

Mengenal Mekanisme Looping Pada PHP

Sony Arianto Kurniawan

sony-ak@sony-ak.com

<http://www.sony-ak.com/>

Lisensi Dokumen:

Copyright © 2004 IlmuKomputer.Com

*Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.*

Setiap bahasa pemrograman pasti memiliki mekanisme untuk melakukan looping (perulangan) dalam code-nya. Bagaimana dengan PHP? PHP Sebagai bahasa pemrograman tentu saja juga memiliki mekanisme ini. Statemen-statemen looping pada PHP bertujuan untuk mengeksekusi suatu blok kode yang sama selama beberapa kali dan bisa ditentukan perulangannya.

PHP memiliki beberapa statemen yang digunakan untuk operasi looping yaitu:

- while, yaitu perulangan pada suatu blok kode selama kondisi tertentu masih terpenuhi.
- do..while, yaitu perulangan yang selalu dijalankan minimal sekali dan baru kemudian mengulangi blok kode nya selama suatu kondisi terpenuhi.
- for, yaitu perulangan yang sifatnya pasti pada suatu blok kode tertentu dan sudah ditentukan sebanyak jumlah tertentu.
- foreach, yaitu perulangan khusus yang digunakan pada array.

Mari kita bahas satu-persatu mengenai teknik looping di atas.

WHILE

Statemen while akan mengeksekusi suatu blok kode "jika dan selama" suatu kondisi looping terpenuhi.

Sintaks :

```
while (kondisi)
kode yang akan dieksekusi;
```

Contoh :

Contoh berikut akan menunjukkan suatu looping yang akan selalu berjalan selama nilai i adalah kurang dari atau sama dengan 5. Nilai i akan di-increase dengan 1 selama looping berjalan.

```
<html>
<body>
```

```
<?php
$i=1;
```

```
while ($i<=5)
{
echo "The number is " . $i . "<br />";
$i++;
}
?>

</body>
</html>
```

DO..WHILE

Statemen do..while akan mengeksekusi suatu blok kode "minimal sekali" dalam perjalanan looping dan kemudian akan berulang "selama" suatu kondisi looping masih terpenuhi.

Sintaks :

```
do
{
kode yang di eksekusi;
}
while (kondisi);
```

Contoh :

Contoh berikut ini akan meng-increase nilai i minimal satu kali, dan kemudian akan lanjut increment nilai i selama nilai i masih lebih kecil dari 5.

```
<html>
<body>

<?php
$i=0;
do
{
$i++;
echo "The number is " . $i . "<br />";
}
while ($i<5);
?>

</body>
</html>
```

FOR

Statemen for digunakan ketika Anda mengetahui dengan pasti jumlah perulangan yang harus dilakukan pada blok kode Anda.

Sintaks :

```
for (inisialisasi; kondisi; increment)
{
code yang hendak dieksekusi;
}
```

Statemen for memiliki tiga parameter. Parameter pertama digunakan untuk inisialisasi variabel. Parameter kedua digunakan sebagai tempat menulis kondisi loop dan parameter ketiga berisi increment yang diperlukan dalam loop. Jika lebih dari satu variabel yang disertakan pada bagian inisialisasi atau pada bagian increment maka hal tersebut harus dipisahkan dengan tanda koma. Parameter kondisi harus mengevaluasi apakah kondisi terpenuhi atau tidak.

Contoh :

Contoh berikut adalah akan menampilkan string "Halo Surabaya !!" sebanyak 3 kali:

```
<html>
<body>

<?php
for ($i=1; $i<=3; $i++)
{
echo "Hello World!<br />";
}
?>

</body>
</html>
```

FOREACH

Statemen foreach digunakan untuk melakukan looping pada elemen-elemen pada suatu array.

Sintaks :

```
foreach (ekspresi_array as $value)
{
    kode yang hendak dieksekusi;
}
```

Contoh :

```
<html>
<body>
<?php
$arr = array("one", "two", "three");
reset ($arr);
while (list(, $value) = each ($arr)) {
    echo "Value: $value<br>\n";
}

foreach ($arr as $value) {
    echo "Value: $value<br>\n";
}
?>
</body>
</html>
```

Demikianlah pelajaran looping dasar pada PHP. Semoga Anda bisa memahami dan bisa segera mencobanya. Segala saran dan komentar langsung sampaikan ke penulis.