



PHP Dasar: Belajar Fungsi (1/3)



PHP

PHP Dasar



n Fungsi

ngsi

ngsi

h dari satu parameter

ta Parameter

utnya

Bagikan:



Kita masih berada di [seri tutorial belajar php dasar](#). Kita telah mempelajari beberapa unsur penting pada PHP: mulai dari [variabel dan tipe data](#), [macam-macam operator](#), hingga [control structure](#) pada PHP.

Pada pertemuan kali ini, kita akan membahas tentang fungsi.

Pengertian Fungsi

Fungsi adalah suatu kumpulan blok kode, yang menerima suatu inputan, melakukan satu tugas tertentu, dan secara opsional ia bisa mengembalikan suatu nilai. Dalam tutorial-tutorial sebelumnya, kita telah mencoba beberapa fungsi bawaan php seperti `var_dump` dan `count`.

PHP memang sudah datang dengan berbagai macam fungsi untuk memudahkan pekerjaan kita. Akan tetapi kita tetap bisa membuat fungsi sendiri, untuk menyelesaikan tugas-tugas tertentu sehingga kita bisa menggunakan fungsi tersebut tanpa harus menulis kode program berulang-ulang.

Cara Mendefinisikan Fungsi



memiliki nama, dan juga bisa menerima parameter.

anda bisa mendefinisikan fungsi sebagai berikut:

```
fungsi () {  
    program
```

Peraturan pemberian nama fungsi, sama dengan nama variabel. Hanya saja ia tidak bisa menggunakan nama fungsi yang sama seperti fungsi bawaan php seperti misalnya `var_dump`, `empty`, `count`, dan lain-lain. Kita juga tidak bisa mendefinisikan satu nama fungsi yang sama sebanyak 2 kali.

Contoh yang lain, berikut ini adalah fungsi yang bertugas untuk menampilkan teks dengan `echo`.

```
<?php
```

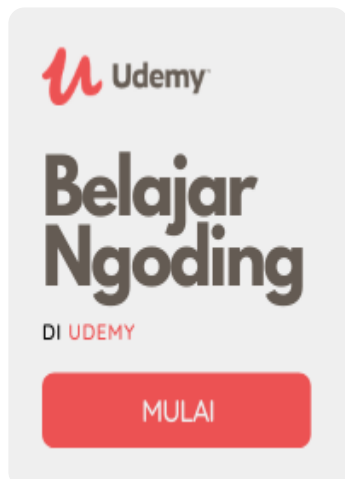
```
function sapaPengunjung () {  
    echo "<h1>Halo, selamat datang!</h1>";  
    echo "<p>Terima kasih telah berkunjung ke situs kami.</p>";  
}
```

Bagaimana cara memanggilnya?

Cara Memanggil Fungsi

Di atas, kita telah membuat satu fungsi dengan nama `sapaPengunjung()`. Kita juga telah mendefinisikan tugas apa yang harus fungsi tersebut lakukan.

Nah.. sekarang kita akan memanggil fungsi tersebut.



in menghasilkan teks html yang di-`echo` melalui fungsi `sapaPengunjung()`.

Parameter Pada Fungsi

Kita sudah singgung pada bagian pengertian fungsi, bahwa fungsi pada PHP bisa menerima suatu nilai atau input. Nilai atau input tersebut, kita katakan sebagai parameter fungsi.

Parameter fungsi adalah suatu nilai yang kita lempar kedalam sebuah fungsi, nilai tersebut bisa berupa apa saja. Bisa berupa `string`, `boolean`, `integer`, bahkan ia juga bisa berupa fungsi yang lainnya (bagian ini insyaallah kita bahas pada tutorial fungsi bagian 2).

Sebagai contoh, kita akan mengubah fungsi `sapaPengunjung()` yang telah kita buat di atas.

Kita akan menambahkan parameter `$nama` yang berisi nama pengunjung. Nama pengunjung tersebut akan kita tampilkan dengan perintah `echo`.

```
<?php
```

```
function sapaPengunjung ($nama) {  
    echo "<h1>Halo {$nama}, selamat datang!</h1>";  
    echo "<p>Terima kasih telah berkunjung ke situs kami.</p>";  
}
```



```
sapaPengunjung("Nurul Huda");
```

anggil fungsi yang telah kita buat tersebut berkali-kali, bahkan parameter yang berbeda-beda!

```
<?php
```

```
sapaPengunjung("Nurul Huda");  
sapaPengunjung("Ibnu Zakariyya");  
sapaPengunjung("Anshori Akbar");
```

Fungsi dengan lebih dari satu parameter

Kita telah mencoba fungsi dengan satu parameter. Sebenarnya, kita juga bisa mendefinisikan lebih dari satu parameter.

Misal pada fungsi `sapaPengunjung()` di atas, kita akan memberikan ucapan khusus bagi pengunjung yang telah mengunjungi situs kita lebih dari 10 kali.

Bagaimana kita tahu bahwa pengunjung tersebut telah mengunjungi lebih dari 10 kali? Kita akan menambahkan parameter ke-2, yaitu jumlah kunjungan.

```
<?php
```

```
function sapaPengunjung ($nama, $jumlahKunjungan) {  
    echo "<h1>Halo {$nama}, selamat datang!</h1>";  
    echo "<p>Terima kasih telah berkunjung ke situs kami.</p>";  
  
    if ($jumlahKunjungan > 10) {
```

```
}
```



gsi, jangan lupa untuk menambahkan parameter kedua.

```
Nurul Huda", 20);
```

Membatasi Tipe Data Parameter

Kita telah membuat fungsi yang menerima 2 buah parameter. Parameter pertama kita beri nama `$nama`, ia dimaksudkan untuk menerima tipe data `string`. Dan parameter kedua kita beri nama `$jumlahKunjungan`, ia dimaksudkan untuk menerima tipe data `integer`.

Akan tetapi.

Jika kita mau perhatikan, sebenarnya kita bisa memasukkan tipe data apa pun di situ. Bisa `string`, `integer`, `boolean`, dan sebagainya.

```
<?php
```

```
sapaPengunjung(true, "Nurul Huda");
```

Itu terjadi karena kita tidak membatasi tipe data apa yang harus dimasukkan ketika memanggil fungsi `sapaPengunjung`.

Tentu saja dengan memasukkan parameter dengan tipe data yang tidak sesuai keinginan, akan membuat fungsi yang telah kita buat tersebut menjadi bekerja tidak seperti yang kita harapkan.



```
    sapaPengunjung (string $nama, int $jumlahKunjungan) {
```

Pada kode program di atas kita mendefinisikan parameter `$nama` dengan tipe data `string`, dan parameter `$jumlahKunjungan` dengan tipe data `integer`.

Coba anda lakukan perintah berikut dan perhatikan pesan error yang anda dapatkan:

```
<?php
```

```
sapaPengunjung("Nurul Huda", "20");
```

Parameter Default

Terkadang kita ingin bahwa suatu fungsi memiliki parameter opsional atau parameter yang tidak wajib diisi. Seperti untuk me-*resize* gambar, kita mungkin akan membuat fungsi dengan nama `resize()` yang menerima tiga parameter: yaitu parameter `$urlGambar`, `$lebar` dan `$tinggi`.

Fungsi tersebut akan mengubah ukuran gambar `$urlGambar` menjadi ukuran baru yaitu sesuai dengan parameter `$lebar` dan `$tinggi`.

Akan tetapi yang kita inginkan, parameter `$tinggi` ini opsional, tidak wajib diijisi. Jika variabel `$tinggi` tidak diisi, maka otomatis gambar akan di-*resize* sesuai dengan ukuran rasionya.

Untuk melakukan hal tersebut, kita bisa melakukan kurang lebih seperti ini:



```
(string $urlGambar, float $lebar, float $tinggi = null) {  
    {  
        eter $tinggi null, maka resize gambar sesuai rasio  
  
        ata variabel $tinggi ada nilainya, maka resize  
        gan nilai tersebut
```

Atau kita bisa ubah fungsi kita sebelumnya yaitu `sapaPengunjung` dengan menjadikan kedua parameternya memiliki nilai default.

```
<?php
```

```
function sapaPengunjung (string $nama = 'Anonimous', int $jumlahKunjungan  
    echo "<h1>Halo {$nama}, selamat datang!</h1>";  
    echo "<p>Terima kasih telah berkunjung ke situs kami.</p>";  
  
    if ($jumlahKunjungan > 10) {  
        echo "<p>Kami memiliki hadiah ebook gratis untuk anda karena anda tel  
    }  
}
```

Pembahasan Selanjutnya

Anda bisa istirahat sebentar sambil menyeduh kopi karena pembahasan kita tentang fungsi masih belum selesai. Kita masih memiliki dua *part* lagi.

Pada [pembahasan selanjutnya](#), kita akan mempelajari fungsi yang mengembalikan nilai, callback/closure, dan arrow function.

Penrograman PHP: Pemula Sampai Mahir.



**Belajar
Ngoding**

DI UDEMY

MULAI

rograman PHP dari pemula sampai mahir disertai studi
kan selalu di-update secara berkala.

(169 Peserta)

[Ambil Kelas](#)



Nurul Huda

Web Developer. FOSS addict. Pengguna Arch Linux (dan Ubuntu). Penyuka kopi saset. Dan pernah kuliah Teknik Informatika sampai lulus.

Dukung Jago Ngoding ☕

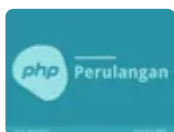
[← Sebelumnya](#)

[Selanjutnya →](#)

[PHP Dasar: Perulangan](#)

[PHP Dasar: Belajar Fungsi \(2/3\)](#)

Artikel Terkait



[PHP Dasar: Perulangan](#)



[PHP Dasar: Logika Percabangan](#)



[PHP Dasar: Macam Macam Operator](#)



**Belajar
Ngoding**

DI UDEMY

MULAI

PHP Dasar: Hello World

by utteranc.es

Write

Preview

Sign in to comment

 Styling with Markdown is supported

Sign in with GitHub

© 2021 Jago Ngoding

Icons made by Freepik from www.flaticon.com