



# APLIKASI TEKNOLOGI ONLINE

## FUNCTION

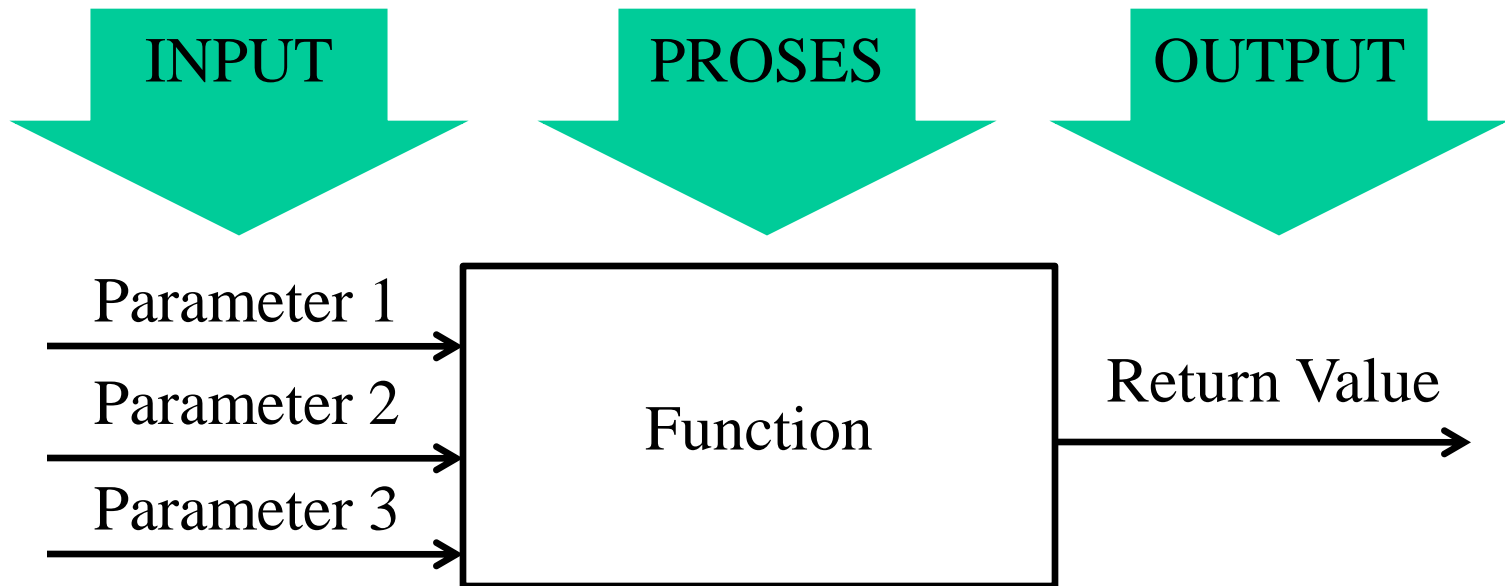
# Functions

- Pendeklarasian Function dan Pemanggilan Function
- Nilai Kembalian (Return Value)
- Variable Function
- Membuat Library Function
- Built-in Function (Function internal dari PHP)

## Definisi Function

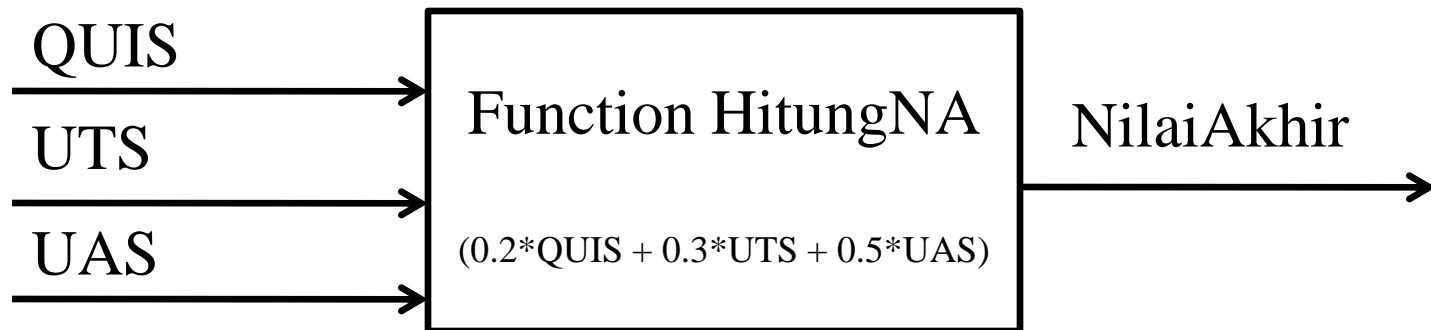
- PHP mendukung modularity program.
- Umumnya modularity suatu program diwujudkan dalam bentuk suatu function
- Suatu fungsi dapat melakukan suatu pengeksekusian sekumpulan kode yang mempunyai kegunaan/fungsi tertentu.
- Suatu function bekerja berdasarkan parameter-parameter yang diinputkan kepadanya.

# Function



Parameter sering juga disebut sebagai Argument

# Function



## Jenis-Jenis Function

Ada 2 jenis function

- Function yang mereturnkan nilai

Function jenis ini biasanya memiliki perintah return di dalam badan functionnya.

- Function yang tidak mereturnkan nilai

Function jenis ini hanya melakukan perintah-perintah yang ada di dalam badan function, tanpa harus memiliki return value. Function jenis ini juga biasa disebut procedure.

## Pembuatan Function

- Pendeklarasian function dapat dilakukan di lokasi manapun dalam file php
- Sebuah pendefinisian function umumnya terdiri dari 3 bagian yaitu :
  - Nama function
  - Daftar parameter yang dipisahkan dengan tanda koma, dan diapit oleh tanda kurung.
  - Isi function yang diapit oleh tanda kurung kurawal ( { } )

# Pembuatan Function

Bentuk umum function adalah

```
function function_name ($param1, $param2, ..., $paramN)
{
    Isi function; //Kode program ditulis disini.
    Isi function;
    Isi function;
    return nilai; // Sifatnya opsional.
}
```



## Pembuatan Function

Aturan-aturan dalam pembuatan function adalah :

- Aturan penamaan function mirip dengan penamaan variable. Terdiri dari huruf, angka dan underscore ( \_ ). Nama function hanya boleh dimulai dengan huruf atau dengan underscore.
- Parameter sifatnya tambahan. Boleh saja function tanpa memiliki parameter.
- Function boleh memiliki nilai kembalian (return value) ataupun tidak.
- Variabel yang dideklarasikan dalam function hanya berlaku dalam function saja

## Pembuatan Function

```
function info_tanggal()←Tanpa Parameter
{
    $hari=array("Minggu","Senin","Selasa",
                "Rabu","Kamis","Jum'at","Sabtu");
    $bulan=array("", "Januari", "Februari", "Maret", "April",
                 "Mei", "Juni", "Juli", "Agustus",
                 "September", "Oktober", "November", "Desember");
    $tgl=getdate();
    echo $hari[$tgl['wday']].", $tgl['mday']-".
        $bulan[$tgl['mon']]."-$tgl['year']";
}
```

```
function pangkat($x, $n)←Dengan Parameter
{
    return pow($x,$n);
}
```

## Memanggil Function

- Pemanggilan function dilakukan cukup dengan memanggil nama functionnya, lengkap dengan paramater yang dibutuhkan (jika ada).
- Jika function yang dipanggil adalah function yang mereturnkan value, maka nilai returnnya dapat dipanggil dalam echo atau diisikan ke suatu variable untuk diolah kemudian.
- Contoh :
  - `info_tanggal()` ; // langsung mencetak tanggal sekarang
  - `echo pangkat(5,2)` ; // menampilkan hasil pangkat  $5^2$
  - `$hasil = pangkat(5,3)` ; // return value disimpan ke variable

## Memanggil Function

Namafile : function1.php

```
<html>
<head><title>Test Fungsi</title></head>
<body>
<?php
    function info_tanggal()
    {
        ....
    }
    function pangkat($x, $n)
    {
        ....
    }
?>
Sekarang : <?php info_tanggal();?> <br>

5<sup>3</sup> = <?php echo pangkat(5,3);?>
</body>
</html>
```

# Memanggil Function

Namafile : function1.php



## Memanggil Function

```
<html>
<head>
<title>Function 1</title>
</head>
<body>
<?php

    function ip_client(){
        return $_SERVER['REMOTE_ADDR'];
    }
    function tulis_identitas($nama, $kota){
        echo "Nama saya $nama dari $kota";
    }

    tulis_identitas("Andri Heryandi","Sweetwater");
    echo "<br>";
    $ip_pengunjung=ip_client(); // simpan nilai kembalian ke variabel
    echo "Anda memiliki IP : $ip_pengunjung"; // tampilkan isi variable

?>
</body>
</html>
```

Namafile : function2.php

# Memanggil Function

Namafile : function2.php



## Variable Function

- Variable function adalah sebuah pemanggilan dinamis terhadap suatu fungsi yang namanya ditentukan pada waktu eksekusi.
- Meskipun tidak terlalu banyak digunakan dalam kebanyakan aplikasi web, variable function dapat mengurangi ukuran kode dan kerumitan kode program, bahkan dapat menghilangkan kondisi statement yang tidak diperlukan.



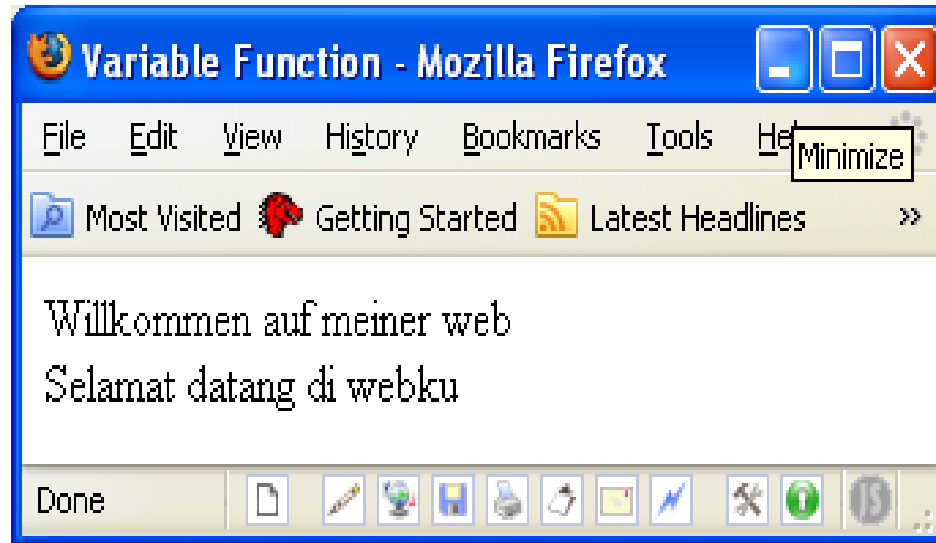
## Variable Function

Namafile : function3.php

```
<html>
<head>
<title>Variable Function</title>
</head>
<body>
<?php
    function indonesia(){ echo "Selamat datang di webku<br>"; }
    function inggris()  { echo "Welcome to my web<br>"; }
    function jerman()   { echo "Willkommen auf meiner web<br>"; }
    function italia()   { echo "Benvenuti nel Mio Web<br>"; }
    $bahasa="jerman";
    $bahasa(); // Akan memanggil function jerman();
    $bahasa="indonesia";
    $bahasa(); // akan memanggil function indonesia();
?>
</body>
</html>
```

## Variable Function

Namafile : function3.php



## Membuat Library Function

- Library Function adalah sekumpulan function yang disimpan dalam file tertentu yang nantinya akan dipanggil oleh file lain.
- Library function akan menghindarkan programmer membuat function yang sama berulang kali dalam file yang berbeda.

## Membuat Library Function

<?php

```
$email_admin="admin@heryandi.net";
$nama_situs="Situs Heryandi.net";
function info_tanggal()
{
    $hari=array("Minggu","Senin","Selasa","Rabu","Kamis","Jum'at","Sabtu");
    $bulan=array("", "Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli",
                "Agustus", "September", "Oktober", "November", "Desember");
    $tgl=getdate();
    echo $hari[$tgl['wday']].", $tgl[mday] - ".$bulan[$tgl['mon']]."-$tgl[year]";
}
function pangkat($x, $n)
{
    return pow($x,$n);
}
function ip_client(){
    return $_SERVER['REMOTE_ADDR'];
}
function tulis_identitas($nama, $kota){
    echo "Nama saya $nama dari $kota";
}
}
```

Namafile : my\_func.php

Function-function dikumpulkan dalam  
sebuah file

?>

## Memanggil Library Function

Memanggil library function dapat dilakukan dengan cara :

1. Memanggil function include atau include\_once
2. Memanggil function require atau require\_once

Include dan require mempunyai kegunaan yang sama.

Include atau require akan menampilkan pesan error jika anda memanggil include/require terhadap file yang sama lebih dari 1 kali. Untuk menghindari kesalahan tersebut, gunakanlah include\_once atau require\_once.

## Memanggil Library Function

Namafile : call\_func.php

```
<html>
<head>
<title>Menggunakan Function Library</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<?php
    include_once("my_func.php"); // memanggil isi library my_func.php

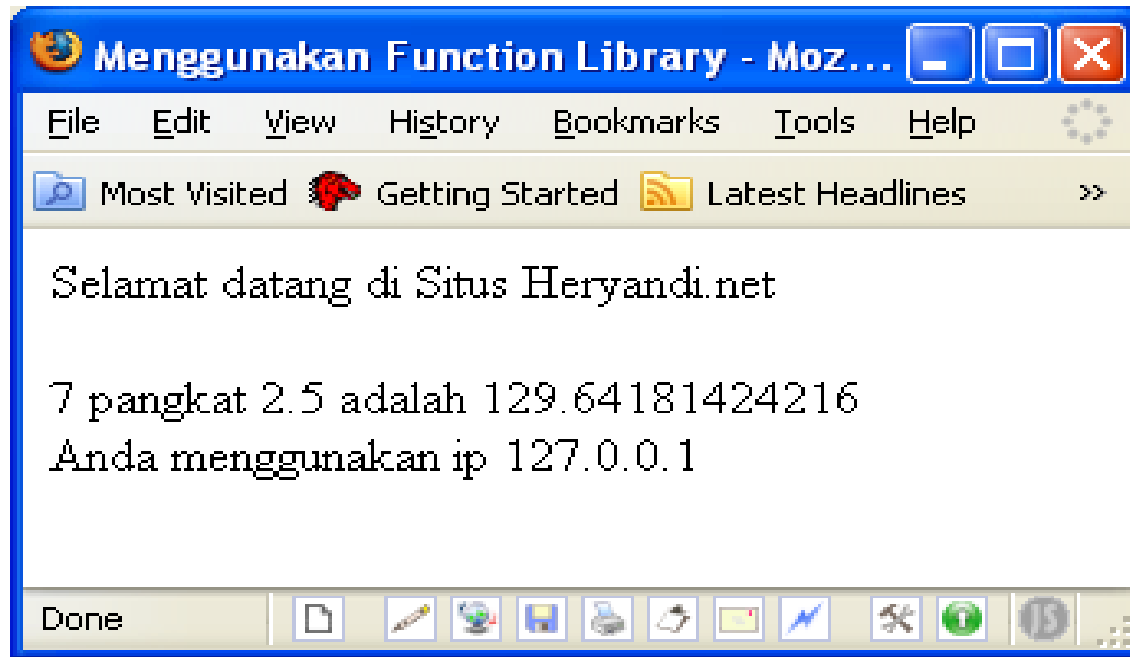
    // Memanggil variable yang dideklarasikan dalam library function
    echo "Selamat datang di $nama_situs<br><br>";

    // Memanggil function yang dideklarasikan dalam library function
    echo "7 pangkat 2.5 adalah ".pangkat(7,2.5)."<br>";
    echo "Anda menggunakan ip ".ip_client();

?>
</body>
</html>
```

# Memanggil Library Function

Namafile : call\_func.php

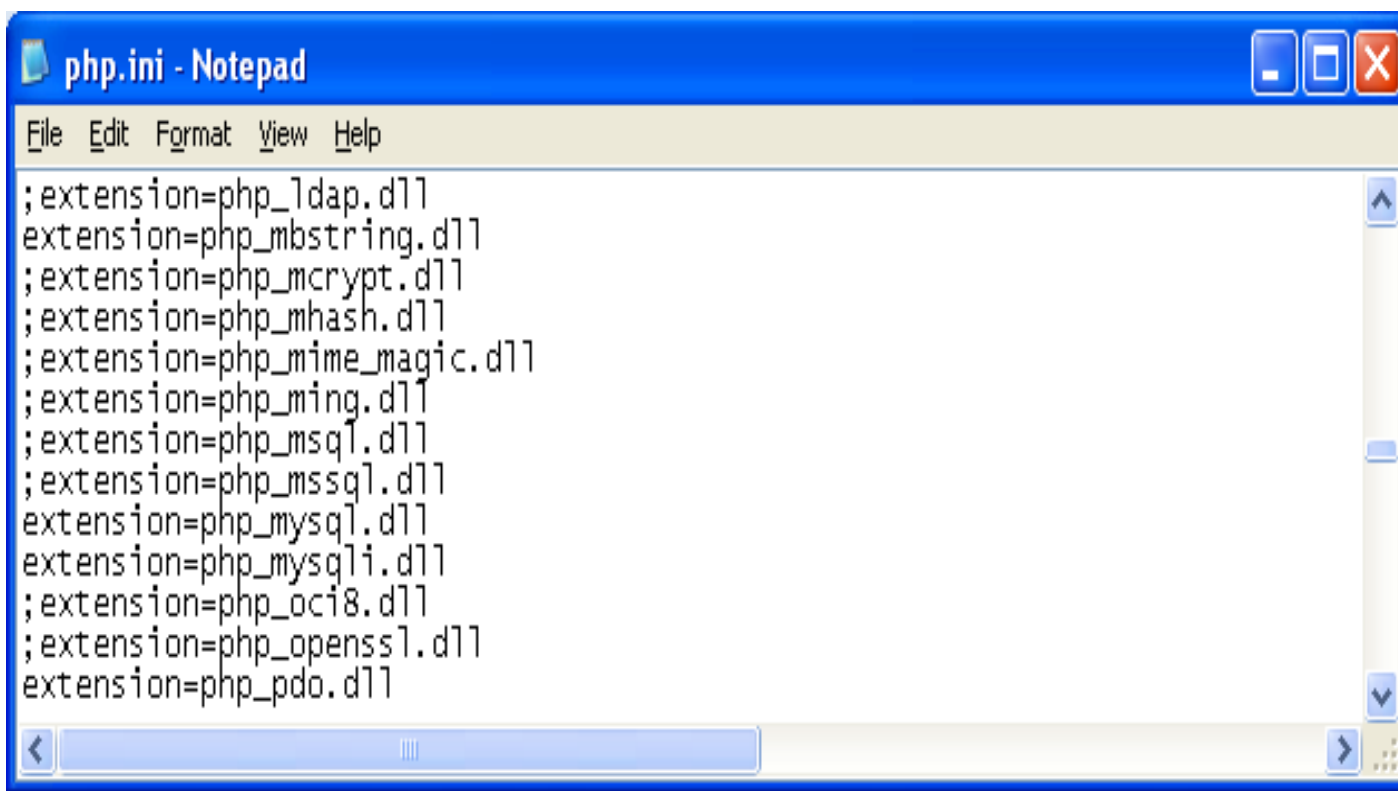


## Built-in Function

- PHP menyediakan banyak fungsi yang dapat digunakan.
- Ada juga fungsi yang membutuhkan extension. File extension harus di-load oleh PHP. Contoh: agar PHP anda ingin mengakses database menggunakan Oracle, maka extension `php_oci8.dll` harus di-load oleh PHP.
- Agar PHP dapat me-load extension tertentu, maka harus ada konfigurasi yang dilakukan pada file `php.ini`.
- Hilangkan tanda titik koma (;) jika extension ingin di-load oleh PHP



## Built-in Function



```

File Edit Format View Help
;extension=php_ldap.dll
extension=php_mbstring.dll
;extension=php_mcrypt.dll
;extension=php_mhash.dll
;extension=php_mime_magic.dll
;extension=php_ming.dll
;extension=php_mysql.dll
;extension=php_mssql.dll
extension=php_mysql.dll
extension=php_mysql_i.dll
;extension=php_oci8.dll
;extension=php_openssl.dll
extension=php_pdo.dll
  
```