

5.1. Fungsi

Sebuah fungsi adalah sejumlah pernyataan yang dikemas dalam sebuah nama. Nama ini selanjutnya dipanggil beberapa kali di beberapa tempat dalam program.

Tujuan membuat fungsi:

Memudahkan dalam mengembangkan program

Menghemat ukuran program. Bermanfaat dan terasa jika ada beberapa deretan instruksi yang sama dan digunakan dalam bagian di dalam program.

5.2. Fungsi Buatan

Suatu fungsi dapat didefinisikan dengan tata cara penulisan berikut:

```
function kosong ($arg_1, $arg_2, ..., $arg_n)
{
    echo "contoh fungsi.\n";
    return $retval;
}
```

Semua kode PHP yang sah dapat berada didalam suatu fungsi, bahkan fungsi dan definisi class lainnya.

5.3. Nilai Balik Fungsi.

Semua Fungsi harus didefinisikan dahulu sebelum digunakan. Umumnya fungsi menerima masukan (argumen atau parameter). Masukan ini selanjutnya diproses oleh fungsi. Hasil akhir dari fungsi berupa sebuah nilai yang disebut nilai balik (return value).

Nilai dikembalikan dengan menggunakan perintah optional return. Sembarang tipe dapat dikembalikan, termasuk tipe list dan objek

```
function pangkat($num)
{
    return $num * $num;
}
echo pangkat(5); // menghasilkan '25'.
```

Fungsi hanya mengembalikan satu nilai, jika nilai yang akan dikembalikan banyak, anda dapat menggunakan suatu list:

```
function kosong() {
    return array (0, 1, 2);
}
list ($zero, $one, $two) = kosong();
```

5.4. Argumen Fungsi

Informasi dapat ditukarkan ke dan dari fungsi dengan argumen, dimana urutan variable atau konstanta dipisahkan dengan tanda koma yang terletak di antara tanda kurung buka dan kurung tutup pada deklarasi dan pemanggilan fungsi.

PHP mendukung argumen yang lewatkan dengan nilai (by value) secara defaultnya, dilewatkan secara acuan (by refrence) , dan argumen bernilai default. Argumen dengan jumlah yang tak tentu tidak didukung, tetapi efek yang sama dapat dicapai dengan menggunakan array.

```
function takes_array($input) {
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
```

Pada defaultnya, argumen fungsi dilewatkan dengan nilai. Tetapi jika anda ingin suatu fungsi dapat melakukan modifikasi terhadap argumennya, anda harus melewatkan mereka dengan acuan

5.5. Class dan Object

Suatu class adalah koleksi dari variabel-variabel dan fungsi yang bekerja dengan variabel-variabel tersebut

```
<?php
class Cart {
    var $items; // Items in our shopping cart

    // Add $num articles of $artnr to the cart

    function add_item ($artnr, $num) {
        $this->items[$artnr] += $num;
    }

    // Take $num articles of $artnr out of the cart

    function remove_item ($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return true;
        } else {
            return false;
        }
    }
}
?>
```

Class adalah tipe, yang mana merupakan cetak biru dari variabel yang sebenarnya. Anda dapat membuat suatu variabel yang diinginkan dengan operator new.

```
$cart = new Cart;
$cart->add_item("10", 1);
```

Hal ini membuat suatu objek \$cart dari class Cart. Fungsi add_item() dari objek tersebut dipanggil untuk menambah 1 item ke nomor artikel 10 ke cart.

Class dapat merupakan perluasan dari class lain. Class perluasan atau turunan memiliki semua variabel dan fungsi dari kelas dasarnya dan apa yang anda tambahkan pada definisi perluasan tersebut. Hal ini dilakukan dengan menggunakan kata kunci extend berikut.

```
class Named_Cart extends Cart {
    var $owner;

    function set_owner ($name) {
        $this->owner = $name;
    }
}
```

Hal ini mendefinisikan suatu class bernama Cart yang memiliki semua variabel dan fungsi dari Cart ditambah suatu variabel tambahan \$owner dan suatu fungsi tambahan set_woner(). Anda membuat suatu nama cart dengan cara biasanya dan sekarang anda dapat menentukan dan mengambil owner dari cart. Anda tetap dapat menggunakan fungsi cart normalnya pada nama cart :

```
$ncart = new Named_Cart; // Create a named cart
$ncart->set_owner ("kris"); // Name that cart
print $ncart->owner; // print the cart owners name
$ncart->add_item ("10", 1); // (inherited functionality from cart)
```

Diantara fungsi atau suatu class variabel \$this berarti objek ini. Anda dapat menggunakan \$this->sesuatu untuk mengakses variabel tertentu atau nama fungsi dalam objek anda.

Konstruktor adalah fungsi dalam suatu class yang mana secara otomatis dipanggil ketika suatu instance baru dari class dibuat. Suatu fungsi menjadi suatu konstruktor ketika memiliki nama yang sama dengan class.

```
class Auto_Cart extends Cart {
    function Auto_Cart () {
        $this->add_item ("10", 1);
    }
}
```

Hal ini mendefinisikan suatu Auto_Cart yang mana adalah suatu Cart ditambah dengan suatu konstruktor yang mana menginisialisasi chart dengan suatu item dari artikel nomor "10" setiap suatu Auto_Cart baru dibuat dengan "new". Konstruktor juga dapat memiliki argumen dan argumen ini dapat berupa optional, yang mana membuat mereka lebih berguna.

```
class Constructor_Cart {
    function Constructor_Cart ($item = "10", $num = 1) {
        $this->add_item ($item, $num);
    }
}
```

```
// Shop the same old boring stuff.
```

```
$default_cart = new Constructor_Cart;
```

```
// Shop for real...
```

```
$different_cart = new Constructor_Cart ("20", 17);
```