

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	13
4	Array	39
5	SQL	45
6	Object Oriented Programming	51

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 Bahasa Pemrograman	2
1.2.1 Kelebihan PHP	2
1.2.2 Penulisan Skript	3
	ix

1.2.3	PHP 7	3
1.3	Web Server	4
2	Judul Bagian Kedua	5
2.1	Pemasangan PHP di Server Web Windows	5
2.1.1	Cara Install XAMPP	6
2.1.2	Menguji Instalasi XAMPP	9
2.2	Localhost	10
3	Dasar Penulisan PHP Sederhana	13
3.1	Aturan Dasar Penulisan Kode PHP	13
3.1.1	Case Sensitivity (perbedaan huruf besar dan kecil) dalam PHP	13
3.1.2	Karakter Spasi dan Tab dalam PHP	14
3.2	Embedded Script dan Non Embedded	14
3.2.1	Embedded Script	14
3.2.2	Non Embedded	15
3.3	Variabel dan Tipe Data	16
3.3.1	Variabel	16
3.3.2	Tipe Data	16
3.3.3	Tipe Data Integer	17
3.3.4	Tipe Data Floting Point	17
3.3.5	Tipe Data String	18
3.3.6	Tipe Data Boolean	18
3.3.7	Tipe Data Objek	18
3.4	Operator PHP	19
3.4.1	Operator Perbandingan	19
3.4.2	Operator PHP Increment dan Decrement	19
3.4.3	Perbedaan Pre Increment dan Post Increment	19
3.4.4	Operator Assignment PHP	19
3.5	Lingkup Variabel	20
3.5.1	Variabel Global	20
3.5.2	Variabel Lokal	21
3.5.3	Variabel Statik	22
3.6	Struktur Kontrol	22
3.6.1	If Statement	22
3.6.2	Perulangan	23
3.6.3	Fungsi	26

3.7	Perulangan	26
3.8	Penanganan Form	26
3.9	String Dan Tanggal	27
3.10	File Dan Direktori	31
3.10.1	File Dan Direktori PHP	33
3.11	Koneksi Dengan Database MySQL	36
3.12	Cookie dan Session	37
4	Array	39
4.1	Array	39
4.1.1	Array Berindeks	40
4.1.2	Array Asosiatif	40
4.2	Membuat Array	41
4.2.1	Mengakses Elemen Array	41
4.2.2	Fungsi-fungsi yang berhubungan dengan array	42
5	SQL	45
5.1	SQL	45
5.1.1	Membuat Database	46
5.1.2	Menampilkan Database	47
5.1.3	Membuka Database	47
5.1.4	Menghapus Database	49
5.1.5	Membuat Tabel	49
5.1.6	Melihat Tabel	50
6	Object Oriented Programming	51
6.1	Aplikasi E-Logbook	51
6.1.1	Merancang Database	52
6.2	Pemrograman Berorientasi Objek Dalam PHP	52
6.2.1	Database	56
6.2.2	Mengaktifkan MySQLi	57
6.2.3	Membuat Database	57

DAFTAR GAMBAR

1.1	Alur Web Server	4
2.1	Versi Terbaru XAMPP	6
2.2	User Account Control	6
2.3	Jendela Awal	7
2.4	Select Component	7
2.5	Installation Folder	8
2.6	Bitnami for XAMPP	8
2.7	Ready to Install	9
2.8	Proses Penginstallan	9
2.9	Windows Security Alert	10
2.10	Selesai Install	10
2.11	XAMPP Control Panel	11
2.12	Localhost Dashboard	11

2.13	Versi PHP	12
3.1	Sistem Bilangan	17
4.1	Fungsi-fungsi yang berhubungan dengan array	43
5.1		46
5.2		46
5.3		46
5.4		47
5.5		47
5.6		47
5.7		48
5.8		48
5.9		49
5.10		50
5.11		50
6.1	XAMPP Control Panel	58
6.2	Halaman Utama	58
6.3	Create Database	59

DAFTAR TABEL

3.1	Operasi Tanggal PHP	30
4.1	(40
4.2	(40
6.1	Tabel Logbook	52
6.2	Tabel Pegawai	52
6.3	Tabel User	53

Listings

src/embedded_script.php	14
src/embedded_script.php	15
src/embedded_script.php	15
src/embedded_script.php	15
src/tag_awal_akhir.php	16
src/tipeDataObjek.php	18
src/lingkupVariabel.php	20
src/lingkupVariabel.php	21
src/lingkupVariabel.php	21
src/lingkupVariabel.php	22
src/array.php	41
src/akses_array.php	42

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[?].

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

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. Dalam pembangunan web terbagi menjadi dua akan tetapi saling berkaitan yaitu: **Bahasa Pemrograman** dan **Web Server**.

1.2 Bahasa Pemrograman

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'.

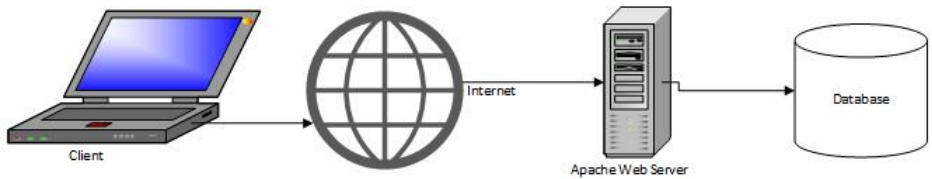
1.2.3 PHP 7

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

1.3 Web Server

Web server merupakan perangkat lunak yang menjadi bagian dari world wide web (www). Web server menunggu permintaan dari client dengan menggunakan sebuah browser seperti UC Browser, Google Chrome, Mozilla Firefox, dan browser lainnya. Jika ada permintaan melalui browser, maka web server akan memproses dan memberikan hasil proses kepada client. Web server ini bisa disebut sebagai perantara kita untuk mengolah web yang dibangun dan mengatur database.



Gambar 1.1 Alur Web Server

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.

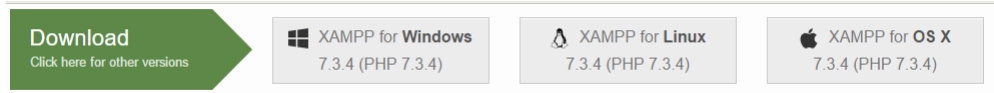
Web server inilah yang akan menerjemahkan kode PHP menjadi HTML dan mengirimnya ke web browser untuk ditampilkan. Kita harus menyewa web server agar kode PHP dapat diproses dan diakses di internet. Namun aplikasi web server ini dapat diinstall di komputer lokal, dan inilah yang akan kita install dalam 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

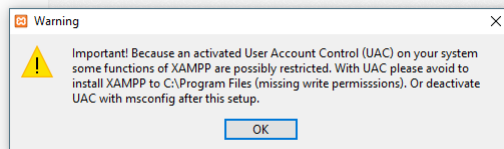
Agar dapat menjalankan sistem yang akan dibuat, kita harus menginstall aplikasi web server yang mendukung PHP ini serta aplikasi untuk database MySQL. Untungnya terdapat banyak aplikasi yang handle program ini, salah satunya yaitu XAMPP. Aplikasi XAMPP adalah aplikasi yang dapat handle banyak aplikasi 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 2.1 Versi Terbaru XAMPP

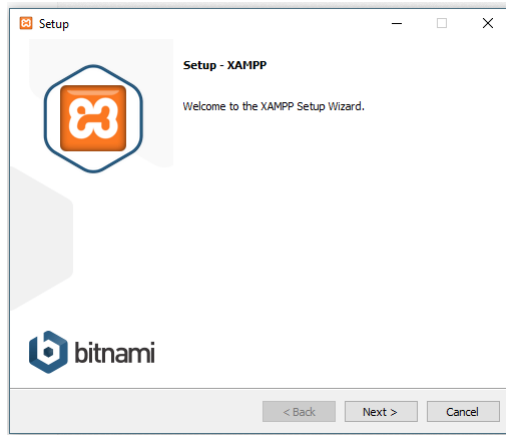
File xampp-windows-x64-7.3.4-0-VC15-installer berukuran cukup besar, sekitar 149MB. Simpanlah file ini dimana kita inginkan. Setelah file XAMPP sudah di download, kita akan menginstallnya dengan cara double klik pada aplikasi tersebut dan akan muncul peringatan sebagai berikut.



Gambar 2.2 User Account Control

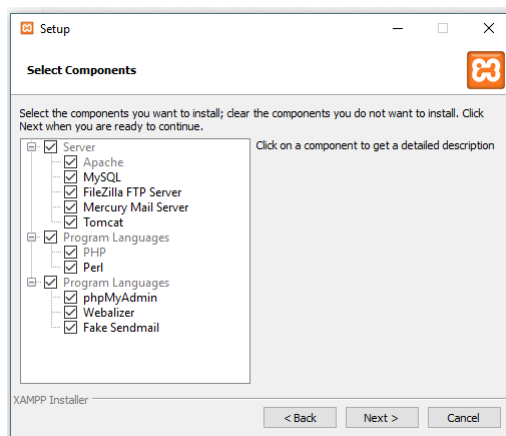
Peringatan ini berkaitan dengan keamanan pada versi Windows Vista keatas dan jika XAMPP akan di install pada folder C mungkin akan terjadi pembatasan hak

akses terhadap XAMPP yang berjalan tidak normal. Silahkan klik tombol OK untuk melanjutkan install maka akan muncul jendela awal install. Silahkan klik next.



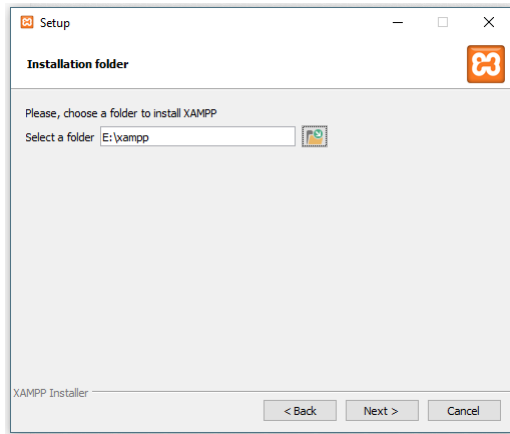
Gambar 2.3 Jendela Awal

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.



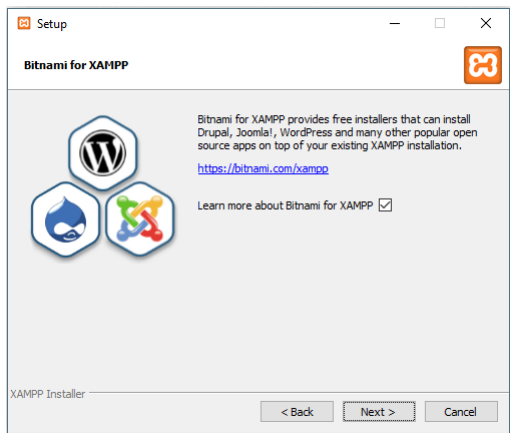
Gambar 2.4 Select Component

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 2.5 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.

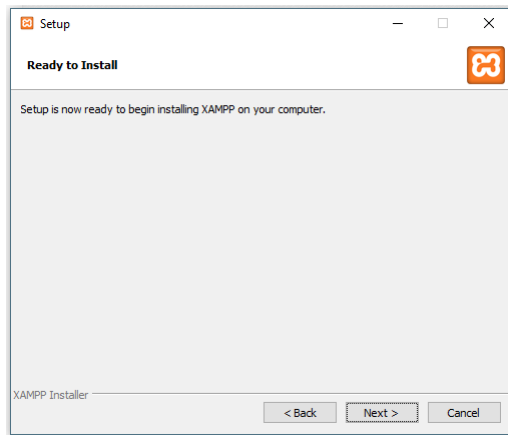


Gambar 2.6 Bitnami for XAMPP

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

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



Gambar 2.7 Ready to Install

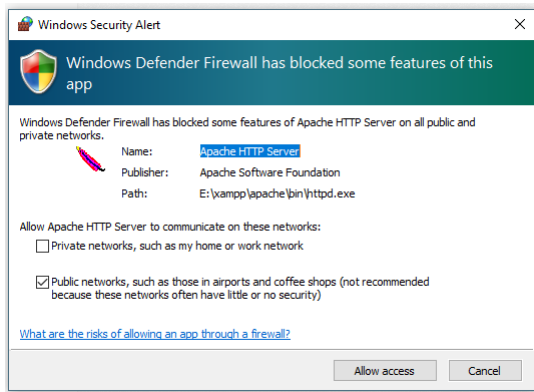


Gambar 2.8 Proses Penginstallan

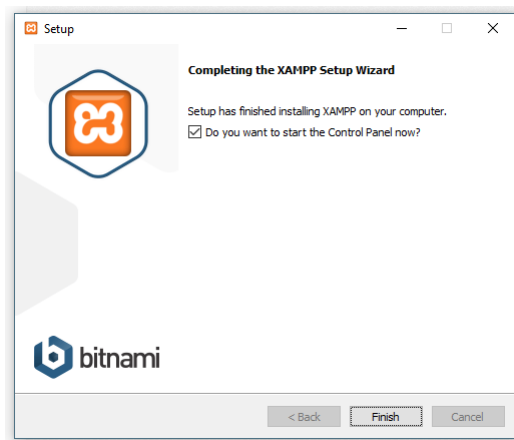
to start the Control Panel now? lalu klik finish. Jika tidak maka hilangkan ceklis dari Do you want to start the Control Panel now? lalu klik finish.

2.1.2 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.



Gambar 2.9 Windows Security Alert



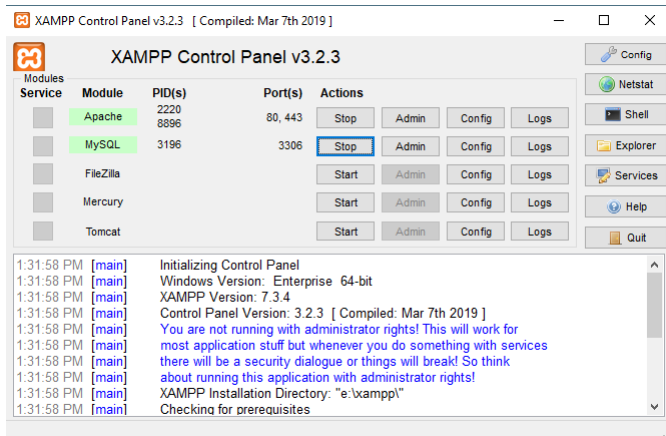
Gambar 2.10 Selesai Install

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.

2.2 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




Gambar 2.11 XAMPP Control Panel



Gambar 2.12 Localhost Dashboard

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.

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	ccrypt holoگو configure ja "--enable-snapshot-build"--enable-debug-pack"--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\v64\instantclient_12_1\edk.shared"--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\v64\instantclient_12_1\edk.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:\vampp\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 2.13 Versi PHP

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. Didalam 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
```

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

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_pesanan() {
9             echo "Cetak Pesanan.";
10        }
11    }
12
13    $test = new objek;
14    $test->do_pesanan();
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: Kondisi IF adalah kondisi dimana sebuah data yang apabila kondisinya jika dan hanya nilai kebenaran dari hasil yang dibuat adalah benar, tetapi jika kondisi yang 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>
```

3.6.2 Perulangan

3.6.2.1 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>
```


3.6.2.2 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 ?>
```

3.6.2.3 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 ?>
```

3.6.2.4 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 ?>
```

3.6.2.5 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 ?>
```

3.6.2.6 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 ?>
```

3.6.2.7 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.

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 );
```

3.7 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 ?>
```

3.8 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.

3.9 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. Konstruksi 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");

```

Tabel 3.1 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

```

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
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 ?>

```

3.10 File Dan Direktori

Dalam manajemen file dan direktori dalam PHP lterdapat 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 ($handle , "Universitas Budi Luhur\n");  
9     //file_put_contents ($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 ?>

```

3.10.1 File Dan Direktori PHP

PHP menyediakan beberapa fungsi yang dapat digunakan agar kita dapat memanipulasi file. Dengan menggunakan PHP kita dapat membuat, menuliskan sesuatu, atau mengedit sebuah file. Kita menggunakan beberapa fungsi dalam pengolahan file dengan PHP untuk dapat membuat sitemap. Jadi kita menambahkan sebuah kategori atau artikel, akan secara otomatis menuliskan url baru pada file sitemap.txt. Tentu saja bukan hanya sitemap yang dapat kita olah, kita dapat menggunakan fungsi-fungsi pengolahan file dalam PHP untuk banyak keperluan.

Dalam menuliskan sesuatu kedalam sebuah file dengan menggunakan PHP, ada 3 fungsi yang perlu diperhatikan. 3 fungsi yang perlu diperhatikan dalam menulis sebuah file dengan PHP yaitu:

1. fopen() - Digunakan untuk membuka file, jika file belum tersedia maka akan membuat file baru.
2. fwrite() - Digunakan untuk menuliskan atau mengedit sesuatu kedalam file.
3. fclose() - Menutup file.

Pada saat kita akan membuka sebuah file dengan menggunakan fungsi fopen(), disini kita harus menentukan tindakan apa yang akan dilakukan terhadap file tersebut. Tindakan yang akan dilakukan terhadap file contohnya, apakah file tersebut hanya akan dibaca, apakah file tersebut akan dibuka dan ditulisi sesuatu, dan lain sebagainya. Tindakan-tindakan terhadap file pada saat kita membuka file tersebut, dituliskan dengan model seperti dibawah ini:

1. r : Membuka file hanya untuk dibaca saja.

2. w : Membuka file hanya untuk ditulis. Isi didalam file tersebut akan dihapus, jika file belum tersedia maka akan menciptakan file baru.
3. a : Membuka file hanya untuk ditulis. Isi didalam file tersebut tidak dihapus, jika file belum tersedia maka akan menciptakan file baru.
4. x : Membuat file baru untuk menulis saja.
5. r+: Membuka file untuk membaca dan juga ditulis.
6. w+: Membuka file untuk membaca dan juga ditulis. Menghapus isi file atau membuat file baru jika tidak ada.
7. a+: Membuka file untuk dibaca dan juga ditulis. Membuat file baru jika file tidak ada
8. x+: Membuat file baru untuk dibaca / ditulis.

Untuk lebih mudah dipahami, mari kita buat contoh kasus dalam mengolah file dengan PHP. Kali ini kita akan mencotohkan cara untuk membuat sebuah file dan menuliskan sesuatu kedalamnya. Dalam contoh ini kita menggunakan model tindakan w.

```
1 <?php
2
3 $filekita = fopen("test.txt", "w");
4 $nama = "Luqman Nurfajri";
5 fwrite($filekita, $nama);
6 fclose($filekita);
7
8 // Hasil didalam test.txt berisi tulisan: Luqman Nurfajri
9
10 ?>
```

Pada contoh mengolah file dengan PHP diatas, kita asumsikan bahwa dokumen PHP diatas berada dalam satu direktori yang sama dengan file yang akan diolah. Maka dapat diuraikan penjelasan sebagai berikut:

1. Disini membuat variabel \$filekita dan didalam variabel tersebut kita membuka sebuah file dengan nama test.txt dan menggunakan model tindakan w. Dengan begitu jika tidak terdapat file dengan nama test.txt maka PHP akan menciptakan file itu. Jika telah terdapat file tersebut, maka isi dari file akan dihapus dan digantikan dengan isi yang akan dituliskan.
2. Variabel \$nama adalah data atau teks yang akan saya tuliskan kedalam file test.txt Kemudian penulisan sintak fwrite(\$filekita, \$nama) adalah proses membuka file dan menuliskan isi kedalamnya.
3. Dengan fungsi fclose(\$filekita) berarti saya telah menutup file yang telah ditulis tersebut.

Memanipulasi Direktori

Direktori adalah lokasi dimana kita menyimpan file pada system yang dapat berisi kumpulan direktori dan direktori. Ada beberapa fungsi yang dapat digunakan di PHP untuk manipulasi direktori:

1. Fungsi `mkdir()` untuk membuat direktori;
2. Fungsi `scandir()` untuk melihat isi direktori;
3. Fungsi `rmdir()` untuk menghapus direktori; `rename()` untuk mengubah nama direktori.

Dari beberapa fungsi diatas ayo kita bahas semuanya.

1. Cara Membuat Direktori di PHP

Fungsi yang digunakan untuk membuat direktori di PHP adalah `mkdir()`. Fungsi ini sama maksudnya dengan perintah `mkdir` di Linux dan `md` pada Windows. Parameter yang diberikan ke fungsi `mkdir()` berupa string. Parameter ini yang akan menjadi nama direktori. Contoh:

```
1 <?php
2 mkdir("contoh_direktori");
3 ?>
```

Atau kita juga bisa membuatnya seperti ini dengan menyebutkan alamat path dan atributnya:

```
1 <?php
2 mkdir("C://xampp/htdocs/test_aja", 1234, true);
3 ?>
```

Keterangan:

1. Parameter 1234 adalah hak akses yang kita berikan kepada direktori;
2. Parameter `true` artinya kita mengizinkan pembuatan direktori secara langsung.

2. Cara Melihat Isi Direktori di PHP

Kali ini fungsi yang akan kita gunakan adalah `scandir()`. Fungsi ini dapat mengembalikan nilai berupa array yang berisi nama-nama dari isi direktori itu sendiri. Contoh:

```
1 <?php
2 $dir = scandir("test_direktori");
3 print_r($dir);
4 ?>
```

Maka variabel `$dir` akan berisi:

```
1 Array
2 (
3     [0] => .
4     [1] => ..
5     [2] => contoh
6 )
```

3. Cara Menghapus Direktori di PHP

Menghapus direktori dalam PHP dapat dilakukan dengan fungsi `rmdir()`. Fungsi ini memiliki parameter berupa string. Parameter tersebut adalah nama direktori yang ingin kita hapus. Contoh:

```
1 <?php
2 rmdir("nama_dir");
3 ?>
```

Fungsi `rmdir()` akan menghasilkan error apabila alamat direktorinya salah atau tidak ditemukan. Untuk mengatasi ini, kita dapat menggunakan fungsi `is_dir()` yang digunakan

```
1 <?php
2 $nama_dir = "test_direktori";
3
4 if( is_dir($nama_dir) ) {
5     rmdir($nama_dir);
6 } else {
7     echo "Direktori tidak ditemukan";
8 }
9 ?>
```

4. Cara Mengubah Nama Direktori di PHP

Kita dapat merubah nama direktori dengan fungsi `rename()`. Fungsi dapat merubah nama direktori dan mengubah nama file juga dapat menggunakan fungsi ini. Contoh:

```
1 <?php
2 rename("test_direktori", " direktori_test");
3 ?>
```

Keterangan:

1. `test_direktori` adalah direktori yang lama;
2. `direktori_test` adalah direktori yang baru.

3.11 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.

3.12 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
3     //memulai session
4     session_start();
5
6     //jika ada session, maka akan diarahkan ke halaman dashboard
7     //admin
8     if(isset($_SESSION['id_user'])){
9
10         //mengarahkan ke halaman dashboard admin
11         header("Location: ./admin.php");
12         die();
13     }
14
15     //mengincludekan koneksi database
16     include "koneksi.php";
17 ?>
```

Session ini sering digunakan untuk:

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

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 Mengakses Elemen Array

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

```

4.2.2 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 indeksnya 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 indeksnya Bentuk umum: bool krsort(array \$array [,int \$sort_flags = SORT_REGULAR])

Gambar 4.1 Fungsi-fungsi yang berhubungan dengan array

BAB 5

SQL

5.1 SQL

SQL (Structured Query Language) adalah bahasa standar untuk berkomunikasi dalam database. Menurut ANSI (American National Standards Institute) adalah bahasa standar yang digunakan untuk sistem manajemen basis data relasional. Pernyataan SQL ini dapat digunakan untuk melakukan perintah-perintah seperti memperbaharui dan mengambil data dari database. Beberapa sistem manajemen basis data relasional umum yang paling banyak digunakan adalah Oracle, Sybase, Microsoft SQL Server, Microsoft Access, dan masih banyak lagi.

Terdapat 2 (dua) jenis perintah SQL, yaitu:

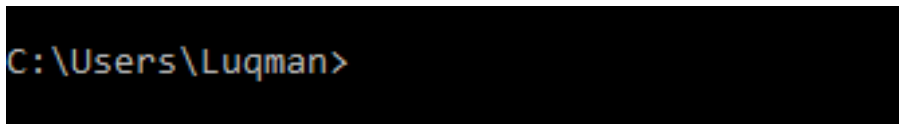
1. DDL (Data Definition Language), DDL adalah perintah SQL yang berhubungan dengan pendefinisian suatu struktur di dalam database. Perintah dasar yang termasuk DDL seperti:
 - CREATE, Perintah yang digunakan untuk membuat database
 - ALTER,
 - RENAME

- DROP,
2. DML (Data Manipulation Language), DML adalah perintah SQL yang digunakan untuk memanipulasi atau pengolahan data yang ada di dalam tabel. Perintah DML antara lain:
- SELECT
 - INSERT
 - UPDATE
 - DELETE

5.1.1 Membuat Database

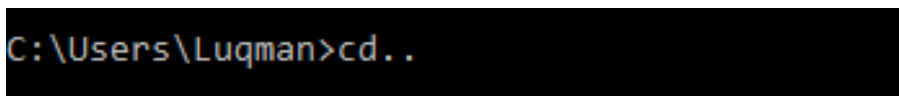
Untuk membuat database MySQL kita hanya menggunakan cmd atau command prompt, yaitu:

- Disini kita dapat menekan tombol keyboard (windows + r) dan ketikan cmd dan muncul gambar



Gambar 5.1

- Jika sudah, selanjutnya kita ketikan perintah cd..



Gambar 5.2

- Tekan tombol enter, lalu ketikan perintah cd..



Gambar 5.3

- Setelah muncul, kita ketikan cd xampp/mysql/bin
- Setelah itu kita ketikan mysql -u root -p jika ada Enter Password enter saja karena secara default password kita kosong

```
C:\>cd xampp/mysql/bin
```

Gambar 5.4

```
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.5.5-10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Gambar 5.5

- Jika sudah, kita dapat membuat database terlebih dahulu dengan mengetikkan kode

```
1 CREATE DATABASE php7;
```

```
mysql> CREATE DATABASE php7;
Query OK, 1 row affected (0.00 sec)
```

Gambar 5.6

Bentuk perintah diatas digunakan untuk membuat sebuah database baru dengan nama php7. Aturan dalam penamaan pada umumnya database sama seperti aturan dengan penamaan sebuah variabel, kita bebas mengkombinasikan huruf, angka, dan underscore (_). Akan tetapi, jika database yang akan kita buat sudah ada maka akan tampil pesan error.

5.1.2 Menampilkan Database

Untuk melihat apa saja database yang sudah kita buat dapat ketikkan perintah sebagai berikut:

```
1 SHOW DATABASES;
```

5.1.3 Membuka Database

Kita harus membuka database MySQL yang ingin kita gunakan, atau mengaktifkannya dengan perintah USE. Silahkan ketik perintah sebagai berikut:


```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| elban    |
| information_schema |
| mysql    |
| performance_schema |
| php7     |
| phpmyadmin |
| test     |
+-----+
7 rows in set (0.65 sec)
```

Gambar 5.7

```
USE php7;
```

Jika benar maka seharusnya tampil 'Database changed', artinya kita menggunakan database yang baru saja dibuat.

```
mysql> USE php7;
Database changed
```

Gambar 5.8

5.1.4 Menghapus Database

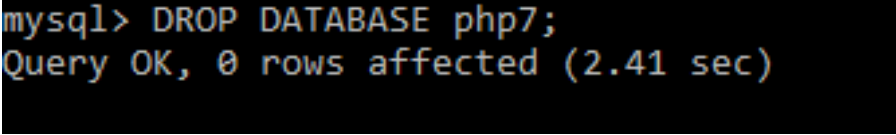
Setelah perintah-perintah dasar mengenai cara pembuatan database lalu bagaimana cara untuk menghapus database itu sendiri. Menghapus database menggunakan perintah DROP, perintah DROP digunakan untuk menghapus database dan seluruh isi atau informasi yang ada di dalam database tersebut. Silahkan ketikkan perintah sebagai berikut untuk menghapus database:

```
1 DROP DATABASE php7;
```

Jika sudah benar, maka seharusnya akan tampil pesan sebagai berikut:

```
1 Query OK, 0 rows affected (2.41 sec)
```

Jika memang sesuai maka itu artinya maka database kita sudah terhapus. Untuk lebih jelasnya silahkan lihat gambar berikut:



```
mysql> DROP DATABASE php7;
Query OK, 0 rows affected (2.41 sec)
```

Gambar 5.9

5.1.5 Membuat Tabel

Untuk dapat membuat tabel kita harus pilih terlebih dahulu database mana yang akan kita gunakan. Contoh database yang akan kita gunakan adalah perusahaan. Cara memilihnya kita gunakan perintah use.

```
1 use nama_database
```

Karena kita akan menggunakan database perusahaan, maka kita mengetikkan

```
1 use perusahaan
```

Jika benar akan muncul tulisan database changed. setelah itu kita siap untuk membuat tabel. Struktur dalam pembuatan tabel sebagai berikut:

```
1 Create table [nama tabel] (
2 nama field 1 type data,
3 nama field 2 type data,,
4 ... nama field n);
```

Contoh apabila kita ingin membuat tabel dengan nama pegawai dengan 5 field sebagai berikut:

```
1 CREATE TABLE pegawai(
2 id INT NOT NULL PRIMARY KEY,
3 nama_pegawai VARCHAR(30) NOT NULL,
4 jenis_kelamin VARCHAR(10) NOT NULL,
5 alamat VARCHAR(30) NOT NULL,
6 jabatan VARCHAR(20) NOT NULL
7 );
```

Untuk saran, penamaan tabel atau field jangan pakai spasi.

```
mysql> CREATE TABLE pegawai(
  -> id INT NOT NULL PRIMARY KEY,
  -> nama_pegawai VARCHAR(30) NOT NULL,
  -> jenis_kelamin VARCHAR(10) NOT NULL,
  -> alamat VARCHAR(30) NOT NULL,
  -> jabatan VARCHAR(20) NOT NULL
  -> );
Query OK, 0 rows affected (0.35 sec)
```

Gambar 5.10

5.1.6 Melihat Tabel

Untuk melihat tabel kita dapat menggunakan perintah

```
1 show tables;
```

Lalu untuk melihat isi dari tabel kita dapat menggunakan perintah

```
1 desc [nama tabel];
```

Sebagai contoh kita menggunakan tabel pegawai maka perintahnya

```
1 desc pegawai;
```

```
mysql> show tables;
+-----+
| Tables_in_perusahaan |
+-----+
| pegawai               |
+-----+
1 row in set (0.05 sec)

mysql> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | NULL    |       |
| nama_pegawai | varchar(30) | NO   |     | NULL    |       |
| jenis_kelamin | varchar(10) | NO   |     | NULL    |       |
| alamat     | varchar(30) | NO   |     | NULL    |       |
| jabatan    | varchar(20) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.17 sec)
```

Gambar 5.11

OBJECT ORIENTED PROGRAMMING

6.1 Aplikasi 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

6.1.1 Merancang Database

Untuk membuat sebuah aplikasi secara sederhana, langkah pertama yang dilakukan adalah merancang tabel-tabel database. Secara sederhana, struktur database dapat dilihat sebagai berikut: Tabel logbook digunakan untuk mencatat seluruh kegiatan

Tabel 6.1 Tabel Logbook

Nama Field	Tipe	Panjang	Keterangan
id_logbook	int	10	primary key dan auto increment
tanggal	date	-	tanggal pada logbook
petugas	varchar	100	nama petugas yang dinas pada hari tersebut
vendor	varchar	100	nama vendor yang dinas pada hari tersebut
penerbangan	varchar	100	daftar penerbangan pada hari tersebut
kegiatan	varchar	500	full kegiatan pegawai pada satu hari
point _{kerusakan}	varchar	200	daftar alat yang rusak sampai hari tersebut
target _{pekerjaan}	varchar	200	target pekerjaan alat yang rusak sampai hari tersebut
id_user	tinyint	1	id user admin

pegawai dan beberapa hal penting lainnya. Tabel pegawai digunakan untuk mendaftarkan

Tabel 6.2 Tabel Pegawai

Nama Field	Tipe	Panjang	Keterangan
id_pegawai	int	10	primary key dan auto increment
pegawai	varchar	100	daftar nama pegawai

daftar pegawai. Tabel user adalah tabel yang mendaftarkan admin untuk masuk ke sistem. Contoh perintah yang digunakan untuk membuat tabel logbook, yaitu:

```

1 CREATE TABLE IF NOT EXISTS 'logbook' (
2   'id_logbook' int(10) NOT NULL,
3   'tanggal' date NOT NULL,

```

Tabel 6.3 Tabel User

Nama Field	Tipe	Panjang	Keterangan
id_user	tinyint	2	primary key dan auto increment
username	varchar	30	username user
password	varchar	35	password user
nama	varchar	50	nama user
alamat	varchar	150	alamat user
hp	varchar	20	nomor handphone user
level	tinyint	1	level user

```

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 ;

```

Contoh perintah yang digunakan untuk tabel pegawai, yaitu:

```

1 CREATE TABLE IF NOT EXISTS 'pegawai' (
2   'id_pegawai' int(10) NOT NULL,
3   'pegawai' varchar(100) NOT NULL
4 ) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=latin1 ;

```

Contoh perintah yang digunakan untuk tabel user, yaitu:

```

1 CREATE TABLE IF NOT EXISTS 'user' (
2   'id_user' tinyint(2) NOT NULL,
3   'username' varchar(30) NOT NULL,
4   'password' varchar(35) NOT NULL,
5   'nama' varchar(50) NOT NULL,
6   'alamat' varchar(150) NOT NULL,
7   'hp' varchar(20) NOT NULL,
8   'level' tinyint(1) NOT NULL
9 ) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1 ;

```

6.2 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) adalah salah satu cara membuat program dengan memecah menjadi beberapa modul-modul

sederhana yang kita sebut sebagai objek. Akan tetapi, setiap memiliki fungsi dan tugas tersendiri. Saat ini OOP telah menjadi sebuah standar dalam dunia pemrograman, termasuk bahasa pemrograman PHP. Kita dapat membuat program PHP walaupun tanpa OOP sama sekali, namun programmer akan beralih menggunakan OOP karena konsepnya yang fleksibel.

Class

Class adalah 'bentuk kasar' dari objek. Class digunakan untuk membuat sebuah kerangka dasar. Yang kita pakai nantinya adalah hasil cetakan dari class yaitu objek. Dalam PHP, penulisan class dapat diawali dengan keyword class kemudian nama dari class tersebut. Aturan penulisan class sama seperti aturan variabel dalam PHP yaitu diawali dengan huruf atau underscore untuk karakter utama, setelahnya dapat berupa selain huruf pada karakter kedua dan selanjutnya. Isi dari class yang berada di dalam kurung kurawal. Contoh:

```
1 <?php
2 class motor {
3     // isi dari class test.
4 }
5 ?>
```

Property

Property atau atribut adalah data yang terdapat di dalam sebuah class. Dalam PHP, aturan tentang tata cara penamaan property ini sama dengan penamaan variabel. Contoh:

```
1 <?php
2 class laptop {
3     var $pemakai;
4     var $merk;
5     var $tipe;
6     // lanjutan isi dari class motor..
7 }
8 ?>
```

Dari contoh diatas \$pemakai, \$merk, dan \$tipe adalah property dari class motor. Seperti yang kita lihat, penulisan property di dalam PHP sama dengan variabel yaitu dalam penggunaan tanda dollar (\$). Ingat sebuah class tidak harus mempunyai property.

Method

Method adalah tindakan apa yang ingin kita lakukan di dalam class. Jika kita menggunakan analogi sebuah motor maka contoh method yaitu: menghidupkan motor, mematikan motor, memvariasi motor, berbagai tindakan lainnya. Method pada dasarnya function atau fungsi yang berada di dalam class. Contoh:

```
1 <?php
2 class motor {
3     function menghidupkan_motor() {
4         //... isi dari method menghidupkan_motor
5     }
6
7     function memvariasi_motor() {
8         //... isi dari method memvariasi_motor
9     }
10 }
```

```

9     }
10
11     ... //isi dari class motor
12 }
13 ?>

```

Dari contoh di atas, function `menghidupkan_motor()` dan function `memvariasi_motor()` adalah method dari class `motor`. Seperti yang kita lihat, bahwa penulisan method di dalam PHP sama dengan cara penulisan function. Sebuah class tidak harus memiliki method.

Object

Object atau Objek adalah hasil atau output dari class. Jika kita menggunakan analogi motor, maka objek adalah `motor_honda`, `motor_yamaha`, `motor_kawasaki`, dan lain-lain. Contoh:

```

1 <?php
2 class motor {
3     //... isi dari class motor
4 }
5
6 $motor_honda = new motor();
7 $motor_kawasaki = new motor();
8 ?>

```

6.2.0.1 Cara Membuat Objek PHP Istilah objek dalam OOP terdiri dari class, property, method dan object. Contoh objek dalam PHP:

```

1 <?php
2 // membuat class motor
3 class motor {
4
5     // buat property untuk class motor
6     var $pemakai;
7     var $merk;
8     var $tipe;
9
10    // buat method untuk class motor
11    function menghidupkan_motor() {
12        return "Menghidupkan Motor";
13    }
14    function memvariasi_motor() {
15        return "Memvariasi Motor";
16    }
17 }
18
19 // buat objek dari class motor
20 $motor_honda = new motor();
21 ?>

```

Dari contoh diatas, kita telah membuat class dengan nama `motor`, selain class kita juga membuat property, method, dan objeknya.

6.2.0.2 Cara Mengakses Objek PHP Cara mengakses objek dalam PHP ini terdiri dari objek, property, dan method. Contoh:


```

1 <?php
2 // membuat class motor
3 class motor {
4
5     // membuat property untuk class motor
6     var $pemakai;
7     var $merk;
8     var $tipe;
9
10    // membuat method untuk class motor
11    function menghidupkan_motor() {
12        return "Menghidupkan Motor";
13    }
14    function memvariasi_motor() {
15        return "Memvariasi Motor";
16    }
17 }
18
19 // membuat objek dari class motor
20 $motor_honda = new motor();
21
22 // atur property
23 $motor_honda->pemakai="Luqman";
24 $motor_honda->merk="Beat";
25 $motor_honda->tipe="CBS";
26
27 // tampilkan property
28 echo $motor_honda->pemakai;
29 echo "<br />";
30 echo $motor_honda->merk;
31 echo "<br />";
32 echo $motor_honda->tipe;
33 echo "<br />";
34
35 // tampilkan method
36 echo $motor_honda->menghidupkan_motor();
37 echo "<br />";
38 echo $motor_honda->memvariasi_motor();
39 ?>

```

Hasil dari contoh diatas adalah sebagai berikut:

```

1 Luqman
2 Beat
3 CBS
4 Menghidupkan Motor
5 Memvariasi Motor

```

6.2.1 Database

Basis Data (database) adalah sebuah kumpulan data yang dapat disimpan secara terstruktur di dalam komputer untuk diolah atau di manipulasi menggunakan sebuah program aplikasi agar dapat menghasilkan informasi. Basis data ini merupakan bagian yang sangat penting dalam pembuatan sistem informasi karena berfungsi sebagai pusat penyimpanan data yang akan di organisasikan. Proses memasukan dan

mengolah data dari media penyimpanan membutuhkan media perangkat lunak yang disebut DBMS (Database Management System). DBMS adalah sistem perangkat lunak yang dapat mengorganisasikan data secara praktis dan efisien.

Beberapa perangkat lunak atau DBMS yang sering digunakan dalam pembuatan aplikasi antara lain:

1. MySQL - <https://www.mysql.com/>
2. Oracle - <https://www.oracle.com/id/index.html>
3. Microsoft SQL Server - <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
4. MariaDB - <https://mariadb.org/>

Akan tetapi, dalam pembuatan aplikasi disini kita menggunakan MySQL.

MySQL adalah sistem manajemen basis data relasional open-source (RDBMS). Namanya adalah kombinasi dari "My", nama co-founder putri Michael Widenius, dan "SQL", singkatan dari Structured Query Language. MySQL adalah perangkat lunak bebas dan sumber terbuka di bawah ketentuan GNU General Public License, dan juga tersedia di bawah berbagai lisensi kepemilikan. MySQL dimiliki dan disponsori oleh perusahaan Swedia MySQL AB, yang dibeli oleh Sun Microsystems (sekarang Oracle Corporation). Pada 2010, ketika Oracle mengakuisisi Sun, Widenius melakukan forking pada proyek open-source MySQL untuk membuat MariaDB. MySQL adalah komponen dari tumpukan perangkat lunak aplikasi web LAMP (dan lainnya), yang merupakan akronim untuk Linux, Apache, MySQL, Perl / PHP / Python. MySQL digunakan oleh banyak aplikasi web berbasis database, termasuk Drupal, Joomla, phpBB, dan WordPress. MySQL juga digunakan oleh banyak situs web populer, termasuk Facebook, Twitter, Flickr, dan YouTube.

Untuk PHP kita menggunakan phpMyAdmin. phpMyAdmin adalah alat perangkat lunak gratis yang ditulis dalam PHP, dimaksudkan untuk menangani administrasi MySQL melalui Web. phpMyAdmin mendukung berbagai operasi di MySQL dan MariaDB. Operasi yang sering digunakan (mengelola basis data, tabel, kolom, hubungan, indeks, pengguna, izin, dll) dapat dilakukan melalui antarmuka pengguna, sementara Anda masih memiliki kemampuan untuk secara langsung menjalankan pernyataan SQL apa pun.

6.2.2 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 dibandingkan 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_mysql.dll` pada file `php.ini`. Namun pada file `php.ini` sudah ada skrip `extension=php_mysql.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.

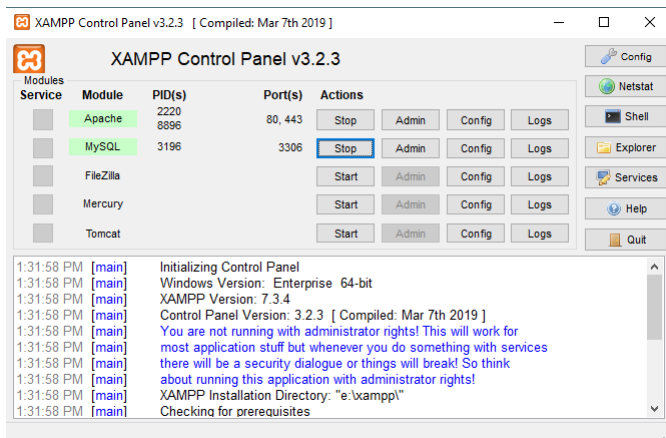
6.2.3 Membuat Database

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.

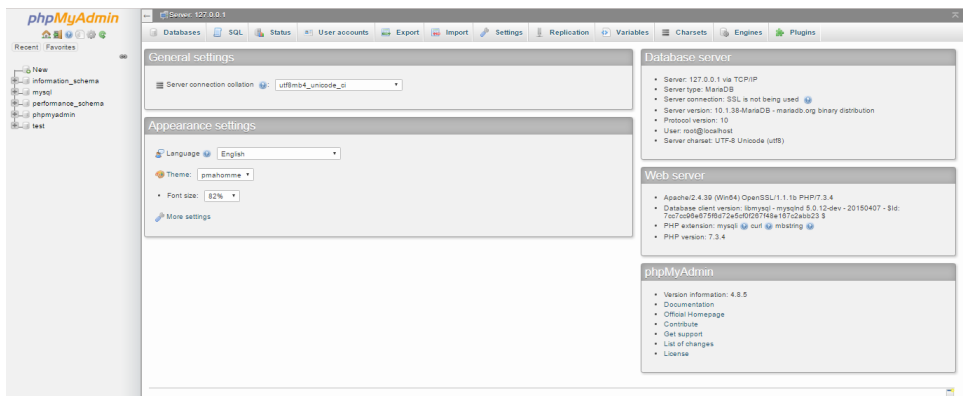
Selain memanfaatkan tag `html`, website yang menggunakan flash juga bisa dikategorikan sebagai web statis, meskipun ada sebagian kecil yang sudah memiliki database dalam mengelola konten tidak perlu membuka sebuah file tertentu namun hanya menambahkan pada form yang telah disiapkan dan tersimpan di database. 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

```
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,
```



Gambar 6.1 XAMPP Control Panel

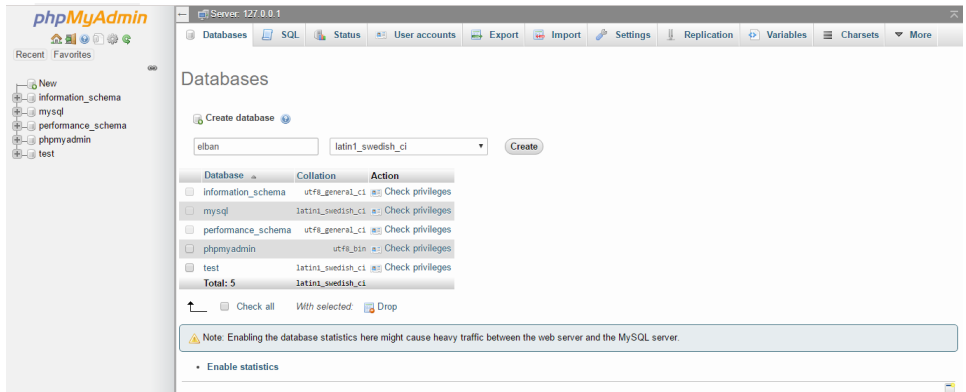


Gambar 6.2 Halaman Utama

```

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 —
16
17 INSERT INTO 'logbook' ('id_logbook', 'tanggal', 'petugas', '
18 vendor', 'penerbangan', 'kegiatan', 'point_kerusakan', '
19 target_pekerjaan', 'id_user') VALUES
20 (2, '2019-03-31', 'M. Arif S, Vania & Rismayadi', 'Anto (CUPPS)
21 Hisyam', 'Citylink KJT – KNO', 'PM di SCP 2 Inter Line 1', '

```



Gambar 6.3 Create Database

```

19      'Master Clok Gate 4, XRAY BHS Internasional', 'Master Clock
      Gate 4', 12),
20      (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);

```

Table structure for table 'pegawai'

```

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;

```

Dumping data for table 'pegawai'

```

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 ALTER TABLE 'logbook'
86   ADD PRIMARY KEY ('id_logbook');
87
88 —
89 — Indexes for table 'pegawai'
90 —
91 ALTER TABLE 'pegawai'
92   ADD PRIMARY KEY ('id_pegawai');
93
94 —
95 — Indexes for table 'user'
96 —
97 ALTER TABLE 'user'
98   ADD PRIMARY KEY ('id_user');
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;
```