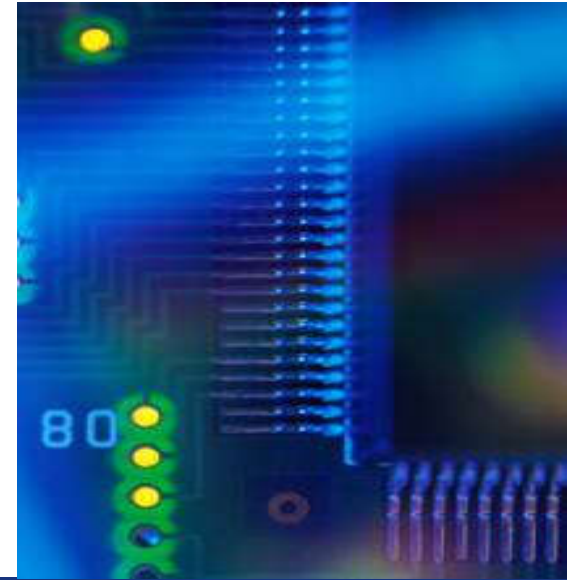




KEMENTERIAN KOMUNIKASI DAN INFORMATIKA  
REPUBLIK INDONESIA

*Menuju Masyarakat Informasi Indonesia*



**JUNIOR WEB DEVELOPER**

# **Mengimplementasikan Pemrograman Terstruktur**

# 1. Menggunakan Tipe Data dan Struktur Kontrol Program

Deskripsi Singkat mengenai Topik

Materi Pelatihan ini memfasilitasi pembentukan kompetensi dalam mengimplementasikan pemrograman terstruktur pada aplikasi berbasis web.

Tujuan Pelatihan

Peserta mampu menggunakan tipe data dan stuktur kontrol program pada aplikasi web yang akan dibangun.

Materi Yang akan disampaikan:

- 1.1. Tipe data
- 1.2. Syntax program
- 1.3. Struktur Kontrol Program

Tugas : *menggunakan tipe data pada aplikasi web yang dibuat & menggunakan struktur kontrol program pada aplikasi web yang dibuat*

Outcome/Capaian Pelatihan

Peserta Kompeten dalam menggunakan tipe data dan struktur kontrol program pada aplikasi berbasis web

## Variabel

## Pelatihan

### Variabel

- Suatu lokasi penyimpanan (di dalam memori) yang berisikan nilai atau informasi
- Kode program yang digunakan untuk menampung nilai tertentu
- Nilai dari variable dapat diisi dengan informasi yang diinginkan dan dapat dirubah nilainya pada saat kode program sedang berjalan.

## Variabel

## Pelatihan

### **Aturan :**

- Diawali dengan karakter dolar ( \$ ) dan diikuti dengan nama pengenalan
- Nama pengenalan dimulai dengan huruf atau garis bawah (\_), tidak boleh diawali dengan angka
- Tidak boleh mengandung spasi dan terdiri dari minimal satu karakter
- Bersifat case sensitive (huruf besar dan kecil dibedakan)
- Tidak diperlukan deklarasi type variable, tetapi type variable akan mengikuti type nilai yg diberikan
- Setiap variabel terbentuk dalam tipe data variant (dapat menampung jenis data apapun)

## Variabel

## Pelatihan

### Contoh penulisan Variabel

```
<?php
    $i;
    $nama;
    $Umur;
    $_lokasi_memori;
    $ANGKA_MAKSIMUM;
?>
```

### Contoh penulisan yang salah

```
<?php
    $4ever; //variabel tidak boleh diawali
dengan angka
    $_salah satu; //varibel tidak boleh
mengandung spasi
    $nama*^; //variabel tidak boleh mengandung
karakter khusus: * dan ^
?>
```

## Tipe Data

## Pelatihan

- **Variable merupakan ‘tempat’ dari data**
- **Data yang diinput kedalam variable memiliki tipe tertentu (angka, desimal, text)**
- **Tipe Data di PHP : integer, floating, string, Boolean, array dan object**

## Tipe Data Integer

## Pelatihan

- Tipe data berupa bilangan bulat
- Misalnya untuk menyimpan data jumlah stock, umur, tinggi badan, nomor sepatu
- Dapat bernilai positif (+) maupun negative (-)
- Dapat dilakukan operasi matematis (tambah, kurang, kali, bagi)

## Tipe Data Integer

## Pelatihan

- **Contoh**

```
<?php
    $umur=21;
    $harga=15000;
    $rugi=-500000;

    echo $umur; //21
    echo "<br />";
    echo $harga; //15000
    echo "<br />";
    echo $rugi; //-500000
```

?

- **Contoh dalam operasi matematis**

```
<?php
    $a=14;
    $b=16;
    $c= $a + $b;
    echo $c; // 30

    $d=$a * $b;
    echo $d; // 224

?>
```



## Tipe Data Float

## Pelatihan

- Tipe data floating point/real number berupa bilangan decimal (pecahan)
- Digunakan untuk variable yang akan berisi angka pecahan seperti nilai IPK, hasil pembagian, atau hasil komputasi numeric yang angkanya tidak bias ditampung oleh tipe data integer
- Contoh angka float: 0.9 , 2.80 , 3.14 , 0.314E1

## Tipe Data Float

## Pelatihan

- **Contoh**

```
<?php
    $angka_float1= 0.78;
    $angka_float2= 14.99;
    $angka_scientific1=0.314E1;
    $angka_scientific2=0.3365E-3;

    echo $angka_float1; // 0.78
    echo "<br />";
    echo $angka_float2; //14.99
    echo "<br />";
    echo $angka_scientific1; //3.14
    echo "<br />";
    echo $angka_scientific2; //0.0003365
?>
```

## Tipe Data Float

## Pelatihan

- **Contoh dalam operasi matematis**

```
<?php
```

```
    $a=10.66;
```

```
    $b=12.4;
```

```
    $c= $a + $b;
```

```
    echo $c; // 23.06
```

```
    $d=$a / $b;
```

```
    echo $d; // 0.85967741935484
```

```
?>
```

## Tipe Data String

## Pelatihan

- Tipe data yang berisi text, kalimat, atau kumpulan karakter
- Penulisannya diapit oleh single quoted/petik satu (') atau double quoted/petik ganda (")
- Contoh:
  - 'a' , 'saya sedang belajar PHP' , 'emailku19@gmail.com'
  - "a" , "saya sedang belajar PHP" , "emailku19@gmail.com"

## Tipe Data String

## Pelatihan

- **Contoh**

```
1
2
3      <?php
4      $string1='Ini adalah string sederhana';
5      $string2='Ini adalah string yang bisa memiliki beberapa baris';
6      $string3='Dia berkata: "I\'ll be back"';
7      $string4="Dia berkata: \"I'll be back\"";
8      $string5="Variabel akan otomatis ditampilkan: $string1 dan
9      $string3";
10
11      echo $string1; echo "<br>";
12      echo $string2; echo "<br>";
13      echo $string3; echo "<br>";
14      echo $string4; echo "<br>";
15      echo $string5; echo "<br>";
16      ?>
17
18
19
```



localhost/digitalent/pelatihan.php

```
echo $string1; echo "<br>" ini adalah string sederhana
echo $string2; echo "<br>" ini adalah string yang bisa memiliki beberapa baris
echo $string3; echo "<br>" dia berkata: "I'll be back"
echo $string4; echo "<br>" Dia berkata: "I'll be back"
echo $string5; echo "<br>" Variabel akan otomatis ditampilkan: ini adalah string sederhana dan dia berkata: "I'll be back"
```

## Tipe Data Boolean

## Pelatihan

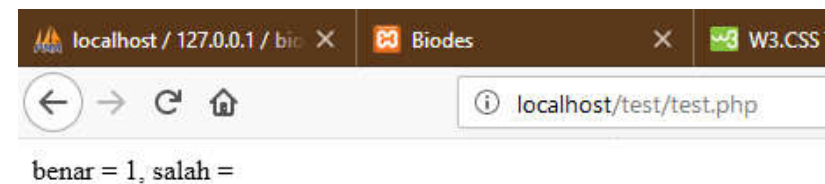
- hanya memiliki 2 nilai : true dan false
- Digunakan dalam operasi logika
- Contoh:

```
<?php
    $benar=true;
    $salah=false;

    echo "benar = $benar, salah = $salah";
    // hasil output: benar = 1, salah =

    $x = FALSE; // false
    $x = ""; // string kosong dianggap false
    $x = " "; // string dengan karakter adalah
true;
    $x = 0; // false
    $x = 1; // true
?>
```

Tampilan di web browser



## 1.2 Syntax program

## Pelatihan

### Syntax PHP : Aturan Penulisan kode program

.Semua skrip harus diapit oleh tanda:

- .- `<?php` dan `?>`
- .- `<script language="php">` dan `</script>`
- .- `<?>` dan `?>`
- .- `<%>` dan `%>`

.Perintah harus diakhiri dengan tanda titik koma ( ; )

Contoh:

```
<html>
  <head>
    <title>Program PHP Pertama Saya</title>
  </head>
  <body>
    <?php echo "Hello World!"; ?>
  </body>
</html>
```



## 1.3 Struktur kontrol program

## Pelatihan

Kontrol Program : bagaimana urutan eksekusi perintah dalam program.

Bentuk:

- Percabangan ( Branching )
- Perulangan ( Looping )
- Lompatan/Perpindahan ( Jumping )

## Percabangan (Branching)

- Disebut juga istilah struktur kontrol keputusan
- Memungkinkan aplikasi untuk memeriksa isi sebuah variabel atau hasil perhitungan dan ekspresi kemudian mengambil tindakan yang sesuai.
- Bentuk:
  - `if`
  - `if - else`
  - `if - elseif - else`
  - `switch`

## if Statement

- Sebuah struktur logika untuk membuat percabangan alur program
- Dapat mengatur apakah sebuah perintah akan dijalankan atau tidak tergantung dengan kondisinya

### Format if

```
if(ekspresi) {  
    statements  
    ...  
}
```

*ekspresi* : kondisi yang harus dipenuhi agar statements dapat dijalankan (bernilai true/false).

## if Statement

### Contoh if

```
<?php
    $usia = 30;
    if($usia > 29)
        echo "Maaf ... Anda tidak memenuhi Syarat untuk mengikuti Pelatihan ini!";
?>
```

Bagaimana outputnya jika baris ke-2 diganti dengan

```
$usia = 20;
```

## If – else Statement

- IF digunakan untuk percabangan alur program dengan 1 pilihan saja
- IF – ELSE dapat membuat percabangan dengan 2 alternatif
- Ketika kondisi IF tidak terpenuhi (ekspresi IF menghasilkan nilai FALSE) maka pernyataan pada bagian else yang akan dijalankan

## if - else Statement

### Format

```
if (ekspresi_logika) {  
    statements1  
    ...  
} else {  
    statements2  
    ...  
}
```

### Contoh:

```
<?php
```

```
$nilai = 80;  
if($nilai >= 60){  
    echo "Selamat anda lulus!";  
} else {  
    echo "Coba lagi semester depan.";  
}
```

```
?>
```

## If – elseif – else Statement

- Jika terdapat lebih dari dua alternatif maka menggunakan bentuk

`if - elseif - else`

- Format:

```
<?php
    if (ekspresiA) {
        statement1;
    } elseif (ekspresiB) {
        statement2;
    } else {
        statement3;
    }
?>
```

## Contoh if – elseif – else Statement

`<?php`

```
$nilai = 80;
if (($nilai >= 85) && ($nilai <= 100)) {
    $index = "A";
} elseif ($nilai >= 70) {
    $index = "B";
} elseif ($nilai >= 50) {
    $index = "C";
} elseif ($nilai >= 30) {
    $index = "D";
} else {
    $index = "E";
}
echo "Nilai anda adalah : ".$nilai;
echo "<br />";
echo "Dengan index      : ".$index;
?>
```



## switch Statement

- Bentuk alternatif dari `if-elseif-else`.
- Ideal untuk membuat keputusan berdasarkan sejumlah kondisi.
- Digunakan *hanya* untuk membandingkan variabel tunggal dengan dengan beberapa kemungkinan nilai-nilai.

## Format switch

```
switch($variable) {  
    case value_1:  
        statement 1;  
        break;  
    case value_2:  
        statement 2;  
        break;  
    case value_3:  
        statement 3;  
        break;  
    default:  
        statement 4;  
        break;  
}
```

## Contoh switch

```
<?php
$plat_nomor = "AB";
switch($plat_nomor){
    case "AB":
        echo "Yogyakarta";
        break;
    case "AD":
        echo "Surakarta";
        break;
    case "BE":
        echo "Lampung";
        break;
    case "B":
        echo "Jakarta";
        break;
    default:
        echo "Plat kendaraan tidak diketahui.";
        break;
}
?>
```

## Perulangan (Loop)

- Struktur Kontrol Perulangan digunakan untuk mengulang satu atau beberapa baris perintah

### •Bentuk:

- for
- while
- do-while
- foreach

## Perulangan (Loop)

- Terdapat 4 komponen :

- *inisialisasi* : kondisi awal perulangan, inisialisasi (pemberian nilai awal) pada variabel counter
- *condition* : kondisi yang harus dipenuhi agar perulangan dijalankan
- *increment* : bagian untuk memproses variable counter agar bisa memenuhi kondisi akhir perulangan
- *statement* : baris perintah yang akan diulang

## for Loop

- Disebut juga *determinate loop*.
- Perulangannya (iterasi) telah ditentukan sejak awal pernyataan
- Format :

```
for (inisialisasi; condition;
increment)
{
    statement;
}
```
- Contoh :

```
<?php
for ($i=1; $i<=10; $i++) {
    echo "Perulangan ke-".$i."<br />";
}
?>
```

## while Loop

- Dikenal juga dengan *indeterminate loop*.
- Penentuan jumlah perulangan tidak ditentukan sebelumnya.
- Perulangan akan dilakukan terus menerus sampai dengan kondisi yang menjadi prasyarat bernilai *false*.

## while Loop

### . Format:

```
inisialisasi;  
while (condition)  
{  
    statement;  
    ...  
    increment;  
}
```

### . Contoh:

```
<?php  
    $i=1;  
    while($i<=10) {  
        echo "Perulangan ke-".$i."<br  
/>";  
        $i++;  
    }  
?>
```



## do-while Loop

- Prinsip kerja sama dengan perulangan `while`.
- Perbedaannya : pengecekan kondisi perulangan dilakukan pada akhir perulangan
- Karena pengecekan kondisi dilakukan di akhir perulangan, maka walaupun kondisi adalah `FALSE`, perulangan akan tetap dijalankan minimal 1 kali.

## do-while Loop

### . Format:

```
inisialisasi;  
do  
{  
    statement;  
    ...  
    increment;  
} while(condition)
```

### . Contoh:

```
<?php  
    $i=1;  
    do{  
        echo "Perulangan ke-".$i."<br/>";  
        $i++;  
    } while($i<=10) ;  
?>
```

## foreach Loop

- . Perulangan khusus untuk array
- . Perulangan untuk mengakses elemen pada array tanpa mengidentifikasi index-nya

## foreach Loop

### . Menggunakan for:

```
<?php
$nama = array("Adi", "Budi", "Carli", "Dedy");
for($i=0; $i<4; $i++) {
    echo "$nama[$i]";
    echo "<br />";
}
?>
```

### . Menggunakan foreach:

```
<?php
$nama = array("Adi", "Budi", "Carli", "Dedy");
foreach($nama as $val) {
    echo "$val";
    echo "<br />";
}
?>
```

## Lompatan (Jumping)

Bentuk:

- Break
- Continue
- Exit

## Break

- Fungsinya untuk keluar dari sebuah perulangan.
- Contoh:

```
<?php
```

```
for ($i=1; $i<=6; $i++) {  
    if ($i == 2) {  
        break;  
    }  
    echo "Nilai i : ".$i."<br />";  
}  
?>
```

## Continue

- Digunakan untuk melewati satu siklus perulangan / iterasi.

- Contoh:

**<?php**

```
for ($i=1; $i<=6; $i++) {  
    if ($i == 3) {  
        continue;  
    }  
    echo "Nilai i : ".$i."<br />";  
}
```

**?>**

## Exit()

- Digunakan untuk keluar dari sebuah program. Walaupun terdapat statements lain dibawah baris kode `exit` yang belum dikerjakan.
- Contoh:

```
<?php
for($i=1;$i<=6;$i++) {
    if($i == 3) {
        exit();
    }
    echo "Nilai i : ".$i."<br />";
}
// statement dibawah ini tidak akan dijalankan
echo "Contoh Penggunaan Exit Pada Program";
?>
```



## 2. Baca Tulis Dari Keyboard.

- Untuk menerima input/masukan dari keyboard menggunakan `<input>` yang merupakan tag form html
- Atribut input:
  - Type (jenis input) : text
  - Name: nama elemen
  - value : nilai isian
- Contoh:

```
<html>
<head>
  <title>Input Text</title>
</head>
<body>
<form>
  Nama: <input type="text"
name="nama"/>
</form>
</body>
</html>
```

### 3.1 Program dengan menggunakan prosedur

- Sebuah blok program yang merupakan sekumpulan statement yang bertujuan untuk menyelesaikan suatu tugas tertentu
- Tujuannya : untuk kepraktisan dan kemudahan membuat program
- Setelah dijadikan prosedur/fungsi, maka untuk menjalankan tugas yang sama, kita tinggal memanggil fungsi tersebut, tanpa perlu membuat kembali kode programnya

## Format Prosedur/Fungsi

```
function nama_fungsi ($parameter1, $parameter2)
{
    // kode program fungsi
    return $nilai_akhir      // untuk fungsi yang mengembalikan
nilai
}
```

- *function* adalah *instruksi* kepada PHP bahwa kita akan membuat fungsi
- *nama\_fungsi* adalah nama dari fungsi yang akan ditulis
- *\$parameter1, \$parameter2* : variabel perantara yang akan menyimpan inputan yang diperlukan dalam pemrosesan fungsi (argumen).
- Jumlah parameter sesuai yang dibutuhkan.
- *return* : perintah khusus untuk fungsi, dimana kata *return* menginstruksikan kepada PHP bahwa pemrosesan fungsi telah selesai. *return \$nilai\_akhir* berarti bahwa fungsi akan '*mengembalikan*' *\$nilai\_akhir* sebagai hasil dari fungsi.

## Contoh prosedur

```
<?php
    function sayHello() {
        echo "Hello..<br />";
    }
    sayHello();
    sayHello();
    sayHello();
?>
```

## 3.2 Program dengan menggunakan fungsi

## Pelatihan

- **Fungsi** : prosedur yang memberikan (mengembalikan) nilai
- **Contoh**:

```
<?php
    function perkalian($bil_1, $bil_2) {
        $hasil = $bil_1 * $bil_2;
        return $hasil;
    }
    echo "5 x 10 = ".perkalian(5, 10);
?>
```

- **Passing Variable By Value**

Yaitu teknik memasukkan paramater ke dalam sebuah fungsi dengan cara membuat copy dari variabel asli, sehingga variabel asli tidak terpengaruh.

- **Passing Variable By Reference**

Memungkinkan kita untuk melakukan manipulasi terhadap variabel yang menjadi parameter melalui sebuah fungsi.

## Parameter Fungsi

## Pelatihan

### Passing Variable By Value

```
<?php

function
tambahSatu($value){
    $value = $value + 1;
}

$a = 5;
tambahSatu($a);
echo $a;

?>
```

Output: 5

### Passing Variable By Reference

```
<?php

function tambahSatu(&$value){
    $value = $value + 1;
}

$a = 5;
tambahSatu($a);
echo $a;

?>
```

Output: 6

Passing Variable By Reference Merubah nilai variabel yang menjadi parameter

## Referensi:

1. PHP 5 Tutorial, diakses dari laman <https://www.w3schools.com/php/>, pada 26 April 2019
2. Tutorial Belajar PHP Duniaikom, diakses dari laman <https://www.duniaikom.com/tutorial-belajar-php-dan-index-artikel-php/>, pada 27 April 2019
3. Anonymous. (n.d.). *MySQL Reference Manual*. Retrieved from <http://downloads.mysql.com/docs/>.



< Topik\_Silabus >

**Pelatihan**

# Terima Kasih