

PHPdetail - Function

Mahadji, kinta

mahadji_kinta@yahoo.com

Lisensi Dokumen:

Copyright © 2003-2008 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Programmer yang handal adalah programmer yang mampu membuat kode program se-efisien mungkin, bayangkan jika kode-kode program yang memiliki fungsi yang sama ditulis berulang-ulang, ditempatkan di setiap kode program, hal ini tentulah tidak efisien dan akan mengakibatkan pembengkakan file dari suatu program.

Fungsi hadir untuk memberikan solusi tersebut, kode-kode program yang sama dibuatkan fungsinya, jika diperlukan maka tinggal panggil saja.

Fungsi pada dasarnya terdiri dari blok-blok program yang dapat dipanggil melalui `include()`, memasukkannya langsung ke kode program utama, dan memanggil nama fungsinya.

Hal terpenting dari fungsi adalah fungsi dapat menerima value dari luar, atau dikenal dengan parameter, value ini kemudian dapat di olah dan hasilnya dapat digunakan di luar fungsi.

Dengan mempelajari fungsi maka Anda telah satu langkah untuk menjadi seorang efisien programmer.

Jenis fungsi

1. Internal function : fungsi yang sudah disediakan pada library PHP, untuk memakainya cukup memanggilnya saja.

2. Define function : fungsi yang dibuat sendiri.

Manfaat menggunakan fungsi

1. Kode program lebih mudah di baca.
2. Jika terdapat kesalahan mudah untuk di koreksi.
3. perintah atau prosedur dalam suatu program dapat langsung diubah dari fungsinya, tanpa harus mengubah program utama, dengan begitu akan meningkatkan efisiensi dalam programming.

Memanggil fungsi

Untuk memanggil suatu fungsi caranya :

```
nama_fungsi ( $parameter );
```

Di mana \$parameter adalah value yang nantinya di olah pada fungsi. \$parameter dapat berupa tipe data integer, float atau \$variabel.

Contoh :

```
fungsi_hitung(2);  
fungsi_hitung(2.7);  
fungsi_hitung($variabel);
```

Note :

\$variabel juga dapat berupa array.

Contoh fungsi dan parameternya

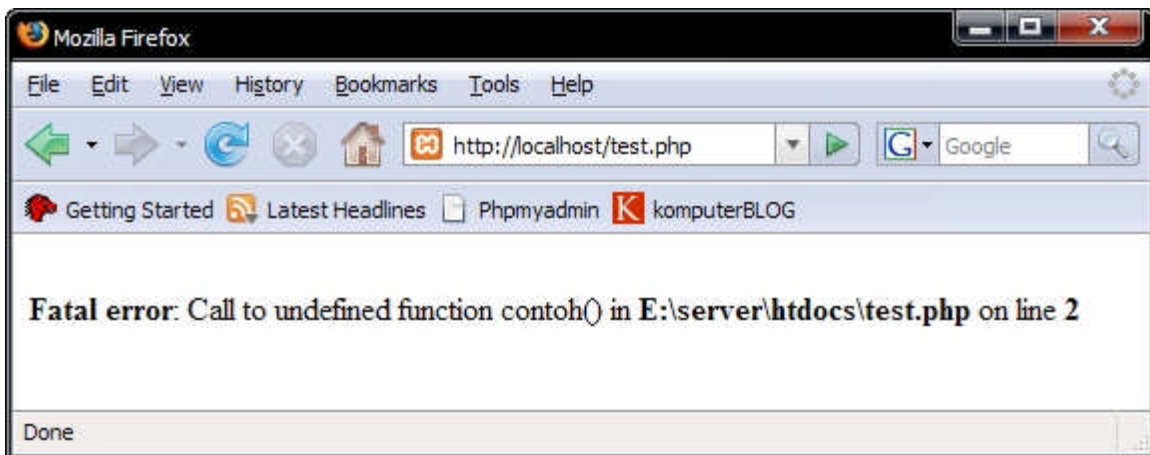
```
strip_tags ( string $str [, string $allowable_tags] )
```

Penjelasan fungsi :

- strip_tags berfungsi untuk membuang tag-tag HTML & PHP dalam suatu string.
- \$str adalah variable yang mau diolah.
- tanda [...] bersifat optional, artinya boleh diberikan atau tidak.

Note :

Pesan error akan muncul pada browser bila fungsi yang dipanggil tidak terdapat dalam library PHP atau salah penamaan, lihat gambar 1.



Gambar 1 : Pesan error ketika memanggil fungsi yang tidak ada atau salah penamaan

Pesan error yang ditampilkan di browser sangatlah berguna, dari pesan error kita dapat mengetahui di mana letak kesalahan ketika melakukan programming.

Ada dua hal yang harus diperhatikan ketika melihat pesan error :

1. Apakah nama fungsi yang dipanggil benar.
2. Jika memakai fungsi yang ada dalam Library PHP, cek apakah fungsi yang dipanggil terdapat pada versi PHP yang digunakan.

Berdasarkan pengalaman penulis, hal yang sering terjadi adalah, penulis lupa fungsi yang mau dipakai memakai tanda pemisah “_” atau tidak, sebagai contoh :

- `str_replace (mixed $search, mixed $replace, mixed $subject [, int &$count])`
- `strtok (string $str, string $token)`

Keduanya memiliki 3 huruf depan yang sama “str”, suatu saat lupa yang memakai pemisah “_” yang mana ? oleh karena itu sampai saat ini, PHP Manual selalu berada dalam USB, manakala dibutuhkan sebagai acuan tinggal search saja ☺.

Fungsi bersifat Case Insensitive

Dalam hal penamaan, fungsi bersifat insensitive, fungsi contoh(); , Contoh(); , CONTOh(); , akan

memberikan hasil yang sama.

Note :

Disarankan, ketika membuat dan memanggil suatu fungsi usahakan untuk menggunakan lowercase (huruf kecil) karena hal ini sesuai dengan acuan dalam Library PHP.

Membuat Fungsi Sendiri

Suatu fungsi yang terdapat dalam Library PHP dapat melakukan berbagai macam hal, seperti menggunakan dan menghubungkan database, membuat grafik/chart, membuat zip on the fly, mengecek suatu format email, memotong dan menggabungkan text, memecah atau menggabungkan array, membuat enkripsi suatu string, mengcopy file dari server ke server, membuat folder, dan macam sebagainya.

Namun tetap saja hal itu memiliki keterbatasan. Dengan membuat fungsi sendiri, Anda dapat dengan bebas menggabungkan, memodifikasi antara satu fungsi dan fungsi lainnya. Dengan begitu diharapkan fungsi yang Anda buat efisien dan sesuai dengan kebutuhan.

Struktur Fungsi

Struktur dasar fungsi dalam PHP sebagai berikut :

```
function nama_fungsi()  
{  
    Taruh kode program di sini;  
}
```

Untuk membuat Fungsi ada beberapa hal yang harus diperhatikan :

- Awali setiap pembuatan fungsi dengan kata `function`.
- Nama fungsi dapat berupa angka, huruf atau campuran keduanya, namun huruf pertama **tidak boleh angka**.

Contoh penamaan fungsi yang salah :

```
function lcontoh ()  
{  
    Contoh;  
}
```

- Nama fungsi tidak boleh sama dengan yang ada dalam Library PHP.

Contoh :

```
Function echo()  
{  
    Kode;  
}
```

- Dalam tanda {...}, merupakan tempat untuk menuliskan kode program.
- Untuk memanggil fungsi, cukup dengan memanggil nama fungsinya.

Parameter

Parameter merupakan nilai yang diberikan kepada fungsi yang kemudian diolah dan hasilnya dapat digunakan di luar fungsi.

Suatu fungsi bisa terdiri dari 1 parameter, 2 parameter, 3 dan seterusnya, bahkan tanpa parameter sekalipun.

Contoh PHP script yang menggunakan parameter :

```
<?PHP  
//menetapkan variable $nilai=5  
$nilai = 5;  
  
//membuat fungsi hitungmath  
Function hitungmath($nilai)  
{  
    $pengali = 5;  
    $total = $pengali * $nilai;  
    echo $total;  
}  
  
//Memanggil fungsi  
Hitungmath($nilai);  
?>
```

Output : 25

Penjelasan :

- \$nilai = 5 adalah variable yang mau di olah dalam fungsi hitungmath.

- Function `hitungmath($nilai)` -> membuat suatu fungsi dengan nama `hitungmath` dan parameter `$nilai`, `$nilai` di sini merupakan nilai yang mau diolah dalam fungsi `hitungmath`, dalam hal ini nilai yang mau diolah sebesar `$nilai` atau 5.
- `Hitungmath($nilai);` -> memanggil fungsi `hitungmath`.

Scope Variable

Scope variable dapat diartikan sebagai sebuah cakupan wilayah yang dapat ditangani oleh variabel.

PHP memiliki 2 jenis scope variable :

1. Lokal variable yaitu variable yang **hanya** dapat digunakan di dalam fungsi.
2. Global variable yaitu variable yang dapat digunakan di luar dan di dalam fungsi.

PHP memiliki beberapa syarat dalam variable scope, diantaranya :

1. Variabel yang dideklarasikan di dalam sebuah function, hanya dapat digunakan pada fungsi tersebut. Jenis scope variable ini dikenal dengan local variable.
2. Variabel yang dideklarasikan di luar function, di kenal dengan global variable.
3. Menggunakan perintah `require()` dan `include()` tidak mempengaruhi area scope variabel. Jika suatu ketika memanggil dengan perintah di atas dan variable terletak di dalam suatu fungsi maka variable yang terdapat didalamnya adalah local variable, namun jika variable terletak diluar fungsi maka variable tersebut akan dianggap sebagai global variable.
4. Suatu variable dalam fungsi dapat dijadikan global variable dengan cara menambahkan kata `global` pada variable yang dideklarasikan.
5. Variable yang terbentuk dapat dihapus dengan cara `unset($nama_variabel)`.

Agar lebih paham mengenai variable scope, simak studi kasus berikut :

variableslope1.php

```
Function scope()  
{  
    $variabel = "nilai";  
}  
echo $variabel;
```

Apa yang ditampilkan jika script di atas dijalankan ? jawabannya tidak ada, lho kenapa ? Jika Anda benar-benar membaca dengan seksama mengenai perbedaan local variable dan global variable maka Anda tidak akan bingung. Ok, tenang saja, kita akan membahasnya kok ☺.

`$variabel` yang ada pada `function scope()`, dinamai local variable, variable ini hanya dapat

digunakan di dalam `function scope()` saja.

\$variabel diluar `function scope()`, dinamai global variable, variable ini bersifat global, karena \$variabel tidak didefinisikan sebelumnya maka tidak ada hal yang ditampilkan.

variablescope2.php

```
<?PHP
function kasus2()
{
    echo "Nilai variable sebelum deklarasi variabel = $var";
    $var = 15;
    echo "nilai variable setelah deklarasi variable = $var";
}

Kasus2();

echo $var;
?>
```

Tebak apa yang keluar tanpa menjalankannya di browser ?

Pass by value dan Pass by reference

Passby1.php

```
<?PHP
function pass($value, $tambah = 1)
{
    $value = $value +$tambah;
}
$value = 10;
pass ($value);
echo $value;
?>

//print out
10
```

Script diatas akan menghasilkan \$value = 10 bukan \$value = 11 mengapa ? hal ini disebabkan karena peraturan scope variable. Kode \$value = 10 akan membuat sebuah variable bernama value dan bernilai 10, kemudian pada baris `pass ($value);` fungsi pass dipanggil. Selanjutnya \$value = 10 diproses, `$value = $value +$tambah;`, dengan begitu nilai dari \$value = 11, sesuai dengan peraturan scope variabel, \$value = 11 hanya akan sampai pada baris terakhir pada fungsi, nilai yang telah diolah tidak akan dapat dipakai di luar, singkatnya bersifat local variable.

Apa itu Pass by value dan Pass by reference?

Pass by value dan Pass by reference adalah cara bagaimana variable diolah.

Pass by value, variable yang diberikan oleh parameter hanya akan diolah didalam fungsi, variable yang berada diluar fungsi akan tetap. Dengan kata lain variable yang dipakai oleh suatu fungsi tidak akan mempengaruhi variable yang ada diluar.

Pass by reference, jika pada pass by value, nilai hasil olahan tidak dapat digunakan diluar fungsi, maka pass by reference sebaliknya, hasil olahan yang telah diproses di dalam fungsi dapat digunakan di luar fungsi, dengan kata lain value yang di luar fungsi akan turut berubah seiring perubahan yang terjadi pada variable didalam fungsi.

Passbyreference.php

```
<?PHP
Function passref(&$value, $tambah=1)
{
    $value = $value + $tambah;
}

$x = 10;
Echo "nilai sebelum fungsi ". $x."<br>";
Passref($x);
Echo "nilai sesudah fungsi ". $x."<br>";
?>
//print out
nilai sebelum fungsi 10
nilai sesudah fungsi 11
```

Note :

Pemahaman yang mendalam mengenai scope variable, pass by value dan pass by reference amatlah penting, karena dalam web development skala besar hal-hal diatas amatlah diperlukan, terutama dalam hal efisiensi kode program.

Fungsi Return

Return berfungsi untuk menghentikan kode program.

Contohreturn.php


```
<?PHP
function datacek($name, $pass)
{
    if(!isset($name) || !isset($pass))
    {
        echo "Data yang Anda isikan tidak lengkap";
        return;
    }
    if(!empty($name) && !empty($pass))
    {
        echo "selamat datang";
    }
}

//variabel

datacek($username, $password);

?>

//input1
$username = "kinta mahadji";
$password = "root";
```

Masukan input1 dibawah kode blok //variable dan lihat hasilnya.

```
//input2
$username = "kinta mahadji";
```

Ganti input1 dengan input 2 dibawah kode blok //variable dan lihat hasilnya.

Penjelasan script **Contohreturn.php**

Fungsi dari isset() adalah untuk mengecek apakah variable sudah terbentuk, dan fungsi dari empty() adalah untuk memastikan bahwa variable yang diberikan tidaklah kosong.

Ketika diberikan input1, variable2 terpenuhi, sehingga kode akan langsung mengecek apakah variable yang diberikan kosong atau tidak, jika tidak kosong maka fungsi echo akan dijalankan.

Ketika diberikan input2, variabel2 yang dibutuhkan tidak terpenuhi, pada saat melakukan pengecekan variable pada baris if(!isset(\$name) || !isset(\$pass)), maka fungsi echo akan dijalankan dan kode akan dihentikan dengan return.

sebagai latihan, saya sertakan script-script yang berhubungan dengan function.

Sampai disini dulu mengenai fungsi, tetap berlatih mulai dari script yang sederhana, menengah sampai expert, agar programming Anda terlatih, jangan malu untuk bertanya karena sesungguhnya tulisan ini masih jauh dari sempurna. Agar penulis bisa menghasilkan tulisan yang

Komunitas eLearning IlmuKomputer.Com
Copyright © 2003-2007 IlmuKomputer.Com

lebih baik lagi, penulis dengan senang jika ada pembaca yang mau mengirimkan saran dan kritiknya, terima kasih, semoga bermanfaat.

Script – script

sample1.php

A function to validate a username and password.

```
<?php
function validatelogin ($username, $password){
//Typically the username and password would be validated against
information
//in the database. For the sake of simplicity in this example, the username
//and password are hard-coded into variables.
$actualuser = "myusername";
$actualpass = "mypassword";
//Now, you do a quick comparison to see if the user
has entered the correct login.
if (strcmp ($username, $actualuser) == 0 &&
strcmp ($password, $actualpass) == 0){
return true;
} else {
return false;
}
}
//You then call the function and pass in the values you want checked.
if (validatelogin ("myusername","mypassword")){
echo "You are logged in correctly";
} else {
echo "You have an incorrect username and/or password";
}
?>
```

sample2.php

A function to return the sum of three values.

```
<?php
function addvalues ($value1 = 0, $value2 = 0, $value3 = 0){
//Now the function takes the three values and adds them.
$total = $value1 + $value2 + $value3;
return $total;
}
```

```
//Now, if you forget a value or two, it will still work.  
echo addvalues (1) . "<br />"; //Echoes a 1.  
//If you pass all the arguments, you will still get a valid result.  
echo addvalues (1,2,3); //Echoes a 6.  
?>
```

sample3.php

A function to concatenate text.

```
<?php  
function attachtext (&$newtext = ""){  
//Now the function attaches the received text.  
$newtext = $newtext . " World!";  
}  
//Here is the current block of text.  
$mystring = "Hello";  
//Then you call the function to attach new text.  
attachtext ($mystring);  
//And when you echo the variable now...  
echo $mystring; //Outputs Hello World!  
?>
```

sample4.php

A function to add up any number of values.

```
<?php  
function addanything (){  
//Default the return value.  
$total = 0;  
//Get the full list of arguments passed in.  
$args = func_get_args ();  
//Loop through the arguments.  
for ($i = 0; $i < count ($args); $i++){  
//Make sure the value is an integer.  
if (is_int ($args[$i])){  
//And add to it if necessary.  
$total += $args[$i];  
}  
}  
//Then return the total.  
return $total;  
}  
//Now, you can pass the function any numbers.  
echo addanything (1,5,7,8,11) . "<br />"; //Outputs 32.  
echo addanything (1,1) . "<br />"; //Outputs 2.  
//It will ignore noninteger values.
```

```
echo addanything (1,1,"Hello World"); //Still outputs 2.  
?>
```

sample5.php

Function that will take in a set of values, calculate them

```
<?php  
then return the values.  
function addandsubtract ($firstvalue, $secondvalue){  
//The first thing we need to do is add the values.  
$firstreturnvalue = ($firstvalue + $secondvalue);  
$secondreturnvalue = ($firstvalue - $secondvalue);  
//Now, you declare an array.  
$myarray = array ();  
//Then put the two return values into the first two indexes of the array.  
$myarray[0] = $firstreturnvalue;  
$myarray[1] = $secondreturnvalue;  
//Then you can return the entire array.  
return $myarray;  
}  
//Now, when you call the function, it will return the two values in array  
format.  
$myarray = array ();  
$myarray = addandsubtract (10, 3);  
echo $myarray[0] . "<br />"; //Will echo 13.  
echo $myarray[1]; //Will echo 7.  
?>
```

Sample6.php

A function to add two values

```
<?php  
function addvalues ($firstvalue = 0, $secondvalue = 0){  
return $firstvalue + $secondvalue;  
}  
//A function to subtract two values.  
function subtractvalues ($firstvalue = 0, $secondvalue = 0){  
return $firstvalue - $secondvalue;  
}  
//A function to multiply two values.  
function multiplyvalues ($firstvalue = 0, $secondvalue = 0){  
return $firstvalue * $secondvalue;  
}  
//And let's assume these are the values you want to work with.  
$firstvalue = 10;
```

```
$secondvalue = 3;
//Let's say this value represents a user-submitted value.
$whattodo = "addvalues";
//You can then call the function as a variable.
echo $whattodo($firstvalue, $secondvalue) . "<br />";
//Let's say this value represents a user-submitted value.
$whattodo = "subtractvalues";
//You can then call the function as a variable.
echo $whattodo($firstvalue, $secondvalue) . "<br />";
//Let's say this value represents a user-submitted value.
$whattodo = "multiplyvalues";
//You can then call the function as a variable.
echo $whattodo($firstvalue, $secondvalue) . "<br />";
?>
```

Referensi :

1. Apress.PHP.5.Recipes.A.Problem.Solution.Approach.Sep.2005
2. <http://www.php.net>
3. PHP and MySQL Web Development
4. PHP 5 Manual

Biografi Penulis



Kinta Mahadji. Biasa disapa dengan kinta. Sekarang sedang studi di Politeknik Negeri Jakarta, Depok, Jurusan Telekomunikasi. Belajar komputer sejak SMA secara otodidak. Menyukai programing dan membaca novel. Turut bekerja juga sebagai seorang freelancer untuk web development.