

Motivasi



Aspersepsi

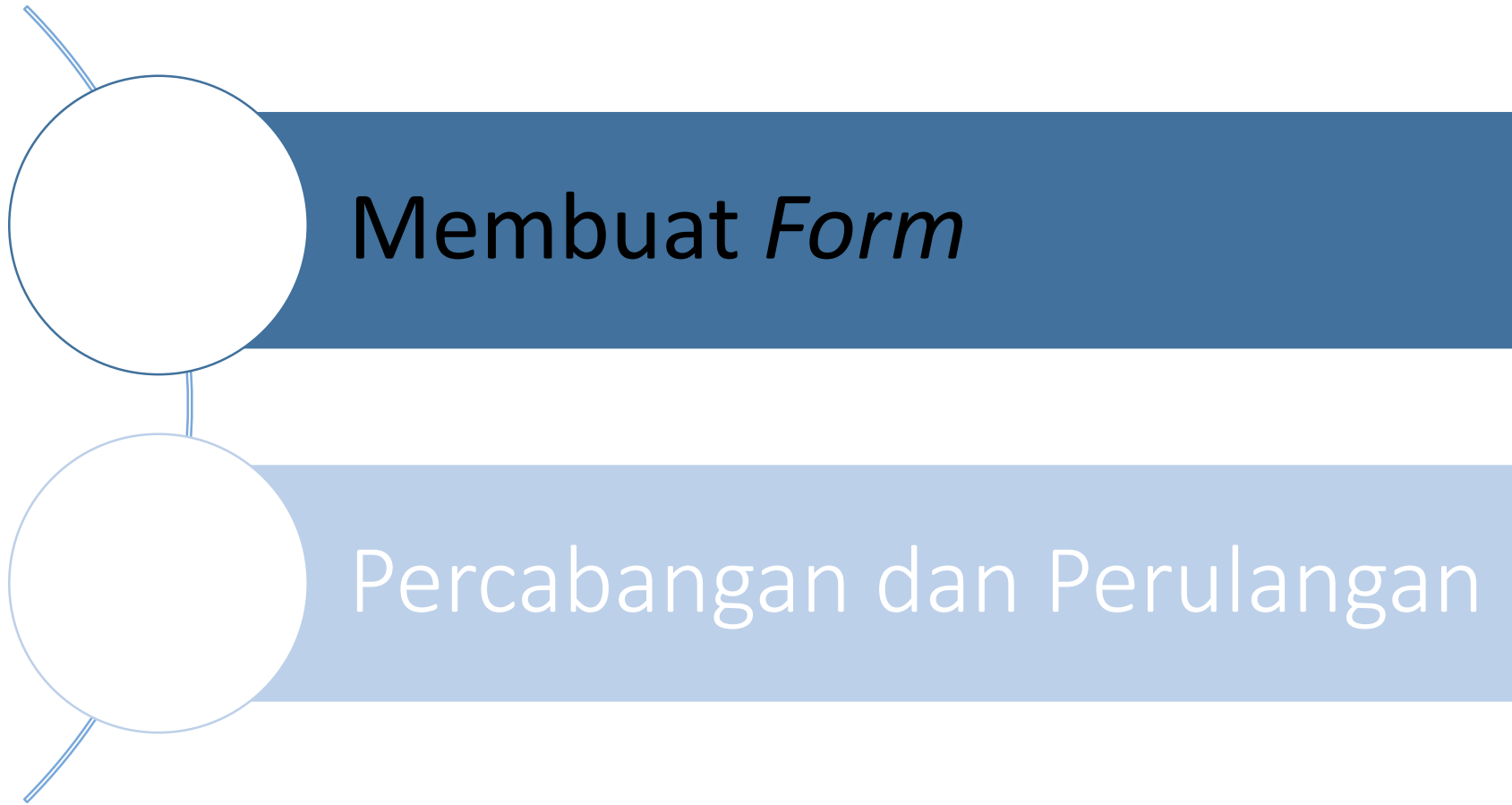


Membuat Program Sederhana

Ringkasan Mata Pelatihan

- Unit Kompetensi Acuan: Mengimplementasikan Pemrograman Terstruktur
- Kode Unit Kompetensi Acuan: J.620100.017.02
- Deskripsi singkat: Mata pelatihan ini menentukan kompetensi, pengetahuan dan sikap kerja yang diperlukan dalam membuat program terstruktur atau prosedural.
- Tujuan Pembelajaran: Peserta dapat membuat program terstruktur atau procedural sesuai dengan paradigma yang berlaku.

Agenda



Membuat Form

Mengenal Struktur Dasar Form HTML

```
<form action="proses.php" method="get">  
  Nama: <input type="text" name="nama" />  
  <br />  
  E-Mail: <input type="text" name="email" />  
  <br />  
  <input type="submit" value="Proses Data" />  
</form>
```

- Dari struktur dasar diatas tersebut, di dalam tag **<form>** terdapat 2 buah atribut. Yakni atribut **action** dan atribut **method**

Atribut Action

- Atribut pertama adalah **action**. Atribut **action** ini diisi dengan nilai berupa alamat halaman PHP dimana kita akan memproses isi form tersebut.
- Isi atribut action adalah alamat dari halaman PHP. Karena atribut *action* pada contoh diatas ditulis **action="proses.php"**, maka file **proses.php** harus berada di dalam 1 folder dengan halaman HTML yang berisi form ini. Namun alamat **proses.php** ini dapat dirubah tergantung dimana file tersebut berada, misalnya menjadi alamat relatif seperti **action="file_php/proses.php"**

Atribut Method

- Atribut kedua yang berkaitan dengan pemrosesan form HTML adalah atribut method. Atribut inilah yang akan menentukan bagaimana cara form 'dikirim' ke dalam halaman proses.php. Nilai dari atribut method hanya bisa diisi dengan 1 dari 2 pilihan, yakni get atau post.

Mengirimkan Nilai Form HTML ke dalam PHP

- Untuk mengambil nilai form HTML, PHP menyediakan 2 buah variabel global yaitu variabel **\$_GET** dan **\$_POST**. Kita menggunakan variabel **\$_GET** jika pada saat pembuatan form menggunakan atribut **method=get**, dan menggunakan variabel **\$_POST** jika form dibuat dengan **method=post**.
- Kedua variabel ini sebenarnya adalah array, sehingga cara mengakses nilai dari form adalah dengan cara: **\$_GET['nama_objek_form']**.
- **'nama_objek_form'** adalah nilai dari atribut **name** di dalam **form**. Jika kita memiliki tag dengan kode HTML **<input type="text" name="nama" />**, maka untuk mengakses nilainya adalah dengan **\$_GET['nama']**, dan untuk tag **<input type="text" name="email" />** diakses dengan nilai **\$_GET['email']**.

Contoh

Belajar.php

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Belajar Form PHP</title>
</head>
<body>
  <h2>Tutorial Belajar Form HTML - PHP </h2>
  <form action="proses.php" method="post">
    Nama: <input type="text" name="nama" />
    <br />
    E-Mail: <input type="text" name="email" />
    <br />
    <input type="submit" value="Proses Data" />
  </form>
</body>
</html>
```

Proses.php

```
<?php
  echo $_POST['nama'];
  echo "<br />";
  echo $_POST['email'];
?>
```

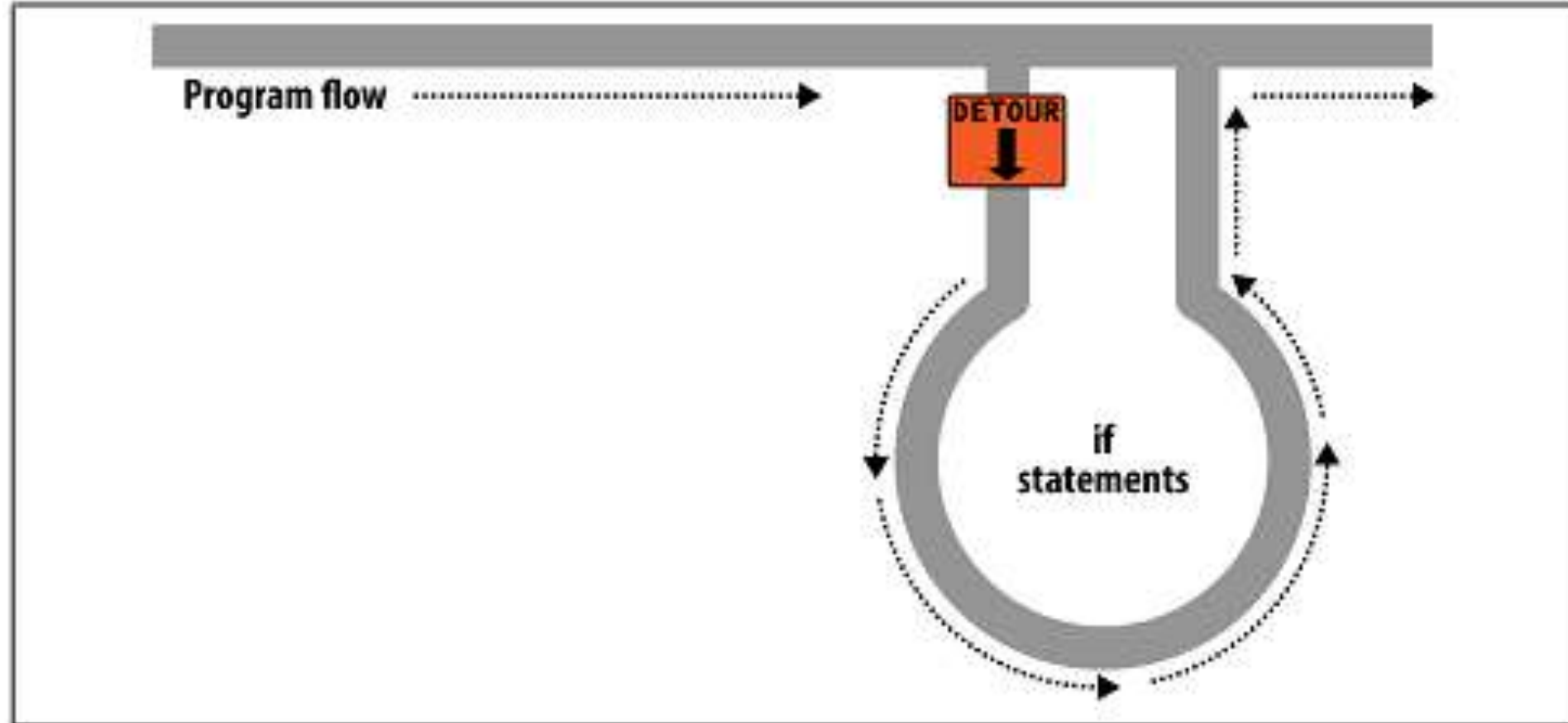
Percabangan dan Perulangan

Pencabangan

- Memungkinkan program melakukan aksi yang berbeda berdasarkan kondisi tertentu.
 - if
 - if...else
 - if...elseif...else
 - switch



Percabangan if



Percabangan if

- Menjalankan suatu perintah jika kondisi benar

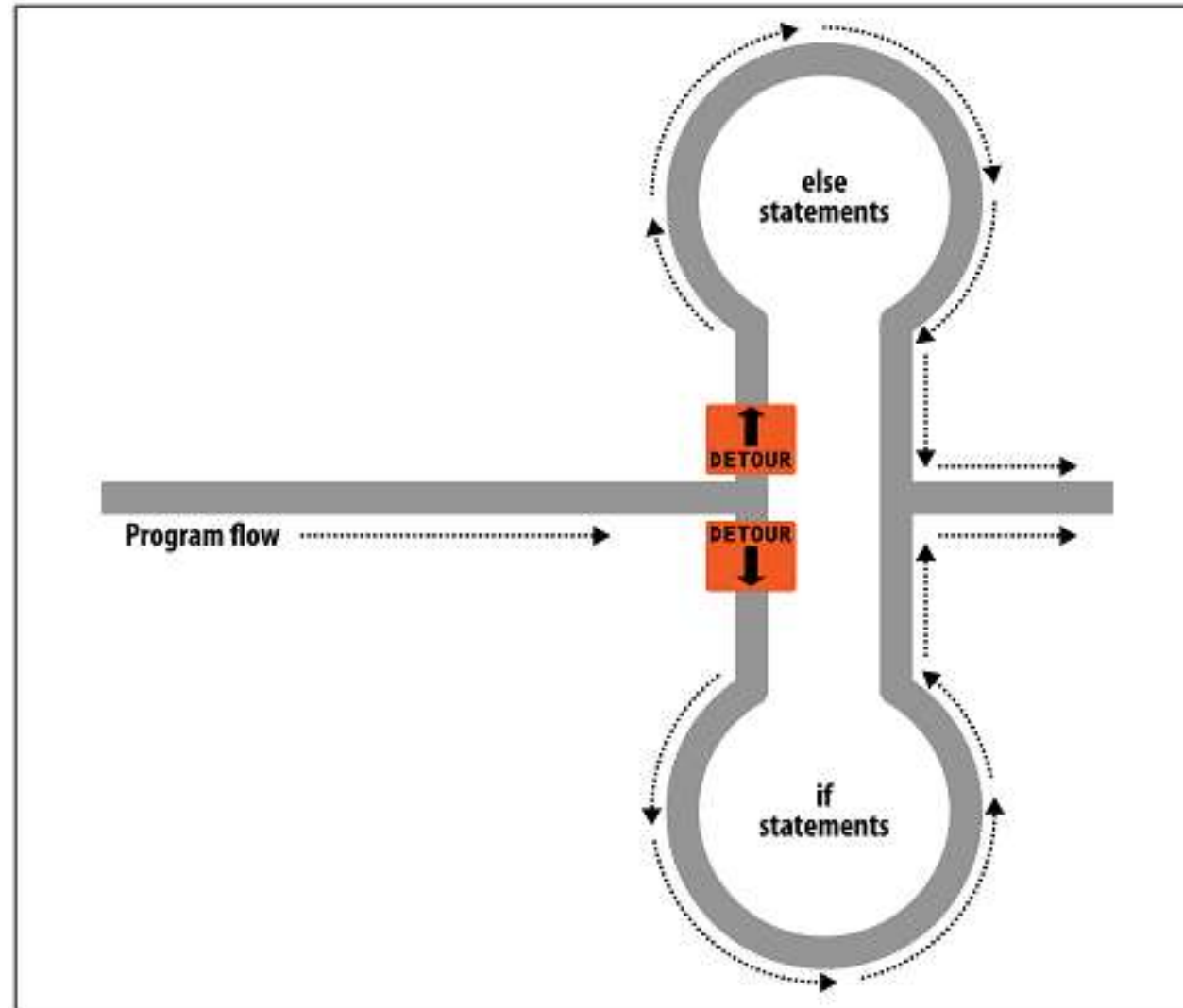
```
if (kondisi benar) {  
    //perintah yang dijalankan jika kondisi benar  
}
```

Contoh

```
echo "<br>";  
$angka = 3;  
if($angka >= 0){  
    echo "Bilangan Positif";  
}
```

Bilangan Positif

Percabangan If...else



Percabangan If...else

- Menjalankan suatu perintah jika kondisi benar dan perintah lain jika kondisi salah

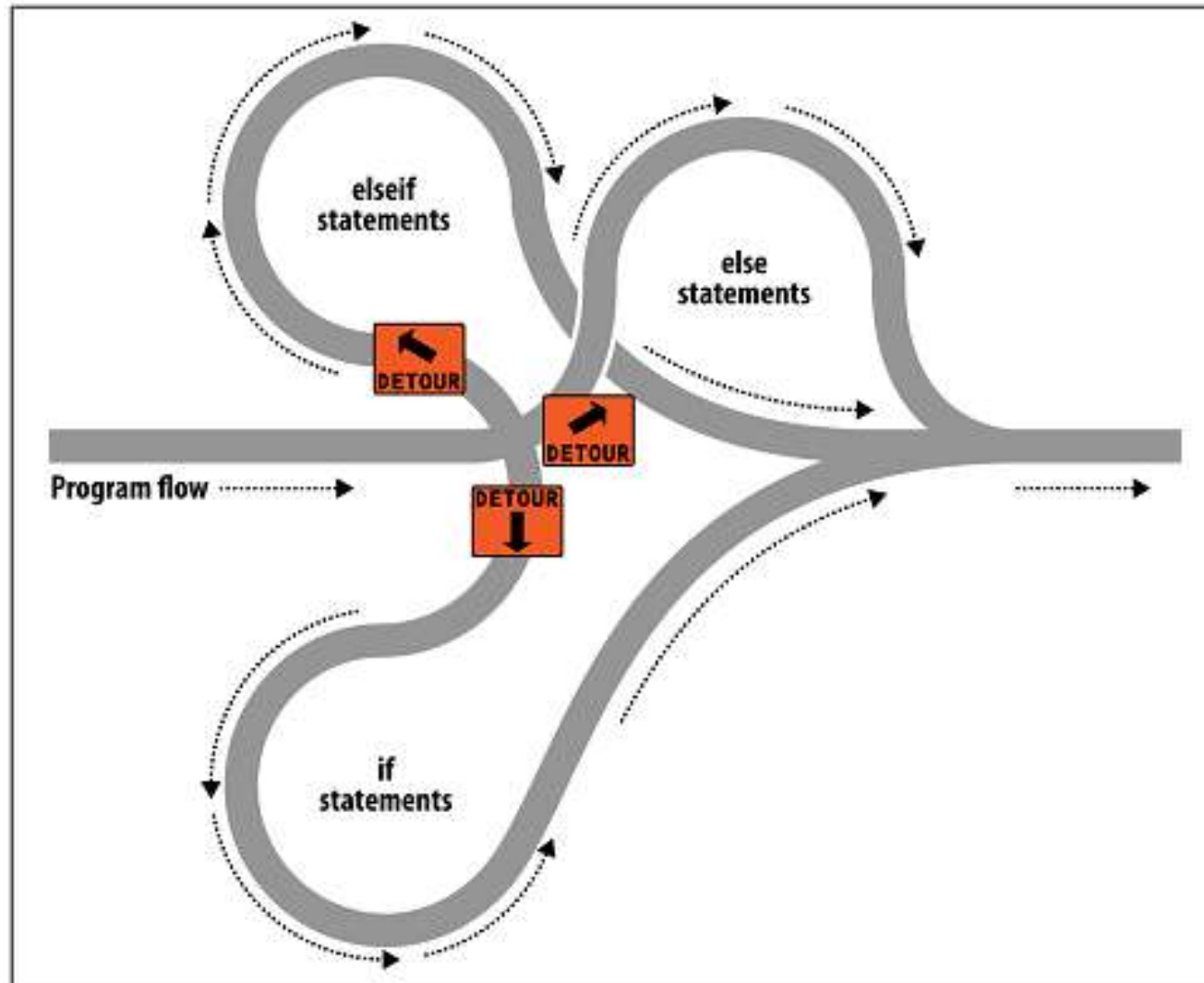
```
if (kondisi benar) {  
    //perintah yang dijalankan jika kondisi benar  
} else {  
    //perintah yang dijalankan jika kondisi salah  
}
```

Contoh

```
echo "<br>";  
$angka = -1;  
if($angka >= 0){  
    echo "Bilangan Positif";  
} else{  
    echo "Bilangan Negatif";  
}
```

Bilangan Negatif

Percabangan if...elseif...else



Percabangan if...elseif...else

- Menjalankan perintah yang berbeda untuk lebih dari dua kondisi

```
if (kondisi1 benar) {  
    //perintah yang dijalankan jika kondisi1 benar  
} elseif (kondisi2 benar) {  
    //perintah yang dijalankan jika kondisi2 benar  
} else {  
    //perintah yang dijalankan jika semua kondisi  
    salah  
}
```

Contoh

```
echo "<br>";  
$lampu = "hijau";  
if($lampu == "hijau"){  
    echo "jalan";  
} elseif ($lampu == "merah"){  
    echo "berhenti";  
} else{  
    echo "hati-hati";  
}
```

jalan

Video *Learning* Konsep Percabangan *if-else statement*

- Berikut merupakan *link* video mengenai konsep percabangan *if-else statement*

<https://youtu.be/OPU5Xei20U8>

Percabangan Switch/Case

- Menjalankan perintah yang berbeda berdasarkan pada kondisi yang berbeda
- Gunakan switch untuk memilih salah satu dari banyak kode yang akan dieksekusi

Percabangan Switch/Case

```
switch (n) {  
    case label1:  
        //perintah yang dijalankan jika n=label1  
        break;  
    case label2:  
        //perintah yang dijalankan jika n=label2  
        break;  
    case label3:  
        //perintah yang dijalankan jika n=label3  
        break;  
    ...  
    default:  
        //perintah yang dijalankan jika n berbeda dari semua label  
}
```

Contoh

```
$halaman = "berita";  
switch ($halaman) {  
case "home":  
    echo "Anda memilih home";  
    break;  
case "berita":  
    echo "Anda memilih berita";  
    break;  
case "artikel":  
    echo "Anda memilih artikel";  
    break;  
default:  
    echo "Halaman yang ada cari tidak tersedia";  
}
```

Perulangan

- Memungkinkan program mengulangi sederetan aksi berdasarkan kondisi tertentu.
- Ada dua jenis perulangan dalam pemrograman:
 - ✓ Counted loop;
 - ✓ Uncounted loop.



Lanj.

Pada PHP ada 4 jenis perulangan yang bisa kita gunakan:

- Perulangan For
- Perulangan While
- Perulangan Do/While
- Perulangan Foreach

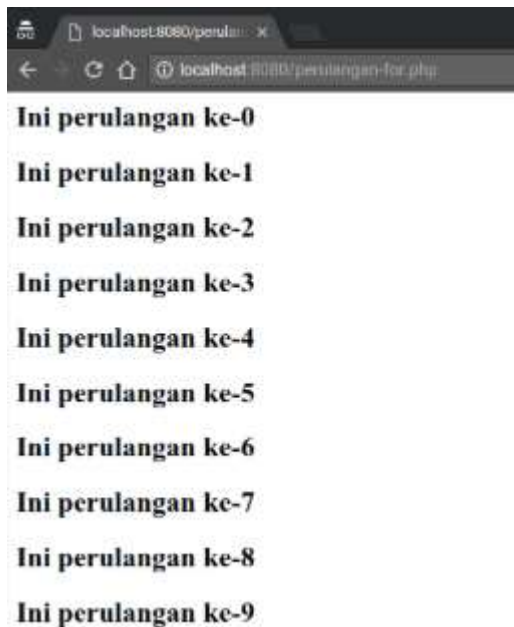
Perulangan For

- Perulangan *For* adalah perulangan yang termasuk dalam *counted loop*, karena kita bisa menentukan jumlah perulangannya.

```
for (init counter; test counter; increment counter) {  
    //perintah yang dijalankan  
}
```

Contoh

```
<?php  
  
for($i = 0; $i < 10; $i++){  
    echo "<h2>Ini perulangan ke-$i</h2>";  
}  
  
?>
```



- Variabel \$i dalam perulangan For berfungsi sebagai counter yang menghitung berapa kali ia akan mengulang.
- Hitungan akan dimulai dari nol (0), karena kita memberikan nilai \$i = 0.
- Lalu, perulangan akan diulang selama nilai \$i lebih kecil dari 10. Artinya, perulangan ini akan mengulang sebanyak 10x.
- Maksud dari \$i++ adalah nilai \$i akan ditambah 1 disetiap kali melakukan perulangan.

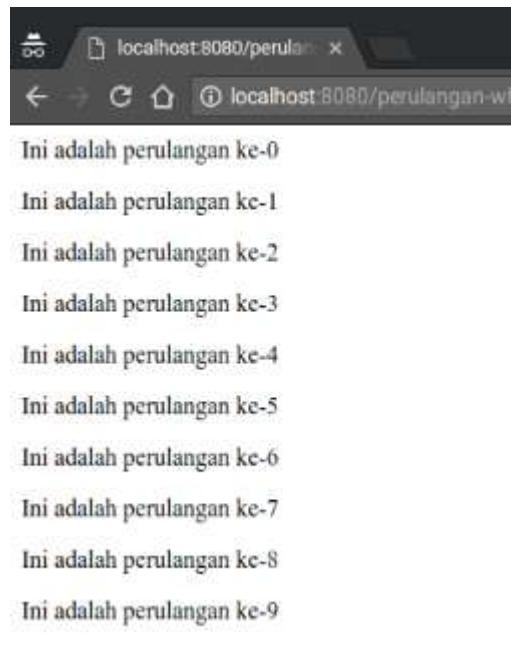
Perulangan While

- Perulangan *while* adalah perulangan yang termasuk dalam *uncounted loop*. Karena biasanya digunakan untuk mengulang sesuatu yang belum jelas jumlah pengulangannya.
- Namun, perulangan *while* juga bisa digunakan seperti perulangan *for* sebagai *counted loop*.

```
while (kondisi benar) {  
    //perintah yang dijalankan selama kondisi benar  
}
```

Contoh

```
<?php  
  
$ulangi = 0;  
  
while($ulangi < 10){  
    echo "<p>Ini adalah perulangan ke-$ulangi</p>";  
    $ulangi++;  
}  
  
?>
```



- Perulangan while akan terus mengulang selama nilai \$ulangi lebih kecil dari 10.
- Lalu di dalam perulangan kita melakukan increment nilai \$ulangi dengan \$ulangi++. Artinya: Tambah 1 disetiap pengulangan.
- Hati-hati, jangan sampai lupa menambahkan increment, atau kode yang akan mempengaruhi pengulangan. Karena kalau tidak, pengulangannya tidak akan pernah berhenti dan akan membuat komputer kita hang.

Video *Learning* Konsep *Looping* atau Perulangan (*For*) dalam Pemograman

- Berikut merupakan *link* video mengenai konsep *looping* atau perulangan (*For*) dalam pemograman

<https://youtu.be/u44gnktW4-U>

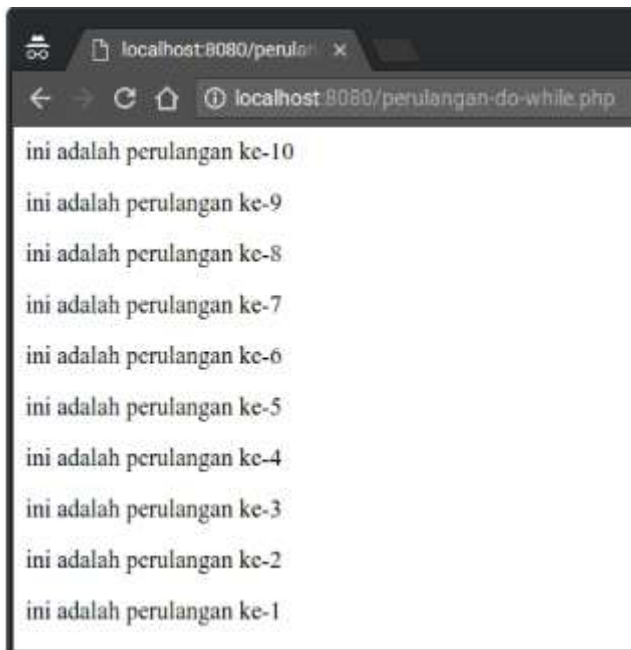
Perulangan Do/While

- Perulangan *Do/While* sama seperti perulangan *while*. Ia juga tergolong dalam *uncounted loop*.
- Perbedaan *Do/While* dengan *while* terletak pada cara iya memulai pengulangan.

```
do {  
    //perintah yang dijalankan  
} while (kondisi benar)
```

Contoh

```
<?php
$ulang = 10;
do {
    echo "<p>ini adalah perulangan ke-$ulang</p>";
    $ulang--;
} while ($ulang > 0);
?>
```



- Perulangan *Do/While* akan selalu melakukan pengulangan sebanyak 1 kali, kemudian melakukan pengecekan kondisi.
- Sedangkan perulangan *while* akan mengecek kondisi terlebih dahulu, baru melakukan pengulangan.

Perulangan Foreach

- Perulangan *foreach* sama seperti perulangan *for*. Namun, ia lebih khusus digunakan untuk mencetak array.

```
<?php  
foreach($array as $data) {  
    echo $data;  
}
```

Contoh

```
<?php
$books = [
    "Panduan Belajar PHP untuk Pemula",
    "Membangun Aplikasi Web dengan PHP",
    "Tutorial PHP dan MySQL",
    "Membuat Chat Bot dengan PHP"
];

echo "<h5>Judul Buku PHP:</h5>";
echo "<ul>";
foreach($books as $buku){
    echo "<li>$buku</li>";
}
echo "</ul>";

?>
```



Judul Buku PHP:

- Panduan Belajar PHP untuk Pemula
- Membangun Aplikasi Web dengan PHP
- Tutorial PHP dan MySQL
- Membuat Chat Bot dengan PHP

Kesimpulan

Kesimpulan

- *Form* mempunyai 2 buah atribut yaitu action dan method yang harus disesuaikan dengan kebutuhan
- Semakin kompleks percabangan atau perulangan maka akan semakin memperlambat proses loading time server.

Referensi / Bacaan Lebih Lanjut

Referensi / Bacaan Lebih Lanjut

- <https://www.duniailkom.com/tutorial-form-php-cara-menampilkan-hasil-form-html-dengan-php/>
- <https://www.w3schools.com/php/>



Kantor:

Balai Pelatihan dan Pengembangan
Teknologi Informasi dan Komunikasi
Kementerian Kominfo

Website: <https://bpptik.kominfo.go.id>

Email: bpptik@kominfo.go.id

Twitter: @bpptik

Facebook: @bpptik

Instagram: @bpptik

Google Plus: +bpptikkemkominfo

Terima Kasih

BPPTIK