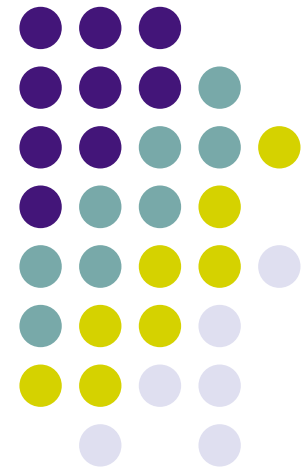


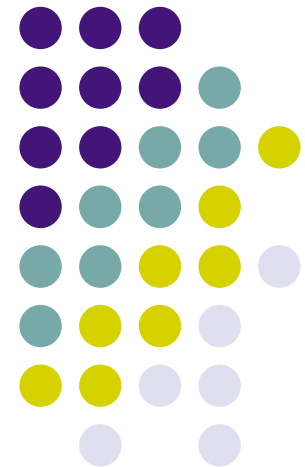
PHP

# Pemrograman Web II

Ganjil 2015 - 2016



# Function





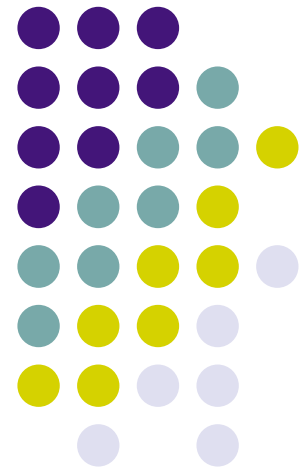
# Function di PHP

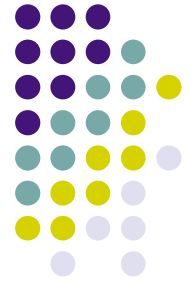
- Built-in Function
- User Defined Function

# Built-in Function

---

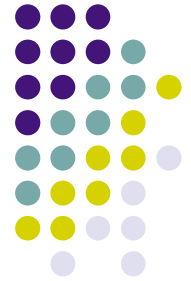
Fungsi-fungsi bantuan yang ada di  
PHP





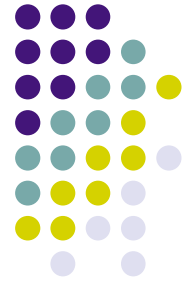
# Date

- <http://id1.php.net/manual/en/function.date.php>
- [http://www.w3schools.com/php/php\\_ref\\_date.asp](http://www.w3schools.com/php/php_ref_date.asp)
- `date()`
- `mktime()`
- `time()`
- `strtotime()`



# String

- <http://php.net/manual/en/ref.strings.php>
- [http://www.w3schools.com/php/php\\_ref\\_string.asp](http://www.w3schools.com/php/php_ref_string.asp)
- `strlen()`
- `strcmp()`
- `strstr()`
- `implode()` / `join()`
- `explode()`
- `htmlspecialchars()`

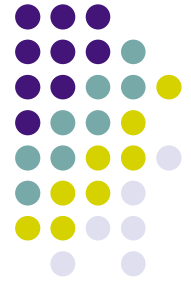


# Math

- <http://php.net/manual/en/ref.math.php>
- [http://www.w3schools.com/php/php\\_ref\\_math.asp](http://www.w3schools.com/php/php_ref_math.asp)
- `pow()`, `log()`, `pi()`
- `rand()`
- `max()`, `min()`
- `floor()`, `ceil()`, `round()`
- `sin()`, `cos()`, `tan()`

# Utility

- `count()`
- `isset()`
- `empty()`







# String Function – strlen()

- Menghitung panjang string / jumlah karakter pada sebuah string (termasuk spasi)

```
$teks1 = "Selamat datang di kuliah pemrograman web 2";  
$teks2 = "Universitas Pasundan";  
$teks3 = "sandhikagalih";
```

```
echo strlen($teks1) . "<br>";  
echo strlen($teks2) . "<br>";  
echo strlen($teks3) . "<br>";
```

42  
20  
13



# String Function – substr ( )

- Mengembalikan sebagian karakter dari string yang diberikan.
- substr(\$string, \$awal\_karakter, \$panjang)

```
$tulisan = "Selamat datang di kuliah pemrograman web 2";  
  
echo substr($tulisan, 8) . "<br>"; Mulai index ke 8, sampai selesai  
echo substr($tulisan, 8, 6) . "<br>"; Mulai index ke 8, sebanyak 6 char  
echo substr($tulisan, -17) . "<br>"; Mulai index ke 17, dari belakang
```

```
datang di kuliah pemrograman web 2  
datang  
pemrograman web 2
```



# String Function – substr\_count()

- Menghitung kemunculan sebuah substring yang ada pada string
- substr\_count(\$string, \$substring);

```
$tulisan = "Selamat datang di kuliah pemrograman web 2";  
  
echo substr_count($tulisan, "pemrograman web 2") . "<br>";  
echo substr_count($tulisan, "am") . "<br>";  
echo substr_count($tulisan, "a") . "<br>";
```

1  
2  
7



# String Function – `str_replace()`

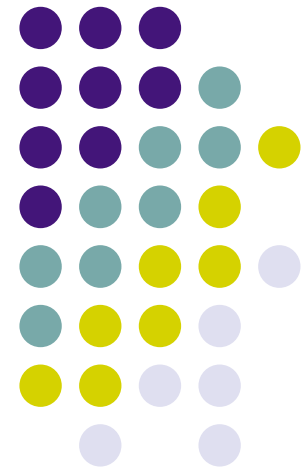
- Mengganti string / substring yang diberikan
- `str_replace($yang_dicari, $diganti_oleh, $string);`

```
$tulisan = "Selamat datang di kuliah pemrograman web 2";  
echo str_replace("a", "*", $tulisan) . "<br>";  
  
$kuliah = "pemrograman web 2";  
echo str_replace($kuliah, "<b>$kuliah</b>", $tulisan);
```

Sel\*m\*t d\*t\*ng di kuli\*h pemrogr\*m\*n web 2  
Selamat datang di kuliah **pemrograman web 2**

# User Defined Function

---





# Pengertian Function

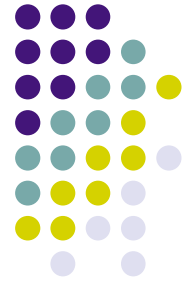


**Masukkan  
'bahan/material'  
ke dalam fungsi**

...

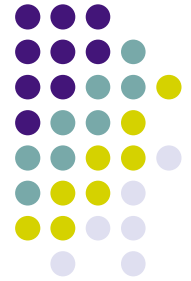
**Lakukan sesuatu  
terhadap 'bahan/  
material' tadi..**

**Akhirnya  
menghasilkan  
sesuatu**

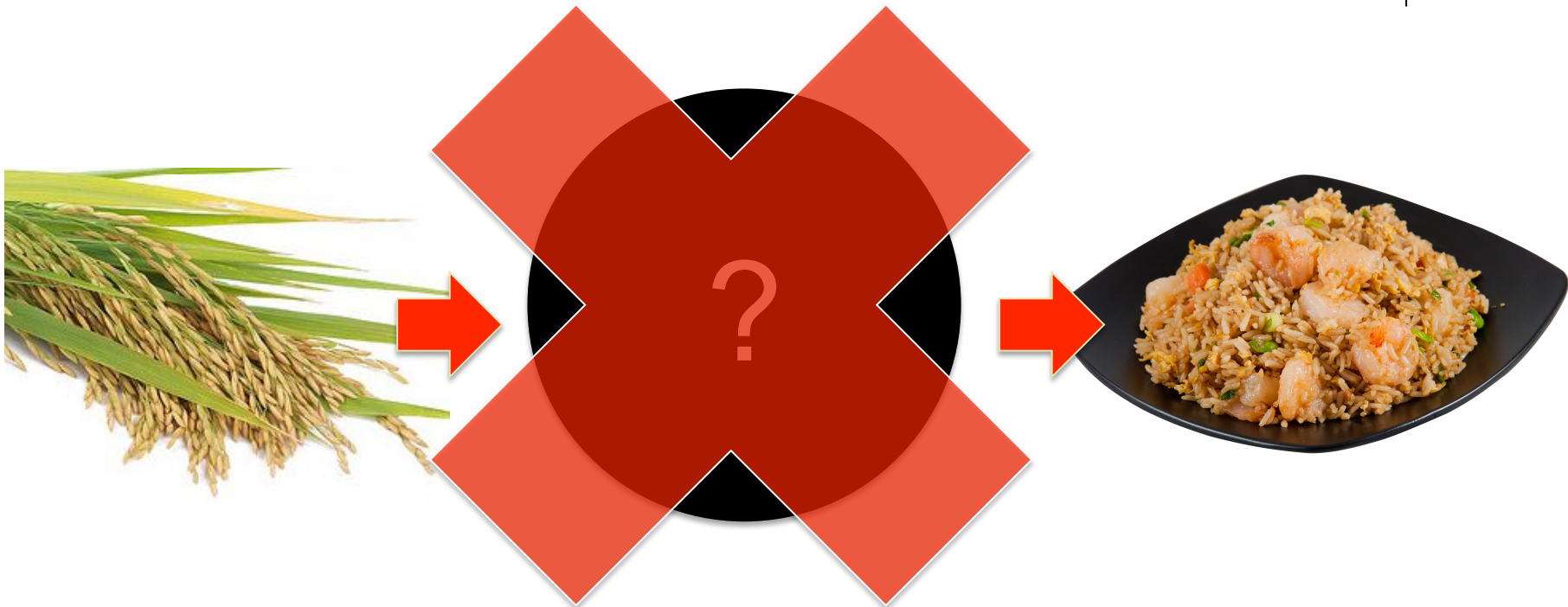


# Pengertian Function

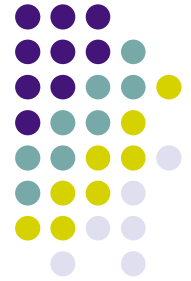
***“Function yang baik adalah function yang hanya mengerjakan 1 hal saja”***



# Pengertian Function

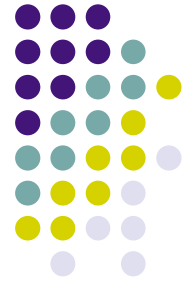






# Pengertian Function





# Pengertian Function





# Pengertian Function

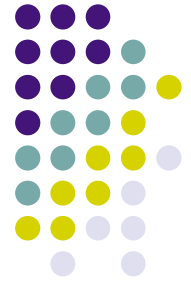


# Pengertian Function





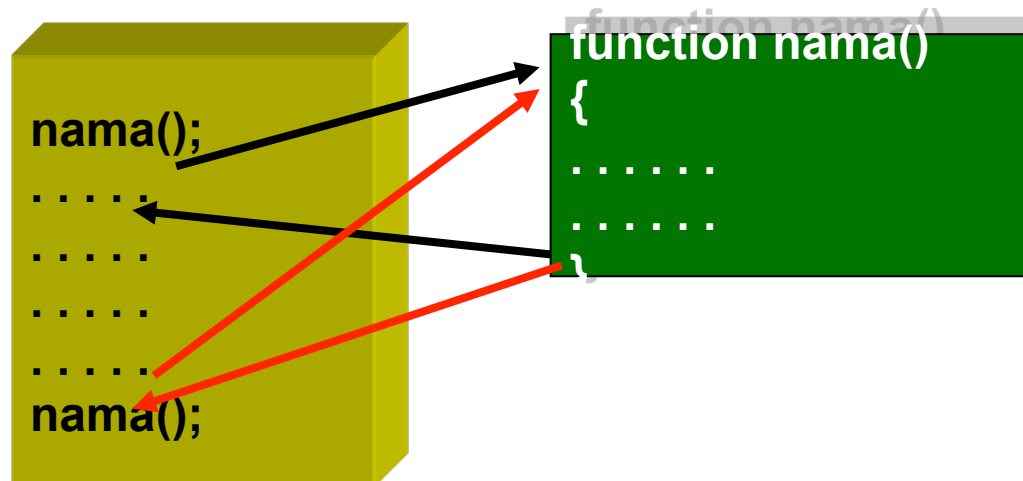
# Pengertian Function





# Pengertian Function

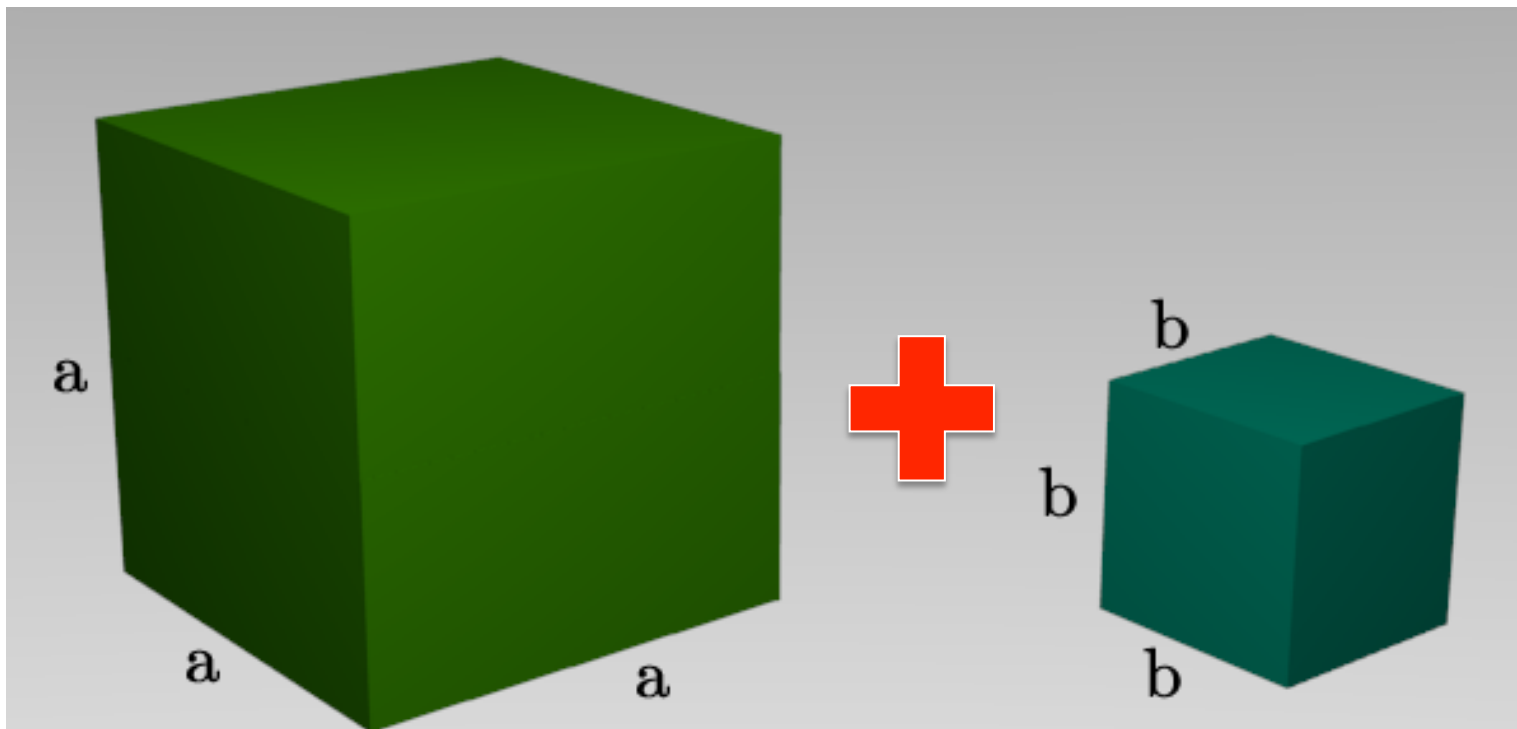
- Blok kode yang ditujukan untuk melaksanakan tugas tertentu
- Function dapat dipanggil berkali-kali
- Memudahkan pelacakan kesalahan & tidak perlu menulis berkali-kali
- Variabel yang dibuat di dalam fungsi, hanya bisa diakses oleh fungsi tersebut

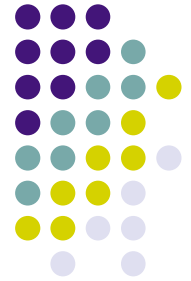




# Contoh

- Menghitung penjumlahan luas 2 buah kubus





# Menjumlahkan luas 2 buah kubus

1. Ketahui sisi kubus

9

4

2. Hitung luas masing-masing

$$9^3 = 729$$

$$4^3 = 64$$

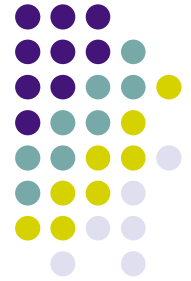
3. Jumlahkan hasilnya

$$729 + 64 = 793$$

4. Kembalikan nilai jawabannya

**793**





# Bagaimana kode PHPnya?

9 → `$a = 9;`

4 → `$b = 4;`

$4^3 = 64$  → `$luas_a = $a * $a * $a;`

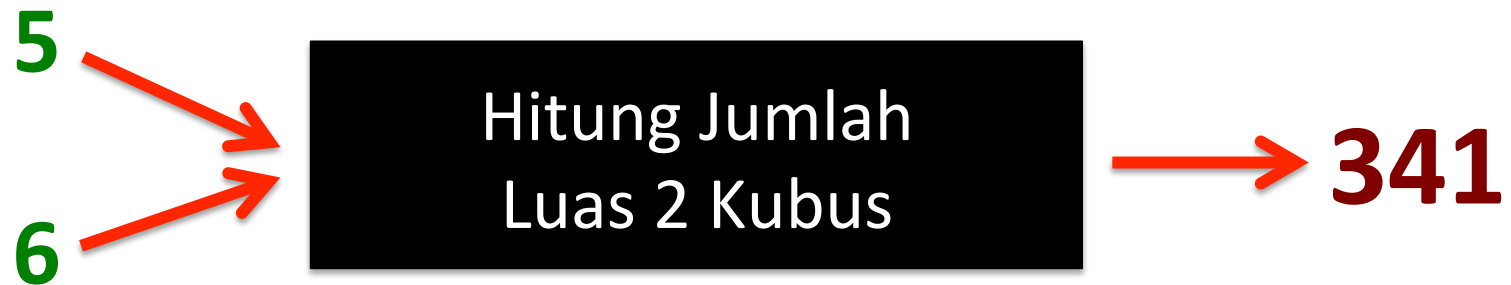
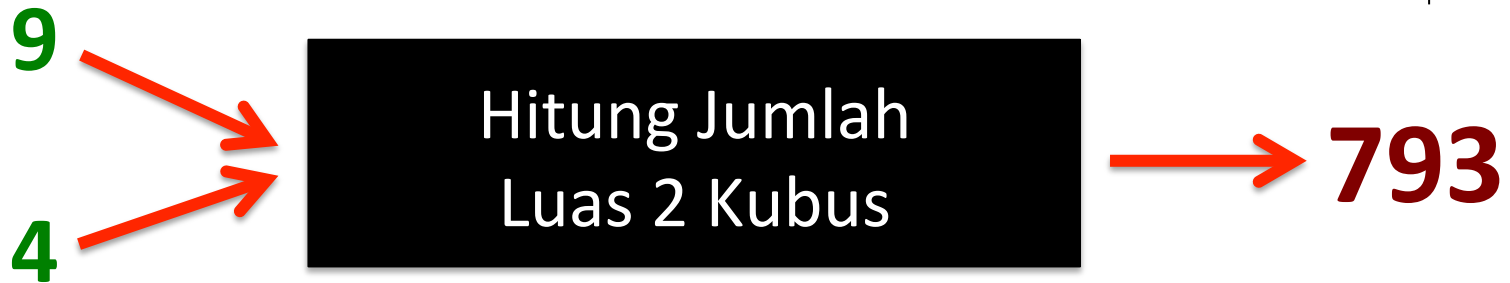
$9^3 = 729$  → `$luas_b = $b * $b * $b;`

$729 + 64 = 793$  → `$total = $luas_a + $luas_b;`

793 → `echo $total;`



# Reusability





# Membuat Function (1)

**function**

Keyword function untuk memberi tahu compiler bahwa kita kan memulai menulis fungsi

}

Selalu bungkus function dengan kurung kurawal

}



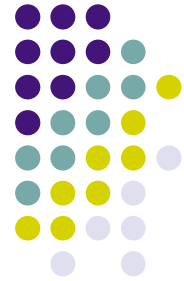
## Membuat Function (2)

```
function luas_dua_kubus($a, $b) {
```

Nama fungsi yang kita berikan harus jelas menggambarkan proses yang ada didalam fungsi tersebut

```
}
```

- Parameter / argumen ditulis di dalam kurung sebelum kurung kurawal pertama.
- Berguna sebagai bahan / material yang akan digunakan di dalam fungsi.
- Parameter boleh ada / tidak
- Jika ada, boleh sebanyak mungkin



## Membuat Function (3)

```
function luas_dua_kubus($a, $b) {  
  
    /* lakukan sesuatu disini */  
  
    return /* sesuatu (jika diperlukan) dari proses fungsi */  
}
```

- Kegunaan dari return adalah untuk memberitahu php 'Ok, kita sudah selesai mengerjakan sesuatu dan ini hasilnya..'
- Return bisa digunakan dimana saja di dalam fungsi untuk menghentikan jalannya fungsi
- Return biasanya ada, tapi boleh juga tidak ada



## Membuat Function (4)

```
function luas_dua_kubus($a, $b) {
```

1. Ketahui sisi kubus

2. Hitung luas masing-masing

3. Jumlahkan hasilnya

```
return $total; ← 4. Kembalikan nilai jawabannya
```

```
}
```



## Membuat Function (5)

```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
    $luas_b = $b * $b * $b;  
  
    $total = $luas_a + $luas_b;  
  
    return $total;  
  
}
```

Ketika parameter dikirim ke dalam fungsi, maka kita bisa menggunakannya sebagai variabel di dalam fungsi tersebut



# Memanggil Function

```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
    $luas_b = $b * $b * $b;  
  
    $total = $luas_a + $luas_b;  
  
    return $total;  
  
}
```

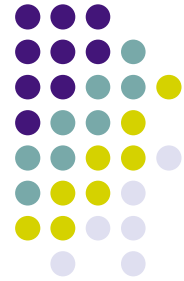
```
echo luas_dua_kubus(9,4);
```

Fungsi hanya mengembalikan nilai \$total, belum mencetaknya ke layar, jadi tetap harus menggunakan echo



# Refactoring (1)

*Membuat code menjadi lebih efisien*



```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
    $luas_b = $b * $b * $b;  
  
    $total = $luas_a + $luas_b;  
  
    return $total;  
  
}
```

Sebetulnya kode ini tidak ada masalah dan berjalan dengan normal, tetapi bisa kita singkat agar irit memory

# Refactoring (2)

*Membuat code menjadi lebih efisien*



```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
    $luas_b = $b * $b * $b;  
  
    $total = $luas_a + $luas_b;  
  
    return $total;  
}
```



```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
    $luas_b = $b * $b * $b;  
  
    return $luas_a + $luas_b;  
}
```



Keyword 'return' bisa langsung menghitung operasi matematik, jadi kita tidak perlu menggunakan \$total

# Refactoring (3)

*Membuat code menjadi lebih efisien*



```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
    $luas_b = $b * $b * $b;  
  
    $total = $luas_a + $luas_b;  
  
    return $total;  
}
```



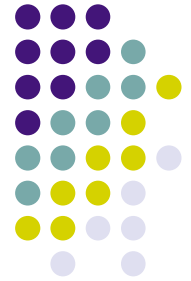
```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
    $luas_b = $b * $b * $b;  
  
    return $luas_a + $luas_b;  
}
```



```
function luas_dua_kubus($a, $b) {  
  
    return $a * $a * $a + $b * $b * $b;  
}
```



```
function luas_dua_kubus($a, $b) {  
  
    $luas_a = $a * $a * $a;  
  
    return $luas_a + $b * $b * $b;  
}
```



# Parameter / Argumen

```
function luas_dua_kubus($a, $b) {  
    return $a * $a * $a + $b * $b * $b;  
}  
  
echo luas_dua_kubus(9,4); // menghasilkan 793  
  
$hasil = luas_dua_kubus(10, 2);  
echo $hasil; // menghasilkan 1008  
  
$x = 4;  
echo luas_dua_kubus($x*3, $x/2); // menghasilkan 1736
```

# Terima Kasih

---

**Pemrograman Web 2 - SP20152016**  
sandhikagalih@unpas.ac.id

