

PHP7: BUKU BELAJAR UNTUK PEMULA DAN AWAM PEMROGRAMAN

PHP7: BUKU BELAJAR UNTUK PEMULA DAN AWAM PEMROGRA

Dalam 24 Jam



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.’
Imam Syafi’i*

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1	Judul Bagian Pertama	1
2	Judul Bagian Kedua	5
3	Dasar Penulisan PHP Sederhana	9
4	Array	23
5	Object Oriented Programming	31

DAFTAR ISI

Daftar Gambar	xiii
Daftar Tabel	xv
Foreword	xix
Kata Pengantar	xxi
Acknowledgments	xxiii
Acronyms	xxv
Glossary	xxvii
List of Symbols	xxix
Introduction	xxx
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Judul Bagian Pertama	1
1.1 Pendahuluan	1
1.2 Pengenalan	2
1.2.1 Kelebihan PHP	2
1.2.2 Penulisan Skript	3
	ix

1.2.3	PHP 7	3
1.3	CGI, FastCGI dan Modul Server Web	3
1.4	PHP7	4
1.5	Framework PHP	4
2	Judul Bagian Kedua	5
2.1	Pemasangan PHP di Server Web Windows	5
2.1.1	Cara Install XAMPP	6
2.2	Server web Eksternal untuk PHP	7
2.3	Pemasangan PHP untuk IIS pada windows 10	7
2.4	Localhost	7
3	Dasar Penulisan PHP Sederhana	9
3.1	Aturan Dasar Penulisan Kode PHP	9
3.1.1	Case Sensitivity (perbedaan huruf besar dan kecil) dalam PHP	9
3.1.2	Karakter Spasi dan Tab dalam PHP	10
3.2	Embedded Script dan Non Embedded	10
3.2.1	Embedded Script	10
3.2.2	Non Embedded	11
3.3	Variabel dan Tipe Data	12
3.3.1	Variabel	12
3.3.2	Tipe Data	12
3.3.3	Tipe Data Integer	12
3.3.4	Tipe Data Floting Point	13
3.3.5	Tipe Data String	14
3.3.6	Tipe Data Boolean	14
3.3.7	Tipe Data Objek	14
3.4	Operator PHP	15
3.4.1	Operator Perbandingan	15
3.4.2	Operator PHP Increment dan Decrement	15
3.4.3	Perbedaan Pre Increment dan Post Increment	15
3.4.4	Operator Assignment PHP	15
3.5	Lingkup Variabel	16
3.5.1	Variabel Global	16
3.5.2	Variabel Lokal	17
3.5.3	Variabel Statik	18
3.6	Struktur Kontrol	18

3.6.1	If Statement	18
3.6.2	Perulangan	19
3.6.3	Fungsi	19
3.6.4	Cookie dan Session	19
3.7	Koneksi Dengan Database MySQL	20
3.8	Perulangan	20
4	Array	23
4.1	Array	23
4.1.1	Array Berindeks	24
4.1.2	Array Asosiatif	24
4.2	Membuat Array	25
4.2.1	File Direktori	25
4.2.2	Pemrograman Berorientasi Objek Dalam PHP	26
4.2.3	Object Dan Class	26
4.2.4	Properties Dan Method	27
4.2.5	Mengakses Elemen Array	27
4.2.6	Fungsi-fungsi yang berhubungan dengan array	28
5	Object Oriented Programming	31
5.1	E-Logbook	31
5.1.1	Mengaktifkan MySQLi	32
5.1.2	Cara Install XAMPP	32
5.1.3	Menguji Instalasi XAMPP	36
5.1.4	Membuat Database	37
5.2	Struktur Dasar PHP	42
5.2.1	Variabel	43
5.2.2	Tipe Data	43
5.2.3	Konstanta	44
5.2.4	Struktur Kondisi Dan Perulangan	44
5.2.5	Penanganan Form	47
5.2.6	Array Dan Fungsi	48
5.2.7	Fungsi	49
5.2.8	String Dan Tanggal	51
5.2.9	File Dan Direktori	55

DAFTAR GAMBAR

3.1	Sistem Bilangan	13
4.1	Fungsi-fungsi yang berhubungan dengan array	29
5.1	Versi Terbaru XAMPP	33
5.2	User Account Control	33
5.3	Jendela Awal	33
5.4	Select Component	34
5.5	Installation Folder	34
5.6	Bitnami for XAMPP	35
5.7	Ready to Install	35
5.8	Proses Penginstallan	36
5.9	Windows Security Alert	36
5.10	Selesai Install	37
5.11	XAMPP Control Panel	37

5.12	Localhost Dashboard	38
5.13	Versi PHP	38
5.14	XAMPP Control Panel	39
5.15	Halaman Utama	39
5.16	Create Database	40

DAFTAR TABEL

4.1	(24
4.2	(24
5.1	Operasi Tanggal PHP	54

Listings

src/embedded_script.php	10
src/embedded_script.php	11
src/embedded_script.php	11
src/embedded_script.php	11
src/tag_awal_akhir.php	12
src/tipeDataObjek.php	14
src/lingkupVariabel.php	16
src/lingkupVariabel.php	17
src/lingkupVariabel.php	17
src/lingkupVariabel.php	18
src/array.php	28
src/akses_array.php	28

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[?].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

JUDUL BAGIAN PERTAMA

1.1 Pendahuluan

Situs web merupakan suatu layanan yang menyajikan informasi menggunakan konsep hyperlink, yang memudahkan pengguna dalam menelusuri atau mencari informasi dari internet untuk mendapatkan informasi, dengan cukup mengklik suatu link berupa teks atau gambar, maka dari teks atau gambar tersebut akan menampilkan informasi yang detail (rinci). Informasi yang akan disajikan dalam halaman web menggunakan konsep multimedia, informasi dapat disajikan dengan menggunakan banyak media (teks, gambar, animasi, suara (audio), dan film). Dalam suatu halaman web, informasi akan disajikan dalam kombinasi dari media-media tersebut yang disajikan dalam suatu halaman.

Situs web berupa kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. yang ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti Internet, ataupun jaringan wilayah lokal (LAN) melalui alamat Internet yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di Internet disebut pula sebagai World Wide Web atau lebih dikenal dengan singkatan WWW. Web merupakan hal yang sangat populer di lingkungan pengguna internet dalam mengakses dan mendapatkan informasi karena kemu-

dahan yang diberikan kepada pengguna internet untuk melakukan penelusuran, penjelajahan dan pencarian informasi.

1.2 Pengenalan

PHP (Hypertext Preprocessor) adalah bahasa pemrograman script server-side yang didesain untuk pembuatan atau pengembangan web. Dengan ini, PHP juga dapat digunakan sebagai bahasa pemrograman umum. PHP sendiri dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan pada akhirnya dikelola oleh The PHP Group. Situs resmi PHP beralamat <http://www.php.net>. PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Pada bulan Juni 1996, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. PHP juga banyak diaplikasikan untuk pembuatan program-program seperti sistem informasi klinik, rumah sakit, akademik, keuangan, manajemen aset, manajemen bengkel dan lain-lain. Dapat dikatakan bahwa program aplikasi yang dulunya hanya dapat dikerjakan untuk desktop aplikasi, PHP sudah dapat mengerjakannya. Penerapan PHP saat ini juga banyak ditemukan pada proyek-proyek pemerintah seperti e-budgetting, e-procurement, e-government dan e lainnya. Website Ubaya ini juga dibuat menggunakan PHP. PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C /VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Wordpress, Mambo, Joomla, Postnuke, Xaraya, dan lain-lain.

1.2.1 Kelebihan PHP

Dalam pembuatan program menggunakan PHP mempunyai kelebihan yaitu:

1. PHP dapat dijalankan diberbagai sistem operasi mulai Windows, Macintosh dan Linux.

2. PHP tidak perlu di compile
3. Mudah diinstall kedalam web server seperti apache sehingga konfigurasi menjadi mudah.
4. Pengembangan relatif mudah karena banyak yang membahas tentang PHP.

1.2.2 Penulisan Skript

```
1 Contoh :  
2 <?php  
3 echo "Hello Word"  
4 ?>
```

PHP ini dapat dijalankan melalui file HTML yang kemudian akan dipanggil melalui web browser seperti Google Chrome, Mozilla Firefox dan UC Browser. Program dalam PHP ditulis dengan ekstensi '.php'.

1.2.3 PHP 7

PHP 7 adalah rilis terbaru dari bahasa pemrograman PHP dan disebut sebagai revolusi dalam aplikasi web dapat dikembangkan dan dikirim untuk mobile. Rilis ini dianggap menjadi perubahan yang paling penting setelah perubahan dari PHP 5 pada tahun 2004. Dalam PHP 7 terdapat beberapa fitur baru yaitu:

1. Improved Performance, memiliki kecepatan dua kali lebih cepat dari PHP 5.
2. Lower Memory Consumption, pemakaian memori yang rendah.
3. Consistent 64-bit support, dukungan konsisten untuk 64-bit mesin arsitektur.

1.3 CGI, FastCGI dan Modul Server Web

CGI(Common Gateway Interface) adalah suatu standar yang menghubungkan (interface) aplikasi internal dengan server web.

FastCGI merupakan standar baru yang menambah dan meningkatkan kemampuan dari program CGI, dimana FastCGI merupakan standar terbuka yang banyak digunakan oleh server web komersial dan opensource.

CGI akan di eksekusi setiap kali program CGI dipanggil, setelah dieksekusi CGI tersebut langsung dihapus dari memori. FastCGI akan terus berada di dalam memori, sampai ada perintah yang secara eksplisit untuk menghentikan program tersebut, sehingga FastCGI akan lebih cepat untuk melayani suatu permintaan. Keunggulan dari FastCGI adalah FastCGI berada pada komputer yang berbeda dengan server webnya.

Modul server web merupakan standar yang mengintegrasikan aplikasi luar menjadi bagian dari server web. PHP merupakan salah satu program yang merupakan bagian dari server web. Tidak semua program CGI dibuatkan menjadi modul untuk server web, dan tidak semua server web menyediakan slot untuk ditambahi modul server web dari luar.

1.4 PHP7

PHP7 dirancang untuk memiliki fitur bahasa pemrograman yang lebih baik dari versi PHP yang sebelumnya dengan kecepatan yang mencapai 2 kali dari kecepatan PHP 5.6.x. PHP7 merupakan versi terbaru yang ditunggu oleh pengembang aplikasi web dengan menggunakan PHP, karena versi PHP sejak tahun 2004 berlabel versi 5 sebagai versi mayornya, dan juga sebagai versi yang terakhir. versi 6 yang ditunggu kehadirannya terpaksa batal untuk dikeluarkan karena masalah teknis dalam pengembangannya.

Berikut merupakan fitur dari PHP7 yang telah ditetapkan:

- Membuang konstruktor nama fungsi yang ada pada saat PHP4.
- Operator baru yaitu spaceship dan interface.
- Mengganti ekstensi JSON yang ada dengan JSOND.
- Deklarasi tipe hasil dari suatu fungsi (return type).

1.5 Framework PHP

Framework PHP adalah suatu kerangka kerja untuk membuat program menggunakan PHP, dengan menggunakan framework dapat memudahkan pengembang dalam membangun suatu web menggunakan PHP sehingga:

- Pengembang aplikasi tidak perlu lagi banyak membuat fungsi tambahan sendiri karena framework sudah menyediakan fungsi yang akan dibutuhkan pengembang.
- Panduan pembuatan aplikasi yang disediakan oleh framework mengacu pada langkah kerja yang umum.
- Framework memiliki teknik pemrograman yang lebih rapih dan terstruktur.

BAB 2

JUDUL BAGIAN KEDUA

2.1 Pemasangan PHP di Server Web Windows

Untuk memasang dan menggunakan PHP di lingkungan windows maka kita harus memasang terlebih dahulu software atau program tambahan yang diperlukan oleh PHP untuk dijalankan didalam sistem operasi Windows. Program atau software tambahan untuk windows tersebut adalah Redistributable Visual C++ dari microsoft. yang harus dipasang, disesuaikan dengan sistem operasi windows yang kita gunakan apakah windows 32 bit atau 64 bit. Server web yang akan digunakan untuk dapat menjalankan program PHP yang kita kembangkan yaitu Server web PHP built-in dan Server web eksternal.

Web server inilah yang akan menerjemahkan kode PHP menjadi HTML dan mengirimnya ke web browser untuk ditampilkan. Kita harus menyewa web server agar kode PHP dapat diproses dan diakses di internet. Namun aplikasi web server ini dapat diinstall di komputer lokal, dan inilah yang akan kita install dalam tutorial kali ini. Untuk aplikasi web server, terdapat beberapa pilihan. Saat ini web server yang sering digunakan adalah Apache, Nginx, dan Microsoft IIS. Apache dan Nginx merupakan aplikasi open source dan dapat digunakan dengan gratis. Namun kali ini kita akan menjalankan PHP menggunakan Apache, karena Apache masih menjadi

aplikasi web server yang paling banyak dipakai saat ini. Akan tetapi, proses instalasi web server Apache dan PHP secara terpisah akan membutuhkan waktu yang cukup lama dan juga butuh pengetahuan tentang konfigurasi Apache. Berita baiknya, terdapat banyak aplikasi yang membundel Apache+PHP. Beberapa diantaranya adalah XAMPP dan WAMP. Pada tutorial belajar PHP ini kita akan menggunakan XAMPP. Aplikasi terakhir yang kita butuhkan adalah web browser. Disini saya akan menggunakan web browser UC Browser dan Goggle Chrome.

2.1.1 Cara Install XAMPP

XAMPP adalah singkatan dari aplikasi dalam paketnya, yaitu: X (berarti cross-platform, maksudnya tersedia dalam berbagai sistem operasi), Apache Web Server, MySQL, PHP dan Perl. Dengan menginstall XAMPP, secara tidak langsung kita telah menginstall keempat aplikasi tersebut. File download dapat di download di link berikut <https://www.apachefriends.org/download.html>. Setelah aplikasi installer XAMPP berhasil di unduh, langkah selanjutnya adalah menginstall XAMPP.

1. Double klik file yang sudah anda download (mungkin nama file XAMPP anda akan sedikit berbeda). Apabila pada komputer terpasang anti virus, akan muncul jendela peringatan untuk sementara mematikan anti virus. Anda boleh mengikuti instruksi ini, atau mengabaikannya dan lanjut ke tahap selanjutnya.
2. Setelah 2 jendela peringatan, jendela tampilan awal instalasi XAMPP akan muncul, klik next untuk melanjutkan.
3. Tampilan berikutnya adalah Select Component, pada layar ini anda bisa memilih aplikasi apa saja yang akan diinstall. Disarankan untuk membiarkan pilihan default, dalam hal ini semua aplikasi akan diinstall. Namun jika anda memiliki kendala akan keterbatasan ruang harddisk, anda bisa memilih aplikasi penting seperti Apache, MySQL, PHP, dan phpmyadmin. Setelah itu, lanjutkan dengan klik tombol next.
4. Pada tampilan berikutnya, Installation Folder anda dapat mengubah folder instalasi default XAMPP ke folder lain, dalam contoh ini saya akan memindahkannya ke C lalu folder xampp. Lalu klik next untuk melanjutkan.
5. Jika semuanya berjalan lancar, tampilan berikutnya adalah Ready to Install, klik tombol next untuk memulai proses instalasi.
6. Proses intallasi akan berjalan beberapa saat, hingga layar tampilan menjadi Completing the XAMPP. Biarkan ceklist Do you want to start the Control Panel now? karena kita akan langsung mencoba XAMPP setelah ini.
7. Akhiri proses instalasi XAMPP dengan klik tombol Finish.

2.2 Server web Eksternal untuk PHP

Server web eksternal adalah server web yang digunakan untuk keperluan publikasi dalam dokumen HTML, dan bisa juga dijadikan sebagai platform untuk aplikasi Internet yang interaktif dengan menggunakan browser web sebagai klien dari aplikasinya. Penggunaan server web eksternal sejak pengembangan aplikasi sangat dianjurkan agar lingkungan pengembangan aplikasi akan sama atau mirip dengan lingkungan server web pada saat implementasi dari aplikasi yang dikembangkan. Berikut adalah beberapa server web yang paling banyak digunakan dan dapat dipilih untuk kepentingan pengembang aplikasi.web dengan menggunakan PHP:

1. IIS (Internet Information System) IIS adalah server web yang disediakan oleh microsoft Pengguna bisa mengaktifkannya pada saat instalasi atau setelah sistem operasi terpasang. Server web ini hanya ada pada sistem operasi windows.
2. Apache Apache (<http://apache.org>) adalah server web open source yang dapat digunakan secara bebas oleh siapa pun yang berminat. Server web ini tersedia pada sistem operasi windows dan *nix/Linux.
3. LightTPD Server web ini tersedia pada sistem operasi windows dan *nix/Linux.
4. Nginx merupakan server web alternatif yang ringan dan tersedia untuk sistem operasi windows dan *nix/Linux.

Aplikasi PHP yang akan dihasilkan nanti dapat dijalankan disemua server web, walaupun kita membuatnya diserver web yang berbeda.

2.3 Pemasangan PHP untuk IIS pada windows 10

Berikut adalah langkah untuk pemasangan PHP di server IIS pada windows 10

1. Pastikan bahwa IIS telah terpasang dengan benar dan aktif pada komputer anda
2. Unduh dan jalankan Web Platform Installer (WPI) pada komputer anda
3. Pilih PHP7 yang akan dijalankan

Dalam pemasangan tersebut kita menggunakan WPI (Web Platform Installer), program pemasangan paket program berbasis web yang disediakan oleh Microsoft, agar kita dapat dengan lebih mudah melakukan pemasangan karena menyediakan proses untuk pemasangan secara otomatis. dalam hal ini WPI sendiri memerlukan adanya koneksi internet karena paket yang dipasangkan harus diunduh oleh WPI dari internet secara langsung.

2.4 Localhost

Localhost adalah server lokal atau sebuah web server yang bekerja pada laptop anda. Alamat IP dari localhost adalah 127.0.0.1. Jadi localhost terletak pada laptop anda.

Anda membutuhkan localhost itu sendiri untuk menjalankan file .php yang telah di jelaskan sebelumnya di pengenalan php. PHP hanya dapat berjalan pada sisi sever dan karenanya PHP disebut sebagai pemrograman server side atau pada sisi server.

Localhost dijadikan server pada saat pengembangan aplikasi yang berbasis web sebelum di hostingkan. Localhost hanya dapat diakses laptop pada browser anda dengan alamat <http://127.0.0.1> atau <http://localhost>. Maka halaman akan ditampilkan ke localhost projek aplikasi yang sedang anda buat.

BAB 3

DASAR PENULISAN PHP SEDERHANA

3.1 Aturan Dasar Penulisan Kode PHP

Seperti bahasa pemrograman yang lain, PHP juga memiliki aturan penulisan seperti case sensitivity (perbedaan antara huruf besar dan kecil), cara mengakhiri sebuah baris perintah, dan pengaruh penggunaan spasi dalam membuat kode program PHP. Berikut adalah aturan dasar penulisan kode PHP:

3.1.1 Case Sensitivity (perbedaan huruf besar dan kecil) dalam PHP

PHP tidak membedakan huruf besar dan kecil (case insensitive) untuk penamaan fungsi (function), nama class, maupun keyword bawaan PHP seperti echo, while, dan class. Ketiga baris berikut akan dianggap sama dalam PHP:

```
1 <?php
2 Echo   Hello World ;
3 ECHO   Hello World ;
4 Echo   Hello World ;
5 ?>
```

Akan tetapi, PHP membedakan huruf besar dan huruf kecil (case sensitive) untuk penamaan variabel, sehingga akan dianggap sebagai variabel yang berbeda. Sering kali error terjadi dikarenakan salah menuliskan nama variabel, yang seharusnya menggunakan huruf kecil, ditulis dengan huruf besar.

```
1 <?php
2 $luqman="Luqman";
3 echo $Luqman; // Notice: Undefined variable: Luqman
4 ?>
```

Untuk mengatasi perbedaan ini, disarankan menggunakan huruf kecil untuk seluruh kode PHP, termasuk variabel, fungsi maupun class. Jika membutuhkan nama variabel yang terdiri dari 2 kata, karakter spasi bisa digantikan dengan underscore.

3.1.2 Karakter Spasi dan Tab dalam PHP

Dalam PHP, karakter seperti spasi dan tab diabaikan di dalam eksekusi program PHP. Anda boleh mencoba sebuah statement menjadi beberapa baris, atau menyatukan beberapa statement dalam sebuah baris yang lumayan panjang. Seperti contoh berikut:

```
1 <?php
2 echo "Saya belajar"; echo "Saya mengerti"; $nama="Men";
3 ?>
```

Baris statement itu sama dengan

```
1 <?php
2     echo "Saya belajar";
3     echo "Saya mengerti";
4     $nama = "Men";
5 ?>
```

Walaupun contoh yang pertama lebih menghemat baris, namun lebih disarankan untuk contoh kedua, karena kita mengusahakan agar setiap statement berada dalam satu baris saja, dan menambahkan beberapa spasi di awal untuk memudahkan membaca kode program.

Keuntungan penghematan baris dan beberapa byte dari sebuah file PHP tidak akan sebanding dengan mencoba memahami kode program yang dibuat dalam beberapa hari kedepan. Menambahkan sebagian komentar pada bagian kode yang lebih rumit sebagai penjelasan juga sangat disarankan.

3.2 Embedded Script dan Non Embedded

3.2.1 Embedded Script

Berikut merupakan contoh dokumen HTML yang akan dihasilkan dengan menggunakan program/script PHP dalam embedded script Ditampilkan dibawah ini :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
```

```

4  <title>Contoh Embedded</title>
5  </head>
6  <body>
7    <?php
8      echo "Hai, Saya dari Script PHP";
9    ?>
10
11 </body>
12 </html>

```

Script diatas menunjukkan contoh script PHP sederhana yang disebut dengan script embedded yang di sisipkan diantara tag-tag HTML. Script tersebut digunakan apabila isi dari suatu dokumen HTML diinginkan dari hasil eksekusi suatu script PHP. jika dilihat dari source-nya dengan menggunakan view source pada web browser maka tampilannya akan berupa seperti berikut

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Contoh Embedded</title>
5  </head>
6  <body>
7    Hai, Saya dari Script PHP
8  </body>
9  </html>

```

Source dokumen HTML yang tampil berupa dokumen HTML yang tidak lagi dari script PHP yang berisi script PHP karena semua menjadi tag HTML, karena pada saat dieksekusi maka bukan scriptnya yang dikirim tetapi eksekusi dari script tersebut yang dikirim

3.2.2 Non Embedded

Script PHP dibawah ini merupakan script murni dari pembuatan program dengan menggunakan PHP, tag dokumen HTML yang dihasilkan untuk membuat dokumen merupakan bagian dari script PHP. di tampilkan dibawah ini:

```

1  <?php
2  echo "<html>";
3  echo "<head>";
4  echo "<title>";
5  echo "Ini merupakan Script Embedded";
6  echo "</title>";
7  echo "</head>";
8  echo "<body>";
9  echo "<p>ini teks yang dihasilkan dengan menggunakan dokumen script
    PHP </p>";
10 echo "</body>";
11 echo "</html>";
12 ?>

```

dan dibawah ini merupakan source dokumen HTML dari tampilan kode diatas

```

1  <html><head><title> Ini merupakan Script Embedded </title></head><
    body>

```

```
2 <p>ini teks yang dihasilkan dengan menggunakan dokumen script PHP</p>  
3 </body></html>
```

Jika diperhatikan dokumen HTML tersebut tidak beraturan ditampilkan. Hal tersebut tidak menjadi masalah, yang penting adalah browser web dapat menampilkannya, karena dokumen tag HTML ini murni dihasilkan dari script PHP.

3.3 Variabel dan Tipe Data

3.3.1 Variabel

Variabel adalah tempat penyimpanan data, variabel memiliki nama. dalam pemrograman variabel merupakan tempat penyimpanan data didalam memori komputer. Di dalam PHP nama variabel diawali dengan karakter dollar diikuti dengan huruf sebagai karakter pertama setelah dollar. kemudian kombinasi karakter dan angka. Tidak boleh ada spasi dan tanda baca dalam penamaannya. kecuali karakter (garis bawah, under score). berikut merupakan penulisan variabel yang benar:

```
1 $namauser  
2 $password  
3 $kota2  
4 $tempat_lahir
```

3.3.2 Tipe Data

Data yang diolah oleh suatu program memiliki berbagai jenis ada data yang menunjukkan jumlah dan menunjukkan nilai benar dan salah, atau tulisan. Jenis tipe data dalam PHP secara mendasar dibedakan menjadi 3 macam yang disebut sebagai tipe data primitif. Tipe data primitif yang diolah oleh PHP:

1. Numerik
2. String
3. Boolean

Tipe data numerik dibedakan menjadi tipe data integer dan floating point. Selain itu tipe data yang lain adalah tipe data compound, terdiri atas:

1. tipe data array
2. tipe data objek

3.3.3 Tipe Data Integer

tipe data integer adalah tipe data yang terdiri dari angka bulat (tidak mengandung nilai pecahan atau nilai desimal). Nilai ini bisa berbentuk angka positif maupun

negatif, contohnya 1, 2, 6, -44, 20000, atau 128730123. Tipe data integer dapat dituliskan dengan notasi sebagai berikut

1. Notasi Desimal adalah susunan bilangan yang mempunyai basis sepuluh. Koefisien bilangan desimal terdiri dari 0,1,2,3,4,5,6,7,8,9. Notasi bilangan desimal dituliskan: (n)10
2. Notasi Oktal adalah susunan bilangan yang mempunyai basis delapan. Koefisien bilangan oktal terdiri dari 0,1,2,3,4,5,6,7. Notasi bilangan oktal dituliskan : (n)8
3. Biner adalah susunan bilangan yang mempunyai basis dua. Basis dua di sini adalah nilai koefisien yaitu 0 dan 1. Notasi bilangan biner dituliskan : (n)2
4. Notasi Heksadesimal adalah susunan bilangan yang mempunyai basis enam belas. Koefisien bilangan heksadesimal terdiri dari 0,1,2,3,4,5,6,7,8,9,A,B,C,D,F. Catatan: A bernilai 10, B bernilai 11, ... , F bernilai 15. Notasi bilangan heksadesimal dituliskan : (n)16

desimal	biner	oktal	heksadesimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Gambar 3.1 Sistem Bilangan

3.3.4 Tipe Data Floting Point

Tipe data float (disebut juga tipe data floating point, atau real number) adalah tipe data angka yang memiliki bagian desimal di akhir angka, atau memiliki floating point (floating point adalah istilah dalam bahasa inggris untuk menyebut tanda titik yang menandakan bilangan desimal). Contoh angka float adalah seperti: 0,9 atau 3,14. Tipe data float cocok digunakan untuk variabel yang akan berisi angka pecahan,

seperti nilai IPK, hasil pembagian, atau hasil komputasi numerik yang angkanya tidak bisa ditampung oleh data integer.

3.3.5 Tipe Data String

Tipe data String adalah tipe data untuk teks yang merupakan gabungan huruf, angka, whitespace (spasi), dan berbagai karakter. Fungsi ini digunakan untuk membuat identifier String/teks. Data string ditulis dengan mengapit data string tersebut dengan tanda petik tunggal atau tanda petik ganda. Tanda petik tunggal umumnya digunakan sebagai konstanta string.

3.3.6 Tipe Data Boolean

Tipe data boolean sebenarnya sangat sederhana. Tipe data ini hanya bisa diisi dengan salah satu dari 2 nilai: TRUE atau FALSE. Tipe data boolean banyak dipakai dalam percabangan kode program, atau untuk memutuskan apa yang harus dijalankan pada sebuah kondisi if else.

3.3.7 Tipe Data Objek

Tipe data dari objek merupakan tipe data baru, merupakan pengembangan PHP untuk mendukung program berorientasi objek. Tipe data objek adalah tipe data yang didalamnya mempunyai data dan method. data yang dipunyai oleh suatu objek populer dengan nama atribut dan method suatu objek umumnya berupa suatu fungsi.

Data objek didefinisikan dengan membuat definisi kelas terlebih dahulu. Suatu variabel yang bertipe objek diinisialisasi (dideklarasikan) dengan menggunakan perintah new kemudian nama objek (berupa nama kelas objek) berikut contohnya:

```

1 <!-- Contoh Tipe data Objek -->
2
3 <?php
4     class objek {
5         function counter(){
6             return 5;
7         }
8         function do_pesan(){
9             echo "Cetak Pesan.";
10        }
11    }
12
13    $test = new objek;
14    $test->do_pesan();
15    echo "</br>";
16    echo $test->counter();
17 ?>

```

3.4 Operator PHP

Pada PHP, terdapat banyak operator beberapa yang sering digunakan.

3.4.1 Operator Perbandingan

Seperti namanya, operator perbandingan digunakan untuk membandingkan beberapa buah nilai pada PHP dan hasilnya berupa boolean `true` yang berarti benar atau `false` yang berarti salah. Contoh:

```
1 <?php
2 if ($_POST['password'] == 'admin')
3 {
4     echo 'Login sukses';
5 }
```

3.4.2 Operator PHP Increment dan Decrement

Operator ini digunakan untuk menambahkan atau mengurangi nilai sebanyak 1 pada suatu variabel.

3.4.3 Perbedaan Pre Increment dan Post Increment

Pada pre increment, nilai variabel akan ditambahkan 1 baru kemudian siap digunakan, sebaliknya, untuk post increment, gunakan dulu nilai variabel kemudian baru ditambahkan dengan 1. Contoh 1:

```
1 <?php
2 $nomor = 1;
3 while($nomor <= 5) {
4     echo $nomor++;
5 }
```

Contoh diatas akan menghasilkan angka 12345.

Contoh 2:

```
1 <?php
2 $nomor = 1;
3 while($nomor <= 5) {
4     echo ++$nomor;
5 }
```

Contoh diatas akan menghasilkan 23456. Lihat, perbedaanya terdapat pada `++` sebelum dan sesudah `$nomor`.

3.4.4 Operator Assignment PHP

Sesuai namanya operator assignment ini digunakan untuk memberikan nilai pada suatu variabel. Operator dasarnya adalah tanda sama dengan (`=`). Dalam praktiknya, operator ini sering digunakan ketika menjumlahkan nilai pada suatu perulangan, seperti ketika menjumlahkan data hasil query database. Contoh:


```
1 <?php
2 $sql = 'SELECT * FROM sales';
3 $query = mysqli_query($sql);
4 $total = 0;
5 while($row = mysqli_fetch_array($query))
6 {
7     $total += $row['jml_bayar'];
8 }
9 ?>
```

3.5 Lingkup Variabel

Variabel Scope (atau ruang lingkup variabel) adalah jangkauan kode program dimana perintah program masih bisa mengakses sebuah variabel. Variabel menunjukkan keberlakuan dan dikenalnya suatu variabel didalam script. Suatu variabel yang didefinisikan didalam sebuah fungsi maka variabel tersebut hanya akan dikenali dan digunakan hanya dalam fungsi tersebut variabel ini dikenal sebagai variabel lokal. karena hanya dikenal pada fungsi tempat variabel tersebut dinyatakan dan digunakan.

Jika kita mendefenisikan sebuah variabel pada satu file PHP, maka variabel tersebut dapat diakses oleh seluruh kode program pada halaman yang sama. Namun jika variabel tersebut di definisikan di dalam sebuah fungsi, variabel itu belum tentu bisa diakses dari luar fungsi tersebut. Hal inilah yang dimaksud dengan Variabel Scope. Variabel akan disebut global apabila variabel tersebut dapat dikenali dan digunakan oleh seluruh bagian script tersebut

Variabel yang didefenisikan di dalam sebuah fungsi, secara default tidak dapat diakses oleh kode program di luar fungsi tersebut. Dan begitu juga sebaliknya, variabel yang didefenisikan di luar fungsi, tidak bisa diakses dari dalam fungsi. Berikut jenis variabel berdasarkan lingkungnya:

3.5.1 Variabel Global

Variabel dan nilainya dikenali dan dapat digunakan oleh semua bagian script yang membutuhkannya. Semua variabel yang dibuat pada bagian utama script bukan pada bagian suatu fungsi, variabel variabel ini akan bersifat global. Pengertian global disini diartikan sebagai variabel dan data yang ada didalamnya hanya dikenali oleh seluruh bagian script, jika script dieksekusi oleh PHP.

Fungsi-fungsi yang ingin menggunakan variabel dan data yang ada pada variabel global, maka dalam fungsi harus dideklarasikan global untuk nama variabel tersebut contohnya:

```
1 <?php
2
3 function test() {
4     global $a;
5     global $b;
6     $b=15;
```

```

7      echo $a; //7
8      echo $b; //15
9  }
10     test();
11     echo $a; //7
12     echo $b; //15
13
14 ?>

```

Variabel array global merupakan variabel asosiatif internal PHP yang mencatat semua variabel global yang dimiliki oleh suatu script. Proses penggunaan suatu variabel global dalam suatu fungsi dapat memanfaatkan array berikut contohnya:

```

1 <?php
2     $a = 1;
3     $b = 2;
4
5     function test() {
6         $GLOBALS['c'] = $GLOBALS['a'] + $GLOBALS['b'];
7     }
8
9     test();
10    echo $c;
11 ?>

```

agar variabel dan data didalamnya dapat dipakai oleh script-script yang saling berhubungan dan membutuhkan maka variabel harus dinyatakan dengan cookie, method GET, ataupun method POST.

3.5.2 Variabel Lokal

Variabel lokal adalah variabel dan data yang dimilikinya hanya dapat digunakan oleh fungsi tempat variabel digunakan dideklarasikan. Nama variabel bisa memiliki nama yang sama dengan nama variabel global yang ada pada script, tetapi tetap yang diacuh oleh fungsi adalah variabel lokal. Variable jenis ini di tulis di dalam function, dan hanya dapat di akses dari dalam function juga, dengan kata lain tidak bisa di akses di luar function, berikut contoh syntax nya :

```

1 <?php
2     function test() {
3         $a = 5; // Variable local
4         echo "<p>Variable a didalam function adalah: $a</p>";
5     }
6     test();
7
8     // Jika kita memanggil variable local maka tidak akan ada nilainya
9     echo "<p>Variable a diluar function adalah: $a</p>";
10 ?>
11
12 // Output program
13 // Variable a didalam function adalah: 5
14 // Variable a diluar function adalah:

```

Syntax diatas dapat terlihat jelas, jika Variable Local di akses di dalam function maka nilainya keluar yakni 5, sedang jika di akses diluar function maka nilai itu tidak ada.

3.5.3 Variabel Statik

Variabel statik dalam fungsi yang memungkinkan nilai yang terakhir yang ada di dalamnya dapat dipertahankan, karena secara biasa setiap fungsi selesai dieksekusi, pada saat dipanggil kembali nilai yang ada pada variabel dalam fungsi tersebut akan diinisialisasi lagi. Pernyataan `static` memungkinkan nilai yang terakhir dipertahankan, sehingga pada saat dipanggil kembali fungsi tersebut masih memiliki nilai yang terakhir. Biasanya ketika sebuah function selesai di jalankan dalam sebuah syntax PHP, semua variable di dalamnya akan di hapus, namun terkadang kita ingin agar variable tersebut tidak di hapus, akan tetapi akan kita gunakan secara lanjut, untuk itu kita harus menggunakan variable statis dalam case ini, berikut syntax nya :

```

1 <?php
2 function test() {
3     static $a = 8;
4     echo $a;
5     $a++;
6 }
7
8 test();
9 test();
10 test();
11 ?>
12
13 // Output
14 // 8
15 // 9
16 // 10

```

Perbedaan dengan jenis variable lain ialah terletak, jika setiap function dipanggil, nilai yang ada di dalam variable adalah nilai terakhir, bukan nilai yang di deskripsikan seperti syntax diatas yakni 8.

3.6 Struktur Kontrol

PHP melakukan eksekusi dengan perintah mulai dari baris pertama kemudian ke baris berikutnya, sampai baris yang terakhir. Struktur kontrol digunakan dalam mengatur alur logika program agar sesuai dengan kenyataan. Struktur kontrol akan melibatkan variabel, tipe data, dan operator.

3.6.1 If Statement

If Statement adalah pernyataan yang hanya akan dijalankan jika suatu kondisi bernilai benar, berfungsi untuk melakukan filter/penyaringan hasil berdasarkan kondisi tertentu. Contoh:

```

1 If (kondisi) {
2     Pernyataan 1;
3     Pernyataan 2;
4     .....

```

```

5 } else {
6     Pernyataan a;
7     Pernyataan b;
8     .....
9 }

```

3.6.2 Perulangan

Perulangan digunakan untuk mengeksekusi suatu pernyataan secara berulang-ulang. Contoh:

```

1 While (kondisi){
2     Pernyataan yang diulang;
3     Counter;
4 }

```

3.6.3 Fungsi

Fungsi adalah serangkaian kode yang terdapat kegunaan khusus dan tertentu, dengan adanya fungsi ini pemrograman dapat dipermudah karena tidak harus menulis berulang-ulang rangkian kode yang sama. Demikian juga dalam pengembangan, jika terjadi kesalahan atau perbaikan kode maka pemrogram hanya dapat melakukan perbaikan pada fungsi tertentu saja, tidak perlu melakukan perbaikan pada banyak kode. Contoh:

```

1 $tanggal = date( format );

```

3.6.4 Cookie dan Session

Dalam php kita kenal session dan cookies yang digunakan untuk menyimpan informasi pengguna. Secara umum memang sulit dibedakan karena dari faktor fungsinya bisa dikatakan sama. Cookies adalah informasi yang dapat disimpan di komputer kita dengan bantuan browser.

Cookies dapat diakses kapanpun melalui halaman-halaman php selama cookies ini masih tersimpan. Cookies disimpan di komputer kita selama dalam sebuah file kecil yang diletakkan pada folder tertentu oleh browser.

```

1 <?php
2     $test = $_COOKIE["Test"];
3     echo $test;
4 ?>

```

Penyimpanan informasi dengan cookies paling sering digunakan untuk:

1. Menyimpan username dan password dalam login agar pengguna tidak selalu harus mengisikannya pada saat membuka halaman.
2. Untuk mencatat konfigurasi yang direkam oleh pengguna, seperti warna tema, jenis huruf, dan pilihan bahasa.

3. Untuk mengetahui apakah pengunjung pernah datang atau belum ke halaman yang sedang dibuka.

Session dapat diartikan sebagai sebuah variabel yang bersifat global yang diciptakan dalam server php pada saat sesi awal membuka sebuah halaman dan berlaku sampai anda menutup halaman tersebut.

```
1 <?php
2     session_start();
3 ?>
```

Session ini sering digunakan untuk:

1. Menyimpan informasi login yang berlaku hanya dalam satu sesi.
2. Menyimpan catatan order barang dalam sistem transaksi online.

3.7 Koneksi Dengan Database MySQL

Caranya buatlah file PHP dengan nama koneksi.php selanjutnya simpan file PHP tersebut di dalam directory localhost kita.

```
1 <?php
2 // isi nama host, username mysql, dan password mysql kita
3 mysqli_connect("localhost","root","");
4
5 // isikan dengan nama database yang akan di hubungkan
6 mysqli_select_db("test");
7 ?>
```

Seperti penjelasan diatas function connect() isikan nama host kita, username mysql dan password mysql kita. Fungsi select db adalah fungsi yang di gunakan PHP untuk memilih database yang ingin kita hubungkan. Jadi kita tinggal memasukan nama database yang akan digunakan kita tinggal memasukkan file koneksi tersebut ke projek yang ada di localhost.

3.8 Perulangan

Struktur perulangan (atau dalam bahasa inggris disebut dengan loop) adalah sebuah instruksi program yang bertujuan untuk mengulang beberapa baris perintah. Dalam merancang ini perulangan kode program, kita setidaknya harus mengetahui 3 komponen, yaitu kondisi awal dari perulangan, perintah program yang akan diulang, serta kondisi akhir dimana perulangan akan berhenti.

Sebagai contoh sederhana untuk perulangan for, saya akan membuat program PHP untuk menampilkan 10 baris kalimat Test. Berikut adalah kode program yang digunakan:

```
1 <?php
2 for ($i= 1; $i <= 10; $i++)
```

```
3 {  
4     echo "Test";  
5     echo "<br />";  
6 }  
7 ?>
```


BAB 4

ARRAY

4.1 Array

Array adalah variabel jamak, yang mempunyai banyak elemen yang diacu dengan satu nama yang sama. Array (atau larik dalam bahasa indonesia) bukanlah tipe data dasar seperti integer atau boolean, Array adalah sebuah tipe data bentukan yang terdiri dari kumpulan tipe data lainnya. Menggunakan array akan memudahkan dalam membuat kelompok data, serta menghemat penulisan dan penggunaan variabel. berikut sebagai contoh

```
1 <?php
2     $a = array("budi", 20, 58.5);
3 ?>
```

Array dalam PHP juga merupakan tipe data, bukan sekedar variabel. Berikut merupakan jenis array dalam PHP:

4.1.1 **Array Berindeks**

Array berindeks adalah array yang diindeks berdasarkan nomor/angka. Indeks array pada umumnya dimulai dari angka 0. Anda bebas mendefinisikan indeks dengan nilai yang Anda tentukan.

Tabel 4.1 (

Array Berindeks

1020304050

1	\$a[0]
1	\$a[1]
1	\$a[2]
1	\$a[3]
1	\$a[4]

Contoh diatas menunjukan array dengan 5 buah elemen. Elemen pertama (*a[0]*) bernilai 10, *a[1]* bernilai 20, dan seterusnya. Dalam array berindeks, antara kunci (indeks) dan nilai tidak memiliki keterkaitan.

4.1.2 **Array Asosiatif**

Array asosiatif adalah array yang diindeks berdasarkan nama tertentu. Letak perbedaan antara array berindeks dan array asosiatif adalah hanya terletak pada penamaan indeksnya saja.

Tabel 4.2 (

Array Asosiatif

10	20	30	40	50
\$a["satu"]	\$a["dua"]	\$a["tiga"]	\$a["empat"]	\$a["lima"]

Array diindeks berdasarkan nama, bukan berdasarkan nomor. Pada contoh diatas indeks array bertipe string. Pada umumnya array asosiatif digunakan untuk merepresentasikan sesuatu yang kunci dan nilainya memiliki keterkaitan, misalnya sebagai berikut.

```
1 <?php
2 $kota = array("jkt" => "jakarta", "bdg" => "bandung", "sby" => "
   surabaya");
3 ?>
```

4.2 Membuat Array

Array dapat dibuat melalui dua cara, yaitu dengan menggunakan fungsi `array()` atau dengan membuat elemen-elemen array dan mengisi nilai-nilai ke dalam elemen-elemen tersebut secara langsung. Berikut ini contoh pembuatan array dengan menggunakan fungsi `array()`.

Untuk Array Berindeks

```
1 <?php
2 $matakuliah = array ("pemograman web", "database", "keamanan
   jaringan",
3 "sistem informasi", "rekayasa perangkat lunak");
4 ?>
```

Untuk Array Asosiatif

```
1 <?php
2 $detailmk = array ("kode" => "TKB5218", "nama" => "pemograman web 2",
   "sks" => 2);
3 ?>
```

Berikut ini contoh pembuatan array dengan cara langsung membuat variabel array dan mengisi nilai ke dalamnya.

Untuk array berindeks

```
1 <?php
2 $matakuliah[0] = "pemograman web";
3 $matakuliah[1] = "database";
4 $matakuliah[2] = "keamanan jaringan";
5 $matakuliah[3] = "sistem informasi";
6 $matakuliah[4] = "rekayasa perangkat lunak";
7 ?>
```

Untuk array asosiatif

```
1 <?php
2 $detailmk["kode"] = "TKB5218";
3 $detailmk["nama"] = "pemograman web 2";
4 $detailmk["sks"] = 2;
5 ?>
```

4.2.1 File Direktori

Dalam management file dan direktori, PHP menyediakan lebih 70 fungsi. Beberapa fungsi utama yang berhubungan dengan management file (create, write, append, dan delete), antara lain : Membuka dan membuat file.

```
1 fopen ($namafile , $mode);
```

Keterangan : `namafile` merupakan nama file yang akan dibuat, sedangkan `mode` merupakan mode akses file. Contoh:

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile , "mode");
4 if (!handle) {
5     echo "<b>File yang ada buat belum ada</b>";
6 } else {
7     echo "<b>File telah berhasil dibuka</b>";
8 }
9 fclose($handle);
10 ?>

```

4.2.2 Pemrograman Berorientasi Objek Dalam PHP

PHP pada awalnya hanya sekumpulan script sederhana. Dalam perkembangannya, dapat ditambahkan berbagai fitur pemrograman berorientasi objek. Hal ini dimulai sejak PHP 4. Dengan lahirnya PHP 5, fitur-fitur pemrograman berorientasi objek semakin mantap dan semakin cepat. Dengan PHP 7, script yang menggunakan konsep object-oriented akan lebih cepat dan lebih efisien.

Pemrograman berorientasi objek atau object-oriented programming (OOP) merupakan suatu inovasi dalam pemrograman yang menggunakan object dan class. Saat ini konsep OOP sudah semakin berkembang. Hampir setiap perguruan tinggi di dunia mengajarkan konsep OOP ini pada mahasiswanya. Pemrograman yang banyak dipakai dalam penerapan konsep OOP adalah Java dan C++. OOP bukanlah sekedar cara penulisan sintaks program yang berbeda, namun dapat lebih dari itu, OOP merupakan cara pandang dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah object. Sebuah object mewakili suatu bagian program yang akan diselesaikan. Beberapa konsep OOP dasar, antara lain :

1. Encapsulation (Class dan Object)
2. Inheritance (Penurunan sifat)
3. Polymorphisme

PHP khususnya PHP 7 sudah mendukung beberapa konsep OOP. Akan tetapi PHP 7 tidak mendukung konsep Multiple-inheritance dan polymorphisme.

4.2.3 Object Dan Class

Bagian dasar dari program yang berorientasi objek adalah objects. Secara tidak langsung kita dapat memahami mengenai object ini. Sebagai contoh, sebuah mobil adalah objek. Sebuah mobil mempunyai properties atau bagian-bagian di dalamnya, seperti warna, mesin, roda, pintu dsb. Sebuah mobil juga dapat melakukan sesuatu, seperti mengisi bensin, menyalakan mesin, berjalan, mengerem dsb.

Biasanya object adalah sebuah kata benda. Orang adalah object. Demikian juga mobil, pohon, bunga, komputer, TV, buku dsb. Namun, object tidak selamanya sebuah objek fisik. Bisa saja sebuah benda abstrak, seperti account bank, sebuah file di komputer, database, pesan email, acara TV, dsb. Class merupakan penjelasan atau

deskripsi dari object. Di dalam class, terdapat penjelasan tentang suatu object termasuk properties yang dimilikinya dan kelakuan atau method yang bisa dilakukan oleh object. Sebagai contoh, class Orang. Class Orang tentu setidaknya memiliki beberapa bagian seperti tangan, kaki, mata, telinga dsb. Class Orang juga setidaknya harus bisa jalan, bisa loncat, bisa lari, bisa melihat, bisa bicara dsb.

Salah satu keuntungan program didefinisikan dengan konsep OOP adalah adanya pengkapsulan (encapsulation) program dalam class dan object, dimana sang programmer yang menggunakan class tidak perlu mengetahui isi dan jalannya class secara detail, hanya perlu tahu bagaimana cara menggunakannya. Sama halnya dengan sebuah mobil misalnya. Seorang pemilik mobil tentunya tidak perlu mengetahui bagian-bagian mobil secara menyeluruh. Dia tidak perlu mengetahui bagaimana mesin mobil melakukan pembakaran dan bagaimana mesin mobil bisa menggerakkan roda, dsb. Dia hanya perlu tahu bagaimana cara menjalankan mobil, bagaimana menghentikan mobil, dan fungsi mobil lainnya.

4.2.4 Properties Dan Method

Setiap class dapat memiliki properties yang kadang disebut juga attributes. Properties sebuah mobil misalnya warna, ukuran, harga dsb. Di dalam class, properties dapat dinyatakan dengan sebuah variabel. Misalnya warna, harga, dsb. Method merupakan sesuatu yang bisa dilakukan oleh object. Method dalam PHP sama halnya dengan sebuah fungsi. Method yang mungkin mempunyai sebuah mobil misalnya, method untuk menghidupkan mobil, menjalankan mobil, menghentikan mobil, dsb. Penamaan properties dan method memiliki sejumlah aturan yang sama dengan penamaan sebuah variabel atau fungsi. Akan tetapi berdasarkan kesepakatan (convention), penamaan properties dan method harus menggunakan camel Caps, dimana tiap kata diawali dengan huruf besar kecuali kata pertama, setiap kata digabung tanpa spasi atau under-score.

Mendefinisikan Class bentuk umum untuk mendefinisikan sebuah class contohnya sebagai berikut:

```
1 class namaClass
2 {
3     Deklarasikan dan definisikan properties di sini
4     Definisikan semua method di sini
5 }
```

Penamaan namaClass pada dasarnya sama dengan penamaan variabel. Penamaannya bebas, boleh apa saja, kecuali stdClass. PHP sudah menggunakan nama stdClass sebagai nama class built-in. Isi tubuh class terletak di antara tanda kurung kurawal buka dan kurawal tutup. Di tubuh class terdapat pendefinisian properties (variabel) dan method-method class.

4.2.5 Mengakses Elemen Array

Setelah array dibuat, langkah selanjutnya adalah mengakses nilai-nilai yang terkandung di dalamnya. Cara mengakses elemen array sangatlah sederhana. Karena el-

emen array berupa nilai, maka kita dapat memperlakukannya seperti layaknya variabel. Kita dapat menempatkan nilai yang diakses ke dalam suatu variabel atau dapat juga diproses secara langsung baik dalam perhitungan maupun ditampilkan langsung. Berikut ini adalah contoh mengakses nilai array ke dalam suatu variabel.

```

1 <?php
2 //menampung elemen array dalam variabel – array berindeks
3 $kategori = $matakuliah[0];
4 echo $kategori;
5 //menampung elemen array dalam variabel – array asosiatif
6 $detail = $detailmk["nama"];
7 echo $detail;
8 ?>

```

Berikut ini adalah contoh mengakses elemen array secara langsung

```

1 <?php
2 //membuat kategori array berindeks
3 $matakuliah[0] = "pemograman web";
4 $matakuliah[1] = "database";
5 $matakuliah[2] = "keamanan jaringan";
6 $matakuliah[3] = "sistem informasi";
7 $matakuliah[4] = "rekayasa perangkat lunak";
8 //membuat kategori array asosiatif
9 $detailmk = array("kode" => "TKB5218", "nama" => "pemograman web 2",
10 "sks"
11 => 2);
12 //mengakses array berindeks
13 echo "DAFTAR MATAKULIAH" . "<br>";
14 echo $matakuliah[0] . "<br>";
15 echo $matakuliah[1] . "<br>";
16 echo $matakuliah[2] . "<br>";
17 echo $matakuliah[3] . "<br>";
18 echo $matakuliah[4] . "<br>";
19 //mengakses array asosiatif
20 echo "<br> . "DETAIL MATAKULIAH" . "<br>";
21 echo "KODE : " . $detailmk["kode"] . "<br>";
22 echo "NAMA : " . $detailmk["nama"] . "<br>";
23 echo "JUMLAH SKS : " . $detailmk["sks"] . "<br>";
24 ?>

```

4.2.6 Fungsi-fungsi yang berhubungan dengan array

Berikut ini beberapa fungsi yang dapat digunakan pada variabel bertipe array.

Fungsi	Kegunaan & Bentuk Umum
is_array()	Untuk memeriksa apakah suatu variabel bertipe array atau bukan Bentuk Umum: is_array(\$nilai)
in_array()	Untuk mencari \$nilai di dalam \$array Bentuk Umum: bool in_array(mixed \$nilai, array \$array [, bool \$strict=false])
array_keys()	Fungsi ini akan mengembalikan daftar nilai kunci (berupa numerik maupun string) yang terdapat di dalam array Bentuk umum: array array_keys(array \$array [,mixed \$nilai [,bool \$strict = false]])
array_key_exists()	Fungsi ini digunakan untuk memeriksa apakah suatu kunci tertentu terdapat di dalam array atau tidak Bentuk umum: bool array_key_exists(mixed \$key, array \$array)
array_values()	Fungsi ini akan mengembalikan daftar nilai yang terdapat di dalam array. Bentuk umum: array array_values(array \$array)

Fungsi	Kegunaan & Bentuk Umum
array_search()	Fungsi ini mengembalikan indeks dari nilai yang ditemukan di dalam array Bentuk umum: mixed array_search(mixed \$nilai, array \$array [, bool \$strict = false])
count() dan sizeof()	Untuk mengetahui jumlah elemen di dalam array Bentuk umum: count(\$nilai) dan/atau sizeof(\$nilai)
sort()	Fungsi ini akan mengurutkan (<i>ascending</i>) array berdasarkan nilai elemennya (dalam pengurutan ini, indeks array akan disusun ulang) Bentuk umum: bool sort(array \$array [,int \$sort_flags = SORT_REGULAR])
asort()	Untuk mengurutkan (<i>ascending</i>) elemen-elemen pada array asosiatif Bentuk umum: bool asort(array \$array [,int \$sort_flags = SORT_REGULAR])
ksort()	Fungsi khusus untuk mengurutkan (<i>ascending</i>) elemen array berdasarkan kunci (<i>key</i>) atau indeksinya Bentuk umum: bool ksort(array \$array [,int \$sort_flags = SORT_REGULAR])
rshot()	Fungsi ini akan mengurutkan (<i>descending</i>) array berdasarkan nilai elemennya (dalam pengurutan ini, indeks array akan disusun ulang) Bentuk umum: bool rsort(array \$array [,int \$sort_flags = SORT_REGULAR])
arsort()	Untuk mengurutkan (<i>descending</i>) elemen-elemen pada array asosiatif Bentuk umum: bool arsort(array \$array [,int \$sort_flags = SORT_REGULAR])
krsort()	Fungsi khusus untuk mengurutkan (<i>descending</i>) elemen array berdasarkan kunci (<i>key</i>) atau indeksinya Bentuk umum: bool krsort(array \$array [,int \$sort_flags = SORT_REGULAR])

Gambar 4.1 Fungsi-fungsi yang berhubungan dengan array

OBJECT ORIENTED PROGRAMMING

5.1 E-Logbook

E-Logbook adalah sebuah buku elektronik untuk mencatat catatan/dokumen penting secara detail setiap aktivitas yang berisi masalah-masalah yang membutuhkan tindak lanjut dari pihak yang terlibat dalam satu hari penuh. Seluruh pegawai sebaiknya membaca buku ini agar mengetahui kegiatan, kerusakan, dan target pekerjaan apa saja yang dilakukan hari sebelumnya. Ada beberapa manfaat e-logbook antara lain:

1. bahan bukti untuk merekap seluruh aktivitas
2. bahan pembuatan laporan kegiatan
3. alat untuk memudahkan pegawai dalam merekap kegiatan

Hal yang perlu di isi dalam e-logbook ini antara lain:

1. Hari, tanggal dan tahun
2. Nama pegawai yang dinas pada hari tersebut
3. Nama vendor yang dinas pada hari tersebut

4. Penerbangan hari ini
5. Kegiatan
6. Kerusakan
7. Target pekerjaan

5.1.1 Mengaktifkan MySQLi

Kenapa disini kita membahas tentang cara mengaktifkan MySQLi karena PHP 5 keatas default-nya menggunakan platform MySQLi untuk menggunakan berbagai fungsi pada database MySQL. MySQLi adalah sebuah class di PHP, jadi pastikan bahwa versi PHP kita sudah 5 keatas yaa. Keunggulan menggunakan MySQLi ke-timbang dengan MySQL:

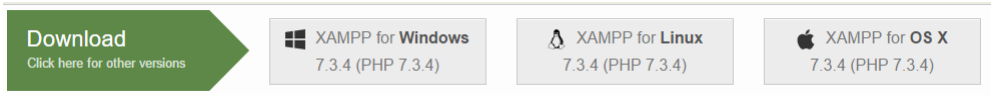
1. Dukungan baru untuk keperluan transaksi
2. Prosedur interface
3. Susunan laporan lebih tersusun
4. Debugging lebih ditingkatkan
5. Dapat memproses dalam waktu yang lebih singkat

Nah itu beberapa keunggulan menggunakan MySQLi. Dan sekarang kita akan membahas cara mengaktifkan MySQLi pada PHP. Untuk mengaktifkan MySQLi, langkah pertama update dahulu versi PHP kita ke PHP 5 keatas. kemudian cari file `php.ini` biasanya terdapat di folder C lalu folder xampp lalu folder php kemudian buka file `php.ini` menggunakan editor seperti notepad++, sublime text, dan adobe dreamweaver. Tambahkan skrip `extension=php_mysqli.dll` pada file `php.ini`. Namun pada file `php.ini` sudah ada skrip `extension=php_mysqli.dll` dan terdapat tanda `;` (tanpa tanda kutip) di depannya, maka kita cukup menghapus tanda `;` tersebut lalu simpan file `php.ini` yang sudah di edit. Jangan lupa untuk restart server apachenya.

5.1.2 Cara Install XAMPP

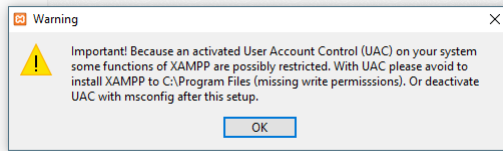
Agar dapat menjalankan sistem yang akan dibuat, kita harus menginstall aplikasi web server yang mendukung PHP ini serta aplikasi untuk database MySQL. Untungnya terdapat banyak aplikasi yang handle program ini, salah satunya yaitu XAMPP. Aplikasi XAMPP adalah aplikasi yang dapat handle banyak aplikasi lain yang dibutuhkan untuk pengembangan web. Nama XAMPP adalah singkatan dari X (huruf X berarti cross-platform), A (Apache web server), M (MySQL), P (PHP), dan P (Perl). Selain beberapa aplikasi tersebut XAMPP menyediakan modul lain seperti OpenSSL dan phpMyAdmin.

Cara mendownload XAMPP terbaru bisa di situs resminya yaitu www.apachefriends.org. Untuk versi terbaru sudah support untuk PHP 7, silahkan pilih download sesuai dengan sistem operasi yang kita pakai.



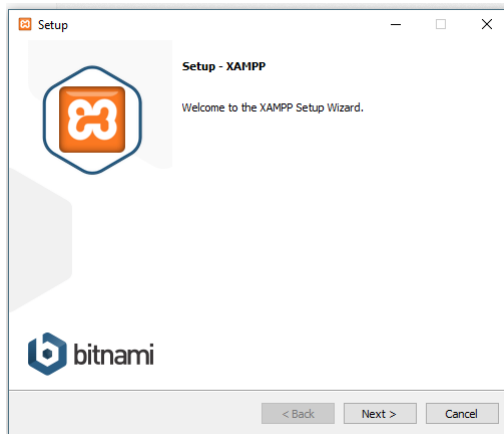
Gambar 5.1 Versi Terbaru XAMPP

File xampp-windows-x64-7.3.4-0-VC15-installer berukuran cukup besar, sekitar 149MB. Simpanlah file ini dimana kita inginkan. Setelah file XAMPP sudah di download, kita akan menginstallnya dengan cara double klik pada aplikasi tersebut dan akan muncul peringatan sebagai berikut.



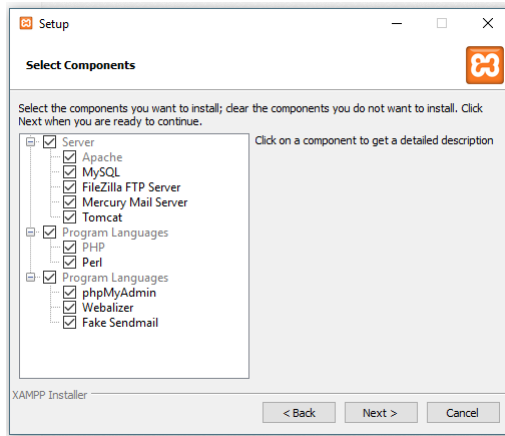
Gambar 5.2 User Account Control

Peringatan ini berkaitan dengan keamanan pada versi Windows Vista keatas dan jika XAMPP akan di install pada folder C mungkin akan terjadi pembatasan hak akses terhadap XAMPP yang berjalan tidak normal. Silahkan klik tombol OK untuk melanjutkan install maka akan muncul jendela awal install. Silahkan klik next.



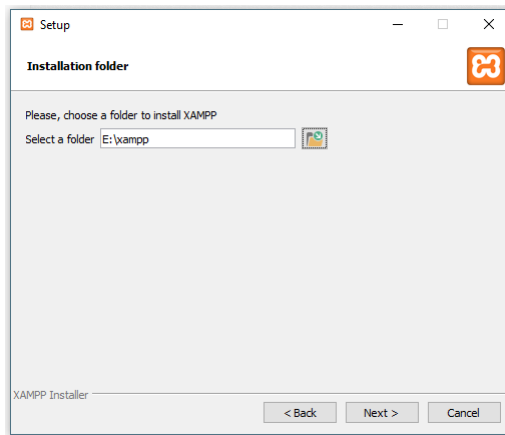
Gambar 5.3 Jendela Awal

Jendela selanjutnya adalah Select Component. Dalam jendela ini kita dapat memilih modul atau aplikasi apa saja yang akan kita install. Dalam tahap ini kita akan menceklis semua pilihan selanjutnya klik next untuk melanjutkan.



Gambar 5.4 Select Component

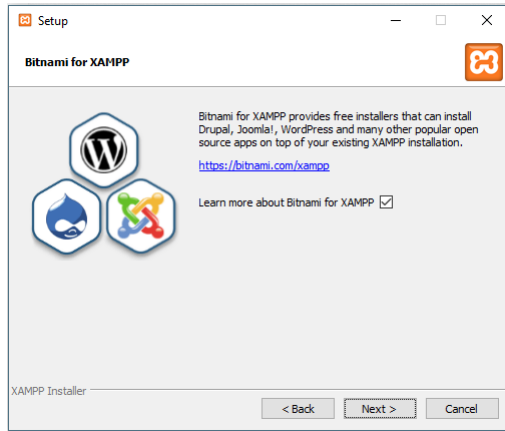
Jendela selanjutnya adalah Installation Folder. Dalam jendela ini kita dapat mengubah lokasi dimana kita akan menyimpan file-file XAMPP. Sebagai contoh kita akan menyimpan file XAMPP di drive E dengan nama folder xampp agar mudah di ingat. Untuk melanjutkan klik next.



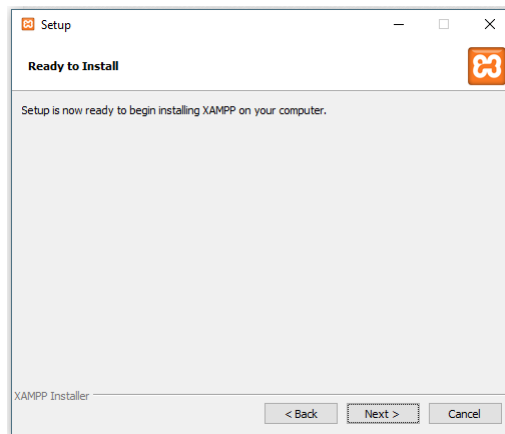
Gambar 5.5 Installation Folder

Jendela berikutnya adalah Bitnami for XAMPP. Dalam hal ini XAMPP menawarkan Bitnami sebagai cara cepat untuk install CMS seperti wordpress, drupal, dan joomla. Kemudian klik next.

Jendela berikutnya adalah pemberitahuan bahwa kita siap untuk menginstall XAMPP. Klik next dan XAMPP akan memulai proses penginstallan.



Gambar 5.6 Bitnami for XAMPP



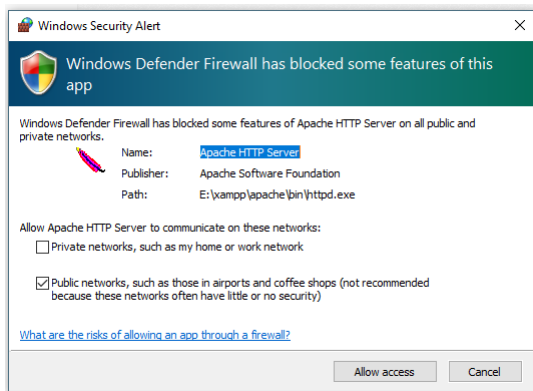
Gambar 5.7 Ready to Install

Setelah proses penginstallan hampir selesai akan muncul jendela Windows Security Alert pada gambar 5.9 karena Windows Defender Firewall mendeteksi Apache HTTP Server. Untuk lanjut klik Allow access.

Proses penginstallan XAMPP sudah selesai pada gambar 5.10 maka akan muncul jendela Completing, dalam bagian ini kita dapat memilih untuk langsung menggunakan XAMPP atau tidak. Jika ingin langsung memakai, ceklis pada Do you want to start the Control Panel now? lalu klik finish. Jika tidak maka hilangkan ceklis dari Do you want to start the Control Panel now? lalu klik finish.



Gambar 5.8 Proses Penginstallan



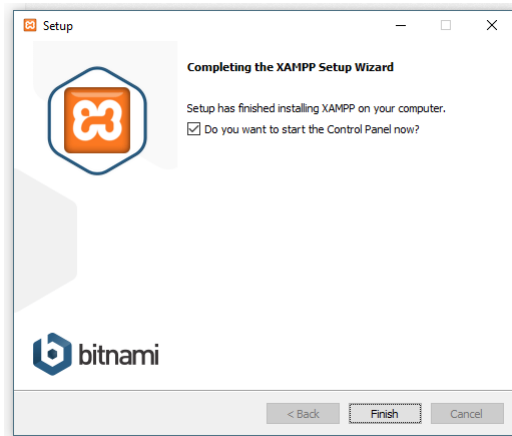
Gambar 5.9 Windows Security Alert

5.1.3 Menguji Instalasi XAMPP

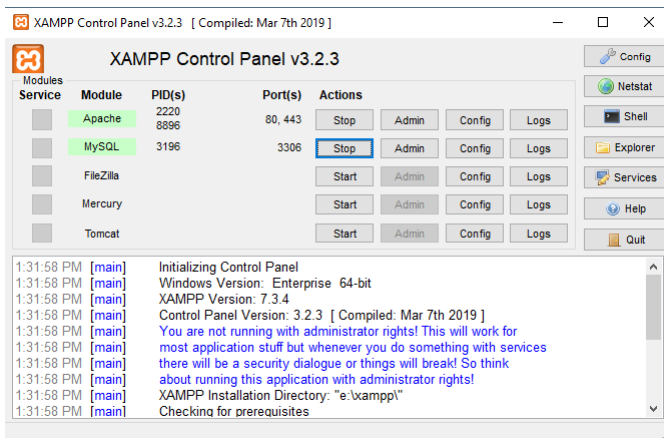
Sesuai dengan namanya, jendela XAMPP Control Panel adalah jendela yang digunakan untuk mengontrol apa saja modul atau aplikasi apa saja yang ingin kita jalankan. Jika ingin membuka manual maka kita dapat membuka dengan cara dari menu Start-*↳*All Programs-*↳*XAMPP-*↳*XAMPP Control Panel. Untuk menguji instalasi XAMPP ini, silahkan klik tombol Start pada modul apache dan MySQL. Jika tidak ada masalah maka akan tampil warna hijau pada bagian modul ini.

Selanjutnya buka web browser dan ketikkan localhost pada address bar kemudian enter maka akan muncul dengan otomatis localhost/dashboard semuanya telah terinstall dengan baik.

Jika ingin melihat versi PHP kita secara mendalam silahkan klik PHPInfo di pojok kanan atas. Disini kita dapat melihat bahwa PHP yang kita pakai sudah versi 7.3.4.



Gambar 5.10 Selesai Install



Gambar 5.11 XAMPP Control Panel

5.1.4 Membuat Database

Secara umum, tipe website bisa dibedakan menjadi dua yaitu web statis dan web dinamis. Web statis adalah web yang tetap dalam arti tampilan, navigasi, dan konten tidak dapat berubah dengan otomatis. Ketika kita ingin mengupdate sebuah kegiatan akan tetapi kita harus membuka file yang aslinya. Umumnya kegiatan yang ditampilkan tetap untuk jangka waktu satu hari - satu hari. Tipe website ini biasanya hanya berupa tag html saja, jadi diperlukan database yang digunakan untuk menyimpan data.

Selain memanfaatkan tag html, website yang menggunakan flash juga bisa dikategorikan sebagai web statis, meskipun ada sebagian kecil yang sudah memiliki database



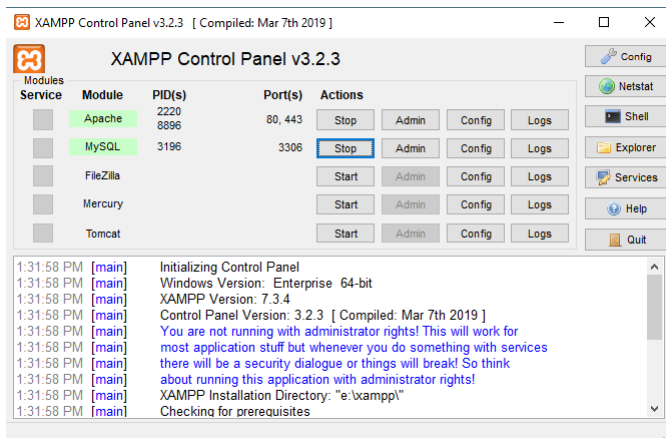
Gambar 5.12 Localhost Dashboard

PHP Version 7.3.4	
System	Windows NT DESKTOP-86L09BN 10.0 build 17134 (Windows 10) AMD64
Build Date	Apr 2 2019 21:50:45
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cmdscript /nologo configure.js "--enable-snapshot-build"--enable-debug-pack"--with-pdo-oci=c:\php-snap-build\deps_aux\oracle64\instantclient_12_1\sdk,shared"--with-oci8-12c=c:\php-snap-build\deps_aux\oracle64\instantclient_12_1\sdk,shared"--enable-object-out-dir=.obj\ "--enable-com-dotnet-shared"--without-analyzer"--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	E:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731.TS.VC15
PHP Extension Build	API20180731.TS.VC15
Debug Build	no
Thread Safety	enabled

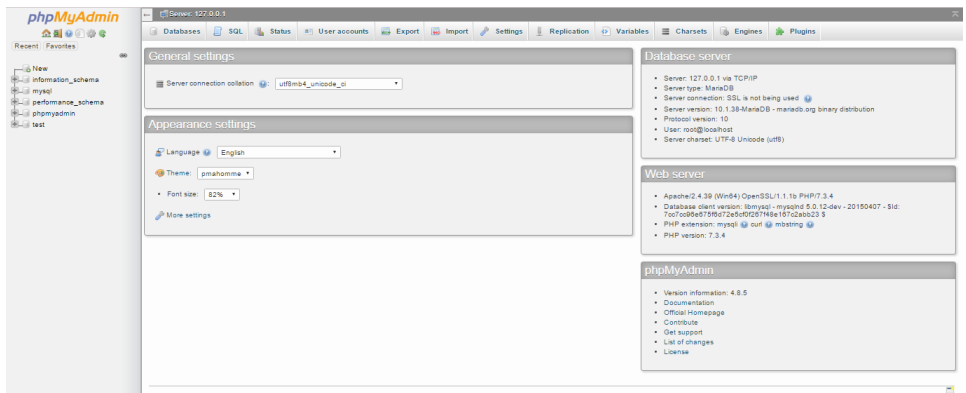
Gambar 5.13 Versi PHP

dalam mengelola konten tidak perlu membuka sebuah file tertentu namun hanya menambahkan pada form yang telah disiapkan dan tersimpan di database. Sama halnya dengan e-logbook ini harus ada database yang dibuat.

1. Untuk membuat database kita pertama kali harus membuka aplikasi XAMPP yang sudah terinstall dan klik start pada Apache serta MySQL.
2. Jika sudah berjalan maka selanjutnya buka web browser kesayangan kita lalu ketikkan <https://localhost/phpmyadmin>
3. Setelah itu, klik databases lalu ketikkan elban lalu klik create
4. Setelah database dibuat lalu belum ada table, klik SQL lalu masukan codingan berikut



Gambar 5.14 XAMPP Control Panel

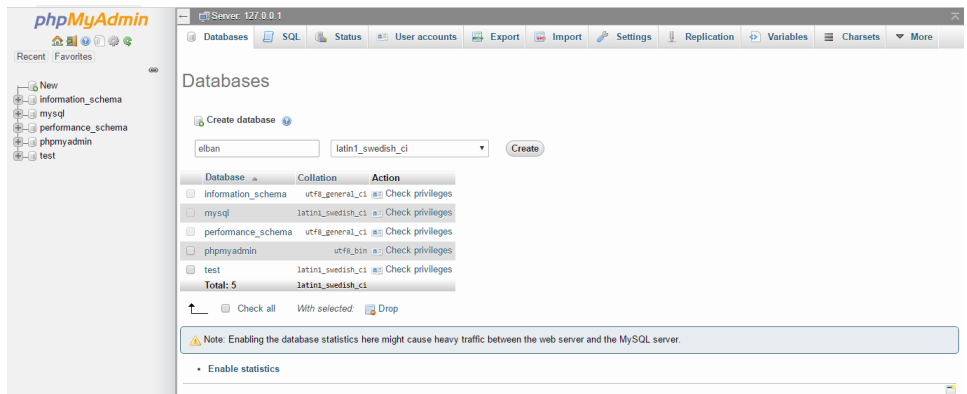


Gambar 5.15 Halaman Utama

```

1 CREATE TABLE IF NOT EXISTS 'logbook' (
2   'id_logbook' int(10) NOT NULL,
3   'tanggal' date NOT NULL,
4   'petugas' varchar(100) NOT NULL,
5   'vendor' varchar(100) NOT NULL,
6   'penerbangan' varchar(100) NOT NULL,
7   'kegiatan' varchar(500) NOT NULL,
8   'point_kerusakan' varchar(200) NOT NULL,
9   'target_pekerjaan' varchar(200) NOT NULL,
10  'id_user' tinyint(1) NOT NULL
11 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
12
13
14 Dumping data for table 'logbook'
15

```

Gambar 5.16 Create Database

```

16
17 INSERT INTO 'logbook' ('id_logbook', 'tanggal', 'petugas', '
    vendor', 'penerbangan', 'kegiatan', 'point-kerusakan', '
    target-pekerjaan', 'id-user') VALUES
18 (2, '2019-03-31', 'M. Arif S, Vania & Rismayadi', 'Anto (CUPPS)
    Hisyam', 'Citylink KJT – KNO', 'PM di SCP 2 Inter Line 1', '
    Master Clok Gate 4, XRAY BHS Internasional', 'Master Clock
    Gate 4', 12),
19 (3, '2019-03-28', 'M. Arif S & Vania', 'Anto (CUPPS) Hisyam', '
    KJT – KNO (Cancelled)', 'PM', 'Master clock gate 4 off', '
    Master CloCk gate 4', 12);
20
21
22
23
24 Table structure for table 'pegawai'
25
26
27 CREATE TABLE IF NOT EXISTS 'pegawai' (
28   'id-pegawai' int(10) NOT NULL,
29   'pegawai' varchar(100) NOT NULL
30 ) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=latin1;
31
32
33 Dumping data for table 'pegawai'
34
35
36 INSERT INTO 'pegawai' ('id-pegawai', 'pegawai') VALUES
37 (1, 'M. Arif S'),
38 (2, 'M. Arif S & Vania'),
39 (3, 'M. Arif S, Vania & Hafiz'),
40 (4, 'M. Arif S, Vania & Rismayadi'),
41 (5, 'M. Arif S & Hafiz'),
42 (6, 'M. Arif S & Rismayadi'),
43 (7, 'M. Arif S, Vania, Hafiz & Rismayadi'),

```

```

44 (8, 'Vania'),
45 (9, 'Vania & Hafiz'),
46 (10, 'Vania & Rismayadi'),
47 (11, 'Vania, Hafiz & Rismayadi'),
48 (12, 'Hafiz'),
49 (13, 'Hafiz & Rismayadi'),
50 (14, 'Rismayadi'),
51 (15, 'M. Arif S, Hafiz & Rismayadi');
52
53 —————
54
55 —
56 — Table structure for table 'user '
57 —
58
59 CREATE TABLE IF NOT EXISTS 'user ' (
60     'id_user' tinyint(2) NOT NULL,
61     'username' varchar(30) NOT NULL,
62     'password' varchar(35) NOT NULL,
63     'nama' varchar(50) NOT NULL,
64     'alamat' varchar(150) NOT NULL,
65     'hp' varchar(20) NOT NULL,
66     'level' tinyint(1) NOT NULL
67 ) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;
68
69 —
70 — Dumping data for table 'user '
71 —
72
73 INSERT INTO 'user ' ('id_user', 'username', 'password', 'nama', '
74     alamat', 'hp', 'level') VALUES
75 (1, 'admin', '21232f297a57a5a743894a0e4a801fc3', 'Luqman Nurfajri
76     ', 'Ciwarugotham', '089634530333', 1),
77 (13, 'vania', '081c2ce8528c443cc4be69d4096c9778', 'Vania R', '
78     Kertajati', '-', 1);
79
80 —
81 —
82 — Indexes for dumped tables
83 —
84 —
85 — Indexes for table 'logbook '
86 —
87 ALTER TABLE 'logbook '
88     ADD PRIMARY KEY ('id_logbook ');
89
90 —
91 — Indexes for table 'pegawai '
92 —
93 ALTER TABLE 'pegawai '
94     ADD PRIMARY KEY ('id_pegawai ');
95
96 —
97 ALTER TABLE 'user '

```

```
97  ADD PRIMARY KEY ('id_user');
98
99  —
100  — AUTO_INCREMENT for dumped tables
101  —
102
103  —
104  — AUTO_INCREMENT for table 'logbook'
105  —
106  ALTER TABLE 'logbook'
107  MODIFY 'id_logbook' int(10) NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=4;
108
109  — AUTO_INCREMENT for table 'pegawai'
110  —
111  ALTER TABLE 'pegawai'
112  MODIFY 'id_pegawai' int(10) NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=16;
113
114  — AUTO_INCREMENT for table 'user'
115  —
116  ALTER TABLE 'user'
117  MODIFY 'id_user' tinyint(2) NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=14;
```

5.2 Struktur Dasar PHP

Pada awalnya PHP adalah kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada saat itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilis kode sumber ini menjadi sumber yang terbuka, maka banyak pemrogram yang ikut tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis lah PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, ada sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut dapat merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hyper-text Preprocessing. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke

arah paradigma berorientasi objek. Server web bawaan ditambahkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall software server. Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 2017.

5.2.1 Variabel

Variabel ini digunakan untuk menyimpan sebuah nilai, data atau informasi.

1. Nama dari variabel diawali dengan tanda dollar \$.
2. Panjang dari variabel tidak terbatas.
3. Setelah tanda \$.
4. Tidak perlu untuk dideklarasikan atau compile.
5. Tidak boleh mengandung spasi.

Contoh:

```
1 <?php
2 $npm="1154054";
3 $nama=' Luqman Nurfajri';
4 echo"npm : ".npm . "<br>";
5 echo"nama : &nama";
6 ?>
```

5.2.2 Tipe Data

Pada PHP, tipe data dari variabel tergantung kondisi yang dialami oleh programmer itu sendiri. Akan tetapi secara otomatis dapat ditentukan oleh PHP. PHP telah mendukung 8 buah tipe data primitif yaitu:

1. Boolean.
2. Integer.
3. Float.
4. String.
5. Array.
6. Object.
7. Resource.
8. NULL.

Contoh:

```

1 <?php
2 $npm = "1154054";
3 $nama = ' Luqman Nurfajri';
4 $umur= 22;
5 $nilai = 98.75;
6 $status = TRUE;
7 echo"NPM : " . $npm . "<br>";
8 echo>Nama: $nama";
9 print "Umur : " . $umur; print "<br>";
10 printf ("Nilai: %.3f<br>"), $nilai);
11 if ($status)
12 echo "Status: Aktif";
13 else
14 echo "Status: Tidak Aktif";
15 ?>

```

5.2.3 Konstanta

Konstanta adalah sebuah variabel konstan yang nilainya tidak berubah-ubah. Untuk mendefinisikan konstanta dalam PHP, menggunakan fungsi define() . Contoh:

```

1 <?php
2 define ("Nama", "Luqman Nurfajri");
3 define ("Nilai", 98.75);
4 echo>Nama : ".Nama .<br>";
5 echo"Nilai: " . Nilai";
6 ?>

```

5.2.4 Struktur Kondisi Dan Perulangan

5.2.4.1 Struktur Kondisi IF Kondisi IF adalah kondisi dimana sebuah data yang apabila kondisinya jika dan hanya nilai kebenaran dari hasil yang dibuat adalah benar, tetapi jika kondisi yah diuji salah maka sistem / program akan tidak menanggapi. Contoh:

```

1 if (bebas) {
2     pernyataan benar
3 }

```

Dari skrip diatas parameter IF ini dapat kita gunakan dalam PHP, buatlah file dengan nama **bebas.php**.

```

1 <html>
2 <head>
3 <title> Test Kondisi IF </title>
4 <body>
5 <?php
6     $bebas ="aih";
7
8     if ($bebas == "aih") {
9         echo "Buku ini semoga bermanfaat";
10    }
11 ?>

```

```

12     </body>
13 </html>

```

5.2.4.2 Struktur Kondisi IF ELSE Kondisi IF ELSE digunakan untuk jika kondisi kita memiliki dua pilihan dari hasil yang berbeda, contohnya hasil yang keluar bernilai benar (*true*) dan bernilai salah (*false*). Secara standar sintaks seperti ini:

```

1 if (bebas) {
2     statement benar
3 } else {
4     statement salah
5 }

```

Dari sintaks diatas kita dapat menyimpulkan bahwa, apabila **bebas** mendapatkan nilai yang sesuai maka *statement* akan benar maka program yang akan dieksekusi benar dan jika **bebas** mendapatkan nilai salah maka yang dieksekusi adalah *statement* salah. Berikut contoh penggunaan kondisi IF dan ELSE. Buatlah file dengan nama **ifelse.php**:

```

1 <html>
2 <head>
3 <title>Test Kondisi IF dan ELSE </title>
4 </head>
5     <body>
6         <?php
7             $makan = "eat";
8             if ($makan=="eat")
9                 echo "Makan adalah bahasa indonesia dari eat";
10            else {
11                echo "Makan bukan bahasa indonesia dari EAt";
12            }
13        ?>
14    </body>
15 </html>

```

5.2.4.3 Struktur Kondisi Switch Dan Case Struktur kondisi SWITCH DAN CASE digunakan saat penyelesaian dari persoalan dengan jumlah kondisi yang banyak. Struktur ini dapat memeriksa nilai suatu variabel dengan SWITCH dan memeriksa kondisi dengan CASE. Contoh:

```

1 switch ($var) {
2 case '1' : statement -1; break;
3 case '2' : statement -2; break;
4 ....
5 }

```

Berikut contoh penggunaan kondisi SWITCH dan CASE. Buatlah file dengan nama **swicase.php**:

```

1 <?php
2 $day =date ("D");
3 switch ($day) {
4     case 'Sun' : $hari= "Minggu" ; break;
5     case 'Mon' : $hari= "Senin" ; break;

```

```

6     case 'Tue' : $hari= "Selasa" ; break;
7     case 'Wed' : $hari= "Rabu" ; break;
8     case 'Thu' : $hari= "Kamis" ; break;
9     case 'Fri' : $hari= "Jum'at" ; break;
10    case 'Sat' : $hari= "Sabtu" ; break;
11    default: $hari = "Kiamat" ;
12 }
13 echo "Hari ini hari <b>$hari </b>";
14 ?>

```

5.2.4.4 Struktur Perulangan For Struktur perulangan digunakan dalam kondisi membatasi perulangan. sebagai contoh disini kita mengulang kalimat "Semoga buku ini bermanfaat" sebanyak 50 kali. Buatlah file dengan nama **for.php**

```

1 <?php
2 for ( $i= 1; $i <= 50; $i++)
3 {
4     echo "Semoga buku ini bermanfaat";
5     echo "<br />";
6 }
7 ?>

```

5.2.4.5 Struktur Perulangan While Struktur perulangan while digunakan pada saat banyaknya perulangan tidak dapat kita pastikan. Disini kita mengulang angka 1 sampai 14 sebagai contoh: Buatlah file dengan nama **while.php**

```

1 <?php
2 $i=1;
3 while ( $i <= 14)
4 {
5     echo "$i";
6     echo "<br />";
7     $i=$i+1;
8 }
9 ?>

```

5.2.4.6 Struktur Perulangan Do While Struktur Do While sebenarnya lanjutan dari perulangan While, perbedaan keduanya dilihat dari posisi pengecekan kondisi. Apabila perulangan While kondisi yang dicek di awal maka perulangan Do While di akhir perulangan. Disini kita mengulang angka 1 sampai 14 sebagai contoh: Buatlah file dengan nama **dowhile.php**

```

1 <?php
2 $i=1;
3 do
4 {
5     echo "$i";
6     echo "<br />";
7     $i=$i+1;
8 } while ( $i <= 10);
9 ?>

```

5.2.4.7 Struktur Perulangan Foreach Array adalah sebuah tipe data yang sering digunakan dalam membuat program menggunakan PHP. Kemampuan array dalam menyimpan banyak data dalam satu variabel akan sangat berguna untuk menyederhanakan dan menghemat penggunaan variabel. Perulangan Foreach adalah perulangan khusus untuk membaca nilai dari array. Buatlah file dengan nama **foreach.php**

```

1 <?php
2 $manusia = array("Luqman","Fajri","Ahmad","Fahmi","Mister");
3
4 foreach ($manusia as $val)
5 {
6     echo "$value";
7     echo "<br />";
8 }
9 ?>

```

5.2.4.8 Struktur Break Dan Continue Struktur **BREAK** dan **CONTINUE** sering digunakan dalam berbagai pekerjaan. Kedua struktur tersebut digunakan untuk mengatur bagaimana jalan dari pengulangan. Struktur **Break** digunakan untuk menghentikan jalan dari pengulangan sedangkan **continue** digunakan untuk melanjutkan ke langkah selanjutnya tanpa menjalankan sisa perintah di dalam skrip pengulangan. Buatlah file dengan nama **breconti.php**

```

1 <?php
2
3 for ($i=1; $i <10 ; $i++) {
4     if ($i == 5)
5         continue;
6     if ($i == 8)
7         break;
8     echo "$i ";
9 }
10
11 ?>

```

Jadi dari skrip diatas dapat disimpulkan bahwa perintah **continue** akan melanjutkan proses pengulangan dan perintah **break** akan menghentikan proses. Dalam proses keduanya maka tidak akan muncul angka 5 dan 8 dalam proses tersebut.

5.2.5 Penanganan Form

Form dalam dunia pemrograman web sudah biasa ditulis menggunakan tag-tag HTML. Untuk halaman form yang berisi tag HTML atau tidak ada skrip lain. Ada tiga komponen penting dalam penanganan form yaitu:

1. Method dalam sebuah form bertanggung jawab untuk dapat menentukan bagaimana data input yang akan di kirim. Ada dua macam method dalam penanganan form ini. Method POST dan GET. Buatlah file dengan nama **get.php**.

```

1 <html>
2 <body>
3     <form method="GET" action="">

```



```

4   <input type="text" name="nama"><br>
5   <input type="text" name="email"><br>
6   <input type="submit" name="submit" value="Submit">
7   </form>
8 </body>
9 </html>

```

Buatlah file dengan nama **post.php**.

```

1 <html>
2 <body>
3   <form method="POST" action="">
4     <input type="text" name="nama"><br>
5     <input type="text" name="email"><br>
6     <input type="submit" name="submit" value="submit">
7   </form>
8
9   <?php
10  if ( $_POST )
11  {
12    echo 'Nama: ' . $_POST[ 'nama' ];
13    echo '<br>';
14    echo 'Email: ' . $_POST[ 'email' ];
15  }
16  ?>
17 </body>
18 </html>

```

2. Action dalam sebuah form bertanggung jawab untuk menentukan dimana data akan diolah. Biasanya action di dalam PHP digunakan untuk mengolah inputan yang diberikan. Jika action dikosongkan dapat dipastikan halaman yang sama pada prosesnya.
3. Submit bertugas sebagai penanda pengiriman data dari form input yang diberikan. Jika tombol submit ditekan maka data dari form input akan dikirim kemudian diproses oleh atribut action yang digunakan.

5.2.6 Array Dan Fungsi

Array mempunyai variabel khusus yang memiliki beberapa nilai sedangkan variabelnya hanya satu yaitu **array** akan tetapi nilainya dapat banyak. Variabel array hanya satu karena variabel tersebut mempunyai indeks. Indeks tersebut dapat berupa angka atau teks. Indeks tersebut dapat secara otomatis diisi oleh program dan bisa juga ditentukan oleh programmer. Buatlah file dengan nama **array1.php**.

```

1 <?php
2 $makna = array( "Luqman", "Galih", "Akbar" )
3 $kawan[1] = "Luqman";
4 $kawan[2] = "Galih";
5 $kawan[3] = "Akbar";
6 ?>

```

Untuk mengakses array yaitu dengan cara menuliskan indeksinya setelah variabel. Buatlah file dengan nama **array2.php**.

```

1 $makna = array("Bagus", "Didi", "Asep", "Ichsan", "Tita");
2 echo $makna[3]; // akan menampilkan Ichsan

```

5.2.6.1 Array Asosiatif Array asosiatif adalah array tetapi indeksnya bukan berupa angka namun berupa teks. Buatlah file dengan nama **arrayaso.php**.

```

1 <?php
2 $data = array ("nama"=>"Luqman", "jenis_kelamin"=>"Pria", "umur"=>22)
3 ;
4 echo $data['nama'] . '<br/>';
5 echo $data['jenis_kelamin'] . '<br/>';
6 echo $data['umur'] . '<br/>';
7 ?>

```

5.2.6.2 Array Multidimensi Array multidimensi adalah array yang ada dalam array itu sendiri. Dalam array tersebut dapat berisi ditambahkan beberapa array lagi. Array multidimensi ini dapat memudahkan membuat sebuah program dikarenakan bisa membuat beberapa array sekaligus sehingga memangkas beberapa perintah operasi. Buatlah file dengan nama **arraymulti.php**.

```

1 <?php
2 $data = array
3 ( array ( "nama_kawan"=>"Luqman", "jenis_kelamin"=>"Pria", "umur"
4 =>22),
5 array ( "nama_kawan"=>"Ahmad", "jenis_kelamin"=>"Pria", "umur"
6 =>22),
7 array ( "nama_kawan"=>"Bung", "jenis_kelamin"=>"Pria", "umur"=>25)
8 );
9 for ($i=0; $i<sizeof($data); $i++)
10 {
11     echo $data[$i]['nama'] . '<br/>';
12     echo $data[$i]['jenis_kelamin'] . '<br/>';
13     echo $data[$i]['umur'] . '<br/>';
14     echo '<br/>';
15 }
16 ?>

```

Fungsi-fungsi Array dalam PHP terdapat banyak bahkan lebih dari 70 untuk menggunakan memanipulasi array. Untuk melihat fungsi array apa saja yang tersedia dapat dilihat di website <https://www.php.net/manual/en/ref.array.php>.

5.2.7 Fungsi

Disini kita akan mencoba beberapa fungsi yang dapat dipakai.

5.2.7.1 Menghitung Panjang String Untuk dapat menghitung jumlah karakter dalam sebuah string kita dapat menggunakan fungsi **strlen**. Buatlah file dengan nama **hitung.php**.

```

1 <?php
2 $jumlah_kata = 'Test';
3 echo 'Jumlah huruf <b>' . $jumlah_kata . '</b> adalah ' . strlen(
4     $jumlah_kata);
5 ?>

```

5.2.7.2 Include Dan Require Untuk dapat menggunakan fungsi serta data yang berulang-ulang PHP memberikan sebuah kemudahan yang dapat meringkas pekerjaan dan meminimalkan kesalahan dalam membuat sebuah program. Penulisan program dalam PHP dapat mengambil data atau perintah yang ada pada file lain yang sama dengan perintah operasinya. Buatlah file dengan nama **teks.php**.

```
1 <?php
2 $teks = 'Masih kosong "teks.php" <br />';
3 ?>
```

Selanjutnya kita buat file dengan nama **include.php**.

```
1 <?php
2 include 'teks.php';
3 echo $teks;
4 echo 'hehe "include.php"';
5 ?>
```

5.2.7.3 Memecah String Untuk dapat memecah string atau memisahkan string dapat menggunakan perintah **explode()** dengan sintaks sebagai berikut: **explode(pola, string, batas)**.

1. **pola** adalah karakter yang akan dijadikan pemisah
2. **string** adalah sebuah data yang akan diproses
3. **batas** adalah jumlah pecahan yang akan diambil

Buatlah file dengan nama **include.php**.

```
1 <?php
2 $data = 'Luqman, Pria, l.nurfajri@gmail, 023122222';
3 $dipecah = explode(',', $data, 2); // 2 adalah jumlah pecahan yang akan
    diproses
4 for ($i=0; $i<=3; $i++)
5 {
6     echo "Pecahan $i = " . $dipecah[$i] . "<br>";
7 };
8 ?>
```

5.2.7.4 Mengubah Kapitalisasi Huruf Disaat kita membutuhkan untuk mengubah kapitalisasi huruf menjadi huruf besar dan kecil semua atau kecil di awal kalimat, maka PHP mempunyai fungsi-fungsi tersebut yang kita tinggal gunakan yaitu: **strtoupper()**, **strtolower()**, **ucfirst()**, dan **ucwords()**.

1. **strtoupper()** adalah fungsi yang digunakan untuk mengubah huruf kecil ke huruf kapital
2. **strtolower()** adalah fungsi yang digunakan untuk mengubah huruf kapital ke huruf kecil
3. **ucfirst()** adalah untuk mengubah menjadi huruf kapital pada awal kalimat

4. **ucwords()** adalah fungsi yang digunakan untuk mengubah huruf kecil menjadi kapital pada setiap awal kata

Buatlah file dengan nama **str.php**.

```

1 <?php
2 $teks = 'Ini kalimat yang diproses oleh fungsi-fungsi kapitalisasi
    string. Ini juga kalimat lainnya. Silakan lihat hasil yang
    ditampilkan.';
3 echo '<p>u>Kalimat asli:</u> <br>';
4 echo "&$teks </p>";
5 echo '<p>u>Diubah menjadi huruf kapital semua:</u> <br>';
6 echo strtoupper("$teks </p>");
7 echo '<p>u>Diubah menjadi huruf kecil semua:</u> <br>';
8 echo strtolower("$teks </p>");
9 echo '<p>u>Diubah menjadi huruf kapital di awal kalimat:</u> <br>';
10 echo ucfirst("$teks </p>"); echo '<p>u>Diubah menjadi huruf kapital di
    setiap awal kata:</u> <br>';
11 echo ucwords("$teks </p>");
12 ?>

```

5.2.8 String Dan Tanggal

String adalah kumpulan dari karakter dalam PHP. Karakter dalam PHP ada 256. PHP tidak mendukung native unicode. Untuk dapat menuliskan sebuah string dalam PHP, kita dapat menggunakan 3 cara, yaitu:

1. Kutip tunggal (')
2. Kutip Ganda (")
3. Heredoc sintaks atau suatu cara untuk menulis blok besar teks dalam PHP
4. Nowdoc sintaks (semenjak PHP 5.3.0)

Pada PHP 7.0.0, tidak ada batasan khusus mengenai panjang string pada build 64-bit. Pada build 32-bit dan dalam versi sebelumnya, sebuah string dapat berukuran hingga 2GB (maksimum 2147483647 bytes). Cara termudah untuk menentukan string adalah dengan melampirkannya dalam tanda kutip tunggal ('). Buatlah file dengan nama **tunggal.php**.

```

1 <?php
2 echo 'ini adalah string sederhana';
3
4 echo 'kita juga dapat menempelkan baris baru dalam string dengan cara
    ini karena tidak apa-apa untuk dilakukan';
5
6
7 echo 'Luqman pernah berkata: "Aku akan kembali"';
8
9 // Outputs: Kita menghapus C:\*..*?
10 echo 'Kita menghapus C:\*..*?';
11

```

```

12 // Outputs: Kita menghapus C:\*.*?
13 echo 'Kita menghapus C:\*.*?';
14
15 // Outputs: Ini tidak akan berubah: \n sebuah baris baru
16 echo 'Ini tidak akan berubah: \n sebuah baris baru';
17
18 // Outputs: Bukan variabel $expand $either
19 echo 'Bukan variabel $expand $either';
20 ?>

```

Jika string dimasukkan dalam tanda kutip ganda ("), PHP akan menafsirkan urutan berikut untuk membentuk karakter khusus:

```

1 <?php
2 $nama = Luqman ;
3
4 echo nama saya $nama ;
5 ?>

```

Cara ketiga untuk membatasi string adalah sintaks **heredoc**. Setelah operator ini, pengidentifikasi yang disediakan, kemudian baris baru. String itu sendiri mengikuti, dan kemudian pengidentifikasi yang sama lagi untuk menutup kutipan. Identifier penutup harus dimulai pada kolom pertama dari tiap baris. Selain itu, pengidentifikasi harus mengikuti aturan penamaan yang sama dengan label lainnya di PHP: ini harus hanya berisi karakter alfanumerik dan garis bawah dan harus dimulai dengan karakter non-digit atau garis bawah.

```

1 <?php
2 $str = <<<EOD
3
4 EOD;
5
6 {
7     var $foo;
8     var $bar;
9
10    function __construct()
11    {
12        $this->foo = 'Foo';
13        $this->bar = array('Bar1', 'Bar2', 'Bar3');
14    }
15 }
16
17 $foo = new foo();
18 $name = 'Luqman';
19
20 echo <<<EOT
21 My name is "$name". I am printing some $foo->foo.
22 Now, I am printing some {$foo->bar[1]}.
23 This should print a capital 'A': \x41
24 EOT;
25 ?>

```

Nowdocs adalah untuk string yang dikutip satu kali, dan heredocs adalah string yang dikutip ganda. Nowdoc ditentukan mirip dengan heredoc, tetapi tidak ada

proses yang dilakukan di dalam nowdoc. Konstruk ini ideal untuk mencocokkan kode PHP atau blok teks besar lainnya.

Nowdoc diidentifikasi dengan urutan yang sama dengan yang digunakan untuk heredocs, tetapi pengidentifikasi yang berikut dilampirkan dalam tanda kutip tunggal. 'EOT'. Semua aturan untuk pengidentifikasi heredoc juga berlaku untuk pengidentifikasi nowdoc, terutama yang berkenaan dengan penampilan pengidentifikasi tertutup.

```

1 <?php
2 class foo
3 {
4     public $foo;
5     public $bar;
6
7     function __construct()
8     {
9         $this->foo = 'Foo';
10        $this->bar = array('Bar1', 'Bar2', 'Bar3');
11    }
12 }
13
14 $foo = new foo();
15 $name = 'Luqman';
16
17 echo <<<'EOT'
18 My name is "$name". I am printing some $foo->foo.
19 Now, I am printing some {$foo->bar[1]}.
20 This should not print a capital 'A': \x41
21 EOT;
22 ?>

```

Fungsi pada operasi tanggal PHP terdapat beberapa jenis akan tetapi fungsi yang sering digunakan adalah date(). Fungsi ini dapat menghasilkan tanggal serta waktu sekarang. Beberapa pilihan parameternya dapat dilihat sebagai berikut:

Buatlah file dengan nama **date1.php**.

```

1 <?php
2 echo "<br>". date("d/m/Y H:i:s");
3 echo "<br>";
4 echo "<br>". date("F j, Y, g:i a");
5 echo "<br>";
6 echo "<br>". date("d.m.y");
7 echo "<br>";
8 echo "<br>". date("Ymd");
9 echo "<br>";
10 echo "<br>". date('j-m-y, it is w Day z ');
11 echo "<br>";
12 echo "<br>". date('\i\t \i\s \t\h\e jS \d\a\y. ');
13 echo "<br>";
14 echo "<br>". date("D M j G:i:s T Y");
15 echo "<br>". date("H:i:s");
16 ?>

```

Buatlah file dengan nama **date2.php**.

```

1 <?php

```

Tabel 5.1 Operasi Tanggal PHP

Parameter	Keterangan	Contoh Nilai
Hari		
		d
Tanggal	01 - 31	
D	Tiga digit	Mon - Sun
j	Tanggal tanpa nol	1 - 31
l	Nama hari lengkap	Monday - Sunday
N	Urutan hari	1 - 7
S	Akhiran angka inggris	sr, nd, rd atau th
w	Urutan hari dalam seminggu	0 - 6
z	Urutan hari dalam setahun	1 - 365
Minggu		
W	Urutan minggu dalam setahun	Contoh: 14, minggu ke 14 dalam tahun ini
Bulan		
F	Nama bulan lengkap	January - December
m	Urutan bulan	01 - 12
M	Tiga digit nama bulan	Jan - Dec
n	Urutan bulan dalam setahun	1 - 12
t	Jumlah hari dalam bulan	14 - 31
Tahun		
Y	4 digit tahun	1997
y	2 digit tahun	97
Waktu		
a	Lowercase	am atau pm
A	Uppercase	AM atau PM
g	Jam format 12	1 - 12
G	Jam format 24	0 - 23
h	Jam format 12	01 - 12
H	Jam format 24	00 - 23
i	Menit	00 - 59
s	Detik	00-59

```

2 // menunjukan tanggal tanggal sekarang
3 $arrDay = array("Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jum'
  at", "Sabtu");
4 $day = date ("w"); //0 - 6 of day
5 echo "Hari ini hari : <b>" . $arrDay[$day]. "</b>";
6 ?>

```

5.2.9 File Dan Direktori

Dalam manajemen file dan direktori dalam PHP terdapat lebih dari 70 fungsi. Beberapa fungsi utama seperti (create, write, append, dan delete), yaitu: Membuka dan Membuat File.

fopen(\$namafile, \$mode);

\$namafile adalah nama file yang ingin kita buat, sedangkan \$mode adalah mode akses file. Mode akses file terdiri dari:

r = Hanya untuk membaca file, pointer berada di awal file.

r+ = Dapat baca dan tulis file, pointer berada di awal file.

w = Hanya untuk tulis file, isi file lama dihapus, jika file belum ada maka akan di-create.

w+ = Untuk baca dan tulis file, isi file lama dihapus, jika file belum ada maka akan di-create.

a = Hanya untuk menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create.

a+ = Untuk membaca dan menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create.

Buatlah file dengan nama **file1.php**

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile , "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>File berhasil dibuka</b>";
8 }
9 fclose ($handle);
10 ?>
```

Buatlah file dengan nama **file2.php**

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile , "w");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>File berhasil dibuka</b>";
8 }
9 fclose ($handle);
10 ?>
```

Buatlah file dengan nama **file3.php**

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile , "w");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     fwrite ($handle , "Fakultas Teknologi Informasi\n");
8     fputs ($handle , "Universitas Budi Luhur\n");
9 }
```



```

9 // file_put_contents ($namafile , "Jakarta");
10 echo "<b>File berhasil ditulis </b>";
11 }
12 fclose($handle);
13 ?>

```

Buatlah file dengan nama **file4.php**

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile , "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     $isi = fgets ($handle , 2048);
8     // $isi2 = fread ($handle , 20);
9     echo "Isi 1 : $isi<br>";
10    //echo "Isi 2 : $isi2<br>";
11 }
12 fclose($handle);
13 ?>

```

Buatlah file dengan nama **file5.php**

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile , "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>Isi file : </b><br>";
8     while ($isi = fgets ($handle , 2048)) {
9         echo "$isi<br>";
10    }
11 }
12 fclose($handle);
13 ?>

```

Buatlah file dengan nama **file6.php**

```

1 <?php
2 $namafile = "data.txt";
3 $handle = @fopen($namafile , "r");
4 if ($handle) {
5     while (!feof($handle)) {
6         $buffer = fgets($handle , 4096);
7         echo $buffer."<br>";
8     }
9     fclose($handle);
10 }
11 ?>

```

Buatlah file dengan nama **file7.php**

```

1 <?php
2 $counter_file="counter.txt";
3 if (!file_exists ($counter_file)) {
4     fopen ($counter_file , "w");
5 }

```

```
6 $file = fopen($counter_file,"r");
7 $counter = fread($file,10);
8 fclose($file);
9 $counter++;
10 echo "<h2>Anda adalah pengunjung ke - $counter </h2>";
11 $file = fopen($counter_file,"w");
12 fwrite($file,$counter);
13 fclose($file);
14 ?>
```

