

## STRUKTUR KENDALI

### 1. TUJUAN:

- Memahami dan menguasai penggunaan Struktur kendali percabangan pada PHP
- Memahami dan menguasai penggunaan struktur kendali Perulangan
- Memahami dan menguasai penggunaan struktur kendali perpindahan

### 2. MATERI:

#### 2.1 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*).

##### 2.1.1 Struktur Kendali Percabangan/Branching

Struktur kendali percabangan merupakan struktur kendali yang berfungsi untuk melakukan pemilihan atas instruksi yang akan dijalankan berdasarkan kondisi tertentu. Ada 4 macam perintah dalam struktur kendali percabangan:

- ✓ IF
- ✓ IF...else
- ✓ IF....else if
- ✓ Switch

##### 2.1.1.1 IF Statement

IF statement digunakan untuk mengeksekusi beberapa code jika satu kondisi terpenuhi/benar.

**Syntax:**

```
if (kondisi) {  
    //kode dijalankan jika kondisi benar/terpenuhi;  
}
```

Kondisi ini bisa berupa operasi logika, operasi perbandingan

Contoh:

- Menggunakan Operasi Perbandingan

```
1 <?php
2     $pertemuan=14;
3     $presensi=10;
4     echo $kehadiran=$presensi/$pertemuan;
5     if($kehadiran>0.75){
6         echo"boleh mengikuti ujian";
7     }
8 ?>
```

- Menggunakan Operasi Logika

```
<?php
    $pertemuan=14;
    $presensi=12;
    $tugas=false;
    echo $kehadiran=$presensi/$pertemuan;
    if($kehadiran>0.75 AND $tugas==true){
        echo "boleh mengikuti ujian";
    }
?>
```

#### 2.1.1.2 IF...ELSE Statement

**IF...ELSE** Statement digunakan untuk mengeksekusi beberapa kode jika kondisi benar dan menjalankan beberapa code jika kondisi tidak terpenuhi/salah.

**Syntax:**

```
if (kondisi) {
    // code dieksekusi jika kondisi terpenuhi
} else {
    //code dieksekusi jika kondisi tidak terpenuhi
}
```

**Contoh:**

```
1  <?php
2      $pertemuan=14;
3      $presensi=10;
4      echo $kehadiran=$presensi/$pertemuan;
5      if($kehadiran>0.75){
6          echo"boleh mengikuti ujian";
7      }else{
8          echo"tidak boleh mengikuti ujian";
9      }
10  ?>
```

```
2.  <?php
      $pertemuan=14;
      $presensi=12;
      $tugas=false;
      echo $kehadiran=$presensi/$pertemuan;
      if($kehadiran>0.75 AND $tugas==true){
          echo "boleh mengikuti ujian";
      }
      else{
          echo "tidak boleh mengikuti ujian";
      }
      ?>
```

IF else juga bisa dibuat dalam bentuk **nested**

**Contoh:**

```
<?php
    $pertemuan=14;
    $presensi=12;
    $kehadiran=$presensi/$pertemuan;
    $tugas=true;
    $nilai=0;
    if($kehadiran>0.75){
        echo "boleh mengikuti ujian";
        if($tugas==true){
            $nilai=$nilai+10;
        }
    }
    else{
        echo "tidak boleh mengikuti ujian";
    }
    ?>
```

### 2.1.1.3 IF...ELSE IF Statement

**IF ...ELSE IF** Statement digunakan untuk mengeksekusi beberapa code yang berbeda untuk lebih dari 2 kondisi

**Syntax:**

```
if (kondisi 1) {  
    //code dijalankan jika kondisi 1 terpenuhi  
}  
elseif (kondisi 2) {  
    //code dijalankan jika kondisi 2 terpenuhi  
}  
else {  
    //code dijalankan jika semua kondisi tidak terpenuhi  
}
```

**Contoh:**

1.

```
1 <?php  
2     $pertemuan=14;  
3     $presensi=10;  
4     $kehadiran=$presensi/$pertemuan;  
5     $tugas_tambahan=true;  
6     if($kehadiran>0.75){  
7         echo"boleh mengikuti ujian";  
8     }elseif($tugas_tambahan==true){  
9         echo"boleh mengikuti ujian";  
10    }else{  
11        echo"tidak boleh mengikuti ujian";  
12    }  
13 ?>
```

2.

```
<?php  
$kode=12;  
if($kode==11){  
    $prodi="S1-Informatika";  
}  
else if($kode==12){  
    $prodi="S1-Sistem Informasi";  
}  
else if($kode==01){  
    $prodi="D3 Teknologi Informasi";  
}  
else if($kode==02){  
    $prodi="D3 Manajemen Informasi";  
}  
else{  
    $prodi="Belum terdaftar";  
}  
echo $prodi;  
?>
```

#### 2.1.1.4 SWITCH Statement

**SWITCH** Statement digunakan untuk memilih beberapa blok dari kode untuk dieksekusi. Perintah `switch` akan menyeleksi kondisi yang diberikan dan membandingkan hasilnya dengan label-label yang berada pada `case`. Pembandingan akan dimulai dari label 1 sampai label terakhir. Jika hasil dari kondisi sama dengan nilai label tertentu, maka pernyataan pada label tersebut akan dijalankan sampai ditemukan pernyataan `break`. Jika hasil dari kondisi tidak ada yang sama dengan label-label yang diberikan, maka pernyataan pada `default` yang akan dijalankan.

**Syntax:**

```
switch (n) {  
    case Label1:  
        //code dijalankan jika n=Label 1;  
        break;  
    case Label2:  
        //code dijalankan jika n=Label 2;  
        break;  
    case Label3:  
        //code dijalankan jika n=Label 3;  
        break;  
    ...  
    default:  
        //code dijalankan jika n berbeda dari semua label  
}
```

Contoh:

```
1  <?php
2      $lampu="merah";
3      switch($lampu){
4          case "merah":
5              echo"harus berhenti";
6              break;
7          case "kuning":
8              echo"harus hati-hati";
9              break;
10         case "hijau":
11             echo"jalan terus";
12             break;
13         default:
14             echo"jalan terus";
15     }
16  ?>
```

### 2.1.2 Struktur Kendali Perulangan/Looping

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

#### 2.1.2.1 Perintah While

Perintah **While** digunakan untuk menjalankan baris program selama kondisi terpenuhi. Pengulangan akan terus dijalankan selama kondisi terpenuhi/benar.

Syntax:

```
while (kondisi terpenuhi) {

    //kode program dijalankan;

}
```

Contoh:

```
1 <?php
2     $x = 1;
3
4     while($x <= 5) {
5         echo "belajar pemograman web ke-".$x."<br>";
6         $x++;
7     }
8 ?>
```

#### 2.1.2.2 Do...while

Perintah **Do..while** digunakan untuk menjalankan baris program 1 kali kemudian mengecek kondisinya dan mengulangi loop jika kondisi terpenuhi.

Syntax:

```
do {
    //code program akan dijalankan;
}
while (kondisi terpenuhi);
```

Perbedaan perintah while dengan do..while adalah terletak pada kondisi yang diperiksa. Pada perintah while, kondisi yang diperiksa terletak di awal perulangan, sehingga sebelum masuk perulangan while, kondisi harus benar. Sedangkan pada perintah do..while, kondisi diperiksa di akhir perulangan. Ini berarti, minimal ada 1 perulangan yang pasti dijalankan oleh perintah do..while, karena untuk masuk ke perulangan tidak ada kondisi yang harus dipenuhi.

Contoh:

```
1 <?php
2     $x = 1;
3     do{
4         echo "belajar pemograman web ke-".$x."<br>";
5         $x++;
6     }
7     while($x <= 5)
8 ?>
```

### 2.1.2.3 For

Perintah for digunakan jika kita tau ingin melakukan perulangan berapa kali.

Syntax:

```
for (init counter; test counter; increment counter) {  
  
    //code akan dijalankan;  
  
}
```

Contoh:

1. Perulangan dengan increment counter

```
<?php  
  
    for($i=1;$i<=5;$i++){  
        echo"belajar PWL pertemuan ke-".$i."<br>";  
    }  
?>
```

```
<?php  
  
    for($i=1;$i<=5;){  
        echo"belajar PWL pertemuan ke-".$i."<br>";  
        $i++;  
    }  
?>
```

2. Perulangan dengan decrement counter

```
<?php  
  
    for($i=5;$i>=1;$i--){  
        echo"belajar PWL pertemuan ke-".$i."<br>";  
    }  
?>
```



```
<?php
    for($i=5;$i>=1;){
        echo"belajar PWL pertemuan ke-".$i."<br>";
        $i--;
    }
?>
```

#### 2.1.2.4 Foreach

Perintah **foreach** merupakan perulangan yang digunakan untuk array, dengan menggunakan pasangan key dan value dari array.

**Syntax:**

```
foreach ($array as $value) {
    //program akan dijalankan;
}
```

**Contoh:**

```
1 <?php
2 $buah = array("apel", "jeruk", "anggur", "strawbery");
3
4 foreach ($buah as $value) {
5     echo "$value <br>";
6 }
7 ?>
```

### 2.1.3 Struktur Kendali Perpindahan/Jumping

#### 2.1.3.1 Break

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

Contoh:

```
<?php
    $i=1;
    while($i<=5){
        if($i==4){
            break;
        }
        echo "belajar PWL pertemuan ke-".$i."<br>";
        $i++;
    }
?>
```

#### 2.1.3.2 Continue

**Continue** merupakan pernyataan yang digunakan untuk melewati proses, penggunaan **continue** hanya berlaku pada perulangan.

Contoh:

```
<?php
    for($i=1;$i<=5;$i++){
        if($i==4){
            continue;
        }
        echo "belajar PWL pertemuan ke-".$i."<br>";
    }
?>
```

#### 2.1.3.3 Return

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

Contoh:

```
<?php
    function luas_persegi($p){
        $luas=$p*$p;
        return $luas;
    }
    echo luas_persegi(5);
?>
```

#### 2.1.3.4 Exit

**Exit** merupakan pernyataan yang digunakan untuk keluar dari sebuah program, walaupun masih ada statement-statement lain di bawah baris code **exit** yang belum dikerjakan.

**Contoh:**

```
<?php
function luas_persegi($p){
    $luas=$p*$p;
    return $luas;
}
echo luas_persegi(5);
exit();
//baris ini tidak akan dieksekusi
echo "sudah keluar dari program";
?>
```