

MODUL

PEMOGRAMAN WEB II

Oleh:

CHALIFA CHAZAR

MODUL 4

Statement Kontrol

Tujuan:

Mahasiswa memahami penggunaan statement kontrol untuk mengontrol pemilihan, mengontrol pengulangan dan memindahkan eksekusi dari/ke bagian tertentu dalam pemrograman PHP.

Pustaka:

Raharjo, B. (2015): Mudah Belajar PHP Teknik Penggunaan Fitur-Fitur Baru Dalam PHP 5. Informatika. Bandung

Buku Pintar Webmaster: Adhi Prasetio, 2015

w3schools.com

Statemen Kontrol adalah statemen-statemen yang digunakan untuk mengendalikan jalannya eksekusi kode program. Statemen kontrol dapat dibedakan menjadi 4, yaitu:

- Statemen mengontrol pemilihan,
- Statemen mengontrol pengulangan,
- Statemen memindahkan eksekusi dari/ke bagian kode tertentu, dan
- Statemen untuk menangani kejadian eksepsi (kesalahan program)

Dalam modul ini kita hanya akan mempelajari sampai dengan 3 statemen kontrol saja.

1. Statemen Pemilihan

Statemen pemilihan adalah blok-blok kode program yang dapat menentukan statemen mana yang akan dieksekusi, tergantung dari kondisi yang didefinisikan di dalamnya.

Dalam PHP, statemen pemilihan dapat dilakukan dengan 2 cara, antara lain:

- Menggunakan statemen **if**
- Menggunakan statemen **switch**

1.1 Statemen if

Berikut ini adalah bentuk umum penggunaan statemen **if** dalam PHP.

Statemen if dengan 1 kondisi
<pre>If (konsisi) { statement; }</pre>
Statemen if dengan 2 Kondisi
<pre>If (konsisi) { statement1; } else { statement2; }</pre>
Statemen if dengan 3 Kondisi atau lebih
<pre>If (konsisi1) { statement1; } elseif (kondisi2) { statement2; } elseif (kondisi3) { statement3; } else { statement4; }</pre>

Contoh:

```
<?php
$nilai = 8;
if ($nilai % 2 == 0) {
    echo "$nilai adalah bilangan genap";
}
?>
```

Latihan:

1. Buat program untuk menentukan bilangan ganjil atau bilangan genap menggunakan statemen if.
2. Buat program untuk menentukan bilangan negatif atau positif atau bernilai 0 menggunakan statemen if.

1.2. Statemen switch

Statemen lain yang dapat digunakan untuk pemilihan adalah **switch**. Berikut ini adalah bentuk umum penggunaan statemen **switch** dalam PHP.

```
Switch (variabel) {  
    Case nilai1: statemen1; break;  
    Case nilai2: statemen2; break;  
    Case nilai3: statemen3; break;  
    Case nilai4: statemen4; break;  
}
```

Contoh:

```
<?php  
$nohari = 6;  
switch ($nohari) {  
    case 1: echo "hari ke-1 : senin\n"; break;  
    case 2: echo "hari ke-2 : selasa\n"; break;  
    case 3: echo "hari ke-3 : rabu\n"; break;  
    case 4: echo "hari ke-4 : kamis\n"; break;  
    case 5: echo "hari ke-5 : jumat\n"; break;  
    case 6: echo "hari ke-6 : sabtu\n"; break;  
    case 7: echo "hari ke-7 : minggu\n"; break;  
}  
?>
```

2. Statemen Pengulangan

Pengulangan adalah eksekusi satu atau beberapa statemen secara berulang sebanyak jumlah yang ditentukan atau didefinisikan.

Dalam PHP, statemen pengulangan dilakukan dengan beberapa cara, antara lain:

- Menggunakan **for**
- Menggunakan **while**
- Menggunakan **do-while**
- Menggunakan **foreach** (untuk array)

2.1. Statemen for

Berikut ini bentuk umum penggunaan statemen for di dalam PHP.

```
for (inisialisasi; ekspresi; increment/decrement)
statement;

// atau dapat ditulis juga:

for (inisialisasi; ekspresi; increment/decrement) {
statement;
}
```

Contoh:

```
<?php
for ($a = 0; $a <= 5; $a++){
    echo $a . '<br>';
}
?>
```

2.2. Statement while

Statemen **while** berguna untuk melakukan pengulangan statemen dengan cara memeriksa kondisi di bagian awal blok. Artinya statemen dalam blok pengulangan akan dilakukan jika kondisi terpenuhi (bernilai **true**).

Bentuk umum penggunaan statemen **while** dalam PHP.

```
while (ekspresi)
statement;
```

```
// atau dapat ditulis juga:  
  
while (ekspresi) {  
    statement;  
}
```

Contoh:

```
<?php  
$a = 0;  
while ($a <= 5) {  
    echo $a . '<br>';  
    $a++;  
}  
?>
```

2.3. Statemen do-while

Bentuk pengulangan jenis ini hampir sama dengan jenis pengulangan yang menggunakan perintah **while**. Perbedaannya, dalam jenis ini kondisi diperiksa di bagian akhir blok. Artinya, pada saat kondisi tidak terpenuhi (bernilai **false**), statemen dalam blok pengulangan **do-while** masih tetap akan dieksekusi satu kali.

Bentuk umum penggunaan statemen **do-while** dalam PHP.

```
do  
statement;  
while (ekspresi);  
  
// atau dapat ditulis juga:  
do {  
    statement;  
} while (ekspresi);
```

Contoh:

```
<?php  
$a = 6;  
do {  
    echo $a . '<br>';
```

```
    $a++;  
} while ($a <= 5);  
>
```

2.4. Statemen foreach

Statemen **foreach** digunakan untuk melakukan pengulangan pada saat mengakses elemen-elemen dalam suatu array.

Bentuk umum penggunaan statemen **foreach** dalam PHP.

```
For ($array as $current)  
statement;  
  
// atau dapat ditulis juga:  
  
foreach ($array as $current) {  
statement;  
}
```

Contoh:

```
<?php  
$pegawai = array ("direktur" => "bimo", "manajer" => "aria",  
"pemasaran" => "santi");  
  
foreach ($pegawai as $nama) {  
    echo "hallo," . $nama . '<br>';  
}  
>
```

3. Statemen Loncat

PHP menyediakan statemen **break**, **continue**, **goto**, dan **return** untuk mengontrol eksekusi kode program agar mengarah ke baris tertentu yang kita kehendaki. Selain itu juga PHP menyediakan fungsi **exit()**. Adapun berikut ini penjelasan tujuan dari masing-masing statemen.

- break** : Berguna untuk menghentikan eksekusi dari suatu struktur **for**, **while**, **do-while**, **foreach**, maupun **switch**.
- continue** : Digunakan untuk mengabaikan eksekusi statemen-statemen yang ada pada index aktif (yang ditulis setelah

statemen **continue**) dan memaksa program untuk melanjutkan proses pengulangan ke index berikutnya.

return : Digunakan di dalam fungsi, dan berguna untuk mengembalikan eksekusi ke baris kode yang melakukan pemanggilan terhadap fungsi bersangkutan.

exit() : Berfungsi untuk menghentikan eksekusi kode di dalam file yang sedang aktif. Fungsi ini dapat dipanggil menggunakan sebuah parameter berupa string.

Tugas.

Buat sebuah program untuk menghitung kelipatan perkalian menggunakan pengulangan. Misalnya, $1 \times 2, \dots, 1 \times \text{ke-}n$.