KOMDIGI
Kementerian Komunikasi dan Digital
Republik Indonesia

digitalent

VSGA Vocational School Graduate Academy

# PERTEMUAN 6-8

## PELATIHAN JUNIOR WEB DEVELOPER

# Mengimplementasikan Pemrograman Terstruktur

(Nama Lokasi/Saluran Daring, Tanggal Penyampaian)

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

- 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.

**Aturan :**

- Diawali dengan karakter dolar ( $ ) dan diikuti dengan nama pengenal
- Nama pengenal 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)

## Contoh penulisan Variabel

```
<?php
    $i;
    $nama;
    $Umur;
    $tinggi_cm;
    $Status_menikah;
?>
```

## 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 ^
?>
```

php Copy code

```php
<?php
  $nama = "John Doe";
  $umur = 25;
  $tinggi_cm = 175.5;
  $status_menikah = true;

  echo "Nama: " . $nama . "<br>";
  echo "Umur: " . $umur . " tahun<br>";
  echo "Tinggi: " . $tinggi_cm . " cm<br>";
  echo "Status Menikah: " . ($status_menikah ? "Sudah Menikah" : "Belum
?>
```

javascript Copy code

```javascript
<script>
  var nama = "John Doe";
  var umur = 25;
  var tinggiCm = 175.5;
  var statusMenikah = true;

  console.log("Nama: " + nama);
  console.log("Umur: " + umur + " tahun");
  console.log("Tinggi: " + tinggiCm + " cm");
  console.log("Status Menikah: " + (statusMenikah ? "Sudah Menikah" : "
</script>
```

Perbedaan penulisan variabel antara PHP dan JavaScript dapat dilihat pada penggunaan karakter underscore (_) pada JavaScript, sedangkan pada PHP karakter yang digunakan adalah dollar sign ($).

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

- 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)

```php
<?php
  $angka_php = 123; // Inisialisasi data integer di PHP
  echo "Angka (PHP): " . $angka_php;
?>
```

```javascript
<script>
  var angka_js = 456; // Inisialisasi data integer di JavaScript
  console.log("Angka (JavaScript): " + angka_js);
</script>
```

Dalam contoh di atas, variabel $angka_php diinisialisasi dengan nilai integer 123 dalam PHP, sedangkan variabel angka_js diinisialisasi dengan nilai integer 456 dalam JavaScript.

php Copy code

```php
<?php
  $angka1 = 10;
  $angka2 = 5;

  // Penjumlahan
  $hasilTambah = $angka1 + $angka2;
  echo "Penjumlahan (PHP): " . $hasilTambah . "<br>";

  // Perkalian
  $hasilKali = $angka1 * $angka2;
  echo "Perkalian (PHP): " . $hasilKali . "<br>";

  // Pembagian
  $hasilBagi = $angka1 / $angka2;
  echo "Pembagian (PHP): " . $hasilBagi . "<br>";
?>
```

javascript Copy code

```javascript
<script>
  var angka1 = 10;
  var angka2 = 5;

  // Penjumlahan
  var hasilTambah = angka1 + angka2;
  console.log("Penjumlahan (JavaScript): " + hasilTambah);

  // Perkalian
  var hasilKali = angka1 * angka2;
  console.log("Perkalian (JavaScript): " + hasilKali);

  // Pembagian
  var hasilBagi = angka1 / angka2;
  console.log("Pembagian (JavaScript): " + hasilBagi);
</script>
```

- **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
?>
```

php Copy code

```php
<?php
  $nilai_php = 123.45; // Inisialisasi float di PHP
  echo "Nilai (PHP): " . $nilai_php;
?>
```

javascript Copy code

```javascript
<script>
  var nilai_js = 678.90; // Inisialisasi float di JavaScript
  console.log("Nilai (JavaScript): " + nilai_js);
</script>
```

Dalam contoh di atas, variabel $nilai_php diinisialisasi dengan nilai float 123.45 dalam PHP, sedangkan variabel nilai_js diinisialisasi dengan nilai float 678.90 dalam JavaScript. Perhatikan bahwa tipe data float pada PHP dan JavaScript dapat menyimpan nilai desimal.

Dalam contoh di atas, variabel $nilai_php diinisialisasi dengan nilai float 123.45 dalam PHP, sedangkan variabel nilai_js diinisialisasi dengan nilai float 678.90 dalam JavaScript. Perhatikan bahwa tipe data float pada PHP dan JavaScript dapat menyimpan nilai desimal.

- 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

- **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
?>
```

- **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
?>
```

```php
<?php
  $teks_php = "Saya sedang belajar Junior Web Developer"; // Inisialisa
  echo "Teks (PHP): " . $teks_php;
?>
```

```javascript
<script>
  var teks_js = "Saya sedang belajar Junior Web Developer"; // Inisiali
  console.log("Teks (JavaScript): " + teks_js);
</script>
```

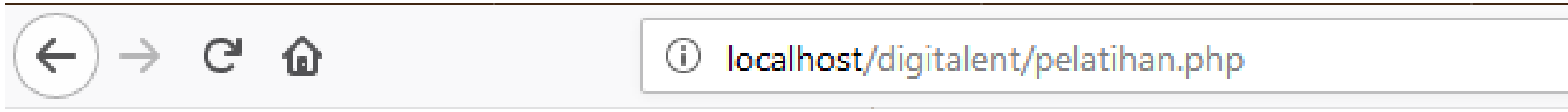
Dalam contoh di atas, variabel $teks_php diinisialisasi dengan nilai string "Saya sedang belajar Junior Web Developer" dalam PHP, sedangkan variabel teks_js diinisialisasi dengan nilai string yang sama dalam JavaScript. Variabel string dapat menyimpan teks atau karakter dalam kedua bahasa pemrograman tersebut.

- 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’ , ‘emailku@gmail.com’
- “a” , “saya sedang belajar PHP” , “emailku@gmail.com”

- **Contoh**

```
<?php
$string1='Ini adalah string sederhana';
$string2='Ini adalah string yang bisa memiliki beberapa baris‘;
$string3='Dia berkata: "I\'ll be back"';
$string4="Dia berkata: \"I'll be back\"“;
$string5="Variabel akan otomatis ditampilkan: $string1 dan
$string3";

echo $string1; echo "<br>";
echo $string2; echo "<br>";
echo $string3; echo "<br>";
echo $string4; echo "<br>";
echo $string5; echo "<br>";
?>
```




```
echo $string1; echo "<br>"
echo $string2; echo "<br>"
echo $string3; echo "<br>"
echo $string4; echo "<br>"
echo $string5; echo "<br>"
```

ini adalah string sederhana
ini adalah string yang bisa memiliki beberapa baris
dia berkata: "I'll be back"
Dia berkata: "I'll be back"
Variabel akan otomatis ditampilkan: ini adalah string sederhana dan dia berkata: "I'll be back"

php Copy code

```php
<?php
  $status_php = true; // Inisialisasi boolean di PHP
  echo "Status (PHP): " . ($status_php ? "True" : "False");
?>
```

javascript Copy code

```javascript
<script>
  var status_js = true; // Inisialisasi boolean di JavaScript
  console.log("Status (JavaScript): " + (status_js ? "True" : "False"))
</script>
```

Dalam contoh di atas, variabel $status_php diinisialisasi dengan nilai boolean true dalam PHP, sedangkan variabel status_js diinisialisasi dengan nilai boolean true dalam JavaScript. Variabel boolean dapat menyimpan nilai true atau false yang mewakili keadaan atau kondisi tertentu dalam program. Dalam contoh ini, kita menggunakan operator ternary untuk menampilkan nilai boolean dengan pesan "True" atau "False" dalam output.

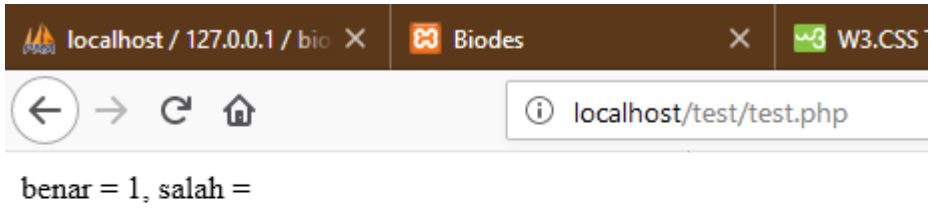
- **hanya memiliki 2 nilai : true dan false**
- **Digunakan dalam operasi logika**
- **Contoh:**

```
<?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



# 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>
```

Kontrol Program : bagaimana urutan eksekusi perintah dalam program.

Bentuk:

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

php Copy code

```php
<?php
  $nilai_php = 75;

  if ($nilai_php >= 70) {
    echo "Selamat, Anda lulus!";
  } else {
    echo "Maaf, Anda tidak lulus.";
  }
?>
```

javascript Copy code

```javascript
<script>
  var nilai_js = 75;

  if (nilai_js >= 70) {
    console.log("Selamat, Anda lulus!");
  } else {
    console.log("Maaf, Anda tidak lulus.");
  }
</script>
```

Dalam contoh di atas, percabangan digunakan untuk mengecek apakah nilai (nilai_php atau nilai_js) lebih besar atau sama dengan 70. Jika benar, maka program akan menampilkan pesan "Selamat, Anda lulus!", dan jika salah, program akan menampilkan pesan "Maaf, Anda tidak lulus.". Ini adalah contoh sederhana dari penggunaan struktur percabangan untuk mengambil keputusan berdasarkan suatu kondisi.

php Copy code

```php
<?php
  $nilai_php = 85;

  if ($nilai_php >= 90) {
    echo "Nilai PHP: A";
  } elseif ($nilai_php >= 80) {
    echo "Nilai PHP: B";
  } elseif ($nilai_php >= 70) {
    echo "Nilai PHP: C";
  } else {
    echo "Nilai PHP: D";
  }
?>
```

javascript Copy code

```javascript
<script>
  var nilai_js = 85;

  if (nilai_js >= 90) {
    console.log("Nilai JavaScript: A");
  } else if (nilai_js >= 80) {
    console.log("Nilai JavaScript: B");
  } else if (nilai_js >= 70) {
    console.log("Nilai JavaScript: C");
  } else {
    console.log("Nilai JavaScript: D");
  }
</script>
```

Dalam contoh di atas, kita menggunakan percabangan if-else if-else untuk mengecek rentang nilai. Jika nilai memenuhi kondisi pertama, maka pesan "A" akan ditampilkan. Jika tidak, program akan melanjutkan ke kondisi berikutnya, dan seterusnya. Jika tidak ada kondisi yang memenuhi, maka blok else akan dieksekusi.

- Disebut juga istilah stuktur 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`

- 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).

# 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` 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

## Format

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

## Contoh:

```
<?php

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

    }
?>
```

- Jika terdapat lebih dari dua alternatif maka menggunakan bentuk

  `if - elseif - else`

- Format:

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

```php
<?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;
?>
```

```php
<?php
// Menetapkan nilai
$nilai_php = 85;

// Menggunakan percabangan switch untuk menentukan nilai
switch (true) {
    case ($nilai_php >= 90):
        echo "Nilai PHP: A";
        break;
    case ($nilai_php >= 80):
        echo "Nilai PHP: B";
        break;
    case ($nilai_php >= 70):
        echo "Nilai PHP: C";
        break;
    default:
        echo "Nilai PHP: D";
}
?>
```

```javascript
// Menetapkan nilai
var nilai_php = 85;
var grade;

// Menggunakan percabangan switch untuk menentukan nilai
switch (true) {
    case (nilai_php >= 90):
        grade = "A";
        break;
    case (nilai_php >= 80):
        grade = "B";
        break;
    case (nilai_php >= 70):
        grade = "C";
        break;
    default:
        grade = "D";
}

console.log("Nilai PHP:", grade);
```

- 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.

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

```php
<?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;
}
?>
```

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

- Bentuk:
  - `for`
  - `while`
  - `do-while`
  - `foreach`

- 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

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

- Format :

```
for (inisialisasi; condition; increment)
{
    statement;
}
```

- Contoh :

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

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

- Format:

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

- Contoh:

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

- 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.

- Format:

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

- Contoh:

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

- **Menggunakan for:**

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

- **Menggunakan foreach:**

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

Bentuk:

- `Break`
- `Continue`
- `Exit`

- Fungsinya untuk keluar dari sebuah perulangan.
- Contoh:

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

- Digunakan untuk melewati satu siklus perulangan / iterasi.
- Contoh:

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

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

```php
<?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”;
?>
```

- Untuk menerima input/masukan dari keyboar 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>
```

- 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

```
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 didibutuhkan.
- *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.

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

php Copy code

```php
<?php
  // Mendefinisikan fungsi sebagai prosedur di PHP
  function sapaan($nama) {
    echo "Halo, " . $nama . "!";
  }

  // Memanggil fungsi sebagai prosedur
  sapaan("John");
?>
```

javascript Copy code

```javascript
<script>
  // Mendefinisikan prosedur di JavaScript
  function sapaan(nama) {
    console.log("Halo, " + nama + "!");
  }

  // Memanggil prosedur
  sapaan("John");
</script>
```

Dalam kedua contoh di atas, kita mendefinisikan fungsi di PHP dan prosedur di JavaScript yang menampilkan pesan sapaan. Perhatikan bahwa di PHP, kita dapat menggunakan echo dalam fungsi untuk menampilkan output langsung ke layar tanpa mengembalikan nilai. Di JavaScript, kita menggunakan console.log untuk mencetak pesan ke konsol.

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

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

php Copy code

```php
<?php
  // Mendefinisikan fungsi di PHP
  function sapaan($nama) {
    return "Halo, " . $nama . "!";
  }

  // Memanggil fungsi
  $hasil_php = sapaan("John");
  echo $hasil_php;
?>
```

javascript Copy code

```javascript
<script>
  // Mendefinisikan fungsi di JavaScript
  function sapaan(nama) {
    return "Halo, " + nama + "!";
  }

  // Memanggil fungsi
  var hasil_js = sapaan("John");
  console.log(hasil_js);
</script>
```

Dalam contoh di atas, kita mendefinisikan sebuah fungsi bernama sapaan yang menerima satu parameter ($nama di PHP dan nama di JavaScript) dan mengembalikan pesan sapaan. Kemudian, kita memanggil fungsi tersebut dengan memberikan argumen ("John") dan menampilkan hasilnya

Buatlah program menggunakan **Procedure** dan **Fuction** untuk menghitung diskon **hitungDiskon()** yang menerima parameter total belanja. Prosesnya adalah sebagai berikut:

1. Jika total belanja lebih dari atau sama dengan Rp. 100.000, maka diskon sebesar 10%.
2. Jika total belanja lebih dari atau sama dengan Rp. 50.000, tapi kurang dari Rp. 100.000, maka diskon sebesar 5%.
3. Jika total belanja kurang dari Rp. 50.000, maka tidak ada diskon.

Setelah itu, prosedur mengembalikan nilai diskon yang kemudian digunakan untuk menghitung total yang harus dibayar.
Pada bagian akhir kode, kita memanggil prosedur tersebut dengan memberikan contoh total belanja sebesar Rp. 120.000, dan menampilkan total belanja, diskon, dan total yang harus dibayar setelah diberikan diskon.

```
// Membuat prosedur untuk menghitung diskon
function hitungDiskon($totalBelanja) {
    $diskon = 0;
    if ($totalBelanja >= 100000) {
        $diskon = 0.1 * $totalBelanja; // Diskon 10% untuk total belanja di
atas atau sama dengan Rp. 100.000
    } elseif ($totalBelanja >= 50000) {
        $diskon = 0.05 * $totalBelanja; // Diskon 5% untuk total belanja di
atas atau sama dengan Rp. 50.000
    }
    return $diskon;
}

// Memanggil prosedur untuk menghitung diskon
$totalBelanja = 120000; // Misalnya total belanja Rp. 120.000
$diskon = hitungDiskon($totalBelanja);

echo "Total belanja: Rp. " . $totalBelanja . "<br>";
echo "Diskon: Rp. " . $diskon . "<br>";
echo "Total yang harus dibayar: Rp. " . ($totalBelanja - $diskon) . "<br>";
```

```php
<?php
// Fungsi untuk menghitung diskon berdasarkan total belanja
function hitungDiskon($totalBelanja) {
    $diskon = 0;

    if ($totalBelanja >= 100000) {
        $diskon = 0.1 * $totalBelanja; //Diskon 10% untuk total belanja di atas
atau sama dengan Rp. 100.000
    } elseif ($totalBelanja >= 50000) {
        $diskon = 0.05 * $totalBelanja; //Diskon 5% untuk total belanja di atas
atau sama dengan Rp. 50.000
    }

    return $diskon;
}
// Contoh penggunaan fungsi untuk menghitung diskon
$totalBelanja = 120000; // Total belanja sebesar Rp. 120.000
$diskon = hitungDiskon($totalBelanja); //Memanggil fungsi untuk menghitung diskon

echo "Total belanja: Rp. " . $totalBelanja . "<br>";
echo "Diskon: Rp. " . $diskon . "<br>";
echo "Total yang harus dibayar: Rp. " . ($totalBelanja - $diskon) . "<br>";
?>
```

- 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.

## 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 Duniailkom, diakses dari laman https://www.duniailkom.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/.

KOMDIGI
Kementerian Komunikasi dan Digital
Republik Indonesia

digitalent

# Terima Kasih

