



# PHP Web

Eko Kurniawan Khannedy

# Eko Kurniawan Khannedy

- Technical architect at one of the biggest ecommerce company in Indonesia
- 10+ years experiences
- [www.programmerzamannow.com](http://www.programmerzamannow.com)
- [youtube.com/c/ProgrammerZamanNow](https://youtube.com/c/ProgrammerZamanNow)





# Eko Kurniawan Khannedy

- Telegram : [@khannedy](https://t.me/khannedy)
- Facebook : [fb.com/ProgrammerZamanNow](https://fb.com/ProgrammerZamanNow)
- Instagram : [instagram.com/programmerzamannow](https://instagram.com/programmerzamannow)
- Youtube : [youtube.com/c/ProgrammerZamanNow](https://youtube.com/c/ProgrammerZamanNow)
- Telegram Channel : [t.me/ProgrammerZamanNow](https://t.me/ProgrammerZamanNow)
- Email : [echo.khannedy@gmail.com](mailto:echo.khannedy@gmail.com)



# Sebelum Belajar

- PHP Dasar
- PHP Object Oriented Programming
- PHP 8 Fitur Baru
- PHP Database MySQL
- HTML
- CSS
- JavaScript



# Agenda

- Pengenalan Web
- Client dan Server
- Membuat Web dengan PHP
- Query Parameter
- Form Post
- Session
- Cookies
- Upload File
- Dan lain-lain

---

# Pengenalan Web

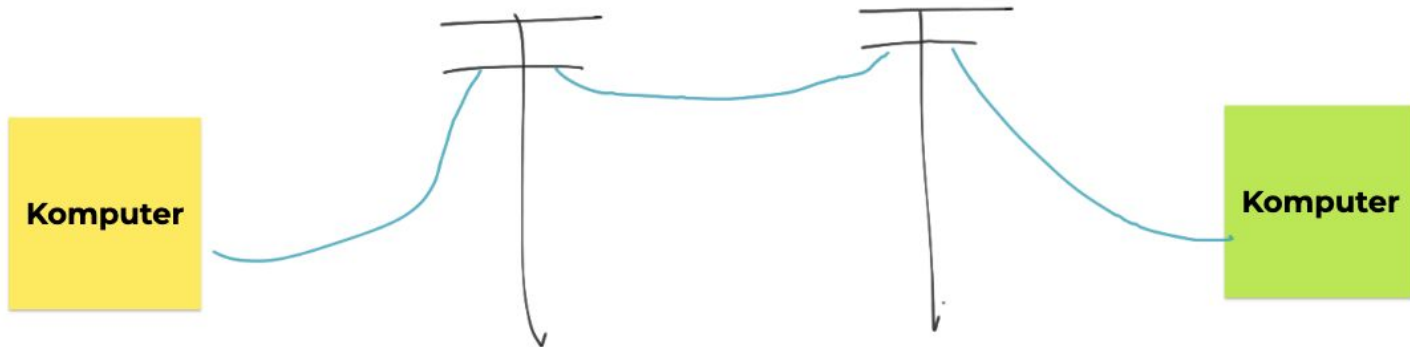


# Kenapa Web?

- Saat ini web digunakan oleh jutaan, bahkan mungkin milyaran orang setiap hari
- Dengan web, kita bisa melakukan belajar online, mendengarkan musik online, nonton video online, belanja online, sampai memesan makanan secara online
- Namun perlu diperhatikan, Web bukanlah Internet

# Internet

- Internet adalah mekanisme komunikasi antar komputer
- Awal internet ada, untuk komunikasi antar komputer, kita membutuhkan jaringan kabel telpon
- Namun sekarang, semenjak berjamurnya jaringan wifi dan sejenisnya, komunikasi antar komputer menjadi lebih cepat dan mudah



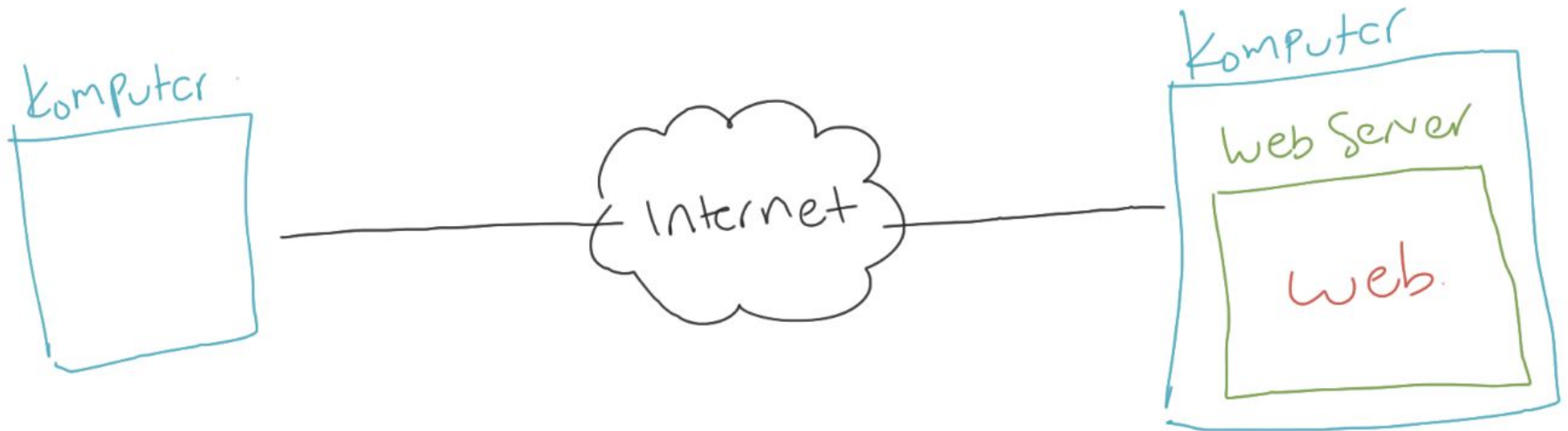




# Web

- Web merupakan kumpulan informasi yang tersedia dalam sebuah komputer yang terkoneksi secara terus menerus melalui internet
- Web bisa berisi informasi dalam bentuk apapun, seperti teks, gambar, audio, video dan lain-lain
- Web berjalan di aplikasi yang bernama Web Server, yaitu aplikasi yang digunakan untuk menyimpan dan menyampaikan isi informasi Web

## Diagram Web



# Web Host

- Pemilik Web, biasanya tidak menjalankan aplikasi Web Server di komputer pribadi nya
- Biasanya mereka akan menyewa komputer di tempat penyedia data center (kumpulan komputer) yang terjamin keandalan dan kecepatan koneksi internetnya
- Pihak penyedia komputer untuk Web Server biasa disebut Web Host





# Domain

- Saat komputer Web terhubung ke internet, biasanya dia memiliki alamat
- Alamat ini bernama ip address, formatnya misal nya 172.217.194.94
- Karena alamat ip address sangat menyulitkan untuk diingat
- Untung saja ada yang namanya nama domain
- Nama domain adalah alamat yang bisa digunakan sebagai alias ke ip address
- Misal seperti google.co.id, blibli.com, dan lain-lain
- Dengan nama domain, sebagai manusia kita akan mudah mengingat dibandingkan ip address
- Namun, saat kita menggunakan nama domain, sebenarnya komputer tetap akan mengakses web menggunakan alamat ip address



# Web Browser

- Jika Web Server adalah aplikasi yang digunakan untuk menyimpan informasi Web
- Web Browser adalah aplikasi yang digunakan untuk mengakses Web melalui internet
- Kita bisa saja mengakses Web secara langsung tanpa bantuan Web Browser, namun Web Server hanya akan memberikan informasi bahasa mesin seperti HTML, JavaScript, CSS, Gambar, Video dan lain-lain
- Dengan menggunakan Web Browser, semua bahasa mesin tersebut bisa ditampilkan secara visual sehingga kita bisa menyerap informasinya dengan lebih mudah

---

# Client dan Server



# Client dan Server

- Web adalah aplikasi berbasis Client dan Server, sekarang pertanyaannya, apa itu Client dan Server?
- Sederhananya client server merupakan konsep arsitektur aplikasi yang menghubungkan dua pihak, sistem client dan sistem server
- Sistem client dan sistem server yang saling berkomunikasi melalui jaringan komputer, internet, atau juga bisa di komputer yang sama

## Diagram Client dan Server







# Tugas Client dan Server

- Aplikasi Client bertugas mengirim request ke Server, dan menerima response dari Server
- Sedangkan, aplikasi Server bertugas menerima request dari Client, memproses data, dan mengembalikan hasil proses data ke Client



# Keuntungan Client dan Server

- Perubahan aplikasi bisa dilakukan dengan mudah di server, tanpa harus membuat perubahan di client, apalagi jika client nya di lokasi yang sulit dijangkau
- Bisa digunakan oleh banyak client pada saat yang bersamaan, walaupun server tidak banyak
- Bisa diakses dari mana saja, asal terhubung satu jaringan dengan server



# Contoh Client dan Server

- Web adalah salah satu contoh arsitektur client server
- Aplikasi yang bertugas sebagai Client adalah Web Browser (Chrome, Firefox, Opera, Edge dan lain-lain)
- Aplikasi yang bertugas sebagai Server adalah Web Server, dimana di dalam web server terdapat kode program Web kita

---

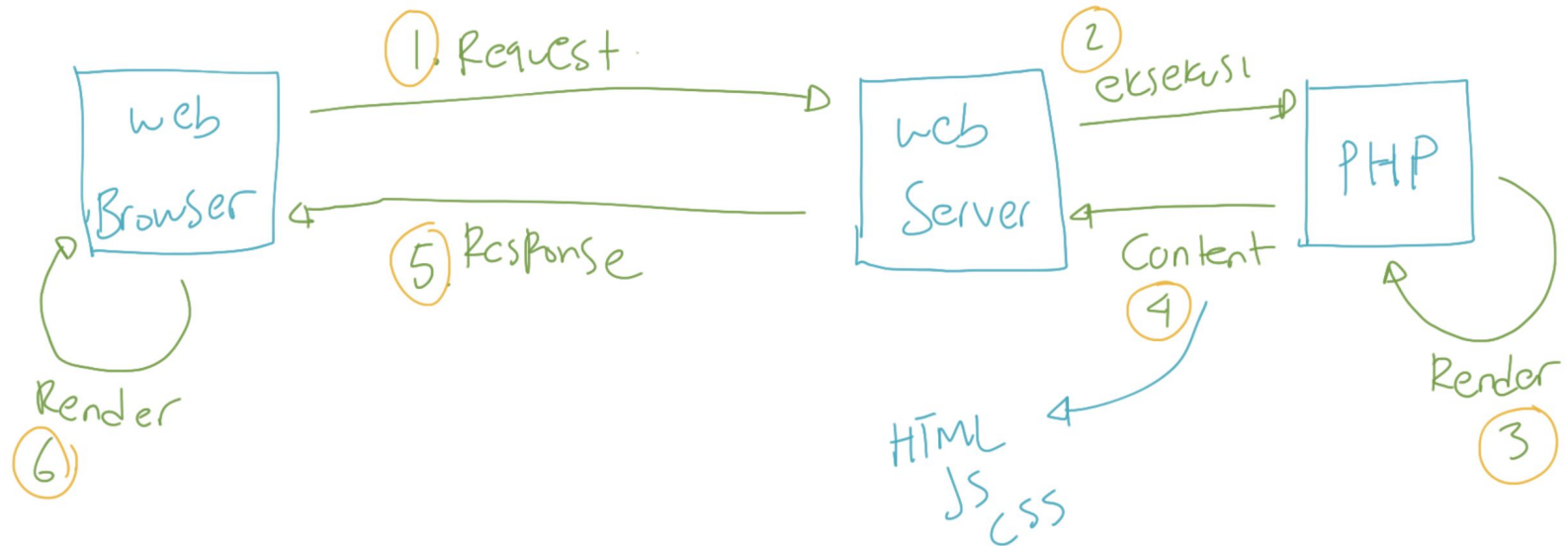
# PHP Web



# PHP Web

- PHP sendiri sejak awal memang diperuntukkan membuat aplikasi Web
- Di materi course-course sebelumnya kita sudah belajar bahasa pemrograman PHP, namun masih menggunakan PHP CLI (command line interface) atau berbasis terminal
- Pada course ini kita akan fokus membuat Web menggunakan PHP

# Diagram Cara Kerja PHP Web





# Cara Kerja PHP Web

1. Web Browser akan melakukan HTTP Request ke Web Server
2. Web Server menerima HTTP Request, lalu mengeksekusi file PHP yang sesuai dengan request tersebut
3. File PHP akan di eksekusi, lalu akan menjalankan kode program PHP, hasilnya akan di render menjadi content yang sesuai dengan yang dilakukan di kode program
4. Hasil content yang dibuat PHP (misal HTML, CSS, JavaScript, dan lain-lain) akan dikembalikan ke Web Server
5. Web Server akan mengembalikan content tersebut sebagai HTTP Response ke Web Browser
6. Web Browser menerima content dari Web Server, lalu me-render content tersebut sesuai dengan tipe content nya

---

# PHP Development Server





# PHP Development Server

- Seperti yang sudah kita bahas sebelumnya, agar web bisa diakses, kita perlu menyimpan web tersebut di dalam aplikasi Web Server
- Saat kita menginstall XAMPP, di dalam nya terdapat aplikasi Apache HTTPD, itu adalah salah satu aplikasi Web Server
- Namun pada course ini, saya tidak akan menggunakan Apache HTTPD, saya akan menggunakan Web Server sederhana yang sudah disediakan langsung oleh PHP, namanya PHP Development Server



# PERINGATAN!

- PHP Development Server adalah aplikasi Web Server untuk membantu mempermudah menjalankan Web Server untuk PHP Web
- Fitur ini hanya untuk proses development, bukan untuk dijalankan di Web Host
- PHP Development Server hanya memiliki 1 thread, artinya hanya bisa menjalankan 1 proses dalam satu waktu



# Menjalankan PHP Development Server

- `php -S host:port`
- Di komputer pribadi, kita bisa gunakan host dengan nama `localhost`
- Untuk nomor port, kita bisa gunakan bebas, asal port nya tidak digunakan oleh aplikasi lain
- Misal : `php -S localhost:8080`
- Artinya PHP Development Server akan bisa diakses di <http://localhost:8080>
- Ingat, kita wajib menjalankan perintah ini di folder tempat kita menyimpan kode program PHP kita



# Hot Reload

- PHP adalah bahasa pemrograman yang tidak butuh di compile terlebih dahulu
- Selain itu PHP juga mendukung fitur hot reload, artinya setiap perubahan yang kita lakukan di kode program PHP, kita tidak butuh melakukan restart aplikasi Web Server nya
- Kita cukup melakukan request ulang dari Web Browser, secara otomatis Web Server akan menjalankan kode program PHP terbaru kita

---

PHP Web Hello World



# PHP Web Hello World

- Sebelumnya di course PHP Dasar, kita sudah coba membuat program Hello World dengan PHP, namun masih berbasis terminal
- Sekarang kita akan coba membuat program Hello World di PHP, namun menggunakan PHP Web



## Kode : PHP Web Hello World



```
hello-world.php x
1 <?php
2 echo "Hello World";
3 |
```

The image shows a code editor window with a tab labeled 'hello-world.php'. The code is written in PHP and consists of two lines: a PHP opening tag and an echo statement that outputs 'Hello World'. The code is syntax-highlighted, with the opening tag in orange and the string in green. A vertical line is visible on the third line of the editor.



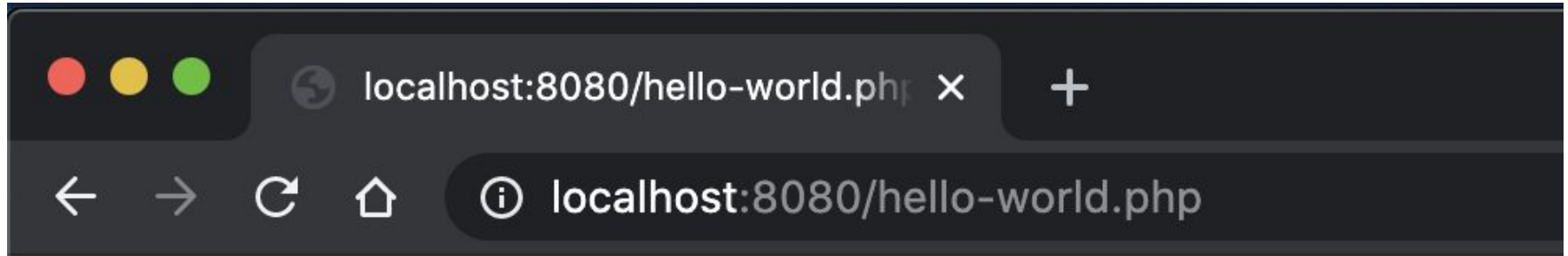
# Menjalankan PHP Web Hello World

- Setelah kita menjalankan PHP Development Server
- Untuk mengakses file PHP Hello World yang sudah kita buat, kita bisa menggunakan Web Browser dengan menggunakan URL : <http://localhost:8080/nama-file.php>
- Kita bisa sesuaikan nama-file.php dengan nama file yang kita buat



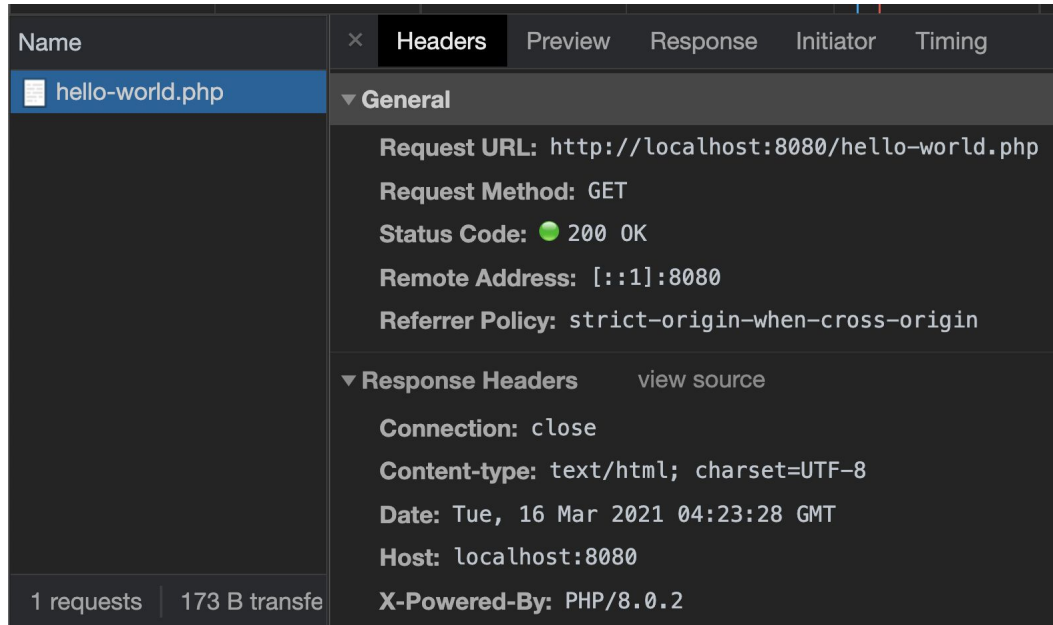


## Tampilan PHP Web Hello World



Hello World

# HTTP Request dan Response



The screenshot displays the 'Headers' tab of a web browser's developer tools. The left sidebar shows a file named 'hello-world.php'. The main panel is divided into two sections: 'General' and 'Response Headers'. The 'General' section provides details about the request, including the URL, method, status code (200 OK), remote address, and referrer policy. The 'Response Headers' section lists the connection status, content type, date, host, and the server software (PHP/8.0.2). At the bottom, a summary bar indicates '1 requests' and '173 B transfe'.

Name	Headers	Preview	Response	Initiator	Timing
hello-world.php	<p><b>General</b></p> <p><b>Request URL:</b> http://localhost:8080/hello-world.php</p> <p><b>Request Method:</b> GET</p> <p><b>Status Code:</b> 200 OK</p> <p><b>Remote Address:</b> [::1]:8080</p> <p><b>Referrer Policy:</b> strict-origin-when-cross-origin</p> <p><b>Response Headers</b> <a href="#">view source</a></p> <p><b>Connection:</b> close</p> <p><b>Content-type:</b> text/html; charset=UTF-8</p> <p><b>Date:</b> Tue, 16 Mar 2021 04:23:28 GMT</p> <p><b>Host:</b> localhost:8080</p> <p><b>X-Powered-By:</b> PHP/8.0.2</p>				

1 requests | 173 B transfe



# Best Practice URL di PHP

- Gunakan lowercase : misal /product.php, /login.php
- Jangan gunakan spasi : misal /hello-world.php, /forgot-password.php

---

# PHP Info



# PHP Info

- PHP memiliki fitur bernama PHP Info
- Fitur ini bisa digunakan untuk mengecek versi PHP dan fitur apa saja yang tersedia di aplikasi PHP yang kita install
- PHP Info bisa dilihat menggunakan terminal dengan perintah `php --info`
- Atau dengan menggunakan PHP Web dengan mengetikkan function `phpinfo()`



## Kode : PHP Info



```
1 <?php
2 phpinfo();
3
```

The image shows a code editor window with a tab labeled 'phpinfo.php'. The code is written in PHP and consists of two lines: a PHP opening tag and a call to the phpinfo() function. The code is syntax-highlighted, with the opening tag in orange and the function call in light blue. The editor has a dark background and a light gray sidebar on the left showing line numbers 1, 2, and 3.

# Tampilan PHP Info

[illegible]

---

# Integrasi dengan HTML





# Integrasi dengan HTML

- PHP adalah bahasa pemrograman yang secara default terintegrasi dengan baik dengan bahasa markup HTML
- Dengan integrasi PHP dan HTML, kita bisa membuat web HTML yang dinamis, dalam artian tidak statis dan bisa berubah-ubah sesuai dengan logika kode program kita
- Secara default ketika kita menggunakan kode `<? dan diakhir ?>`, artinya di dalamnya kita bisa memasukkan kode PHP, namun diluar kode tersebut, kita bisa memasukkan kode HTML seperti biasa

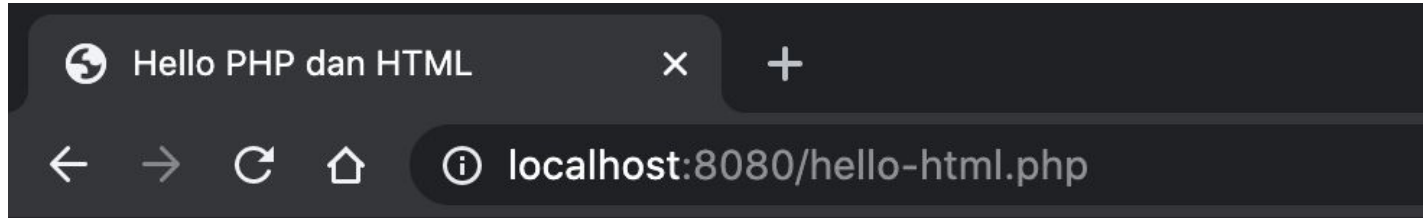
## Kode : Integrasi dengan HTML

```
<?php
$title = "Hello PHP dan HTML";
$body = "Hello PHP dan HTML";
?>

<html>
<head>
    <title><?php echo $title ?></title>
</head>
<body>
    <h1><?php echo $body ?></h1>
</body>
</html>
```



## Tampilan Web



# Hello PHP dan HTML



# Source Code HTML

← → ↻ 🏠 ⓘ view-source:localhost:8080/hello-html.php

```
1
2 <html>
3 <head>
4   <title>Hello PHP dan HTML</title>
5 </head>
6 <body>
7 <h1>Hello PHP dan HTML</h1>
8 </body>
9 </html>
```

10



## Kode <?=

- Kadang saat integrasi PHP dan HTML, kita sering sekali melakukan kode sederhana seperti ini :
  - `<?php echo $title ?>`
  - `<?php echo $body ?>`
- Hal itu bisa dipersingkat menggunakan tanda <?=?, misal :
  - `<?= $title ?>`
  - `<?= $body ?>`

## Kode : Tanda <? =

```
<?php
$title = "Hello PHP dan HTML";
$body = "Hello PHP dan HTML";
?>

<html>
<head>
    <title><? = $title ?></title>
</head>
<body>
    <h1><? = $body ?></h1>
</body>
</html>
```



# Tidak Hanya HTML

- Walaupun secara default, PHP terintegrasi dengan HTML, namun bukan berarti hanya bisa membuat content HTML yang dinamis
- PHP bisa mengembalikan konten apapun, seperti CSS, JavaScript, Image, Video dan lain-lain

---

# Global Variable SERVER





# **\$\_SERVER**

- Di PHP, terdapat global variable bernama \$\_SERVER
- \$\_SERVER merupakan array yang berisikan informasi seperti HTTP Header, path, lokasi script php, dan lain-lain
- \$\_SERVER dibuat ketika ada request masuk, sehingga bisa jadi tiap request akan berbeda isinya
- Beberapa key yang ada di \$\_SERVER mengikuti spesifikasi RFC 3875 - The Common Gateway Interface (CGI) Version 1.1 : <https://tools.ietf.org/html/rfc3875>

## Kode : Menampilkan Isi \$\_SERVER

```
<h1>$_SERVER</h1>
<table border="1">
  <?php foreach ($_SERVER as $key => $value) { ?>
    <tr>
      <td><?= $key ?></td>
      <td><?= $value ?></td>
    </tr>
  <?php } ?>
</table>
```

---

# Query Parameter



# Query Parameter

- Saat mengakses Web, kita akan menggunakan URL, dan di dalam URL, terdapat bagian yang bernama query parameter
- Query parameter adalah informasi tambahan yang bisa kita kirimkan dari client ke server secara dinamis
- Untuk menambahkan query parameter, cukup gunakan ? diikuti dengan key=value pada URL
- Di PHP, semua query parameter akan secara otomatis disimpan dalam global variable bernama `$_GET`



## **\$\_GET**

- \$\_GET adalah global variable berupa array yang berisikan key sesuai dengan nama query parameter, dan value sesuai dengan value query parameter
- Jika kita mengirim query parameter name=Eko, artinya akan ada key name di dalam \$\_GET yang bernilai Eko
- Perlu diperhatikan, query parameter adalah data yang dikirim oleh client, jadi pastikan sebelum menggunakannya kita harus cek apakah datanya ada atau tidak, karena bisa jadi datanya tidak dikirim oleh client

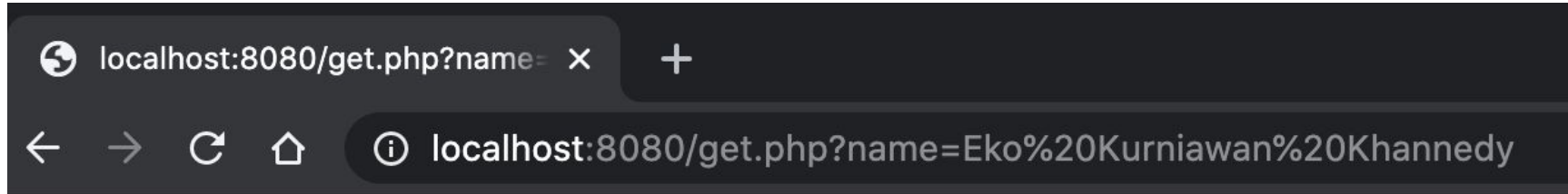


## Kode : Query Parameter

```
<?php
$say = "Hello " . $_GET['name'];
?>
<html>
<body>
<h1><?= $say ?></h1>
</body>
</html>
```



## Tampilan Query Parameter



**Hello Eko Kurniawan Khannedy**



# Multiple Query Parameter

- Query parameter di URL bisa memiliki banyak parameter
- Jika kita ingin mengirim banyak query parameter, kita cukup gunakan tanda & diikuti dengan query parameter selanjutnya

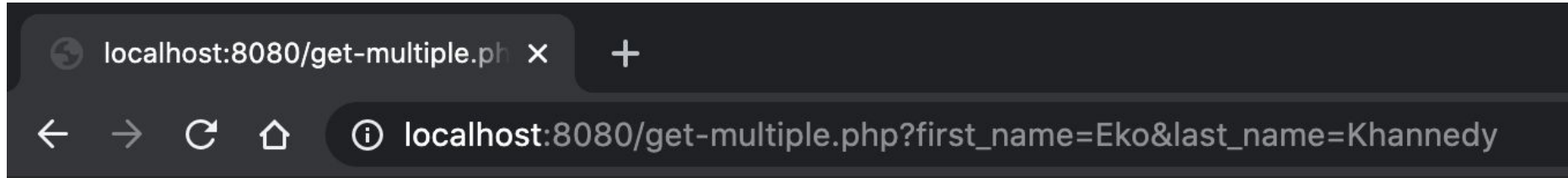


## Kode : Multiple Query Parameter

```
<?php
$say = "Hello " . $_GET['first_name'] . " " . $_GET['last_name'];
?>
<html>
<body>
<h1><?= $say ?></h1>
</body>
</html>
```



## Tampilan Multiple Query Parameter



**Hello Eko Khannedy**



## Query Parameter Array

- Secara default, query parameter berisikan data key=value, artinya 1 key akan berisi 1 value
- di PHP ada kemampuan untuk mengirim query parameter dengan data lebih dari satu
- Data tersebut akan otomatis dikonversi menjadi Array oleh PHP
- Caranya adalah kita wajib menambahkan tanda [] diakhir nama query parameter nya
- Dan kita bisa menambah query parameter dengan nama yang sama berkali-kali, misal
- `name[]=Eko&name[]=Budi&name[]=Joko`, maka secara otomatis parameter name akan berisi [Eko, Budi dan Joko]



## Kode : Array Query Parameter

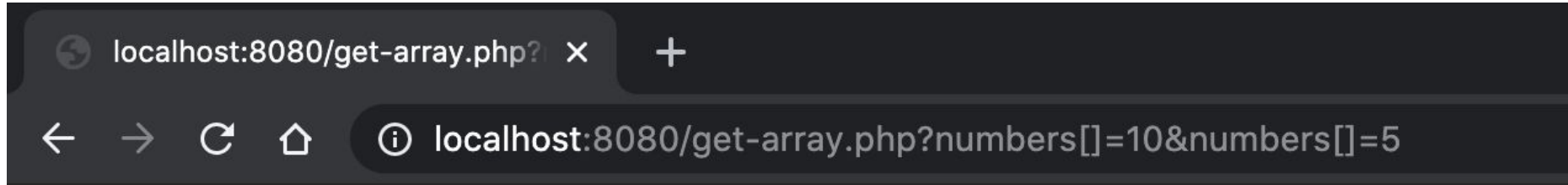
```
<?php
$numbers = $_GET['numbers'];
$total = 0;

foreach ($numbers as $number) {
    $total += $number;
}

?>
<html>
<body>
<h1><?= "Total = $total" ?></h1>
</body>
</html>
```



## Tampilan Array Query Parameter



**Total = 15**

---

# XSS (Cross-site scripting)



# Cross Site Scripting

- XSS merupakan celah keamanan yang biasanya dieksploitasi oleh penyerang dengan cara mengirim script pada parameter
- Jika sampai pembuat kode salah melakukan render halaman web, maka script tersebut akan di render di halaman web, dan bisa menyebabkan penyerang mengirimkan kode JavaScript
- Hal ini sangat berbahaya, karena penyerang bisa mencuri session pengguna yang sedang login di web kita

# Contoh XSS

name=Eko<h1><script>alert("Ups di Heck")</script><h1>

```
20di%20Heck")</script><h1>
```

localhost:8080 says

Ups di Heck

← → ↻ 🏠 ⓘ view-source:localhost:8080/get.php?name=Eko<h1><scr

```
1 <html>
2 <body>
3 <h1>Hello Eko<h1><script>alert("Ups di Heck")</script><h1></h1>
4 </body>
5 </html>
6
```





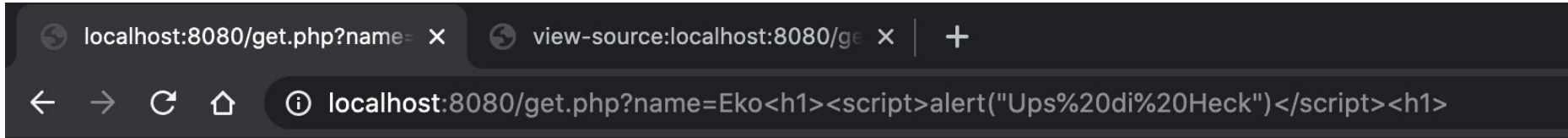
# Cara Mencegah XSS

- Cara mencegah XSS adalah, kita perlu berhati-hati ketika ingin menampilkan text yang inputnya berasal dari user
- Jika kita tidak berharang sebuah text berisi tag HTML, kita menggunakan function `htmlspecialchars(value)`
- Function `htmlspecialchars(value)` secara otomatis akan melakukan encode spesial character di html agar di render menjadi text biasa
- <https://www.php.net/manual/en/function htmlspecialchars.php>

## Kode : Menangani XSS

```
<?php
$say = "Hello " . htmlspecialchars($_GET['name']);
?>
<html>
<body>
<h1><?= $say ?></h1>
</body>
</html>
```

# Tampilan Menangani XSS



**Hello Eko<h1><script>alert("Ups di Heck")</script><h1>**

```
1 <html>
2 <body>
3 <h1>Hello Eko<h1><script>alert('Ups di Heck')</script><h1>
4 </body>
5 </html>
6
```

---

# Form Post



# Form Post

- Saat kita belajar HTML, kita tahu bahasa di HTML terdapat form
- Form biasanya digunakan untuk mengirim data dari Client ke Server
- Secara default, method di form adalah GET, sehingga saat kita submit, data akan dikirim dalam bentuk Query Parameter yang bisa ditangkap menggunakan `$_GET` di PHP
- Namun jika kita gunakan Form dengan method POST, maka secara otomatis request dari Client ke Server akan berupa HTTP POST, hal ini menyebabkan data form yang dikirim akan dikirim melalui Body, bukan Query Parameter, dan untuk menerima datanya di PHP, kita perlu menggunakan `$_POST`



# **\$\_POST**

- \$\_POST adalah global variable yang berisikan data array yang dikirim dari Client ke Server dalam bentuk form post
- Cara penggunaan \$\_POST sama seperti \$\_GET

## Kode : Form Post

```
<form action="post.php" method="post">
  <label>First Name :
    <input type="text" name="first_name"/>
  </label>
  <br/>
  <label>Last Name :
    <input type="text" name="last_name"/>
  </label>
  <br/>
  <input type="submit" value="Register">
</form>
```

## Kode : Menerima Form Post

```
<table>
  <tr>
    <td>First Name :</td>
    <td><?= $_POST['first_name'] ?></td>
  </tr>
  <tr>
    <td>Last Name :</td>
    <td><?= $_POST['last_name'] ?></td>
  </tr>
</table>
```





Header



# Header

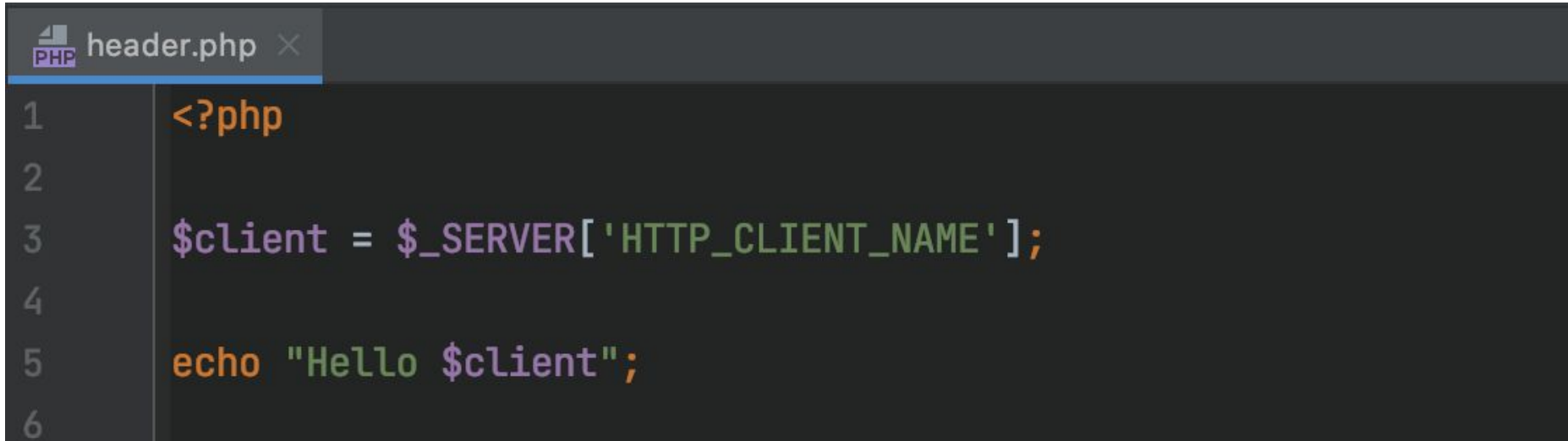
- Dalam HTTP request dan response, terdapat data yang bernama Header
- Header biasanya digunakan sebagai tempat menyimpan informasi tambahan diluar URL, METHOD dan BODY
- Di PHP, kita bisa menangkap data header yang dikirim oleh Client atau membuat header di response dari Server



# Menerima Header Request

- Semua header yang dikirim oleh Client, secara otomatis akan dimasukkan ke global variable `$_SERVER`
- Namun key untuk header akan secara otomatis di konversi menjadi UPPERCASE, dan jika terdapat spasi atau - , akan otomatis diganti menjadi \_
- Selain itu untuk membedakan request header dan lainnya, khusus request header, akan ditambah prefix `HTTP_`
- Misal header `Content-Type` akan menjadi `HTTP_CONTENT_TYPE`
- Header `Accept-Language` akan menjadi `HTTP_ACCEPT_LANGUAGE`

## Kode : Menangkap Header



```
1 <?php
2
3 $client = $_SERVER['HTTP_CLIENT_NAME'];
4
5 echo "Hello $client";
6
```

# Mengirim Header Menggunakan Postman

GET

http://localhost:8080/header.php

Authorization

Headers (1)

Body

Pre-request Script

Tests

	Key	Value
<input checked="" type="checkbox"/>	Client-Name	Belajar PHP
	New key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

HTML

i 1

Hello Belajar PHP



# Menambah Header Response

- Kadang kita ingin menambah informasi tambahan di HTTP response
- Seperti versi aplikasi kita atau lainnya
- Kita juga bisa menambahkan header dengan menggunakan function header(key, value) di PHP
- Perlu diingat, di spesifikasi HTTP, header berada pada posisi atas sebelum content dibuat, oleh karena itu, pastikan menambah header sebelum kita membuat content



## Kode : Menambah Header Response

```
<?php  
  
header('Application: Belajar PHP Web');  
header('Author: Eko Kurniawan');  
  
$client = $_SERVER['HTTP_CLIENT_NAME'];  
echo "Hello $client";
```

# Tampilan Header Response dengan Postman

GET

http://localhost:8080/header.php

Authorization

Headers (1)

Body

Pre-request Script

Tests

Key	Value
<input checked="" type="checkbox"/> Client-Name	Belajar PHP
New key	Value

Body

Cookies

Headers (7)

Test Results

application → Belajar PHP Web

author → Eko Kurniawan





Redirect



# Redirect

- Dalam pembuatan halaman web, melakukan redirect dari satu halaman ke halaman lain adalah hal biasa
- Misal setelah sukses login, kita akan redirect halaman web nya ke halaman member atau admin
- Atau bahkan melakukan redirect ke domain yang berbeda, misal ketika login dengan Google, setelah selesai, akan melakukan redirect ke halaman web kita lagi



# HTTP Redirect

- Redirect sendiri sudah merupakan sesuatu yang standard di HTTP
- Untuk melakukan redirect, kita hanya butuh menggunakan response header Location yang berisi url redirect nya, bisa dalam domain yang sama, atau ke domain berbeda, misal :
- Location : /admin.php => artinya redirect ke halaman admin di domain yang sama
- Location : <https://www.google.com> => artinya redirect ke halaman google



## Kode : Redirect

```
<?php  
  
header('Location: /phpinfo.php');  
exit(); // pastikan tidak ada content
```

---

# Response Code



# Response Code



- Dalam spesifikasi HTTP, response dari server biasanya memiliki code, atau dikenal HTTP Response Code
- Secara default, di PHP, response code adalah 200 OK
- Ada banyak sekali HTTP Response Code :  
<https://developer.mozilla.org/id/docs/Web/HTTP/Status>
- Dalam pembuatan website mungkin kita jarang sekali melakukan perubahan response code, namun jika nanti kita ingin membuat API menggunakan PHP, response code sangat penting digunakan

## Kode : Response Code

```
<?php
if (!isset($_GET['name']) || $_GET['name'] == "") {
    http_response_code(400);
    echo "Name is Required";
    exit();
}

$say = "Hello " . htmlspecialchars($_GET['name']);
?>
<html>
<body>
<h1><?= $say ?></h1>
```

# Hasil Response Code

Name	× Headers	Preview	Response	Initiator	Timing
 <code>get.php</code>	<div data-bbox="629 554 832 598">▼ General</div> <div data-bbox="701 642 1705 689">Request URL: <code>http://localhost:8080/get.php</code></div> <div data-bbox="701 721 1153 768">Request Method: <code>GET</code></div> <div data-bbox="701 801 1400 848">Status Code:  <code>400 Bad Request</code></div> <div data-bbox="701 881 1319 927">Remote Address: <code>:::1:8080</code></div> <div data-bbox="701 959 1789 1006">Referrer Policy: <code>strict-origin-when-cross-origin</code></div>				



---

# Session



# Session

- Session adalah mekanisme yang biasa digunakan untuk mengingat interaksi sebelumnya dari Client
- Di spesifikasi HTTP, tidak dikenal yang namanya session, karena HTTP di desain stateless (tidak menyimpan state, data atau informasi apapun)
- Session adalah teknik dimana kita bisa menyimpan informasi client di server, sehingga ketika ada request dari client yang sama, kita bisa tahu dari server
- Contohnya session biasanya digunakan ketika proses login, setelah login sukses, kita akan membuat session, dan request selanjutnya dari client akan membawa informasi session tersebut



# Memastikan Fitur Session di PHP

- Untuk menggunakan fitur session di PHP, kita harus pastikan bahwa fitur tersebut aktif, kita bisa melihatnya di phpinfo
- <https://www.php.net/manual/en/book.session.php>

## session

Session Support	enabled
Registered save handlers	files user
Registered serializer handlers	php_serialize php php_binary



# **\$\_SESSION**

- \$\_SESSION adalah global variable array yang berisi data di session
- Setelah session kita start, kita bisa menggunakan \$\_SESSION untuk menyimpan data di session, atau mengambil data di session
- Jika session belum di start, perubahan yang kita lakukan di \$\_SESSION atau pengambilan data di \$\_SESSION, tidak akan berpengaruh apa-apa

## Kode : Menjalankan Session (1)

```
session_start();

if($_SESSION['login'] == true){
    header('Location: /session/member.php');
    exit();
}

$error = "";
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if ($_POST['username'] == 'eko' && $_POST['password'] = 'eko') {
        $_SESSION['login'] = true;
        $_SESSION['username_login'] = 'eko';
        header('Location: /session/member.php');
        exit();
    } else {
        $error = "Login Gagal";
    }
}
```

## Kode : Menjalankan Session (2)

```
<body>
<h1>Login</h1>
<?php
if ($error != "") {
    echo "<p>$error</p>";
}
?>
<form action="login.php" method="post">
    <label>First Name :
        <input type="text" name="username"/>
    </label>
    <br/>
    <label>Last Name :
        <input type="password" name="password"/>
    </label>
    <br/>
    <input type="submit" value="Login">
```

## Kode : Mengambil Session

```
session_start();
if ($_SESSION['login'] != true) {
    header('Location: /session/login.php');
    exit();
}
$say = "Hello " . $_SESSION['username_login'];
?>

<html>
<body>
<h1><?= $say ?></h1>
<a href="logout.php">Logout</a>
```



## Kode : Menghapus Session

```
session_start();  
session_destroy();  
  
header('Location: /session/login.php');  
exit();|
```

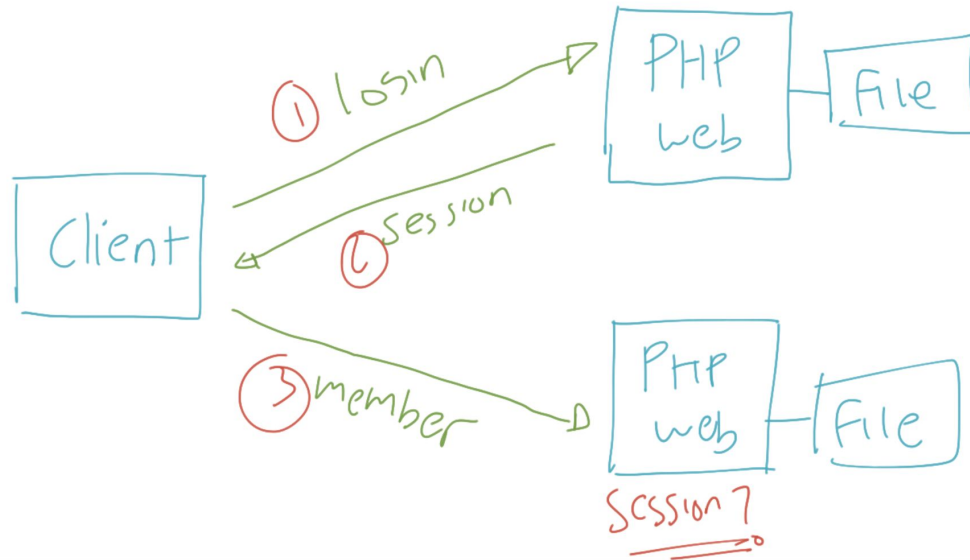




# Masalah dengan PHP Session

- PHP menyimpan session di File
- Hal ini tidak direkomendasikan untuk menyimpan data yang terlalu besar di session, karena setiap kita melakukan `session_start`, maka secara otomatis session akan di load dari file, dan setelah request selesai, session akan di save di file
- Selain itu penggunaan File menyebabkan masalah scalability, karena jika kita menjalankan 2 Web PHP secara bersamaan di server yang berbeda, bisa jadi session nya menghilang ketika request dari client masuk ke server yang berbeda
- Oleh karena itu, penggunaan fitur session di PHP sebenarnya jarang sekali digunakan di web yang sudah besar, biasanya akan menggunakan alternatif lain untuk management session, seperti Secure Cookie atau JWT (Json Web Token)

## Session Stateful di PHP



---

# Cookie



# Cookie

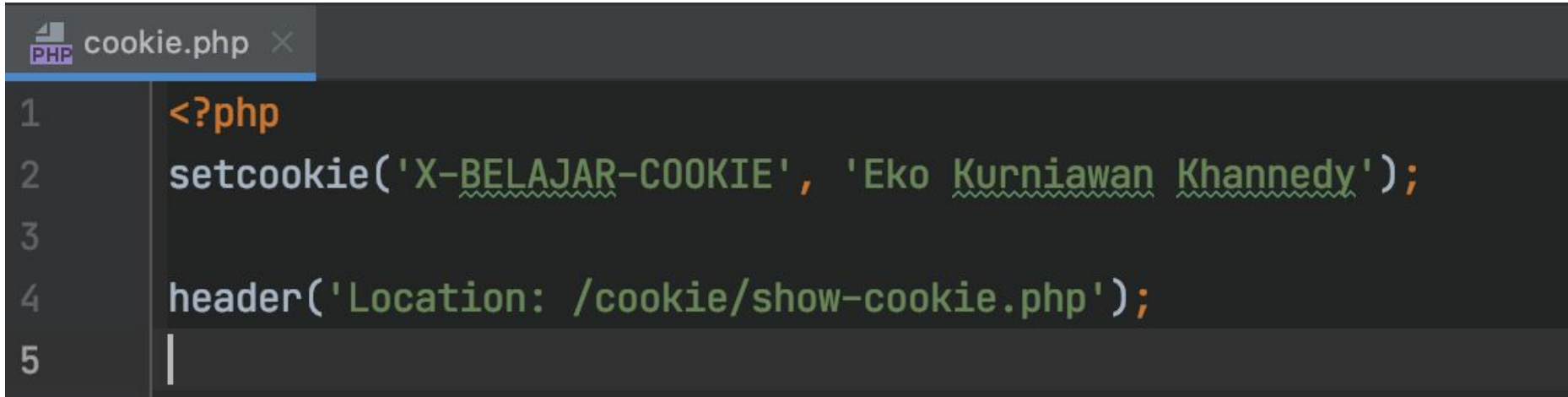
- Cookie adalah data dalam bentuk key-value yang dikirim oleh Server pada HTTP Response untuk disimpan di Client (Web Browser)
- Ketika Web Browser menerima cookie, secara otomatis, request selanjutnya cookie tersebut akan dikirim di setiap HTTP Request yang dilakukan ke Server
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>



# Cookie di PHP

- Untuk membuat cookie di PHP, kita bisa menggunakan function `setcookie(...)`
- <https://www.php.net/manual/en/function.setcookie.php>
- Cookie merupakan bagian dari HTTP Header, jadi pastikan cookie dibuat sebelum content di render di PHP
- Untuk membaca cookie yang dikirim oleh client, kita bisa menggunakan global variable `$_COOKIE`
- <https://www.php.net/manual/en/reserved.variables.cookies.php>

## Kode : Membuat Cookie



The image shows a code editor window with a tab labeled 'cookie.php'. The code is written in PHP and consists of five lines. Line 1 is the opening PHP tag. Line 2 sets a cookie named 'X-BELAJAR-COOKIE' with the value 'Eko Kurniawan Khannedy'. Line 3 is empty. Line 4 sends a header to redirect to '/cookie/show-cookie.php'. Line 5 is empty.

```
1 <?php
2 setcookie('X-BELAJAR-COOKIE', 'Eko Kurniawan Khannedy');
3
4 header('Location: /cookie/show-cookie.php');
5 |
```



## Kode : Mengambil Cookie

```
show-cookie.php x
1  <?php
2
3  $belajarCookie = $_COOKIE['X-BELAJAR-COOKIE'];
4
5  echo $belajarCookie;
6
```



# Hati-Hati Menggunakan Cookie

- Jangan terlalu banyak data di cookie, karena semua data di cookie akan selalu dikirim di tiap HTTP Request, semakin banyak, semakin lambat karena data yang harus dikirim banyak
- Cookie bisa diubah oleh user, jadi jangan mudah percaya dengan isi cookie, misal menyimpan informasi hak akses di cookie



---

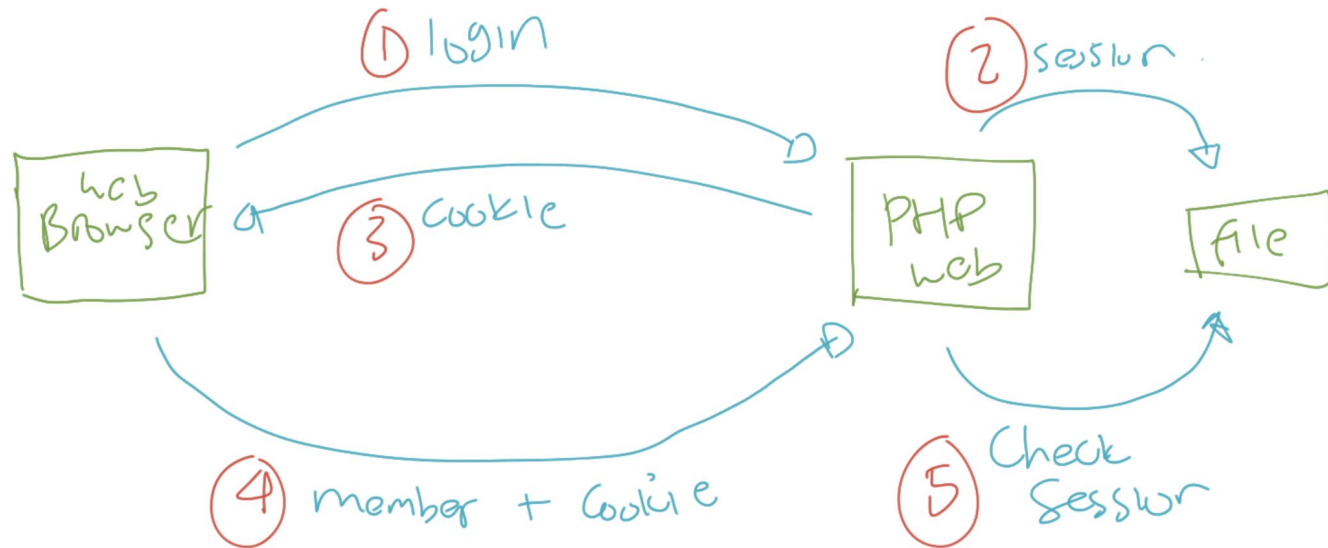
# Session dan Cookie



# Session dan Cookie

- Sebelumnya kita sudah belajar session di PHP, sebenarnya session id di PHP disimpan dalam cookie
- Ketika kita membuat session, PHP akan membuat cookie session id, cookie session id tersebut digunakan sebagai id untuk file session di server PHP
- Ketika web browser mengirim cookie berisi session id tersebut, PHP akan mengecek apakah nama file dengan session id tersebut ada atau tidak, jika ada, otomatis session nya valid dan akan di ambil semua data session di dalam file tersebut

## Diagram Session di PHP



# Session Cookie PHP

SourcesNetworkPerformanceMemoryApplicationSecurityLight

Filter

Only show

Name▲	Value	Dom
PHPSESSID	1fnhjquvd4sq54cnv67nugi4t5	loca
X-BELAJAR-COOKIE	Eko%20Kurniawan%20Khannedy	loca

---

**Upload File**



# Upload File

- Selain mengirim data berupa Query Param dan Form Post, kadang kita juga butuh melakukan upload file dari client ke server
- Dan PHP Web juga menerima upload file



# PHP Upload File

- Di PHP, untuk file yang di upload dari server, kita bisa menangkapnya dari global variable `$_FILES`
- Setiap ada file yang di upload, secara otomatis `$_FILES` akan berisi informasi seputar file yang di upload
- <https://www.php.net/manual/en/features.file-upload.post-method.php>
- Saat di upload ke server, file akan secara otomatis disimpan di temporary folder di server, dari situ kita bisa pindahkan ke lokasi yang kita inginkan



# **`$_FILES`**

## `$_FILES['userfile']['name']`

The original name of the file on the client machine.

## `$_FILES['userfile']['type']`

The mime type of the file, if the browser provided this information. An example would be "image/gif". This mime type is however not checked on the PHP side and therefore don't take its value for granted.

## `$_FILES['userfile']['size']`

The size, in bytes, of the uploaded file.

## `$_FILES['userfile']['tmp_name']`

The temporary filename of the file in which the uploaded file was stored on the server.

## `$_FILES['userfile']['error']`

The error code associated with this file upload.





# PHP Info - Upload Configuration

<b>unserialize_callback_func</b>	<i>no value</i>	<i>no value</i>
<b>upload_max_filesize</b>	40M	40M
<b>upload_tmp_dir</b>	/Applications/XAMPP/xamppfiles/temp/	/Applications/XAMPP/xamppfiles/temp/
<b>user_dir</b>	<i>no value</i>	<i>no value</i>

## Kode : Form Upload

```
<html>
<body>
<form action="upload.php" method="post" enctype="multipart/form-data">
  <label>File :
    <input type="file" name="upload_file">
  </label>
  <input type="submit" value="Upload">
</form>
</body>
</html>
```



## Kode : Menerima File Upload

```
$file_name = $_FILES['upload_file']['name'];  
$file_type = $_FILES['upload_file']['type'];  
$file_size = $_FILES['upload_file']['size'];  
$file_tmp_name = $_FILES['upload_file']['tmp_name'];  
$file_error = $_FILES['upload_file']['error'];
```



# Memindahkan File Upload

- Secara default, file upload akan disimpan di temporary folder
- Kadang kita ingin memindahkan file tersebut ke lokasi dimana kita bisa mengakses file tersebut
- Untuk melakukan hal tersebut, PHP memiliki banyak function untuk memanipulasi file dan folder
- <https://www.php.net/manual/en/ref.filesystem.php>



## Kode : Memindahkan File Upload

```
$file_name = $_FILES['upload_file']['name'];  
$file_type = $_FILES['upload_file']['type'];  
$file_size = $_FILES['upload_file']['size'];  
$file_tmp_name = $_FILES['upload_file']['tmp_name'];  
$file_error = $_FILES['upload_file']['error'];  
  
move_uploaded_file($file_tmp_name, __DIR__ . '/file/' . $file_name);
```

---

**Download File**



# Download File

- Saat membuat web, kadang-kadang kita butuh mengembalikan content berupa file yang butuh di download oleh client
- Sebenarnya, kita bisa langsung mengakses file tersebut melalui URL, namun kita mau, kita juga bisa menggunakan PHP untuk membuat content dalam bentuk file dan memaksa client untuk mendownload nya



# Download File di PHP

- Memaksa client untuk melakukan download file, sebenarnya sudah ada di standard HTTP Header, ini menggunakan Content-Disposition :  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Disposition>
- Jadi di PHP kita cukup dengan menambahkan header tersebut, maka kita bisa secara otomatis memaksa browser untuk download file yang kita buat menggunakan PHP
- Untuk membaca file dan me-load nya di PHP, kita bisa menggunakan function readfile(filename) :  
<https://www.php.net/manual/en/function.readfile.php>



## Kode : Download File di PHP

```
if (isset($_GET['key']) && $_GET['key'] == 'rahasia') {  
    header('Content-Disposition: attachment; filename="profile.jpg"');  
    header('Content-Type: image/jpg');  
    readfile(__DIR__ . '/upload/file/profile.jpg');  
    exit();  
} else {  
    echo "Invalid Link";  
    exit();  
}
```

---

# Materi Selanjutnya



# Materi Selanjutnya

- PHP Composer
- PHP Unit Test