



# Pertemuan 9 : Server Side Scripting

---

## Pemrograman Web

Agung Nugroho, M.Kom

Teknik Informatika – S1

Fakultas Teknik

Universitas Pelita Bangsa



# Agung Nugroho, S.Kom, M.Kom

- 1994 | SDN Pulau Panggung, OKU Sumsel
- 1997 | MTs Lab Fak Tarbiah IAIN SUKA, Yogyakarta
- 2000 | SMK PIRI 1, Yogyakarta
- 2004 | Ilmu Komputer, Universitas Ahmad Dahlan, Yogyakarta
- 2016 | Magister Komputer, STMIK Eresha, Jakarta

- 2012 - Present | Freelance Web Developer
- 2011 - 2012 | Web Developer at BP Indonesia
- 2010 - 2011 | OSS Core Engineer at PT Ericsson Indonesia
- 2008 - 2009 | Radio Database Planner at PT. NextWave subcon NSN
- 2005 - 2008 | Software Developer at PT Gamatechno Indonesia
- 2004 - 2005 | Web Programmer at PT Reftindo Sarana



- [www.linkedin.com/in/kangmasagung](http://www.linkedin.com/in/kangmasagung)
- [www.fb.me/agung.n](http://www.fb.me/agung.n)
- [www.koding.web.id](http://www.koding.web.id)



# Server Side Scripting

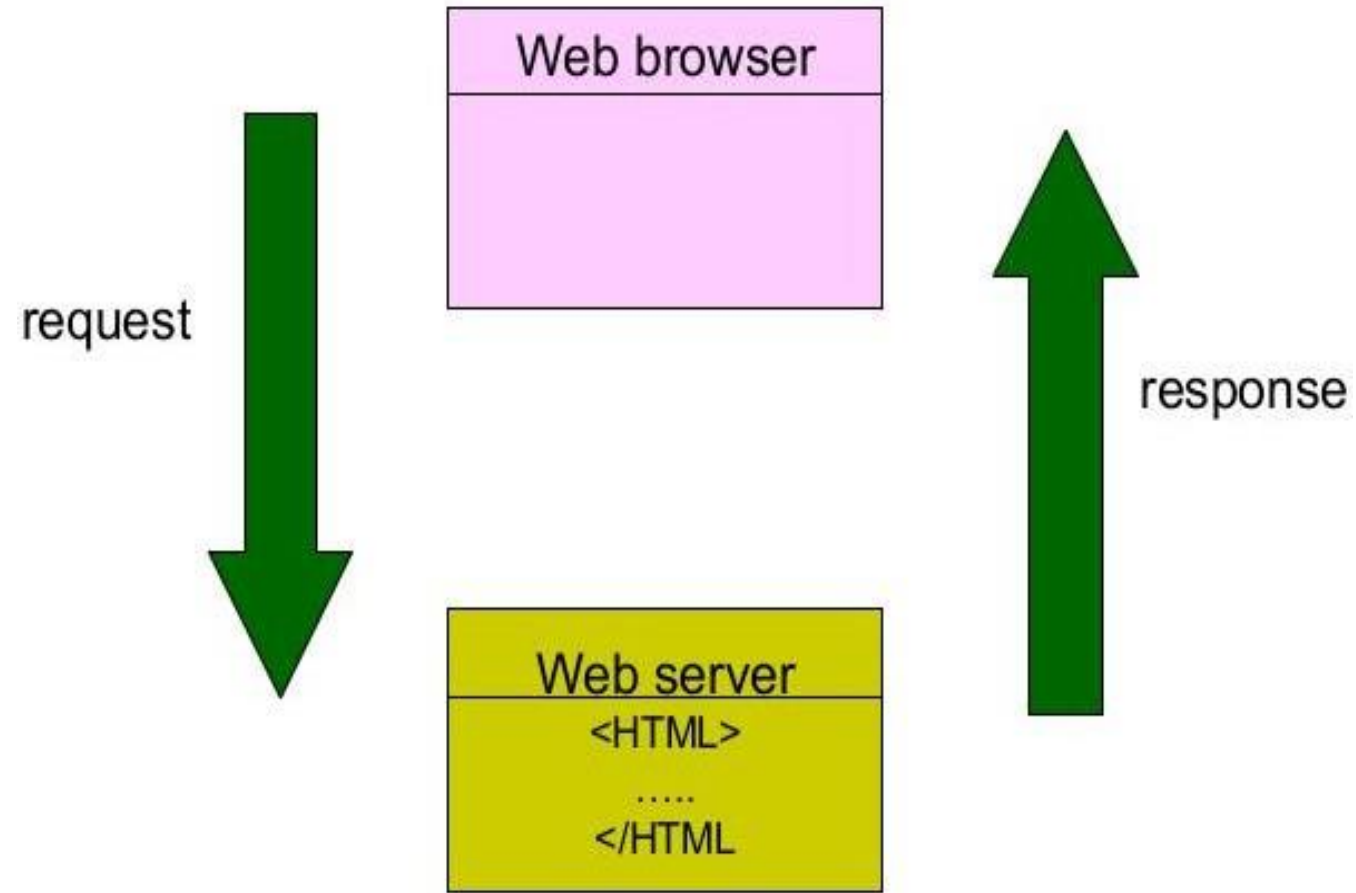
## Pertemuan 9



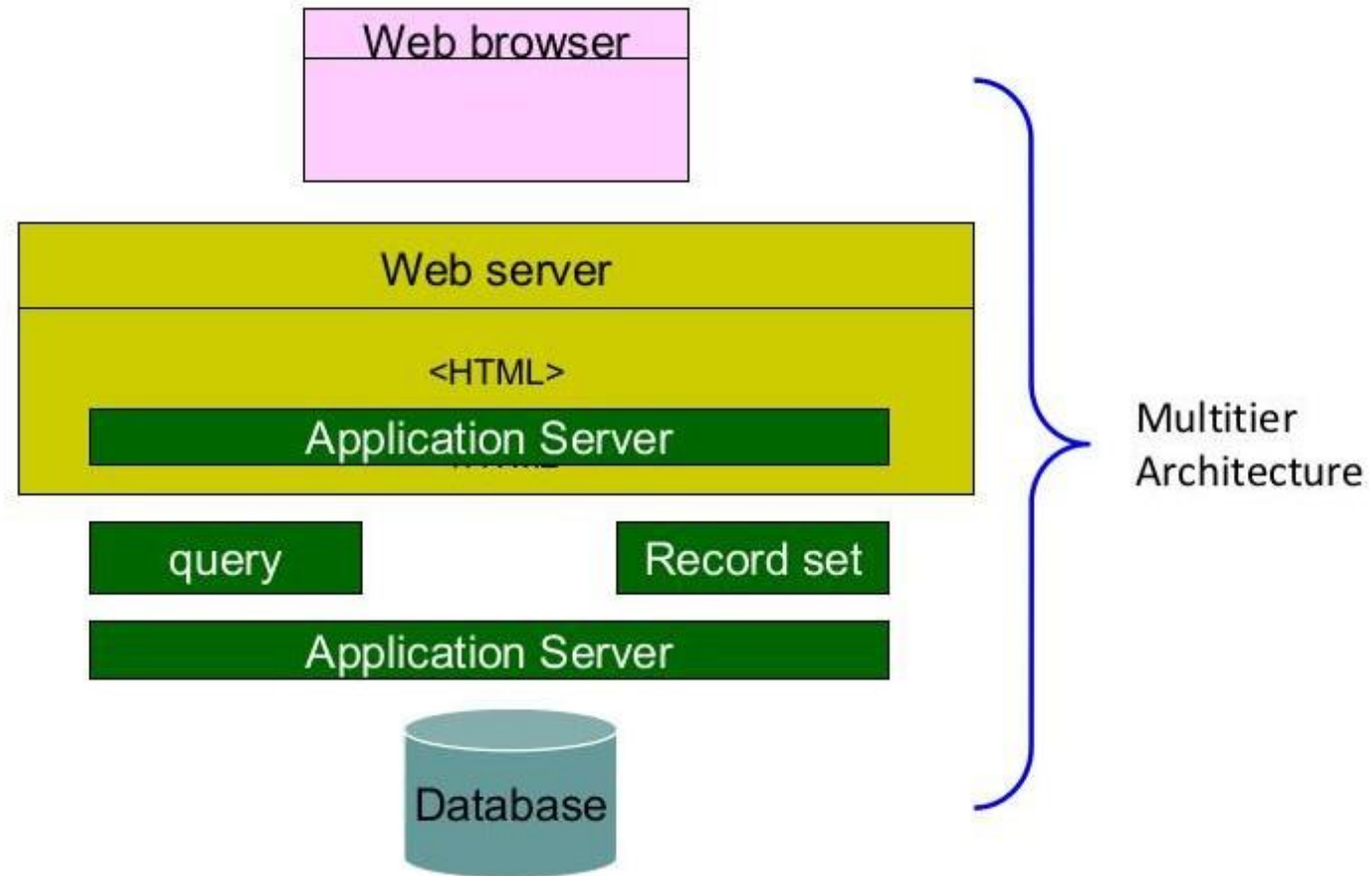
# Jenis Web

- **Static Website**, situs web yang isinya disimpan dalam sebuah file statis (HTML).
- **Dynamic Website**, situs web yang isi dan tampilannya disimpan dalam suatu database.

# Static Website



# Dynamic Website





# Multitier Architecture

- Apa itu multitier architecture?
  - Dalam metode pengembangan aplikasi, Multi-tier Architecture (atau dikenal N-tier Architecture) adalah arsitektur klien – server yang mana tingkat / tier presentasi (tampilan interaksi dengan user), aplikasi pemroses data, dan manajemen data secara logika merupakan proses-proses yang berbeda.
- Keunggulan
  - Pengembang lebih leluasa dalam membuat aplikasi yang fleksibel dan dapat digunakan kembali (reuseable application). Modifikasi hanya dilakukan pada bagian yang diinginkan, tidak keseluruhan.
- Yang paling banyak digunakan adalah ***Three-tier Architecture***



# Three-tier Architecture

- Presentation Tier
  - Tingkatan paling atas dari aplikasi. Memberikan informasi kepada pengguna aplikasi. Berinteraksi dengan tingkatan lainnya, contohnya mengirimkan permintaan user kemudian menampilkan datanya.
- Application Tier
  - *Middle tier, Business Logic*, operasi logika, akses dan proses data. Tingkatan ini memiliki fungsi untuk mengontrol fungsionalitas aplikasi dan melakukan proses yang mendetil.
- Data Tier
  - Tingkat ini mengacu ke Database Server. Pengelolaan data yang independen seperti ini, meningkatkan skalabilitas dan performa aplikasi.





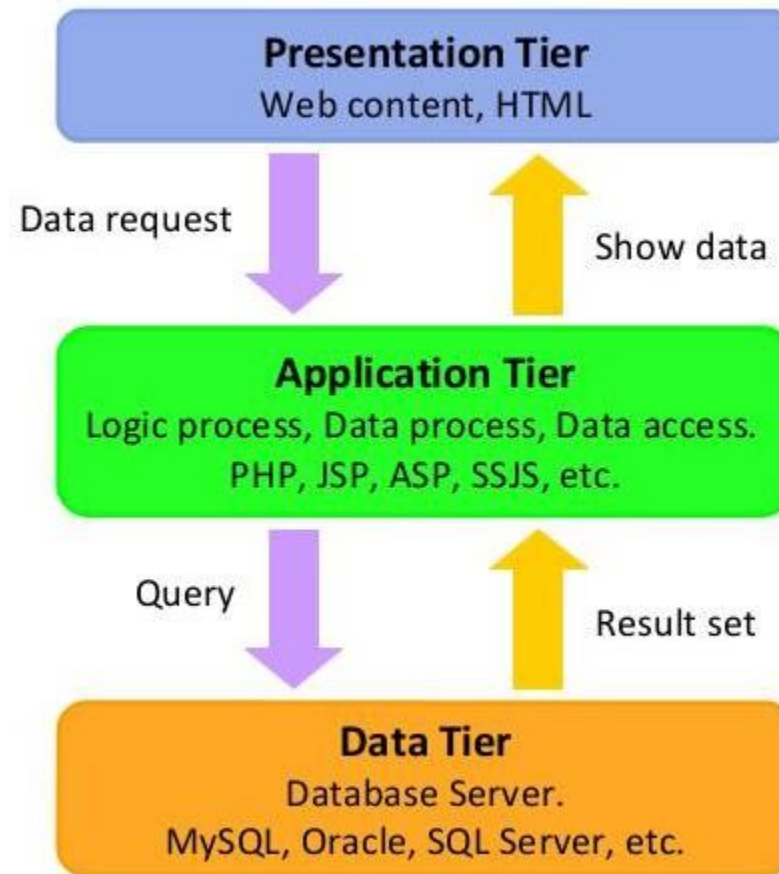
# Three-tier Architecture

Penggunaan pada pengembangan website

- Presentation Tier (*Front-end*)
  - Konten yang di-generate oleh browser.
- Application Tier (*Server Side Script*)
  - Server Side Scripting, seperti PHP, JSP, ASP, SSJS (Server Side JS), dll.
- Data Tier (*Back-end*)
  - Database server.



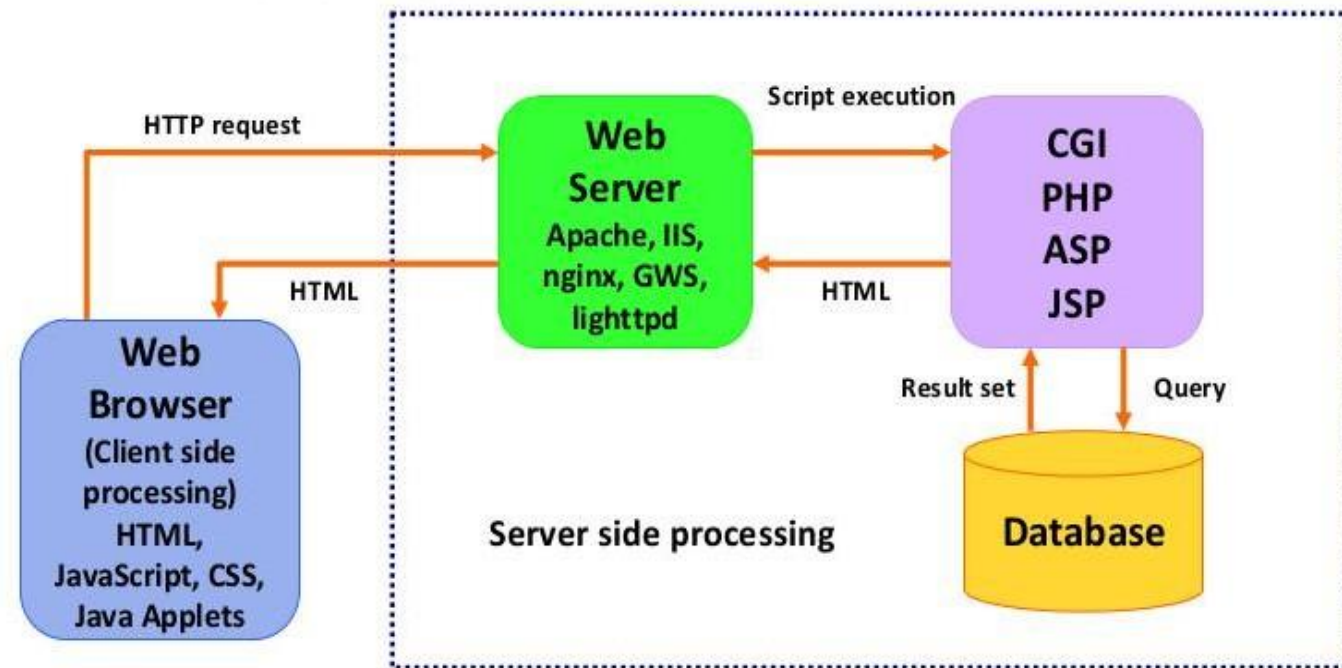
# Three-tier Architecture



# Server Side Scripting

Apa itu *Server Side Scripting*?

- Script yang dieksekusi pada web server.
- Biasa digunakan untuk website dinamis yang menggunakan database atau media penyimpanan data lainnya.





# Web Server

Apa itu web server ?

- Sebuah program komputer yang mengirimkan konten ke client seperti halaman web, menggunakan protokol HTTP melalui World Wide Web.
- Web server juga dapat diartikan pada mesin komputer atau mesin virtual (virtual machine) yang menjalankan aplikasi web server.

Vendor	Product	Web Sites Hosted (millions)	Percent
Apache	Apache	111	54%
Microsoft	IIS	50	24%
Igor Sysoev	nginx	16	8%
Google	GWS	15	7%
lighttpd	lighttpd	1	0.46%

Sumber : Netcraft survey, January 2010



# Server Side Scripting - PHP

- **PHP: Hypertext Preprocessor**, bahasa scripting yang didesain untuk pengembangan website dinamis.
- Dibuat oleh Rasmus Lerdorf, tahun 1995. Saat itu masih bernama FI (*Form Interpreted*)
- PHP adalah bahasa skrip open source yang banyak digunakan
- Skrip PHP dijalankan di server
- PHP gratis untuk diunduh dan digunakan
- Tag php dimulai dengan **<?php** dan diakhiri dengan **?>**





# Apa itu File PHP?

- File PHP dapat berisi teks, HTML, CSS, JavaScript, dan kode PHP
- Kode PHP dijalankan di server, dan hasilnya dikembalikan ke browser sebagai HTML biasa
- File PHP memiliki ekstensi "**.php**"



# Sintaks PHP Dasar

- Skrip PHP dapat ditempatkan di mana saja dalam dokumen.
- Skrip PHP dimulai dengan **<?php** dan diakhiri dengan **?>**

```
<?php  
// kode PHP disini  
?>
```

- Ekstensi file default untuk file PHP adalah **".php"**.
- File PHP biasanya berisi tag HTML, dan beberapa kode skrip PHP.



# Embedded Script

- Script PHP disisipkan diantara tag HTML

```
<html>
  <body>
    <?php
      echo "Hello world";
      echo "Sedang belajar PHP";
    ?>
  </body>
</html>
```



# Non Embedded Script

- Dalam satu dokumen, murni script PHP.
- Tag HTML merupakan bagian dari script PHP

```
<?php
    echo "<html>
        <body>
            Hello world
            Sedang belajar PHP
        </body>
    </html>";
?>
```



# Komentar di PHP

- Komentar dalam kode PHP adalah baris yang tidak dieksekusi sebagai bagian dari program. Tujuan satu-satunya adalah untuk dibaca oleh seseorang yang melihat kode tersebut.
- *Single-line Commentor*, menggunakan karakter “**//**”

```
<?php
// ini adalah komentar
?>
```

- *Multi-line Commentor*, menggunakan karakter “**/\***” dan “**\*/**”

```
<?php
/* ini adalah blok komentar
 * dapat dibuat lebih dari satu baris komentar */
?>
```





# Variable PHP

- Diawali dengan karakter \$
- Nama variabel bisa terdiri dari karakter huruf, angka, dan underscore.
- Setelah karakter \$, harus diikuti karakter huruf atau underscore “\_”.
- *Case Sensitive*

```
$_name  
$first_name  
$name3  
$lastName  
global $_name → variabel global
```



# Indirect References Variable

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

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



# Predefined Variable

- Nama variabel yang sudah dipakai oleh PHP.
- Beberapa Predefined Variable:

```
$GLOBAL → Mengacu ke semua variabel global.  
$_SERVER → Informasi konfigurasi lingkungan server.  
$_GET → Variabel HTTP GET.  
$_POST → Variabel HTTP POST.  
$_FILES → Variabel HTTP File Upload.  
$_REQUEST → Variabel HTTP Request.  
$_SESSION → Variabel Session.  
$_COOKIE → Variabel HTTP Cookie.  
$php_errormsg → Pesan error terakhir.  
$http_response_header → Header respon dari HTTP request.
```



# Predefined Variable

## Variable **\$GLOBAL**

- Mengacu ke variabel global pada suatu script.
- Tipe data array.

```
<?php
function test() {
    $foo = "local variable";

    echo '$foo in global scope: ' . $GLOBALS["foo"] . "\n";
    echo '$foo in current scope: ' . $foo . "\n";
}

$foo = "Example content";
test();
?>
```



# Predefined Variable

## Variable **`$_SERVER`**

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

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

Nama Server: localhost

IP Address: 127.0.0.1

Script Path: /kuliah/predefined.php





# Predefined Variable

## Variable `$_GET`

- Variabel yang berasal dari parameter URL.
- Tipe data array.

## Contoh

- Buat satu file dengan nama predefined\_get.php

```
echo 'Selamat Datang ' . $_GET['nama'];|
```

- Akses file tersebut di browser, tambahkan parameter [http://localhost/predefined\\_get.php?name=Agung](http://localhost/predefined_get.php?name=Agung)
  - Lokasi file disesuaikan dengan tempat penyimpanan.



# Predefined Variable

## Variable `$_POST`

- Variabel yang berasal dari HTTP POST.
- Tipe data array.
- Penerapan pada HTML Form
- Value properti “name” elemen input, menjadi index array `$_POST`

**Contoh**, file : form.php

```
<html>
  <body>
    <form method="post">
      <label>Nama </label>
      <input type="text" name="nama" /><br />
      <input type="submit" value="Kirim" />
    </form>
  </body>
</html>
<?php
echo 'Selamat Datang ' . $_POST['nama'];
?>
```

Nama

Selamat Datang agung



# 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



# Type 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

- **NULL**

- Hanya dapat berisi nilai NULL. Menandakan variabel kosong. Contoh :  
\$bilangan = NULL;

- **Resource**

- PHP extension resource, seperti hasil query, open file, koneksi database, dll.

- **Array**



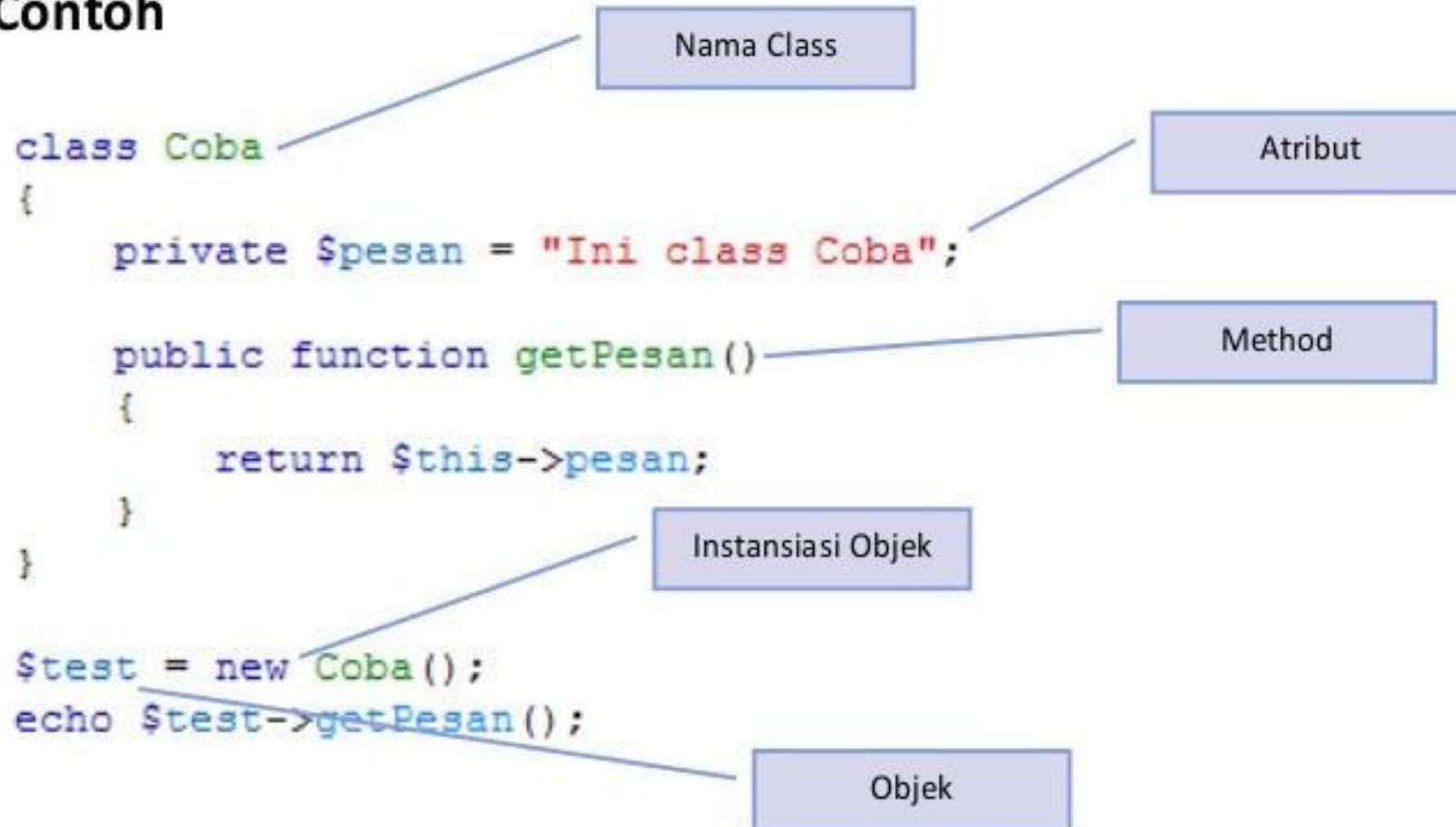


# Tipe Data Objek

- Dukungan terhadap OOP.
- Memiliki Data dan Method.
- Data yang dipunyai oleh suatu objek, populer dengan nama atribut.
- Method suatu objek umumnya berupa fungsi.
- Pendefinisian Class.  
*class NamaClass*
- Suatu variabel yang bertipe objek, diinisialisasi dengan perintah NEW
- kemudian NAMA OBJEK (berupa nama kelas objek)

# Type Data Object

## Contoh





# Escape Character

<code>\n</code>	Linefeed (LF or 0x0A in ASCII)
<code>\r</code>	Carriage return (CR or 0x0D in ASCII)
<code>\t</code>	Horizontal TAB (HT or 0x09 in ASCII)
<code>\\</code>	Backslash
<code>\\$</code>	Dollar sign
<code>\"</code>	Double-quote
<code>\</code>	[0-7] {1,3} the sequence of character matching the regular expression is a character in octal notation
<code>\x</code>	[0-9 A-F a-f] {1,2} the sequence of character matching the regular expression is a character in hexadecimal notation



# Operator pada PHP



# 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>	20
/=	<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

Operator	Deskripsi	Contoh	Hasil
+	Penjumlahan ( <i>Addition</i> )	$\$x = 1 + 2;$	3
-	Pengurangan ( <i>Substraction</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 Pembandingan

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
<b>&amp;&amp;</b>	Dan (and)	$(x < 10 \ \&\& \ y > 1)$	True
<b>  </b>	Atau (or)	$(x == 5 \    \ y == 3)$	True
<b>!</b>	Tidak (not)	$!(x == y)$	True



# Operator Cast

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

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



## 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;
```



# 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);
```



# 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");
```



# Flow Control



# Kondisi IF

- Syntax

```
if (expression)  
    statement
```

- Contoh:

```
if ($jurusan == 'informatika')  
    echo "benar";
```





# Kondisi IF ELSE

- Syntax

```
if (expression)
    statement true
else
    statement false
```

- Contoh:

```
if ($jurusan == 'informatika')
    echo "benar";
else
    echo "salah";
```



# Kondisi IF ELSE IF

- Syntax

```
if (expression)
    statement1
elseif (expression2)
    statement2
```

- Contoh

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

# Kondisi SWITCH

- Syntax

```
switch (var) {  
    case value1:  
        statement1  
        break;  
    case value2:  
        statement2  
        break;  
    case valueN:  
        statementN  
        break;  
    default:  
        statement default;  
}
```

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



## Ternary

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

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



# Perulangan

## Macan-macam Perulangan

- **while - loop** → melalui blok kode selama kondisi yang ditentukan benar
- **do...while** → mengulang melalui satu blok kode satu kali, dan kemudian mengulang perulangan selama kondisi yang ditentukan benar
- **for - loop** → melalui blok kode beberapa kali
- **foreach - loop** → melalui blok kode untuk setiap elemen dalam array



# Perulangan FOR

- Syntax

```
for ($var=nilaiAwal; $var<=nilaiAkhir; $var_increment)
{
    block statement;
}
```

***\$var\_increment***: penambahan atau pengurangan (\$var++ / \$var--)

- Contoh

```
echo "Perulangan 1 sampai 10 <br />";
for ($i=1; $i<=10; $i++) {
    echo "Perulangan ke: " . $i . ' <br />';
}
```



# Perulangan WHILE

- Syntax

```
$var = nilaiAwal;  
while ($var<=nilaiAkhir) {  
    Block statement;  
    $var_increment;  
}
```

- Contoh

```
echo "Perulangan 1 sampai 10 <br />";  
$i=1;  
while ($i<=10) {  
    echo "Perulangan ke: " . $i . '<br />';  
    $i++;  
}
```



# Perulangan DO .. WHILE

- Syntax

```
$var = nilaiAwal;  
do {  
    Block statement;  
    $var_increment;  
} while ($var<=nilaiAkhir);
```

- Contoh

```
echo "Perulangan 1 sampai 10 <br />";  
$i=1;  
do {  
    echo "Perulangan ke: " . $i . '<br />';  
    $i++;  
} while ($i<=10);
```





## Break Statement

- Perintah break akan menghentikan pengulangan.
- Contoh

```
echo "Perulangan 1 sampai 10 <br />";  
for ($i=1; $i<=10; $i++) {  
    if ($i == 3)  
        break;  
    echo "Perulangan ke: " . $i . '<br />';  
}  
echo "Akhir Perulangan";
```

Perulangan 1 sampai 10  
Perulangan ke: 1  
Perulangan ke: 2  
Akhir Perulangan

Pengulangan berhenti ketika variabel \$i bernilai 3.



## Continue Statement

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

```
echo "Perulangan 1 sampai 10 <br />";  
for ($i=1; $i<=10; $i++) {  
    if ($i == 3)  
        continue;  
    echo "Perulangan ke: " . $i . '<br />';  
}  
echo "Akhir Perulangan";
```

Perulangan 1 sampai 10  
Perulangan ke: 1  
Perulangan ke: 2  
Perulangan ke: 4  
Perulangan ke: 5  
Akhir Perulangan

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



# Perulangan FOREACH

- Menelusuri seluruh elemen array.
- Syntax

```
foreach($var_array as $index => $value) {  
    Block program;  
}
```

\$var\_array: Nama variabel array

\$index: nama variabel yang menyimpan informasi index array

\$value: nama variabel yang menyimpan nilai dari tiap index array



# Perulangan FOREACH

- Contoh

```
$kota = array('Jakarta', 'Bandung', 'Semarang');  
foreach ($kota as $index => $nama) {  
    echo 'Kota ke-' . $index . ': ' . $nama . '<br />';  
}
```

Kota ke-0: Jakarta

Kota ke-1: Bandung

Kota ke-2: Semarang



Question?



# CodeLabs | Experiment



# Persiapan

## Tools:

- **Text Editor:** Sublime Text, VS Code, Atom, Dll
- **Web Browser:** Google Chrome, Mozilla Firefox, Dll
- **Web Server:** Apache, IIS, NGINX, Dll





# Terimakasih

---

Agung Nugroho

[agung@pelitabangsa.ac.id](mailto:agung@pelitabangsa.ac.id)

[www.koding.web.id](http://www.koding.web.id)