Mari Belajar Pemrograman Web
9 </body>
10
11 </html>
12
```

Gambar 3.7 File struktur.html

- Setelah seluruh script html diatas anda ketik, silahkan lakukan penyimpanan dengan membuat nama file: **struktur.html** dan jangan lupa merubah pengaturan penyimpanan pada pilihan *save as type*: menjadi **"All types"**. Lihat pada gambar 3.8 dibawah ini:

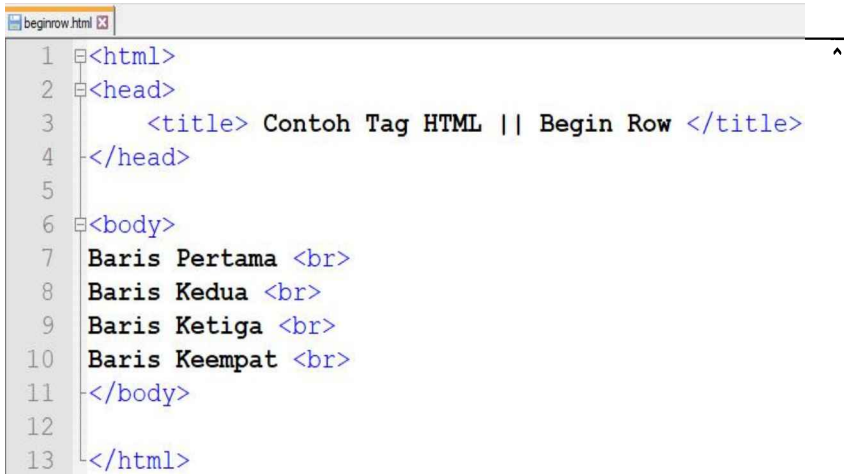


Gambar 3.8 Pengaturan *save as type* menjadi *All types*

2. Latihan HTML

Penggunaan Tag *Begin Row*

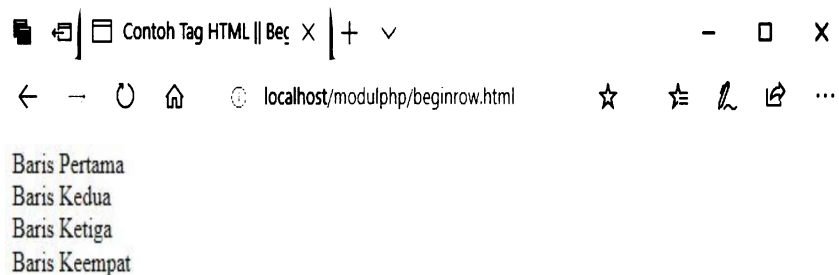
- Contoh Tag `
` yang berfungsi untuk membuat baris baru pada sebuah paragraph, buatlah dengan nama file: **beginrow.html**



```
1 <html>
2 <head>
3     <title> Contoh Tag HTML || Begin Row </title>
4 </head>
5
6 <body>
7     Baris Pertama <br>
8     Baris Kedua <br>
9     Baris Ketiga <br>
10    Baris Keempat <br>
11 </body>
12
13 </html>
```

Gambar 3.9 Tag `
` sebagai pembuat baris baru

- Tampilan hasil pengkodean dengan fungsi *begin row* `
`



Gambar 3.10 Hasil Tampilan Tag `
`.

Penggunaan Tag font

- Contoh Tag `...` ini dapat digunakan untuk mengatur ukuran, warna dan jenis pada huruf, sebagai contoh buatlah dengan nama file: **font.html**

Tabel 3.2. Tag `` dan atribut yang digunakan

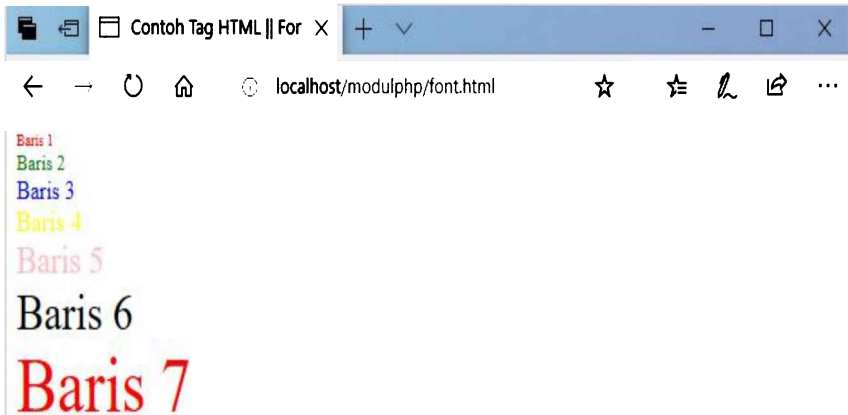
Tag Pembuka	Tag Penutup	Atribut
<code></code>	<code></code>	Size; Color; Face;



```
1 <html>
2 <head>
3     <title> Contoh Tag HTML || Font </title>
4 </head>
5
6 <body>
7     <font size=1 color=red> Baris 1 </font><br>
8     <font size=2 color=green> Baris 2 </font><br>
9     <font size=3 color=blue> Baris 3 </font><br>
10    <font size=4 color=yellow> Baris 4 </font><br>
11    <font size=5 color=pink> Baris 5 </font><br>
12    <font size=6 color=black> Baris 6 </font><br>
13    <font size=7 color=red> Baris 7 </font><br>
14
15 </body>
16
17 </html>
```

Gambar 3.11 Tag `...`

- Pada gambar diatas, kita menggunakan dua fungsi sekaligus yaitu fungsi `` dengan atribut: **size** dan **color**, beserta fungsi `
`
- Tampilan hasilnya dapat dilihat pada gambar dibawah ini:



Gambar 3.12 Hasil Tampilan Tag `...`

- Penggunaan Tag `` dapat dilihat pada gambar dibawah ini, buat dengan nama: **font-face.html**



Gambar 3.13 Tag ` `

Anda dapat menambahkan berbagai jenis *font style* dengan memanfaatkan Tag ``, sesuaikan dengan kebutuhan anda. Adapun hasilnya dapat anda lihat pada gambar 3.14 dibawah ini:



Gambar 3.14 Hasil Tampilan Penggunaan Tag ``

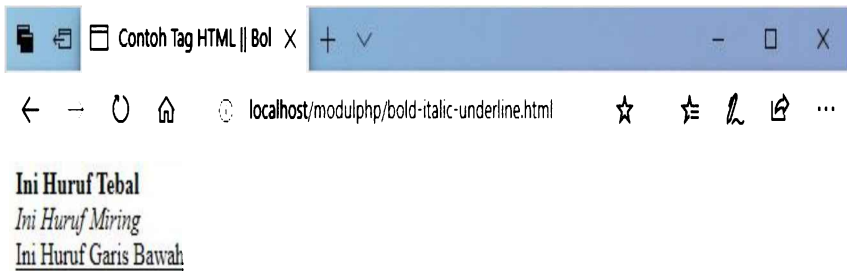
Penggunaan Tag (**bold**, *italic* dan underline)

- Contoh Tag ``, `<i></i>` dan `<u></u>`

Fungsi ini digunakan sebagai pembuatan huruf tebal, miring dan garis bawah, silahkan buat dengan nama file: **bold-italic-underline.html** lihat contoh berikut:



Gambar 3.15 Penggunaan Tag ``, `<i></i>`, dan `<u></u>`



Gambar 3.16 Hasil Tampilan Tag ``, `<i></i>`, dan `<u></u>`

Penggunaan Tag *Horizontal Line*

- Contoh Tag `<hr>`

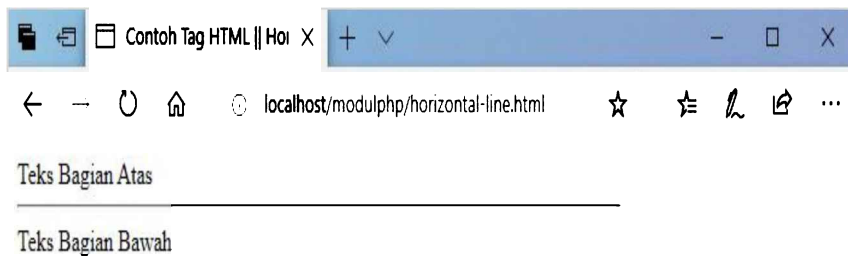
Tabel 3.3. Tag `<hr>` dan atribut yang digunakan

Tag	Atribut
<code><hr></code>	<code>width;</code> <code>align;</code>

Tag `<hr>` berfungsi membuat sebuah garis datar, yang dapat anda lihat pada contoh dibawah, nama file: **horizontal-line.html**



Gambar 3.17 Penggunaan Tag `<hr>`



Gambar 3.18 Hasil Tampilan Tag `<hr>`

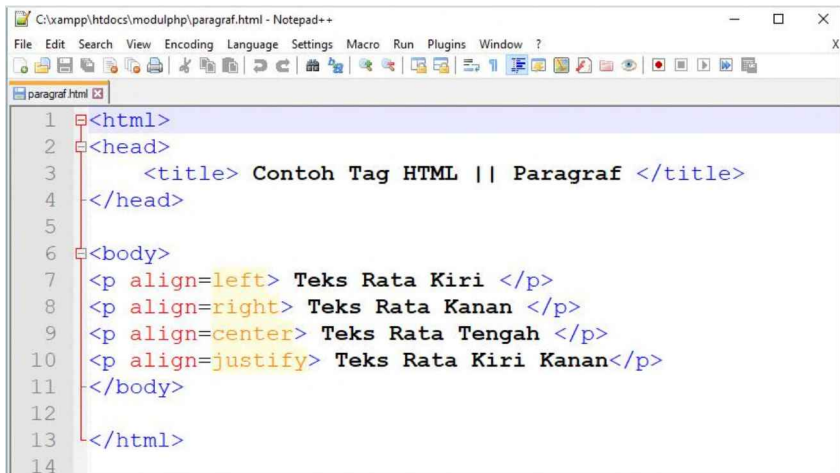
Penggunaan Tag *Paragraph*

- Contoh Tag `<p></p>`

Tag `<p></p>` digunakan untuk membuat paragraf, jika tanpa tag penutup hasilnya seperti menggunakan 2 kali `
`, buat dengan nama file: **paragraf.html**

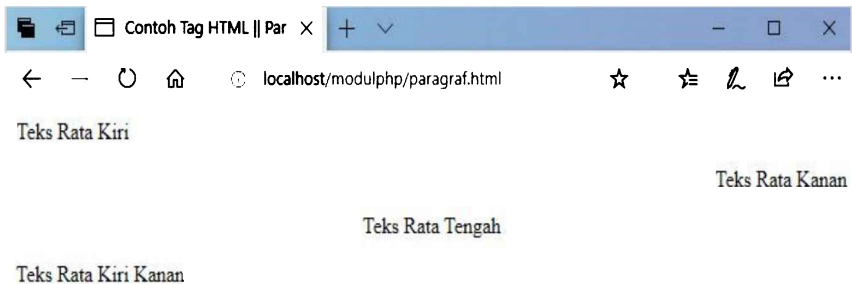
Tabel 3.4. Tag `<p></p>` dan atribut yang digunakan

Tag Pembuka	Tag Penutup	Atribut
<code><p></code>	<code></p></code>	<code>align;</code>



```
1 <html>
2 <head>
3   <title> Contoh Tag HTML || Paragraf </title>
4 </head>
5
6 <body>
7   <p align=left> Teks Rata Kiri </p>
8   <p align=right> Teks Rata Kanan </p>
9   <p align=center> Teks Rata Tengah </p>
10  <p align=justify> Teks Rata Kiri Kanan</p>
11 </body>
12
13 </html>
14
```

Gambar 3.19 Penggunaan Tag <p></p>



Gambar 3.20 Hasil Tampilan Tag <p></p>

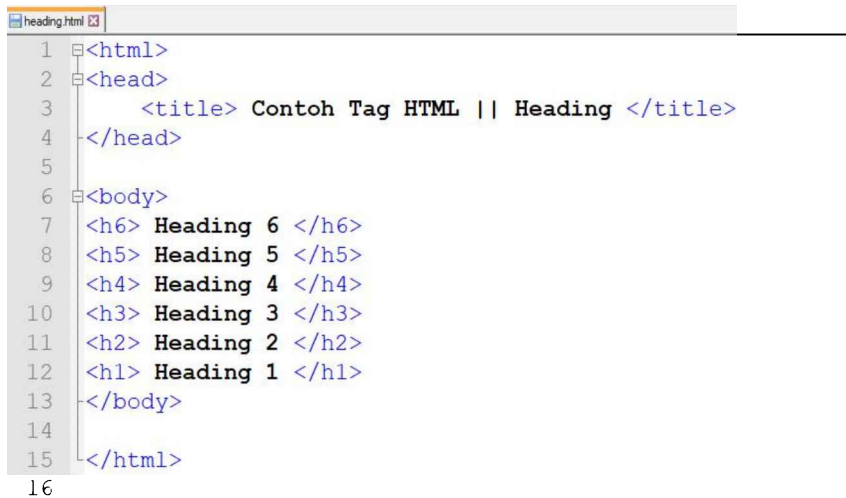
Penggunaan Tag *Heading*

- Tag *Heading* berfungsi untuk mendefinisikan judul, bukan untuk memperbesar atau menebalkan teks.

Tabel 3.5. Tag *heading*

Tag Pembuka	Tag Penutup
<h1>	</h1>
<h2>	</h2>
<h3>	</h3>
<h4>	</h4>
<h5>	</h5>
<h6>	</h6>

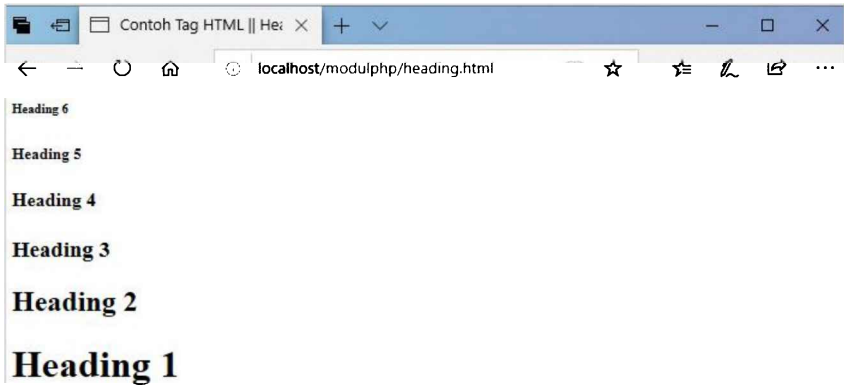
- Tag *Heading* memiliki tingkat besar ukuran, mulai dari <h1> yang paling besar sampai <h6> yang paling kecil, silahkan buat nama file: **heading.html** seperti contoh berikut:



```
1 <html>
2 <head>
3     <title> Contoh Tag HTML || Heading </title>
4 </head>
5
6 <body>
7     <h6> Heading 6 </h6>
8     <h5> Heading 5 </h5>
9     <h4> Heading 4 </h4>
10    <h3> Heading 3 </h3>
11    <h2> Heading 2 </h2>
12    <h1> Heading 1 </h1>
13 </body>
14
15 </html>
```

Gambar 3.21 Penggunaan Tag *heading* <h1> s.d <h6>

- Hasilnya dapat dilihat pada gambar berikut:



Gambar 3.22 Hasil Penggunaan Tag *heading* <h1> s.d <h6>

Penggunaan Tag *Order List*

- Contoh Tag

Order list (biasa dikenal dengan *numbering*) biasa digunakan untuk membuat urutan pada sesuatu.

Tabel 3.6. Tag *Order List*

Tag Pembuka	Tag Penutup	Atribut
		Type: 1, A, a, I, i
		

- Silahkan buat file baru, dengan nama file: **order-list.html**

```

1 <html>
2 <head>
3   <title> Contoh Tag HTML || Order List </title>
4 </head>
5
6 <body>
7   <ol type="I">
8     <li> Urutan Satu </li>
9     <li> Urutan Dua </li>
10    <li> Urutan Tiga </li>
11  </ol>
12  <ol type="A">
13    <li> Urutan Satu </li>
14    <li> Urutan Dua </li>
15    <li> Urutan Tiga </li>
16  </ol>
17  <ol type="1">
18    <li> Urutan Satu </li>
19    <li> Urutan Dua </li>
20    <li> Urutan Tiga </li>
21  </ol>
22 </body>
23
24 </html>

```

Gambar 3.23 Penggunaan Tag *Order List* `` `` dan `` ``

- Pada contoh diatas, terdapat tiga contoh penggunaan atribut type yaitu: I, A dan 1. Anda dapat menambahkan type atribut lain, sesuai dengan kebutuhan anda, misalnya “a dan i”. Hasil dari code diatas, dapat anda lihat pada gambar berikut ini:

I. Urutan Satu
 II. Urutan Dua
 III. Urutan Tiga

A. Urutan Satu
 B. Urutan Dua
 C. Urutan Tiga

1. Urutan Satu
 2. Urutan Dua
 3. Urutan Tiga

Gambar 3.24 Hasil Penggunaan Tag ```` dan ````

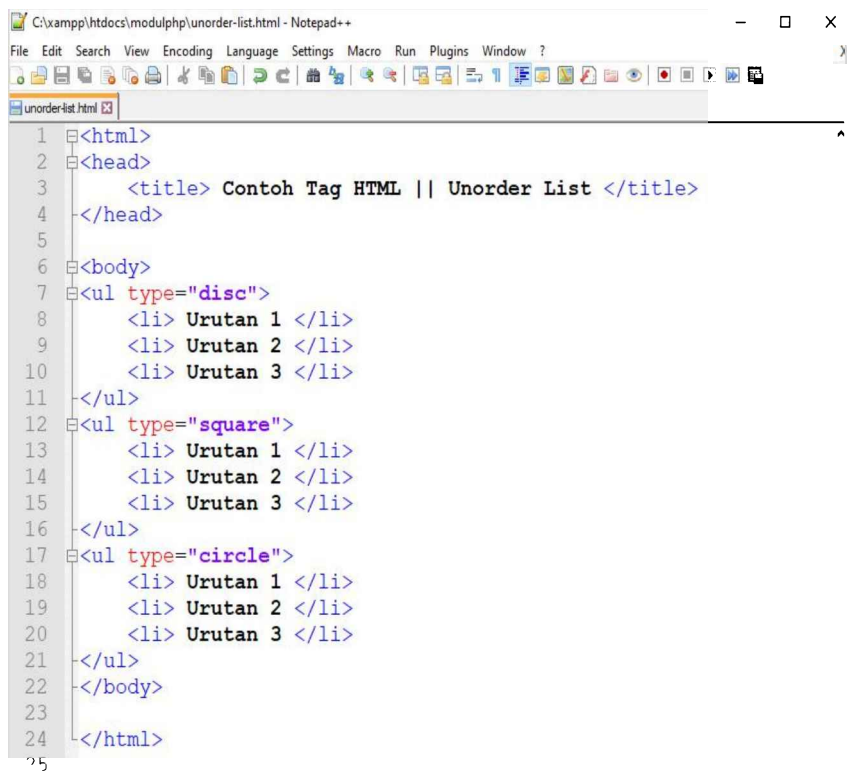
Penggunaan Tag *Unorder List*

- Contoh Tag ``, *Unorder List* biasa dikenal dengan Bullet.

Tabel 3.7. Tag *Unorder list*

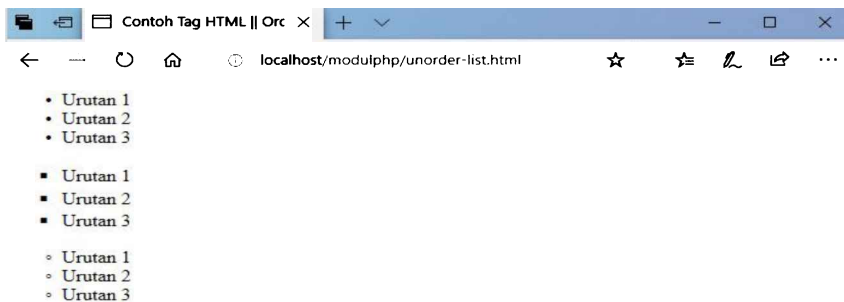
Tag Pembuka	Tag Penutup	Atribut
<code></code>	<code></code>	Type: disc; square; circle;
<code></code>	<code></code>	

- Silahkan buat file baru, dengan nama file: **unorder-list.html**



```
1 <html>
2 <head>
3   <title> Contoh Tag HTML || Unorder List </title>
4 </head>
5
6 <body>
7   <ul type="disc">
8     <li> Urutan 1 </li>
9     <li> Urutan 2 </li>
10    <li> Urutan 3 </li>
11  </ul>
12  <ul type="square">
13    <li> Urutan 1 </li>
14    <li> Urutan 2 </li>
15    <li> Urutan 3 </li>
16  </ul>
17  <ul type="circle">
18    <li> Urutan 1 </li>
19    <li> Urutan 2 </li>
20    <li> Urutan 3 </li>
21  </ul>
22 </body>
23
24 </html>
```

Gambar 3.25 Penggunaan Tag `/ul>` dan ``



Gambar 3.26 Hasil Penggunaan Tag `` dan ``

Penggunaan Tabel

▪ Contoh Tag `<table></table>`

Pembuatan tabel berfungsi sebagai salah satu pembentuk struktur dari sebuah website, selain dari pada `<div>`. Struktur `<table>` terdiri dari `<tr></tr>` dan `<td></td>`.

Tabel 3.8. Tag tabel dan atribut yang digunakan

Tag Pembuka	Tag Penutup	Atribut
<code><table></code>	<code></table></code>	<code>border; align; width; height; bgcolor;</code>
<code><tr></code>	<code></tr></code>	<code>align; width; height; bgcolor; colspan;</code>
<code><td></code>	<code></td></code>	<code>align; width; height; bgcolor;</code>

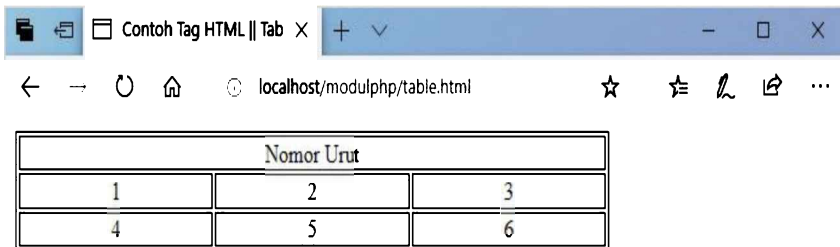
- Pada tabel diatas terlihat ada tiga struktur pembentuk tabel. Tag `<table>` sebagai pembentuk struktur atau kerangka utama. Tag `<tr>` pembentuk *row* atau baris, sedangkan Tag `<td>` sebagai pembentuk *column*. Silahkan buat latihan berikut dengan nama file: **table.php**

```

1 <html>
2 <head>
3   <title> Contoh Tag HTML || Tabel </title>
4 </head>
5
6 <body>
7   <table width=500 border=1>
8     <tr>
9       <td colspan=3 align=center>Nomor Urut</td>
10    </tr>
11    <tr>
12      <td align=center>1</td>
13      <td align=center>2</td>
14      <td align=center>3</td>
15    </tr>
16    <tr>
17      <td align=center>4</td>
18      <td align=center>5</td>
19      <td align=center>6</td>
20    </tr>
21  </table>
22 </body>
23 </html>
24

```

Gambar 3.27 Penggunaan Tag <table>, <tr>, <td>



Nomor Urut		
1	2	3
4	5	6

Gambar 3.28 Hasil Penggunaan Tag <table>, <tr> dan <td>

Pada gambar di atas, terlihat penggunaan Tag <table> disertai dengan atribut “width dan border” dan juga Tag <td> disertai “colspan dan align”. Atribut lain dapat juga anda gunakan sesuai dengan kebutuhan.

Penggunaan Image

- Contoh Tag

Tag berfungsi menampilkan gambar.

Tabel 3.9. Tag image dan atribut yang digunakan

Tag Pembuka	Tag Penutup	Atribut
	-	src; align; width; height; alt or title;

Keterangan:

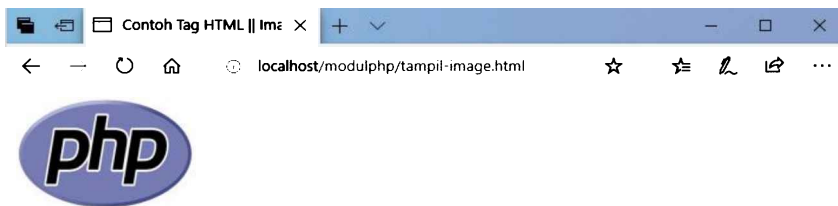
- *src* (letak file gambar)
- *align* (perataan gambar)
- *width* (lebar gambar dlm persen atau pixel)
- *height* (tinggi gambar dlm persen atau pixel)
- *alt* atau *title* (keterangan gambar)

Buatlah latihan dengan contoh berikut, nama file: **tampil-image.html**

```
tampil-image.html
1 <html>
2 <head>
3   <title> Contoh Tag HTML || Image </title>
4 </head>
5
6 <body>
7   <img src=img/php.png align=left alt=GambarPHP width=150 height=100>
8 </body>
9
10 </html>
11
```

Gambar 3.29 Penggunaan Tag beserta atribut

Terlihat pada gambar di atas, atribut “src” berfungsi mengarah pada lokasi tempat gambar tersimpan, pada contoh ini gambar tersimpan di dalam folder modulphp > folder **img**, dan sertakan nama file gambar seperti contoh di sini adalah file php.png (lokasi file gambar harus cocok).



Gambar 3.30 Hasil Tampilan Tag

Penggunaan Tag Link

- Contoh Tag Link <a>

Link berfungsi menghubungkan satu halaman dengan halaman lainnya.

Tabel 3.10. Tabel Tag link dan atribut yang digunakan

Tag Pembuka	Tag Penutup	Atribut
<a>		href; name;

Contoh latihan: Buatlah dua buah file yang nantinya akan kita hubungkan, pada setiap halaman terdapat link yang menghubungkan halaman 1 ke halaman 2, sehingga ke dua file tersebut menjadi saling terhubung.

Nama file:

- halaman1.html
- halaman2.html

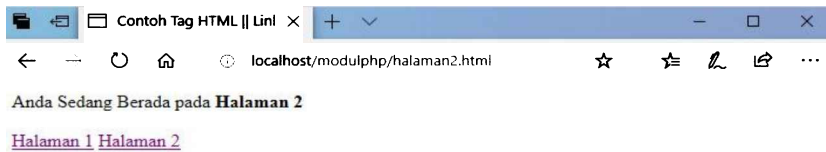
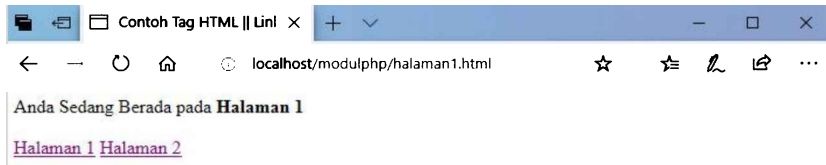
Pada gambar 3.31 script untuk “halaman1.html”, dan pada gambar 3.32 script untuk “halaman2.html”.

```
halaman1.html x
1 <html>
2 <head>
3     <title> Contoh Tag HTML || Link </title>
4 </head>
5
6 <body>
7     Anda Sedang Berada pada <b>Halaman 1</b> <br><br>
8     <a href=halaman1.html>Halaman 1</a>
9     <a href=halaman2.html>Halaman 2</a>
10 </body>
11
12 </html>
13
```

Gambar 2.31 Penggunaan Tag link pada halaman1.html

```
halaman2.html x
1 <html>
2 <head>
3     <title> Contoh Tag HTML || Link </title>
4 </head>
5
6 <body>
7     Anda Sedang Berada pada <b>Halaman 2</b> <br><br>
8     <a href=halaman1.html>Halaman 1</a>
9     <a href=halaman2.html>Halaman 2</a>
10 </body>
11
12 </html>
13
```

Gambar 3.32 Penggunaan Tag link pada halaman2.html



Gambar 3.33 Hasil Penggunaan Tag *link* pada halaman1 dan halaman2

Pada gambar di atas, dua buah halaman telah dihubungkan menggunakan link `<a>` sehingga dengan mengklik *link* pada salah satu halaman, maka anda akan langsung di hantar kehalaman yang dituju.

E. Jenis-jenis Tag HTML

Tag merupakan sesuatu yang dipergunakan untuk menandai elemen-elemen dalam suatu dokumen HTML. HTML membutuhkan cara untuk memberitahu sebuah *web browser* apa fungsi dari teks. Apakah teks itu ditulis sebagai sebuah *paragraf*, *list*, atau sebagai *link*? Dalam HTML, tanda ini dikenal dengan istilah *tag*. *Tag* merupakan suatu pasangan

yang terdiri 2 bagian yang disebut dengan *tag* pembuka dan *tag* penutup.

Tabel 3.11. Jenis-jenis tag HTML

Tag Pembuka	Tag Penutup	Keterangan
<a>		Menyisipkan <i>hyperlink</i>
		Menebalkan teks
<big>	</big>	Membesarkan ukuran teks
<blink>	</blink>	Teks berkedip
<blockquote>	</blockquote>	Menandai sebagai kutipan
<body>	</body>	Tubuh dari dokumen
 	</br>	Baris baru
<caption>	</caption>	Judul untuk tabel
<center>	</center>	Perataan teks tengah
<cite>	</cite>	Mendefinisikan kutipan
<code>	</code>	Mendefinisikan bagian dari kode komputer
		Memberi garis tengah teks
<div>	</div>	Mendefinisikan sebuah <i>section</i> dalam dokumen
		Fungsi <i>tag</i> ini sama dengan <i>tag</i> <i> (teks miring)
<embed>	</embed>	Menambahkan <i>sound</i> atau <i>file</i> avi kehalaman web
		Menentukan atribut <i>font</i>
<form>	</form>	Mendefinisikan <i>form input</i>
<frame>	</frame>	Mendefinisikan <i>frame</i>
<h1> to <h6>	</h1> to </h6>	Merupakan awal dari sebuah <i>header</i> /judul dokumen.
<head>	</head>	Mendefinisikan <i>head document</i>
<hr>	</hr>	Membuat garis <i>horizontal</i>
<i>	</i>	Membuat teks miring
	-	Menyisipkan gambar

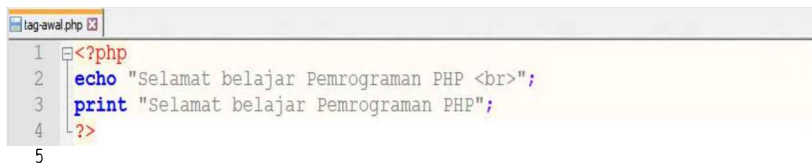
<ins>	</ins>	Membuat teks bergaris bawah
<label>	</label>	Mendefinisikan label untuk sebuah elemen <input>
<legend>	</legend>	Mendefinisikan sebuah <i>caption</i> untuk elemen <fieldset>
		Daftar item
		Daftar terurut
<option>	</option>	Menampilkan beberapa pilihan dalam daftar <i>dropdown</i>
<p>	</p>	Membuat sebuah <i>paragraph</i>
<param>	</param>	Mendefinisikan parameter untuk objek
<small>	</small>	Memperkecil ukuran teks dari ukuran <i>default</i>
<script>	</script>	Mendefinisikan <i>client-side-script</i> (pada javascript)
<select>	</select>	Membuat daftar <i>dropdown</i> pada <option>
		Mendefinisikan sebuah <i>section</i> dalam dokumen
		Membuat teks tebal, sama dengan fungsi tag
<style>	</style>	Mendefinisikan informasi <i>style</i> dokumen HTML
<table>	</table>	Membuat tabel
<td>	</td>	Menyisipkan dat sel tabel (kolom pada tabel)
<textarea>	</textarea>	Mendefinisikan sebuah <i>control input multiline</i>
<th>	</th>	Mendefinisikan sel <i>header</i> dalam tabel
<tr>	</tr>	Mendefinisikan baris tabel (<i>row</i> tabel)
<u>	</u>	Membuat garis bawah (<i>underline</i>)
		Daftar tidak terurut (dalam bentuk <i>bullets</i>)
<var>	</var>	Mendefinisikan sebuah <i>variable</i>

Bab 4

Dasar Penulisan PHP

A. Tag Awal PHP

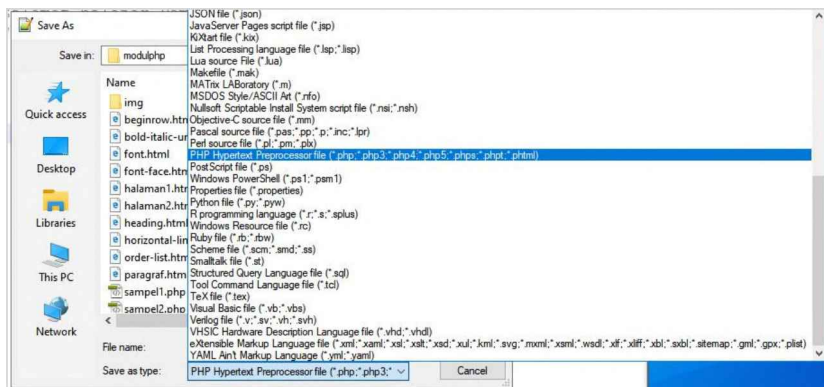
Penulisan php diawali `<?php` dan di akhiri `?>`, penulisan PHP lazimnya ditulis diantara blok `<?php ... ?>`.



```
1 <?php
2 echo "Selamat belajar Pemrograman PHP <br>";
3 print "Selamat belajar Pemrograman PHP";
4 ?>
```

Gambar 4.1 Penulisan dasar PHP

Perintah `echo` dan `print` pada PHP sama-sama berfungsi menampilkan teks. Kemudian proses penyimpanan file pada notepad++ dengan cara memilih save as type “PHP” atau “All Types” seperti gambar berikut:

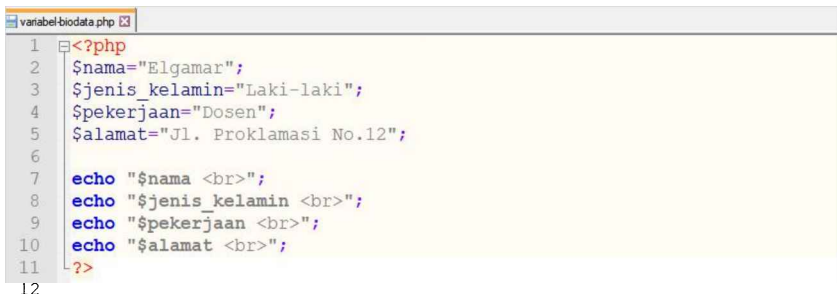


Gambar 4.2 Penulisan dasar PHP

B. Penggunaan Variabel

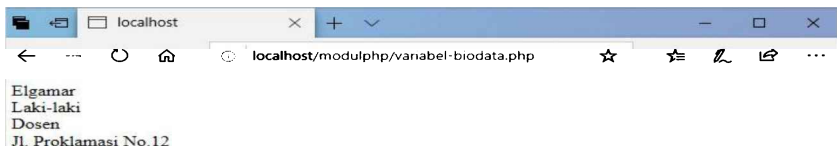
Fungsi variabel dapat menyimpan nilai atau informasi ke dalam memori. Variabel dapat kita gunakan berkali-kali sesuai kebutuhan dan nilainya dapat kita ganti. Nama variable diawali dengan tanda \$ (string). Misalnya \$nama, \$salamat.

Nama variable bersifat *case sensitive*, \$A tidak sama dengan \$a. Jenis variable bisa berupa boolean (true atau false) \$ (string) (huruf, angka dan karakter), integer (angka bulat), float atau double (angka desimal/pecahan), array, object, NULL. Variabel string bisa diapit tanda petik dua, atau petik satu. Buatlah contoh variabel, dan simpan dengan nama file **variabel-biodata.php**



```
1 <?php
2 $nama="Elgamar";
3 $jenis_kelamin="Laki-laki";
4 $pekerjaan="Dosen";
5 $salamat="Jl. Proklamasi No.12";
6
7 echo "$nama <br>";
8 echo "$jenis_kelamin <br>";
9 echo "$pekerjaan <br>";
10 echo "$salamat <br>";
11 ?>
```

Gambar 4.3 Penggunaan Variabel \$ (string)



localhost

localhost/modulphp/variabel-biodata.php

Elgamar
Laki-laki
Dosen
Jl. Proklamasi No.12

Gambar 4.4 Tampilan Penggunaan Variabel

Contoh variabel dengan angka:



```
1 <?php
2 $a="7";
3 $b="9";
4
5 echo $a+$b; //hasil
6 ?>
```

Gambar 4.5 Penggunaan Variabel dengan Angka

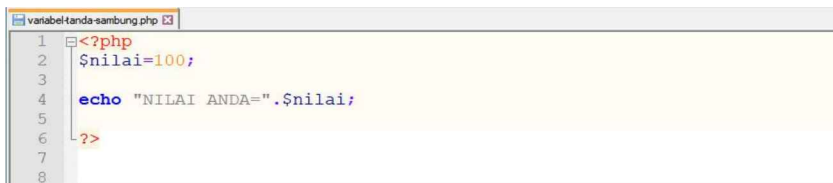
Penggunaan fungsi echo untuk menampilkan hasil perhitungan operator matematika, maka echo tidak perlu menggunakan tanda "" (kutip dua). Berikut hasilnya:



Gambar 4.6 Penggunaan Variabel dengan Angka

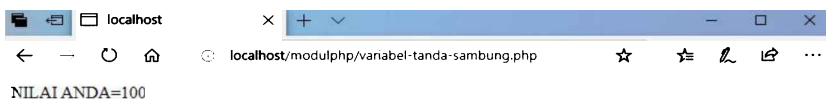
C. Tanda Sambung

Untuk menyambung antara variabel yang satu dengan yang lain, atau menggabungkan teks gunakan tanda "titik" buat file dengan nama **variabel-tanda-sambung.php**. Perhatikan contoh tanda sambung:



```
1 <?php
2 $nilai=100;
3
4 echo "NILAI ANDA=".$nilai;
5
6 ?>
```

Gambar 4.7 Penggunaan Tanda Sambung



Gambar 4.8 Hasil Penggunaan Tanda Sambung titik (.)

D. Operator Matematika

Penggunaan operator matematika diperlukan saat anda melakukan perhitungan dengan menggunakan angka, buat file dengan nama **var-operator-matematika.php** perhatikan pada tabel berikut:

Tabel 4.1. Jenis-jenis operator Matematika

Operator	Keterangan	Contoh	Hasil
+	Penambahan	x=2 jika: x+2	4
-	Pengurangan	x=2 Jika: 8-x	6
*	Perkalian	x=2 Jika: x*7	14
/	Pembagian	x=6 Jika: 12/x	2
%	Modulus (sis a pembagian)	5%2 10%8 10%2	1 2 0
++	Increment (Penambahan dengan 1)	x=5 x++	x=6 dst
--	Decrement (pengurangan dengan 1)	x=5 x--	x=4 dst



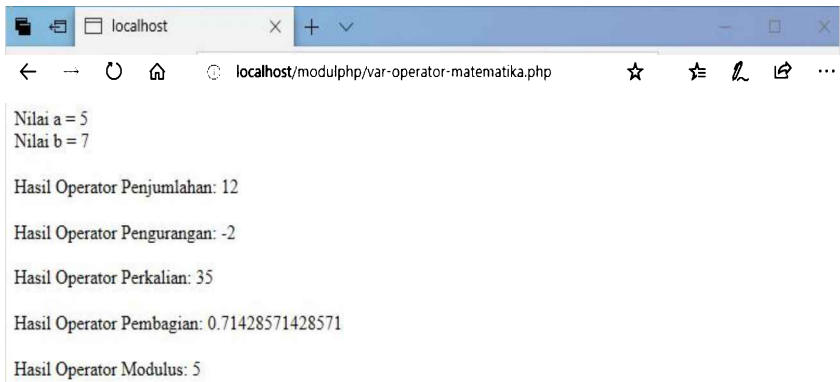
```

1 <?php
2 $nilai_a=5;
3 $nilai_b=7;
4
5 echo "Nilai a = $nilai_a <br>";
6 echo "Nilai b = $nilai_b <br><br>";
7
8 echo "Hasil Operator Penjumlahan: ";
9 echo $nilai_a+$nilai_b;
10 echo "<br><br>";
11
12 echo "Hasil Operator Pengurangan: ";
13 echo $nilai_a-$nilai_b;
14 echo "<br><br>";
15
16 echo "Hasil Operator Perkalian: ";
17 echo $nilai_a*$nilai_b;
18 echo "<br><br>";
19
20 echo "Hasil Operator Pembagian: ";
21 echo $nilai_a/$nilai_b;
22 echo "<br><br>";
23
24 echo "Hasil Operator Modulus: ";
25 echo $nilai_a%$nilai_b;
26 ?>

```

Gambar 4.9 Penggunaan Operator Matematika

Pada contoh diatas sangat jelas bahwa anda harus memberikan value terlebih dahulu pada variabel **nilai_a** dan **nilai_b**, setelah itu baru diberikan fungsi operator matematika “penambahan (+), pengurangan (-), perkalian (*), pembagian (/), modulus/sisa bagi (%). Lihat hasil penggunaan fungsi operator matematika, pada gambar berikut:



Gambar 4.10 Hasil Penggunaan Operator Matematika

Dari jenis-jenis operator tersebut, anda dapat mencoba dengan berbagai bentuk latihan lainnya, agar anda dapat lebih memahami fungsi dari operator matematika.

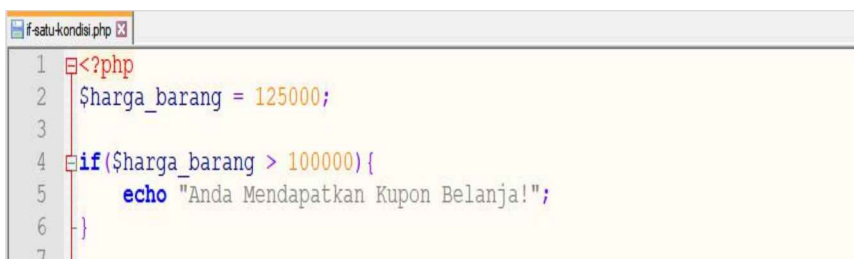
Dasar Penulisan PHP Fungsi *If*, *Switch Case*, Operator Perbandingan

A. Fungsi IF

Fungsi if menyatakan suatu kondisi, dimana dapat disesuaikan dengan kondisi yang diinginkan. Misal kita dapat menentukan alur program untuk mengisi password dengan benar dan alur program untuk mengisi password yang salah.

1. Penulisan fungsi if dengan 1 kondisi

Penggunaan fungsi if dengan satu kondisi, maka kode akan dieksekusi dan menampilkan sesuai dengan kondisi yang ditetapkan, tapi apabila tidak sesuai maka tidak akan menampilkan kondisi apapun. Silahkan buat nama file **if-satu-kondisi.php**

A screenshot of a code editor window titled 'if-satu-kondisi.php'. The code is written in PHP and uses an if statement to check if a variable is greater than a certain value. The code is as follows:

```
1 <?php
2 $harga_barang = 125000;
3
4 if($harga_barang > 100000){
5     echo "Anda Mendapatkan Kupon Belanja!";
6 }
7
```

Gambar 5.1 Penggunaan fungsi if satu kondisi

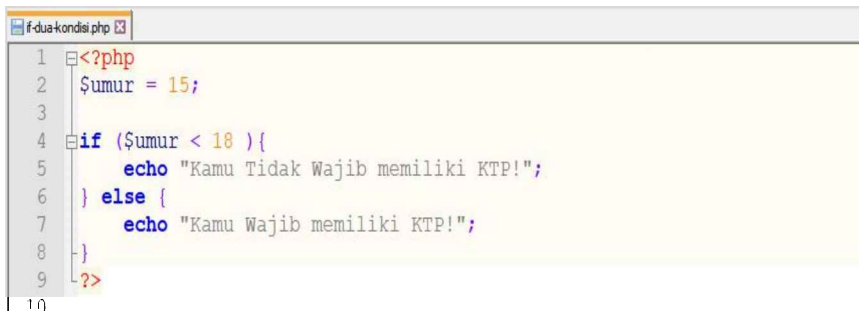
Silahkan kamu coba dengan value `$harga_barang` yang berbeda, amati jika ada perubahan dan seperti apa hasilnya. Hasil penulisan fungsi if satu kondisi dapat di lihat pada gambar berikut:



Gambar 5.2 Hasil Penggunaan fungsi if dengan satu kondisi

2. Penulisan fungsi if dengan 2 kondisi

Fungsi if dengan dua kondisi, memberikan pilihan dengan dua ketentuan, dimana jika kondisi pertama tidak benar (*false*), maka akan menampilkan kondisi ke dua. Contoh code, dapat anda lihat pada gambar berikut, buat dengan nama file **if-dua-kondisi.php**



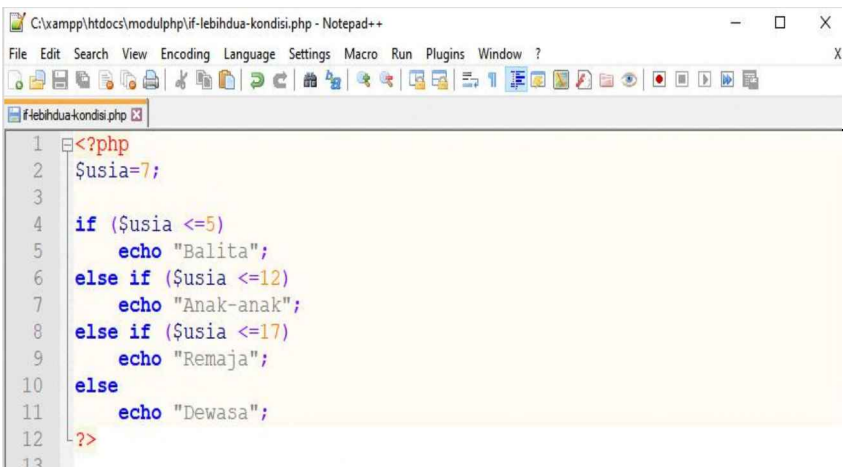
Gambar 5.3 Penggunaan fungsi if dengan dua kondisi



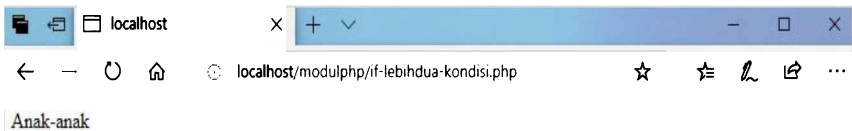
Gambar 5.4 Hasil Penggunaan fungsi if dengan dua kondisi

3. Penulisan fungsi if dengan lebih dari 2 kondisi

Dengan menggunakan fungsi if lebih dari dua kondisi, dapat memungkinkan anda untuk memberikan banyak pilihan terhadap kondisi yang anda inginkan, buat dengan nama file **if-lebihdua-kondisi.php**, lihat pada contoh berikut ini:



Gambar 5.5 Penggunaan fungsi if dengan dua kondisi



Gambar 5.6 Penggunaan fungsi if dengan lebih dari dua kondisi

B. Switch Case

Switch case merupakan bentuk lain dari fungsi *if else* dan memiliki kegunaan yang sama, yaitu memeriksa kondisi suatu variabel. Switch case berguna dalam pengambilan tindakan atau keputusan sesuai dengan kondisi yang telah ditetapkan. Lihat contoh berikut, nama file **switch-case.php**

```
1 <?php
2
3 $level = 'S1'; //statment yang dirubah
4 switch($level)
5 {
6     case 'SMA':
7         echo "Anda Tamatan SMA/SMK Sederajat";
8         break;
9     case 'S1':
10        echo "Anda Tamatan Sarjana (Strata-1)";
11        break;
12    case 'S2':
13        echo "Anda Tamatan Magister (Strata-2)";
14        break;
15    case 'S3':
16        echo "Anda Tamatan Doktorat (Strata-3)";
17        break;
18    default:
19        echo "Belum Bersekolah";
20 }
21 ?>
22
```

Gambar 5.7 Penggunaan fungsi switch case



Gambar 5.8 Hasil Penggunaan fungsi *switch case*

C. Operator Perbandingan

Operator perbandingan memiliki fungsi yang berkaitan dengan IF (kondisi), banyak sekali contoh kasus yang menjadikan operator perbandingan sebagai kondisi tertentu. Lihat jenis-jenis operator perbandingan pada tabel berikut:

Tabel 5.1. Jenis-jenis Operator Perbandingan

Operator	Keterangan	Contoh
==	Sama dengan	4==8; hasil false
!=	Tidak sama dengan	4!=8; hasil true
>	Lebih Besar	4>8; hasil false
<	Lebih Kecil	4<8; hasil true
>=	Lebih Besar sama dengan	4>=8; hasil false
<=	Lebih Kecil sama dengan	4<=8; hasil true

Dasar Penulisan PHP Dengan Fungsi Perulangan (*Looping*)

A. Fungsi Perulangan (*Looping*)

Perulangan (*looping*) merupakan barisan instruksi program yang bertujuan untuk mengulang beberapa baris perintah sesuai dengan kondisi yang diberikan. Perancangan barisan instruksi perulangan harus memenuhi 3 kondisi, yaitu kondisi awal dari perulangan, perintah program yang akan diulang, serta kondisi akhir dimana perulangan akan berhenti. Fungsi perulangan pada bahasa pemrograman PHP di bagi menjadi menjadi 4 jenis, lihat pada tabel berikut:

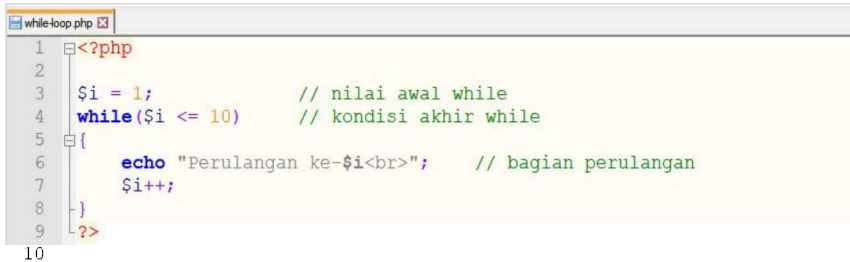
Tabel 6.1. Jenis-jenis fungsi perulangan

Jenis Perulangan	Keterangan
<i>While</i>	<i>uncounted loop</i>
<i>Do-While</i>	<i>uncounted loop</i>
<i>For</i>	<i>counted loop</i>
<i>For each</i>	<i>counted loop</i>

Counted loop merupakan jenis perulangan yang jumlah pengulangannya telah diketahui. Sedangkan *Uncounted loop* jenis perulangan yang tidak pasti berapa kali perintah akan mengulangi.

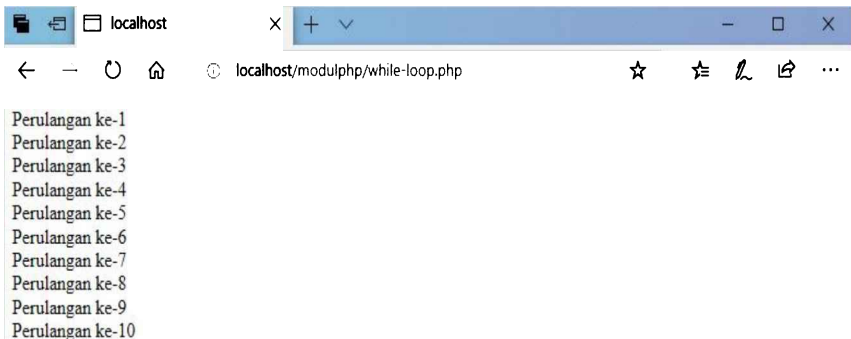
1. Perulangan *while*

Buatlah contoh perulangan *while* seperti gambar berikut, nama file **while-loop.php**



```
1 <?php
2
3 $i = 1;           // nilai awal while
4 while($i <= 10)   // kondisi akhir while
5 {
6     echo "Perulangan ke-$i<br>"; // bagian perulangan
7     $i++;
8 }
9 ?>
```

Gambar 6.1 Perulangan *while*



localhost

localhost/modulphp/while-loop.php

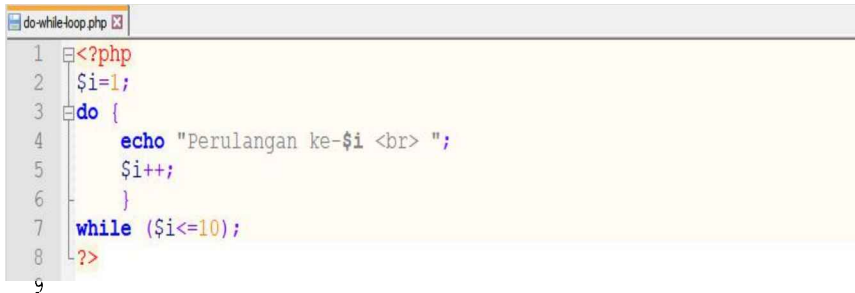
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
Perulangan ke-6
Perulangan ke-7
Perulangan ke-8
Perulangan ke-9
Perulangan ke-10

Gambar 6.2. Hasil Perulangan *while*

2. Perulangan *Do-While*

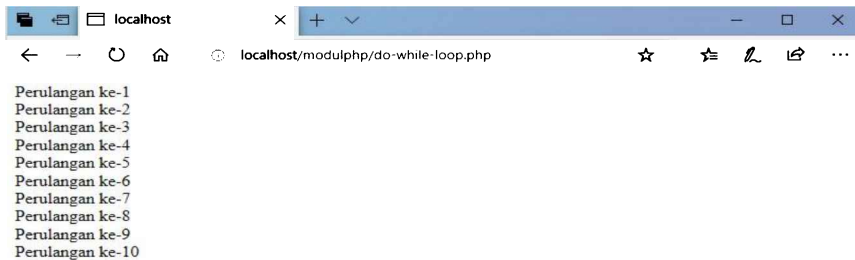
Perulangan dengan perintah *Do-While* merupakan kebalikan dari struktur perulangan *While*. Pada struktur *While*, pembacaan

ekspresi dilakukan pada awal perulangan, sedangkan pada struktur *Do-While* pembacaan ekspresi dilakukan setelah proses satu *loop* dilakukan. Buatlah sebuah file dengan **do-while-loop.php**.



```
1 <?php
2 $i=1;
3 do {
4     echo "Perulangan ke-$i <br> ";
5     $i++;
6 }
7 while ($i<=10);
8 ?>
```

Gambar 6.3 Perulangan *do-while*



localhost

localhost/modulphp/do-while-loop.php

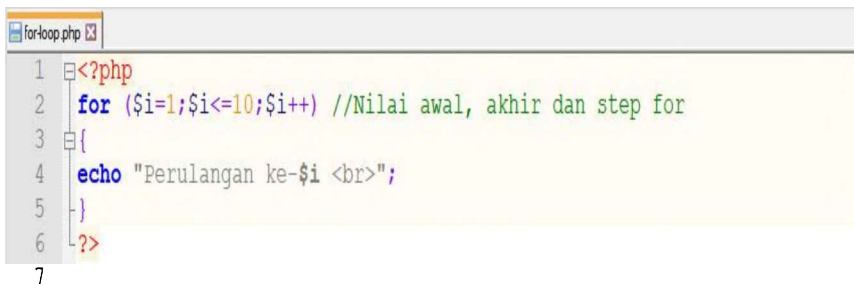
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
Perulangan ke-6
Perulangan ke-7
Perulangan ke-8
Perulangan ke-9
Perulangan ke-10

Gambar 6.4 Hasil Perulangan *do-while*

Perbedaan perintah *while* dan *do-while* terletak pada kondisi yang diperiksa. Pada perintah *while*, kondisi yang diperiksa terletak di awal perulangan. Sedangkan pada perintah *do-while*, diperiksa di akhir perulangan.

3. Perulangan *for*

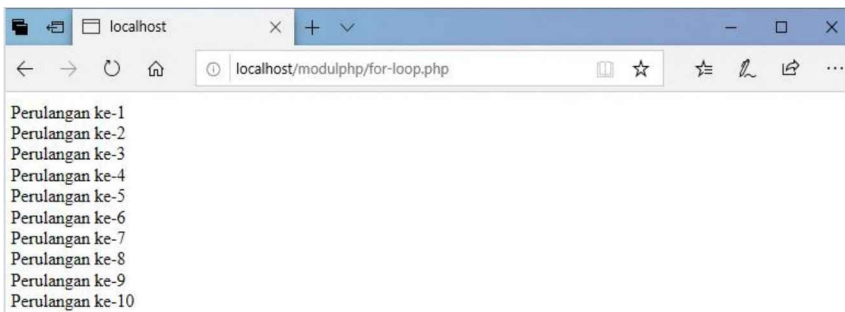
Perulangan *for* ditentukan oleh jumlah perulangan dan kelipatan perulangannya. Kita hanya perlu menuliskan nilai awal perulangan dan nilai batas perulangannya. Nilai variabel akan otomatis bertambah atau berkurang setiap kali proses perulangan dilakukan. Untuk lebih jelasnya buatlah sebuah file dengan nama **for-loop.php** dan perhatikan dengan seksama.



```
1 <?php
2 for ($i=1;$i<=10;$i++) //Nilai awal, akhir dan step for
3 {
4     echo "Perulangan ke-$i <br>";
5 }
6 ?>
```

7

Gambar 6.5 Perulangan *for*



localhost

localhost/modulphp/for-loop.php

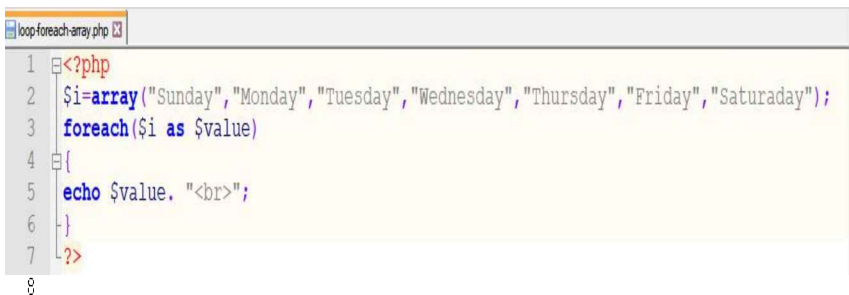
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
Perulangan ke-6
Perulangan ke-7
Perulangan ke-8
Perulangan ke-9
Perulangan ke-10

Gambar 6.6 Hasil Perulangan *for*

B. Perulangan *for* dan *foreach* dengan *Array*

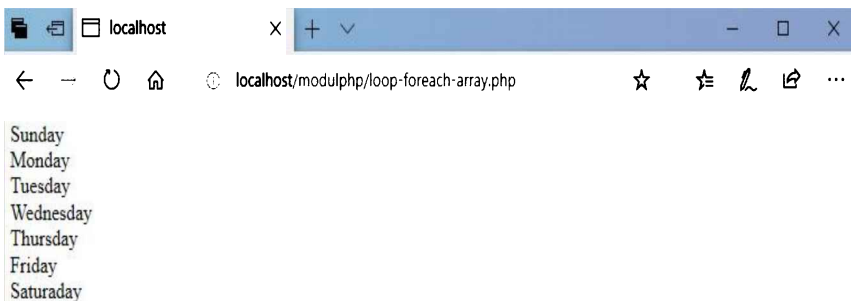
Perulangan *foreach* menggunakan *array*. *Array* merupakan tipe data yang sering digunakan dalam bahasa pemrograman PHP. *Array* mampu menyimpan banyak data dalam satu variabel dan akan sangat berguna untuk menyederhanakan dan menghemat penggunaan *variabel*. Proses menampilkan data *array*, kita bisa memanfaatkan perulangan *foreach* dan *for*, seperti contoh berikut ini:

- Buatlah sebuah file dengan nama **loop-foreach-array.php** seperti gambar berikut:



```
1 <?php
2 $i=array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday");
3 foreach($i as $value)
4 {
5     echo $value, "<br>";
6 }
7 ?>
```

Gambar 6.7 Perulangan *array-foreach*



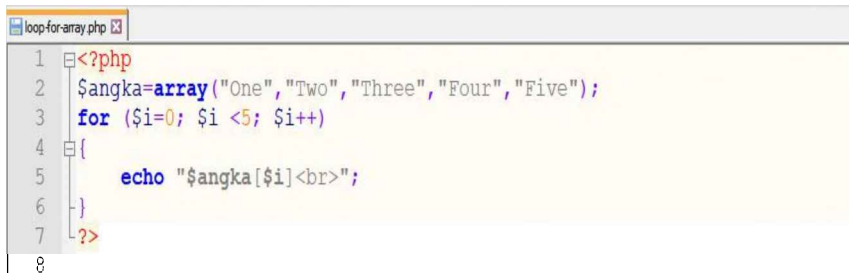
localhost

localhost/modulphp/loop-foreach-array.php

Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

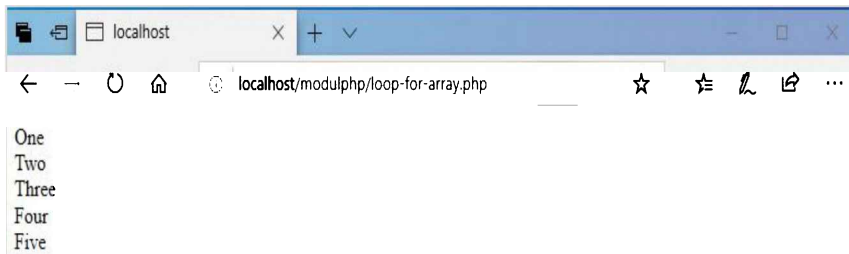
Gambar 6.8 Hasil Perulangan *array-foreach*

- Buatlah perulangan *array* dengan jenis *for*, dengan nama **loop-for-array.php** seperti gambar berikut:



```
1 <?php
2 $angka=array("One","Two","Three","Four","Five");
3 for ($i=0; $i <5; $i++)
4 {
5     echo "$angka[$i]<br>";
6 }
7 ?>
8
```

Gambar 6.9 Hasil Perulangan *array-for*



localhost

localhost/modulphp/loop-for-array.php

One
Two
Three
Four
Five

Gambar 6.10 Hasil Perulangan *array-for*

Dasar Penulisan PHP Dengan Form Input

A. PHP dengan Form

Sebuah aplikasi tidak terlepas dari pembuatan sebuah form input. Form input tersebut juga digunakan sebagai interaksi antara user dengan sistem. Pada PHP terdapat berbagai contoh form input/masukan antara lain dapat dilihat pada tabel berikut:

Tabel 7.1. Jenis-jenis form input/masukan

Jenis Form	Fungsi
1. Submit and Reset Button	Berfungsi sebagai <i>button</i> eksekusi pada <i>form input</i>
2. Text Field	Berfungsi sebagai <i>form input short text</i>
3. Text Area	Berfungsi sebagai <i>form input long text</i>
4. List Menu	Berfungsi sebagai <i>form input optional</i>
5. Radio Button	Berfungsi sebagai <i>form input optional</i>
6. Check Box	Berfungsi sebagai <i>form input optional ganda</i>

B. Form dan Implementasi

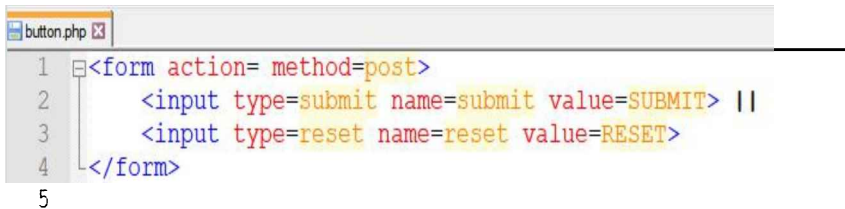
1. *Submit dan Reset Button*

Submit dan *Reset* merupakan sebuah *button* atau tombol eksekusi sebuah perintah pada form input. Aplikasi berbasis web maupun aplikasi berbasis desktop menggunakan tombol ini sebagai eksekusi suatu perintah. Button ini dapat kita lihat pada contoh berikut :



Gambar 7.1 Contoh *Button Submit* dan *Reset*

Langkah selanjutnya bagaimana cara kita membuat tombol di atas, lihat pada gambar berikut, buatlah sebuah file dengan nama **button.php**

The image is a screenshot of a code editor window titled 'button.php'. It displays the following HTML code:

```
1 <form action= method=post>
2   <input type=submit name=submit value=SUBMIT> ||
3   <input type=reset name=reset value=RESET>
4 </form>
5
```

The code is color-coded: tags are blue, attributes and values are orange, and the text '||' is black.


Gambar 7.2 Contoh *Button Submit* dan *Reset*

Pada code di atas, terlihat penggunaan fungsi button tidak memerlukan tag `<?php ?>`. Kemudian bagian terpenting dari pembuatan *form input* adalah tag pembuka `<form>` dan tag penutup `</form>`. Pada tag pembuka perlu disertakan atribut *method* dan *action*, disini kita menggunakan atribut *method* “post” dan atribut *action* masih kosong, karena kita tidak mengarahkannya pada file manapun juga. Button Submit dan Reset akan dikombinasikan dengan jenis *form input* yang ada.

2. Text Field

Text field banyak digunakan sebagai form input/masukan berupa, **nama, nim, nik, nip, no_handphone, email** dan sebagainya. Seperti contoh dibawah ini:

Buatlah sebuah file pada notepad++ dengan nama "**text-field.php**", kemudian ketiklah code dibawah ini, kemudian lihatlah hasilnya pada browser.

The image shows a screenshot of a code editor with a file named 'text-field.php'. The code is as follows:

```
1 <form action=text-field.php method=post>
2
3 NPM: <input type=text name=npm><br><br>
4 Nama: <input type=text name=nama><br><br>
5 Email: <input type=text name=email><br><br>
6 <input type=submit name=submit value=SUBMIT> ||
7 <input type=reset name=reset value=RESET>
8
9 </form>
10
11 <?php
12 if(isset($_POST['submit'])) {
13     echo "NPM: $_POST[npm] <br>";
14     echo "Nama: $_POST[nama] <br>";
15     echo "Email: $_POST[email]";
16 }
17 ?>
18
```

Gambar 7.3 Form Input dengan *textfield*

Code `<form action=text-field.php method=post>`, yaitu mengarahkan sebuah proses perintah kepada suatu halaman, dan pada contoh ini halaman yang dituju adalah halaman itu sendiri. Fungsi `$_POST[npm]`, `$_POST[nama]`, `$_POST[email]` untuk mengambil nilai dari data yang dimasukkan pada form input, berikut hasilnya.

localhost/modulphp/text-field.php

NPM:

Nama:

Email:

NPM: 1022108702
 Nama: Elgamar
 Email: elgamar.mkom@gmail.com

Gambar 7.4 Hasil *Form Input* dengan *textfield*

3. Text Area

Penggunaan form input *textarea* biasanya digunakan untuk memasukan alamat, isi berita, komentar, keterangan atau catatan, yang memiliki jumlah karakter input yang panjang. Buatlah sebuah file dengan nama “**textarea.php**”, kemudian ketiklah code dibawah ini, kemudian lihatlah hasilnya pada browser.

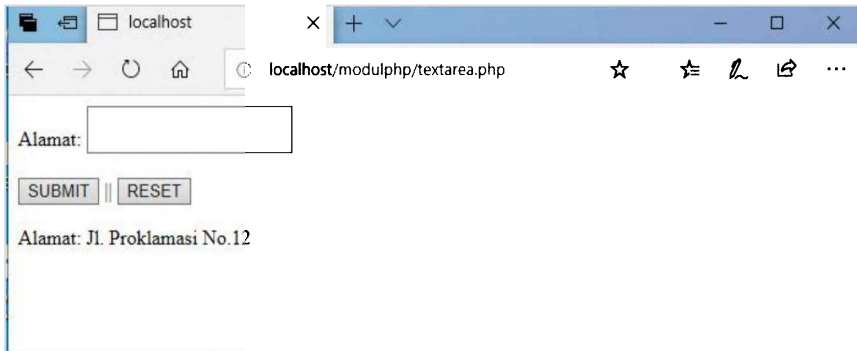
```

1 <form action=textarea.php method=post>
2  Alamat: <textarea name=alamat></textarea><br><br>
3  <input type=submit name=submit value=SUBMIT> ||
4  <input type=reset name=reset value=RESET>
5 </form>
6
7 <?php
8 if(isset($_POST['submit'])){
9     echo "Alamat: $_POST[alamat] <br>";
10 }
11 ?>

```

Gambar 7.5 *Form Input* dengan *textarea*


`<textarea nama="alamat"></textarea>`, adalah script pembentuk form input textarea, berikut hasilnya.

A screenshot of a web browser window. The address bar shows 'localhost/modulphp/textarea.php'. The page content includes a label 'Alamat:' followed by a text area input field. Below the input field are two buttons: 'SUBMIT' and 'RESET'. Below the buttons, the text 'Alamat: Jl. Proklamasi No.12' is displayed, indicating the value entered in the text area.

Gambar 7.6 Hasil *Form Input* dengan *textarea*

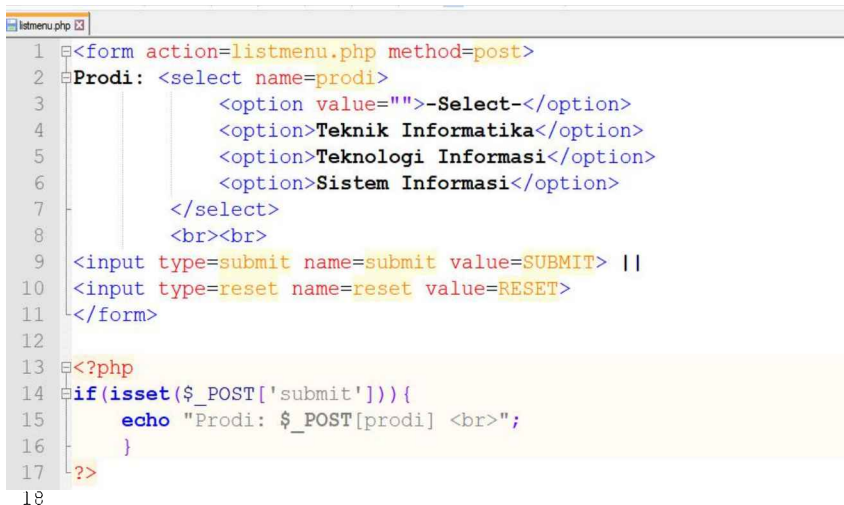
4. List Menu

Form input list menu merupakan *form input/masukan* yang berguna untuk memilih salah satu pilihan yang disediakan, berbentuk list drop down. Umumnya list menu digunakan pada contoh seperti: pemilihan jurusan, provinsi, kabupaten, agama, kategori dan sebagainya. Lihatlah contoh berikut ini:

A screenshot of a web browser window. The address bar shows 'localhost/modulphp/listmenu.php'. The page content includes a label 'Prodi:' followed by a dropdown menu (list menu) with the text '-Select-' and a downward arrow. Below the dropdown menu are two buttons: 'SUBMIT' and 'RESET'.

Gambar 7.7 Form input dengan *listmenu*

Pada contoh, kita dapat menambahkan beberapa item lagi, sesuai dengan kebutuhan. Form input *list menu* ini disediakan untuk pemilihan salah satu item dengan beberapa pilhan. Berikut ini contoh code pembuatan *form input list meni*, buatlah dengan nama file **listmenu.php**



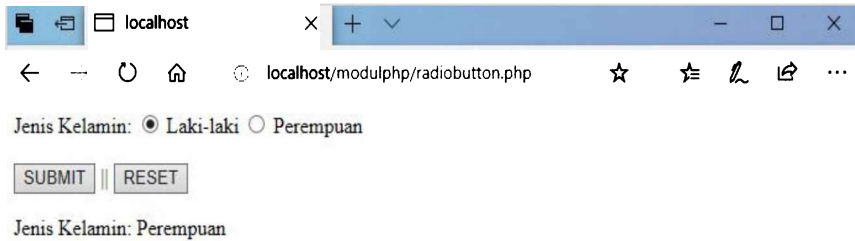
```
1 <form action=listmenu.php method=post>
2 Prodi: <select name=prodi>
3     <option value="">-Select-</option>
4     <option>Teknik Informatika</option>
5     <option>Teknologi Informasi</option>
6     <option>Sistem Informasi</option>
7 </select>
8 <br><br>
9 <input type=submit name=submit value=SUBMIT> ||
10 <input type=reset name=reset value=RESET>
11 </form>
12
13 <?php
14 if(isset($_POST['submit'])){
15     echo "Prodi: $_POST[prodi] <br>";
16 }
17 >
18
```

Gambar 7.8 Form Input dengan *list menu*

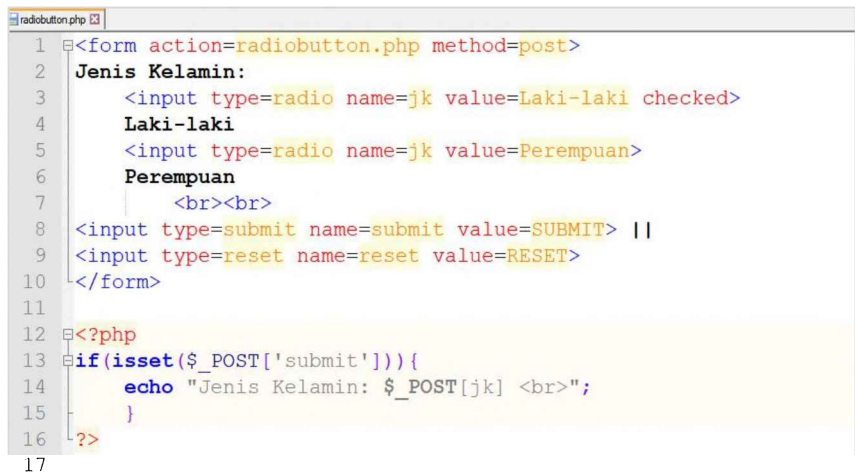
`<select name=prodi></select> <option>Item1</option>`, adalah struktur code pembentuk *form input list menu*.

5. Radio Button

Radio button merupakan *form input* yang berfungsi sebagai *option* atau pilihan. Umumnya radio *button* dibuat untuk memilih salah satu pilihan yang bersifat terbatas seperti: Jenis kelamin atau kondisi (*true or false*) / (*yes or no*) / (aktif or non aktif).



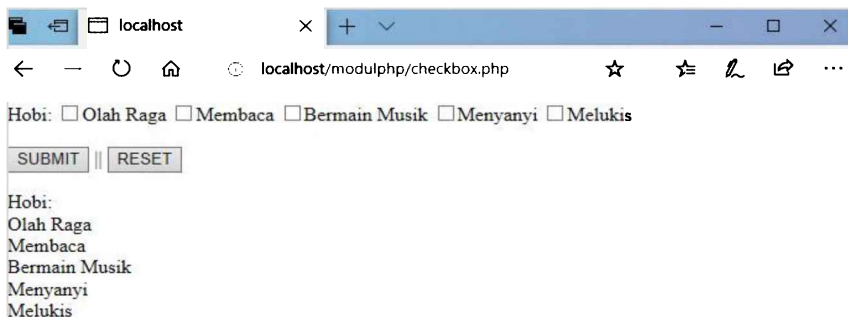
Gambar 7.9 Form Input Radio Button



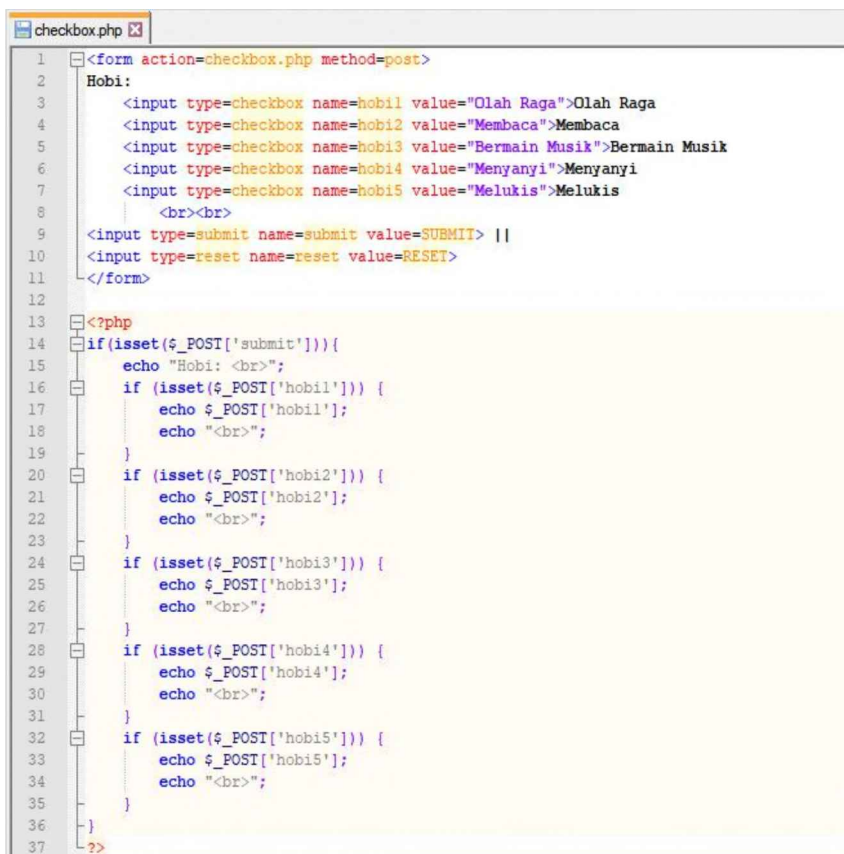
Gambar 7.10 Hasil Form Input dengan Radio Button

6. Check Box

Fungsi *Check box* sebagai *form input* yang bersifat multi, sehingga *user* dapat memberikan lebih dari satu pilihan. Adapun contoh *check box* seperti, “hobi”, “menu” dan sebagainya.



Gambar 7.11 Form input *checkbox*



Gambar 7.12. Form Input dengan *Checkbox*

`<input type="checkbox" name="hobi1" value="Olah raga">`, adalah code yang membentuk struktur dari *checkbox*, pada masing-masing pilihan, "`name`" dari sebuah *checkbox* dibedakan menjadi `hobi1`, `hobi2`, `hobi3`, `hobi4` dan `hobi5`. Form input optional *checkbox* sedikit berbeda dengan jenis form input optional lainnya seperti, *listmenu*, dan *radio button*.

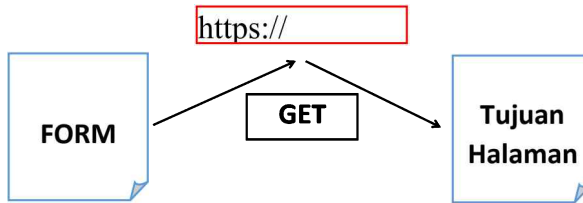
Dasar Penulisan PHP Dengan Method POST dan GET

A. Method POST dan GET

Method merupakan suatu cara yang dapat digunakan untuk mengirimkan data atau suatu nilai ke halaman lain untuk diproses. Terdapat dua Method dalam bahasa pemrograman PHP yang dapat digunakan dalam memproses sebuah form, yaitu method POST dan GET. Pada dasarnya, method POST dan GET memiliki kegunaan yang sama yaitu untuk mengirimkan value variabel ke halaman lain atau mengirimkan ke database. Perbedaan kedua method tersebut sangat kecil namun sangat terlihat dengan jelas. Dalam mengirimkan data Method GET akan menambahkan pada URL, sedangkan method POST dikirim ke server dengan disertakankan pada sebuah permintaan (*body of request*) dan bukan melalui URL dan hasil dari method POST ini tidak bersifat *cacheable* (dapat disimpan dalam *cache*), perhatikan perbedaan dari gambar berikut:



Gambar 8.1 Pemrosesan Method POST



Gambar 8.2 Pemrosesan Method GET

Silahkan lihat contoh berikut bagaimana method POST dan GET bekerja dengan menggunakan form login sederhana. Buatlah dua buah file dengan **form-login.php** dan **proses.php** dan tambahkan script java script.

```

1  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
2  <script>
3  $(document).ready(function() {
4
5      $("#input[type=radio]").change(function() {
6          var method = $(this).val();
7          $("#form").attr("method", method);
8      });
9
10     $("#submit").click(function() {
11         var fuser = $("#fuser").val();
12         var lpass = $("#lpass").val();
13
14         if( fuser != '' || lpass != '' ) {
15             return true;
16         }
17         else {
18             alert("Silahkan isi form");
19             return false;
20         }
21     });
22 });
23 </script>
24
25 <form method="get" action="form-login.php" id="form">
26     <h2>POST VS GET</h2><hr>
27     Username:
28     <input type="text" name="fuser" id="fuser" />
29     Password:
30     <input type="password" name="lpass" id="lpass" />
31     Pilih Method :
32     <span><input type="radio" name="method" value="get" checked> GET &nbsp;&nbsp;& #
33     <input type="radio" name="method" value="post" > POST </span>
34     <input type="submit" name="submit" id="submit" value="Submit">
35 </form>
36 <?php include "proses.php";?>
37

```

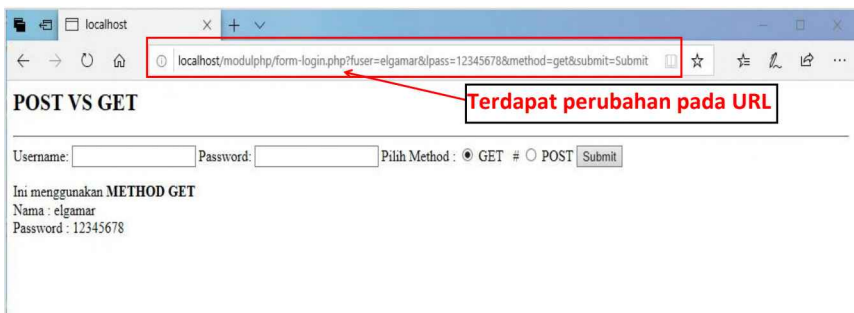
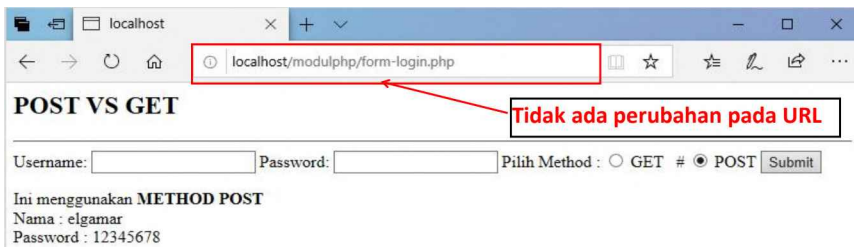
Gambar 8.3 Code script form-login.php

```

1 <?php
2 if(isset($_POST['fuser']))
3 {
4     $fuser = $_POST['fuser'];
5     $lpass = $_POST['lpass'];
6     echo "<span class='success'>Ini menggunakan <b>METHOD POST</b></span><br/>";
7     echo "Nama : ".$fuser."<br/>Password : ".$lpass;
8 }
9 //-----
10 if(isset($_GET['fuser']))
11 {
12     $fuser = $_GET['fuser'];
13     $lpass = $_GET['lpass'];
14     echo "<span class='success'>Ini menggunakan <b>METHOD GET</b></span><br/>";
15     echo "Nama : ".$fuser."<br/>Password : ".$lpass;
16 }
17 ?>

```

Gambar 8.4 Code script proses.php



Gambar 8.5 Hasil Perbedaan Method POST dan Method GET

Tabel 8.1. Perbedaan Method POST dan Method GET

Perbedaan	Method POST	Method GET
History	Parameter/nilai tidak tersimpan di browser.	Parameter/nilai tersimpan di browser dan bisa diakses ulang karena nilai menjadi kesatuan URL.
Keamanan	Method POST lebih aman karena data langsung dikirim ke server.	Method GET dapat diketahui data apa saja yang dikirim.
Penggunaan	Method POST digunakan untuk mengirim data yang lebih rahasia seperti password.	Digunakan lebih untuk keperluan mengambil data.
Besar data yang dikirim	Standard 8 Mb, tergantung konfigurasi server.	7607 karakter.

B. Kesimpulan POST dan GET

Adapun kesimpulan yang dapat diambil dari perbedaan Method POST dan Method GET adalah :

1. Method POST tidak menampilkan nilai *variabel* pada URL dan Method GET menampilkan nilai *variabel* yang dikirimkan.
2. Method POST lebih aman dan Method GET kurang aman (pada contoh diatas berarti password ditampilkan pada URL).
3. Jika Method GET dibatasi panjang string hingga 2047 karakter, Method POST tidak.
4. Perbedaan pengambilan data jika Method POST menggunakan \$_POST sedangkan Method GET menggunakan \$_GET.

5. Method POST biasanya digunakan untuk input dari FORM, Method GET menggunakan input dari LINK atau akses menggunakan link.
6. Method POST digunakan untuk mengirimkan data rahasia seperti password, Method GET digunakan untuk mengirimkan/mengambil data publik seperti id_user atau id_halaman.

LAMPIRAN

Link download soft file latihan via google drive:

<https://bit.ly/BukuAjarWebJilid1>

DAFTAR PUSTAKA

- Kadir, A. (2017). *Dasar Logika Pemrograman Komputer*. Elex Media Computindo.
- Marisa, F. (2017). *Web Programming (Client Side and Server Side)*. Deepublish.
- Prasetio, A. (2012). *Buku Pintar Pemrograman Web*. Jakarta: mediakita.
- Raharjo & Istiyanto, J. E. (2003). *Keamanan Akses ke PostgreSQL melalui PHP (Menggunakan Apache Web Server pada GNU/Linux)*. Penerbit Andi.
- Solichin, A. (2016). *Pemrograman web dengan PHP dan MySQL*. Penerbit Budi Luhur.
- Zaki, A., & Comunity, S. (2008). *36 menit belajar komputer PHP dan MySQL*. Jakarta: PT Elex Media Komputindo.

PROFIL PENULIS



Elgamar, S.Kom., M.Kom merupakan salah satu Dosen pengajar pada Program Studi Teknik Informatika di Universitas Islam Kuantan Singingi (UNIKS) Telukkuantan - Riau. Latar belakang pendidikan penulis adalah sebagai dosen dibidang ilmu komputer (*computer science*). Penulis telah merintis karirnya di bidang pendidikan sebagai dosen sejak tahun 2012, dan telah banyak mengajar diberbagai kampus di Riau. Penulis telah

menjadi dosen tetap di Universitas Islam Kuantan Singingi sejak tahun 2014 awal sampai sekarang. Sekarang penulis sedang melanjutkan pendidikan S3 Doktorat (Ph.D) di Kampus Universiti Utara Malaysia (UUM) Kedah Darul Aman, Malaysia sejak 2019 akhir. Penulis berharap buku ajar **Konsep Dasar Pemrograman Website Dengan PHP** (Jilid 1) ini menjadi langkah awal penulis untuk fokus menghasilkan karya-karya berupa buku ajar, buku referensi dan buku monograf dimasa mendatang.

Adapun buku ajar ini berisi tentang:

1. Memahami Dasar-dasar Web dan Perangkat Lunak Pendukungnya.
2. Memahami Dasar Pembangunan Web dan HTML.
3. Memahami Fungsi Tag HTML.
4. Memahami Dasar Pemrograman PHP.
5. Memahami Konsep Basisdata dalam Pemrograman PHP
6. Memahami Konsep Pembangunan Web dengan Fungsi Form Input.
7. Memahami Konsep dan Fungsi Method POST dan GET.



Penerbit:

CV. Multimedia Edukasi

Jl. Ki Ageng Gribig, Gang Kaserin MU No. 36
Kota Malang 65138, Telp: +6285232777747
www.multidukasi.co.id

ISBN 978-523-7531-97-5

