

eBook PHP5

Easy Step to Learn PHP5

Bahasa Indonesia



Imam Ali Mustofa

Mengenai Hak Cipta

Segala hak cipta termasuk cover adalah milik penulis, kecuali yang disebutkan.

Sejarah Singkat PHP



PHP adalah singkatan dari "**PHP: Hypertext Preprocessor**", yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan **HTML**. **PHP** diciptakan oleh *Rasmus Lerdorf* pertama kali

tahun 1994. Pada awalnya **PHP** adalah singkatan dari "*Personal Home Page Tools*". Selanjutnya diganti menjadi FI ("*Forms Interpreter*"). Sejak versi 3.0, nama bahasa ini diubah menjadi "**PHP: Hypertext Preprocessor**" dengan singkatannya "**PHP**". PHP versi terbaru adalah versi ke-5. Berdasarkan survey Netcraft pada bulan Desember 1999, lebih dari sejuta site menggunakan PHP, di antaranya adalah *NASA*, *Mitsubishi*, dan *RedHat*. (Sumber: *wikibuku*)

Pengertian PHP Menurut Ahli dalam Bukunya

PHP

yang merupakan bahasa program yang bersifat compiler. Definisi **PHP** menurut *Kasiman Peranginangin* dalam bukunya yang berjudul ***Aplikasi Web dengan PHP dan MySQL*** menjelaskan bahwa "**PHP** singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa scrip server-side dalam pengembangan web yang disisipkan pada dokumen **HTML**" (Kasiman, 2006:2)

Definisi lain dari **PHP** menurut *Anhar*, menjelaskan bahwa "*php adalah script yang digunakan untuk membuat halaman website yang dinamis*" (2010:03). Berdasarkan definisi di atas maka penulis menyimpulkan bahwa **PHP** adalah bahasa pemrograman yang digunakan sebagai bahasa dalam membangun atau mengembangkan sebuah web. (**Google**).

Intro

PHP adalah server bahasa pemrograman yang nantinya script / program tersebut akan dijalankan atau dieksekusi oleh server, dan alat yang ampuh untuk membuat halaman Web dinamis serta interaktif. **PHP** adalah alternatif yang digunakan secara luas, bebas, dan efisien sama halnya seperti **Microsoft ASP**.

Skrip PHP yang dijalankan oleh server.

Apa yang harus Anda ketahui sebelum melanjutkan tutorial ini? Utamanya anda harus memiliki pemahaman dasar sebagai berikut:

- HTML
- CSS
- JavaScript

Jika anda belum mengerti tentang beberapa mata pelajaran di atas, anda dapat melihatnya di [Tutorial HTML](#) dan [Tutorial CSS](#) (untuk sementara, **Javascript** belum kami tulis artikelnnya).

Apa itu PHP?

PHP adalah akronim dari "PHP: Hypertext Preprocessor". PHP adalah bahasa scripting yang digunakan secara luas dan terbuka. Skrip PHP yang dijalankan pada server, PHP bebas untuk didownload dan digunakan.

PHP adalah Bahasa yang Menakjubkan dan Populer!

Hal ini cukup kuat untuk berada di inti dari sistem blogging terbesar di web (**WordPress**) dan sosial media seperti (**Facebook**)! PHP juga cukup mudah untuk dipelajari bagi seorang pemula!

Apa itu PHP File?

File PHP dapat berisi teks, **HTML**, **CSS**, **JavaScript**, dan kode PHP. Kode PHP yang dijalankan pada server, dan hasilnya akan ditampilkan ke browser sebagai plain HTML. File PHP memiliki ekstensi ".php". Sebagai contoh lihat gambar di bawah:



```
<!DOCTYPE html>
<html>
<body>

<?php
    echo "Ini script PHPku yang Petama!";
?>

</body>
</html>
```

Apa yang Bisa Anda Lakukan dengan PHP?

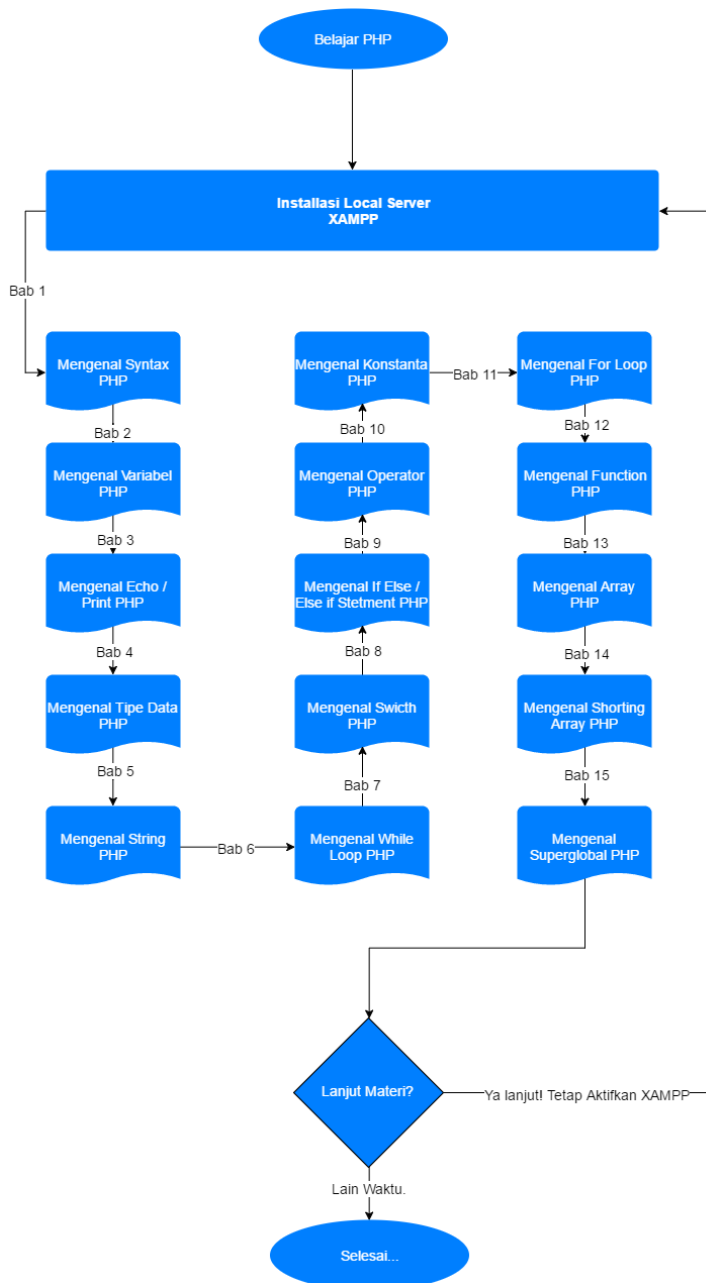
- PHP dapat menghasilkan konten dinamis,
- PHP dapat membuat, membuka, membaca, menulis, menghapus dan menutup file di server,
- PHP dapat mengumpulkan data formulir,
- PHP dapat mengirim dan menerima cookie
- PHP dapat menambah, menghapus, mengubah data dalam database,
- PHP dapat digunakan untuk mengontrol akses pengguna,
- PHP dapat mengenkripsi data,
- Dengan PHP tidak terbatas untuk output HTML. Anda bisa output gambar, file PDF, dan bahkan Flash film. Anda juga bisa output teks, seperti XHTML dan XML.
-

Mengapa Menggunakan PHP?

- PHP berjalan pada berbagai platform (Windows, Linux, Unix, Mac OS X, dll.).
- PHP kompatibel dengan hampir semua server digunakan hingga hari ini seperti (Apache, IIS, dll.).
- PHP mendukung berbagai database.
- PHP gratis. Download dari sumber PHP resmi: www.php.net
- PHP mudah dipelajari dan berjalan efisien pada sisi server.

Langkah selanjutnya setelah kita mengenal sedikit tentang PHP, sekarang kita akan mengetahui langkah – langkah dan proses yang akan kita lakukan dalam eBook ini. Berhubung dengan tidak adanya “*Daftar Isi*” yang menyertakan nomor halaman untuk setiap babnya, sebenarnya penulis memiliki tujuan sederhana kenapa tidak menyertakan daftar isi pada eBook ini, tujuannya adalah sebagai berikut:

- Penulis berharap kepada pembaca untuk tidak melewatkan setiap langkah demi langkah dalam proses belajar.
 - Penulis menginginkan situasi belajar lebih terfokus, sehingga pembaca bisa memahami apa materi yang ditulis dan disampaikan dalam eBook ini.
-



Download Dan Install XAMPP Sebagai Local Server

Seperti yang kami sampaikan pada tulisan sebelumnya yaitu [Intro](#) pada kesempatan kali ini kita akan belajar untuk menginstall **XAMPP** di Windows (kebetulan penulis artikel ini menggunakan Windows), untuk menjalankan script **PHP** dibutuhkan Server yang akan mengeksekusi script, kebanyakan pemrogram atau developer **PHP** menggunakan Local Server sebagai compiler dan juga menggunakan **MySQL** sebagai database, kedua fitur tersebut didownload dan digunakan tanpa harus beli alias gratis tis tis tis.

Anda akan repot jika harus menginstallnya satu per satu, belum lagi install software pengelola database MySQL seperti phpMyAdmin. Nah masalah tersebut bisa ditangani dengan hadirnya XAMPP yang merupakan paket server lengkap.

Apa itu XAMPP?

Apa saja yang terdapat dalam paket XAMPP ini? Kita bahas sedikit tentang XAMPP terlebih dahulu, supaya anda dan saya ingat bahwa ilmu yang bermanfaat dan ilmu yang berguna adalah ilmu yang kita dapat tidak percuma alias mengerti apa makna dan tujuannya.

XAMPP

XAMPP merupakan singkatan dari *Cross-Platform* (X), *Apache* (A), *MySQL* (M), *PHP* (P) dan *Perl* (P). XAMPP didesain lengkap dengan tujuan memudahkan para Web Developer mendapat Local Server sebagai tempat pengujian.

XAMPP menyediakan semua yang anda butuhkan untuk membuat sebuah web. Server aplikasi (*Apache*), database (*MySQL*), dan bahasa scripting (*PHP*). XAMPP juga cross-platform, yang berarti bisa bekerja sama dengan baik di Linux, Mac dan Windows.

Karena sebagian besar penyebaran web server yang sebenarnya (secara online) menggunakan komponen yang sama seperti

XAMPP, sehingga membuat transisi dari server lokal ke server online sangat mudah.

Download XAMPP untuk Windows

Jika Anda ingin men-download yang versi 1.8.3, Anda dapat men-download instaler XAMPP versi tersebut dengan mengunjungi url download dibawah ini:

www.bettadev-tutorial.blogspot.com/download/xampp-installer.exe

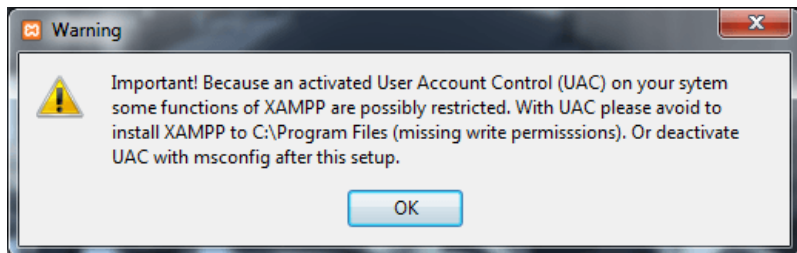
Atau anda ingin men-download yang versi terbaru dengan mengunjungi url halaman:

<https://www.apachefriends.org/download.html>

Install XAMPP di Windows

Langkah 1: Non-aktifkan anti-virus karena dapat menyebabkan beberapa komponen XAMPP tidak bisa di Install dengan lancar.

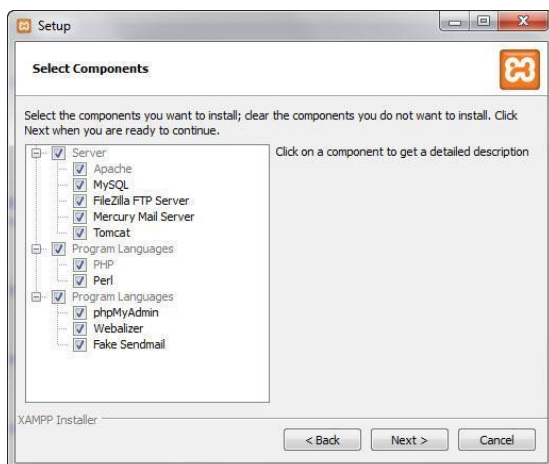
Langkah 2: Untuk pengguna Windows 7, Anda akan melihat jendela pop up, peringatan tentang User Account Control (UAC) yang aktif pada sistem. Klik "OK" untuk melanjutkan instalasi.



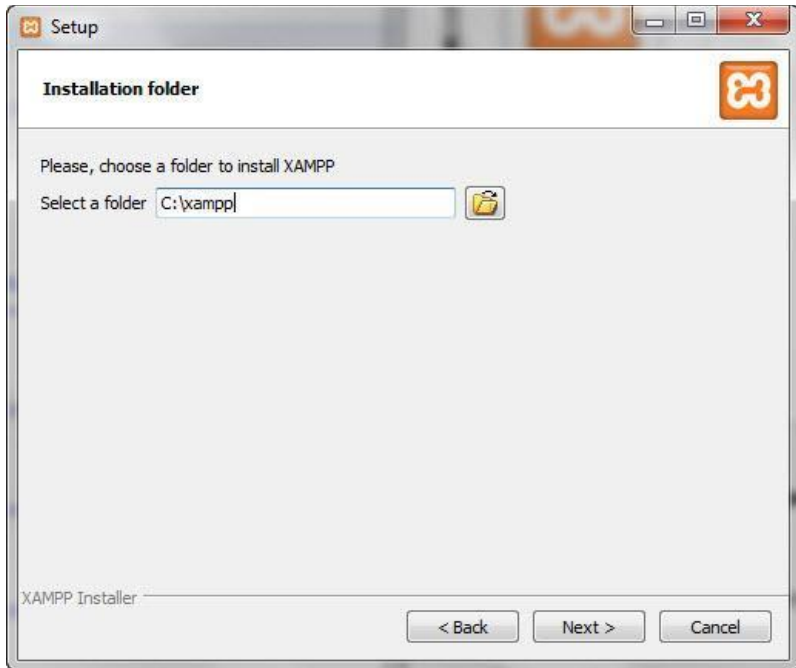
Langkah 3: Mulai proses instalasi dengan klik dua kali pada instaler XAMPP. Klik 'Next' setelah splash screen.



Langkah 4: Di sini, kita dapat memilih komponen yang ingin kita instal. Pilih pilihan default dan klik 'Next'.



Langkah 5: Pilih folder sebagai tempat XAMPP akan diinstal, di folder ini akan menampung semua file aplikasi web kita, jadi pastikan untuk memilih drive yang memiliki banyak ruang(space).



Langkah 6: Layar berikutnya adalah promo untuk BitNami, sebuah toko aplikasi untuk server perangkat lunak. Hapus kotak centang 'Learn more about BitNami for XAMPP'.



Langkah 7: Sekarang Setup sudah siap untuk menginstall XAMPP. Klik Next dan tunggu instaler untuk membongkar paket-nya dan memasang komponen yang dipilih. Mungkin memakan waktu beberapa menit. Nanti mungkin kita akan diminta untuk menyetujui akses Firewall untuk komponen tertentu (seperti Apache) selama proses instalasi.

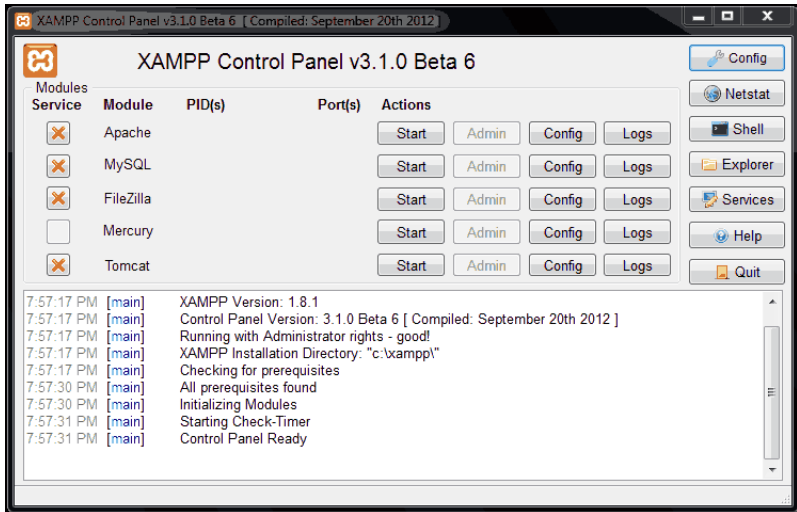


Langkah 8: Proses Install sudah selesai! Pilih Kotak centang 'Do you want to start the Control Panel now?' untuk membuka panel kontrol XAMPP.

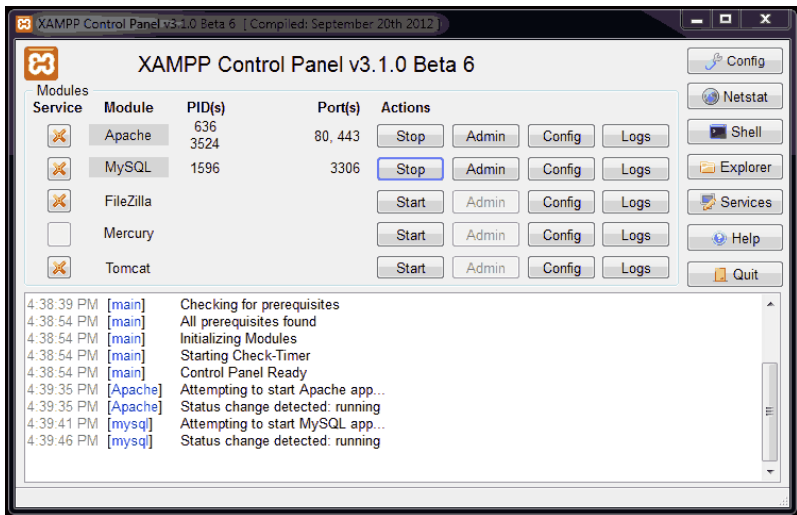
Cara menggunakan XAMPP Control Panel

XAMPP control panel memberi kita kontrol penuh atas semua komponen XAMPP yang telah di Install. Kita dapat menggunakan Control Panel untuk memulai / menghentikan modul yang berbeda, meluncurkan Unix shell, membuka Windows explorer sampai melihat semua operasi yang sedang berjalan di balik layar.

Berikut ini adalah gambaran singkat dari Control Panel. Untuk saat ini, kita hanya perlu tahu bagaimana untuk memulai dan menghentikan server Apache.



Klik Start untuk memulai XAMPP Control Panel.



Dan sekarang XAMPP Local Server anda sudah berjalan.

Referensi <http://www.nyekrip.com/cara-install-xampp-web-server-di-windows/> .



Bab 1

PHP Syntax

Basic PHP

PHP adalah bahasa pemrograman yang dieksekusi langsung oleh server. Membuat file atau script **PHP** cukup lah mudah. Hanya dengan **<?php** dan ditutup **?>** anda sudah bisa menuliskan script **PHP** diantara start dan end elemen tersebut. Sebagai contoh:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Menenal Sintaks PHP</title>
5  </head>
6  <body>
7
8    <h1>My first PHP page</h1>
9
10   <?php
11     echo "Hello World!";
12   ?>
13
14   </body>
15   </html>
```

Dan jika anda ingin menambahkan komentar pada PHP anda cukup menggunakan `"// "` atau `"/ * komentar */ "` dan komentar tidak akan dieksekusi oleh server. mudah bukan?! tentu mudah.. Next kita bahas **Variable** dalam **PHP**. Contohnya:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Mengenal Sintaks PHP</title>
5 </head>
6 <body>
7
8 <h1>My first PHP page</h1>
9
10 <?php
11 echo "Hello World!"; //Ini adalah Komentar satu baris
12
13 /*
14 Ini adalah
15 komentar
16 multi baris
17 */
18 ?>
19
20 </body>
21 </html>

```

Sebelum script dieksekusi atau dicompile / di-ru pada browser. Perlu diperhatikan sebagai catatan! Anda wajib menyimpan file / script PHP pada folder **htdocs** yang terletak pada folder **XAMPP** dimana anda menginstallnya. Simpan dengan ekstensi (.php)

Bab 2

PHP Variabel

Variabel adalah "wadah" untuk menyimpan informasi.

Membuat (deklarasi) Variabel PHP

Dalam PHP, variabel diawali **\$** tanda, dan diikuti oleh nama variabel:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Variable PHP</title>
5 </head>
6 <body>
7
8 <h2>Mendeklarasikan Variable</h2>
9
10 <?php
11 $hallo = "Hello world!";
12 $lima = 5;
13 $sepuluhkomalima = 10.5;
14
15 echo $hallo;
16 echo "<br>";
17 echo $lima;
18 echo "<br>";
19 echo $sepuluhkomalima;
20 ?>
21
22 </body>
23 </html>
```

Setelah mengeksekusi pernyataan di atas, variabel **\$txt** akan menampilkan nilai Hello world!, variabel **\$x** menampilkan nilai 5 dan variabel **\$y** akan menampilkan nilai 10.5.

Catatan: Bila Anda menetapkan nilai teks (string) ke variabel, gunakanlah tanda kutip. Tidak seperti bahasa pemrograman lain, PHP memiliki perintah untuk mendeklarasikan variabel. Hal ini dibuat saat pertama kali Anda menetapkan nilai itu.

{Pikirkan variabel sebagai wadah untuk menyimpan data.}

Variabel PHP

Sebuah variabel dapat memiliki nama pendek (seperti x dan y) atau nama lebih deskriptif (usia, carname, total_volume).

Aturan variabel PHP:

- Variabel diawali tanda \$ dan diikuti oleh nama variabel
- Nama variabel diawali dengan huruf atau karakter garis bawah
- Nama variabel tidak dapat mulai dengan angka
- Nama variabel hanya dapat mengandung karakter alfa-numerik dan garis bawah (AZ, 0-9 dan _)
- Nama variabel bersifat case-sensitive (\$age dan \$AGE adalah dua variabel yang berbeda)

{Ingat bahwa nama – nama variabel PHP bersifat case – sensitive!}

Variabel output

PHP echo pernyataan ini sering digunakan untuk menampilkan data ke layar. Contoh berikut akan menunjukkan bagaimana output variabel:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Variable PHP</title>
5  </head>
6  <body>
7  <h2>Menyisipkan Variable Output</h2>
8
9  <?php
10 $txt = "Beta Dev Indonesia";
11 echo "I love $txt!";
12 ?>
13
14 </body>
15 </html>

```

Contoh berikut akan menghasilkan output yang sama seperti contoh di atas:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Variable PHP</title>
5  </head>
6  <body>
7  <h2>Menyisipkan Variable Output</h2>
8
9  <?php
10 $txt = "Beta Dev Indonesia";
11 echo "I love " . $txt . "!";
12 ?>
13
14 </body>
15 </html>

```

Contoh berikut akan menampilkan jumlah dari dua variabel:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Variable PHP</title>
5 </head>
6 <body>
7
8 <h2>Menjumlahkan Dua Variable</h2>
9
10 <?php
11 $x = 5;
12 $y = 4;
13 echo "Hasil dari 5 + 4 = ";
14 echo $x + $y;
15 ?>
16
17 </body>
18 </html>
```

Catatan: Anda akan belajar lebih banyak tentang pernyataan echo dan bagaimana untuk output data ke layar dalam bab berikutnya.

PHP adalah Bahasa yang Mudah Diketik

Dalam contoh di atas, perhatikan kita tidak perlu memberitahu PHP variabel tipe data yang akan mengeluarkan hasil penjumlahan.

tidak perlu menuliskannya seperti ini, meskipun hasilnya tetap sama:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Variable PHP</title>
5 </head>
6 <body>
7
8 <h2>Menjumlahkan Dua Variable</h2>
9
10 <?php
11 $x = 5;
12 $y = 7;
13 $j = $x + $y;
14
15 echo "Hasil dari 5 + 4 = ".$j;
16
17 ?>
18
19 </body>
20 </html>
```

PHP secara otomatis mengkonversi variabel tipe data yang benar, tergantung pada nilai.

Dalam bahasa lain seperti C, C++ dan Java, programmer harus menyatakan nama dan jenis variabelnya sebelum digunakan.

Lingkup variabel PHP

Dalam PHP, variabel dapat dideklarasikan di mana saja dalam script. Lingkup variabel adalah bagian dari script variabel yang dapat dirujuk / digunakan.

PHP memiliki tiga cakupan variabel yang berbeda:

- lokal
- global
- statis

Lingkup global dan lokal

Sebuah variabel yang dinyatakan di luar fungsi memiliki lingkup **GLOBAL** dan hanya dapat diakses di luar fungsi:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Variable PHP</title>
5 </head>
6 <body>
7
8 <h2>Lingkup Global dan Local</h2>
9
10 <?php
11 $x = 5; // lingkup global
12
13 function xVariable() {
14     // menggunakan x dalam fungsi ini akan menghasilkan kesalahan
15     echo "<p>variabel x di dalam fungsi ini adalah: $x</p>";
16 }
17 xVariable();
18
19 echo "<p>variabel x di luar fungsi adalah: $x</p>";
20 ?>
21
22 </body>
23 </html>
```

dan variable sebaliknya dari global adalah **LOKAL**:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Variable PHP</title>
5  </head>
6  <body>
7
8  <h2>Lingkup Global dan Local</h2>
9
10 <?php
11 function xVariable() {
12     $x = 5; // lingkup local
13     echo "<p>Variable x di dalam fungsi ini adalah: $x</p>";
14 }
15 xVariable();
16
17 // menggunakan x dalam fungsi ini akan menghasilkan kesalahan
18 echo "<p>Variable x di luar fungsi adalah: $x</p>";
19 ?>
20
21 </body>
22 </html>
```

Catatan: Anda dapat memiliki variabel lokal dengan nama yang sama dalam fungsi yang berbeda, karena variabel lokal hanya diakui oleh fungsi di mana mereka dinyatakan.

PHP kata kunci global

Kata kunci **global** digunakan untuk mengakses variabel **global** dari dalam fungsi. Untuk melakukannya, gunakan kata kunci **global** sebelum variabel (di dalam fungsi):

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Variable PHP</title>
5  </head>
6  <body>
7
8  <h2>Lingkup Global Variable</h2>
9
10 <?php
11 $x = 5;
12 $y = 10;
13
14 function Testing() {
15     global $x, $y;
16     $y = $x + $y;
17 }
18
19 Testing();
20 echo $y; // outputnya 15
21 ?>
22
23 </body>
24 </html>
```

PHP juga menyimpan semua variabel **global** dalam **array** yang disebut **\$GLOBALS [indeks]**. Indeks memegang nama variabel. Array juga diakses dari dalam fungsi dan dapat digunakan untuk memperbarui variabel **global** secara langsung.

Contoh di atas dapat ditulis ulang seperti ini:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Variable PHP</title>
5 </head>
6 <body>
7
8 <h2>Lingkup Global Variable</h2>
9
10 <p style="width:350px;text-align:justify;">PHP juga menyimpan semua variabel global dalam array
11 yang disebut $GLOBALS [indeks]. Indeks memegang nama variabel.
12 Array juga diakses dari dalam fungsi dan dapat digunakan untuk memperbarui variabel global secara langsung.
13 Contoh di atas dapat ditulis ulang seperti ini:</p>
14
15 <?php
16 $x = 5;
17 $y = 10;
18
19 function Testing() {
20     $GLOBALS['x'] = $GLOBALS['x'] + $GLOBALS['y'];
21 }
22
23 Testing();
24 echo $y; // outputnya 15
25
26 </body>
27 </html>
```

PHP Kata Kunci Static

Biasanya, ketika fungsi selesai/dieksekusi, semua variabel akan dihapus. Namun, kadang-kadang kita ingin variabel lokal untuk tidak dihapus. untuk pekerjaan lebih lanjut.

Untuk itu, kita gunakan kata kunci yang **static** ketika anda pertama kali menyatakan variabel:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Variable PHP</title>
5 </head>
6 <body>
7
8 <h2>Static Variable</h2>
9 <p style="width:350px;text-align:justify;">
10 Biasanya, ketika fungsi selesai dieksekusi, semua variabel akan dihapus. Namun,
11 kadang-kadang kita ingin variabel lokal tidak dihapus, untuk pekerjaan lebih lanjut.
12 Untuk melakukannya, gunakan kata kunci <i>static</i> ketika pertama kali menyatakan variabel:
13 </p>
14
15 <?php
16 function Testing() {
17     static $x = 0;
18     echo $x;
19     $x++;
20 }
21
22 Testing();
23 Testing();
24 Testing();
25 ?>
26
27 </body>
28 </html>

```

Kemudian, setiap kali fungsi disebut, variabel akan masih memiliki informasi yang terkandung dari terakhir kali fungsi dipanggil.

Bab 3 dan 4

PHP Echo / Print dan Tipe Data

Echo / Print

Echo atau **Print** dalam **PHP** adalah suatu perintah untuk menampilkan suatu data yang sudah disiapkan atau dikemas dalam sebuah variabel atau data langsung yang berupa karakter maupun angka. Contoh sederhana penggunaan Echo atau Print adalah sebagai berikut, sekaligus kami terangkan tentang [Tipe Data dalam PHP](#). Sebelumnya kita akan membahas sedikit apa itu [Tipe Data dalam PHP](#).

Tipe Data PHP

Variabel dapat menyimpan beberapa jenis data, dan tipe data yang berbeda akan menghasilkan hal yang berbeda pula.

PHP mendukung tipe data berikut:

- String

- Integer
- Float
- Boolean
- Array
- Object
- NULL
- Resource

PHP String

String adalah urutan karakter, seperti "***Hello world!***". String dapat ditulis dalam tanda kutip. Anda dapat menggunakan tanda kutip tunggal atau ganda, contohnya:

```
echo "-----<br/>";
echo "<b>Ini adalah Contoh String</b><br/>";

$string = 'Hello World!';

echo "Ini adalah Contoh String + Variabel ($string)<br/>";
echo "-----<br/>";

/*
atau dengan print
*/

print "-----<br/>";
print "<b>Ini adalah Contoh String</b><br/>";

$string = 'Hello World!';

print "Ini adalah Contoh String + Variabel ($string)<br/>";
print "-----<br/>";
```

PHP Integer

Tipe data integer adalah nomor bebas-desimal antara -2,147,483,648 dan 2,147,483,647.

Aturan bilangan bulat:

1. Integer harus memiliki setidaknya satu digit
2. Integer tidak harus memiliki titik desimal
3. Integer dapat menjadi positif atau negatif

Bilangan bulat dapat ditentukan dalam tiga format: desimal, heksadesimal atau oktal.

Dalam contoh berikut variabel (**\$contoh**) adalah bilangan bulat. Fungsi `var_dump()` dalam **PHP** mengembalikan jenis data dan nilainya:

```
echo "-----<br/>";
echo "<b>Ini adalah contoh Integer</b><br/>";
echo "contohnya: ";
$contoh = 5985;
var_dump($contoh);
echo "<br/>-----<br/>";

/*
atau dengan print
*/

print "-----<br/>";
print "<b>Ini adalah contoh Integer</b><br/>";
print "contohnya: ";
$contoh = 5985;
var_dump($contoh);
print "<br/>-----<br/>";
```

PHP Float

Float (floating nomor atau angka dengan titik) adalah nomor desimal dengan angka dalam bentuk eksponensial.

Dalam contoh berikut \$x adalah float. Fungsi var_dump() dalam PHP mengembalikan jenis data dan nilainya:

```
echo "-----<br/>";
echo "<b>Ini adalah contoh Float</b><br/>";
echo "contohnya: ";
$x = 10.365;
var_dump($x);
echo "<br/>-----<br/>";

/*
atau dengan print
*/

print "-----<br/>";
print "<b>Ini adalah contoh Float</b><br/>";
print "contohnya: ";
$x = 10.365;
var_dump($x);
print "<br/>-----<br/>";
```

PHP Boolean

Boolean mewakili satu posisi yang tersedia, yaitu: benar atau tidak.

```
$x = true;
$y = false;
```

Boolean sering digunakan dalam pengujian bersyarat. Anda akan mempelajari lebih lanjut tentang pengujian bersyarat dalam bab selanjutnya dari tutorial ini. Selanjutnya....

PHP Array

Array menyimpan beberapa nilai dalam satu variabel. Dalam contoh berikut \$merk_mobil adalah array. Fungsi var_dump() dalam PHP mengembalikan jenis data dan nilainya:

```

echo "-----"
echo "<b>Ini adalah contoh Array</b><br/>";
echo "contohnya: ";
$merk_mobil = array("Ferrari", "BMW", "Toyota");
var_dump($merk_mobil);
echo "<br/>-----"

/*
atau dengan print
*/

print "-----"
print "<b>Ini adalah contoh Array</b><br/>";
print "contohnya: ";
$merk_mobil = array("Ferrari", "BMW", "Toyota");
var_dump($merk_mobil);
print "<br/>-----"

```

PHP object

Object adalah jenis data yang dapat menyimpan data dan informasi pada sebagaimana proses data tersebut didefinisikan.

Dalam **PHP**, object harus secara eksplisit dinyatakan.

Pertama kita harus mendeklarasikan sebuah **Class Object**. Untuk itu,

kita menggunakan kata kunci **Class**. **Class** adalah struktur yang dapat berisi properti dan metode, contohnya:

```

echo "-----<br>";
echo "<b>Ini adalah contoh dari PHP Object</b>";
echo "<br/>-----<br>";
class Mobil {
    function Mobil() {
        $this->model = "VN";
    }
}

// buat satu object
$herbie = new Mobil();

// tampilkan properti object
echo "Jenis atau Model dari Class Mobil adalah = ";
echo $herbie->model;
echo "<br/>-----<br>";

/*
atau dengan print
*/

print "-----<br>";
print "<b>Ini adalah contoh dari PHP Object</b>";
print "<br/>-----<br>";
class Mobil {
    function Mobil() {
        $this->model = "VN";
    }
}

// buat satu object
$herbie = new Mobil();

// tampilkan properti object
print "Jenis atau Model dari Class Mobil adalah = ";
print $herbie->model;
print "<br/>-----<br>";

```

#Anda bisa belajar PHP Object lainnya pada tutorial ini.

PHP Tipe Data NULL

Null adalah tipe data khusus yang memiliki nilai hanya satu yaitu **NULL** (nol).

Variabel tipe data **NULL** adalah variabel yang tidak memiliki nilai dan memang ditugaskan untuk itu.

Catatan: Jika sebuah variabel dibuat tanpa nilai, maka secara otomatis ditetapkan nilai **NULL**.

Variabel dapat juga dikosongkan dengan menetapkan nilai **NULL** contoh:

```
echo "-----<br>";
echo "<b>Ini adalah tipe data NULL</b><br>";
echo "-----<br>";
$x = "Hello world!";
$x = null;
var_dump($x);
echo "<br>-----<br>";

]/*
atau dengan print
*/

print "-----<br>";
print "<b>Ini adalah tipe data NULL</b><br>";
print "-----<br>";
$x = "Hello world!";
$x = null;
var_dump($x);
print "<br>-----<br>";
```

PHP Resource

Tipe data jenis **Resource** adalah tipe data istimewa bukan tipe data aktual. Tipe data ini adalah untuk menyimpan referensi ke fungsi dan sumber daya eksternal dalam **PHP**. Contoh umum menggunakan tipe data **Resource** adalah panggilan database. Kami tidak akan berbicara tentang tipe **Resource** di sini, karena itu adalah topik yang berbeda.

Bab 5

PHP String

PHP5 String

String adalah urutan karakter, seperti "***Hello world!***".

Fungsi String PHP

Dalam bab ini kita akan melihat beberapa fungsi yang sering digunakan untuk memanipulasi string.

Mendapatkan panjang dari **String**. Fungsi dalam **PHP** seperti **strlen()** adalah mengembalikan nilai dari panjang suatu **string**. Contoh di bawah ini adalah fungsi untuk mengembalikan panjang dari string "***Hello world!***":

```
26
27 echo "-----<br>";
28 echo "<b>Contoh Fungsi String (strlen)</b><br>";
29 echo "Panjang dari Hello World adalah = ";
30 echo strlen("Hello world!")." Karakter.<br>";
31 // outputnya adalah 12 panjang dari karakter "Hello World!"
32 echo "-----<br>";
33
```

Menghitung Jumlah Kata dalam String

Fungsi **PHP** **str_word_count()** untuk menghitung jumlah kata dalam **string**, contohnya:

```
39
40 echo "-----<br>";
41 echo "<b>Contoh Fungsi String str_word_count()</b><br>";
42 echo "Jumlah kata dalam Hello World adalah = ";
43 echo str_word_count("Hello world!")." Karakter.<br>";
44 // outputnya adalah 2 jumlah kata dalam "Hello World!"
45 echo "-----<br>";
46
```

Membalikkan String (Reverse)

Fungsi `strrev()` PHP membalikkan string, contohnya:

```
52 echo "-----<br>";
53
54 echo "<b>Contoh Fungsi String strrev()</b><br>";
55 echo "Membalik kata Hello world menjadi = ";
56 echo strrev("Hello world!")."<br>"; // kata "Hello World!" dibalik.
57 echo "-----<br>";
58
```

Mencari Teks Tertentu dalam String

`Strpos()` adalah fungsi dalam PHP untuk mencari teks tertentu dalam suatu string.

Jika kecocokan ditemukan, fungsi mengembalikan posisi karakter dari yang dilihatnya. Jika tidak menemukan kecocokan, maka akan kembali posisinya. Contoh di bawah ini adalah pencarian teks "world" dalam string "Hello world!":

Tip: Posisi karakter pertama dalam string adalah 0 (bukan 1).
misal:

H a l l o
0 1 2 3 4

```
72
73 echo "-----<br>";
74 echo "<b>Contoh Fungsi String Strpos()</b><br>";
75 echo "Mencari nilai dari string world = ";
76 echo strpos("Hello world!", "world")."<br>"; // maka outputnya adalah 6.
77 echo "-----<br>";
78
```


Mengganti Teks dalam String

Fungsi `str_replace()` dalam **PHP** akan menggantikan beberapa karakter dengan beberapa karakter lainnya yang sudah ditentukan dalam string. Contoh di bawah ini kita akan menggantikan teks "world" dengan "Betta":

```
85
86
87
88 echo "-----<br>";
89 echo "<b>Contoh Fungsi String str_replace()</b><br>";
90 echo "menggantikan teks world dengan Betta = ";
91 echo str_replace("world", "Betta", "Hello world!")."<br>";
92 // outputnya adalah Hello Betta!
93 echo "-----<br>";
94
95
96
```

Dan masih banyak lagi fungsi – fungsi string dalam PHP. Untuk referensi lebih lanjut silahkan hubungi kami di www.bettadev-indonesia.blogspot.com

Definisi dan penggunaan **Addslashes()**

Fungsi `addslashes()` mengembalikan sebuah string dengan garis miring terbalik di depan karakter tertentu.

Catatan: `Addslashes()` fungsi case-sensitive.

Catatan: Hati-hati menggunakan `addslashes()` pada 0 (NULL), r (mengembalikan ke karakter), n (baris baru), f (form feed), t (tab), dan v (vertikal tab).

sintaks : `addslashes(string,characters);`

```
14
15 <?php
16
17 $str = addslashes("Hello World!", "W");
18 echo "Contoh : " . ($str);
19
20 ?>
```

Contoh lainnya adalah sebagai berikut:

```
<?php
$str = "Who's Betta Dev Indonesia?";
echo $str . " This is not safe in a database query.<br>";
echo addslashes($str) . " This is safe in a database query.";
?>
```

Bab 6

PHP 5 konstanta

Konstanta hampir mirip dengan variabel ketika mereka sudah didefinisikan maka tidak dapat mengubah atau undefined.

Konstanta PHP

Konstanta adalah sebuah identifier (nama) untuk nilai sederhana. Nilai tidak berubah selama berada dalam script.

Nama konstan yang valid yaitu diawali dengan huruf atau underscore (bukan tanda \$ sebelum nama konstanta).

Catatan: Tidak seperti variabel, konstanta otomatis global di seluruh script.

Membuat konstan PHP

Untuk membuat konstan, gunakan fungsi `define()`.

Sintaks

`define(nama, nilai, case-sensitive)`

Parameter:

Nama: menentukan nama dari konstan

nilai: menentukan nilai konstan

case-sensitive: menentukan apakah nama konstan harus case-sensitive. jika Default maka false.

Contoh di bawah menciptakan konstan dengan nama case-sensitive:

```
<?php
echo "-----<br>";
echo "<b>Konstanta Default</b><br>";
echo "-----<br>";
define("SELAMAT", "Selamat datang di Betta Dev Indonesia!");
echo SELAMAT;
echo "<br>-----<br>";
?>
```

Contoh di bawah menciptakan konstan dengan nama case-sensitive:

```
<?php
/*
Contoh di bawah menciptakan konstan dengan nama case-sensitive:
*/

echo "-----<br>";
echo "<b>Konstanta dengan nama Case-sensitive</b><br>";
echo "-----<br>";
define("SELAMAT", "Selamat datang di Betta Dev Indonesia!", true);
echo selamat;
echo "<br>-----<br>";
?>
```

Konstanta Global

Konstanta global secara otomatis dapat digunakan di seluruh script. Contoh di bawah menggunakan konstanta di dalam fungsi, bahkan jika itu didefinisikan di luar fungsi:

```
<?php
echo "-----<br>";
echo "<b>Konstanta Global</b><br>";
echo "-----<br>";
define("HAI", "Selamat datang di Betta Dev Indonesia!");

function Test() {
    echo HAI;
}

Test();
echo "<br>-----<br>";
?>
```

Bab 7

PHP operator

Operator yang digunakan untuk melakukan operasi pada variabel dan nilai-nilai.

PHP membagi operator di grup berikut:

1. Aritmatika operator
2. Penetapan operator
3. Operator perbandingan
4. Kenaikan/penurunan operator (Increment/Decrement)
5. Operator logika
6. String operator
7. Array operator

PHP aritmatika operator

Operator aritmatika PHP yang digunakan dengan nilai-nilai numerik untuk melakukan operasi aritmetika umum, seperti penambahan, pengurangan, perkalian dll. dijelaskan sebagai berikut dalam contoh PHP Operator Aritmatika:

```
<?php
echo "-----<br>";
echo "<b>Aritmatika Penambahan (penjumlahan),<br>Pengurangan, Perkalian dll.</b><br>";
echo "-----<br>";
$x = 10;
$y = 6;
echo "Hasil dari Penjumlahan 10 + 6 = ";
echo $x + $y."<br>";
echo "Hasil dari Pengurangan 10 - 6 = ";
echo $x - $y."<br>";
echo "Hasil dari Perkalian 10 * 6 = ";
echo $x * $y."<br>";
echo "Hasil dari Pembagian 10 / 6 = ";
echo $x / $y."<br>";
echo "Sisa x dibagi dengan y 10 % 6 = ";
echo $x % $y;
echo "<br>-----<br>";
?>
```

PHP Penetapan Operator

Penetapan operator yang digunakan dengan nilai-nilai numerik untuk menulis nilai ke variabel. Operator dasar dari

penetapan nilai di PHP adalah "=". Yang berarti operan kiri akan ditetapkan ke nilai tugas ekspresi di sebelah kanan.

```
<?php
echo "-----<br>";
echo "<b>PHP Penetapan Operator (Assignment Operators)</b><br>";
echo "-----<br>";
$a = 10;
echo "A ditetapkan dengan nilai ". $a . "<br>";
$b = 30;
$b += 100;
echo "B ditambahkan variabel sama yang memiliki nilai berbeda ". $b . "<br>";
$c = 50;
$c -= 30;
echo "C dikurangi variabel sama yang memiliki nilai berbeda ". $c . "<br>";
$d = 10;
$d *= 3;
echo "D dikalikan variabel sama yang memiliki nilai berbeda ". $d . "<br>";
$e = 50;
$e /= 2;
echo "E dibagi variabel sama yang memiliki nilai berbeda ". $e . "<br>";
$f = 45;
$f %= 20;
echo "F Sisa habis dibagi variabel sama yang memiliki nilai berbeda ". $f;
echo "<br>-----<br>";
?>
```

Operator perbandingan

Operator perbandingan PHP yang digunakan untuk membandingkan nilai-nilai (nomor atau string):

```

<?php
echo "-----<br>";
echo "<b>Operator Perbandingan</b><br>";
echo "-----<br>";
$x = 100;
$x1 = 80;
$y1 = 80;
$y = "100";
echo "Nilainya sama jadi dianggap True == ";
var_dump($x == $y); //mengembalikan nilai true karena nilai-nilai sama
echo "<br>";
echo "Jenisnya berbeda jadi dianggap False === ";
var_dump($x === $y); // mengembalikan nilai false karena jenisnya tidak sama
echo "<br>";
echo "Nilainya sama jadi dianggap False != ";
var_dump($x != $y); // mengembalikan False karena nilai-nilai sama
echo "<br>";
echo "Nilainya sama jadi dianggap False < > ";
var_dump($x < $y); // mengembalikan nilai true karena nilai-nilai sama
echo "<br>";
echo "Jenisnya tidak sama jadi dianggap True !== ";
var_dump($x !== $y); // mengembalikan nilai true karena jenis tidak sama
echo "<br>";
echo "Nilainya dianggap True karena $x lebih besar dari $y1 > ";
var_dump($x > $y1); // mengembalikan nilai true karena $x lebih besar dari $y1
echo "<br>";
echo "Nilainya dianggap True karena $y1 lebih kecil dari $x < ";
var_dump($y1 < $x); // mengembalikan nilai true karena $y1 lebih kecil dari $x
echo "<br>";
echo "Nilainya dianggap True $x1 lebih besar dari atau sama dengan $y1 >= ";
var_dump($x1 >= $y1); // mengembalikan nilai true karena $x1 lebih besar dari atau sama dengan $y1
echo "<br>";
echo "Nilainya dianggap True karena $x1 kurang dari atau sama dengan $y1 <= ";
var_dump($x1 <= $y1); // mengembalikan nilai true karena $x1 kurang dari atau sama dengan $y1
echo "<br>-----<br>";
?>

```

PHP Increment / Decrement Operators

Operator Increment PHP yang digunakan untuk peningkatan nilai variabel.

Operator Decrement PHP yang digunakan untuk pengurangan nilai variabel.

```

<?php
echo "-----<br>";
echo "<b>PHP Increment / Decrement Operators</b><br>";
echo "-----<br>";
$a = 10;
$a1 = ++$a;
echo "Pre-Increment (++$) dari 10 adalah: ".$a1."<br>";
$b = 10;
$b1 = $b++;
echo "Post-Increment ($++) dari 10 adalah: ".$b1."<br>";
$c = 10;
$c1 = --$c;
echo "Pre-Decrement (--$) dari 10 adalah: ".$c1."<br>";
$d = 10;
$d1 = $d--;
echo "Post-Decrement ($--) dari 10 adalah: ".$d1;
echo "<br>-----<br>";
?>

```

PHP Logical operator

Operator logika PHP yang digunakan untuk menggabungkan pernyataan bersyarat.

```
<?php
echo "-----<br>";
echo "<b>PHP Logical operator</b><br>";
echo "-----<br>";
$x = 100;
$y = 50;

if ($x == 100 and $y == 50) { // jika benar $x sama dengan 100 dan $y sama dengan 50, maka
    echo "Benar! variabel X adalah 100 dan Y adalah 50 (and)"; // tampilkan Benar!
}
echo "<br>";
if ($x == 100 or $y == 90) { // jika salah satu benar antara $x sama dengan 100 atau $y sama dengan 90, maka
    echo "Benar! variabel X adalah 100 (or)"; // tampilkan Variabel ada yang sama yaitu, variabel X dengan value 100!
}
echo "<br>";
if ($x == 100 xor $y == 80) { // jika salah satu benar antara $x sama dengan 100 dan $y sama dengan 90, maka
    echo "Benar! variabel X adalah 100 (xor)."; // tampilkan Variabel ada yang sama yaitu, variabel X dengan value 100!
}
echo "<br>";
if ($x == 100 && $y == 50) { // jika $x sama dengan 100 dan $y sama dengan 50, maka
    echo "Jawaban anda Benar! (&&)"; // tampilkan tulisan jawaban anda benar!
}
echo "<br>";
if ($x == 100 || $y == 50) { // jika $x sama dengan 100 atau $y sama dengan 50, maka
    echo "Jawaban anda Benar! (||)"; // tampilkan tulisan jawaban anda benar!
}
echo "<br>";
if ($y != 70) { // jika $y tidak sama dengan 70, maka
    echo "Jawaban anda Benar! (!=)"; // tampilkan tulisan jawaban anda benar!
}
echo "<br>-----<br>";
?>
```

PHP String operator

PHP memiliki dua operator yang khusus dirancang untuk string. Contohnya sebagai berikut:

```
<?php
echo "-----<br>";
echo "<b>PHP String operator</b><br>";
echo "-----<br>";
$text1 = "Hello";
$text2 = " world!";
echo $text1 . $text2;
echo "<br>";
$t1 = "Hai";
$t2 = " Dude!";
$t1 .= $t2;
echo $t1;
echo "<br>-----<br>";
?>
```

Operator Array PHP

Operator array PHP yang digunakan untuk membandingkan array. Contohnya sebagai berikut:


```

<?php
echo "<br>-----<br>";
echo "<br>Operator Array PHP<br><br>";
echo "<br>-----<br>";
$a = array("a" => "red", "b" => "green");
$b = array("c" => "blue", "d" => "yellow");
print "1. perataan a dan b ";
print_r($a + $b); // perataan $a dan $b
print "<br>";
$d = array("a" => "red", "b" => "green");
$d = array("c" => "blue", "d" => "yellow");
print "2. kesetaraan variabel c dan d "; //jika variabel c dan d memiliki nilai yang sama, maka akan True, jika tidak False.
var_dump($c == $d);
print "<br>";
$ka = array("a" => "red", "b" => "green");
$ya = array("c" => "blue", "d" => "yellow");
print "3. Identifikasi variabel ka dan ya "; //jika variabel ka dan ya memiliki nilai yang sama, maka akan True, jika tidak False.
var_dump($ka == $ya);
print "<br>";
$ko = array("a" => "red", "b" => "green");
$yo = array("c" => "blue", "d" => "yellow");
print "4. Ketidaksamaan variabel ko dan yo "; //jika variabel ka dan ya tidak memiliki nilai yang sama, maka akan False, jika tidak True.
var_dump($ko != $yo);
print "<br>";
$ko = array("a" => "red", "b" => "green");
$yo = array("c" => "blue", "d" => "yellow");
print "5. Ketidaksamaan variabel ko dan yo "; //jika variabel ka dan ya tidak memiliki nilai yang sama, maka akan False, jika tidak True.
var_dump($ko <> $yo);
print "<br>";
$ko = array("a" => "red", "b" => "green");
$yo = array("c" => "blue", "d" => "yellow");
print "6. Non-identitas variabel ko dan yo "; //jika variabel ka dan ya tidak memiliki nilai yang sama, maka akan False, jika tidak True.
var_dump($ko !== $yo);
echo "<br>-----<br>";
?>

```

Bab 8

PHP 5 Pernyataan if..else..elseif

Pernyataan bersyarat yang digunakan untuk melakukan berbagai tindakan berdasarkan kondisi yang berbeda.

Pernyataan bersyarat PHP

Sangat sering bila Anda menulis kode, Anda ingin untuk melakukan tindakan yang berbeda untuk kondisi yang berbeda. Anda dapat menggunakan pernyataan bersyarat dalam kode Anda untuk melakukan hal ini.

Dalam PHP, kami memiliki pernyataan bersyarat berikut:

- If statement - mengeksekusi beberapa kode jika satu syarat true
- if...else statement - mengeksekusi beberapa kode jika kondisi yang benar dan kode lain jika kondisi itu false
- if...elseif...else statement - mengeksekusi kode yang berbeda untuk kondisi yang lebih dari dua
- swith statement - memilih salah satu dari banyak blok kode untuk dijalankan

sintaks:

```
if (syarat) {  
    kode akan dieksekusi jika syarat benar (true);  
}
```

Berikut adalah contohnya:

```
$txt = 100;  
  
echo "Jika benar angka dalam <br>  
variabel adalah kurang dari 200 maka  
";  
  
if ($txt < 200) {  
    echo "angka $txt keluar";  
}
```

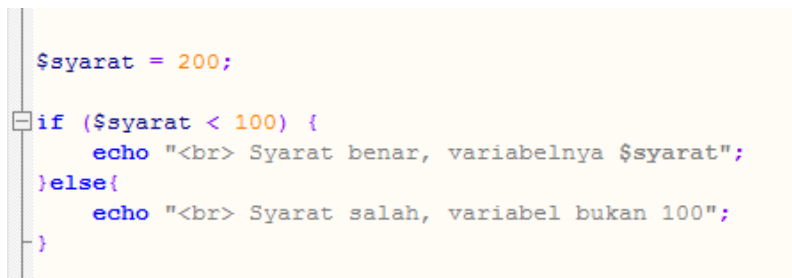
PHP Pernyataan if..else

Pernyataan if..else digunakan untuk mengeksekusi sebuah perintah, jika salah satu syarat benar maka akan keluar pernyataan yang benar (true) dan jika salah maka akan keluar pernyataan salah (false)

sintaks:

```
if (syarat) {  
    kode akan dieksekusi jika syarat benar (true);  
} else {  
    kode akan dieksekusi jika syarat salah false;  
}
```

Contohnya sebagai berikut:

A screenshot of a code editor with a light yellow background. The code is written in PHP and uses syntax highlighting. It defines a variable \$syarat with the value 200, then uses an if-else statement to check if \$syarat is less than 100. If true, it echoes "Syarat benar, variabelnya \$syarat"; otherwise, it echoes "Syarat salah, variabel bukan 100".

```
$syarat = 200;  
  
if ($syarat < 100) {  
    echo "<br> Syarat benar, variabelnya $syarat";  
} else {  
    echo "<br> Syarat salah, variabel bukan 100";  
}
```

```
if (syarat) {  
    kode akan dieksekusi jika syarat benar (true);  
} elseif (syarat) {  
    kode akan dieksekusi jika syarat benar (true);  
} else {  
    kode akan dieksekusi jika syarat salah (false);  
}
```

```
$t = date("H");
```

```
if ($t < "10") {  
    echo "<br> Selamat Pagi!";  
} elseif ($t < "20") {  
    echo "<br> Hari yang indah!";  
} else {  
    echo "<br> Selamat Malam!";  
}
```

Bab 9

PHP 5 Switch Statement

Switch digunakan untuk melakukan berbagai tindakan berdasarkan kondisi yang berbeda.

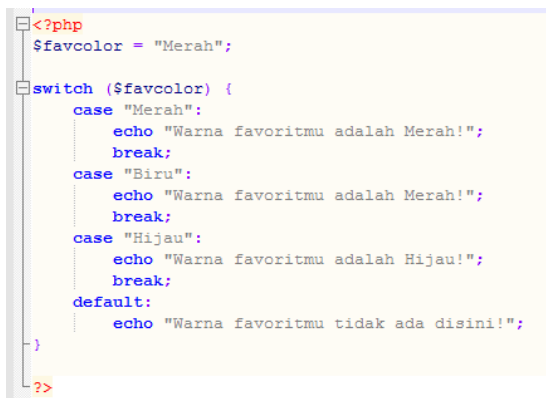
PHP switch Statement

Switch statement digunakan untuk memilih salah satu dari banyak blok kode untuk dijalankan.

```
switch (n) {  
    case nilai1:  
        kode untuk dijalankan jika n = label1;  
        break;  
    case nilai2:  
        kode untuk dijalankan jika n = label2;  
        break;  
    case nilai3:  
        kode untuk dijalankan jika n = label3;  
        break;  
    ...  
    default:  
        kode untuk dijalankan jika n tidak termasuk dalam label;  
}  

```

Contoh kodenya adalah sebagai berikut:



```
<?php  
$favcolor = "Merah";  
  
switch ($favcolor) {  
    case "Merah":  
        echo "Warna favoritmu adalah Merah!";  
        break;  
    case "Biru":  
        echo "Warna favoritmu adalah Merah!";  
        break;  
    case "Hijau":  
        echo "Warna favoritmu adalah Hijau!";  
        break;  
    default:  
        echo "Warna favoritmu tidak ada disini!";  
}  

```

Bab 10

PHP loop

Sering bila Anda menulis kode, Anda ingin blok kode untuk menjalankan lagi dan lagi dalam baris yang sama. Daripada menambahkan beberapa hampir sama dengan baris kode dalam skrip, kita dapat menggunakan loop untuk melakukan tugas seperti ini.

Dalam PHP, kami memiliki pernyataan perulangan berikut:

- Sementara - loop melalui blok kode selama kondisi benar
- do...while - loop melalui blok kode sekali dan kemudian mengulangi loop selama kondisi benar
- untuk - loop melalui blok kode tertentu beberapa kali
- foreach - loop melalui blok kode untuk setiap elemen dalam array

Sementara loop mengeksekusi sebuah blok kode selama kondisi benar.

Sintaks:

```
while (kondisi true) {  
    code yang dieksekusi;  
}
```

Contoh di bawah menetapkan variabel `$x` adalah 1 (`$x = 1`). Kemudian, while loop akan terus berjalan selama `$x` adalah kurang dari, atau sama dengan 10 (`$x <= 10`). `$x` akan bertambah 1 setiap kali berjalan loop (`$x++`):

```
25 |  
26 |  
27 |<?php  
28 |     $i = 1;  
29 |     echo "<h2>While Loop</h2>";  
30 |     while($i <= 10) {  
31 |         echo "Ini adalah nomor: $i |<br>";  
32 |         $i++;  
33 |     }  
34 |     echo "-----+";  
35 |>  
36 |
```

Do...while loop akan mengeksekusi blok kode sekali, itu kemudian akan memeriksa kondisi, dan ulangi loop sementara kondisi benar.

Sintaks:

```
do {  
    code yang dieksekusi;  
} while (kondisi true);
```

Contoh di bawah menetapkan variabel `$x` adalah 1 (`$x = 1`). Kemudian, do while loop akan menulis beberapa output, dan kemudian variabel `$x` ditambah dengan 1. Maka kondisi dicek (apakah `$x` kurang dari, atau setara dengan 5?), dan loop akan terus berjalan selama `$x` adalah kurang dari, atau setara dengan 5:

```
<?php  
$x = 1;  
echo "<h2>Do While Loop</h2>";  
do {  
    echo "Ini adalah nomor: $x <br>";  
    $x++;  
} while ($x <= 5);  
echo "-----+";  
?>
```

Perhatikan bahwa di do while loop kondisi diuji setelah mengeksekusi pernyataan didalamnya. Ini berarti do while loop akan mengeksekusi pernyataan yang setidaknya sekali, bahkan jika kondisi tersebut adalah false pertama kalinya.

Contoh di bawah menetapkan variabel `$x` adalah 6, maka loop dirun pada browser, kemudian memeriksa kondisi tersebut hasilnya akan nampak seperti contoh:

```
<?php  
$x = 6;  
echo "<h2>Do While Loop</h2>";  
do {  
    echo "Ini adalah nomor: $x <br>";  
    $x++;  
} while ($x <= 5);  
echo "-----+";  
?>
```

Bab 11

PHP For Loop

PHP For loop mengeksekusi sebuah blok kode tertentu beberapa kali.

PHP For Loop

Untuk loop digunakan ketika Anda tahu di halaman berapa kali script harus dijalankan.

Sintaks

```
for (init counter counter tes; kenaikan counter) {}  
kode untuk dijalankan;  
}
```

Parameter:

- Init counter: menginisialisasi nilai counter loop
- menguji counter: dievaluasi untuk setiap iterasi pengulangan. Jika itu mengevaluasi ke TRUE, loop terus. Jika itu mengevaluasi ke FALSE, loop berakhir.
- kenaikan counter: meningkatkan nilai counter loop

Contoh di bawah menampilkan nomor dari 0 sampai 10:

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
echo "<br>";  
?>
```

PHP foreach Loop

Foreach loop bekerja hanya pada array, dan digunakan untuk loop melalui setiap pasangan kunci/nilai dalam array.

Sintaks

```
foreach ($array sebagai $value) {  
    kode untuk dijalankan;  
}
```

Untuk setiap pengulangan, nilai elemen array saat ini ditetapkan ke `$value` dan array kursor digerakkan oleh satu, hingga mencapai elemen array terakhir.

Contoh berikut menunjukkan lingkaran itu akan menampilkan nilai-nilai array yang diberikan (`$colors`):

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

Bab 12

PHP Functions

Kekuatan yang sesungguhnya dari PHP berasal dari fungsinya; memiliki lebih dari 1000 built-in fungsi.

Mendefinisikan Fungsi PHP

Selain fungsi built-in PHP, kita dapat membuat fungsi kita sendiri. Fungsi adalah sebuah blok pernyataan yang dapat digunakan berulang kali dalam program. Fungsi tidak akan mengeksekusi segera ketika halaman load. Fungsi ini akan dijalankan oleh panggilan ke fungsi.

Membuat Definisi Fungsi PHP

Deklarasi fungsi ditetapkan pengguna dimulai dengan kata "function":

Sintaks

```
function functionName() {  
    kode untuk dijalankan;  
}
```

Catatan: Nama fungsi dapat mulai dengan huruf (bukan nomor).

Tip: Berikanlah nama fungsi sesuai fungsi yang akan digunakan!

Catatan: nama fungsi tidak case-sensitive.

Dalam contoh di bawah ini, kami menciptakan sebuah fungsi yang bernama "writeMsg()". Pembukaan kurung kurawal ({} menunjukkan awal kode fungsi dan penutupan keriting penjepit (}) menunjukkan akhir fungsi. Fungsi output "Hello world!". Untuk memanggil fungsi, hanya tuliskan nama:

```
<?php
function writeMsg() {
    echo "Hello world!";
}
writeMsg(); // memanggil fungsi
print("<br><br>");
?>
```

Argumen fungsi PHP

Informasi dapat dilewatkan ke fungsi melalui argumen. Sebuah argumen adalah seperti sebuah variabel.

Argumen ditentukan setelah nama fungsi, di dalam kurung. Anda dapat menambahkan beberapa argumen yang Anda inginkan, hanya, pisahkan dengan koma.

Contoh berikut memiliki fungsi dengan satu argumen (`$fname`). Ketika fungsi `familyName()` disebut, kami juga menyampaikan nama (misalnya Jani), dan nama ini digunakan dalam fungsi, yang output beberapa nama pertama yang berbeda, tetapi nama belakang sama:

```
<?php
function familyName($fname) {
    echo "$fname Refsnes.<br>";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
print("<br>");
?>
```

Contoh berikut memiliki fungsi dengan dua argumen (`$fname` dan `$year`):

```

<?php
function MyFamily($fname, $year) {
    echo "$fname Refsnes. Born in $year <br>";
}

MyFamily("Hege", "1975");
MyFamily("Stale", "1978");
MyFamily("Kai Jim", "1983");
print("<br>");
?>

```

Nilai Argumen Default PHP

Contoh berikut menunjukkan cara menggunakan default parameter. Jika kita memanggil fungsi `setHeight()` tanpa argumen dibutuhkan nilai default sebagai argumen:

```

<?php
function setHeight($minheight = 50) {
    echo "The height is : $minheight <br>";
}

setHeight(350);
setHeight(); // ini adalah settingan default, hasilnya 50
setHeight(135);
setHeight(80);
print("<br>");
?>

```

Fungsi PHP Mengembalikan Nilai

Untuk membiarkan sebuah fungsi yang mengembalikan nilai, gunakan kembali pernyataan berikut:

```

<?php
function sum($x, $y) {
    $z = $x + $y;
    return $z; //mengembalikan hasil ketika fungsi dipanggil
}

echo "5 + 10 = " . sum(5, 10) . "<br>";
echo "7 + 13 = " . sum(7, 13) . "<br>";
echo "2 + 4 = " . sum(2, 4);
?>

```

Bab 13

PHP Array

Array menyimpan beberapa nilai di satu variabel, contoh:

```
<?php
echo "<h1>PHP Array</h1>";

$makan = array("Mie Goreng", "Nasi Goreng", "Ramesan");
echo "I like " . $makan[1] . ", " . $makan[2] . " and " . $makan[0] . " .";
?>
```

Apa itu Array?

Array adalah variabel khusus, yang dapat menampung lebih dari satu nilai pada suatu waktu.

Jika Anda memiliki daftar item (daftar nama-nama Makanan, misalnya), menyimpan nama - nama makanan dalam satu variabel bisa terlihat seperti ini:

```
$makan1 = "Mie Goreng";
$makan2 = "Nasi Goreng";
$makan3 = "Ramesan";
```

Namun, bagaimana jika Anda ingin mengulang variabel melalui `$makan` dan menemukan satu tertentu? Dan bagaimana jika Anda memiliki lebih dari 3 makanan, tetapi 300?

Solusinya adalah menciptakan sebuah array!

Array dapat memegang banyak nilai-nilai di bawah satu nama, dan Anda dapat mengakses nilai dengan mengacu pada nomor indeks.

Membuat sebuah Array di PHP

Dalam PHP, fungsi `array()` digunakan untuk membuat sebuah array:

Contoh: `array();`

Dalam PHP, ada tiga jenis array:

1. Array - array dengan indeks numerik
2. Array asosiatif - array dengan nama kunci
3. Array multidimensi - array yang berisi satu atau lebih array

PHP Diindeks Array

Ada dua cara untuk menciptakan array:

Indeks dapat diberikan secara otomatis (indeks selalu dimulai pada 0), seperti ini:

```
$makan = array("Mie Goreng", "Nasi Goreng", "Ramesan");
```

atau indeks dapat ditetapkan secara manual seperti ini:

```
$makan[0] = "Mie Goreng";  
$makan[1] = "Nasi Goreng";  
$makan[2] = "Ramesan";
```

Contoh berikut menciptakan array-diindeks bernama \$makan, menetapkan tiga elemen itu dan kemudian mencetak teks yang terkandung dalam nilai array:

```
<?php  
$makan = array("Mie Goreng", "Nasi Goreng", "Ramesan");  
echo "<br><br>I like " . $makan[0] . ", " . $makan[1] . " and " . $makan[2] . " .";  
?>
```

Dapatkan Panjang Array - dengan Fungsi Count()

Fungsi `count()` digunakan untuk mengembalikan panjang (jumlah elemen) dari sebuah array:

```
<?php  
$makan = array("Mie Goreng", "Nasi Goreng", "Ramesan");  
echo "<br><br>".count($makan);  
?>
```

Loop untuk mengambil Array diindeks

Untuk loop dan mencetak semua nilai array diindeks, yang dapat Anda lakukan, seperti ini:

```
<?php
$makan = array("Mie Goreng", "Nasi Goreng", "Ramesan");
$arrlength = count($makan);
echo "<br><br>";
for($x = 0; $x < $arrlength; $x++) {
    echo $makan[$x];
    echo "<br>";
}
?>
```

PHP array asosiatif

Array asosiatif juga array yang menggunakan nama kunci yang Anda tetapkan untuk mereka.

Ada dua cara untuk menciptakan array asosiatif:

```
$umur = array("Dam" => "35", "Kus" => "37", "Dul"=>"43");
```

atau seperti ini,

```
$umur['Dam'] = "35";
$umur['Kus'] = "37";
$umur['Dul'] = "43";
```

Nama kunci kemudian dapat digunakan dalam skrip:

```
<?php
echo "<br>";
$umur = array("Dam" => "35", "Kus" => "37", "Dul"=>"43");
echo "Dam is " . $umur['Dam'] . " years old.";
?>
```

Loop melalui Array asosiatif

Untuk loop melalui dan mencetak semua nilai array asosiatif, Anda dapat menggunakan foreach loop, seperti ini:

```

<?php
$umur = array("Dam" => "35", "Kus" => "37", "Dul"=>"43");
echo "<br><br>";
foreach($umur as $x => $x_value) {
    echo "Name=" . $x . ", Age=" . $x_value;
    echo "<br>";
}
?>

```

PHP 5 Array Multidimensi

Sebelumnya dalam tutorial ini, ada array yang satu daftar pasangan kunci/nilai yang telah kita dijelaskan. Namun, kadang-kadang Anda ingin menyimpan nilai-nilai dengan lebih dari satu kunci. Hal tersebut bisa anda lakukan dalam array multidimensi.

PHP - Array Multidimensi

Adalah array yang berisi satu atau lebih array.

PHP memahami Array Multidimensi seperti dua, tiga, empat, lima, atau lebih. Namun, array lebih dari tiga level akan sulit untuk dikelola bagi kebanyakan orang.

Tip: Dimensi array menunjukkan jumlah indeks Anda untuk memilih elemen.

Untuk array dua dimensi, Anda memerlukan dua indeks untuk memilih sebuah elemen. Untuk tiga dimensi array Anda perlu tiga indeks untuk memilih sebuah elemen.

PHP - dua-dimensi array

Dua-dimensi array adalah array dari array (tiga dimensi array adalah array dalam array dari array).

Pertama, lihatlah tabel berikut:

Tabel Stok Makanan		
Nama	Stok	Terjual
Mie Telor	12	6
Kerupuk Udang	50	45
Sambel ABC	22	21
Kecap Bangau	31	23

Kita dapat menyimpan data dari tabel di atas dalam sebuah array dua dimensi, seperti ini:

```
$makanan = array
(
    array("Mie Telor",22,18),
    array("Kerupuk",15,13),
    array("Sambel ABC",5,2),
    array("Kecap Bangau",17,15)
);
```

Sekarang array dua dimensi \$makanan berisi empat array, dan memiliki dua indeks: baris dan kolom. Untuk mendapatkan akses ke elemen array \$makanan kita harus menunjukkan dua indeks (baris dan kolom):

```
<?php
echo "<div style='width:40%;border:1px solid #222;margin:10px auto;padding:10px;'>";
echo "<h2>Contoh 1</h2>";
$makanan = array
(
    array("Mie Telor",12,6),
    array("Kerupuk Udang",50,45),
    array("Sambel ABC",22,21),
    array("Kecap Bangau",31,23)
);

echo $makanan[0][0].": Stok: ".$makanan[0][1].", Terjual: ".$makanan[0][2]."<br>";
echo $makanan[1][0].": Stok: ".$makanan[1][1].", Terjual: ".$makanan[1][2]."<br>";
echo $makanan[2][0].": Stok: ".$makanan[2][1].", Terjual: ".$makanan[2][2]."<br>";
echo $makanan[3][0].": Stok: ".$makanan[3][1].", Terjual: ".$makanan[3][2]."<br>";
echo "</div>";
?>
```

Kita juga dapat menggunakan loop untuk menempatkan elemen array `$makanan`:

```
<?php
echo "<div style='width:40%;border:1px solid #222;margin:10px auto;padding:10px;'>";
echo "<h2>Contoh 2</h2>";
for ($row = 0; $row < 4; $row++) {
    echo "<p><b>Nomor Urutan $row</b></p>";
    echo "<ul>";
    for ($col = 0; $col < 3; $col++) {
        echo "<li>".$makanan[$row][$col]."</li>";
    }
    echo "</ul>";
}
echo "</div>";
?>
```

Bab 14

PHP Superglobals

PHP Sorting Array

Elemen dalam array dapat diurutkan dalam urutan abjad atau numerik, turun atau naik.

PHP - fungsi Sort untuk array

Dalam bab ini, kita akan berbuat sesuatu melalui array menurut fungsi PHP berikut:

- `sort()` - semacam array dalam urutan naik.
- `rsort()` - semacam array dalam urutan.
- `asort()` - semacam array asosiatif dalam urutan naik, sesuai dengan nilai.
- `ksort()` - semacam array asosiatif dalam urutan naik, menurut kunci.
- `arsort()` - semacam array asosiatif dalam urutan, sesuai dengan nilai.
- `krsort()` - semacam array asosiatif dalam urutan, menurut kunci.

Semacam Array dalam urutan naik - `sort()` Berikut contoh macam elemen array `$kendaraan` dalam urutan abjad:

```
$kendaraan = array("Motor", "Sepeda", "Becak");
sort($kendaraan);

$length = count($kendaraan);
echo "<ul>";
for($x = 0; $x < $length; $x++) {
    echo "<li>".$kendaraan[$x]."</li>";
}
echo "</ul>";
```

Berikut contoh macam elemen array `$nomor` dalam urutan numerik:

```

$nomor = array(4, 6, 2, 22, 11);
sort($nomor);

$arlength = count($nomor);
for($x = 0; $x < $arlength; $x++) {
    echo $nomor[$x];
    echo "<br>";
}

```

Semacam Array dalam urutan menurun - `rsort()` Berikut contoh macam elemen array `$kendaraan1` dalam urutan abjad:

```

$kendaraan1 = array("Motor", "Sepeda", "Becak");
rsort($kendaraan1);

$klenght1 = count($kendaraan1);
echo "<ul>";
for($x = 0; $x < $klenght1; $x++) {
    echo "<li>".$kendaraan1[$x]."</li>";
}
echo "</ul>";

```

Berikut contoh macam elemen array `$nomor1` dalam urutan numerik:

```

$nomor1 = array(4, 6, 2, 22, 11);
rsort($nomor1);
$arlength1 = count($nomor1);
for($x = 0; $x < $arlength1; $x++) {
    echo $nomor1[$x];
    echo "<br>";
}

```

Menyortir Array (Ascending Order)

Menurut nilai - `asort()` Berikut contoh macam array asosiatif dalam urutan naik, sesuai dengan nilai:

```
$umur = array("Mahmud"=>"35", "Tejo"=>"37", "Darsono"=>"43");
asort($umur);

foreach($umur as $x => $x_value) {
    echo "Nama=" . $x . ", Umur=<b>" . $x_value . "</b>"; //nama sebagai kata kunci dan umur adalah nilainya.
    echo "<br>";
}
```

Menyortir Array (Ascending Order)

Menurut kunci - ksort() Berikut contoh macam array asosiatif dalam urutan naik, menurut kunci:

```
$umur1 = array("Mahmud"=>"35", "Tejo"=>"37", "Darsono"=>"43");
ksort($umur1);

foreach($umur1 as $x1 => $x_value1) {
    echo "Nama=<b>" . $x1 . "</b>, Umur=" . $x_value1; //nama sebagai kata kunci dan umur adalah nilainya.
    echo "<br>";
}
```

Menyortir Array (urutan menurun)

Menurut nilai - arsort() Berikut contoh macam array asosiatif dalam urutan, sesuai dengan nilai:

```
$umur2 = array("Doni"=>"35", "Ela"=>"37", "Mahmud"=>"43");
arsort($umur2);

foreach($umur2 as $x2 => $x_value2) {
    echo "Nama=" . $x2 . ", Umur=<b>" . $x_value2 . "</b>"; //nama sebagai kata kunci dan umur adalah nilainya.
    echo "<br>";
}
```

Menyortir Array (urutan menurun)

Menurut kunci - krsort() Berikut contoh macam array asosiatif dalam urutan, menurut kunci:

```
$umur3 = array("Ela"=>"35", "Mahmud"=>"37", "Herman"=>"43");
krsort($umur3);

foreach($umur3 as $x3 => $x_value3) {
    echo "Nama=<b>" . $x3 . "</b>, Umur=" . $x_value3; //nama sebagai kata kunci dan umur adalah nilainya.
    echo "<br>";
}
```

Bab 15

PHP 5 variabel Global – Superglobals

Superglobals diperkenalkan di PHP 4.1.0, sebagai built-in variabel yang selalu tersedia dalam semua cakupan.

PHP variabel Global - Superglobals

Beberapa variabel yang telah ditetapkan dalam PHP adalah superglobals, yang berarti bahwa mereka selalu dapat diakses, terlepas dari lingkup - dan Anda dapat mengaksesnya dari fungsi, kelas atau file tanpa harus melakukan sesuatu yang istimewa.

Variabel superglobal PHP adalah:

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_ENV`
- `$_COOKIE`
- `$_SESSION`

Bab ini akan menjelaskan beberapa superglobals, dan sisanya akan dijelaskan dalam bab-bab selanjutnya.

PHP `$GLOBALS`

`$GLOBALS` adalah variabel super-global PHP yang digunakan untuk mengakses variabel global dari mana saja dalam naskah PHP (juga dari dalam fungsi atau metode).

PHP menyimpan semua variabel global dalam sebuah array yang disebut `$GLOBALS` [indeks]. Indeks memegang nama variabel.

Contoh di bawah menunjukkan bagaimana menggunakan `$GLOBALS` variabel super-global:

```

echo "<div style='width:60%;border:1px solid black;margin:10px auto;padding:10px;'>";
echo "Contoh dari penjumlahan dua variabel yaitu \$a + \$b = \$j:<br><br>";
$a = 75;
$b = 25;

function Penjumlahan() {
    $GLOBALS['j'] = $GLOBALS['a'] + $GLOBALS['b'];
}

Penjumlahan();
echo "\$a ($a) + \$b ($b) = \$j ($j)";
echo "</div><br>";

```

*Catatan: Dalam contoh di atas, karena j adalah tambahan / hadiah dari variabel dalam **\$GLOBALS** array, hal ini juga dapat diakses dari luar fungsi!*

PHP \$_SERVER

\$_SERVER adalah variabel super-global PHP yang memegang informasi tentang header, jalan, dan lokasi script. Contoh di bawah ini menunjukkan cara menggunakan beberapa elemen dalam **\$_SERVER**:

```

Lokasi File : /materi_tutorial/PHP/bab16_superglobals/superglobals.php
Nama Server : localhost
Nama Hosting : localhost
Halaman Aktif : http://localhost/materi_tutorial/PHP/bab16_superglobals/
Browser Digunakan : Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.103 Safari/537.36
Script Aktif : /materi_tutorial/PHP/bab16_superglobals/superglobals.php

```

Di bawah ini adalah script untuk menampilkan data di atas:

```

echo "<div style='width:80%;border:1px solid black;margin:10px auto;padding:10px;'>";
echo "<b>Lokasi File : </b>".$_SERVER['PHP_SELF'];
echo "<br>";
echo "<b>Nama Server : </b>".$_SERVER['SERVER_NAME'];
echo "<br>";
echo "<b>Nama Hosting : </b>".$_SERVER['HTTP_HOST'];
echo "<br>";
echo "<b>Halaman Aktif : </b>".$_SERVER['HTTP_REFERER'];
echo "<br>";
echo "<b>Browser Digunakan </b>: ".$_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo "<b>Script Aktif : </b>".$_SERVER['SCRIPT_NAME'];
echo "</div>";

```

Tabel berikut mencantumkan unsur-unsur paling penting yang dapat masuk ke dalam **\$_SERVER**:

Elemen kode	Deskripsi
<code>\$ _SERVER ['PHP_SELF']</code>	Mengembalikan nama file script yang digunakan saat ini
<code>\$ _SERVER ['GATEWAY_INTERFACE']</code>	Mengembalikan versi dari Common Gateway Interface (CGI) server
<code>\$ _SERVER ['SERVER_ADDR']</code>	Mengembalikan alamat IP server host
<code>\$ _SERVER ['SERVER_NAME']</code>	Mengembalikan nama host server (seperti www.contoh.com)
<code>\$ _SERVER ['SERVER_SOFTWARE']</code>	Pengembalian string identifikasi server (seperti Apache/2.2.24)
<code>\$ _SERVER ['SERVER_PROTOCOL']</code>	Mengembalikan nama dan revisi dari protokol informasi (seperti HTTP/1.1)
<code>\$ _SERVER ['REQUEST_METHOD']</code>	Mengembalikan metode yang digunakan untuk mengakses halaman (seperti POST)
<code>\$ _SERVER ['REQUEST_TIME']</code>	Mengembalikan timestamp awal (seperti 1377687496)
<code>\$ _SERVER ['QUERY_STRING']</code>	Mengembalikan string jika halaman yang diakses melalui string kueri
<code>\$ _SERVER ['HTTP_ACCEPT']</code>	Mengembalikan menerima header saat ini
<code>\$ _SERVER ['HTTP_ACCEPT_CHARSET']</code>	Mengembalikan Accept_Charset header saat ini (seperti utf-8, ISO-8859-1)
<code>\$ _SERVER ['HTTP_HOST']</code>	Mengembalikan Host header dari permintaan saat ini
<code>\$ _SERVER ['HTTP_REFERER']</code>	Mengembalikan URL lengkap halaman aktif (tidak dapat diandalkan karena tidak semua pengguna-agensi mendukung ini)
<code>\$ _SERVER ['HTTPS']</code>	Adalah script yang bertanya melalui protokol HTTP aman
<code>\$ _SERVER ['REMOTE_ADDR']</code>	Mengembalikan alamat IP dari mana pengguna yang melihat Halaman ini
<code>\$ _SERVER ['REMOTE_HOST']</code>	Mengembalikan nama Host dari mana pengguna melihat Halaman ini
<code>\$ _SERVER ['REMOTE_PORT']</code>	Mengembalikan port yang digunakan pada komputer pengguna untuk berkomunikasi dengan web server
<code>\$ _SERVER ['SCRIPT_FILENAME']</code>	Mengembalikan nama path absolut script saat ini
<code>\$ _SERVER ['SERVER_ADMIN']</code>	Mengembalikan nilai yang diberikan kepada SERVER_ADMIN direktif di web server file konfigurasi (jika skrip Anda berjalan pada virtual host, itu akan menjadi nilai ditetapkan untuk virtual host) (seperti name@beta.com)
<code>\$ _SERVER ['SERVER_PORT']</code>	Mengembalikan port pada mesin server yang digunakan oleh web server (seperti 80)
<code>\$ _SERVER ['SERVER_SIGNATURE']</code>	Mengembalikan versi server dan virtual host nama yang ditambahkan ke server
<code>\$ _SERVER ['PATH_TRANSLATED']</code>	Mengembalikan path file sistem berbasis ke script saat ini
<code>\$ _SERVER ['SCRIPT_NAME']</code>	Mengembalikan jalan script saat ini
<code>\$ _SERVER ['SCRIPT_URI']</code>	Mengembalikan URI halaman saat ini

PHP `$ _REQUEST`

PHP `$ _REQUEST` digunakan untuk mengumpulkan data setelah mengirimkan sebuah form HTML.

Contoh di bawah menunjukkan suatu bentuk dengan sebuah field input dan tombol kirim. Ketika pengguna mengirimkan data dengan mengklik 'Kirim', formulir data dikirim ke file yang ditentukan dalam tindakan atribut dari tag. Dalam contoh ini, kita menunjukkan file ini sendiri untuk pengolahan data formulir. Jika Anda ingin menggunakan file PHP yang lain untuk proses data formulir, mengganti bahwa dengan nama file pilihan Anda. Kemudian, kita dapat menggunakan super global variabel `$ _REQUEST` untuk mengumpulkan nilai dari input field:

[...Back](#)

Masukkan Nama:

Imam Ali Mustofa

Nama yang sudah diketikan pada field di atas akan dikirim pada halaman yang sama jika user menekan tombol submit pada form. Dan scriptnya adalah sebagai berikut, contoh dari penggunaan variabel super-global:

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
    Masukkan Nama: <input type="text" name="fname">
    <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // koreksi inputan
    $nama = $_REQUEST['fname'];
    if (empty($nama)) {
        echo "<br>Nama Kosong..";
    } else {
        echo "<br>". $nama;
    }
}
?>
```

PHP \$_GET

PHP `$_GET` juga dapat digunakan untuk mengumpulkan data formulir setelah mengirimkan sebuah form HTML dengan metode = 'get'. `$_GET` juga dapat mengumpulkan data yang dikirim dalam URL. Asumsikan kita memiliki halaman HTML yang berisi hyperlink dengan parameter, klik link dibawah:

Pertama, anda harus membuat file dengan nama index.php:

```
<?php
echo "<div style='width:60%;border:1px solid black;margin:10px auto;padding:10px;'>";
echo "<a href='test_get.php?subject=PHP&web=Betta Dev Indonesia'>Contoh <b>PHP \$_GET</b></a>";
echo "</div>";
?>
```

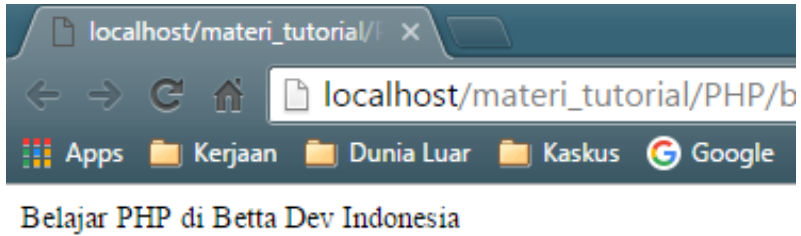
Kemudian buat satu file lagi dengan nama test_get.php:

```
<?php
echo "Belajar " . $_GET['subject'] . " di " . $_GET['web'];
?>
```

Maka hasilnya akan seperti ini:

Contoh PHP \$ _GET

Ketika link tersebut diklik, maka akan tampil halaman test_get.php dan tampilannya seperti ini:



Sepertinya sudah cukup saya menjelaskan tentang sejarah PHP dan mengungkap siapa dalang dibalik PHP itu sendiri. Bilamana ada yang kurang jelas bisa kunjungi channel Youtube kami di Betta Dev atau kunjungi Blog kami di www.bettadev-indonesia.blogspot.com segala bentuk kritik dan saran serta pertanyaan bisa anda kirimkan melalui email di bettadevindonesia@gmail.com dan jangan lupa subscribe channel kami dan like fanspage [@bettadevindonesia](https://www.facebook.com/bettadevindonesia).



Tentang Penulis

Imam Ali Mustofa, pemuda aktif, energik dan ceria. Seorang dengan passion aneh dan terkadang membuat kita bingung saat mengetahui jalan pemikirannya yang berbalik arah dari ekspektasi dan paradigma orang lain mengenai sesuatu atau malah persepsi orang kepadanya. Imam Ali Mustofa

adalah pemuda kelahiran Tegal 22 Juni 1995, di usianya yang sangat muda, dia aktif dalam berbagai kegiatan seperti Nyanggar di padepokannya (Medium Sastra dan Budaya Indonesia Kabupaten Tegal) Sekaligus personil Teater Asah Manah yang juga berasal dari padepokannya, mendaki gunung dan menjadi porter adalah selingan yang kadang dilakukannya (hobi yang menghasilkan, kenapa tidak), sempat menjadi wartawan muda di sebuah surat kabar di kotanya yaitu Kota Tegal dan sempat pula menjadi jurnalis sebuah website yang bergerak dalam bidang Travel dan Adventure, tidak berhenti disitu saja Imam Ali Mustofa juga sangat gemar menulis lirik – lirik puisi dan cerpen, memang ketertarikannya terhadap dunia sastra sangat tinggi sampai waktu dia ingin menempuh pendidikan *Sastra Indonesia* yang kemudian kandas karena biaya (pada saat itu dia uang tabungannya tidak mencukupi). Yang pada akhirnya dia memutuskan untuk berhenti satu sejenak dan mengumpulkan dana untuk menggapai cita – citanya.

Tentang Pendidikan Penulis

Saat itu Imam Ali Mustofa bekerja dan mencari penghidupannya sendiri dirantau. Tabungan sudah ada dan tekad sudah bulat. Penerimaan mahasiswa baru sudah di buka dan dia mendaftar di Fakultas Sastra Indonesia sebuah Universitas Negeri di Kota Magelang (tanpa sepengetahuan orang tua) yang pada akhirnya dia tidak bisa terima lagi, saat mengkalkulasi biaya per-semester dan biaya hidupnya – sendiri yang dijalani sehari – hari.

Akhirnya, dia memilih alternatif lain untuk mendaftar disebuah yayasan pendidikan swasta **STMIK Bina Patria** magelang yang alhamdulillah biayanya bisa ditanggung sendiri. Nah dari

situlah dia berfikir, *jika Sastra Indonesia adalah belajar membuat sebuah tulisan dengan tata bahasa yang baik dan benar, begitu juga dengan Sistem Informasi yang merupakan bagian penting dalam hidup. Memenajemen abstraksi menjadi sebuah alur yang indah, tak sekedar dilihat, tapi didengar dan dirasakan. Sistem adalah harmoni hidup yang indah.*

Keputusan itu terpikir begitu saja olehnya, dan tanggapannya dibidang pemrograman juga menarik:

“ Sastra Indonesia dan Bahasa Pemrograman adalah studi menarik sekaligus seni dalam menulis, sastra adalah studi bahasa dan tata bahasa, pemrograman juga sama. membuat alur sebuah sistem mengalir seperti cerita, memiliki rima irama.”

“ The best moment in my life is... When i encountered an error and i can fix it.”

Imam Ali Mustofa

Referensi

www.php.net

www.Wikibuku.com

www.w3shcools.com

www.bettadev-indonesia.blogspot.com

www.nyekrip.com

www.youtube.com



Betta Dev Indonesia

Betta Dev Indonesia

Jl. Ketepeng No. 9 Trunan – Tidar Selatan

0858 – 7843 – 4313 Kode Pos 56125