

"PHP: Powerful, Hot, and Painless Programming"

JEFFREY MCMANUS - ELANCE.COM



<?PHP

Web Programmer
HEROES?>

theSpora Press

Robbi Hendriyanto



PHP Heroes

Robbi Hendriyanto

PHP Heroes

Daftar Isi

Memulai Dengan PHP	1
Apa itu PHP?	1
<i>Proses dalam menjalankan sebuah script PHP</i>	1
<i>Kenapa memilih PHP ?</i>	2
Instalasi Web Server di Windows	2
Instalasi Web Server di Linux	5
Instalasi Web Server di Mac OSX	7
Menyapa dunia dengan PHP	8
PHP Dasar	10
Variabel dan Aturan Penulisan	10
<i>Aturan penulisan sederhana</i>	10
<i>Variabel</i>	10
<i>Konstanta (Constants)</i>	11
Tipe Data dan Operator	12
<i>Tipe Data</i>	12
<i>Operator</i>	16
Kondisional dan Perulangan	19
Kondisional di PHP	19
<i>if statement</i>	19
<i>else dan elseif statement</i>	20

<i>switch statement</i>	20
Menggunakan Perulangan	21
<i>while...</i>	21
<i>do....while</i>	22
<i>for..</i>	22
<i>foreach</i>	23
<i>break dan continue</i>	23
Perulangan Bersarang (Nested Loops)	24
Fungsi String	25
Bermain dengan Fungsi String	25
<i>Membuat Berbagai Macam String</i>	25
<i>Beberapa Fungsi String yang Populer</i>	26
Formatting String	29
Fungsi Array	33
Berkenalan dengan Array di PHP	33
Mengakses Element Array	33
<i>Mengganti elemen pada Array</i>	34
<i>Mengambil sebagian elemen Array dengan array_slice()</i>	36
<i>Menghitung elemen pada Array</i>	37
<i>"Pointer" pada Array</i>	37
Array Multidimensi	38
Pengurutan "Sorting" Array	39
"Function" alias Fungsi	41
Fungsi di PHP	41
Membuat Fungsi Sederhana	41

Fungsi Rekursif	42
Include & Require	43
Mengolah Form	44
Mengakses Data dari Form	44
Membuat Form Registrasi	45
Cek Data dengan Form Validation	46
Membuat Form Upload File	47
Sessions dan Cookies	49
Session	49
Membuat, Mengakses dan Menghapus Session	49
Cookie	50
Membuat, Mengakses dan Menghapus Cookie	50
Bermain dengan File	51
Mengakses File	51
Membaca File Perbaris	52
Membaca File Perkarakter	53
Menulis File	53
File Permissions	54
Menduplikasi, Mengubah Nama dan Menghapus File	54
PHP dan MySQL	55
Form Tambah Mahasiswa	58
Kode PHP untuk Menambah Data	59
Kode PHP untuk Memodifikasi Data	59
Kode PHP untuk Menghapus Data	59
Object Oriented dengan PHP	60

Dasar OOP pada PHP	60
Constructors dan Destructors pada PHP	60
Kenali “Visibility” pada OOP di PHP	61
Bermain dengan Methods	62
Inheritance atau Pewarisan	62

Memulai Dengan PHP

Apa itu PHP?

PHP atau yang dikenal dengan Hypertext Preprocessor merupakan bahasa pemrograman (*server side scripting language*) yang berbasis script dimana untuk menjalankan tidak dibutuhkan kompailer tetapi langsung menggunakan Web Server. Web Server akan menjalankan setiap permintaan yang diminta oleh client dalam hal ini adalah web browser atau command-line, dan akan menyediakan halaman web yang diminta oleh pengunjung. Halaman web yang dimaksud adalah halaman yang terdiri atas client side scripting language (HTML dan juga dapat disertai dengan javascript dan CSS).

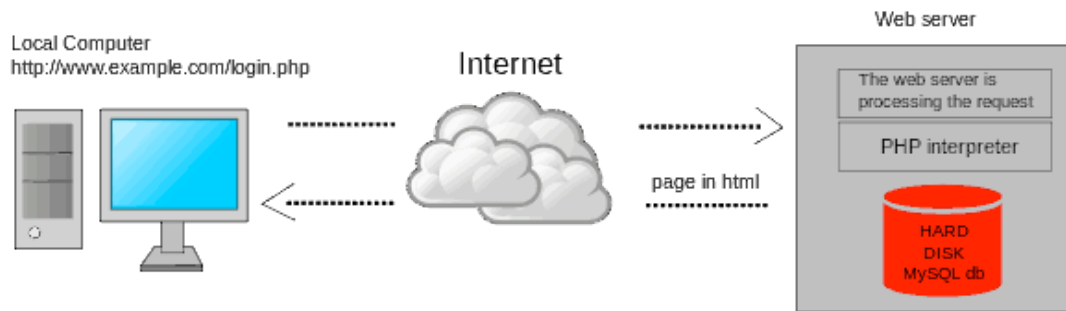
Apa yang dapat dilakukan dengan PHP?

- PHP dapat menghasilkan konten yang dinamis dalam sebuah website
- PHP dapat membuat, membuka, membaca, menulis dan menutup file pada server
- PHP dapat mengumpulkan data
- PHP dapat mengirimkan dan menerima cookies (dari client)
- PHP dapat membuat, menambahkan, menghapus, mengubah (CRUD) data dari Database
- PHP dapat membatasi hak akses pada sebuah halaman pada website
- PHP dapat mengenkripsi data

Proses dalam menjalankan sebuah script PHP

Secara sederhana untuk menjalankan sebuah script PHP dirangkum dalam 3 buah tahap sebagai berikut :

1. Pengunjung web me-*request* sebuah halaman web dengan mengetikkan URL pada sebuah browser atau command-line. Web browser akan mengirimkan data ke web server pada waktu yang sama baik menggunakan permintaan secara langsung atau melalui AJAX (Asynchronous JavaScript And XML)
2. Data yang dikirim oleh client kemudian ditangkap oleh server, kemudian PHP Engine akan mengolah permintaan dari client
3. Hasil yang dihasilkan setelah selesai akan dikirimkan kembali ke client melalui web browser dalam bentuk HTML, CSS, Javascript atau konten web lainnya baik berupa gambar ataupun file multimedia.



Alur proses untuk menjalankan sebuah script PHP

Kenapa memilih PHP ?

PHP merupakan scripting yang paling banyak digunakan di Internet dan banyak didukung oleh Internet Service Provider yang menyediakan layanan web hosting yang mendukung PHP. Dengan menggunakan PHP tidak diperlukan spesifikasi hardware yang besar, ataupun sistem operasi tertentu. PHP dapat berjalan diberbagai sistem operasi baik Windows, Linux, Mac OSX, dan sistem operasi lain yang berbasis UNIX. Untuk menjalankan sebuah script PHP diperlukan sebuah Web server yang memiliki PHP Module, ada beberapa Web server yang mendukung PHP seperti Apache, IIS, Zeus, Lighttpd. PHP juga sangat mudah untuk dipelajari jika dibandingkan dengan bahasa untuk membuat web lainnya seperti Perl, Java (JSP), Ruby (RoR).

Instalasi Web Server di Windows

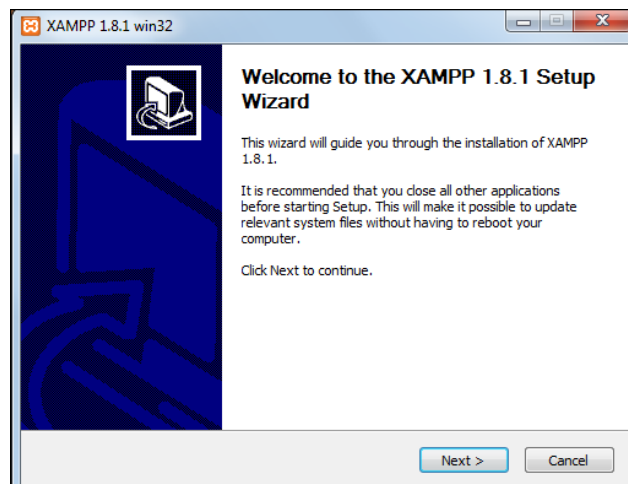
Web Server yang tersedia pada Windows beraneka ragam, mulai dari Apache ataupun IIS, Proses instalasi dapat dilakukan secara satu-demi satu artinya proses instalasi dilakukan terpisah-pisah, yaitu dengan menginstall Apache terlebih dahulu, lalu meng-copy-kan PHP modul kedalamnya dan melakukan beberapa konfigurasi pada file konfigurasi yang tersebia baik di Apache dan PHP. Cara cepatnya untuk dapat menjalankan Web server pada sistem operasi Windows, dapat menggunakan paketan instalasi yang sudah tersedia, salah satu paketan yang cukup populer adalah XAMPP. XAMPP merupakan singkatan dari X (Cross Platform) A (Apache HTTP Server) M (MySQL) P (PHP) P (Perl)

Download XAMPP terbaru : <http://www.apachefriends.org/en/xampp-windows.html>

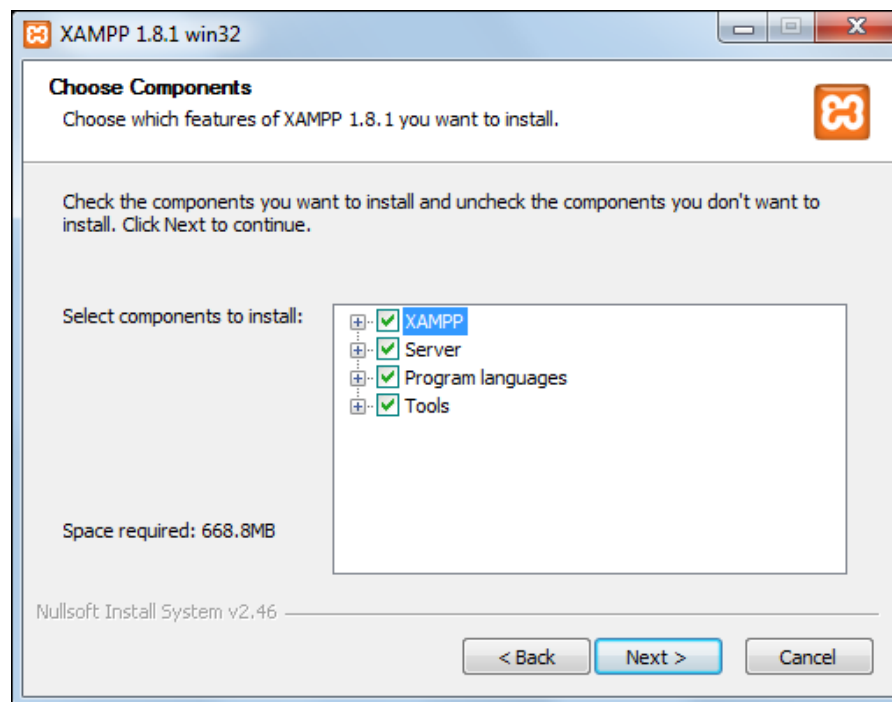
Berikut ini adalah langkah-langkah proses instalasi XAMPP di Windows :

1. Buka file hasil yang telah didownload lalu klik dua kali untuk memulai proses instalasi
2. Pilih Bahasa kemudian klik OK

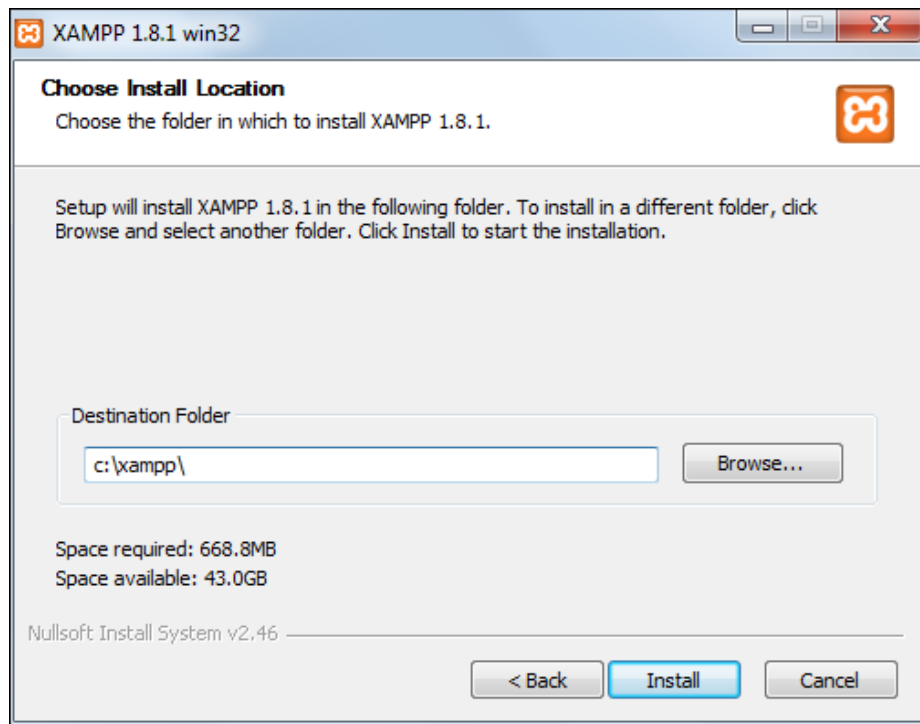
3. Klik Next untuk melanjutkan proses instalasi



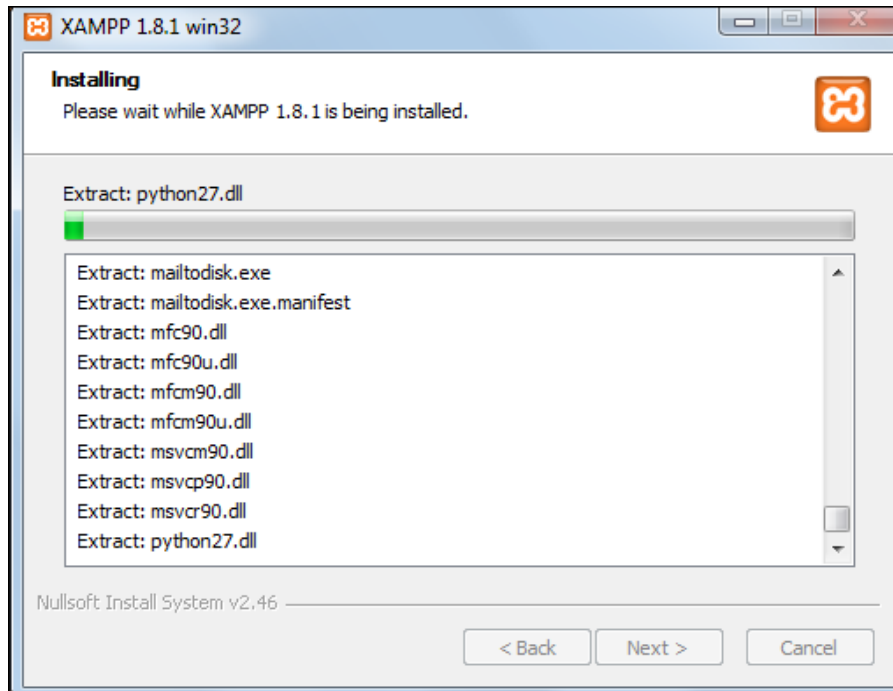
4. Setelah itu maka akan muncul dialog untuk memilih komponen yang akan di Install, pilih komponen yang dihendaki, lalu klik Next



5. Kemudian akan muncul window untuk memilih Folder tempat menyimpan hasil instalasi, pilih folder kemudian klik Install

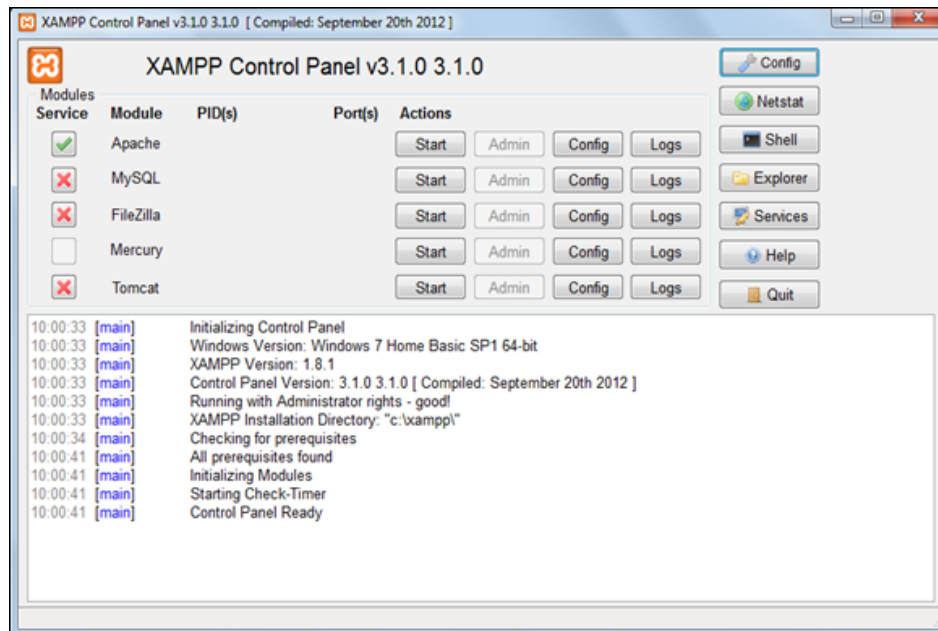


6. Lalu setelah mengklik Install maka proses instalasi akan berjalan



7. Sesudah proses instalasi selesai silakan mengklik Finish untuk mengakhiri

8. XAMPP yang sudah berjalan



9. File web disimpan didalam folder htdoc (default), dan untuk membuka ketik <http://localhost>

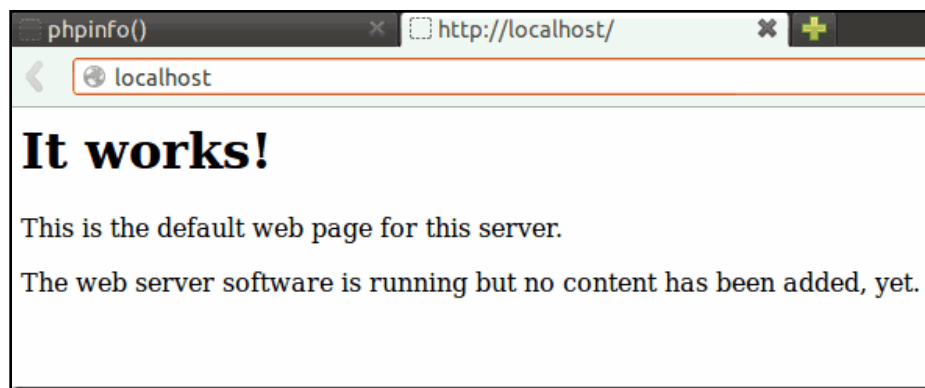
Instalasi Web Server di Linux

Sama seperti pada sistem operasi Windows, pada Linux juga terdapat paketan yang dikenal dengan nama LAMP (Linux, Apache, MySQL, PHP). Berikut ini cara instalasi LAMP pada Ubuntu 13.02

1. Buka Terminal (Ctrl-Alt-T)
2. Sesudah muncul Terminal, ketik perintah dibawah ini

```
sudo apt-get install lamp-server^
```

3. Setelah proses selesai, coba membuka browser lalu ketikkan <http://localhost>



4. Ketikkan perintah dibawah ini pada Terminal untuk mengecek apakah MySQL sudah berjalan dengan memasukkan password yang dibuat untuk MySQL pada proses instalasi

```
mysql -u root -p
```

Cara lainnya bisa dengan mengikuti langkah sebagai berikut :

1. Ketikkan di terminal (untuk masuk ke root)

```
sudo su
```

2. Lalu install MySQL 5 terlebih dahulu dengan mengetikkan perintah

```
apt-get install mysql-server mysql-client
```

3. Setelah itu saat proses instalasi akan diminta untuk memasukkan password

```
New password for the MySQL "root" user: <-- yourrootsqlpassword
```

```
Repeat password for the MySQL "root" user: <-- yourrootsqlpassword
```

4. Setelah instalasi MySQL berhasil lalu dilanjutkan dengan instalasi Apache, dengan mengetikkan perintah

```
apt-get install apache2
```

5. Dilanjutkan dengan instalasi PHP5

```
apt-get install php5 libapache2-mod-php5
```

6. Lalu restart Apache dengan perintah

```
/etc/init.d/apache2 restart
```

7. Integrasi MySQL ke dalam PHP5

- cari modul php5 yang tersedia dengan mengetikkan perintah

```
apt-cache search php5
```

- pilih beberapa modul dan lakukan instalasi dengan menggunakan perintah berikut

```
apt-get install php5-mysql php5-curl php5-gd php5-intl php-pear php5-imagick php5-imap php5-mcrypt php5-memcache php5-ming php5-ps php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl
```

- restart Apache

```
/etc/init.d/apache2 restart
```

8. Install phpmyadmin

```
apt-get install phpmyadmin
```

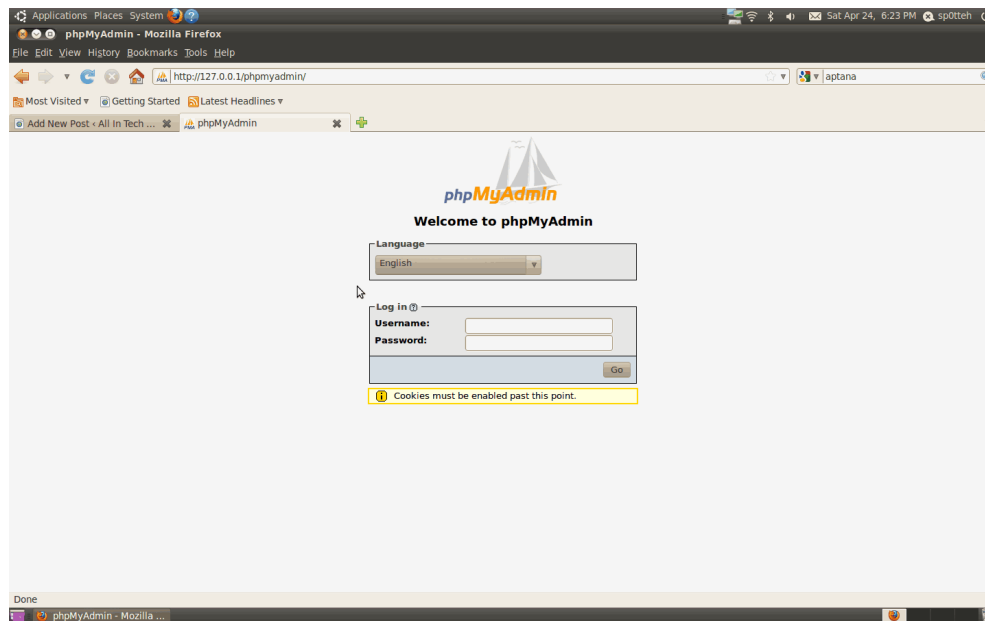
9. Jawab pertanyaan saat proses instalasi phpmyadmin

```
Web server to reconfigure automatically: <-- apache2
```

```
Configure database for phpmyadmin with dbconfig-common? <-- No
```

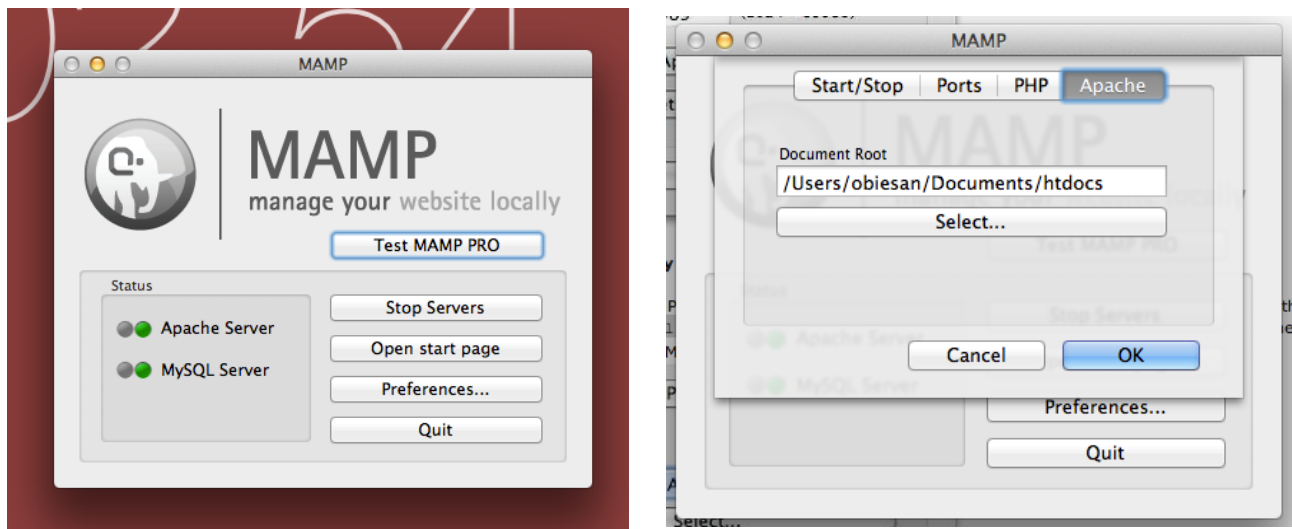
10. Mengecek apakah phpmyadmin berhasil diinstal dengan mengetikkan

```
http://localhost/phpmyadmin/
```



Instalasi Web Server di Mac OSX

Pada Mac OSX paketan Web server yang paling *uptodate* adalah MAMP (Macintosh Apache MySQL PHP). Paketan ini dapat di download di <http://www.mamp.info/en/downloads/index.html>. Setelah selesai mendownload, proses instalasi cukup dengan mendrag MAMP ke dalam folder application dan setelah itu jalankan MAMP dan klik Start Server. Untuk menyimpan dokumen web dapat disimpan di folder Application/ MAMP/htdocs dan halaman awal untuk MAMP dapat diakses dengan alamat : `http://localhost:8888`



Untuk mengubah folder default penyimpanan dokumen web dapat dilakukan dengan cara mengklik tombol Preferences, kemudian pilih tab Apache lalu pilih folder yang dihendaki dan diakhiri dengan mengklik OK.


Menyapa dunia dengan PHP

Sebelum memulai membuat script dalam PHP, hal yang biasa dilakukan adalah mengecek apakah PHP sudah terinstall dan bekerja dengan baik sesuai apa yang diharapkan. Buka text editor kemudian ketik script di bawah ini :

```
<?php
    phpinfo();
?>
```

Simpan file dengan nama info.php, lalu jalankan dengan mengetikkan URL pada browser :

<http://localhost/info.php>

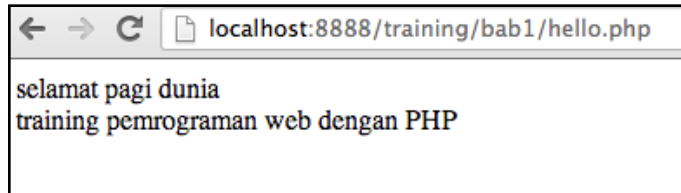
PHP Version 5.4.3	
	
System	Darwin robbis-MacBook-Pro.local 12.4.0 Darwin Kernel Version 12.4.0: Wed May 1 17:57:12 PDT 2013; root:xnu-2050.24.15~1/RELEASE_ARM_T8040 x86_64
Build Date	Jun 18 2012 17:03:13
Configure Command	'./configure' '--with-mysql=/Applications/MAMP/Library/' '--with-apxs2=/Applications/MAMP/Library/bin/apxs' '--with-gd' '--with-jpeg-dir=/Applications/MAMP/Library/' '--with-png-dir=/Applications/MAMP/Library/' '--with-zlib' '--with-freetype-dir=/Applications/MAMP/Library/' '--prefix=/Applications/MAMP/bin/php/php5.4.3' '--exec-prefix=/Applications/MAMP/bin/php/php5.4.3' '--sysconfdir=/Applications/MAMP/bin/php/php5.4.3/conf' '--with-config-file-path=/Applications/MAMP/bin/php/php5.4.3/conf' '--enable-ftp' '--enable-gd-native-ttf' '--with-bz2=/usr' '--with-ldap' '--with-mysqli=/Applications/MAMP/Library/bin/mysqli_config' '--with-t1lib=/Applications/MAMP/Library/' '--enable-mbstring=all' '--with-curl=/Applications/MAMP/Library/' '--enable-sockets' '--enable-bcmath' '--with-imap=shared,/Applications/MAMP/Library/lib/imap-2007f' '--enable-soap' '--with-kerberos' '--enable-calendar' '--with-pgsql=shared,/Applications/MAMP/Library/pg' '--enable-exif' '--with-libxml-dir=/Applications/MAMP/Library/' '--with-gettext=shared,/Applications/MAMP/Library/' '--with-xsl=/Applications/MAMP/Library/' '--with-pdo-mysql=shared,/Applications/MAMP/Library/' '--with-pdo-pgsql=shared,/Applications/MAMP/Library/pg' '--with-mcrypt=shared,/Applications/MAMP/Library/' '--with-openssl' '--enable-zip' '--with-iconv=/Applications/MAMP/Library'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/Applications/MAMP/bin/php/php5.4.3/conf
Loaded Configuration File	/Applications/MAMP/bin/php/php5.4.3/conf/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

Setelah memastikan PHP sudah berjalan dengan baik, berarti sudah dapat melakukan pemrograman web dengan menggunakan PHP. Setiap bahasa pada saat pertama kali melakukan pembelajaran memiliki sebuah

kebiasaan atau ritual untuk memulai pemrograman dengan menyapa dunia terlebih dahulu dengan membuat script sebagai berikut :

```
<?php  
  
    echo "selamat pagi dunia";  
    print "<br>training pemrograman web dengan PHP";  
  
?>
```

output :



Jika sudah berhasil maka dengan ini siap untuk mempelajari materi berikutnya.

PHP Dasar

Variabel dan Aturan Penulisan

Di dalam PHP terdapat aturan dan konsep-konsep dasar yang hampir sama dengan bahasa pemrograman lainnya, antara lain di PHP terdapat variabel, tipe data, operator, konstanta dan aturan-aturan lainnya. Tapi ada yang menarik di dalam aturan pada PHP, di PHP kita dapat melakukan hal “jorok” yaitu kita dapat bebas membuat variabel di mana saja. Hal tersebut terkadang dapat memudahkan tetapi dapat menyulitkan juga ketika aplikasi yang dibangun lebih kompleks dan tidak rapi dalam membuat script akan menimbulkan kesulitan pada tahap pengembangan ataupun perawatan.

Aturan penulisan sederhana

Dalam menulis script PHP dimulai dengan tanda “lebih kecil tanda tanya” lalu dituliskan “php” dan ditutup dengan “tanda tanya lebih besar”. Kode PHP dapat diletakkan dimana saja (perlu diingat proses pembacaan code dari paling kiri atas ke kanan bawah). Artinya kode PHP dapat diletakkan ditengah-tengah script HTML asalkan disimpan dengan file berekstensi .php.

Dalam penulisan komentar di PHP menggunakan double slash “//” untuk 1 baris dan /* untuk membuka komentar yang diakhiri dengan */ untuk menutup komentar. Untuk mengakhiri setiap perintah diharuskan menggunakan titik koma “;” dan penggunaan tanda kurung untuk mengambil parameter inputan pada sebuah fungsi.

```
<?php
    echo "coba lagi";
    //latihan penulisan sederhana
    /*ini komentar untukkd3817sc
       lebih dari 1 baris*/
?>
```

Aturan terakhir dalam penulisan pada PHP adalah Case Sensitive, berarti penulisan variabel yang akan dibahas pada bagian selanjutnya pada buku ini harus membedakan antara penggunaan huruf besar dan huruf kecil.

Variabel

Variabel digunakan untuk menyimpan data dan memanipulasi data dalam script. Ada beberapa hal yang harus diperhatikan dalam penamaan variabel, yaitu :

1. Variabel harus dimulai dengan tanda dolar (\$)
2. Karakter utama harus huruf (A-Z/a-z) atau sebuah underscore “_”
3. Karakter kedua dan seterusnya terdiri atas (boleh kombinasi) huruf, angka dan underscore.

Contoh penamaan variabel :

```
$coba_var  
$cobavar  
$_cobavar  
$_123  
$c
```

Contoh case-sensitive variabel :

```
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$color="red";  
echo "My car is " . $color . "<br>";  
echo "My house is " . $COLOR . "<br>";  
echo "My boat is " . $coLOR . "<br>";  
?>  
  
</body>  
</html>
```

Output

```
My car is red  
My house is  
My boat is
```

Konstanta (*Constants*)

Konstanta atau Constants sebenarnya mirip dengan sebuah variabel cuman yang membedakannya adalah aturan bahwa sebuah konstanta tidak boleh dirubah nilainya atau tidak didefinisikan terlebih dahulu. Cara membuat konstanta tidak menggunakan tanda dollar (\$) sebelum nama konstanta. Nama konstanta dapat dimulai dengan **huruf** atau **underscore**. Dan konstanta bersifat global tidak seperti variabel yang bersifat lokal.

Berikut ini contoh konstanta

```
<?php  
  
define("MERDEKA", "Merdeka hidup atau mati");  
echo MERDEKA; // output Merdeka hidup atau mati  
  
echo "<br>";  
  
define("merdeka", "Sudah merdeka?", true);  
echo merdeka; // output Sudah merdeka?  
echo "<br>";  
echo MERDEKA; // output Merdeka hidup atau mati  
  
?>
```

Tipe Data dan Operator

Seperti bahasa pemrograman lainnya, PHP memiliki tipe data untuk mendefinisikan jenis variabel yang akan digunakan dan mengenal beberapa operator untuk mempermudah proses yang diperlukan dalam suatu aplikasi web.

Tipe Data

a. String

String adalah tipe data yang terdiri atas sekumpulan karakter seperti “Coba Latihan 2”

Contoh :

```
<?php
$x = "Hello world! <br>";
echo $x;
$x = 'Hello world!';
echo $x;
?>
```

output :

```
Hello world!
Hello world!
```

b. Integer

Integer adalah tipe data yang terdiri atas number dengan desimal point. Integer dimulai dengan karakter pertama hanya dari 0-1-9. Integer tidak boleh mengandung karakter tanda koma “,” atau kosong (blank). Integer tidak harus mengandung nilai desimal saja. Integer terdiri dari negatif dan positif. Integer terdiri atas desimal (basis 10) atau hexadesimal (basis 16 yang dimulai dengan karakter 0x) atau terdiri atas octal (basis 8 yang dimulai dengan karakter 0)

Contoh :

```
<?php
$x = 5985;
var_dump($x);
echo "<br>";
$x = -345; // negative number
var_dump($x);
echo "<br>";
$x = 0x8C; // hexadecimal number
var_dump($x);
echo "<br>";
$x = 047; // octal number
var_dump($x);
?>
```

output :

```
int(5985)
int(-345)
int(140)
int(39)
```

c. Float

Float adalah tipe data yang terdiri atas number dengan desimal point atau number di dalam form eksponensial.

Contoh :

```
<?php
$x = 10.365;
var_dump($x);
echo "<br>";
$x = 2.4e3;
var_dump($x);
echo "<br>";
$x = 8E-5;
var_dump($x);
?>
```

output :

```
float(10.365)
float(2400)
float(8.0E-5)
```

d. Boolean

Tipe data boolean adalah tipe data yang hanya terdiri atas 2 buah nilai yaitu TRUE atau FALSE

Contoh :

```
<?php
$x=true;
$y=false;

echo $x;
echo "<br>".$y;

?>
```

output :

```
1
```

e. Array

Array adalah tipe data yang dapat menyimpan berbagai nilai dalam sebuah variabel

Contoh :

```
<?php
$cars=array("Volvo","BMW","Toyota");
var_dump($cars); //digunakan untuk mengeluarkan tipe data dan nilai di dalam array
?>
```

output :

```
array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }
```

f. Object

Object adalah tipe data yang menyimpan data dan informasi dan bagaimana untuk memproses data tersebut. Tipe data object akan ditemui pada PHP OOP.

Contoh :

```
<?php
class Car
{
    var $color;
    function Car($color="green")
    {
        $this->color = $color;
    }
    function what_color()
    {
        return $this->color;
    }
}

function print_vars($obj)
{
    foreach (get_object_vars($obj) as $prop => $val)
    {
        echo "\t$prop = $val\n";
    }
}

// instantiate one object
$herbie = new Car("white");

// show herbie properties
echo "\herbie: Properties\n";
print_vars($herbie);
?>
```

output :

```
herbie: Properties color = white
```

g. Resource

Resource adalah tipe data spesial yang berisi referensi ke sumber eksternal, seperti file atau database

h. NULL

Tipe data NULL merepresentasikan sebuah variabel yang tidak memiliki isi bukan berisi karakter kosong ataupun spasi. Jika variabel memiliki nilai NULL maka akan bernilai kosong.

Contoh :

```
<?php
$x="Hello world!";
$x=null;
var_dump($x);
?>
```

output :

```
NULL
```

Berikut ini adalah latihan mengenai Variabel dan fungsi-fungsi di dalam PHP untuk melakukan validasi atau manipulasi pada penggunaan variabel.

Latihan 1 : penggunaan fungsi gettype()

```
<?php

$test_var;
echo gettype($test_var) . "<br/>"; // Displays "NULL"
$test_var = 15;
echo gettype($test_var) . "<br/>"; // Displays "integer"
$test_var = 8.23;
echo gettype($test_var) . "<br/>"; // Displays "double"
$test_var = "Hello, world!";
echo gettype($test_var) . "<br/>"; // Displays "string"

?>
```

Latihan 2 : mengubah tipe data suatu variabel

```
<?php

$test_var = 8.23;
echo $test_var . "<br/>"; // Displays "8.23"
settype($test_var, "string");
echo $test_var . "<br/>"; // Displays "8.23"
settype($test_var, "integer");
echo $test_var . "<br/>"; // Displays "8"
settype($test_var, "float");
echo $test_var . "<br/>"; // Displays "8"
settype($test_var, "boolean");
echo $test_var . "<br/>"; // Displays "1"

?>
```

Latihan 3 : mengubah tipe data menggunakan Casting

```
<?php

$test_var = 8.23;
echo $test_var . "<br/>"; // Displays "8.23"
echo (string) $test_var . "<br/>"; // Displays "8.23"
echo (int) $test_var . "<br/>"; // Displays "8"
echo (float) $test_var . "<br/>"; // Displays "8.23"
echo (boolean) $test_var . "<br/>"; // Displays "1"

?>
```

Operator

Setelah mempelajari mengenai variabel selanjutnya materi yang akan dibahas adalah operator. Apa itu operator, operator adalah cara untuk memanipulasi konten dari satu atau lebih variabel sehingga menghasilkan suatu nilai baru. Beberapa jenis operator yang dimiliki oleh PHP adalah sebagai berikut

a. Arithmetic

Arithmetic adalah operator yang terdiri dari penjumlahan, pengurangan, perkalian, pembagian dan sisa bagi.

```
$x=15;
$y=6;
echo ($x + $y); // output 21
echo "<br>";
echo ($x - $y); // output 9
echo "<br>";
echo ($x * $y); // output 90
echo "<br>";
echo ($x / $y); // output 2.5
echo "<br>";
echo ($x % $y); // output 3
?>
```

b. Assignment

Assignment adalah operator untuk memberikan suatu nilai atau mengubah suatu nilai di sebuah variabel. Dimana operator sebelah kiri akan mengubah nilai disebelah kanan.

```
<?php
$x=5;
echo $x; // outputs 5
echo "<br>";

$y=10;
$y += 100;
echo $y; // output 110
echo "<br>";

$z=25;
$z -= 7;
echo $z; // output 18
echo "<br>";

$i=10;
$i *= 3;
echo $i; // output 30
echo "<br>";

$j=10;
$j /= 3;
echo $j; // output 3.333333
echo "<br>";

$k=15;
$k %= 7;
echo $k; // output 1
?>
```

c. String

String adalah operator yang berfungsi untuk mengubah, menambahkan atau menggabungkan nilai yang bertipe string atau kedalam tipe data string.

```
<?php
$a = "Salam";
$b = $a . " Kenal";
echo $b; // output Salam Kenal
echo "<br>";
$x="Salam";
$x .= " Kenal";
echo $x; // output Salam Kenal
echo "<br>";
?>
```

d. Increment / Decrement

Increment / Decrement adalah operator yang berfungsi untuk menambahkan atau mengurangi suatu nilai dalam variabel

```
<?php
$x=5;
echo ++$x; // output 6
echo "<br>";
$y=5;
echo $y++; // output 5
echo "<br>";
$z=6;
echo --$z; // output 5
echo "<br>";
$i=6;
echo $i--; // output 6
?>
```

e. Comparison

Comparison adalah operator yang berfungsi untuk membandingkan antara variabel bisa dalam nilainya dan tipe data suatu variabel

```
<?php
$x=50;
$y="50";
var_dump($x == $y); //output bool(true)
echo "<br>";
var_dump($x === $y); //output bool(false)
echo "<br>";
var_dump($x != $y); //output bool(false)
echo "<br>";
var_dump($x !== $y); //output bool(true)
echo "<br>";
$a=50;
$b=90;
var_dump($a > $b); //output bool(false)
echo "<br>";
var_dump($a < $b); //output bool(true)
?>
```


f. Logical

Logical adalah operator yang berfungsi untuk memberikan nilai logik dari hasil perbandingan, nilai logik yang dihasilkan yaitu TRUE dan FALSE (boolean)

```
<?php
$x = 10; $y = 15; $z = "10";
echo ( ($x > 9) && ($x < 15) ) . "<br/>"; // Display 1 (true)
echo ( ($x == 10) or ($y == 0) ) . "<br/>"; // Display 1 (true)
echo ( ($x == 0) xor ($y == 3) ) . "<br/>"; // Display "" (false) karena keduanya true
echo ( !($x == 5) ) . "<br/>"; // Display 1 (true) karena $x bukan 5 adalah false dan
dinyatakan
echo ($x === $z); // Display "" (false) karena tipe data tidak sama
?>
```

g. Array

Operator ini sebenarnya operator logika yang digunakan untuk variabel array. Detail dari penggunaannya adalah sebagai berikut :

Operator	Name	Example	Result
+	Union	<code>\$x + \$y</code>	Union of \$x and \$y (but duplicate keys are not overwritten)
==	Equality	<code>\$x == \$y</code>	True if \$x and \$y have the same key/value pairs
===	Identity	<code>\$x === \$y</code>	True if \$x and \$y have the same key/value pairs in the same order and of the same types
!=	Inequality	<code>\$x != \$y</code>	True if \$x is not equal to \$y
<>	Inequality	<code>\$x <> \$y</code>	True if \$x is not equal to \$y
!==	Non-identity	<code>\$x !== \$y</code>	True if \$x is not identical to \$y

```
<?php
$x = array("a" => "sapi", "b" => "kuda");
$y = array("c" => "kerbau", "d" => "keledai");
$z = $x + $y; // union $x dan $y
var_dump($z);
//output array(4) { ["a"]=> string(4) "sapi" ["b"]=> string(4) "kuda" ["c"]=> string(6)
"kerbau" ["d"]=> string(7) "keledai" }
echo "<br/>";

var_dump($x == $y); //output bool(false)
echo "<br/>";
var_dump($x === $y); //output bool(false)
echo "<br/>";
var_dump($x != $y); //output bool(true)
echo "<br/>";
var_dump($x <> $y); //output bool(true)
echo "<br/>";
var_dump($x !== $y); //output bool(true)
?>
```

Kondisional dan Perulangan

Kondisional di PHP

Sama seperti bahasa pemrograman lain, PHP memiliki perintah untuk melakukan suatu pemilihan kondisi atau yang dikenal dengan nama kondisional (*conditional* atau *decisions*).

if statement

“if” statement akan menjalankan kode perintah di dalamnya jika kondisi pada “if” bernilai true, berikut adalah kode dari “if”

```
if (condition){
    code to be executed if condition is true;
}
```

contoh :

```
<?php

$nilai = 69;

if ($nilai == 69)
    echo "Index nilai yang diperoleh adalah BC";

//display "Index nilai yang diperoleh adalah BC"
```

Penulisan “if” jika kode di dalam kondisi hanya berupa 1 perintah diperbolehkan tidak menggunakan kurung kurawal {}

```
<?php

$nilai = 72;

if ($nilai >= 70 && $nilai <75) {
    echo "Index nilai yang diperoleh adalah B";
}

//display "Index nilai yang diperoleh adalah B"

if ($nilai >=70){
    if ($nilai <75) {
        echo "Index nilai yang diperoleh adalah B";
    }
}

//display "Index nilai yang diperoleh adalah B"

?>
```

else dan elseif statement

Pada kondisional “if” dapat ditambahkan perintah “else” dan “elseif”. Perintah ini sebagai alternatif jika kondisi pada if tidak terpenuhi maka akan masuk ke dalam kondisi “elseif” dan jika tidak terpenuhi maka akan masuk ke kondisi terakhir yaitu “else”. Perhatikan contoh berikut ini untuk membedakan penggunaan “else” dan “else if”

```
<?php

$nilai = 67;

if ($nilai >= 80) {
    echo "Index nilai yang diperoleh adalah A";
}elseif ($nilai <80 && $nilai >=75){
    echo "Index nilai yang diperoleh adalah AB";
}elseif ($nilai <75 && $nilai >=70){
    echo "Index nilai yang diperoleh adalah B";
}elseif ($nilai <70 && $nilai >=65){
    echo "Index nilai yang diperoleh adalah BC";
}elseif ($nilai <65 && $nilai >=60){
    echo "Index nilai yang diperoleh adalah C";
}elseif ($nilai <60 && $nilai >=55){
    echo "Index nilai yang diperoleh adalah D";
}else{
    echo "Index nilai yang diperoleh adalah E";
}
// output "Index nilai yang diperoleh adalah BC"
?>
```

switch statement

Pada kondisional “switch” berfungsi untuk menyatakan suatu kondisi dengan banyak kondisi nilai yang dibandingkannya. Mirip seperti menggunakan “if” dan “elseif” / “else”, untuk lebih jelasnya dapat melihat contoh sebagai berikut :

```
<?php

$menu = "print";

switch($menu){
    case "new" :
        echo "Buka menu New File";
        break;
    case "open" :
        echo "Buka menu Open File";
        break;
    case "print" :
        echo "Buka menu Print";
        break;
    case "exit" :
        echo "Keluar dari aplikasi";
        break;
    default :
        echo "kode yang dimasukkan salah";
}
// output "Buka menu Print"
?>
```

Penggunaan kondisional “Switch” harus menggunakan kode “break;” disetiap akhir dari perintah “case”, jika tidak menggunakan “break” maka akan terjadi kesalahan pada output yang dihasilkan, berikut ini contoh switch yang salah

```
<?php
$menu = "print";

switch($menu){
    case "new" :
        echo "Buka menu New File";
    case "open" :
        echo "Buka menu Open File";
    case "print" :
        echo "Buka menu Print";
    case "exit" :
        echo "Keluar dari aplikasi";
    default :
        echo "kode yang dimasukkan salah";
}
// output "Buka menu PrintKeluar dari aplikasikode yang dimasukkan salah"
?>
```

Menggunakan Perulangan

Perulangan adalah perintah untuk melakukan eksekusi suatu kode atau baris program secara terus menerus sampai pada suatu kondisi tertentu yang didefinisikan pada perintah perulangan. Pada PHP sama seperti bahasa pemrograman lainnya memiliki perintah untuk perulangan “looping” antara lain seperti while, do-while, for, beberapa turunan dari for (foreach)

while...

Perintah while akan terus mengeksekusi kode atau baris program di dalamnya selama kondisi bernilai sama dengan apa yang didefinisikan pada while (true),

```
while(kondisi)
{
    code yang akan di eksekusi
}
```

contoh :

```
<?php
$x = 1;
while($x < 4){
    echo "Perulangan ke $x<br>";
    $x++;
}
?>
```

output :

```
Perulangan ke 1
Perulangan ke 2
Perulangan ke 3
```

do...while

Perintah ini mirip seperti “while” tetapi dalam “do-while” kode atau baris program akan dieksekusi terlebih dahulu sebanyak satu kali kemudian baru akan mengecek kondisi yang didefinisikan pada “do-while”. Bila kondisi true maka perulangan akan terus dilakukan.

```
do
{
    code yang akan di eksekusi
}
```

contoh

```
<?php
lebar = 1;
$panjang = 1;

do {
    $lebar++;
    $panjang++;
    $luas = $panjang * $lebar;
} while ( $luas < 10 );

echo "luas persegi dengan nilai $luas memiliki panjang $panjang cm dan lebar $lebar cm";

//output : "luas persegi dengan nilai 16 memiliki panjang 4 cm dan lebar 4 cm"

?>
```

for..

Perulangan for adalah perulangan dengan jumlah yang sudah ditentukan pada penetapan perintah for.

```
for (inisialisasi, kondisi, increment/decrement) {
    code yang akan di eksekusi
}
```

contoh

```
<?php

for ($x=0; $x<=5; $x++){
    echo "Nilai : $x <br>";
}

?>
```

Output

```
Nilai : 0
Nilai : 1
Nilai : 2
Nilai : 3
Nilai : 4
Nilai : 5
```

foreach

foreach merupakan perulangan khusus untuk variabel yang bertipe array, sintaknya sebagai berikut

```
foreach ($array as $value)
{
    code to be executed;
}
```

Contoh

```
<?php
$colors = array("red","green","blue","yellow");
foreach ($colors as $value)
{
    echo "$value <br>";
}
?>
```

Output

```
red
green
blue
yellow
```

break dan continue

kode “break” digunakan untuk memerhentikan kondisi perulangan sedangkan continue digunakan untuk melanjutkan perulangan dengan cara meng-skip perulangan dan kemudian dilanjutkan mengeksekusi perintah selanjutnya.

```
<?php
$count = 0;
while ( true ) {
    $count++;
    echo "mulai menghitung: $count <br/>";
    if ( $count == 5 ) break;
}

for ($i=1; $i <= 5; $i++) {
    if ( $i == 4 ) continue;
    echo "mulai menghitung: $i <br/>";
}
echo "selesai";

?>
```

Output

```
mulai menghitung: 1
mulai menghitung: 2
mulai menghitung: 3
mulai menghitung: 4
mulai menghitung: 5
```

```
mulai menghitung: 1  
mulai menghitung: 2  
mulai menghitung: 3  
mulai menghitung: 5  
selesai
```

Perulangan Bersarang (*Nested Loops*)

Perulangan bersarang atau lebih dikenal dengan istilah “nested loop” adalah perulangan yang terdiri atas perulangan di dalam baris program, singkatnya di dalam sebuah perulangan ada perulangan lagi. Perulangan bersarang dapat mengkombinasikan berbagai jenis perulangan tidak harus dalam satu jenis kode perintah, misalnya menggabungkan for di dalam sebuah while. Contoh sederhananya seperti berikut ini

```
<?php  
for ( $tens = 0; $tens < 3; $tens++ ) {  
    for ( $units = 0; $units < 3; $units++ ) {  
        echo $tens . $units . "<br/>";  
    }  
}  
?>
```

Output

```
00  
01  
02  
10  
11  
12  
20  
21  
22
```

Fungsi String

Bermain dengan Fungsi String

Seperti yang sudah dibahas pada materi sebelumnya mengenai tipe data string, string adalah sekumpulan karakter yang membentuk suatu nilai tertentu. Misalnya sebuah kalimat yang tersusun atas beberapa huruf, angka, dan simbol. Berapa jumlah minimal karakter disebut sebuah string yaitu 0. Jika suatu variabel diberikan nilai kosong dengan menggunakan tanda petik, contoh : `$x=""`; maka `$x` adalah sebuah string yang bernilai kosong. PHP memiliki banyak fungsi untuk mendukung pemrograman antara lain yaitu fungsi untuk memanipulasi string. Untuk melihat list fungsi string secara lengkap bisa mengunjungi URL berikut ini www.php.net/manual/en/ref.strings.php.

Membuat Berbagai Macam String

Cara untuk membuat sebuah string sangat sederhana, cukup dengan mengetikkan deretan karakter didalam sebuah kutip satu atau kutip dua, misalnya 'saya' atau "kamu". Berikut ini beberapa contoh untuk membuat suatu string.

Membuat string sederhana :

```
$iniString = "Latihan buat string";  
$iniString = 'latihan buat string';
```

Perbedaan menggunakan kutip dua dalam membuat string yaitu mampu membaca nilai sebuah variabel string dan mampu menerjemahkan penggunaan karakter spesial. Yang dimaksud karakter spesial seperti `\n` untuk pindah baris atau `\t` untuk tab horisontal dan beberapa karakter spesial lainnya. Contohnya perbedaan antara penggunaan kutip satu dan kutip dua :

```
<?php  
$iniString = 'Jakarta';  
echo "Hello, $iniString! <br/> "; // Displays "Hello, Jakarta!"  
echo 'Hello, $iniString! <br/> '; // Displays "Hello, $iniString!"  
echo "<pre>Hi\tJakarta! </pre> "; // Displays "Hi Jakarta!"  
echo '<pre>Hi\tJakarta! </pre> '; // Displays "Hi\tJakarta!"  
?>
```

Membuat string lebih dari 1 baris :

```
<?php  
$iniString = "Latihan membuat String, sangat mudah  
            bahkan tidak sampai beberapa menit  
            silakan unuk dicoba";  
?>
```

Membuat string dengan Delimiters

```
$pekerjaan = 'Mahasiswa';
```



```
//contoh delimiter heredoc berfungsi seperti kutip dua sedangkan untuk kutip satu
tinggal menambahkan kutip satu contoh : 'MULAI_TULISAN'

$iniString = <<<MULAI_TULISAN
"Setiap tahunnya semua $pekerjaan diharuskan mengikuti acara yang namanya 'orientasi'
hal ini berguna supaya $pekerjaan mengenal lingkungannya yang baru, tentunya orientasi
harus diselenggarakan dengan 'cerdas' dan 'mendidik'"
MULAI_TULISAN;

echo "<pre> $iniString </pre>";
```

Beberapa Fungsi String yang Populer

a. Menghitung panjang string

Panjang suatu string dapat dihitung dengan menggunakan fungsi **strlen()** sedangkan untuk menghitung banyak kata dapat menggunakan **fungsi str_word_count()** dengan contoh sebagai berikut

```
<?php
$iniString = "Pada hari minggu aku pergi ke kota";

echo strlen ($iniString) . "<br/>";           // Output 34
echo strlen ("kota Bandung") . "<br/>";       // Output 12
echo str_word_count($iniString) . "<br/>";    // Output 7
?>
```

b. Mengakses karakter pada string

Dalam PHP diberikan keleluasaan untuk mengakses karakter langsung pada string, script untuk proses tersebut dapat dilihat pada script di bawah ini

```
<?php
$iniString = "Pada hari minggu aku pergi ke kota";

echo $iniString[0] . "<br/>";
// Displays 'P'

echo $iniString[12] . "<br/>";
// Displays 'n'

$iniString[10] = '?';
echo $iniString . "<br/>";
// Displays 'Pada hari mi?ggu aku pergi ke kota'
?>
```

Fungsi **substr()** adalah fungsi untuk mengambil karakter sebagian atau keseluruhan baik dengan mendefinisikan karakter yang diminta, posisi dimulai untuk ekstraksi karakter dan banyaknya karakter yang akan diekstrak.

```
<?php
$iniString = "Pada hari minggu aku pergi ke kota";

echo substr( $iniString, 1, 6 ) . "<br/>";
// Displays 'ada ha'

echo substr( $iniString, 9 ) . "<br/>";

// Displays 'minggu aku pergi ke kota'
```

```

echo substr( $iniString, -1 ) . "<br/>";
// Displays 'a'

echo substr( $iniString, -9, -1 ) . "<br/>";
// Displays 'i ke kot'
?>

```

c. Mencari string

Di PHP terdapat banyak fungsi untuk mencari sebuah karakter atau string di suatu string. Fungsi **strstr()** adalah fungsi untuk mencari kata atau teks pada string atau tulisan. Hasil yang dihasilkan dari fungsi ini adalah berupa teks secara lengkap (kata) atau dapat mengembalikan nilai FALSE jika string yang dicari tidak ditemukan. Fungsi tersebut bersifat *case-sensitive*, jika untuk menghiraukan penggunaan huruf besar dan kecil bisa menggunakan versi *case-insensitive* yaitu **striistr()**. Contohnya :

```

<?php
$iniString = "Pada hari minggu aku pergi ke kota";

echo strstr( $iniString, "har" ) . "<br/>";
// Displays 'hari minggu aku pergi ke kota'

echo (strstr( $iniString, "per" ) ? "Yes" : "No" ) . "<br/>";
// Displays 'Yes'

echo strstr( $iniString, "aku pergi", true );
// Displays 'Pada hari minggu'
?>

```

Fungsi **strpos()** merupakan fungsi untuk mencari dimana posisi string yang dicari sedangkan fungsi **strrpos()** memiliki fungsi yang sama dengan strpos() tetapi dimulai dari deret paling kanan atau paling akhir. Kedua fungsi tersebut bersifat *case-sensitive*, jika untuk menghiraukan penggunaan huruf besar dan kecil bisa menggunakan versi *case-insensitive* dengan fungsi **stripos()** dan **strripos()**.

```

<?php
$iniString = "Pada hari minggu aku pergi ke kota";

echo strpos($iniString,"a"). "<br/>"; //output 1
echo strrpos($iniString,"a"). "<br/>"; //output 33

echo strpos($iniString,"a",7). "<br/>"; //output 17
echo strpos($iniString,"minggu"). "<br/>"; //output 10
?>

```

Fungsi **strpbrk()** merupakan fungsi untuk mencari string yang mengandung suatu deret karakter, contohnya sebagai berikut :

```

<?php
$iniString = "Pada hari minggu aku pergi ke kota";

echo strpbrk($iniString, "ui")."<br/>";
//output "i minggu aku pergi ke kota"

echo strpbrk($iniString, "zkt");

```

```
//output "ku pergi ke kota"
echo "<br/>";
?>
```

d. Mengganti karakter pada string

Ada tiga fungsi yang sering digunakan untuk mengganti karakter pada sebuah string, yaitu **str_replace()** atau fungsi *case-insensitive* yang bernama **stri_replace()** yang berfungsi untuk mengganti semua deret karakter pada teks atau string menjadi deret karakter yang diinginkan, fungsi **substr_replace()** adalah fungsi untuk mengganti deret karakter pada posisi yang telah ditentukan dan fungsi **strtr()** yaitu fungsi untuk menerjemahkan suatu deret karakter menjadi deret karakter yang baru. Berikut ini adalah contoh untuk memudahkan pemahaman.

```
<?php
$iniString = "Pada hari minggu aku pergi ke kota";

echo str_replace("kota","desa",$iniString)."<br/>";
//output "Pada hari minggu aku pergi ke desa";

str_replace("kota","desa",$iniString,$banyak)."<br/>";
echo "Jumlah kata yang diganti sebanyak $banyak kata<br/>";
//output "Jumlah kata yang diganti sebanyak 1 kata"

echo str_replace("des","kota",$iniString)."<br/>";
//output "Pada hari minggu aku pergi ke kota"

/*----- */

$iniString = "Pada hari minggu aku pergi ke kota ";

echo substr_replace($iniString,"tidur",17)."<br/>";
//output Pada hari minggu tidur

echo substr_replace($iniString,"tidur",17,9)."<br/>";
//output Pada hari minggu tidur ke kota

echo substr_replace($iniString,"tidur",17,-9)."<br/>";
//output "Pada hari minggu tiduri ke kota"
echo "<br/>";

/*----- */

$iniString = "Pada hari minggu 'aku' pergi ke kota";
echo strtr( $iniString, " '", "+-" )."<br/>";
//output Pada+hari+minggu+aku-+pergi+ke+kota
?>
```

e. Upper & Lowercase

Untuk memodifikasi penggunaan huruf kapital dan huruf kecil dapat menggunakan fungsi-fungsi seperti pada contoh script dibawah ini

```
<?php
$iniString = "pada hari minggu, aku Pergi ke kota";

echo strtolower( $iniString )."<br/>";
// Output "pada hari minggu, aku pergi ke kota"
```

```

echo strtoupper( $iniString )."<br/>";
// Output "PADA HARI MINGGU, AKU PERGI KE KOTA"

echo ucfirst( $iniString )."<br/>";
// Output "Pada hari minggu, aku Pergi ke kota"

echo lcfirst( $iniString )."<br/>";
// Output "pada hari minggu, aku Pergi ke kota"

echo ucwords( $iniString )."<br/>";
// Output "Pada Hari Minggu, Aku Pergi Ke Kota"
echo "<br/>";
?>

```

Formatting String

a. printf() & sprintf()

Fungsi printf() dan sprintf() adalah fungsi yang dapat digunakan untuk meng-format string ke berbagai bentuk format mulai dari biner sampai hexadesima, memberikan nilai 0 didepan number, menentukan panjang karakter default, menentukan panjang difit desimal, dll.

Cara penulisan fungsi ini : printf (format,arg1,arg2,arg++) dan sprintf(format,arg1,arg2,arg++)

Parameter	Description
<i>format</i>	<p>Required. Specifies the string and how to format the variables in it.</p> <p>Possible format values:</p> <ul style="list-style-type: none"> • %% - Returns a percent sign • %b - Binary number • %c - The character according to the ASCII value • %d - Signed decimal number (negative, zero or positive) • %e - Scientific notation using a lowercase (e.g. 1.2e+2) • %E - Scientific notation using a uppercase (e.g. 1.2E+2) • %u - Unsigned decimal number (equal to or greater than zero) • %f - Floating-point number (local settings aware) • %F - Floating-point number (not local settings aware) • %g - shorter of %e and %f • %G - shorter of %E and %f • %o - Octal number • %s - String • %x - Hexadecimal number (lowercase letters) • %X - Hexadecimal number (uppercase letters) <p>Additional format values. These are placed between the % and the letter (example %.2f):</p> <ul style="list-style-type: none"> • + (Forces both + and - in front of numbers. By default, only negative numbers are marked) • ' (Specifies what to use as padding. Default is space. Must be used together with the width specifier. Example: %'x20s (this uses "x" as padding) • - (Left-justifies the variable value) • [0-9] (Specifies the minimum width held of to the variable value) • .[0-9] (Specifies the number of decimal digits or maximum string length) <p>Note: If multiple additional format values are used, they must be in the same order as above.</p>
<i>arg1</i>	Required. The argument to be inserted at the first %-sign in the format string
<i>arg2</i>	Optional. The argument to be inserted at the second %-sign in the format string
<i>arg++</i>	Optional. The argument to be inserted at the third, fourth, etc. %-sign in the format string

Berikut ini contoh penggunaan printf dan sprintf ;

```
<?php

$num1 = 123456789;
$num2 = -123456789;
$char = 50; // The ASCII Character 50 is 2

// Note: The format value "%" returns a percent sign
printf("%b = %b <br>", $num1); // Binary number
printf("%c = %c <br>", $char); // The ASCII Character
printf("%d = %d <br>", $num1); // Signed decimal number
printf("%d = %d <br>", $num2); // Signed decimal number
printf("%e = %e <br>", $num1); // Scientific notation (lowercase)
printf("%E = %E <br>", $num1); // Scientific notation (uppercase)
printf("%u = %u <br>", $num1); // Unsigned decimal number (positive)
printf("%u = %u <br>", $num2); // Unsigned decimal number (negative)
printf("%f = %f <br>", $num1); // Floating-point number (local settings aware)
printf("%F = %F <br>", $num1); // Floating-point number (not local settings aware)
printf("%g = %g <br>", $num1); // Shorter of %e and %f
printf("%G = %G <br>", $num1); // Shorter of %E and %f
printf("%o = %o <br>", $num1); // Octal number
printf("%s = %s <br>", $num1); // String
printf("%x = %x <br>", $num1); // Hexadecimal number (lowercase)
printf("%X = %X <br>", $num1); // Hexadecimal number (uppercase)
printf("%+d = %+d <br>", $num1); // Sign specifier (positive)
printf("%+d = %+d <br>", $num2); // Sign specifier (negative)

$number = 123;
printf("With 2 decimals: %1$.2f
<br>With no decimals: %1$u", $number);

$number = 123;
$txt = sprintf("With 2 decimals: %1$.2f
<br>With no decimals: %1$u", $number);
echo $txt;

?>
```

b. Trimming

Fungsi ini merupakan fungsi untuk menghilangkan karakter spasi atau karakter yang telah didefinisikan pada parameter fungsi trim(), ltrim(), rtrim().

```
<?php

$kalimat = " 1. Nilai sebuah variabel dapat diganti ";

echo $kalimat."<br/>";
//output "1. Nilai sebuah variabel dapat diganti "

echo trim($kalimat)."<br/>";
//output "1. Nilai sebuah variabel dapat diganti"

echo ltrim($kalimat)."<br/>";
//output "1. Nilai sebuah variabel dapat diganti "
```

```

echo rtrim($kalimat)."<br/>";
//output "1. Nilai sebuah variabel dapat diganti"

echo ltrim($kalimat, "0..9. ")."<br/>";
//output "Nilai sebuah variabel dapat diganti "

echo rtrim($kalimat, "a..o. ")."<br/>";
//output "1. Nilai sebuah variabel dapat digant"

?>

```

c. Padding Strings

Fungsi **str_pad()** ini berguna untuk membuat deret string yang berfungsi sebagai padding, contoh script sebagai berikut :

```

<?php

echo str_pad( "Hello, world!", 20, "*" ) . "<br>";
// Displays "Hello, world!*****"

echo str_pad( "Hello, world!", 20, "123" ) . "<br>";
// Displays "Hello, world!1231231"

echo str_pad( "Hello, world!", 20, "*", STR_PAD_BOTH ) . "<br>";
// Displays "****Hello, world!****"

?>

```

d. Wrapping Lines

Fungsi wordwrap adalah fungsi untuk membuat teks berpindah baris jika sudah mencapai batas karakter dalam 1 baris.

```

<?php
$katakata = "Serangan badai debu menyapu wilayah Phoenix,
AS di awal pekan ketiga Agustus 2013. Fenomenaalam yang
tak biasa ini dilaporkan memutus aliran listrik serta
menumbangkan pepohonan hingga ke kawasan Tempe, Arizona.";

echo wordwrap($katakata,30, "<br/>");

?>

```

output

```

Serangan badai debu menyapu
wilayah Phoenix, AS di awal
pekan ketiga Agustus 2013.
Fenomenaalam yang tak biasa
ini dilaporkan memutus aliran
listrik serta menumbangkan
pepohonan hingga ke kawasan
Tempe, Arizona.

```

e. Formating Number

Fungsi `number_format` adalah fungsi untuk memberikan format pada number, contohnya :

```
<?php

$number = 1234.56;

$english_format_number = number_format($number);
echo $english_format_number."<br/>";
// output 1,235

$france_format_number = number_format($number, 2, ',', ' ');
echo $france_format_number."<br/>";
// output1 234,56

$number = 1234.5678;

// english notation without thousands separator
$english_format_number = number_format($number, 2, '.', '');
echo $english_format_number."<br/>";
// output 1234.57

?>
```

Fungsi Array

Berkenalan dengan Array di PHP

Array adalah suatu jenis variabel yang spesial dimana variabel tersebut dapat menyimpan lebih dari satu nilai pada waktu yang sama. Mempelajari array sangat penting karena dalam pembuatan suatu program dengan menggunakan array akan menjadi lebih fleksibel dan ringkas. Array memiliki spesifikasi dua buah fitur yang membuat array lebih baik dalam menyimpan suatu data, yaitu array dapat dibuat sepanjang atau sebanyak apapun dari satu sampai jutaan ilai yang tersimpan pada sebuah varibel dan array sangat mudah dimanipulasi isinya seperti melakukan “sorting” atau mencari data dll.

Array terdiri atas 2 buah bagian yaitu value untuk mennyimpan data atau nilai suatu data dan key sebagai petunjuk posisi dari suatu value. Array pada PHP terdiri atas 2 jenis yaitu array yang berindeks (array yang referensi “key”-nya menggunakan number dan biasanya dimulai dari angka 0) dan array asosiatif yaitu array yang menggunakan suattu string pada indexnya.

Contoh :

```
$x_array[0] = 1; // array berindeks
$x_array["nilai"] = 1; // array asosiatif
```

Ada beberapa cara untuk membuat array di PHP, berikut ini contoh untuk membuat sebuah array

```
<?php

$merk = array("Honda", "Suzuki", "Daihatsu", "Nissan", "Toyota");

$mobil = array ("merk" => "Honda",
                "jenis" => "MPV",
                "model" => "Freed",
                "tahun" => 2012);

?>
```

Mengakses Element Array

Setelah mampu membuat sebuah array, yang harus dilakukan selanjutnya adalah mampu untuk mengakses sebuah array. Seperti bagaimana memasukkan suatu nilai yang berasal dari array ke sebuah variabel, bagaimana untuk mengambil sebagian nilai dari array. Berikut ini contoh untuk mengakses array dan pembahasan detail fungsi-fungsi untuk melakukan manipulasi dan akses terhadap elemen dari suatu array.

Contoh :

```
<?php

$merkmobil = $merk[0];
$modelmobil = $mobil["model"];
echo $merkmobil." ".$modelmobil; //output "Honda Freed"
echo "<br/>";

$posisi = 2;
$merk[$posisi+2]="jeep";
echo $merk[4]."<br/>"; //output jeep

?>
```

Mengganti elemen pada Array

Mengganti elemen array adalah fungsionalitas yang sering digunakan, berikut ini contoh-contoh proses untuk mengganti elemen pada array berikut penjelasannya

```
?php

echo "<pre>";
$merk = array("Honda", "Suzuki", "Daihatsu", "Nissan", "Toyota");
$merk[3] = "Kia";
$merk[5] = "Hyundai";
$merk[] = "Subaru";

print_r($merk);

// cara 0
$merk = array("Honda", "Suzuki", "Daihatsu", "Nissan", "Toyota");

// cara 1
$merk1[0] = "Honda";
$merk1[1] = "Suzuki";
$merk1[2] = "Daihatsu";
$merk1[3] = "Nissan";
$merk1[4] = "Toyota";

// cara 2
$merk2[] = "Honda";
$merk2[] = "Suzuki";
$merk2[] = "Daihatsu";
$merk2[] = "Nissan";
$merk2[] = "Toyota";

print_r($merk);
print_r($merk1);
print_r($merk2);
echo "<pre>";
?>
```

Contoh diatas adalah contoh untuk mendefinisikan suatu array atau mengubah value dari elemen array yang sudah ada. Pada cara 0 merupakan proses pembuatan array langsung menggunakan fungsi array(). Sama seperti pada cara 1 cuman perbedaanny disana adalah melakukan assignment satu-satu pada variabel aarray yang sudah didefinisikan indexnya atau dapat juga dilakukan tanpa mendefinisikan terlebih dahulu dengan cara 2. Cara 2 secara otomatis index akan berurutan dari 0 untuk yang pertama kali di definisikan.

```

<?php

//----- Mengganti Array Asosiatif -----

// cara a
$mobil = array ("merk" => "Honda",
               "jenis" => "MPV",
               "model" => "Freed",
               "tahun" => 2012);

// cara b

$mobil1["merk"] = "Honda";
$mobil1["jenis"] = "MPV";
$mobil1["model"] = "Freed";
$mobil1["tahun"] = 2012;

print_r($mobil);
print_r($mobil1);

echo "<pre>";

?>

```

Pada contoh berikutnya (di atas) adalah mendefinisikan suatu array asosiatif yang dibagi menjadi 2 cara yang hampir sama persis dengan pendefinisian array menggunakan index. Cara a adalah pembuatan array yang mendefinisikan secara sekaligus dengan menggunakan fungsi array(). Sedangkan cara b sama seperti cara 1 pada contoh sebelumnya, yaitu membuat array dengan mengisi nilai satu demi satu ke dalam variabel array.

Jika dilihat dari kedua contoh di atas, mungkin akan ada pertanyaan kode print_r() fungsinya buat apa? jawabannya adalah fungsi print_r() merupakan fungsi untuk mengeluarkan nilai dari suatu array langsung pada arraynya tanpa terlebih dahulu melakukan ekstraksi pada masing-masing elemen. Fungsi print_r() akan menampilkan nama variabel array lengkap dengan key dan valuenya.

```

Array
(
    [0] => Honda
    [1] => Suzuki
    [2] => Daihatsu
    [3] => Kia
    [4] => Toyota
    [5] => Hyundai
    [6] => Subaru
)

Array
(
    [merk] => Honda
    [jenis] => MPV
    [model] => Freed
    [tahun] => 2012
)

```

Mengambil sebagian elemen Array dengan array_slice()

Fungsi array_slide() sesuai dengan namanya adalah fungsi untuk mengambil sebagian (mengeksktraksi) elemen pada sebuah array. Untuk menggunakan harus mendefinisikan posisi elemen pertama dan jumlah elemen yang akan diambil / diekstrak.

Contoh

```
<?php

echo "<pre>";
$mobil = array ( "merk" => "Honda",
                 "jenis" => "MPV",
                 "model" => "Freed",
                 "tahun" => 2012);

$mobil1 = array_slice ( $mobil,1,3);

print_r($mobil1);

$merk = array("Honda", "Suzuki", "Daihatsu", "Nissan", "Toyota");
print_r(array_slice ( $merk,2,2 ));
print_r(array_slice ( $merk,2,2, true));

echo "</pre>";

?>
```

Output

```
Array
(
    [jenis] => MPV
    [model] => Freed
    [tahun] => 2012
)

hasil ke 1
Array
(
    [0] => Daihatsu
    [1] => Nissan
)

hasil ke 2
Array
(
    [2] => Daihatsu
    [3] => Nissan
)
```

Pada contoh diatas adalah proses untuk mengambil sebagian nilai dari array menggunakan array_slide, jika hasil dari array slide tidak ditampung terlebih dahulu pada suatu variabel maka nilai index pada key jika langsung dioutputkan akan berurutan seperti pada hasil 1 dan 2.

Menghitung elemen pada Array

Proses menghitung array sangat penting dalam pembuatan program yang menggunakan array sebagai penampungan data sementara, misalny pada pengolahan database. PHP sudah menyediakan fungsi untuk menghitung sebuah array, seperti fungsi **count()**. Berikut contoh penggunaan fungsi count

```
<?php

$merk = array("Honda", "Suzuki", "Daihatsu", "Nissan", "Toyota");

$jumlah = count($merk);
$indexterakhir = count($merk)-1;

echo "Jumlah : ".$jumlah."<br>";
// display "Jumlah : 5"

echo "Posisi indeks terakhir : ".$indexterakhir;
// display "Posisi indeks terakhir : 4"

?>
```

“Pointer” pada Array

Jika pernah mempelajari keilmuan mengenai struktur data pada informatika, pasti sudah tidak asing lagi dengan istilah pointer. Pointer adalah petunjuk posisi dalam sebuah data atau file, begitu juga dalam array di PHP yang dapat diakses dengan menggunakan cara yang hampir sama dengan konsep sebuah pointer. Pointer akan menginisialisasikan posisi pertama dan terakhir dalam setiap pengambilan atau pembacaan dari data. Berikut ini ada beberapa fungsi yang mendukung konsep pointer pada PHP.

Function	Description
<code>current()</code>	Returns the value of the current element pointed to by the pointer, without changing the pointer position.
<code>key()</code>	Returns the index of the current element pointed to by the pointer, without changing the pointer position.
<code>next()</code>	Moves the pointer forward to the next element, and returns that element's value.
<code>prev()</code>	Moves the pointer backward to the previous element, and returns that element's value.
<code>end()</code>	Moves the pointer to the last element in the array, and returns that element's value.
<code>reset()</code>	Moves the pointer to the first element in the array, and returns that element's value.

Contoh :

```
<?php

$merk = array("Honda", "Suzuki", "Daihatsu", "Nissan", "Toyota");

echo current($merk) . "<br>"; //Honda
echo next($merk) . "<br>"; //Suzuki
echo next($merk) . "<br>"; //Daihatsu
echo key($merk) . "<br>"; // 2
```

```

echo current($merk) . "<br>"; //Daihatsu
echo end($merk) . "<br>"; //Toyota
echo reset($merk) . "<br>"; //Honda
echo next($merk) . "<br>"; //Suzuki
echo prev($merk) . "<br>"; //Honda
echo current($merk) . "<br>"; //Honda

?>

```

Array Multidimensi

Array multidimensi adalah array yang memiliki dimensi lebih dari 1 dimensi artinya di dalam array terdapat elemen yang memiliki tipe data array dan tanpa batasan berapa maksimal dimensi yang dapat dilakukan. Nilai lebih dari pembuatan array di PHP adalah cenderung mudah, sederhana dan tidak serumit bahasa pemrograman lainnya. Berikut ini adalah contoh untuk pembuatan array multidimensi

```

<?php

$listmobil = array(
    array ("merk" => "Honda",
          "jenis" => "MPV",
          "model" => "Freed",
          "tahun" => 2012),
    array ("merk" => "Toyota",
          "jenis" => "MPV",
          "model" => "Inova",
          "tahun" => 2013),
    array ("merk" => "Toyota",
          "jenis" => "Sedan",
          "model" => "Camry",
          "tahun" => 2011)
);

echo "<pre>";
print_r($listmobil);
echo "</pre>";

?>

```

Output

```

Array
(
    [0] => Array
        (
            [merk] => Honda
            [jenis] => MPV
            [model] => Freed
            [tahun] => 2012
        )

    [1] => Array
        (
            [merk] => Toyota
            [jenis] => MPV
            [model] => Inova

```

```

        [tahun] => 2013
    )

    [2] => Array
    (
        [merk] => Toyota
        [jenis] => Sedan
        [model] => Camry
        [tahun] => 2011
    )

)

```

Sesudah berhasil membuat array multidimensi, langkah selanjutnya adalah bagaimana cara mengakses array tersebut. Jawabannya sebenarnya caranya sama saja dengan array 1 dimensi cuman berbedaannya jumlah keynya lebih banyak.

Contoh

```

<?php
echo "<pre>";
print_r($listmobil);
echo "</pre>";

echo "Mobil di list ke 2 adalah bermerk ".$listmobil[1]["merk"]."<br/>";
echo "Yang berjenis ".$listmobil[1]["jenis"]." dengan model ".$listmobil[1]["model"]."
tahun ".$listmobil[1]["tahun"];
?>

```

Output

```

Mobil di list ke 2 adalah bermerk Toyota
Yang berjenis MPV dengan model Inova tahun 2013

```

Pengurutan “Sorting” Array

PHP juga menyediakan berbagai fungsi untuk melakukan sorting atau pengurutan pada array, sehingga memudahkan programmer untuk membuat sebuah program tanpa harus memikirkan atau membuat algoritma pengurutan secara manuall. Berikut ini fungsi “sorting” pada PHP antara lain :

- `sort()` - pengurutan arrays dari nilai kecil ke besar
- `rsort()` - pengurutan arrays dari nilai besar ke kecil
- `asort()` - pengurutan associative arrays dari nilai kecil ke besar, berdasarkan value
- `ksort()` - pengurutan associative arrays dari nilai kecil ke besar, berdasarkan key
- `arsort()` - pengurutan associative arrays dari nilai besar ke kecil, berdasarkan value
- `krsort()` - pengurutan associative arrays dari nilai besar ke kecil, berdasarkan key

Contoh

```
<?php

$merk = array("Honda", "Suzuki", "Daihatsu", "Nissan", "Toyota");
$jumlah = array ("honda" => 4,
                  "toyota" => 2,
                  "nissan" => 5,
                  "suzuki" => 3);

sort($merk);
print_r($merk);
echo "<br>";

rsort($merk);
print_r($merk);
echo "<br>";

asort($jumlah);
print_r($jumlah);
echo "<br>";

ksort($jumlah);
print_r($jumlah);
echo "<br>";

arsort($jumlah);
print_r($jumlah);
echo "<br>";

krsort($jumlah);
print_r($jumlah);

?>
```

Output

```
Array ( [0] => Daihatsu [1] => Honda [2] => Nissan [3] => Suzuki [4] => Toyota )
Array ( [0] => Toyota [1] => Suzuki [2] => Nissan [3] => Honda [4] => Daihatsu )
Array ( [toyota] => 2 [suzuki] => 3 [honda] => 4 [nissan] => 5 )
Array ( [honda] => 4 [nissan] => 5 [suzuki] => 3 [toyota] => 2 )
Array ( [nissan] => 5 [honda] => 4 [suzuki] => 3 [toyota] => 2 )
Array ( [toyota] => 2 [suzuki] => 3 [nissan] => 5 [honda] => 4 )
```

"Function" alias Fungsi

Fungsi di PHP

Fungsi merupakan subrutin atau dalam bahasa sederhananya program kecil yang memiliki fungsi yang spesifik. Fungsi di PHP dapat mengembalikan suatu nilai atau berupa fungsi yang tidak mengembalikan nilai. Fungsi menerima satu atau lebih parameter yang dapat akan digunakan dalam pemrosesan logic di dalam fungsi. Kegunaan dari fungsi antara lain memudahkan saat hendak digunakan pada halaman atau file lainnya, memudahkan saat proses perawatan dalam mencari kesalahan dan kekurangan dari aplikasi, fungsi dapat mendetailkan dan memudahkan saat membuat aplikasi yang besar dan detail.

Format penulisan fungsi di PHP

```
iniFungsi ( )  
iniFungsi ( argument1, argument2 )  
// contoh penulisan untuk menerima pengembalian nilai dari fungsi  
$stampung = iniFungsi ( argument1, argument2 );
```

Membuat Fungsi Sederhana

Berikut ini contoh untuk membuat sebuah fungsi sederhana yang akan mengembalikan sebuah nilai, ataupun yang tidak mengembalikan nilai :

```
<?php  
  
function hitungLuasLingkaran($jari){  
    $luas = $jari*$jari*3.14;  
    return $luas;  
}  
  
function kelilingLingkaran($jari){  
    $kel = 2*3.14*$jari;  
    echo "Keliling Lingkaran : ".$kel."<br>";  
}  
  
$r = 7;  
  
echo "Jari-jari Lingkaran : ".$r."<br>";  
  
kelilingLingkaran($r);  
  
echo "Luas Lingkaran : ".hitungLuasLingkaran($r)."<br>";  
  
?>
```


Output

```
Jari-jari Lingkaran : 7  
Keliling Lingkaran : 43.96  
Luas Lingkaran : 153.86
```

Contoh fungsi dengan default parameter

```
<?php  
  
function luasPersegi($sisi,$lebar=0){  
    if($lebar>0){  
        $luas = $sisi*$lebar;  
    }else{  
        $luas = $sisi*$sisi;  
    }  
    return $luas;  
}  
  
$r = 5;  
$l = 10;  
  
echo "Luas Persegi : ".luasPersegi($r)."<br>";  
echo "Luas Persegi Panjang : ".luasPersegi($r,$l)."<br>";  
  
?>
```

Output

```
Luas Persegi : 25  
Luas Persegi Panjang : 50
```

Fungsi Rekursif

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri dan akan berhenti jika tidak memenuhi kondisi yang dibentuk dengan bantuan kondisional yang dituliskan di dalam fungsi tersebut. Fungsi rekursif akan menghemat baris dalam pembuatan suatu aplikasi tetapi untuk membuat fungsi rekursif membutuhkan “cost” yang lebih besar dari pada membuat fungsi biasa.

```
<?php  
  
function contohRekursif($nilai){  
    if($nilai<50){  
  
        $nilai=contohRekursif($nilai+5);  
  
    }  
    return $nilai;  
}  
  
$x = contohRekursif(1);  
echo $x;  
?>
```

Include & Require

Dalam mengakses suatu file baik yang berisi sebuah fungsi atau object di dalam PHP membutuhkan fungsi untuk memanggil file tersebut. Beberapa fungsi pada PHP yang dapat memanggil fungsi tersebut adalah include , include_once, require dan require_once. Apa perbedaan keempat fungsi tersebut, berikut ini pembahasan singkat mengenai keempat modul tersebut

a. include

Fungsi Include digunakan di PHP untuk menyertakan suatu file kedalam proses yang memanggilnya, pada include dibutuhkan alamat file yang hendak dipanggil.

```
include "hal_fungsi.php";
```

b. include_once

Fungsi Require berfungsi seperti fungsi Include tetapi perbedaannya file yang dipanggil hanya akan digunakan 1 kali saja.

```
include_once "hal_fungsi.php";
```

c. require

Fungsi Require berfungsi seperti fungsi Include tetapi perbedaannya jika file tidak tersedia maka akan mengeluarkan PHP Error. Fungsi ini dibutuhkan untuk aplikasi yang bekerja dengan benar.

```
require "hal_fungsi.php";
```

d. require_once

Fungsi Require berfungsi seperti fungsi Require tetapi perbedaannya file yang dipanggil hanya akan digunakan 1 kali saja.

```
require_once "hal_fungsi.php";
```

Mengolah Form

Mengakses Data dari Form

Proses input data dari client ada biasanya menggunakan form sebagai media pengiriman data dan media input dari client. Untuk mengakses data yang dikirim oleh form, PHP sebagai server side programming akan menangkap data yang ditembakkan pada suatu URL tertentu, dimana URL tersebut merupakan tempat untuk mengeksekusi / menangkap data yang dikirimkan. Proses untuk menangkap data dengan menampung suatu data pada suatu variable dengan memperhatikan metoda pengiriman data dari form apakah berbentuk POST atau GET.

Berikut ini contoh bentuk Form untuk mengirimkan data :

```
<form action="" method="POST">
    Username : <input name="username" type="text"><br>
    Password : <input name="password" type="password"><br>
    <input name=" kirim" type="submit" value=" kirim">
</form>
```

Pada contoh form akan menangkap data username dan password yang akan dikirimkan untuk diproses di file yang sama (dikarenakan pada parameter action tidak diisi oleh nama file) dengan menggunakan method "POST". Berikutnya adalah contoh untuk menangkap data dari form

```
<?php

if(isset($_POST[' kirim'])){

    $uname = $_POST['username'];
    $upass = $_POST['password'];

    echo "username yang login : ".$uname."<br>";
    echo "password yang dimasukkan ".$upass."<br>";

}else{

?>

<form action="" method="POST">
    Username : <input name="username" type="text"><br>
    Password : <input name="password" type="password"><br>
    <input name=" kirim" type="submit" value=" kirim">
</form>

<?php
}
?>
```

Proses penangkapan data dapat menggunakan \$_POST jika menggunakan method POST, \$_GET jika form menggunakan method GET dan \$_FILES untuk menangkap jenis data berbentuk file. yang akan dijelaskan pada subbab berikutnya.

Membuat Form Registrasi

Setelah mampu membuat sebuah form sederhana, langkah berikutnya adalah contoh untuk membuat form registrasi dengan data yang lebih banyak. Dengan contoh dibawah ini sudah cukup untuk memahami bagaimana cara membuat sebuah form tanpa upload file.

```
<?php

if(isset($_POST[' kirim'])) {

    $uname = $_POST['username'];
    $upass = $_POST['password'];
    $email = $_POST['email'];
    $sex = $_POST['sex'];
    $alamat = $_POST['alamat'];

    echo "username yang login : ".$uname."<br>";
    echo "password yang dimasukkan ".$upass."<br>";
    echo "email yang dimasukkan ".$email."<br>";

    if($sex=="1") {
        echo "sex yang dimasukkan laki-laki<br>";
    } else {
        echo "sex yang dimasukkan wanita<br>";
    }

    echo "alamat yang dimasukkan ".$alamat."<br>";

} else {

?>

<h1>Form Registrasi Sederhana</h1>

<form action="" method="POST">
    Username : <input name="username" type="text"><br>
    Password : <input name="password" type="password"><br>
    Email : <input name="email" type="text"><br>
    Sex : <input type="radio" name="sex" value="1"> Laki-laki <input type="radio"
name="sex" value="p"> Perempuan<br>
    Alamat : <textarea name="alamat"></textarea><br>
    <input name=" kirim" type="submit" value=" kirim">
</form>

<?php
}
?>
```

Output

```
username yang login : robbi  
password yang dimasukkan 123456  
email yang dimasukkan obiesan@gmail.com  
sex yang dimasukkan laki-laki  
alamat yang dimasukkan jl.veteran
```

Cek Data dengan Form *Validation*

Data yang sudah diinput kemudian ditangkap oleh server sebaiknya harus dilakukan validasi terlebih dahulu. Proses validasi data dapat dilakukan dengan dua cara pertama validasi di client side dengan menggunakan javascript dan di server side dengan menggunakan PHP. Pada subbab ini akan mencoba membuat validasi dengan server side

```
<?php  
  
if(isset($_POST[' kirim'])) {  
  
    $uname = $_POST['username'];  
    $upass = $_POST['password'];  
    $email = $_POST['email'];  
    $sex = $_POST['sex'];  
    $alamat = $_POST['alamat'];  
  
    if (($uname!="") and ($upass!="") and ($email!="")) {  
  
        if (filter_var($email, FILTER_VALIDATE_EMAIL)) {  
  
            echo "username yang login : ".$uname."<br>";  
            echo "password yang dimasukkan ".$upass."<br>";  
            echo "email yang dimasukkan ".$email."<br>";  
  
            if($sex=="1") {  
                echo "sex yang dimasukkan laki-laki<br>";  
            }else{  
                echo "sex yang dimasukkan wanita<br>";  
            }  
  
            echo "alamat yang dimasukkan ".$alamat."<br>";  
        }else{  
            echo "Email salah format";  
        }  
  
    }else{  
        echo "Inputan name, password dan email wajib diisi";  
    }  
}else{  
  
?>  
<h1>Form Registrasi Sederhana</h1>  
<form action="" method="POST">  
    Username : <input name="username" type="text"><br>  
    Password : <input name="password" type="password"><br>  
    Email : <input name="email" type="text"><br>
```

```

        Sex : <input type="radio" name="sex" value="1"> Laki-laki <input type="radio"
name="sex" value="p"> Perempuan<br>
        Alamat : <textarea name="alamat"></textarea><br>
        <input name=" kirim" type="submit" value=" kirim">
</form>

<?php
}
?>

```

Output jika inputan tidak diisi

Inputan name, password dan email wajib diisi

Output jika inputan telah diisi semua tetapi format email salah

Email salah format

Membuat Form Upload File

Saat ini form tidak hanya mengirimkan data berbasis teks saja tetapi juga mengirimkan sebuah file untuk diupload dan disimpan diserver. Berikut ini adalah contoh script untuk mengupload file

```

<?php

if(isset($_POST['submit'])){

    $allowedExts = array("gif", "jpeg", "jpg", "png");
    $temp = explode(".", $_FILES["file"]["name"]);
    // perintah untuk memisahkan nama dan ekstensi

    $extension = end($temp);

    if ((($_FILES["file"]["type"] == "image/gif")
    || ($_FILES["file"]["type"] == "image/jpeg")
    || ($_FILES["file"]["type"] == "image/jpg")
    || ($_FILES["file"]["type"] == "image/pjpeg")
    || ($_FILES["file"]["type"] == "image/x-png")
    || ($_FILES["file"]["type"] == "image/png"))
    && ($_FILES["file"]["size"] < 2000000)
    && in_array($extension, $allowedExts))
    {
        //$_FILES["file"]["type"] untuk mengambil type sebuah file

        if ($_FILES["file"]["error"] > 0){
            echo "Return Code: " . $_FILES["file"]["error"] . "<br>";
        }else
        {
            echo "Upload: " . $_FILES["file"]["name"] . "<br>"; //Mengambil nama file
            echo "Type: " . $_FILES["file"]["type"] . "<br>"; //Mengambil tipe file
            echo "Size: " . ($_FILES["file"]["size"] / 1024) . " kB<br>";
            echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br>";
        }
    }
}

```

```

        if (file_exists("upload/" . $_FILES["file"]["name"]))
        {
            echo $_FILES["file"]["name"] . " already exists. ";
        }
        else
        {
            move_uploaded_file($_FILES["file"]["tmp_name"],
            "upload/" . $_FILES["file"]["name"]);
            echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
        }
    }
}
else
{
    echo "Invalid file";
}

}else{

?>
<html>
<body>

<form action="" method="post" enctype="multipart/form-data">
    <label for="file">Filename:</label>
    <input type="file" name="file" id="file"><br>
    <input type="submit" name="submit" value="Submit">
</form>

</body>
</html>

<?php
}
?>

```

Sessions dan Cookies

Session

Apa itu session? mungkin bagi para pengguna Internet yang rajin browsing terutama yang menggunakan internet banking sudah sering mendengar kata “session expire” yang mengakibatkan kita otomatis logout dan kembali ke halaman untuk login ulang. Session adalah merupakan penyimpanan data sementara yang disimpan di server web, biasanya penggunaan session yang paling sering digunakan untuk penyimpanan data user sesudah melakukan proses autentifikasi.

Membuat, Mengakses dan Menghapus Session

Berikut ini contoh untuk membuat dan mengakses sebuah session

```
<?php
session_start();
// store session data
$_SESSION['views']=1;
?>

<html>
<body>

<?php
//retrieve session data
echo "Pageviews=". $_SESSION['views']; //output pageviews = 1
?>

</body>
</html>
```

```
<?php
session_start();

if(isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views']+1;
else
$_SESSION['views']=1;
echo "Views=". $_SESSION['views'];
?>
```

Untuk membuat dan mengakses session dapat dilakukan secara sederhana seperti membuat sebuah variabel dengan terlebih dahulu memanggil session_start() kemudian menggunakan variabel \$_SESSION['nama session']

Menghapus session yang sudah tidak digunakan

```
<?php
session_start();
if(isset($_SESSION['views']))
    unset($_SESSION['views']);
?>
```

Cookie

Cookie sebenarnya berfungsi seperti session, tetapi penyimpanan file berada di client biasanya disimpan pada file temporary sebuah browser. File yang disimpan pada cookie berukuran lebih kecil dibandingkan dengan session. Cookie lebih rentan untuk penyimpanan data-data yang bersifat krusial ataupun rahasia karena disimpan di sisi client.

Membuat, Mengakses dan Menghapus Cookie

Berikut ini contoh untuk membuat sebuah cookie

```
<?php
setcookie("user", "Alex Porter", time()+3600);
?>
```

Menggunakan fungsi setcookie untuk membuat cookie sekaligus mendefinisikan waktu timeoutnya

Contoh untuk mengakses cookie yang sudah dibuat

```
<html>
<body>

<?php
if (isset($_COOKIE["user"]))
    echo "Welcome " . $_COOKIE["user"] . "!\n";
else
    echo "Welcome guest!\n";
?>

</body>
</html>
```

Untuk mengakses dapat menggunakan variabel \$_COOKIE

Menghapus cookie yang sudah tidak digunakan

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time()-3600);
?>
```

Untuk menghapus dapat menggunakan dengan cara mengeset waktu minus

Bermain dengan File

Mengakses File

Pada bagian ini, *file* yang akan diakses adalah *file* bertipe .txt. Sebelum menggunakan beberapa fungsi *built-in* PHP untuk menangani *file* , buatlah sebuah *file* .txt dengan nama **myfile.txt** dan simpan ke salah satu Drive yang ada pada komputer. Pada contoh kali ini, *file* teks tersebut akan disimpan pada :

```
D:\myfile.txt
```

Isi dari *file* teks tersebut adalah :

```
1. Joko
2. Anton
3. Budi
```

PHP memiliki fungsi *built-in* untuk mengakses sebuah *file* yaitu `fopen(filename, mode)`. Fungsi ini memiliki dua parameter :

1. *filename* : Parameter ini berisi nama *file* atau *pathfile lengkap* yang akan diakses.
2. *mode* : Parameter ini berisi mode yang akan dijalankan pada saat mengakses sebuah *file*.

Berikut ini adalah mode yang dapat dilakukan pada saat mengakses *file* :

Mode	Deskripsi
r	Read only. Dimulai dari awal file.
r+	Read/Write. Dimulai dari awal file.
w	Write only. Digunakan untuk membuat file baru. Mode ini akan membuka dan mengosongkan isi file. Jika file belum ada, dengan mode ini, sebuah file baru akan dibuat.
w+	Read/Write. Digunakan untuk membuat file baru. Mode ini akan membuka dan mengosongkan isi file. Jika file belum ada, dengan mode ini, sebuah file baru akan dibuat.
a	Append. Menuliskan data pada akhir sebuah <i>file</i> . Jika <i>file</i> belum ada, dengan mode ini, sebuah <i>file</i> baru akan dibuat.

a+	Read/Append. Membuka <i>file</i> agar dalam kondisi siap ditulis data baru pada akhir <i>file</i> .
x	Write Only. Membuat sebuah <i>file</i> baru dan akan mengembalikan nilai FALSE jika <i>file</i> sudah ada.
x+	Read/Write. Membuat sebuah <i>file</i> baru dan akan mengembalikan nilai FALSE jika <i>file</i> sudah ada.

Berikut ini adalah contoh kode PHP untuk membuka sebuah *file* yang disimpan di `D:\myfile.txt` :

```
<?php
    $filename = "D:\\myfile.txt";
    $file = fopen($filename,"r");
    fclose($file);
?>
```

Membaca File Perbaris

Untuk membaca isi dari sebuah *file* per baris, diperlukan sebuah fungsi **fgets** (mengambil isi *file* per baris) dan **feof** (mendeteksi akhir *file*). Dengan fungsi ini, data pada sebuah *file* akan terus dicetak sebelum proses pembacaan mencapai akhir *file*. Berikut ini adalah contoh kode untuk membaca isi *file* perbaris :

```
<?php
    $filename = "D:\\myfile.txt";
    $file = fopen($filename,"r");
    while(!feof($file))
    {
        echo fgets($file)."<br/>";
    }
    fclose($file);
?>
```

Output :

```
1.Joko
2.Anton
3.Budi
```

Membaca File Per karakter

PHP juga memiliki fungsi yang dapat membaca isi *file* per karakter yaitu **fgetc**. Berikut ini adalah contoh kode PHP untuk membaca isi *file* per karakter :

```
<?php
    $filename = "D:\\myfile.txt";
    $file = fopen($filename,"r");
    while(!feof($file))
    {
        echo fgetc($file). "<br/>";
    }
    fclose($file);
```

Output :

```
1
.
J
o
k
o
...
```

Menulis File

Untuk menuliskan sebuah data ke sebuah file, kita dapat menggunakan fungsi `fwrite()`. Berikut adalah contoh penggunaan fungsi `fwrite()` :

```
<?php
    $filebaru = "D:\\newfile.txt";
    $handle = fopen($filebaru, "w" );
    fwrite( $handle, "Isi File Baru" );
?>
```

Output :



File Permissions

File permissions adalah sebuah izin yang melekat pada sebuah file. File permission ini akan menentukan apakah sebuah *file* itu hanya dapat dibaca saja atau dapat dibaca dan dimodifikasi. PHP memiliki fungsi **chmod** yang dapat digunakan untuk mengubah permission sebuah file. Berikut ini adalah contohnya :

```
<?php
    $file = "D:\\newfile.txt";
    chmod($file,0755);
?>
```

Fungsi chmod diatas dapat berfungsi dengan baik pada operating sistem linux.

Menduplikasi, Mengubah Nama dan Menghapus File

Untuk menduplikasi, mengubah nama dan menghapus file memerlukan tiga fungsi PHP yaitu :

- copy();
- rename();
- unlink();

Berikut ini adalah contoh penggunaan ketiga fungsi diatas :

```
<?php
    $file = "D:\\newfile.txt";

    $copy = "D:\\copyfile.txt";
    $copytorename = "D:\\copytorename.txt";
    $copytodel = "D:\\copytodel.txt";

    $rename = "D:\\renfile.txt";

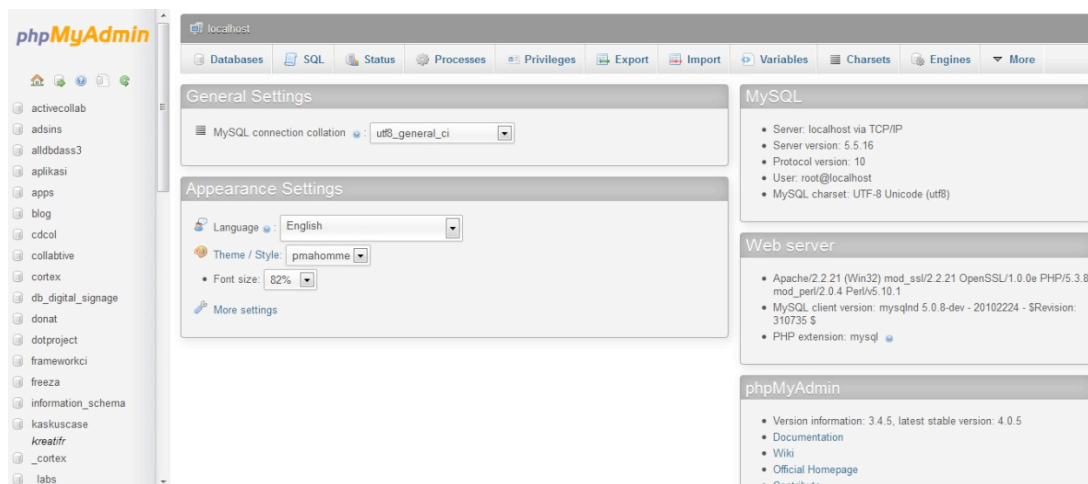
    copy($file,$copy);
    copy($file,$copytorename);
    copy($file,$copytodel);

    rename($copytorename,$rename);
    unlink($copytodel);
```

Outputnya adalah menghasilkan 4 buah file

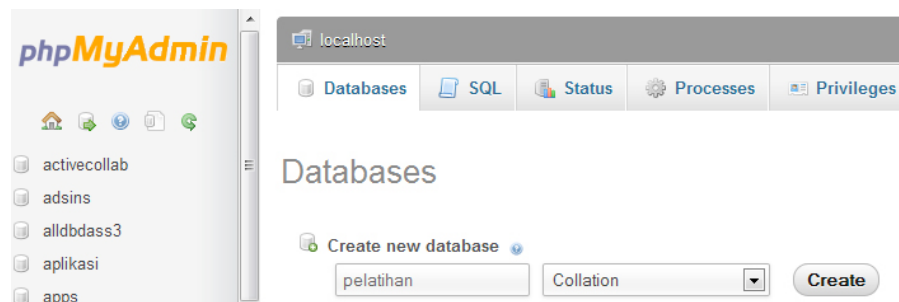
PHP dan MySQL

Untuk mengakses MySQL dengan PHP kita memerlukan dua fungsi PHP yaitu **mysql_connect()** dan **mysql_select_db()**. Fungsi **mysql_connect()** digunakan untuk membuka koneksi MySQL. Sedangkan **mysql_select_db()** digunakan untuk memilih database yang akan digunakan. Sebelum melakukan koneksi ke MySQL, buka browser terlebih dahulu dan akses URL : <http://localhost/phpmyadmin>.



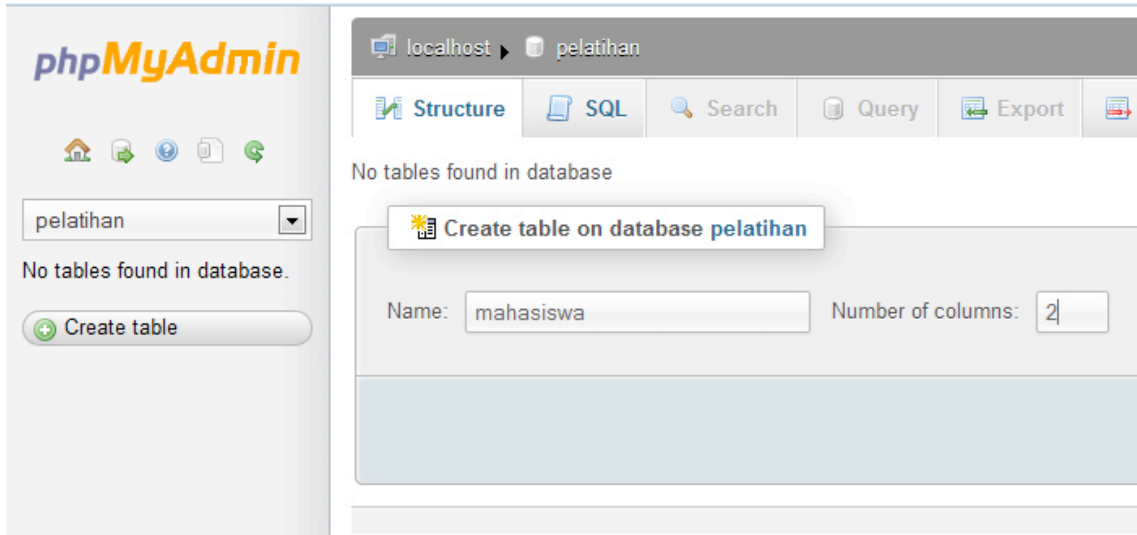
Gambar. phpmyadmin

Buatlah database baru dengan nama pelatihan seperti gambar berikut ini :



Gambar. Membuat Database

Setelah membuat database pelatihan, buatlah satu tabel mahasiswa yang terdiri dari 2 field, yaitu nim dan nama seperti pada gambar berikut ini :



Gambar. Membuat Tabel Mahasiswa

Table name:

mahasiswa

Structure

Column	nim	nama
Type	VARCHAR	VARCHAR
Length/Values ¹	8	30
Default ²	None	None
Collation		
Attributes		

Gambar. Struktur Tabel Mahasiswa

Setelah disimpan, buatlah sebuah file **mahasiswa.php** yang berisi kode berikut ini :

```
<table width="300px" border="1">
<tr>
    <th>No</th>
    <th>Nim</th>
    <th>Nama</th>
    <th>Action</th>
</tr>
<?php
    $conn = mysql_connect("localhost","root","");
    if($conn)
    {
        mysql_select_db("pelatihan");
    }else{
        echo "error";
    }

    $sql = "SELECT * FROM mahasiswa";
    $hasil = mysql_query($sql);

    if(mysql_num_rows($hasil) > 0)
    {
        $no = 1;
        while($data = mysql_fetch_array($hasil))
        {
            ?>
<tr>
    <td><?php echo $no; ?></td>
    <td><?php echo $data['nim']; ?></td>
    <td><?php echo $data['nama']; ?></td>
    <td>
        <a href="">Edit</a> | <a href="">Delete</a>
    </td>
</tr>
<?php
    $no++;
    }
    }else{
        echo "<tr><td colspan='4'>Data belum tersedia !!!</td></tr>";
    }
    ?>
</table>
```


Output (Jika data belum ada) :

No	Nim	Nama	Action
Data belum tersedia !!!			

Output (Jika data ada) :

No	Nim	Nama	Action
1	30110217	Aldilla	Edit Delete
2	30110265	Yugo	Edit Delete

Form Tambah Mahasiswa

Gambar berikut ini adalah gambar form tambah mahasiswa :

Tambah Mahasiswa

Nim

Nama

Berikut ini adalah kode untuk membuat form tambah mahasiswa :

```
<h3>Tambah Mahasiswa</h3>
<form action="mahasiswa" method="post">
<table>
  <tr>
    <td>Nim</td>
    <td><input type="text" name="nim" /></td>
  </tr>
  <tr>
    <td>Nama</td>
    <td><input type="text" name="nama" /></td>
  </tr>
  <tr>
    <td><input type="submit" name="btn-simpan" value="simpan" /></td>
  </tr>
</table>
```

Kode PHP untuk Menambah Data

```
<?php
    $nim = $_POST['nim'];
    $nama = $_POST['nama'];

    $sql = "INSERT INTO mahasiswa VALUES ('$nim', '$nama')";
    mysql_query($sql);
?>
```

Kode PHP untuk Memodifikasi Data

```
<?php
    $nim = $_POST['nim'];

    $sql = "UPDATE mahasiswa SET nama = 'Andi' WHERE nim = '$nim';
    mysql_query($sql);
?>
```

Kode PHP untuk Menghapus Data

```
<?php
    $nim = $_POST['nim'];

    $sql = "DELETE FROM mahasiswa WHERE nim = '$nim'";
    mysql_query($sql);
?>
```

Object Oriented dengan PHP

Dasar OOP pada PHP

Dalam memahami Konsep Object Oriented, kita perlu memahami terlebih dahulu mengenai :

1. **Kelas.** adalah sebuah kerangka, pola atau model umum dari sesuatu. Kerangka, pola atau model yang digambarkan oleh sebuah kelas merupakan karakteristik (*characteristic*) dan perilaku (*behaviour*) dari sesuatu yang akan kita buat. Kelas bisa juga kita sebut sebagai cetak biru (*blueprint*) dari sesuatu.
2. **Objek.** Adalah sebuah *instance* spesifik dari sebuah kelas.
3. **Property.** Dalam konteks OOP, property adalah sebuah variabel yang dideklarasikan didalam sebuah kelas.
4. **Method.** Pada sebuah kelas, mirip dengan fungsi. Method menggambarkan aksi apa saja yang bisa dilakukan oleh sebuah kelas. Pada kenyataannya, method kelas di PHP didefinisikan dengan menggunakan pernyataan function.

Berikut ini adalah contoh sederhana dari sebuah kelas PHP :

```
<?php
    class Mobil {

    }

    $m1 = new Mobil();
    $m2 = new Mobil();
?>
```

Contructors dan Destructors pada PHP

Contructors adalah sebuah method dengan nama khusus dan akan selalu dieksekusi pertama kali ketika di-*instance*. Berikut ini adalah contoh kode untuk membuat contructors di PHP :

```
<?php
    class MyClass {
        function __construct() {
            echo "Tes Constructor ";
        }
    }
    $obj = new MyClass;
?>
```

Destructors adalah sebuah method untuk membersihkan objek yang telah dibuat. Berikut adalah contoh kode untuk membuat destructors di PHP :

```
<?php
    class MyClass {
        public function save() {
            echo "Menyimpan object ";
        }

        public function __destruct() {
            $this->save();
        }
    }
?>
```

Kenali “Visibility” pada OOP di PHP

Pada OOP terdapat tiga tipe visibility yang akan digunakan untuk properti, yaitu :

1. *Public Property*. Semua property yang dideklarasikan public berarti dapat diakses oleh siapapun. Baik oleh kelas sendiri maupun kelas lain.
2. *Private Property*. Semua property yang didklarasikan private, hanya dapat diakses dari dalam kelas saja.
3. *Protected Property*. Semua property yang didklarasikan protected, hanya dapat diakses dari luar kelas yang masih dalam satu package atau kelas turunannya.

Bermain dengan Methods

Berikut ini adalah contoh kode PHP yang menjelaskan mengenai penggunaan sebuah kelas, property dan method :

```
<?php
class Product{

    public $name = 'nama produk';
    public $price = 0;
    protected $discount = 0; //persen

    public function __construct($productName, $price){
        $this->name = $productName;
        $this->price = $price;
    }

    public function getDiscount(){
        return $this->discount; //mengakses properti
    }

    public function getPrice(){
        $price = $this->price - ($this->price * $this->getDiscount());
        return $price;
    }

    public function setDiscount($discount){
        $this->discount = $discount; // setting nilai properti
    }
}

$product = new Product('Buku Belajar PHP Object', 99.5);
echo $product->name;
//output 'Buku Belajar PHP Object' bukan 'default product name'

echo $product->getPrice();
//output '99.5' bukan '0'

?>
```

Inheritance atau Pewarisan

Inheritance atau Pewarisan adalah sebuah mekanisme dimana satu atau lebih class diturunkan dari sebuah class lain. Class yang mewarisi atau diturunakn dari class lainnya disebut subclass dari class yang diwarisi. Berikut ini adalah contoh kode yang mengimplementasi inheritance :

```

<?php
class CDMusik extends Product{

    public $artis;
    public $durasi = 0;
    public $daftarLagu = array();

    function tambahLagu($lagu){
        if(is_array($lagu)){
            $this->daftarLagu = array_merge($this->daftarLagu, $lagu);
        }else{

            $this->daftarLagu[] = $lagu;
        }
    }

    function setDurasi($durasi){
        $this->durasi = $durasi;
    }

    function setArtis($artis){
        $this->artis = $artis;
    }

}

$cdAndra = new CDMusik('Andra and The Backbone', 40);
$cdAndra->tambahLagu('Main Hati');
$cdAndra->tambahLagu('Hitamku');
$cdAndra->setArtis('Andra');
$cdAndra->setDurasi(60);

echo 'Artis: '.$cdAndra->artis;
echo '<br/>Durasi: '.$cdAndra->durasi;
echo '<br/>Lagu: ';
echo '<ol>';
foreach($cdAndra->daftarLagu as $lagu){
    echo '<li>'.$lagu.'</a>';
}
echo '</ol>';?>

```