

MODUL 4

PERULANGAN

Dalam pembuatan script PHP sering kita menjalankan kode program yang sama berulang-ulang dalam satu urutan proses. Jika ditulis secara manual mungkin akan membutuhkan baris script yang sangat banyak. Untuk mempermudah dan meringkas penulisan kita bisa menggunakan perintah perulangan (looping). Perulangan pada PHP dapat dilakukan dengan perintah WHILE, DO...WHILE, FOR dan FOREACH.

1. Perulangan WHILE

perulangan kode program berdasarkan suatu kondisi yang bernilai benar. Jika kondisi tersebut bernilai salah maka perulangan berhenti.

Cara Penulisan WHILE:

```
while (kondisi) {  
    kode yang akan dijalankan;  
}
```

Contoh Penggunaan WHILE:

```
01. <?php  
02. $x=1;  
03.  
04. while($x<=5) {  
05.     echo "Angka: $x <br>";  
06.     $x++;  
07. }  
08. ?>
```

Penjelasan:

Pada baris pertama nilai variabel \$x diberi nilai 1 dengan perintah \$x=1. Kemudian nilai variabel \$x akan ditambahkan 1 dengan perintah \$x++ selama nilai variabel \$x lebih kecil atau sama dengan 5. Setelah itu perulangan \$x++ berhenti.

2. Perulangan DO – WHILE

Perintah untuk mengerjakan kode program lalu mengulang proses tersebut selama kondisi bernilai benar. Perulangan berhenti jika kondisi tidak terpenuhi lagi.

Cara Penulisan DO – WHILE

```
do {  
    kode yang akan dijalankan;  
}  
while (kondisi);
```

Contoh Penggunaan DO – WHILE

```
01. <?php
02. $x=1;
03.
04. do {
05.     echo "Angka: $x <br>";
06.     $x++;
07. } while ($x<=5);
08. ?>
```

Penjelasan

Pertama kali variabel \$x diberi nilai 1 (\$x=1). Kemudian menampilkan nilai \$x dan menambah \$x dengan 1 melalui perintah \$x++. Selanjutnya cek kondisi setelah while, apakah nilai \$x lebih kecil atau sama dengan 5. Jika ya maka akan diulang proses diatas sampai kondisi pada while tercapai.

“Perbedaan Antara WHILE Dan DO..WHILE”

Sepintas terlihat sama antara WHILE dan DO...WHILE, namun ada perbedaan mendasar yaitu terletak pada penempatan WHILE. Jika pada perulangan WHILE letak WHILE pada baris pertama perulangan maka pada DO...WHILE terletak pada akhir perulangan. Pada DO...WHILE sebuah proses pasti dikerjakan setidaknya satu kali, sedang pada WHILE ada kemungkinan tidak dikerjakan.

3. Perulangan FOR

Mengulang kode program dengan jumlah angka perulangan tertentu. Perulangan berhenti setelah tercapai angka perulangan yang diset pada inisialisasi FOR. Perulangan FOR dipakai jika kita sudah mengetahui berapa jumlah perulangan yang diinginkan.

Cara Penulisan FOR

```
for (nilai awal; tes kondisi nilai; operasi perubahan nilai)
{
    kode yang akan dijalankan;
}
```

Keterangan:

- **nilai awal**, adalah nilai yang diberikan saat FOR *pertama kali* dijalankan.
- **tes kondisi nilai**, adalah proses pengecekan nilai selama loop berlangsung. Jika bernilai benar maka proses loop diteruskan dan jika bernilai salah maka loop berhenti.
- **operasi perubahan nilai**, adalah operasi yang mengubah nilai selama proses loop yang membuat nilai mencapai kondisi nilai tes.

Contoh Penggunaan FOR

```
01. <?php
02. for ($x=0; $x<=10; $x++)
03. {
04.     echo "Angka: $x <br>";
05. }
06. ?>
```

Penjelasan

Pada program tampak perulangan FOR mengacu pada perubahan nilai variabel \$x. Untuk nilai awal \$x=0 lalu tes kondisi perulangan jika \$x kurang dari atau sama dengan 10 kemudian operasi perubahan nilai berupa \$x++. Dengan demikian jika program dijalankan akan tampil angka 1 sampai 10.

4. Perulangan FOREACH

Perulangan kode program untuk setiap elemen data di dalam array. Perulangan berhenti setelah data terakhir array. Perulangan FOREACH hanya bekerja pada array.

Cara Penulisan FOREACH

```
foreach ($array as $value)
{
    kode yang dijalankan;
}
```

Contoh Penggunaan FOREACH

```
01. <?php
02. $warna = array("merah", "hijau", "biru", "kuning");
03.
04. foreach ($warna as $value) {
05.     echo "$value <br>";
06. }
07. ?>
```

Penjelasan

Jika dijalankan, program diatas akan menampilkan semua data pada variabel array \$warna. Pada contoh, variabel array berisi nama-nama warna yaitu merah, hijau, biru dan kuning. Dengan perintah FOREACH maka semua data pada array \$warna akan ditampilkan satu-persatu sesuai urutan data pada array.