



Web Dinamis: Pengantar PHP

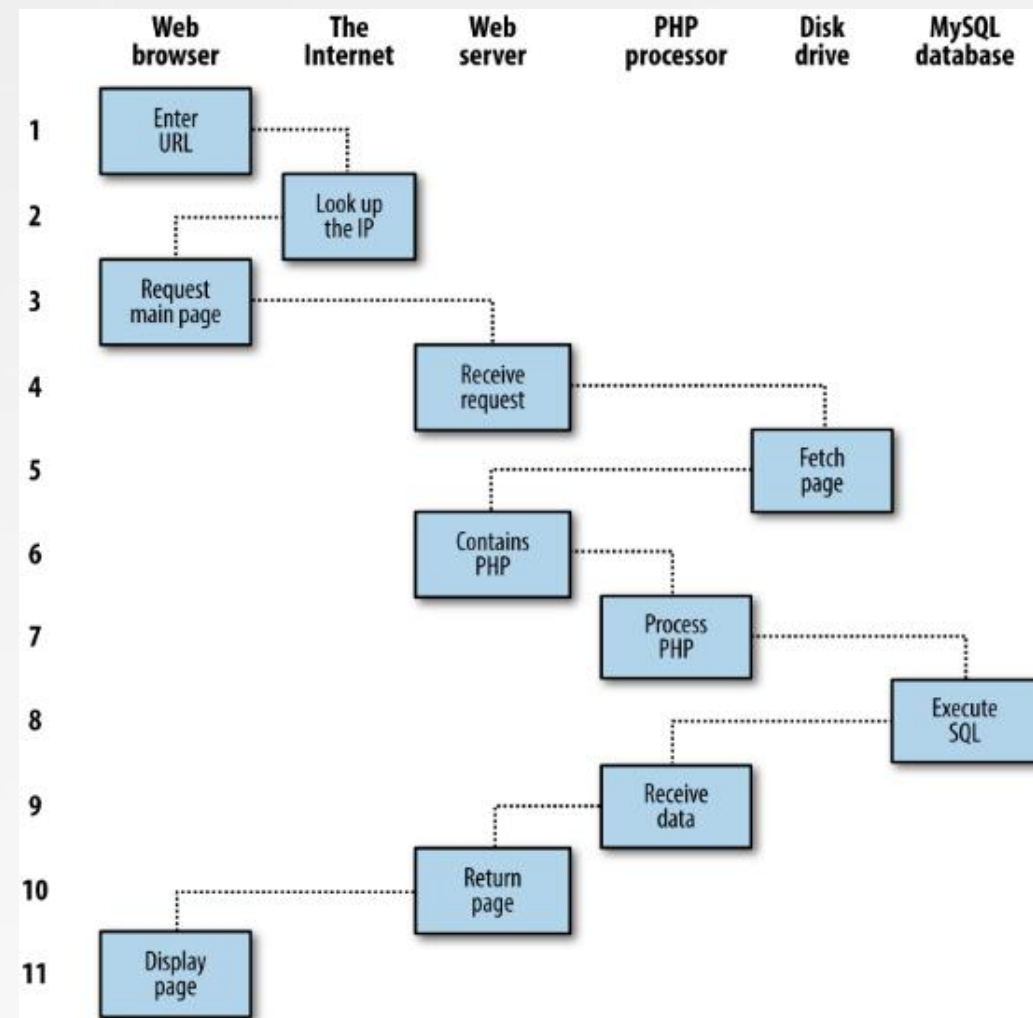
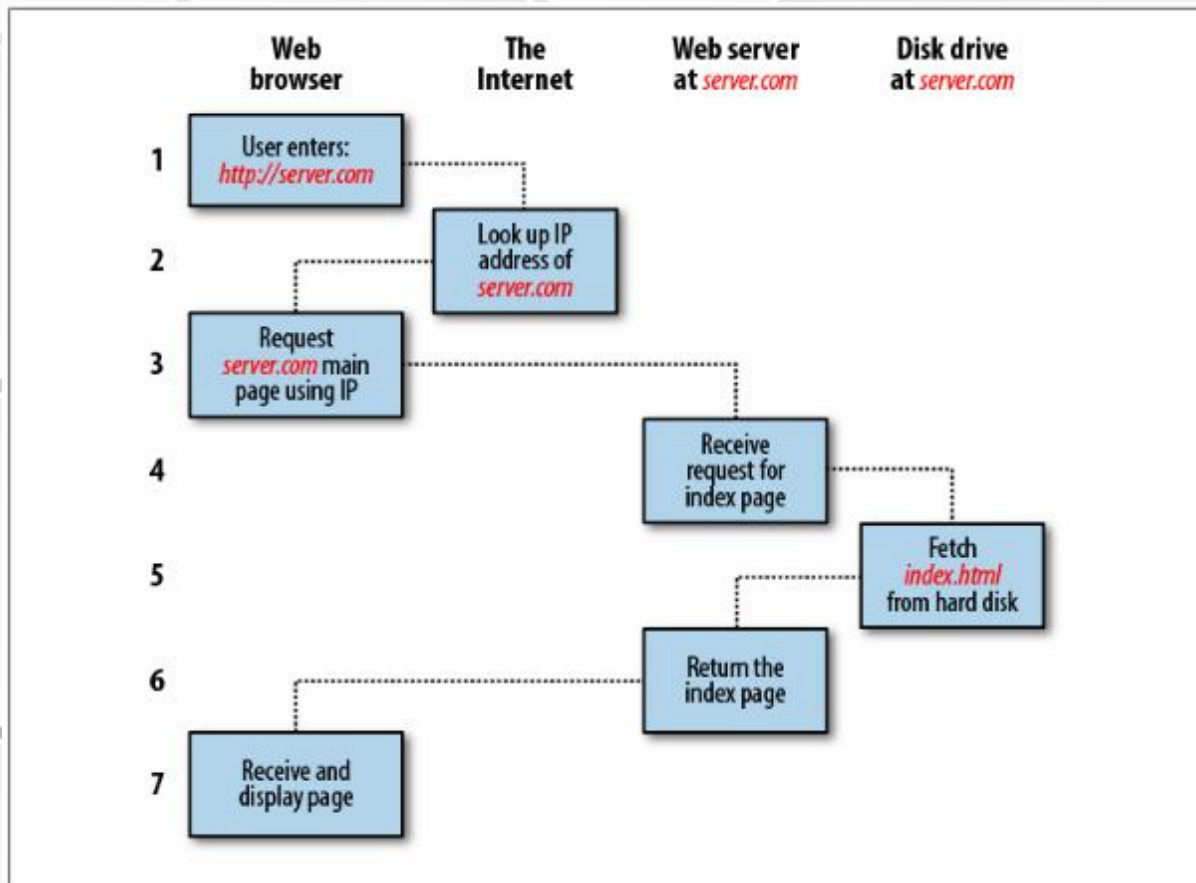
adesutedi@sttgarut.ac.id



Perbedaan Web Statis dan Dinamis

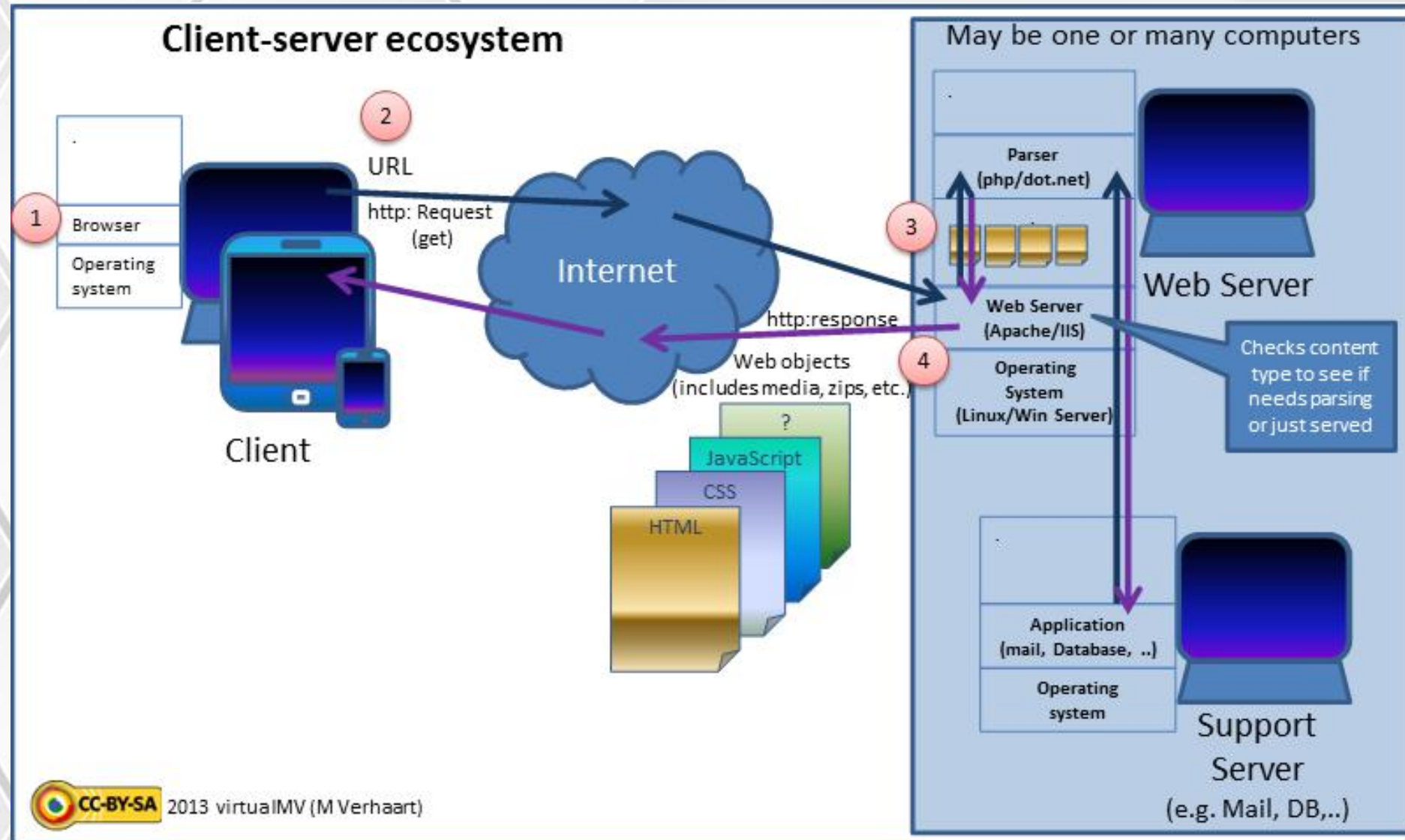
Static	Dynamic
Prebuilt content is same every time the page is loaded.	Content is generated quickly and changes regularly.
It uses the HTML code for developing a website.	It uses the server side languages such as PHP, SERVLET, JSP, and ASP.NET etc. for developing a website.
It sends exactly the same response for every request.	It may generate different HTML for each of the request.
The content is only changed when someone publishes and updates the file (sends it to the web server).	The page contains "server-side" code which allows the server to generate the unique content when the page is loaded.
Flexibility is the main advantage of static website.	Content Management System (CMS) is the main advantage of dynamic website.

Web Statis dan Dinamis



Web Dinamis

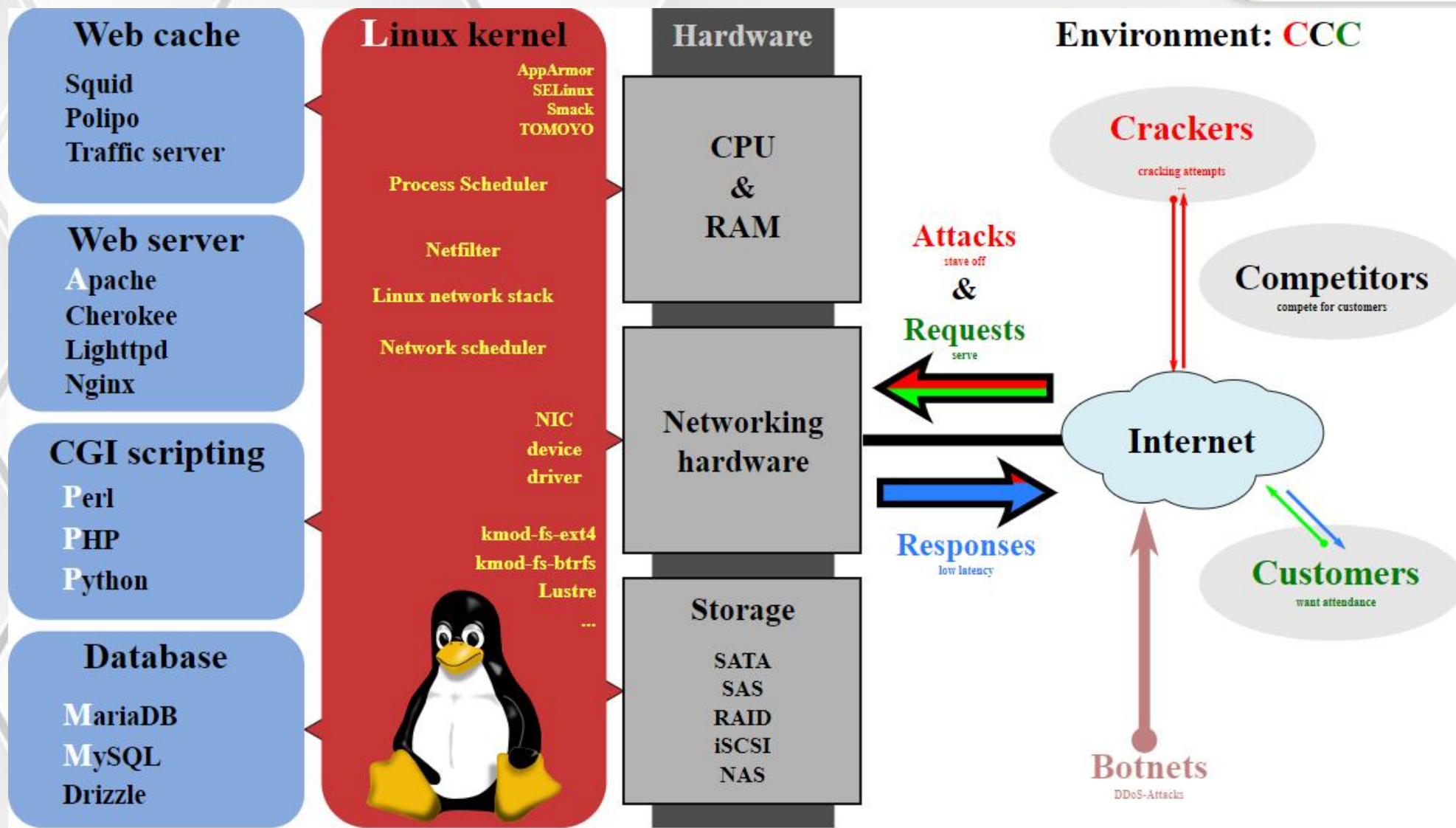
- Laman **web dinamis sisi server** adalah laman web yang konstruksinya dikontrol oleh skrip sisi server yang memproses aplikasi server.
- Dalam pembuatan skrip sisi server, parameter menentukan bagaimana proses perakitan setiap halaman web baru, termasuk pengaturan pemrosesan sisi klien lainnya.
- Halaman **web dinamis sisi klien** memproses halaman web menggunakan skrip HTML yang berjalan di browser saat dimuat.
- JavaScript dan bahasa skrip lainnya menentukan cara HTML di halaman yang diterima diurai menjadi Model Objek Dokumen, atau DOM, yang mewakili halaman web yang dimuat.
- Teknik sisi klien yang sama kemudian dapat memperbarui atau mengubah DOM secara dinamis dengan cara yang sama.



Web Server

- **Web Server** adalah perangkat lunak server, atau sistem dari satu atau lebih komputer yang didedikasikan untuk menjalankan perangkat lunak yang dapat memenuhi permintaan HTTP klien di WWW publik atau juga di LAN dan WAN pribadi.

Product	Vendor	Percent
Apache	Apache	34.5%
nginx	NGINX, Inc.	33.7%
Cloudflare Server	Cloudflare, Inc.	17.6%
LiteSpeed Web Server	LiteSpeed Technologies	8.0%
IIS	Microsoft	7.2%
GWS	Google	1.2%
Node.js	OpenJS Foundation	1.1%



PHP: Hypertext Preprocessor

- PHP adalah bahasa *scripting* untuk keperluan pengembangan web. awalnya dibuat oleh programmer Denmark-Kanada Rasmus Lerdorf pada tahun 1994.
- Kode PHP biasanya diproses di server web yang diimplementasikan sebagai modul, *daemon* atau sebagai *Common Gateway Interface* (CGI) yang dapat dieksekusi.
- Pengembangan PHP dimulai pada tahun 1994 ketika **Rasmus Lerdorf** menulis beberapa program Common Gateway Interface (CGI) di C, dan memperluasnya untuk bekerja dengan web form dan untuk berkomunikasi dengan database, dan menyebut penerapan ini "*Personal Home Page / Forms Interpreter*" atau *PHP / FI*.

Struktur/sintaks penulisan PHP

- Skrip PHP dieksekusi pada server, dan HTML mengirimkan hasil eksekusi ke Web browser
- Skrip PHP dapat ditulis dimanapun dalam suatu halaman yang dimulai dengan **<?php** dan diakhiri dengan **?>**

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
echo "My first PHP script!";
```

```
?>
```

```
</body>
```

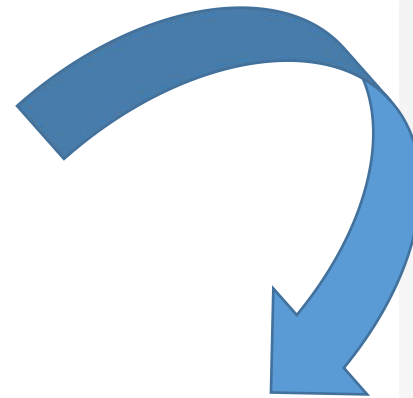
```
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```



My first PHP page

Hello World!



Dasar-dasar PHP

- Tipe data
- Variabel
- Operator
- Komentar
- Array 1 dan 2 dimensi
- Array asosiatif



Penulisan Komentar

- Komentar dalam kode PHP merupakan suatu baris yang tidak dibaca/ dieksekusi sebagai bagian dari program
- Kegunaan komentar yaitu:
 - Memudahkan orang memahami apa yang ditulis dalam kode PHP
 - Mengingatkan programmer akan semua yang ditulinya dalam program

```
<!DOCTYPE html>
<html>
<body>

<?php
// This is a single-line comment

# This is also a single-line comment

/*
This is a multiple-lines comment block
that spans over multiple
lines
*/

// You can also use comments to leave out parts of a code line
$x = 5 /* + 15 */ + 5;
echo $x;
?>

</body>
</html>
```


Echo dan Print

- Ada dua cara dasar untuk menampilkan hasil (output): **echo** dan **print**.
- Kebanyakan yang dipakai dalam PHP adalah **echo**.

```
<?php
```

```
echo "<h2>PHP is Fun!</h2>";  
echo "Hello world!<br>";  
echo "I'm about to learn PHP!<br>";
```

```
?>
```

```
<?php
```

```
Print "<h2>PHP is Fun!</h2>";  
Print "Hello world!<br>";  
Print "I'm about to learn PHP!<br>";
```

```
?>
```

Tipe Data dalam PHP

- Suatu variable dapat meyimpan tipe data yang berbeda, dan tipe data yang berbeda bisa melakukan hal-hal yang berbeda pula
- Tipe data yang mendukung dalam penulisan PHP antara lain:
 - String
 - Integer
 - Float (floating point number – biasa disebut duoble)
 - Boolean
 - Array
 - Object
 - NULL
 - Resource

Tipe Data String

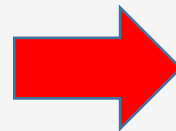
- String merupakan rangkaian suatu karakter, seperti “Hallo, Temans!”
- String merupakan teks didalam tanda petik
- Contoh penulisan:

```
<!DOCTYPE html>
<html>
  <body>

    <?php
      $x = "Hello world!";
      $y = 'Hello world!';

      echo $x;
      echo "<br>";
      echo $y;
    ?>

  </body>
</html>
```



Hello world!
Hello world!

Tipe Data Integer

- Tipe data integer merupakan bilangan asli non-desimal antara -2,147,483,648 sampai 2,147,483,647
- Aturan dalam tipe data integer
- Harus mempunyai sedikitnya satu digit
- Bukan bilangan desimal (tidak memiliki angka dibelakang koma → 0,1 – 10,123)
- Integer dapat bernilai positif atau negatif
- Integer dapat ditulis dalam tiga format yaitu basis 10 (desimal), basis 8 (oktal), basis 16 (heksadesimal)

```
• <!DOCTYPE html>  
<html>  
<body>
```

```
<?php  
$x = 5985;  
  
var_dump($x);  
?>
```

```
</body>  
</html>
```


Tipe Data Float

- Float digunakan ketika diperlukan angka desimal atau angka dalam bentuk eksponensial.
- Dalam contoh berikut \$ x adalah float. Fungsi PHP var_dump () mengembalikan tipe data dan nilai:

```
<?php  
$x = 10.365;  
var_dump($x);  
?>
```

Tipe Data Boolean

- Boolean merepresentasikan dua kondisi kemungkinan state yaitu true atau false
- Boolean biasa digunakan untuk proses pengujian kondisi
 - \$x = true;
\$y = false;

Tipe Data Array

- Array menyimpan banyak nilai dalam satu variabel tunggal.
- Dalam contoh berikut, **\$cars** adalah sebuah array.
- Fungsi PHP `var_dump ()` mengembalikan tipe data dan nilai:

```
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
var_dump($cars);  
?>
```

Tipe Data Objek

- Objek adalah tipe data yang menyimpan data dan informasi tentang cara memproses data itu.
- Dalam PHP, suatu objek harus dinyatakan secara eksplisit.
- Pertama kita harus mendeklarasikan kelas objek. Untuk ini, kami menggunakan kata kunci kelas. Kelas adalah struktur yang dapat berisi properti dan metode:

```
<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}

// create an object
$herbie = new Car();

// show object properties
echo $herbie->model;
?>
```


Variable

- Variable ditulis menggunakan tanda \$ kemudian diikuti dengan nama variable
- Contoh: $\$x = 4$; $\$nama = \text{"Sebuah Nama"};$
- Macam-macam variable:
 - Variable String
 - Variable numerik
- Cakupan Variable
 - local
 - global
 - static

Variable String

- Variable string adalah variable yang digunakan untuk mewakili suatu nilai string
- Penulisannya menggunakan quotes (tanda petik)

```
<?php // test1.php
$username = "Fred Smith";
echo $username;
echo "<br />";
$current_user = $username;
echo $current_user;
?>
```

Variable numerik

- Variable numerik adalah variable yang digunakan untuk mewakili suatu nilai angka
- Penulisannya tanpa menggunakan quotes (tanda petik)

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 5;
$y = 4;
echo $x + $y;
?>

</body>
</html>
```

Variable Local

- variable yang dideklarasikan didalam function dan hanya dapat diakses oleh function itu sendiri

```
<?php
function myTest() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}

myTest();

// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
```


Variable Global

- variable yang dideklarasika di luar function dan hanya dapat diakses diluar function

```
<?php
$x = 5; // global scope

function myTest() {
    // using x inside this function will generate an error
    echo "<p>Variable x inside function is: $x</p>";
}

myTest();

echo "<p>Variable x outside function is: $x</p>";
?>
```

- Variable global dapat digunakan dalam suatu function ketika ditulis dengan menyertakan \$GLOBALS[indeks]

```
<?php
$x = 5;
$y = 10;

function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}

myTest();
echo $y; // outputs 15
?>
```

Variable Statis

- variable yang masih dapat digunakan pada proses eksekusi selanjutnya

```
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}

myTest();
myTest();
myTest();
?>
```

□ Hasil nya

- 0
- 1
- 2
- ...
- ...

Operator

- Operator digunakan untuk melakukan operasi pada variabel dan nilai.
- PHP membagi operator dalam grup berikut:
 - ✓ Arithmetic operators
 - ✓ Assignment operators
 - ✓ Comparison operators
 - ✓ Increment/Decrement operators
 - ✓ Logical operators
 - ✓ String operators
 - ✓ Array operators
 - ✓ Conditional assignment operators

Arithmetic Operators

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

Assignment Operators

- Operator penugasan PHP digunakan dengan nilai numerik untuk menulis nilai ke variabel.
- Operator penugasan dasar dalam PHP adalah "=". Artinya bahwa nilai operan berada di sebelah kiri sedangkan ekspresi penugasan di sebelah kanan.

Assignment	Same as...	Description
$x = y$	$x = y$	The left operand gets set to the value of the expression on the right
$x += y$	$x = x + y$	Addition
$x -= y$	$x = x - y$	Subtraction
$x *= y$	$x = x * y$	Multiplication
$x /= y$	$x = x / y$	Division
$x \% = y$	$x = x \% y$	Modulus

Comparison Operators



- Operator perbandingan PHP digunakan untuk membandingkan dua nilai (angka atau string):

Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type
!=	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y
<>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y
<=>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. Introduced in PHP 7.



Conditional Statements

- **if** statement - mengeksekusi beberapa kode jika satu syarat benar
- **if ... else** statement - mengeksekusi beberapa kode jika suatu kondisi benar dan kode lain jika kondisi itu salah
- **if ... elseif ... else** statement - mengeksekusi kode yang berbeda untuk lebih dari dua syarat
- **Switch statement** - memilih salah satu dari banyak blok kode yang akan dieksekusi

IF

Syntax

```
if (condition) {  
    code to be executed if condition is  
true;  
}
```

Contoh:

```
<?php  
$t = date("H");  
  
if ($t < "20") {  
    echo "Have a good day!";  
}  
?>
```


If...Else

Syntax

```
if (condition) {  
    code to be executed if condition is true;  
} else {  
    code to be executed if condition is false;  
}
```

Contoh:

```
<?php  
$t = date("H");  
  
if ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```


If...Elseif...Else

Syntax

```
if (condition) {  
    code to be executed if this condition is true;  
} elseif (condition) {  
    code to be executed if first condition is false and this  
condition is true;  
} else {  
    code to be executed if all conditions are false;  
}
```

Contoh:

```
<?php  
$t = date("H");  
  
if ($t < "10") {  
    echo "Have a good morning!";  
} elseif ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```

Switch

Syntax

```
switch (n) {  
    case label1:  
        code to be executed if n=label1;  
        break;  
    case label2:  
        code to be executed if n=label2;  
        break;  
    case label3:  
        code to be executed if n=label3;  
        break;  
    ...  
    default:  
        code to be executed if n is different from all  
labels;  
}
```

```
<?php  
$favcolor = "red";  
  
switch ($favcolor) {  
    case "red":  
        echo "Your favorite color is red!";  
        break;  
    case "blue":  
        echo "Your favorite color is blue!";  
        break;  
    default:  
        echo "Your favorite color is neither  
red, blue, nor green!";  
}  
?>
```

Loop

- Loop digunakan untuk mengeksekusi blok kode yang sama berulang kali, selama kondisi tertentu benar.
- Di PHP, kami memiliki jenis loop berikut:
 - **while** – loop melalui blok kode selama kondisi yang ditentukan benar
 - **do ... while** – loop melalui blok kode sekali, dan kemudian mengulangi loop selama kondisi yang ditentukan benar
 - **for** – loop melalui blok kode beberapa kali
 - **foreach** – loop melalui blok kode untuk setiap elemen dalam array

While

- While mengeksekusi blok kode selama kondisi yang ditentukan benar.

- **Syntaks**

```
while (condition is true) {  
    code to be executed;  
}
```

Contoh:

```
<?php  
$x = 0;
```

```
while($x <= 100) {  
    echo "The number is: $x <br>";  
    $x+=10;  
}  
?>
```

Do...While

- Sintaks

```
do {  
    code to be executed;  
} while (condition is true);
```

Contoh:

```
<?php  
$x = 1;
```

```
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
?>
```


For

- Sintaks

```
for (init counter; test counter; increment  
counter) {  
    code to be executed for each iteration;  
}
```

Contoh:

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

Foreach

- Sintaks

```
foreach ($array as $value) {  
    code to be executed;  
}
```

Contoh:

```
<?php  
$colors  
= array("red", "green", "blue", "yellow");
```

```
foreach ($colors as $value) {  
    echo "$value <br>";  
}  
?>
```



References

- Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5
- <https://www.php.net/>
- <https://www.w3schools.com/php/>





Terima Kasih

