

STATEMENT KONTROL



Statement IF ...

Sintaks dari statement tersebut dalam PHP adalah"

```
if (syarat)  
{  
  statement1;  
  statement2;  
  .  
  .  
}
```

Contoh

```
<?
$my_name = "d3";
if ($my_name == "d3")
{
echo "Your name is ".$my_name."!"<br>";
}
echo "Welcome to my homepage!";
?>
```

-
- ❑ Untuk menyatakan syarat, biasanya digunakan operator perbandingan seperti yang telah dibahas sebelumnya.
 - ❑ Apabila syarat bernilai TRUE maka statement-statement yang diapit dengan tanda kurung kurawal akan dijalankan.
-

Bentuk lain if (dgn else)

```
if (syarat)
{
statement1;
statement2;
.
.
}
else
{
statement3;
statement4;
.
.
}
```

Contoh dengan else

```
<?
$number = 3;
if ($number == 4)
{
echo "Benar";
}
else
{
echo "Salah";
}
?>
```

Bentuk Lain if dengan elseif

```
if (syarat1)
{
statement11;
statement12;
}
elseif (syarat2)
{
statement21;
statement22;
}
else
{
statement1;
statement2;
}
```

Contoh dengan elseif

```
<?
$karyawan = "Bob";
if($karyawan == "Tanner")
{
echo "Hello Tanner!";
}
elseif($karyawan == "Bob")
{
echo "Hello Bob!";
```

Statement SWITCH

```
switch (variabel)
{
case option1:
statement11;
statement12;
break;
case option2:
statement21;
statement22;
break;
default:
statementdefault1;
statementdefault2;
break;
}
```

Contoh

```
<?php
$tujuan = "Tokyo";
echo "Biaya Perjalanan Menuju $tujuan adalah ";
switch ($tujuan){
case "Las Vegas":
echo " $500";
break;
case "Tokyo":
echo " $900";
break;
case "Caribbean Islands":
echo " $700";
break;
}
?>
```

Switch dengan default

```
<?php
$tujuan = "New York";
echo "Biaya Perjalanan Menuju $tujuan adalah ";
switch ($tujuan){
case "Las Vegas":
echo " $500";
break;
case "Tokyo":
echo " $900";
break;
case "Caribbean Islands":
echo " $700";
break;
default:
echo " $100";
break;
}
<?
```

Statement WHILE

- Statement ini digunakan untuk mengerjakan suatu statement secara berulang-ulang sampai suatu syarat dipenuhi. Sintaksnya adalah

```
while (syarat)  
{  
  statement;  
  statement;  
  .  
  .  
}
```

-
- ❑ Pada sintaks di atas, selama syarat bernilai TRUE maka statement-statement di dalam while akan terus dijalankan secara berulang-ulang. Perulangan baru akan berhenti apabila syarat bernilai FALSE. Sebelum statement yang diulang-ulang dilakukan, terlebih dahulu akan dicek syarat nya apakah bernilai TRUE atau FALSE. Apabila TRUE maka statement akan dijalankan. Sedangkan apabila FALSE, perulangan akan langsung berhenti. Dengan kata lain, statement dalam WHILE bisa jadi tidak akan pernah dilakukan, yaitu apabila syaratnya langsung bernilai FALSE.
-

Contoh

```
<?
$harga_sikat = 1500;
$jumlah_sikat = 10;
while ( $jumlah_sikat <= 100 )
{
echo "Jumlah Sikat : $jumlah_sikat <br>";
echo "Harga          : ".$harga_sikat * $jumlah_sikat.
"<br>";
$jumlah_sikat = $jumlah_sikat + 10;
}
?>
```

-
- ❑ Kode di atas akan menampilkan hasil di browser yang berisi jumlah sikat dan harganya, dengan asumsi harga sebuah sikat adalah Rp. 1.500. Jumlah sikat yang ditampilkan adalah kelipatan 10 dengan batas sampai 100 buah.
-

Statement FOR

Statement FOR mirip dengan WHILE yang memiliki sintaks berikut ini

for (inisialisasi counter; syarat;
increment/decrement counter)

```
{  
    statement;  
    .  
    .  
}
```

-
- Untuk memperjelas pemahaman tentang FOR, berikut ini adalah contoh kode dengan for untuk menghasilkan tampilan yang sama dengan contoh while sebelumnya (tentang jumlah sikat dan harganya). Coba bandingkan dengan kode contoh while sebelumnya.
-

Contoh for

```
<?
$harga_sikat = 1500;
for ($jumlah_sikat = 10; $jumlah_sikat <=
    100; $jumlah_sikat+=10)
{
    echo "Jumlah Sikat : $jumlah_sikat <br>";
    echo "Harga          : ".$harga_sikat *
        $jumlah_sikat.
        "<br>";
}
```

Statement DO WHILE

Statement ini merupakan bentuk modifikasi dari WHILE. Sintaksnya adalah sebagai berikut

```
do  
{  
statement;  
.  
.  
}  
while (syarat);
```

Perbandingan While dan Do While

```
<?
$kue = 0;
while($kue > 1)
{
echo "statement"
}
?>

<?
$kue = 0;
do
{
echo "statement"
} while ($kue > 1);
?>
```

-
- ❑ Pada kode WHILE, teks "Statement " tidak akan ditampilkan karena syaratnya langsung bernilai FALSE (perulangan berhenti). Sedangkan pada DO WHILE, teks akan ditampilkan dahulu kemudian perulangan berhenti (syarat bernilai FALSE).
-

ARRAY



-
- Dalam PHP, indeks untuk array dapat berupa numerik dan dapat pula berupa suatu nilai atau yang sering disebut dengan array assosiatif.
-

Array Berindeks Numerik

- Sintaks umum untuk menyatakan suatu array berindeks numerik beserta nilainya adalah

`$nama_array[$key] = value;`

- Dalam hal ini `$key` berupa bilangan bulat mulai dari 0, 1, 2, ...
-

Contoh Array Berindeks Numerik

<?

```
$karyawan[0] = "Bob";
```

```
$karyawan[1] = "Sally";
```

```
$karyawan[2] = "Charlie";
```

```
$karyawan[3] = "Clare";
```

```
echo "Berikut ini adalah 2 orang karyawan saya,  
yaitu "
```

```
. $karyawan[0] . " & " . $karyawan[1];
```

```
echo "<br>Dua orang karyawan saya yang lain  
adalah "
```

```
. $karyawan[2] . " & " . $karyawan[3];
```

```
?>
```

Cara lain....

```
$karyawan=array('bob','sally','  
    Charlie',' Clare');
```

Array multidimensi

```
$a[1][1]=11;
```

```
$a[1][2]=12;
```

```
$a[2][1]=21;
```

```
$a[2][2]=22;
```

```
$a=array(array(11,12),array(21,22));
```

Array Asosiatif

- Untuk array asosiatif, sintaksnya sama dengan array berindeks numerik namun perbedaannya adalah pada \$key. Pada array asosiatif, \$key dapat berupa suatu string.
-

Contoh Array Asosiatif

```
<?
$gaji["Bob"] = 2000;
$gaji["Sally"] = 4000;
$gaji["Charlie"] = 600;
$gaji["Clare"] = 0;
echo "Bob digaji - $" . $gaji["Bob"] . "<br>";
echo "Sally digaji - $" . $gaji["Sally"] . "<br>";
echo "Charlie digaji - $" . $gaji["Charlie"] . "<br>";
echo "dan Clare digaji - $" . $gaji["Clare"];
?>
```

Statement Foreach

Misalkan Anda punya data berupa array asosiatif yang akan diproses secara berulang-ulang, maka PHP menyediakan statement foreach yang mudah digunakan.

Sintaksnya adalah:

```
foreach(variabelarray as kunci => value)  
{  
statement;  
.  
.  
}
```

Contoh Foreach

```
<?
$UsiaKaryawan["Lisa"] = "28";
$UsiaKaryawan["Jack"] = "16";
$UsiaKaryawan["Ryan"] = "35";
$UsiaKaryawan["Rachel"] = "46";
$UsiaKaryawan["Grace"] = "34";
foreach($UsiaKaryawan as $Nama => $umur)
{
echo "Nama Karyawan: $Nama, Usia: $umur"." th
    <br>";
}
?>
```

Fungsi count() dan sizeof()

Kedua fungsi ini memiliki kegunaan yang sama, yaitu menghitung jumlah elemen yang dimiliki oleh sebuah array. Sintaksnya adalah sebagai berikut:

```
count($array);
```

```
sizeof($array);
```

Contoh:

```
$a[0] = 1;
```

```
$a[1] = 3;
```

```
$a[2] = 5;
```

```
$jumlah = count($a);
```

```
//variabel $jumlah akan bernilai 3
```

```
$b["buah"] = "semangka";
```

```
$b["sayur"] = "wortel";
```

```
$b["daging"] = "ayam";
```

```
$b["utama"] = "nasi";
```

```
$jumlah = sizeof($b);
```

```
//variabel $jumlah akan bernilai 4
```

Fungsi array_push()

- ❑ Fungsi ini digunakan untuk menyisipkan elemen baru ke dalam sebuah array. Elemen baru ini disisipkan pada bagian paling akhir. Sintaksnya adalah sebagai berikut:
 - ❑ `Array_push($array,nilai[nilai,...])`
 - ❑ Nilai adalah nilai-nilai baru yang akan disisipkan ke dalam array.
-

Contoh:

```
$program = array ("HTML", "PHP");  
array_push ($program, "CSS",  
"JavaScript");  
// anggota $program sekarang  
menjadi "HTML", "PHP", "CSS",  
"JavaScript"
```

Tugas

- ❑ Carilah 10 fungsi array yang lain dan berikan contoh penggunaanya, tulis dalam word dengan format (tugasarray_nim_masing2.docx) upload ke perkuliahan teori di spada
-