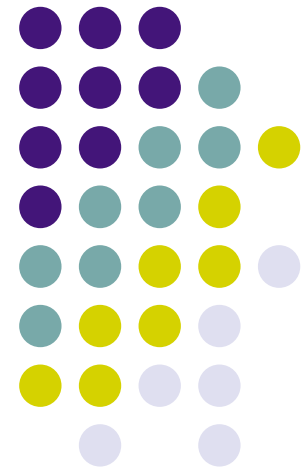


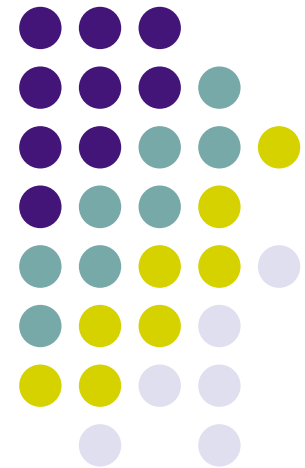
PHP

Pemrograman Web II

Ganjil 2013 - 2014



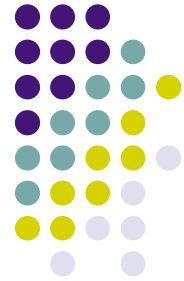
Variabel





Penamaan Variabel

- Diawali dengan karakter \$
- Nama variabel bisa terdiri dari karakter huruf, angka, dan *underscore*.
- Setelah karakter \$, harus diikuti karakter huruf atau *underscore* “_”.
- *Case Sensitive*
- Contoh :
 - \$_name
 - \$first_name
 - \$name3
 - \$lastName
 - global \$_name → variabel global



Variable Variables

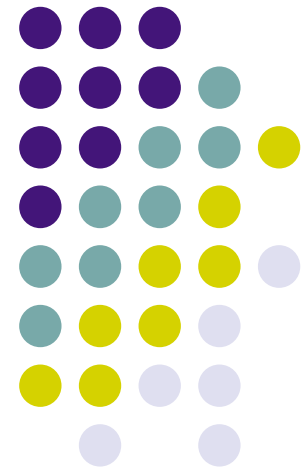
- Variabel yang diberi nama dari isi variabel lainnya.
- Dibuat saat script dieksekusi (*runtime*).
- Contoh

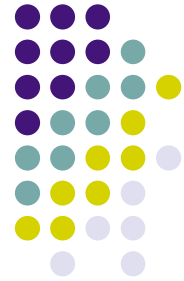
```
$name = "John";  
$$name = "Registered User";  
  
echo $John;
```

- Hasil

Registered User

Tipe Data





Tipe Data Standar

- **Numerik**

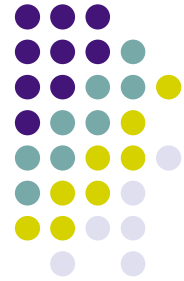
- Integer : 5, -7, 2000, 012 → octet, 0xff → hexadecimal
- Floating point : 2005.14, 14e+12

- **String**

- ‘teks dengan apostrophe’
- “teks dengan quote”

- **Escape character**

- Karakter khusus untuk mewakili karakter ASCII dengan fungsi khusus
- Diawali dengan tanda ‘\’
- Ex: ganti baris, tabulasi dll → \n



Tipe Data Standar

- **Boolean**
 - True (1) / False (0)

| Data Type | False Values | True Values |
|----------------|--|--|
| Integer | 0 | All non-zero values |
| Floating point | 0.0 | All non-zero values |
| Strings | Empty strings ("" The zero string ("0") | All other strings |
| Null | Always | Never |
| Array | If it does not contain any elements | If it contains at least one element |
| Object | Never | Always |
| Resource | Never | Always |



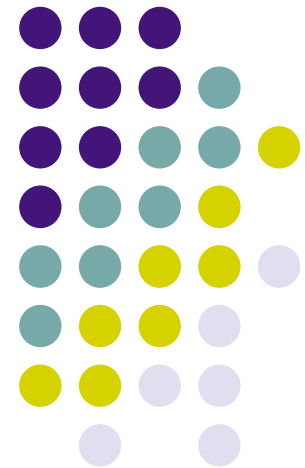
Tipe Data Standar

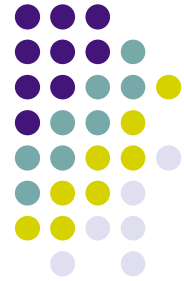
- Array

```
<?php
    /*
        deklarasi ARRAY pada PHP
    */

    $bulan = array('januari', 'februari', 'maret', 'april');
?>
```


Operator

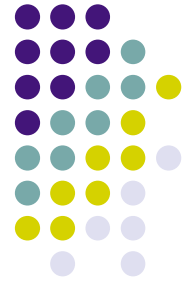




Operator Assignment

- Misal $x = 5$, $y = 3$

| Operator | Contoh | Sama Dengan | Hasil |
|----------|--------------------------------------|-------------------------------|-------------|
| = | <code>\$text = "Belajar PHP";</code> | | Belajar PHP |
| += | <code>\$x += 5;</code> | <code>\$x = \$x + 5;</code> | 10 |
| -= | <code>\$x -= \$y;</code> | <code>\$x = \$x - \$y;</code> | 2 |
| *= | <code>\$x *= \$y;</code> | <code>\$x = \$x * \$y;</code> | 15 |
| /= | <code>\$x /= 2;</code> | <code>\$x = \$x / 2;</code> | 2.5 |
| .= | <code>\$x .= \$y;</code> | <code>\$x = \$x . \$y;</code> | 53 |
| %= | <code>\$x %= \$y;</code> | <code>\$x = \$x % \$y;</code> | 2 |



Operator Numerik

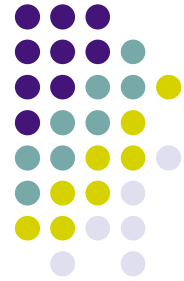
\$x-nya nyambung terus.

| Operator | Deskripsi | Contoh | Hasil |
|----------|-------------------------------------|-------------------|-------|
| + | Penjumlahan (<i>Addition</i>) | $\$x = 1 + 2;$ | 3 |
| - | Pengurangan (<i>Subtraction</i>) | $\$x = \$x - 1;$ | 2 |
| * | Perkalian (<i>Multiplication</i>) | $\$x = 4 * \$x;$ | 8 |
| / | Pembagian (<i>Division</i>) | $\$x = \$x / 2;$ | 4 |
| % | Modulus | $\$x = 9 \% \$x;$ | 1 |
| ++ | Increment | $\$x++;$ | 2 |
| -- | Decrement | $\$x--;$ | 1 |



Operator String

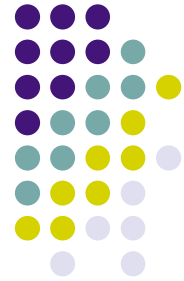
| Operator | Deskripsi | Contoh | Hasil |
|----------|---|-------------------------------------|------------|
| . | Menggabungkan 2 string (<i>Concatenates</i>) | <code>\$x = "Belajar"."PHP";</code> | BelajarPHP |



Operator Pembanding

- Misal \$x = 20

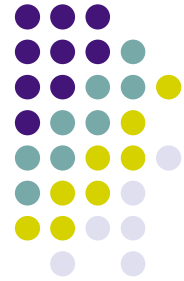
| Operator | Deskripsi | Contoh | Hasil |
|----------|--|--------------|-------|
| == | Sama dengan | \$x == 20 | True |
| === | Sama persis dengan (tipe data dan nilai) | \$x === "20" | False |
| != | Tidak sama dengan | \$x != 20 | False |
| !== | Tidak sama persis dengan | \$x !== "20" | True |
| > | Lebih besar | \$x > 20; | False |
| < | Lebih kecil | \$x < 100; | True |
| >= | Lebih besar dari | \$x >= 20; | True |
| <= | Lebih kecil dari | \$x <= 20; | True |



Operator Logika

- Misal \$x = 6\$ dan \$y = 3\$

| Operator | Deskripsi | Contoh | Hasil |
|-------------------|-------------|---------------------------|-------|
| && | Dan (and) | $(x < 10 \ \&\& \ y > 1)$ | True |
| | Atau (or) | $(x == 5 \ \ y == 3)$ | True |
| ! | Tidak (not) | $!(x == y)$ | True |



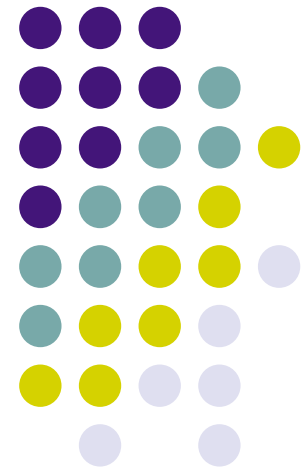
Operator Cast

- Berfungsi untuk memaksa merubah tipe data suatu variabel.
- Misal \$x = "123";

| Operator | Merubah Tipe Ke | Contoh | Hasil |
|----------------------------------|-----------------|--------------|-------------------------------------|
| (int), (integer) | Integer | (int)\$x; | 123 |
| (float), (real), (double) | Floating point | (float)\$x; | 123,00 |
| (string) | String | (string)\$x; | "123" |
| (bool), (boolean) | Boolean | (bool)\$x; | True |
| (array) | Array | (array)\$x; | x[0]="123" |
| (object) | Obejek | (object)\$x; | stdClass Object ([scalar] => 123) |

Standar Output

ECHO, PRINT



ECHO

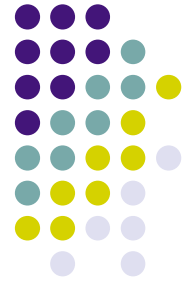


- Data yang dicetak berupa **string** atau **numerik**.
 - `echo ($data);`
 - `echo $data;`
 - `echo ("text");`
 - `echo "text";`
 - `echo "Text-nya adalah : $data";`

PRINT



- Data yang dicetak berupa **string**, **numerik**, atau **objek**.
 - `print ($data);`
 - `print $data;`
 - `print "text";`
 - `print ('text');`
 - `print_r($variabel_array);`



Standar Output

- **Single Quote ‘**
 - Isi quote di-*parsing* oleh PHP per karakter.
 - Lebih hemat memori
- **Double Quote “**
 - Isi quote di-*parsing* oleh PHP seluruhnya, kemudian disimpan atau dicetak.
 - Menggunakan lebih banyak memori dibanding *single quote*.

Saran :

- Gunakan operator *concatenation* “.” untuk mencetak variabel agar hemat memori.
- echo “Nama saya : “.\$nama;



Standar Output - Contoh

```
<html>
<head>
  <title>Contoh Echo dan Print</title>
</head>
<body>
  <?php
    echo "teks ini dengan perintah echo";
  ?>
  <br />
  <?php
    print "teks ini dengan perintah print";
  ?>
  <br />
</body>
</html>
```



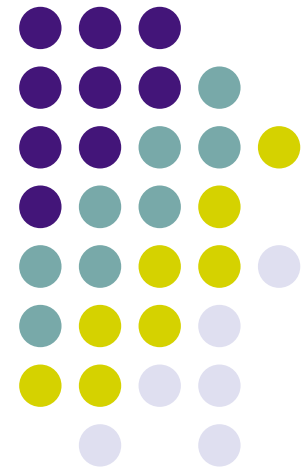
Standar Output – Contoh 2

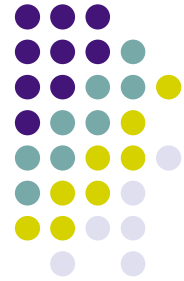
```
<html>
<head>
    <title>Contoh Echo dan Print</title>
</head>
<body>
    <?php
        echo "<h3>CETAK VARIABEL</h3>";

        $a = 1000;

        echo "Isi variabel \$a adalah $a";
        echo "<br />";
        echo 'Isi variabel $a adalah ' . $a;
    ?>
</body>
</html>
```

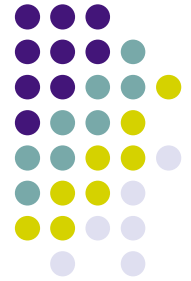
Konstanta





Konstanta

- Aturan penamaan konstanta sama seperti variabel.
- Nama konstanta tidak boleh sama dengan keyword PHP, seperti echo, break, empty, list, dll.
- Hanya dapat dideklarasikan 1 kali.
- Nilainya tidak dapat dirubah.
- Tipe data kecuali array dan objek.
- Case sensitive
- Pendefinisian konstanta
 - `define("nama_konstanta","nilai");`



Konstanta

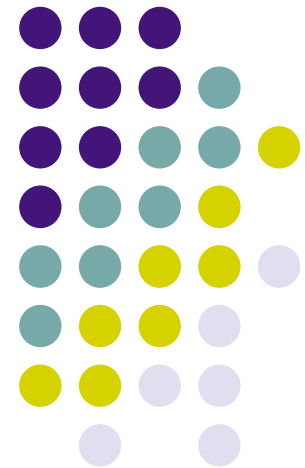
- Contoh

```
define("TEST", "testing konstanta");  
define("TEST", "test2");  
  
echo TEST;
```

- Hasil

testing konstanta

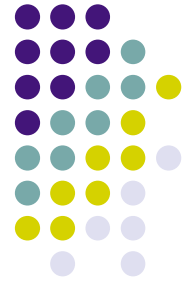
SUPER GLOBALS





Variabel SUPER GLOBAL

- `$_GET` → Variabel HTTP GET.
- `$_POST` → Variabel HTTP POST.
- `$_REQUEST` → Variabel HTTP Request.
- `$_SESSION` → Variabel Session.
- `$_COOKIE` → Variabel HTTP Cookie.
- `$_SERVER` → Informasi konfigurasi lingkungan server.
- `$_FILES` → Variabel HTTP File Upload.



Variabel SUPER GLOBAL

- **\$_SERVER**

- Berisi nilai yang berkaitan dengan informasi server.
- Tipe data array.
- Dokumentasi lengkap : <http://www.php.net/manual/en/reserved.variables.server.php>

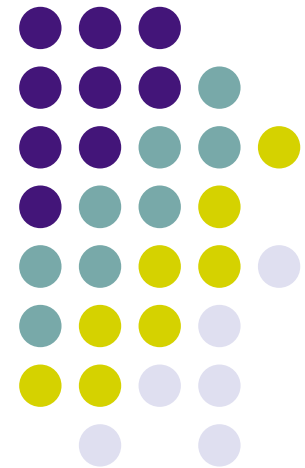
- Contoh

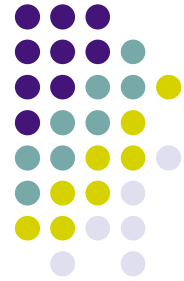
```
echo "Nama Server : " . $_SERVER['SERVER_NAME'] . "<br />";  
echo "IP Pengunjung : " . $_SERVER['REMOTE_ADDR'] . "<br />";  
echo "Script Path : " . $_SERVER['SCRIPT_NAME'];
```

- Hasil

```
Nama Server : localhost  
IP Pengunjung : 127.0.0.1  
Script Path : /kuliah/web2/pertemuan4/predefined_variable.php
```

Struktur Kontrol





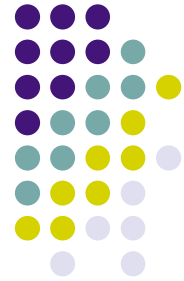
Pengkondisian - IF

- Syntax

```
if (expression)
    statement
```

- Contoh :

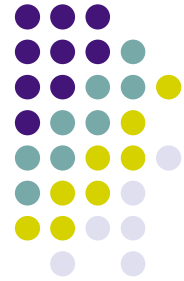
```
if ($userid == "informatika")
{
    echo "benar";
}
```



Pengkondisian – IF ELSE

- Contoh

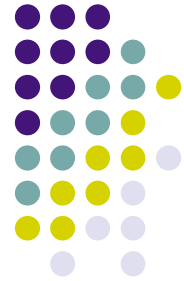
```
if ($userid == "informatika")
{
    echo "benar";
}
else
{
    echo "salah";
}
```



Pengkondisian – IF ELSE IF

- Contoh

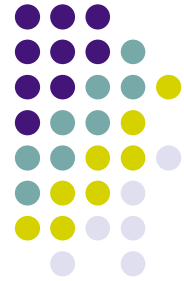
```
$nama_hari = date("\l");  
if ($nama_hari == "Sunday")  
{  
    echo "Minggu";  
}  
elseif ($nama_hari == "Monday")  
{  
    echo "Senin";  
}  
else  
{  
    echo "Selasa";  
}
```



Pengkondisian - SWITCH

- Contoh

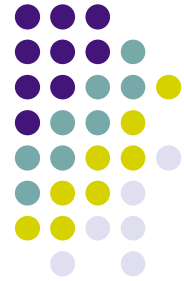
```
$nama_hari = date("l");  
switch ($nama_hari)  
{  
    case "Sunday" : { echo "Minggu"; }  
                    break;  
    case "Monday" : echo "Senin";  
                    break;  
    case "Tuesday" : echo "Selasa";  
                    break;  
    default : echo "Sabtu";  
}
```

Pengkondisian - SWITCH

- Contoh

```
$nama_hari = date("l");  
switch ($nama_hari)  
{  
    case "Monday" :  
    case "Tuesday" :  
    case "Wednesday" : echo "Hari Kerja";  
                        break;  
    case "Saturday" :  
    case "Sunday" : echo "Hari Libur";  
                    break;  
}
```



Ternary

- Menggunakan tanda “?”
- Merupakan model penyederhanaan dari **if...else**
- Cocok untuk melakukan proses pengisian variabel suatu hasil pengujian.

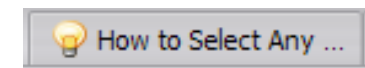
Sintaks

```
namaVariabel = (kondisi)? Nilai1 : Nilai2;
```

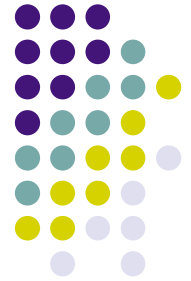
Contoh

```
$greeting = (date("H") <= 12)? "Selamat Pagi" : "Selamat Siang";  
echo $greeting;
```

Hasil

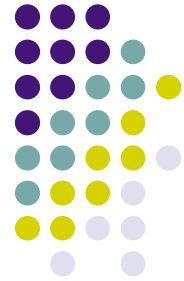


Selamat Siang



Pengulangan

- **Macam-macam pengulangan pada PHP :**
 - for
 - Pengulangan sebanyak jumlah yang ditentukan.
 - while
 - Pengulangan selama pengkondisian menghasilkan nilai “true”.
 - do while
 - Hampir sama dengan while, hanya pengecekan kondisi diakhir tiap pengulangan.
 - foreach
 - Pengulangan sebanyak elemen array



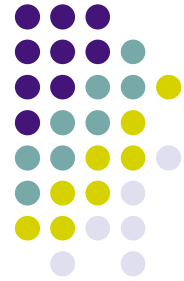
Pengulangan – FOR

- Sintax

```
for ($var=startValue; $var<=endValue; $var=$var+increment)
{
    blok program
}
```

- Contoh

```
echo "<h2>Pengulangan 1 - 10</h2>";
for ($i=1; $i<=10; $i++)
{
    echo $i."<br />";
}
```



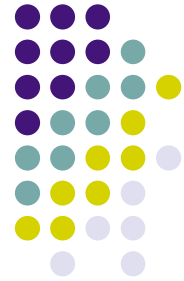
Pengulangan – WHILE

- **Sintax**

```
$var=startValue;  
while ($var<=endValue)  
{  
    source code;  
    $val = $val+increment;  
}
```

- **Contoh**

```
echo "<h2>Pengulangan 1 - 10</h2>";  
$i=1;  
while ($i<=10)  
{  
    echo $i."<br />";  
    $i++;  
}
```



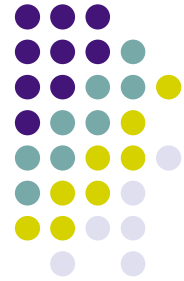
Pengulangan – DO .. WHILE

- **Sintax**

```
$var=startValue;  
do  
{  
    source code;  
    $val = $val+increment;  
} while ($var<=endValue);
```

- **Contoh**

```
echo "<h2>Pengulangan 1 - 10</h2>";  
$i = 1;  
do  
{  
    echo $i."<br />";  
    $i++;  
}  
while ($i<=10);
```



Break Statement

- Perintah ***break*** akan menghentikan pengulangan.

- **Contoh**

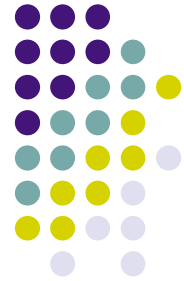
```
echo "<h2>Break Statement</h2>";  
for ($i=1; $i<=5; $i++)  
{  
    if ($i == 3)  
        break;  
  
    echo $i."<br />";  
}  
echo "Akhir pengulangan";
```

Hasil

Break Statement

```
1  
2  
Akhir pengulangan
```

- Pengulangan berhenti ketika variabel ***\$i*** bernilai **3**.



Continue Statement

- Perintah ***continue*** akan melewati *value* pengulangan yang ditentukan, kemudian melanjutkan ke *value* berikutnya.

- Contoh**

```
echo "<h2>Continue Statement</h2>";
for ($i=1; $i<=5; $i++)
{
    if ($i == 3)
        continue;

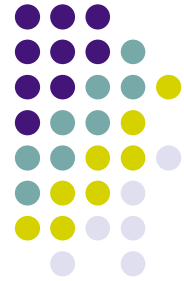
    echo $i."<br />";
}
echo "Akhir pengulangan";
```

Hasil

Continue Statement

```
1
2
4
5
Akhir pengulangan
```

- Ketika variabel ***\$i*** bernilai **3**, blok program dalam pengulangan tidak dieksekusi.



Exit Statement

- Perintah ***exit*** akan menghentikan eksekusi blok program selanjutnya.

- **Contoh**

```
echo "<h2>Exit Statement</h2>";
for ($i=1; $i<=5; $i++)
{
    if ($i == 3)
        exit;

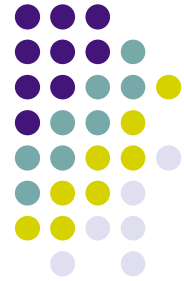
    echo $i."<br />";
}
echo "Akhir pengulangan";
```

Hasil

Exit Statement

1
2

- Baris perintah setelah pengulangan tidak dieksekusi.

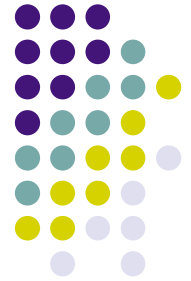


Pengulangan – FOREACH

- Menelusuri seluruh elemen array.
- **Sintax 1**

```
foreach ($nama_array as $value)
{
    blok program;
}
```

- **\$nama_array**
 - Nama variabel bertipe array.
- **\$value**
 - Nama variabel yang menyimpan nilai dari tiap index array.



Pengulangan - FOREACH

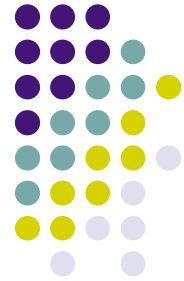
- **Contoh**

```
$ibu_kota = array("Aceh", "Pekanbaru", "Jayapura");

foreach ($ibu_kota as $nama)
{
    echo "Kota : ".$nama."<br />";
}
```

- **Hasil**

```
Kota : Aceh
Kota : Pekanbaru
Kota : Jayapura
```

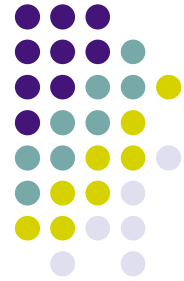


Pengulangan – FOREACH

- Menelusuri seluruh elemen array.
- **Sintax 2**

```
foreach ($nama_array as $key => $value)
{
    blok program;
}
```

- **\$nama_array**
 - Nama variabel bertipe array.
- **\$nama_index**
 - Nama variabel yang menyimpan informasi index array.
- **\$value**
 - Nama variabel yang menyimpan nilai dari tiap index array.



Pengulangan - FOREACH

- **Contoh**

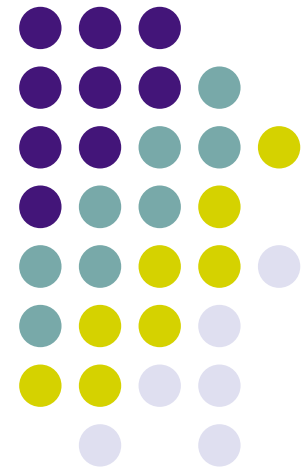
```
$ibu_kota = array("Aceh", "Pekanbaru", "Jayapura");

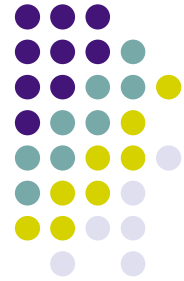
foreach ($ibu_kota as $kota => $nama)
{
    echo "Kota ke-".$kota." : ".$nama."<br />";
}
```

- **Hasil**

```
Kota ke-0 : Aceh
Kota ke-1 : Pekanbaru
Kota ke-2 : Jayapura
```

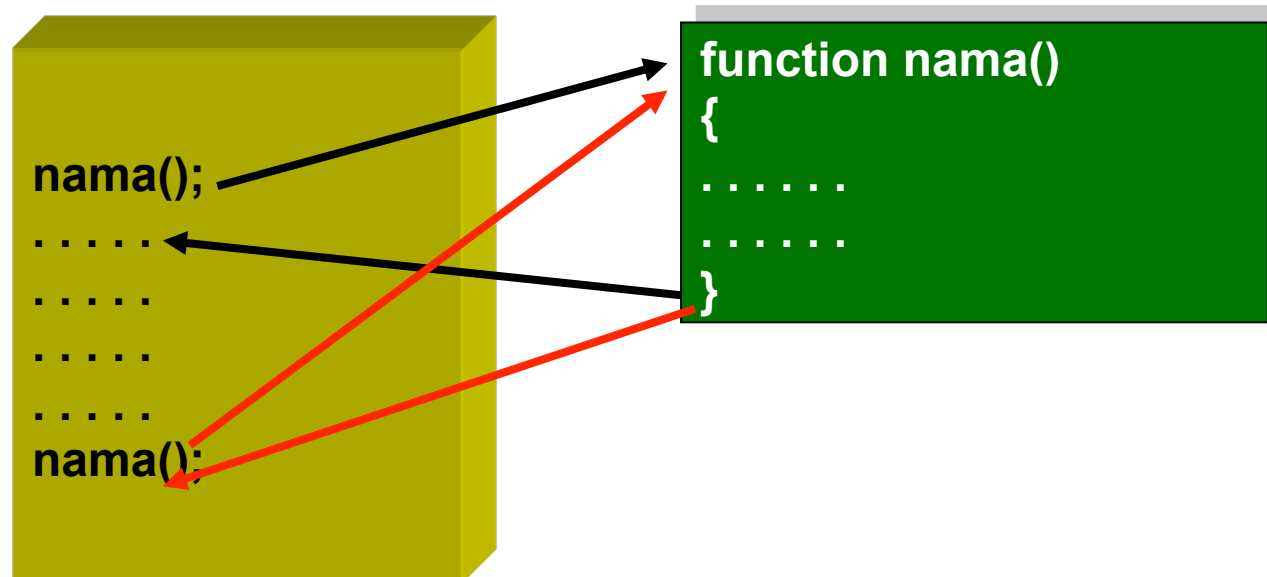
Fungsi





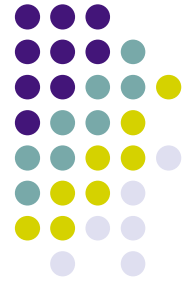
Pengertian Fungsi

- Blok kode yang ditujukan untuk melaksanakan tugas tertentu
- Fungsi dapat dipanggil berkali-kali
- Memudahkan pelacakan kesalahan

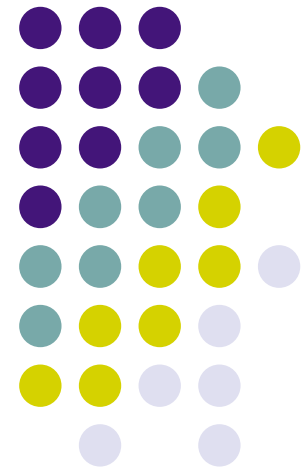


Fungsi

- User Defined Function
- Built-in Function



User Defined Function





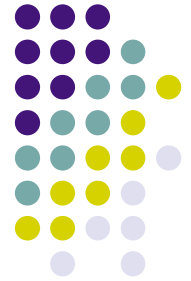
UDF

- **Syntax :**

```
function nama_fungsi([parameter])  
{  
    statement . . . ;  
    statement . . . ;  
}
```

- **Contoh**

```
function tulis_tebal($teks)  
{  
    echo "<b>".$teks."</b>";  
}
```



UDF

- Fungsi dengan **return value**.
- **Contoh**

```
function selisih($a, $b)
{
    if ($a >= $b)
        return $a - $b;
    else
        return $b - $a;
}
```



UDF

- Sintaks fungsi tanpa return value :

```
nama_fungsi([parameter]);
```

- Sintaks fungsi dengan return value :

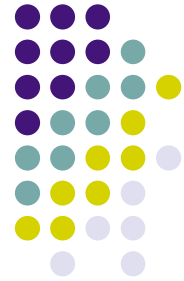
```
$var_hasil = nama_fungsi([parameter]);
```



Parameter Default

- **Contoh**

```
function tulis_teks($teks="", $jumlah = 1)
{
    for ($i=1; $i <= $jumlah; $i++)
        echo $teks;
}
tulis_teks("A", 5);
echo "<br />";
tulis_teks("F");
```



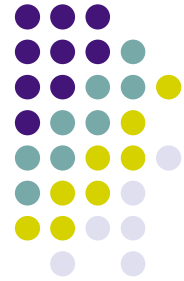
Scope / Lingkup variabel

- Contoh

```
function ganti()  
{  
    $bunga = "Mawar";  
}  
$bunga = "Melati";  
ganti();  
echo $bunga;
```

- Hasil

- Teks yang tampil **Melati**. Karena pengisian **Mawar** ke dalam variabel \$bunga di dalam fungsi ganti() bersifat variabel lokal.



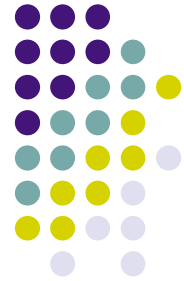
Scope / Lingkup variabel

- Contoh

```
function ganti()  
{  
    global $bunga;  
    $bunga = "Mawar";  
}  
$bunga = "Melati";  
ganti();  
echo $bunga;
```

- Hasil

- Teks yang tampil **Mawar**. Karena variabel **\$bunga** di dalam fungsi **ganti()** merupakan variabel global.



Scope / Lingkup variabel

- Contoh

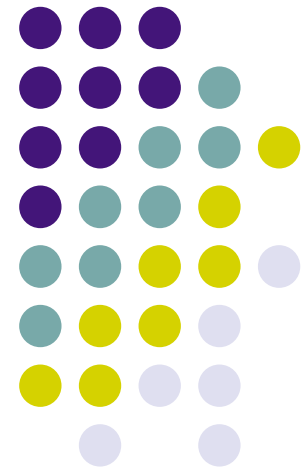
```
function ganti()  
{  
    $GLOBALS['bunga'] = "Mawar";  
}  
$bunga = "Melati";  
ganti();  
echo $bunga;
```

- Hasil

- Teks yang tampil **Mawar**. Karena **Mawar** diisikan ke variabel **global** milik PHP. (lihat slide pertemuan 3 – Predefined Variable)

Fungsi Built-in

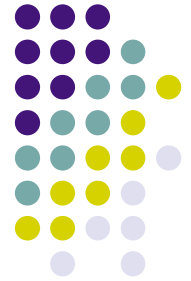
Fungsi-fungsi bantuan yang ada di
PHP





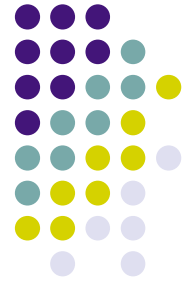
Date

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



String

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

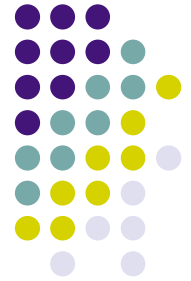


Math

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

Utility

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





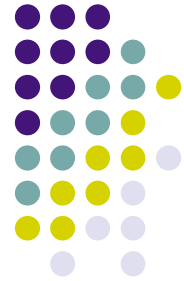
Tugas 2

- Buat halaman php yang mengimplementasikan tiap-tiap fungsi built-in yang sebelumnya disebutkan
- 1 halaman untuk tiap kategori (date, string, math dan utility)
- Total jadi 4 halaman (date.php, string.php, math.php dan utility.php)
- Contoh kasus untuk tiap-tiap fungsi bebas
- BERI KOMENTAR untuk menjelaskan penggunaan tiap-tiap fungsi



Tugas 2

- Sebelum pertemuan selanjutnya (sesi pertama, bukan praktikum)
- Kirim ke email dosen & asisten masing-masing
- Subject: pw2–t2–kelas–nrp
- Simpan semua file php di dalam folder dengan nama nrp-tugas2
- Lalu kompres (zip/rar) dengan nama yang sama



Referensi

- Candra Utama, “Pemrograman Web 2”. Teknik Informatika Universitas Pasundan Bandung. 2012
- Arief H. Suardi, “*Pemrograman Web*”. Teknik Informatika Universitas Pasundan Bandung.
- Gutmans, Andi, Stig Saether Bakken, Derick Rethans, “*PHP 5 Power Programming*”, Prentice Hall, 2005
- Valade, Janet, “*PHP 5 For Dummies*”, Wiley Publishing, 2004
- www.w3school.com
- www.php.net

Terima Kasih

Pemrograman Web 2 - 20132014

sandhikagalih@unpas.ac.id

erik@unpas.ac.id

acep.hendra@unpas.ac.id

