

# Kontrol Struktur II: Pengulangan (Loop)

Donny Reza, S.Kom.  
Aplikasi IT 2  
Program Studi Akuntansi  
Universitas Komputer Indonesia  
2013

# Pengulangan (Loop)

Loop (pengulangan) di dalam PHP digunakan untuk melakukan suatu proses yang dilakukan secara berulang.

Perintah loop di dalam PHP terdiri dari:

1. [while](#)
2. [do – while](#)
3. [for](#)
4. [foreach](#)

# while

Pengulangan **while** merupakan tipe pengulangan (*loop*) yang paling sederhana.

Syntax:

```
while ( ekspresi ){  
    statement;  
}
```

Proses: selama **ekspresi** bernilai **True**, maka **statement** akan dieksekusi.

# while

Contoh:

```
$n = 1;
```

```
while ( $n <= 10 ){  
    echo " nilai n = ".$n;  
    $n++;  
}
```

Proses:

1. PHP akan memeriksa ekspresi ( `$n <= 10` ). Jika bernilai **True**, maka proses di dalam blok **while** akan dieksekusi:
  - Eksekusi perintah `echo`,
  - Eksekusi perintah (*statement*) `$n++`, perintah ini akan menambahkan 1 angka ke dalam variabel `$n`;
2. Periksa kembali ekspresi ( `$n <= 10` ), jika masih bernilai **True**, proses di atas diulangi. Sedangkan jika bernilai **False**, proses loop dihentikan.

# do – while

Pengulangan **do – while** tidak berbeda dengan **while**. Hanya saja, pada **do – while** proses evaluasi terhadap ekspresi diperiksa di akhir, sementara pada **while** dilakukan di awal proses.

Syntax:

```
do {  
    statement;  
} while( ekspresi )
```

Proses evaluasi nilai **ekspresi** yang dilakukan di akhir, akan menyebabkan **statement** pasti diproses minimal satu kali.

# do - while

Contoh:

```
$n = 10;  
do {  
    echo "nilai n = ".$n;  
    $n++;  
} while ( $n <= 10 );
```

Proses:

1. Eksekusi perintah-perintah yang ada di dalam blok **do {}**,
2. Periksa ekspresi ( **\$n <= 10** ). Jika bernilai **True**, eksekusi kembali proses di dalam blok **do {}**. Jika bernilai **False**, hentikan proses.

# for

Pengulangan **for** merupakan yang paling kompleks di dalam PHP. Dilihat dari prosesnya, **for** lebih mirip dengan **while**

Syntax:

```
for ( expr1; expr2; expr3 ){  
    statement;  
}
```

Keterangan:

**expr1**, biasanya merupakan inisialisasi variabel kontrol.

**expr2**, merupakan pemeriksaan nilai kebenaran kondisi (True/False).

**expr3**, proses meningkatkan atau menurunkan nilai variabel kontrol, untuk dievaluasi pada kondisi **ekpr2**.

# for

Contoh:

```
for ( $i = 1; $i <=5; $i++ ){  
    echo "UNIKOM ";  
}
```

**Hasil:** Script di atas akan menampilkan tulisan UNIKOM sebanyak 5 kali.

Langkah Proses:

1. ekspresi `$i = 1` dikerjakan, biasanya merupakan proses inisialisasi variabel.
2. lakukan proses evaluasi pada ekspresi `$i <= 5`, jika bernilai **True**, eksekusi proses **echo "UNIKOM"**. Jika bernilai **False**, proses dihentikan.
3. lakukan proses `$i++`, yang merupakan proses *incremental* (menambah 1 pada variabel `$i`, sehingga nilai `$i` akan terus bertambah)
4. lakukan kembali proses pada **nomor 2**.



# foreach

**foreach** digunakan untuk melakukan iterasi pembacaan nilai yang terdapat pada sebuah Array.

Syntax 1:

```
foreach( $variabel_array as $namavariabel ){  
    statement;  
}
```

Syntax 2:

```
foreach( $variabel_array as $namaindex => $namavariabel ){  
    statement;  
}
```

# foreach

Contoh1:

```
$nilai = Array( 1,2,3 );
```

```
foreach ( $nilai as $nomor ){  
    echo $nomor;  
    echo "<br>";  
}
```

# foreach

Contoh2:

```
$nilai = Array(  
    1 => "Senin",  
    2 => "Selasa",  
    3 => "Rabu"  
);
```

```
foreach ( $nilai as $idx => $hari ){  
    echo "hari ke-".$idx." adalah hari ".$$hari;  
    echo "<br>";  
}
```