

MODUL PRAKTIKUM
PABW MINGGU-8
Materi : Fungsi

FUNGSI

1. Penjelasan

Fungsi adalah suatu blok program yang mempunyai tujuan tertentu. Fungsi dibuat agar suatu pekerjaan dapat dilakukan berulang-ulang tanpa menulis baris kode yang sama.

1. Struktur Fungsi

```
function functionName (parameter) {  
    ...  
    statement  
    ...  
}
```

The diagram illustrates the structure of a PHP function. It shows the keyword **function**, followed by the function name *functionName* (labeled 'Nama Fungsi'), then the parameter list *(parameter)* (labeled 'Parameter'), and finally the function body enclosed in curly braces { } (labeled 'Function Body'). The function body contains an ellipsis (...) and a *statement*.

Struktur fungsi diatas dapat dijelaskan sebagai berikut:

a. Nama fungsi merupakan deklarasi nama fungsi yang akan kita buat.

Deklarasi fungsi harus memenuhi syarat-syarat sebagai berikut:

- Tidak boleh sama dengan fungsi yang sudah ada didalam PHP. Ini dikarenakan PHP tidak mendukung fasilitas *overloading* yaitu kondisi dimana fungsi dapat menggunakan nama yang sama namun memiliki keluaran yang berbeda dengan syarat parameter pada fungsi tersebut berbeda
- Hanya terdiri dari huruf, angka dan garis bawah (underscore).
- Tidak boleh diawali dengan angka.

b. Parameter adalah data / nilai masukan yang akan diolah oleh fungsi.

c. Function body adalah statements / perintah yang akan dikerjakan oleh fungsi.

2. Implementasi Fungsi

```
1  <?php
2  function multiply($num_1,$num_2){
3      $result = $num_1 * $num_2;
4      return $result;
5  }
6
7  echo '3 x 3 = '.multiply(3,3);
8  ?>
```

Gambar 1 Implementasi dan pemanggilan fungsi

Penjelasan dari fungsi diatas adalah sebagai berikut :

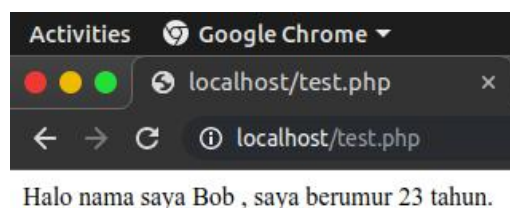
1. Deklarasi sebuah fungsi dengan nama **multiply** dan memiliki parameter input **\$num_1** dan **\$num_2**
2. Fungsi **multiply** berfungsi melakukan operasi perkalian antara **\$num_1** dan **\$num_2** lalu menyimpannya pada variable **\$result**.
3. Pemanggilan fungsi **multiply** akan mengembalikan variabel **\$result**.

Contoh : Pemanggilan fungsi **multiply(3,3)** akan mengembalikan nilai **9**.

Note: Fungsi bisa tanpa mempunyai nilai kembalian dan parameter (*void*)

```
1  <?php
2  function sayHi($name,$age){
3      $message = "Halo nama saya ".$name.", saya berumur ".$age." tahun.";
4      echo $message;
5  }
6
7  sayHi('Bob',23)
8  ?>
```

Gambar 2 Fungsi tanpa nilai kembalian



3. Scope Variabel

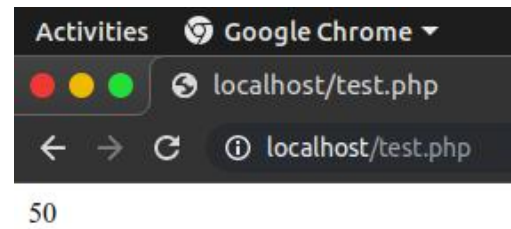
A. Local Scope

Setiap fungsi memiliki ruang lingkup kerja tersendiri, sehingga jika variabel didalam fungsi memiliki nama yang sama dengan variabel diluar fungsi tidak akan terjadi gangguan. Perhatikan contoh berikut :

```

1 <?php
2 $nilai_asli = 50;
3
4 function palsukan_nilai($nilai){
5     $nilai_asli = $nilai;
6 }
7
8 palsukan_nilai(95);
9 echo $nilai_asli;
10 ?>
11

```



Variabel *\$nilai_asli* yang berada di luar fungsi *palsukan_nilai(\$nilai)* tidak terpengaruh oleh proses yang terjadi didalam fungsi *palsukan_nilai(\$nilai)* meski menggunakan nama variabel yang sama. Variabel *\$nilai_asli* yang berada didalam fungsi hanya bisa diakses dan dioperasikan didalam fungsi *palsukan_nilai(\$nilai)*.

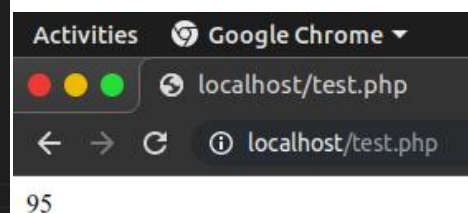
B. Global Scope

Bagaimanapun *PHP* juga menyediakan mekanisme agar variabel yang dideklarasikan diluar *function body* dapat dikenali dan dioperasikan / diubah nilainya melalui *function body*.

```

1 <?php
2 $nilai_asli = 50;
3
4 function palsukan_nilai($nilai){
5     global $nilai_asli ;
6     $nilai_asli = $nilai;
7 }
8
9 palsukan_nilai(95);
10 echo $nilai_asli;
11 ?>
12

```



Variabel *\$nilai_asli* pada program diatas akan bernilai 95 setelah pemanggilan fungsi *palsukan_nilai()*. Dengan kata lain, variabel *\$nilai_asli* terkena pengaruh operasi yang dilakukan *palsukan_nilai()*. Ini dikarenakan adanya statement *global \$nilai_asli;* yang dijalankan pada *function body* *palsukan_nilai()*. Statement tersebut memberitahukan kepada *PHP* interpreter bahwa variabel *\$nilai_asli* pada *function body* tersebut adalah sebuah variabel global. Dengan demikian variabel *\$nilai_asli* yang sudah dideklarasikan sebelumnya akan dikenali dan dapat diubah nilainya melalui fungsi tersebut.

4. Passing Parameter

A. By Value

Passing by value ialah teknik memasukkan parameter ke dalam sebuah fungsi dengan cara membuat salinan dari variabel asli, sehingga variabel asli tidak terpengaruh. Untuk lebih jelasnya, perhatikan contoh latihan berikut :



The screenshot shows a code editor on the left with the following PHP code:

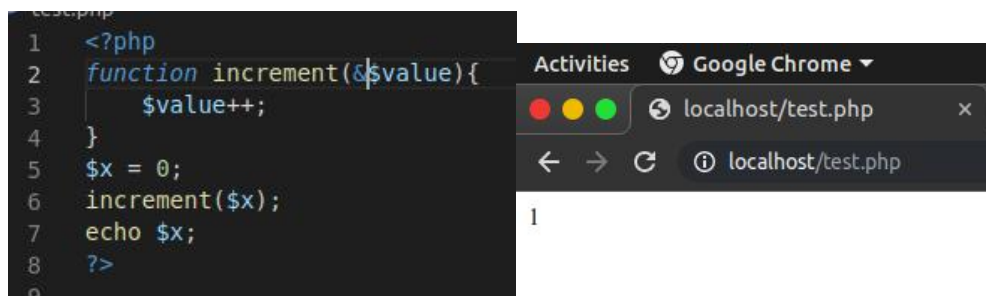
```
1 <?php
2 function increment($value){
3     $value++;
4 }
5 $x = 0;
6 increment($x);
7 echo $x;
8 ?>
```

On the right, a Google Chrome browser window is open to `localhost/test.php`. The output displayed on the page is `0`.

Awalnya variabel `$x` memiliki nilai 0, ketika menjalankan fungsi `increment`, maka nilai dari variabel `$x` akan disalin ke variabel `$value`. Maka operasi yang dilakukan oleh fungsi `increment` adalah operasi yang dilakukan terhadap variabel `$value` yang merupakan salinan dari variabel `$x`. Sehingga nilai variabel `$x` tidak berubah.

B. By Reference

Passing parameter by reference memungkinkan kita untuk melakukan manipulasi terhadap variabel yang menjadi parameter melalui sebuah fungsi. Untuk lebih jelasnya, perhatikan contoh latihan berikut yang merupakan modifikasi dari skrip program sebelumnya :



The screenshot shows a code editor on the left with the following PHP code:

```
1 <?php
2 function increment(&$value){
3     $value++;
4 }
5 $x = 0;
6 increment($x);
7 echo $x;
8 ?>
```

On the right, a Google Chrome browser window is open to `localhost/test.php`. The output displayed on the page is `1`.

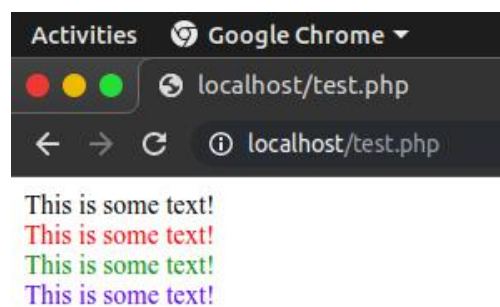
Pada passing parameter by reference, variabel `$x` menjadi variabel referensi bagi variabel `$value`. Sehingga setiap perubahan yang terjadi pada variabel `$value` akan berimbas pada variabel `$x`. Untuk melakukan passing parameter by reference, cukup menambahkan tanda “&” pada awal parameter.

C. Default Parameter

Default parameter adalah nilai yang otomatis digunakan jika saat pemanggilan fungsi parameter tersebut tidak berisi nilai.

Contoh :

```
1 <?php
2 function bar($color="black"){
3     echo '<font color="'. $color. '>This is some text!</font><br>';
4 }
5
6 bar();
7 bar('red');
8 bar('green');
9 bar('blue');
10 ?>
11
```



Secara default, teks tersebut akan berwarna hitam jika fungsi bar dipanggil tanpa ada parameter. Namun jika pemanggilan fungsi disertai dengan parameter \$color, maka parameter default akan diabaikan dan menggunakan parameter yang dimasukkan.

5. Fungsi built-in PHP

PHP menyediakan fungsi built-in yang melimpah. Lebih dari 700 buah fungsi built-in yang tersedia pada PHP 5.x yang akan memudahkan kita untuk mengembangkan sebuah sistem.

A. include & require

Fungsi include/require memungkinkan kita untuk memasukkan/menyertakan isi sebuah file kedalam sebuah program. Perbedaan antara keduanya yaitu ketika include gagal memanggil sebuah file, PHP akan menampilkan pesan error, namun tetap menjalankan program utamanya. Berbeda dengan require yang akan langsung berhenti jika terdapat masalah pada saat pemanggilan sebuah file.

header.php

```

1 <?php
2 function printHeader(){
3     echo "Ini Header <br>";
4 }
5
6 printHeader();
7 ?>

```

footer.php

```

1 <?php
2 function printFooter(){
3     echo "<br>Ini Footer <br>";
4 }
5
6 printFooter();
7 ?>

```

Main.php

```

1 <?php
2 include('header.php');
3
4 echo "ini main";
5
6 require('footer.php');
7 ?>
8

```

Activities Google Chrome ▾

localhost/main.php

← → ↻ ⓘ localhost/main.php

Ini Header
ini main
Ini Footer

B. strlen

Fungsi strlen berguna untuk mengetahui panjang / banyaknya karakter dalam sebuah string / variabel.

```

1 <?php
2
3 $alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
4
5 echo "Jumlah alfabet : ".strlen($alphabet);
6 ?>

```

Activities Google Chrome ▾

localhost/main.php

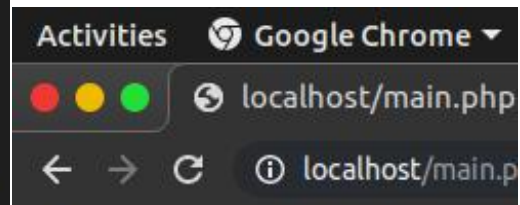
← → ↻ ⓘ localhost/main.php

Jumlah alfabet : 26

C. isset

Fungsi `isset` digunakan memeriksa apakah sebuah variabel sudah dideklarasikan sebelumnya. Ia akan mengembalikan nilai `true` jika sebuah variabel sudah dideklarasikan dan `false` jika belum pernah dideklarasikan.

```
main.php
1  <?php
2
3  if(isset($alphabet)) {
4      echo "True";
5  }else{
6      echo "False";
7  }
8
9  echo "<br>";
10 $alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
11
12 if(isset($alphabet)) {
13     echo "True";
14 }else{
15     echo "False";
16 }
17 <?>
```



False
True

Latihan 1

Buatlah sebuah fungsi untuk menukar (swap) nilai 2 variabel dengan menggunakan *passing by reference*!

Latihan 2

Buatlah fungsi untuk menghitung harga bet berdasarkan panjang nama dan menampilkannya dengan warna font yang diinginkan. Jika parameter warna kosong, maka gunakan warna merah.

Panjang nama 1 - 10 : Rp. 300/karakter

Panjang nama 11-20 : Rp. 500/karakter

Panjang nama >20 : Rp. 700/karakter

Ouput dari fungsi adalah nama dan harga total dengan warna font yang diinginkan.

Latihan 3

Buatlah fungsi untuk menghitung faktorial dari suatu bilangan!

Tugas Mingguan

Buatlah sebuah form yang memiliki keterangan berikut :

- Nama (Textfield) - tidak boleh kosong
- Alamat (Textfield)
- Jenis Kelamin (Radio Button)
- Golongan Darah (Combo Box)
- Hobby (Checkbox)
- Keterangan (TextArea)

Kemudian keterangan diatas akan diproses oleh fungsi pencetakan data.

Simpan dalam file TM_7form.php dan TM_7hasil.php