

Materi 10

PHP

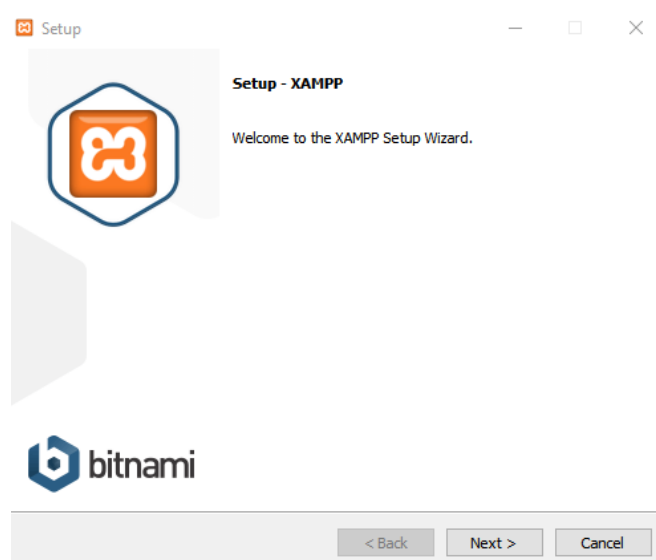
A. Pendahuluan

Apa itu XAMPP?

XAMPP adalah lingkungan pengembangan PHP paling populer. XAMPP adalah distribusi Apache yang sepenuhnya gratis dan mudah dipasang yang berisi MariaDB, PHP, dan Perl. Paket open source XAMPP telah diatur agar sangat mudah dipasang dan digunakan.

Anda dapat mengunduh XAMPP di <http://www.apachefriends.org/download.html>.

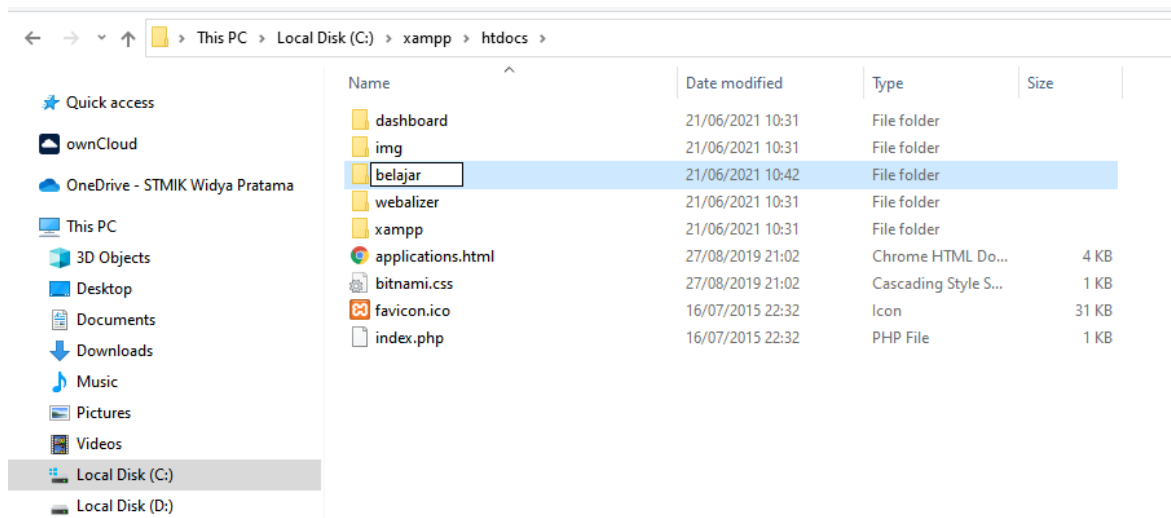
Install pada komputernya



Mengenal Folder htdocs

Agar dapat mengakses halaman PHP dari web browser, maka kita harus menempatkan file PHP dalam folder khusus yang merupakan folder home dari web server. Pada aplikasi XAMPP, folder tersebut adalah folder htdocs yang berada di C:\xampp\htdocs.

Seluruh file PHP harus diletakkan dalam folder htdocs ini. Agar memudahkan pencarian file, dalam contoh akan dibuat folder “belajar” di dalam folder htdocs. Nantinya file kita akan ada di dalam folder belajar ini.



Cara Membuat File PHP

File PHP pada dasarnya merupakan file text biasa yang bisa dibuat menggunakan aplikasi notepad bawaan windows, namun untuk penggunaan yang lebih baik, kita bisa pakai aplikasi text editor khusus programming seperti Notepad++, Atom, VS Code atau Sublime Text.

Teks editor yang paling ringan adalah **Notepad++**, aplikasi ini hanya berukuran 4MB dan sudah cukup untuk membuat perintah PHP sederhana.

Kembali ke folder belajar yang berada di C:\xampp\htdocs\belajar, kita akan membuat sebuah file baru. Untuk pembuatan file ini bisa menggunakan notepad bawaan Windows, Notepad++, maupun VS Code.

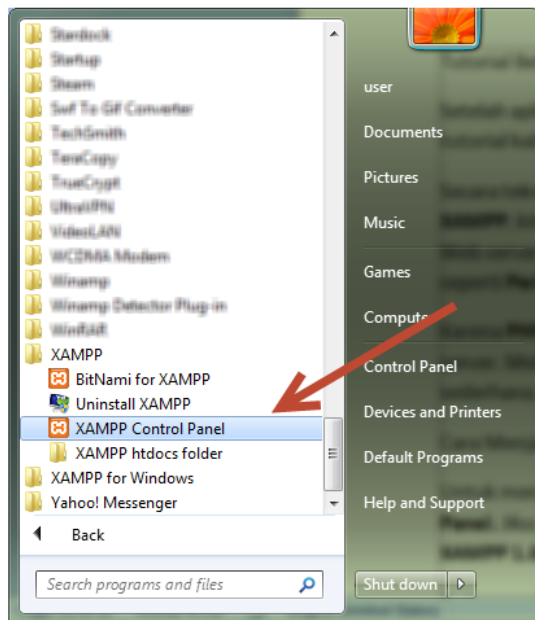
Untuk isi dari file tersebut, silahkan ketik kode berikut:

```
1 <?php
2     echo "Saya sedang belajar PHP di STMIK Widya Pratama";
3     echo "<br>";
4     echo "Hello World...";
5 ?>
```

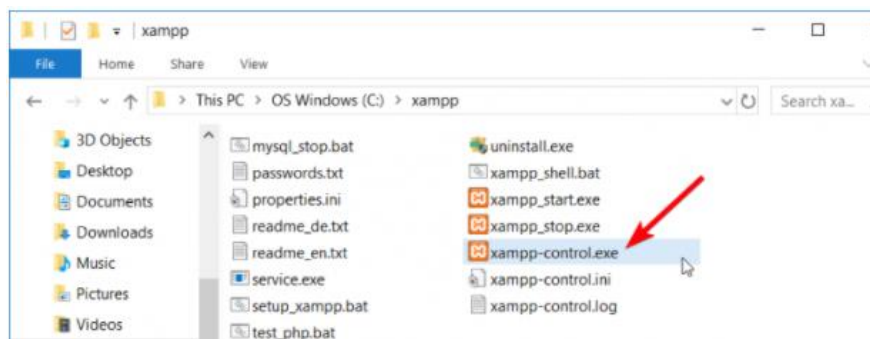
Lalu save sebagai coba.php. File tersebut akan berada pada alamat C:\xampp\htdocs\belajar\coba.php.

Cara Menjalankan File PHP

Untuk menjalankan web server Apache, pertama kali kita harus menjalankan aplikasi XAMPP Control Panel. Jika jendela tersebut belum terbuka, anda bisa menjalankannya dari START → All Program → XAMPP → XAMPP Control Panel.

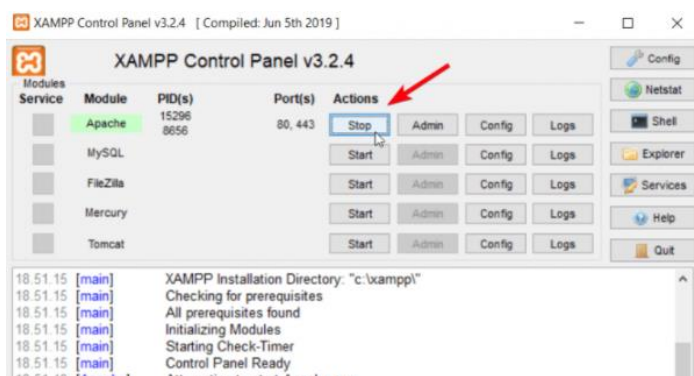


Atau jika anda tidak menemukannya di Start Menu, bisa juga buka langsung folder instalasi XAMPP di **C:\xampp**, lalu double klik file **xampp-control.exe**.

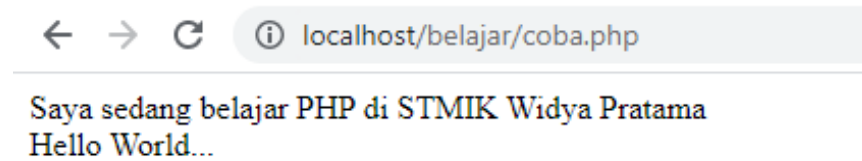


Pada tampilan XAMPP Control Panel, kita dapat melihat modul atau program apa saja yang bisa dijalankan dari XAMPP, seperti *Apache Web Server*, *MySQL*, *Program FTP Filezilla*, dan beberapa aplikasi lainnya.

Untuk menjalankan Apache, silahkan klik tombol Start pada kolom Action untuk menu Apache. Warna background dari text **Apache** akan beralih menjadi kuning yang menandakan proses sedang dimulai. Tunggu lah beberapa saat sampai warna background text Apache menjadi hijau yang menandakan web server tersebut telah berjalan di latar belakang.



Untuk menjalankannya, harus menggunakan web browser. ketik alamat berikut: localhost/belajar/coba.php pada address bar dan tekan Enter. Apabila anda mengikutinya langkah-langkah di atas dengan benar, maka di dalam web browser akan tampil seperti tampilan di berikut:



B. Sintaks PHP Dasar

Skrip PHP dapat ditempatkan di mana saja dalam dokumen.

Skrip PHP dimulai dengan `<?php` dan diakhiri dengan `?>`:

```
<?php
// PHP code goes here
?>
```

Ekstensi file default untuk file PHP adalah ".php".

File PHP biasanya berisi tag HTML, dan beberapa kode skrip PHP.

Di bawah ini, kami memiliki contoh file PHP sederhana, dengan skrip PHP yang menggunakan fungsi bawaan PHP "echo" untuk menampilkan teks "Hello World!" di halaman web:

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

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

Catatan: Pernyataan PHP diakhiri dengan titik koma (;).

PHP Case Sensitivity

Dalam PHP, kata kunci (misalnya if, else, while, echo, dll.), kelas, fungsi, dan fungsi yang ditentukan pengguna tidak peka huruf besar-kecil.

Dalam contoh di bawah, ketiga pernyataan echo di bawah ini adalah sama dan legal:

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

<?php
```

```
ECHO "Hello World!<br>";  
echo "Hello World!<br>";  
EcHo "Hello World!<br>";  
?>
```

```
</body>  
</html>
```

Catatan: Namun; semua nama variabel peka huruf besar-kecil!

Lihatlah contoh di bawah ini; hanya pernyataan pertama yang akan menampilkan nilai variabel \$color! Ini karena \$color, \$COLOR, dan \$coLOR diperlakukan sebagai tiga variabel berbeda:

```
<!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>
```

Komentar di PHP

Komentar dalam kode PHP adalah baris yang tidak dieksekusi sebagai bagian dari program. Satu-satunya tujuannya adalah untuk dibaca oleh seseorang yang melihat kodenya.

Komentar dapat digunakan untuk:

- Biarkan orang lain memahami kode Anda
- Ingatkan diri Anda tentang apa yang Anda lakukan - Sebagian besar programmer telah mengalami kembali ke pekerjaan mereka sendiri satu atau dua tahun kemudian dan harus mencari tahu kembali apa yang mereka lakukan. Komentar dapat mengingatkan Anda tentang apa yang Anda pikirkan ketika Anda menulis kode

PHP mendukung beberapa cara berkomentar:

- Sintaks untuk komentar satu baris:

```
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
/*  
This is a multiple-lines comment block  
that spans over multiple  
lines
```

```
*/  
?>
```

```
</body>  
</html>
```

- Sintaks untuk komentar beberapa baris:

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

```
<?php  
// You can also use comments to leave out parts of a code line  
$x = 5 /* + 15 */ + 5;  
echo $x;  
?>
```

```
</body>  
</html>
```

C. PHP Variables

Variabel adalah "wadah" untuk menyimpan informasi.

Membuat (Mendeklarasikan) Variabel PHP

Dalam PHP, sebuah variabel dimulai dengan tanda \$, diikuti dengan nama variabel:

```
<?php  
$txt = "Hello world!";  
$x = 5;  
$y = 10.5;  
?>
```

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

Catatan: Saat Anda menetapkan nilai teks ke variabel, beri tanda kutip di sekitar nilai.

Catatan: Tidak seperti bahasa pemrograman lain, PHP tidak memiliki perintah untuk mendeklarasikan variabel. Itu dibuat saat Anda pertama kali menetapkan nilai padanya.

Pikirkan variabel sebagai wadah untuk menyimpan data.

Variabel PHP

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

Aturan untuk variabel PHP:

- Sebuah variabel dimulai dengan tanda \$, diikuti dengan nama variabel
- Nama variabel harus dimulai dengan huruf atau karakter garis bawah
- Nama variabel tidak boleh dimulai dengan angka
- Nama variabel hanya boleh berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan _)
- Nama variabel peka terhadap huruf besar-kecil (\$age dan \$AGE adalah dua variabel yang berbeda)

Ingat bahwa nama variabel PHP peka huruf besar-kecil!

Variabel Keluaran

Pernyataan echo PHP sering digunakan untuk menampilkan data ke layar.

Contoh berikut akan menunjukkan cara menampilkan teks dan variabel:

```
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
?>
```

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

```
<?php
$txt = "W3Schools.com";
echo "I love " . $txt . "!";
?>
```

Contoh berikut akan menampilkan jumlah dari dua variabel:

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

PHP adalah Bahasa yang Diketik Secara Longgar

Pada contoh di atas, perhatikan bahwa kita tidak perlu memberi tahu PHP tipe data mana dari variabel tersebut.

PHP secara otomatis mengaitkan tipe data ke variabel, tergantung pada nilainya. Karena tipe data tidak diatur secara ketat, Anda dapat melakukan hal-hal seperti menambahkan string ke bilangan bulat tanpa menyebabkan kesalahan.

Di PHP 7, deklarasi tipe ditambahkan. Ini memberikan opsi untuk menentukan tipe data yang diharapkan saat mendeklarasikan suatu fungsi, dan dengan mengaktifkan persyaratan ketat, itu akan memunculkan "Kesalahan Fatal" pada ketidakcocokan tipe.

Anda akan mempelajari lebih lanjut tentang persyaratan ketat dan tidak ketat, dan deklarasi tipe data.

PHP Variables Scope

Dalam PHP, variabel dapat dideklarasikan di mana saja dalam skrip.

Ruang lingkup variabel adalah bagian dari skrip di mana variabel dapat direferensikan/digunakan.

PHP memiliki tiga cakupan variabel yang berbeda:

- local
- global
- static

Global and Local Scope

Variabel yang dideklarasikan di luar fungsi memiliki LINGKUP GLOBAL dan hanya dapat diakses di luar fungsi:

```
<?php
$x = 5; // global scope

function myTest() {
    // using x inside this function will generate an error
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();

echo "<p>Variable x outside function is: $x</p>";
?>
```

Variabel yang dideklarasikan dalam suatu fungsi memiliki LINGKUP LOKAL dan hanya dapat diakses di dalam fungsi itu:

```
<?php
function myTest() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();

// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
```

Anda dapat memiliki variabel lokal dengan nama yang sama dalam fungsi yang berbeda, karena variabel lokal hanya dikenali oleh fungsi di mana mereka dideklarasikan.

PHP global Keyword

Kata kunci global digunakan untuk mengakses variabel global dari dalam suatu fungsi.

Untuk melakukan ini, gunakan kata kunci global sebelum variabel (di dalam fungsi):

```
<?php
$x = 5;
$y = 10;

function myTest() {
    global $x, $y;
    $y = $x + $y;
}

myTest();
echo $y; // outputs 15
?>
```

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

Contoh di atas dapat ditulis ulang seperti ini:

```
<?php
$x = 5;
$y = 10;

function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}

myTest();
echo $y; // outputs 15
?>
```

PHP static Keyword

Biasanya, ketika suatu fungsi selesai/dieksekusi, semua variabelnya akan dihapus. Namun, terkadang kami ingin variabel lokal TIDAK dihapus. Kami membutuhkannya untuk pekerjaan lebih lanjut.

Untuk melakukannya, gunakan kata kunci static saat pertama kali mendeklarasikan variabel:

```
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}
```

```
}  
  
myTest();  
myTest();  
myTest();  
?>
```

Kemudian, setiap kali fungsi dipanggil, variabel itu akan tetap memiliki informasi yang dikandungnya sejak terakhir kali fungsi dipanggil. Catatan: Variabel masih lokal ke fungsi.

D. PHP echo and print Statements

Dengan PHP, ada dua cara dasar untuk mendapatkan output: echo dan print.

Dalam tutorial ini kita menggunakan echo atau print di hampir setiap contoh. Jadi, bab ini berisi sedikit lebih banyak info tentang dua pernyataan keluaran tersebut.

PHP echo and print Statements

echo dan print kurang lebih sama. Keduanya digunakan untuk menampilkan data ke layar.

Perbedaannya kecil: echo tidak memiliki nilai kembalian sedangkan print memiliki nilai kembalian 1 sehingga dapat digunakan dalam ekspresi. echo dapat mengambil beberapa parameter (walaupun penggunaan seperti itu jarang terjadi) sementara print dapat mengambil satu argumen. echo sedikit lebih cepat daripada print.

PHP echo statement

Pernyataan echo dapat digunakan dengan atau tanpa tanda kurung: echo atau echo().

Display Text

Contoh berikut menunjukkan cara menampilkan teks dengan perintah echo (perhatikan bahwa teks dapat berisi markup HTML):

```
<?php  
echo "<h2>PHP is Fun!</h2>";  
echo "Hello world!<br>";  
echo "I'm about to learn PHP!<br>";  
echo "This ", "string ", "was ", "made ", "with multiple parameters.";  
?>
```

Display Variables

Contoh berikut menunjukkan cara menampilkan teks dan variabel dengan pernyataan echo:

```
?php  
$txt1 = "Learn PHP";  
$txt2 = "W3Schools.com";  
$x = 5;  
$y = 4;
```

```
echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

PHP print Statement

Pernyataan print dapat digunakan dengan atau tanpa tanda kurung: print atau print().

Display Text

Contoh berikut menunjukkan cara menampilkan teks dengan perintah print (perhatikan bahwa teks dapat berisi markup HTML):

```
<?php
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

Display Variables

Contoh berikut menunjukkan cara menampilkan teks dan variabel dengan pernyataan cetak:

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>
```

E. PHP if...else...elseif Statements

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

PHP - Pernyataan if

Pernyataan if mengeksekusi beberapa kode jika satu kondisi benar.

Output "Have a good day!" jika waktu saat ini (JAM) kurang dari 20:

```
<?php
$t = date("H");
```

```
if ($t < "20") {
    echo "Have a good day!";
}
?>
```

PHP - Pernyataan if...else

Pernyataan if...else mengeksekusi beberapa kode jika suatu kondisi benar dan kode lain jika kondisi itu salah.

Output "Have a good day!" jika waktu saat ini kurang dari 20, dan "Have a good night!" jika tidak:

```
<?php
$t = date("H");

if ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

PHP - Pernyataan if...elseif...else

Pernyataan if...elseif...else mengeksekusi kode yang berbeda untuk lebih dari dua kondisi.

output "Have a good morning!" jika waktu saat ini kurang dari 10, dan "Have a good day!" jika waktu saat ini kurang dari 20. Jika tidak, itu akan menampilkan "Have a good night!":

```
<?php
$t = date("H");

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

PHP switch Statement

Gunakan pernyataan switch untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
```

```

case "blue":
    echo "Your favorite color is blue!";
    break;
case "green":
    echo "Your favorite color is green!";
    break;
default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>

```

F. PHP Loop

Seringkali ketika Anda menulis kode, Anda ingin blok kode yang sama dijalankan berulang-ulang beberapa kali. Jadi, alih-alih menambahkan beberapa baris kode yang hampir sama dalam skrip, kita dapat menggunakan loop.

Loop digunakan untuk mengeksekusi blok kode yang sama berulang kali, selama kondisi tertentu benar.

Di PHP, kami memiliki jenis loop berikut:

- while - loop melalui blok kode selama kondisi yang ditentukan benar
- do... while - loop melalui blok kode sekali, dan kemudian mengulangi loop selama kondisi yang ditentukan benar
- for - loop melalui blok kode beberapa kali
- foreach - loop melalui blok kode untuk setiap elemen dalam array

PHP while Loop

Loop while mengeksekusi blok kode selama kondisi yang ditentukan benar.

Contoh di bawah ini menampilkan angka dari 1 hingga 5:

```

<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>

```

Penjelasan Contoh

`$x = 1;` - Inisialisasi penghitung loop (`$x`), dan atur nilai awal ke 1

`$x <= 5` - Lanjutkan loop selama `$x` kurang dari atau sama dengan 5

`$x++;` - Tingkatkan nilai penghitung loop sebesar 1 untuk setiap iterasi

PHP do...while Loop

Perulangan do... while akan selalu mengeksekusi blok kode satu kali, kemudian akan memeriksa kondisi, dan mengulangi perulangan saat kondisi yang ditentukan benar.

Contoh di bawah ini pertama-tama menetapkan variabel \$x ke 1 (\$x = 1). Kemudian, loop do while akan menulis beberapa output, dan kemudian menambah variabel \$x dengan 1. Kemudian kondisi diperiksa (apakah \$x kurang dari, atau sama dengan 5?), dan loop akan terus berjalan selama \$x kurang dari, atau sama dengan 5:

```
<?php
$x = 1;

do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
?>
```

Catatan: Dalam perulangan do... while, kondisi diuji SETELAH mengeksekusi pernyataan di dalam perulangan. Ini berarti bahwa perulangan do... while akan mengeksekusi pernyataannya setidaknya sekali, bahkan jika kondisinya salah.

PHP for Loop

Perulangan for digunakan ketika Anda mengetahui sebelumnya berapa kali skrip harus dijalankan.

contoh di bawah ini menampilkan angka dari 0 sampai 10:

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

Penjelasan Contoh

\$x = 0; - Inisialisasi penghitung loop (\$x), dan atur nilai awal ke 0

\$x <= 10; - Lanjutkan loop selama \$x kurang dari atau sama dengan 10

\$x++ - Tingkatkan nilai penghitung loop sebesar 1 untuk setiap iterasi

PHP foreach Loop

Loop foreach hanya bekerja pada array, dan digunakan untuk mengulang setiap pasangan kunci/nilai dalam array.

Contoh berikut akan menampilkan nilai dari array yang diberikan (\$colors):

```
<?php
$colors = array("red", "green", "blue", "yellow");
```

```
foreach ($colors as $value) {  
    echo "$value <br>";  
}  
?>
```

PHP Break

Anda telah melihat pernyataan break yang digunakan dalam bab sebelumnya. Itu digunakan untuk "melompat keluar" dari pernyataan switch.

Pernyataan break juga dapat digunakan untuk melompat keluar dari loop.

Contoh ini melompat keluar dari loop ketika x sama dengan 4:

```
<?php  
for ($x = 0; $x < 10; $x++) {  
    if ($x == 4) {  
        break;  
    }  
    echo "The number is: $x <br>";  
}  
?>
```

Output :

The number is: 0
The number is: 1
The number is: 2
The number is: 3

PHP Continue

Pernyataan continue memecah satu iterasi (dalam loop), jika kondisi tertentu terjadi, dan berlanjut dengan iterasi berikutnya dalam loop.

Contoh ini melewati nilai 4:

```
<?php  
for ($x = 0; $x < 10; $x++) {  
    if ($x == 4) {  
        continue;  
    }  
    echo "The number is: $x <br>";  
}  
?>
```

Output :

The number is: 0

The number is: 1

The number is: 2

The number is: 3

The number is: 5

The number is: 6

The number is: 7

The number is: 8

The number is: 9