

Pertemuan 3



PERNYATAAN KONDISIONAL PHP

Yunia Ikawati

Workshop Pemrograman Web

PHP CONDITIONAL

Dalam PHP, kita memiliki pernyataan kondisional berikut:

- **Pernyataan if** - mengeksekusi beberapa kode jika satu kondisi benar
 - **Pernyataan if...else** - mengeksekusi beberapa kode jika suatu kondisi benar dan kode lain jika kondisi itu salah
 - **Pernyataan if...elseif...else** - mengeksekusi kode yang berbeda untuk lebih dari dua kondisi
 - **Pernyataan switch** - memilih satu dari banyak blok kode yang akan dieksekusi
-

SYNTAX IF

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

CONTOH PERINTAH IF

```
<html>
<body>
<?php
if (5 > 3) {
    echo "Have a good day!";
}
?>
</body>
</html>
```

Have a good day!

CONTOH PERINTAH IF

```
<html>
<body>
<?php
if (5 < 3) {
    echo "Have a good day!";
}
?>
</body>
</html>
```



localhost/mywebsite/perintah_if.php

COMPARISON OPERATORS

- Untuk membandingkan dua nilai, kita perlu menggunakan operator perbandingan.
- Berikut ini adalah operator perbandingan PHP yang dapat digunakan dalam pernyataan if:

Operator	Name	Result
==	Equal	Mengembalikan true jika nilainya sama
===	Identical	Mengembalikan true jika nilai dan tipe data identik
!=	Not equal	Mengembalikan true jika nilainya tidak sama
<>	Not equal	Mengembalikan true jika nilainya tidak sama
!==	Not identical	Mengembalikan true jika nilai atau tipe data tidak identik
>	Greater than	Mengembalikan true jika nilai pertama lebih besar dari nilai kedua
<	Less than	Mengembalikan true jika nilai pertama lebih kecil dari nilai kedua
>=	Greater than or equal to	Mengembalikan true jika nilai pertama lebih besar dari, atau sama dengan, nilai kedua
<=	Less than or equal to	Mengembalikan true jika nilai pertama lebih kecil dari, atau sama dengan, nilai kedua

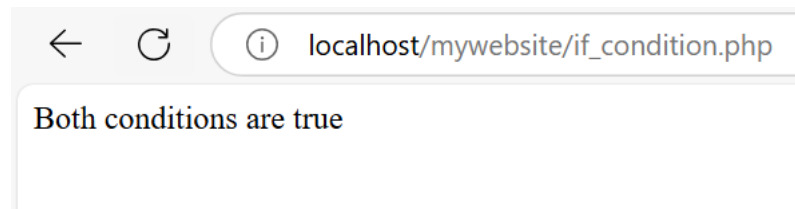
OPERATOR LOGIKA PHP

Operator	Name	Description
and	And	True if both conditions are true
&&	And	True if both conditions are true
or	Or	True if either condition is true
	Or	True if either condition is true
xor	Xor	True if either condition is true, but not both
!	Not	True if condition is not true

CONTOH COMPARISON OPERATORS

```
<html>
<body>
<?php
    $a = 200;
    $b = 33;
    $c = 500;

    if ( $a > $b && $a < $c ) {
        echo "Both conditions are true";
    }
?>
</body>
</html>
```



SYNTAX IF - ELSE

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

Pernyataan tersebut mengeksekusi beberapa kode jika suatu kondisi benar, dan kode lain jika kondisi tersebut salah.

```
<html>  
<body>  
  
<?php  
$t = date("H");  
  
if ($t < "4") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>  
  
</body>  
</html>
```

```
Have a good night!
```

SYNTAX IF – ELSEIF - IF

```
if (condition) {  
    code to be executed if this condition is true;  
} elseif (condition) {  
    // code to be executed if first condition is false and this condition is true;  
} else {  
    // code to be executed if all conditions are false;  
}
```

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

```
<html>  
<body>  
  
<?php  
$t = date("H");  
echo "<p>The hour (of the server) is " . $t;  
echo ", and will give the following message:</p>";  
  
if ($t < "10") {  
    echo "Have a good morning!";  
} elseif ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>  
  
</body>  
</html>
```

```
The hour (of the server) is 10, and will give the following message:  
Have a good day!
```

SHORT HAND IF...ELSE

Pernyataan if...else juga dapat ditulis dalam satu baris, tetapi sintaksnya sedikit berbeda.

```
<html>
<body>

<?php
$a = 13;

$b = $a < 10 ? "Hello" : "Good Bye";

echo $b;
?>

</body>
</html>
```



Good Bye

NESTED IF

Anda dapat memiliki pernyataan if di dalam pernyataan if, ini disebut pernyataan if bersarang.

```
<html>
<body>

<?php
$a = 13;

if ($a > 10) {
    echo "Above 10";
    if ($a > 20) {
        echo " and also above 20";
    } else {
        echo " but not above 20";
    }
}
?>

</body>
</html>
```

Above 10 but not above 20

PERNYATAAN SWITCH PHP

- Pernyataan switch digunakan untuk melakukan berbagai tindakan berdasarkan berbagai kondisi
- Gunakan pernyataan switch untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

```
switch (expression) {  
    case label1:  
        //code block  
        break;  
    case label2:  
        //code block;  
        break;  
    case label3:  
        //code block  
        break;  
    default:  
        //code block  
}
```

1. Ekspresi dievaluasi sekali
2. Nilai ekspresi dibandingkan dengan nilai setiap case
3. Jika ada kecocokan, blok kode terkait dieksekusi
4. Kata kunci break keluar dari blok switch
5. Blok kode default dieksekusi jika tidak ada kecocokan

CONTOH SWITCH PHP

```
<html>
<body>

<?php
$favcolor = "red";

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

Your favorite color is red!Your favorite color is blue!

CONTOH SWITCH PHP

```
<html>
<body>

<?php
$d = 4;

switch ($d) {
    default:
        echo "Looking forward to the Weekend";
        break;
    case 6:
        echo "Today is Saturday";
        break;
    case 0:
        echo "Today is Sunday";
}
?>

</body>
</html>
```

Looking forward to the Weekend

PHP LOOPS

- ✓ Loop digunakan untuk mengeksekusi blok kode yang sama berulang-ulang, selama kondisi tertentu berlaku.
 - ✓ Dalam PHP, kita memiliki tipe loop berikut:
 - **while** - melakukan loop melalui blok kode selama kondisi yang ditentukan berlaku
 - **do...while** - melakukan loop melalui blok kode sekali, lalu mengulang loop selama kondisi yang ditentukan berlaku
 - **for** - melakukan loop melalui blok kode sejumlah waktu tertentu
 - **foreach** - melakukan loop melalui blok kode untuk setiap elemen dalam array
-

WHILE LOOP


- Perulangan while - Melakukan pengulangan pada blok kode selama kondisi yang ditetapkan masih bernilai benar.
- Perulangan while tidak berjalan dalam jumlah tertentu, tetapi memeriksa setelah setiap iterasi apakah kondisinya masih benar.

```
<html>
<body>

<?php
$i = 1;

while ($i < 6) {
    echo $i;
    $i++;
}
?>

</body>
</html>
```



12345

WHILE LOOP

Break

```
<html>
<body>

<?php
$i = 1;

while ($i < 6) {
    if ($i == 3) break;
    echo $i;
    $i++;
}
?>

</body>
</html>
```

12

Continue

```
<?php
$i = 0;

while ($i < 6) {
    $i++;
    if ($i == 3) continue;
    echo $i;
}
?>

</body>
</html>
```

12456

DO...WHILE

- Perulangan do...while akan selalu mengeksekusi blok kode minimal satu kali, kemudian memeriksa kondisinya, dan mengulang perulangan selama kondisi yang ditentukan bernilai benar.

```
<html>
<body>

<?php
$i = 1;

do {
    echo $i;
    $i++;
} while ($i < 6);
?>

</body>
</html>
```

12345

PHP FOR LOOP

- Perulangan for digunakan ketika mengetahui berapa kali script harus dijalankan.
 1. ekspresi1 (nilai awal) dievaluasi sekali
 2. ekspresi2 (batasan) dievaluasi sebelum setiap iterasi
 3. ekspresi3 (meningkatkan nilai) dievaluasi setelah setiap iterasi

```
for (expression1, expression2, expression3) {  
    // code block  
}
```

CONTOH PHP FOR LOOP

```
<html>
<body>

<?php
for ($x = 0; $x <= 100; $x+=10) {
    echo "The number is: $x <br>";
}
?>

</body>
</html>
```

```
The number is: 0
The number is: 10
The number is: 20
The number is: 30
The number is: 40
The number is: 50
The number is: 60
The number is: 70
The number is: 80
The number is: 90
The number is: 100
```

PERULANGAN PHP FOREACH

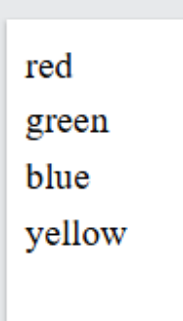
- Perulangan foreach - Melakukan perulangan melalui blok kode untuk setiap elemen dalam array atau setiap properti dalam objek.

```
<html>
<body>

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

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

</body>
</html>
```



```
red
green
blue
yellow
```

PHP BREAK

- Pernyataan break dapat digunakan untuk keluar dari berbagai jenis perulangan.

```
<html>
<body>

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

</body>
</html>
```

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

PHP CONTINUE

- Pernyataan continue dapat digunakan untuk keluar dari iterasi loop saat ini, dan melanjutkan ke iterasi berikutnya.

```
<html>
<body>

<?php
$x = 0;

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

</body>
</html>
```

```
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
The number is: 10
```

PHP FUNCTION

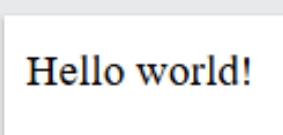
- Fungsi adalah blok pernyataan yang dapat digunakan berulang kali dalam suatu program.
- Fungsi tidak akan dijalankan secara otomatis saat halaman dimuat.
- Fungsi akan dijalankan dengan memanggil fungsi tersebut.
- PHP memiliki lebih dari 1000 fungsi bawaan yang dapat dipanggil secara langsung, dari dalam skrip, untuk melakukan tugas tertentu.
- Selain fungsi bawaan PHP, Anda dapat membuat fungsi Anda sendiri.
- Nama fungsi harus diawali dengan huruf atau garis bawah.
- Nama fungsi TIDAK Case Sensitive.

```
<html>
<body>

<?php
function myMessage() {
    echo "Hello world!";
}

myMessage();
?>

</body>
</html>
```



Hello world!

PHP FUNCTION ARGUMENT

- Informasi dapat diteruskan ke fungsi melalui argumen.
- Argumen sama seperti variabel.
- Argumen ditentukan setelah nama fungsi, di dalam tanda kurung.

```
<html>
<body>

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

familyName("Yani");
familyName("Tia");
familyName("Rina");
familyName("Kay");
familyName("Berli");
?>

</body>
</html>
```

```
Yani Salwa.
Tia Salwa.
Rina Salwa.
Kay Salwa.
Berli Salwa.
```

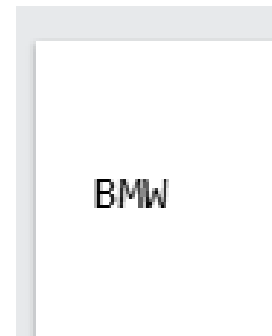
PHP ARRAYS

- Array menyimpan beberapa nilai dalam satu variabel.
- Array adalah variabel khusus yang dapat menampung banyak nilai di bawah satu nama.
- Array dapat mengakses nilai-nilai tersebut dengan merujuk ke nomor indeks atau nama.

```
<html>
<body>
<pre>

<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "$cars[1]";
?>

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



VARIABEL GLOBAL PHP - SUPERGLOBAL

- Beberapa variabel yang telah ditetapkan sebelumnya dalam PHP adalah "superglobal", yang berarti variabel tersebut selalu dapat diakses, apa pun cakupannya – kita dapat mengaksesnya dari fungsi, kelas, atau berkas apa pun tanpa harus melakukan hal khusus apa pun.

```
<html>
<body>

<?php
$x = 75;

function myfunction() {
    echo $GLOBALS['x'];
}

myfunction()
?>

</body>
</html>
```



75

PHP REGULAR EXPRESSIONS

- Ekspresi reguler dapat berupa karakter tunggal, atau pola yang lebih rumit.
- Ekspresi reguler dapat digunakan untuk melakukan semua jenis operasi pencarian teks dan penggantian teks.

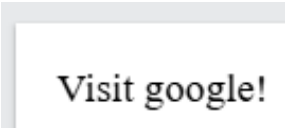
Menggunakan preg_replace()

Fungsi preg_replace() akan mengganti semua kecocokan pola dalam string dengan string lain.

```
<html>
<body>

<?php
$str = "Visit Microsoft!";
$pattern = "/microsoft/i";
echo preg_replace($pattern, "google", $str);
?>

</body>
</html>
```



Visit google!

PRAKTIKUM

- Silahkan kerjakan PERCOBAAN dan LATIHAN dan buatlah laporan praktikumnya.
- Masing-masing percobaan dan Latihan silahkan di jelaskan atau Analisa singkat

Percobaan-1

if01.php

```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri")
echo "Have a nice weekend!";
else
echo "Have a nice day!";
?>

</body>
</html>
```

Percobaan-2

if02.php

```
<html>
<body>

<?php
$x=10;
if ($x==10)
{
echo "Hello<br />";
echo "Good morning<br />";
}
?>

</body>
</html>
```

Percobaan-3

example : switch01.php

```
<html>
<body>

<?php
switch ($x)
{
case 1:
    echo "Number 1";
    break;
case 2:
    echo "Number 2";
    break;
case 3:
    echo "Number 3";
    break;
default:
    echo "No number between 1 and 3";
}
?>

</body>
</html>
```

Percobaan-4

while01.php

```
<html>
<body>

<?php
$i=1;
while($i<=5)
{
echo "The number is " . $i . "<br />";
$i++;
}
?>

</body>
</html>
```

Percobaan-5

dowhile01.php

```
<html>
<body>

<?php
$i=0;
do
{
    $i++;
    echo "The number is " . $i . "<br />";
}
while ($i<5);
?>

</body>
</html>
```

Percobaan-6

for01.php

```
<html>
<body>

<?php
for ($i=1; $i<=5; $i++)
{
echo "Hello World!<br />";
}
?>

</body>
</html>
```

Percobaan-7

foreach01.php

```
<html>
<body>

<?php
$arr=array("one", "two", "three");
foreach ($arr as $value)
{
echo "Value: " . $value . "<br />";
}
?>

</body>
</html>
```

Percobaan-8

visitorinfo.php

```
<?php
$ip = $_SERVER['REMOTE_ADDR'];
$agent = $_SERVER['HTTP_USER_AGENT'];

if(strpos($agent, 'Opera') !== false)
    $agent = 'Opera';
else if(strpos($agent, "MSIE") !== false)
    $agent = 'Internet Explorer';

echo "Your computer IP is $ip and you are using $agent";
?>
```

Percobaan-9

while02.php

```
<?php
$number = 1;
while ($number < 10)
{
    echo $number . '<br>';
    $number += 1;
}
?>
```

Percobaan-10

Example :break01.php

```
<?php
$number = 1;

while ($number < 10)
{
echo $number . '<br>';

  if ($number == 6)
  {
    break;
  }

  $number += 1;
}
?>
```

PERCOBAAN 11

```
<html>
<body>

<?php
$cars = array (
    array("Volvo",22,18),
    array("BMW",15,13),
    array("Saab",5,2),
    array("Land Rover",17,15)
);

echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2]."<br>";
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2]."<br>";
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2]."<br>";
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2]."<br>";
?>

</body>
</html>
```

PERCOBAAN 12

```
<html>
<body>

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

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```

PERCOBAAN 13

```
<html>
<body>

<a href="demo_phpfile.php?subject=PHP&web=google.com">Test $GET</a>

</body>
</html>
```

LATIHAN

- Buatlah program PHP yang menampilkan output seperti berikut: Data mahasiswa diganti data nama keluarga masing-masing

Data Mahasiswa

Nama	Nilai	Grade
Andi	85	B
Budi	78	C
Citra	92	A
Dewi	65	D
Eka	50	E
