

## Bab 3

# Struktur Kondisi dan Perulangan

- ❖ Struktur Kondisi
  - + Struktur Kondisi If
  - + Struktur Kondisi If...Else
  - + Struktur Kondisi Khusus ? :
  - + Struktur Kondisi Switch...Case
- ❖ Struktur Perulangan
  - + Struktur Kondisi For
  - + Struktur Kondisi While
  - + Struktur Kondisi Do...While
  - + Struktur Kondisi Foreach
- ❖ Struktur Break dan Continue

### Struktur Kondisi

---

#### Struktur Kondisi If

```
if (kondisi) {  
    statement-jika-kondisi-true;  
}
```

##### Keterangan :

- **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.

#### Struktur Kondisi If ... Else

```
if (kondisi) {  
    statement-jika-kondisi-true;  
} else {  
    statement-jika-kondisi-false;  
}
```

##### Keterangan :

- **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya. Jika kondisinya TRUE maka statemen yang berada di blok if akan dieksekusi, sebaliknya jika kondisinya FALSE maka statemen yang berada di blok else yang akan dieksekusi.

#### Struktur Kondisi Khusus ? :

```
(kondisi) ? benar : salah;
```

##### Keterangan :

- **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya. Statement pada blok benar dan salah hanya boleh satu statemen saja

#### Struktur Kondisi Switch ... Case

```
switch ($var) {  
    case '1' : statement-1; break;  
    case '2' : statement-2; break;  
    ....  
}
```

Keterangan :

- **\$var** merupakan variabel yang akan diperiksa isi atau nilainya. Tipe data variabel ini tidak dibatasi.
- Value pada **case** juga bisa berupa string, integer, boolean, bahkan bisa berupa conditional-statement. Boleh memakai kutip tunggal maupun kutip ganda.

Program 3.1

Nama File : *if.php*

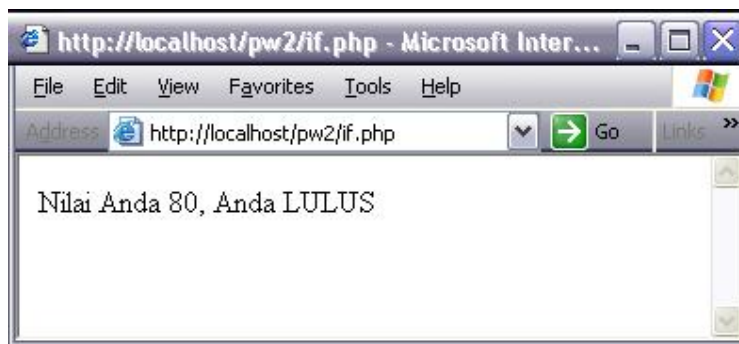
Deskripsi : *Program Struktur Kondisi If.*

```

1 <?php
2 $nilai = 80;
3 if ($nilai >= 60) {
4     echo "Nilai Anda $nilai, Anda LULUS";
5 }
6 ?>
```

Penjelasan Program 3.1

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka *statement* pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka *statement* pada baris ke-4 tidak akan dijalankan.



Gambar 3.1. Tampilan Program 3.1

Program 3.2

Nama File : *if\_else.php*

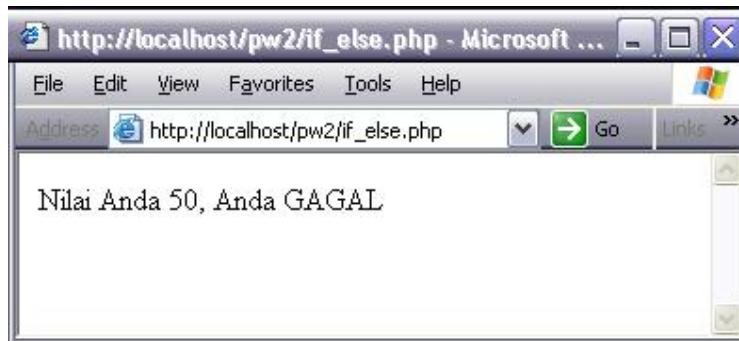
Deskripsi : *Program Struktur Kondisi If..Else.*

```

1 <?php
2 $nilai = 50;
3 if ($nilai >= 60) {
4     echo "Nilai Anda $nilai, Anda LULUS";
5 } else {
6     echo "Nilai Anda $nilai, Anda GAGAL";
7 }
8 ?>
```

Penjelasan Program 3.2

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka *statement* pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka *statement* pada baris ke-6 yang akan dijalankan. Pada contoh program di atas, kondisi pada baris ke-3 bernilai **FALSE** karena isi variabel \$nilai adalah 50.



Gambar 3.2. Tampilan Program 3.2

Program 3.3

Nama File : *if\_else2.php*

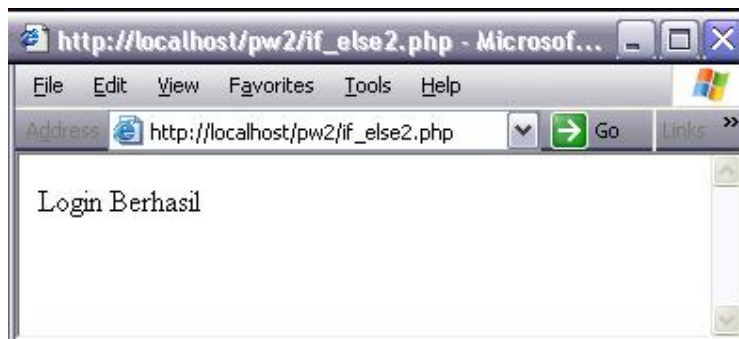
Deskripsi : Program Memeriksa username dan password dengan If..Else.

```

1 <?php
2 $user = "achmatim";
3 $pass = "123"
4 if ($user == "achmatim" && $pass == "123") {
5     echo "Login Berhasil";
6 } else {
7     echo "Login Gagal";
8 }
9 ?>
    
```

Penjelasan Program 3.3

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** sama dengan "achmatim" **dan** apakah isi variabel **\$pass** sama dengan "123" (baris 4). Jika kedua kondisi tersebut bernilai **TRUE** maka statement pada baris ke-5 akan dijalankan. Sebaliknya jika salah satunya bernilai **FALSE**, maka statement pada baris ke-7 yang akan dijalankan.



Gambar 3.3. Tampilan Program 3.3

Program 3.4

Nama File : *if\_var.php*

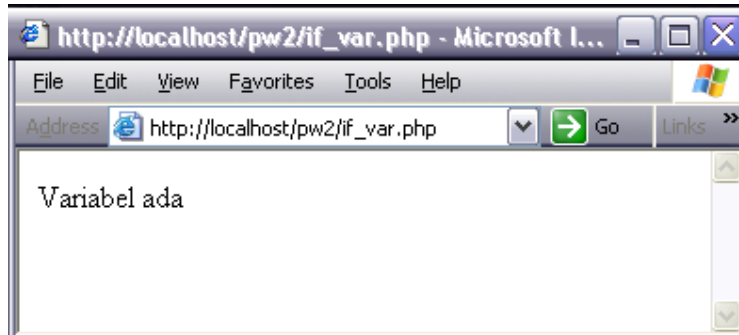
Deskripsi : Program Memeriksa suatu variabel ada atau tidak.

```

1 <?php
2 $user="";
3 if (!isset($user)) {
4     echo "variabel tidak ada/belum terbentuk";
5 } else {
6     echo "variabel ada";
7 }
8 ?>
    
```

### Penjelasan Program 3.4

Pada program di atas, terdapat fungsi **isset()** yang merupakan fungsi untuk memeriksa apakah suatu variabel sudah terbentuk (ada) atau belum. Fungsi akan bernilai **TRUE** jika variabelnya ada. Jadi jika baris ke-2 dari program di atas di-remark atau dihilangkan maka *statement* pada baris ke-4 yang akan dijalankan.



Gambar 3.4. Tampilan Program 3.4

### Program 3.5

Nama File : *switch.php*

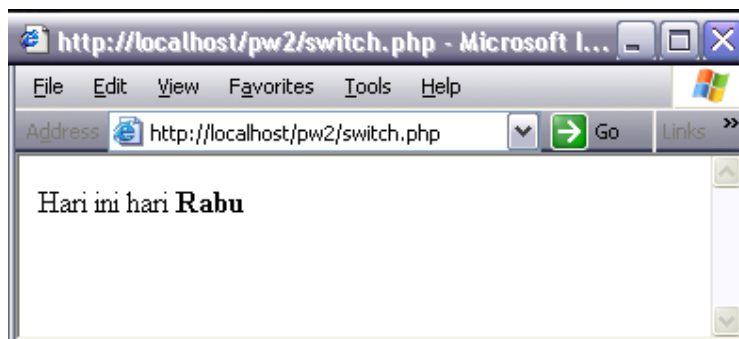
Deskripsi : Program Struktur Switch..Case untuk menampilkan nama hari dalam bahasa Indonesia.

```

1 <?php
2 $day = date ("D");
3 switch ($day) {
4     case 'Sun' : $hari = "Minggu"; break;
5     case 'Mon' : $hari = "Senin"; break;
6     case 'Tue' : $hari = "Selasa"; break;
7     case 'Wed' : $hari = "Rabu"; break;
8     case 'Thu' : $hari = "Kamis"; break;
9     case 'Fri' : $hari = "Jum'at"; break;
10    case 'Sat' : $hari = "Sabtu"; break;
11    default  : $hari = "Kiamat";
12 }
13 echo "Hari ini hari <b>$hari</b>";
14 ?>
    
```

### Penjelasan Program 3.5

Pada program di atas, variabel **\$day** (baris ke-2) akan berisi 3 (tiga) digit pertama dari nama hari dalam bahasa Inggris. Dengan fungsi **date()**, kita akan memperoleh informasi tanggal, hari dan waktu sekarang. Fungsi ini akan dipelajari lebih lanjut pada Bab mendatang. Selanjutnya isi variabel **\$day** akan diperiksa dengan switch (baris ke-3), jika isinya adalah 'Sun' maka **\$hari** adalah "Minggu" dan seterusnya.



Gambar 3.5. Tampilan Program 3.5

**Program 3.6**

Nama File : *if\_khusus.php*

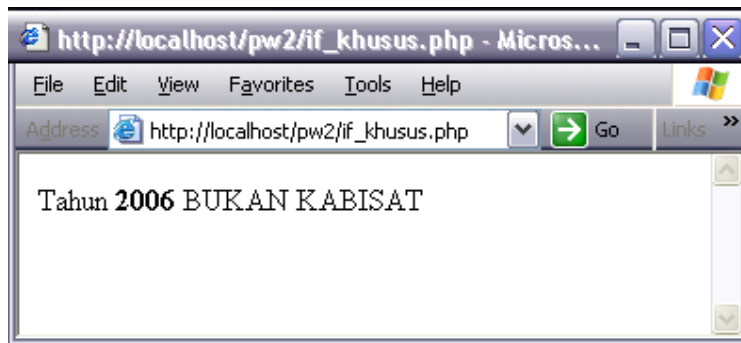
Deskripsi : *Program Struktur Kondisi Khusus ? untuk memeriksa tahun kabisat.*

```

1 <?php
2 $tahun = date ("Y");
3 $kabisat = ($tahun%4 == 0) ? "KABISAT" : "BUKAN KABISAT";
4 echo "Tahun <b>$tahun</b> $kabisat";
5 ?>
    
```

**Penjelasan Program 3.6**

Pada program di atas, fungsi **date()** dengan parameter "Y" (baris ke-2) akan menghasilkan 4 digit tahun sekarang. Pada baris ke-3 akan diperiksa apakah isi variabel \$tahun jika dimodulus dengan 4 akan menghasilkan nilai 0. Jika **TRUE** maka \$kabisat akan berisi "KABISAT" dan jika **FALSE** \$kabisat akan berisi "BUKAN KABISAT"



Gambar 3.6. Tampilan Program 3.6

**Struktur Perulangan**

Struktur Perulangan For

```

for (init_awal, kondisi, counter) {
    statement-yang-diulang;
}
    
```

Keterangan :

- **init\_awal** merupakan inisialisasi atau nilai awal variable.
- **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.

Struktur Perulangan While

```

init_awal;
while (kondisi) {
    statement-yang-diulang;
    counter;
}
    
```

Keterangan :

- **init\_awal** merupakan inisialisasi atau nilai awal variable.
- **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.

### Struktur Perulangan Do ... while

```

init_awal;
do {
    statement-yang-diulang;
    counter;
} while (kondisi);
    
```

Keterangan :

- **init\_awal** merupakan inisialisasi atau nilai awal variable.
- **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- **counter** merupakan penambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.
- Pada struktur do...while, pemeriksaan kondisi ada di bawah, sehingga statement yang berada dalam block do...while setidaknya akan dieksekusi sebanyak satu kali.

### Struktur Perulangan Foreach

```

foreach (array_expression as $value)
    statement;

foreach (array_expression as $key => $value)
    statement;
    
```

Keterangan :

- Struktur foreach biasanya digunakan untuk melakukan perulangan berdasarkan isi suatu array. Perulangan akan berakhir jika isi array telah habis.

### Program 3.7

Nama File : for.php

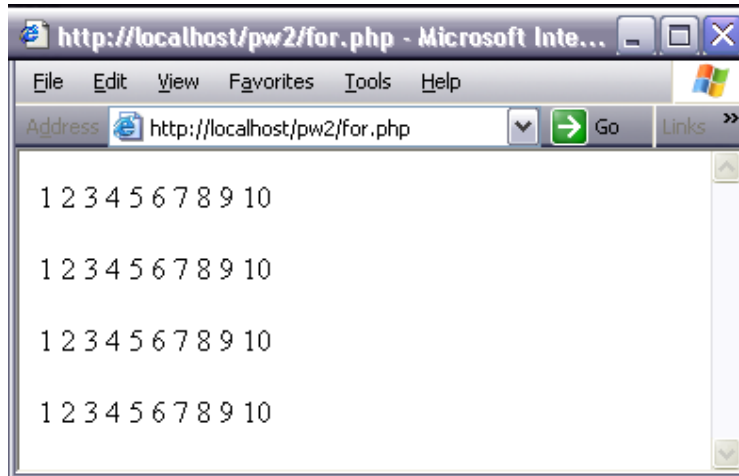
Deskripsi : Program Struktur Perulangan For dan beberapa variasinya.

```

1  <?php
2
3  /* contoh 1 */
4  for ($i = 1; $i <= 10; $i++) {
5      echo "$i ";
6  }
7  echo "<br><br>";
8
9  /* contoh 2 */
10 for ($i = 1; ; $i++) {
11     if ($i > 10) {
12         break;
13     }
14     echo "$i ";
15 }
16 echo "<br><br>";
17
18 /* contoh 3 */
19 $i = 1;
20 for (; ; ) {
21     if ($i > 10) {
22         break;
23     }
24     echo "$i ";
25     $i++;
26 } echo "<br><br>";
27
28 /* contoh 4 */
29 for ($i = 1; $i <= 10; print "$i ", $i++);
30
31 ?>
    
```

### Penjelasan Program 3.7

Program di atas merupakan bentuk-bentuk perulangan dengan menggunakan for. Contoh yang pertama (baris 4-6) merupakan bentuk yang paling umum. Pada contoh 2, batas akhir perulangan tidak disebutkan dalam for, tapi diatur dengan if dan **break** (baris 11-13).



Gambar 3.7. Tampilan Program 3.7

### Program 3.8

Nama File : *while.php*

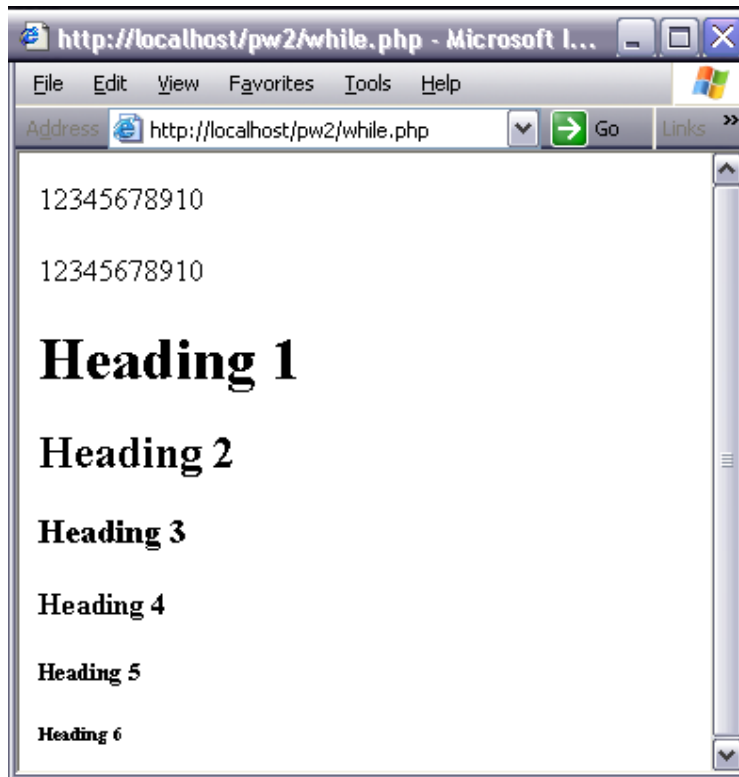
Deskripsi : *Program Struktur Perulangan While dan beberapa variasinya.*

```

1 <?php
2 /* contoh 1 */
3 $i = 1;
4 while ($i <= 10) {
5     echo $i++;
6 }
7 echo "<br><br>";
8
9 /* contoh 2 */
10 $i = 1;
11 while ($i <= 10):
12     echo "$i";
13     $i++;
14 endwhile;
15 echo "<br><br>";
16
17 /* contoh 3 */
18 $i = 1;
19 while ($i <= 6) {
20     echo "<h$i>Heading $i</h$i>";
21     $i++;
22 }
23 ?>
    
```

### Penjelasan Program 3.8

Program di atas merupakan beberapa variasi perulangan dengan while. Contoh yang pertama (baris 3-6) merupakan bentuk yang paling umum dipakai. Blok perintah (baris 5) akan dijalankan selama pemeriksaan kondisi pada baris ke-4 bernilai **TRUE**. Contoh kedua merupakan bentuk lain dari while, bentuk ini jarang digunakan. Pada contoh ketiga, penggunaan perulangan untuk menampilkan tulisan dengan format <H1> sampai <H6>



Gambar 3.8. Tampilan Program 3.8

Program 3.9

Nama File : dowhile.php

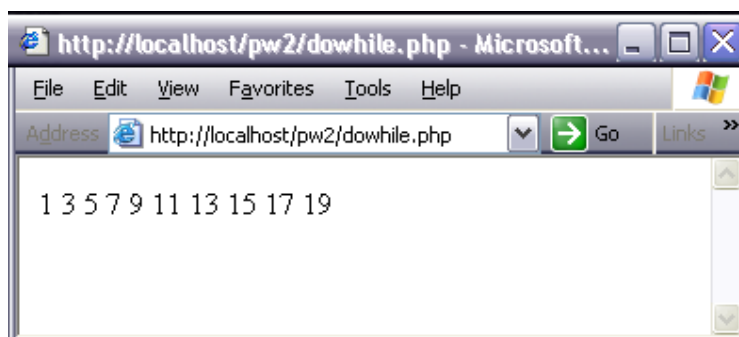
Deskripsi : Program Struktur Perulangan dengan Do...While.

```

1 <?php
2
3 $i = 1;
4 do {
5     echo "$i ";
6     $i+=2;
7 } while ($i <= 20);
8
9 ?>
```

Penjelasan Program 3.9

Program di atas akan menampilkan bilangan ganjil antara 1 sampai 20 menggunakan struktur perulangan **do..while**. Pada struktur perulangan do...while, pemeriksaan kondisi berada di bawah.



Gambar 3.9. Tampilan Program 3.9

## Struktur Break dan Continue

---

### Program 3.10

Nama File : *break.php*

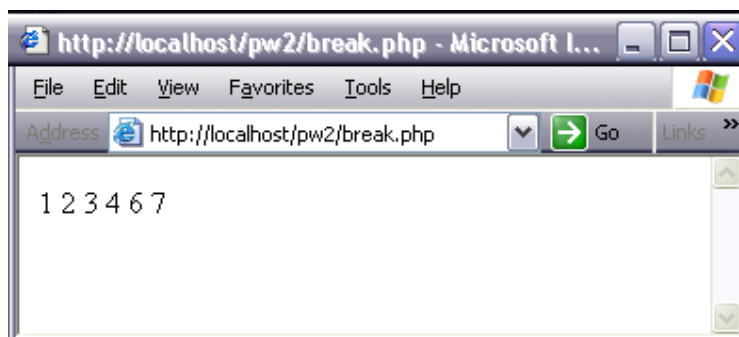
Deskripsi : *Program Struktur Break dan Continue.*

```

1  <?php
2
3  for ($i=1; $i<10; $i++) {
4      if ($i == 5)
5          continue;
6      if ($i == 8)
7          break;
8      echo "$i ";
9  }
10
11 ?>
```

### Penjelasan Program 3.10

Dari program di atas, dapat disimpulkan bahwa perintah **continue** akan melanjutkan proses perulangan tanpa melewati (meng-eksekusi) baris perintah di bawahnya. Jadi perintah pada baris 6-8 akan dilewati (5 tidak akan tercetak) Sedangkan dengan perintah **break** akan menyebabkan program menghentikan perulangan (langsung keluar dari perulangan)



Gambar 3.10. Tampilan Program 3.10