

MODUL PRAKTIKUM
PEMROGRAMAN WEB



Materi 6:
Function

Dosen :

Febrianta Surya Nugraha, M.Kom

S1 INFORMATIKA
STMIK AMIKOM SURAKARTA
2020

FUNCTION

1. PENDAHULUAN FUNCTION

- **Fungsi (Function)** adalah bagian kode program PHP yang berfungsi untuk melaksanakan tugas-tugas spesifik tertentu.
- Fungsi digunakan supaya code program bisa lebih mudah dibaca, dipahami, dipelihara, dan **digunakan ulang untuk keperluan lain** (selama prosesnya sama).
- Fungsi berdasarkan siapa pembuatnya dapat dibagi menjadi 2, yaitu **Built-in function** dan **User Defined Function**.
- **Built-in function** merupakan fungsi bawaan PHP, yang berjumlah banyak. Contohnya array, sort, dll. Untuk mengetahui fungsi-fungsi bawaan dari PHP dapat dilihat di tautan: <https://www.php.net/manual/en/indexes.functions.php>.
- **User Defined Function** merupakan fungsi yang dideklarasikan oleh user.

Syntax penulisan Fungsi :

```
function NamaFungsi ([parameter, ..]) {  
    // kode yang akan dijalankan  
    ..  
}
```

Penulisan **nama fungsi** harus mengikuti ketentuan sebagai berikut:

1. **Harus diawali huruf atau underscore (_) kemudian dapat diikuti dengan huruf, angka, dan underscore.**
2. **Case in-sensitive** (tidak membedakan huruf kecil dan besar).

Parameter fungsi ditulis dalam tanda kurung dan dapat berupa tipe data apapun baik string, array, object, boolean, dsb., selain itu argumen juga **dapat dikosongkan**.

Kita dapat **mendefinisikan nilai default dari parameter**, sehingga memudahkan pemanggilan fungsi karena tidak perlu menulis argumen terlalu banyak.

Jenis fungsi ada 2, yaitu:

1. **Fungsi yang tidak mengembalikan nilai (void function):** Fungsi yang tidak mengembalikan nilai adalah fungsi yang hanya melakukan proses tertentu, tidak menghasilkan nilai.

```
<?php  
function cetak_judul() {  
    echo "<h2>Pemrograman Web </h2>";  
}  
?>
```

2. **Fungsi yang mengembalikan nilai:** Fungsi yang mengembalikan nilai memiliki perintah *return* untuk mengembalikan nilai yang ditampung dalam sebuah variabel.

```
<?php
function pangkat_dua($x) {
    $hasil=$x*$x;
    return $hasil;
}
?>
```

```
<?php
function gabung_string($string1,
$string2){
    $hasil = $string1 . $string2;
    return $hasil;
}
?>
```

Memanggil Fungsi

Cara memanggil fungsi:

- Untuk fungsi yang tidak mengembalikan nilai, cukup tuliskan nama fungsinya dan daftar parameternya (bila ada).

```
cetak_judul(); // memanggil fungsi cetak_judul()
```

- Untuk fungsi yang mengembalikan nilai, sediakan variabel bantu untuk menampung nilai kembalian fungsi atau dapat dengan langsung memproses hasil dari fungsi tersebut.

```
// menggunakan variabel bantu untuk penampung
$gabung = gabung_string ("Bimo","Aulia");
echo $gabung . "<br>";
// nilai kembalian fungsi langsung diproses
echo gabung_string ("Bimo","Aulia");
```

Ketika melakukan pemanggilan fungsi, maka penulisan parameter harus lengkap, jika fungsi terdiri dari 3 parameter, maka kita harus menuliskan ketiganya,

Latihan Fungsi

```
<html>
<body>
<?php
// Tanpa parameter
function cetak_judul() {
    echo "<h2>Pemrograman Web</h2>";
}
cetak_judul(); //memanggil fungsi cetak judul

echo "<br>";

//Dengan parameter
function gabung_string($string1, $string2){
    $hasil = $string1 . ' ' . $string2;
    return $hasil;
}
// menggunakan variabel bantu untuk penampung
$gabung = gabung_string ("Saya","Belajar");
echo $gabung . "<br>";
// nilai kembalian fungsi langsung diproses
echo gabung_string ("Fungsi","di PHP");

echo "<br><br>";

// menambahkan nilai default
function nama_bulan($bulan, $tahun = 2020) {
    echo $bulan . ' ' . $tahun;
}
nama_bulan('mei');
?>
</body>
</html>
```

MELEWATKAN PARAMETER KE DALAM FUNGSI

Parameter adalah suatu nilai yang dilewatkan (masuk dan keluar) ke dalam fungsi. Kegunaannya adalah untuk membuat fungsi tersebut menjadi dinamis (nilainya berbeda sesuai dengan nilai parameter yang dilewatkan).

Cara melewatkan Parameter ke dalam Fungsi pada PHP antara lain:

1. Pass by value (Melewatkan Parameter berdasarkan nilai).

Secara default, fungsi pada php menggunakan pendekatan ini, yaitu **proses di dalam fungsi tidak mempengaruhi keadaan di luar fungsi**. Berikut adalah contoh kode fungsi untuk menunjukkan hal tersebut.

```

<html>
<body>
<?php
function tambahsatu($bilangan) {
    $hasil=$bilangan++;
    return $hasil;
}
$x=4;
echo "nilai \$x sebelum pemanggilan fungsi : $x" . "<br>";
echo " hasil fungsi:" . tambahsatu($x) . "<br>";
echo "nilai \$x setelah pemanggilan fungsi : $x <br>";
?>
</body>
</html>

```

Output:

```

nilai $x sebelum pemanggilan fungsi : 4
hasil fungsi:4
nilai $x setelah pemanggilan fungsi : 4

```

2. Pass by Reference (Melewatkan Parameter berdasarkan referensi).

Dalam **pass by reference**, argumen harus berupa **variabel**. Pada saat pendefinisian fungsinya, membubuhkan tanda **ampersand (&)** di **depan parameter yang bersangkutan**. Berbeda dengan pass by value, parameter yang dilewatkan berdasarkan referensi ini akan mempengaruhi keadaan di luar fungsi

```

<html>
<body>
<?php
function tambahsatu(&$bilangan) {
    $hasil=$bilangan++;
    return $hasil;
}
$x=4;
echo "nilai \$x sebelum pemanggilan fungsi : $x" . "<br>";
echo " hasil fungsi:" . tambahsatu($x) . "<br>";
echo "nilai \$x setelah pemanggilan fungsi : $x <br>";
?>
</body>
</html>

```

Output:

```

nilai $x sebelum pemanggilan fungsi : 4
hasil fungsi:4
nilai $x setelah pemanggilan fungsi : 5

```

MEMANGIL FUNGSI DI DALAM FUNGSI

```
<?php
// membuat fungsi
function hitungUmur($lahir, $thn){
    $umur = $thn - $lahir;
    return $umur;
}
function perkenalan($nama){
    echo "Perkenalkan, nama saya ".$nama."<br/>";
    // memanggil fungsi lain
    echo "Saya berusia ". hitungUmur(1999, 2020) ." tahun<br/>";
    // ganti tahun lahir
}
// memanggil fungsi perkenalan
perkenalan("Isi nama disini");
?>
```

FUNGSI REKURSIF

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Fungsi ini biasanya digunakan untuk menyelesaikan masalah seperti faktorial, bilangan fibonacci, pemrograman dinamis, dll.

```
<?php
function faktorial($angka) {
    if ($angka < 2) {
        return 1;
    } else {
        // memanggil dirinya sendiri
        return ($angka * faktorial($angka-1));
    }
}
$x=8;
// memanggil fungsi
echo "faktorial $x adalah " . faktorial($x);
?>
```

PHP RETURN TYPE DECLARATIONS

Pada PHP 7 mensupport return tipe deklarasi. Untuk mendeklarasikan tipe return, gunakan tanda titik dua (:) dan tipe data sebelum tanda kurung kurawal ({ } pada fungsi.

```
<?php
function penjumlahan(float $a, float $b) : float {
    return $a + $b;
}
echo penjumlahan(1.2, 5.2);
?>
```

Coba ganti tipe float sebelum kurawal yang bukan tipe data parameter menjadi **int** (Interger).

LATIHAN

Ubahlah latihan percabangan if else aplikasi sederhana menentukan nilai akhir dengan menggunakan **fungsi**.

```
<!DOCTYPE html>
<html>
<head>
<title>Nilai Akhir elseif</title>
</head>
<body>
<form method="POST" action="namafile.php">
Masukkan Nilai Tugas = <input type="text" name="tugas" placeholder="0-100"><br/>
Masukkan Nilai UTS = <input type="text" name="UTS" placeholder="0-100"><br/>
Masukkan Nilai UAS = <input type="text" name="UAS" placeholder="0-100"><br/>
<input type="submit" name="oke" value="kirim">
</form>
<?php
if(isset($_POST['oke']))
{
    $tugas = $_POST['tugas'];
    $UTS = $_POST['UTS'];
    $UAS = $_POST['UAS'];
    $nilai=(0.3* $tugas)+ (0.2*$UTS)+ (0.5*$UAS);
    echo "Nilai Tugas Anda adalah $tugas <br>" ;
    echo "Nilai UTS Anda adalah $UTS <br>" ;
    echo "Nilai UAS Anda adalah $UAS <br>" . "<br>" ;
    if($nilai <1) {
        echo "Nilai Anda adalah $nilai, = E";
    }
    elseif ($nilai >=1 && $nilai <=55) {
        echo "Nilai Anda adalah $nilai, = D";
    }
    elseif ($nilai >=56 && $nilai <=75) {
        echo "Nilai Anda adalah $nilai, = C";
    }
    elseif ($nilai >=76 && $nilai <=85) {
        echo "Nilai Anda adalah $nilai, = B";
    }
    elseif ($nilai >=86 && $nilai <=100) {
        echo "Nilai Anda adalah $nilai, = A";
    }
    else {
        echo " Coba periksa Input data";
    }
}
?>
</body>
</html>
```