

## PERTEMUAN 3

### Pokok Bahasan :

- ◆ Struktur Kendali
- ◆ Struktur Kendali Percabangan
- ◆ Struktur Kendali Pengulangan
- ◆ Struktur Kendali Perpindahan

#### 1.1 Struktur Kendali

Struktur kendali merupakan pengatur aliran program, mempunyai rangkaian perintah yang harus ditulis untuk memenuhi beberapa keadaan, yaitu :

- ✓ Mengulang suatu perintah jika terpenuhi suatu kondisi.
- ✓ Melanjutkan sebuah pernyataan bila kondisi terpenuhi.
- ✓ Memilih sebuah pilihan dari beberapa alternatif bila kondisi terpenuhi.

Ada beberapa contoh struktur kendali, yaitu struktur kendali percabangan (*branching*), struktur kendali pengulangan (*looping*), dan perpindahan (*jumping*).

#### 1.2 Struktur Kendali Percabangan

Struktur kendali percabangan (pengambilan keputusan) struktur kendali yang berfungsi untuk melakukan pemilihan atas perintah yang akan dijalankan sesuai dengan kondisi tertentu. Ada empat perintah percabangan dalam PHP, yaitu **if**, **if..else**, **if..elseif**, dan **switch**.

##### 1.2.1 Perintah IF

Perintah **if** digunakan untuk menjalankan satu atau lebih perintah berdasarkan suatu kondisi. Sintaks penulisan perintah if adalah sebagai berikut :

```
if (kondisi)
{
    pernyataan yang akan dijalankan apabila kondisi benar
}
```

Pada bentuk ini, bagian **pernyataan akan dijalankan** hanya kalau bagian kondisi bernilai **benar**.

Contoh penggunaan struktur kendali **if** :

File : **if.php**

```
<html>
<head>
  <title>Struktur Kendali IF</title>
</head>
<body>
  <?php
    $nilai = 90;
    if($nilai >= 60)
    {
        echo "Nilai Anda $nilai, Anda Lulus";
    }
  ?>
</body>
</html>
```

Penjelasannya :

Pada program diatas, terdapat pemeriksaan kondisi apakah isi variabel \$nilai lebih dari sama dengan 60. Jika kondisi ini bernilai **TRUE** (variabel \$nilai lebih dari sama dengan 60) maka perintah echo “Nilai Anda \$nilai, Anda Lulus” akan di jalankan. Sebaliknya jika kondisinya **FALSE** maka perintah tersebut tidak dijalankan.

### 1.2.2 Perintah IF - ELSE

Perintah **if...else** digunakan untuk memilih salah satu pernyataan berdasarkan suatu kondisi. Perintah ini akan menjalankan pernyataan tertentu bila kondisi bernilai benar dan akan menjalankan pernyataan yang lain jika kondisi bernilai salah. Sintaks penulisannya adalah sebagai berikut :

```
if (kondisi)
{
    pernyataan_1
}
else
{
    pernyataan_2
}
```

Pada bentuk ini **pernyataan\_1** dijalankan kalau kalau kondisi bernilai **benar**, dan **pernyataan\_2** dijalankan apabila kondisi bernilai **salah**.

Contoh penggunaan struktur kendali **if...else** :

File : **ifelse.php**

```
<html>
<head>
  <title>Struktur Kendali IF..ELSE</title>
</head>
<body>
  <?php
    $nilai = 40;
    if($nilai >= 60) {
      echo "Nilai Anda $nilai, Anda LULUS";
    }
    else {
      echo "Nilai Anda $nilai, Anda GAGAL";
    }
  ?>
</body>
</html>
```

File : **ifelsedinamis.php**

```
<html>
<head>
  <title>Struktur Kendali IF..ELSE</title>
</head>
<body>
  <form method="post" action="ifelsedinamis.php">
    Masukkan Nilai = <input type="text" name="nilai"><br/>
    <input type="submit" name="oke" value="Kirim">
  </form>

  <?php
    if(isset($_POST['oke']))
    {
      $nilai = $_POST['nilai'];

      if($nilai >= 60) {
        echo "Nilai Anda $nilai, Anda LULUS";
      }
      else {
        echo "Nilai Anda $nilai, Anda GAGAL";
      }
    }
  ?>
</body>
</html>
```

### 1.2.3 Perintah IF - ELSEIF

Perintah **if...elseif** digunakan untuk menjalankan suatu pernyataan dengan melibatkan lebih dari satu kondisi. Sintaks penulisannya sebagai berikut :

```
if (kondisi_1)
{
    pernyataan_1
}
elseif (kondisi_2)
{
    pernyataan_2
}
else
{
    pernyataan_3
}
```

Pada bentuk ini **pernyataan\_1** dijalankan kalau kalau **kondisi\_1** bernilai benar, dan **pernyataan\_2** dijalankan apabila **kondisi\_2** bernilai benar. Sedangkan **pernyataan\_3** akan dijalankan apabila **kondisi\_1** dan **kondisi\_2** bernilai salah.

Contoh penggunaan struktur kendali **if...elseif** :

File : **ifelseif.php**

```
<html>
<head>
  <title>Struktur Kendali IF..ELSEIF</title>
</head>
<body>
  <?php
    $jurusan = "SI";
    if ($jurusan == "KA")
    {
      echo "Komputerisasi akuntansi";
    }
    elseif ($jurusan == "MI")
    {
      echo "Manajemen Informatika";
    }
    elseif ($jurusan == "SI")
    {
      echo "Sistem Informasi";
    }
    elseif ($jurusan == "TI")
    {
      echo "Teknik Informatika";
    }
    else
    {
      echo "Jurusan $jurusan tidak terdaftar";
    }
  ?>

</body>
</html>
```

File : **ifelseifdinamis.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>Struktur Kendali IF..ELSEIF</title>
</head>
<body>
<?php
echo "<form method=GET action=ifelseifdinamis.php>
Masukkan Kode Jurusan : <input type=text name='jurusan' />
<input type=submit name='submit' value=Kirim />
</form>";

if(isset($_GET['submit']))
{
    if ($_GET['jurusan'] == "KA")
    {
        echo "Komputerisasi Akuntansi";
    }
    elseif ($_GET['jurusan'] == "MI")
    {
        echo "Manajemen Informatika";
    }
    elseif ($_GET['jurusan'] == "SI")
    {
        echo "Sistem Informasi";
    }
    elseif ($_GET['jurusan'] == "TI")
    {
        echo "Teknik Informatika";
    }
    else
    {
        echo "Jurusan $_GET[jurusan] tidak terdaftar";
    }
}
?>
</body>
</html>
```

### 1.2.4 Perintah Switch

Perintah `switch` digunakan sebagai alternatif pengganti dari perintah `if...elseif`. Dengan perintah ini program percabangan akan lebih mudah dibuat dan dipelajari. Sintaks penulisan perintah `switch` adalah sebagai berikut :

```
switch (kondisi)
{
    case konstanta_1:
        pernyataan_1;
        break;
    case konstanta_2:
        pernyataan_2;
        break;
    default:
        pernyataan_3;
}
```

Perintah `switch` akan menyeleksi kondisi yang diberikan dan membandingkan hasilnya dengan konstanta-konstanta yang berada pada `case`. Pembandingan akan dimulai dari `konstanta_1` sampai konstanta terakhir. Jika hasil dari kondisi sama dengan nilai konstanta tertentu, maka pernyataan pada konstanta tersebut akan dijalankan sampai ditemukan pernyataan `break`. Jika hasil dari kondisi tidak ada yang sama dengan konstanta-konstanta yang diberikan, maka pernyataan pada `default` yang akan dijalankan.

Contoh penggunaan struktur kendali **`switch`** :

File : **`switch.php`**

```
<?php
$jurusan = "SI";
switch($jurusan)
{
    case "KA":
        echo "$jurusan = Komputerisasi akuntansi";
        break;
    case "MI":
        echo "$jurusan = Manajemen Informatika";
        break;
    case "SI":
        echo "$jurusan = Sistem Informasi";
        break;
    case "TI":
        echo "$jurusan = Teknik Informatika";
        break;
    default:
        echo "$jurusan tidak terdaftar";
}
?>
```

### 1.3 Struktur Kendali Perulangan

Struktur kendali pengulangan digunakan untuk mengulang suatu perintah sebanyak yang diinginkan. Contohnya : menampilkan nomor bilangan 1 s/d 10. Kalau kita menuliskan secara manual, maka sangat tidak efisien apalagi jika kita ingin menampilkan bilangan 1 s/d 1000. Ada beberapa perintah pengulangan yang dapat digunakan, diantaranya :

**For, While, Do ... While, dan Foreach.**

#### 1.3.1 Perintah For

Perintah **for** digunakan untuk mengulangi suatu perintah dengan jumlah pengulangan yang sudah diketahui. Pada perintah ini tidak perlu menuliskan suatu kondisi untuk diuji. Kita hanya perlu menuliskan nilai awal dan akhir variabel penghitung. Nilai variabel penghitung akan secara otomatis bertambah atau berkurang tiap kali sebuah pengulangan dilaksanakan. Sintaks penulisan perintah **for** adalah sebagai berikut :

```
for (nilai_awal; nilai_akhir; penambahan/pengurangan)  
{  
    pernyataan yang dijalankan  
}
```

Contoh struktur pengulangan dengan **for** :

File : **for1.php**

```
<html>  
<head>  
    <title>Struktur pengulangan dengan FOR</title>  
</head>  
<body>  
    <?php  
        for($x=1;$x<=100;$x++)  
        {  
            echo "Angka $x<br>";  
        }  
    ?>  
</body>  
</html>
```



**File : for2.php**

```
<html>
<head>
  <title>Struktur pengulangan dengan FOR</title>
</head>
<body>
  <?php
    for($x=100;$x>=1;$x--)
    {
        echo "Angka $x<br>";
    }
  ?>
</body>
</html>
```

**File : for3.php**

```
<html>
<head>
  <title>Struktur pengulangan dengan FOR</title>
</head>
<body>
  <?php
    for($r=1; $r<=10; $r++)
    {
        for($s=1; $s<$r; $s++)
        {
            echo "$s";
        }
        echo "<br>";
    }
  ?>
</body>
</html>
```

**Catatan Penting!**

Pada perulangan **for** ada operator perbandingan, jika pada pengulangan increment atau menaik bentuk operatornya adalah '<=', namun dalam pengulangan decrement atau menurun bentuk operatornya adalah '>=', apabila peraturan ini tidak dipatuhi maka hasilnya adalah error.

### 1.3.2 Perintah While

Perintah **while** digunakan untuk mengulangi suatu perintah sampai jumlah yang belum bisa ditentukan. Pengulangan akan terus berjalan selama kondisi masih bernilai benar. Sintaks penulisan perintah **while** adalah sebagai berikut :

```
while (kondisi)
{
    pernyataan yang akan dijalankan
}
```

Contoh struktur pengulangan dengan **while** :

File : **while.php**

```
<html>
<head>
    <title>Struktur pengulangan dengan WHILE</title>
</head>
<body>
    <?php
        $i=1;
        while ($i <= 6)
        {
            echo "<h$i>Ini adalah contoh heading $i</h$i>";
            $i++;
        }
    ?>
</body>
</html>
```

### 1.3.3 Perintah Do ... While

Perintah **do...while**, proses pengulangan akan terus dikerjakan jika kondisi yang diperiksa di **while** masih bernilai benar. Proses pengulangan akan dihentikan jika kondisi sudah bernilai salah. Sintaks penulisannya sebagai berikut :

```
do
{
    pernyataan yang dijalankan
}
while (kondisi);
```

Perbedaan antara perintah **while** dengan **do...while** adalah terletak dari kondisi yang diperiksa. Pada perintah **while**, kondisi yang diperiksa terletak di awal perulangan, sehingga sebelum masuk ke dalam perulangan **while** kondisi harus bernilai benar.

Sedangkan pada perintah **do...while**, kondisi diperiksa di akhir perulangan. Ini berarti bahwa paling sedikit sebuah perulangan akan dilakukan oleh perintah **do...while**, karena untuk masuk ke perulangan tidak ada kondisi yang harus dipenuhi.

Contoh struktur pengulangan dengan **do...while** :

File : **dowhile.php**

```
<html>
<head>
  <title>Struktur pengulangan dengan do...while</title>
</head>
<body>
  <?php
    $i = 1;
    do
    {
      echo "Ini juga pengulangan yang ke-$i<br>";
      $i++;
    }
    while ($i <= 6);

    $j = 5;
    do
    {
      echo "<br>Perulangan ini kondisinya tidak terpenuhi<br>";
      $j++;
    }
    while($j <= 3);
  ?>
</body>
</html>
```

### 1.3.4 Perintah Foreach

Perintah **foreach** merupakan sebuah perulangan pada variabel bertipe array. Sintaks penulisan **foreach** sebagai berikut :

```
foreach (variabel array as variabel lain)
{
  pernyataan yang dijalankan
}
```

Contoh struktur pengulangan dengan **foreach** :

File : **foreach.php**

```
<html>
<head>
  <title>Struktur pengulangan dengan foreach</title>
</head>
<body>
  <?php
    $a = array("Cirebon", "Jakarta", "Bandung", "Yogyakarta");
    foreach ($a as $data)
    {
        echo "$data <br>";
    }
  ?>
</body>
</html>
```

## 1.4 Struktur Kendali Perpindahan

### 1.4.1. Break

Break adalah pernyataan yang digunakan untuk keluar dari proses, penggunaan Break hanya berlaku pada percabangan ataupun perulangan.

Contoh struktur perpindahan dengan **break** :

File : **break.php**

```
<html>
<head>
  <title>Struktur perpindahan dengan break</title>
</head>
<body>
  <?php
    $x= 1;
    while ($x <= 5) {
        if ($x==3) {
            break;
        }
        echo "Nilai x : ".$x."<br />";
        $x++;
    }
  ?>
</body>
</html>
```

### 1.4.2. Continue

Continue adalah pernyataan yang digunakan untuk melewati proses, penggunaan Continue hanya berlaku pada perulangan.

Contoh struktur perpindahan dengan **continue** :

File : **continue.php**

```
<html>
<head>
  <title>Struktur perpindahan dengan continue</title>
</head>
<body>
  <?php
    for($x=1;$x<=5;$x++){
      if($x == 3){
        continue;
      }
      echo "Nilai x : ".$x."<br />";
    }
  ?>
</body>
</html>
```

### 1.4.3. Return

Return adalah pernyataan yang digunakan untuk keluar dari alur dan mengembalikan nilai dari suatu fungsi, penggunaan Return hanya berlaku pada fungsi.

Contoh struktur perpindahan dengan **return** :

File : **return.php**

```
<html>
<head>
  <title>Struktur perpindahan dengan return</title>
</head>
<body>
  <?php
    function perkalian($x, $y){
      return $x * $y;
    }
    echo perkalian(3,5);
  ?>
</body>
</html>
```

#### 1.4.4. Exit

Exit adalah pernyataan yang digunakan untuk keluar dari sebuah program. Walaupun terdapat statements-statements lain dibawah baris kode exit yang belum dikerjakan.

Contoh struktur perpindahan dengan **exit** :

File : **exit.php**

```
<html>
<head>
  <title>Struktur perpindahan dengan exit</title>
</head>
<body>
  <?php
    for($x=1;$x<=5;$x++){
      if($x == 3){
        exit();
      }
      echo "Nilai x : ".$x."<br />";
    }
    //statement dibawah ini tidak akan dijalankan
    echo "STMIK AMIKOM Yogyakarta";
    echo "Tempat kuliah orang berdasi";
  ?>
</body>
</html>
```

## Referensi

Arief, M. R. 2011. *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi Offset.