

BAB 10

Sessions Dan Cookies

Secara default, halaman HTML dan web server tidak menyimpan informasi yang telah dimasukkan dalam sebuah halaman ketika browser user memproses halaman lain. Hal ini membuat melakukan hal yang berkaitan dengan penggunaan informasi dari user yang sama dalam beberapa halaman terasa sulit.

Session membantu memecahkan masalah ini, dengan menyimpan data pada saat user mengunjungi web site, dari halaman satu ke halaman lain. Tiap-tiap session mampu menyimpan beberapa variabel yang akan tetap tersimpan dalam session tersebut.

Berikut ini akan diberikan contoh tentang penggunaan session.

Contoh:

Diberikan source code dengan beberapa halaman sbb.

Form.php

```
<form action=submit.php method=post>
Username    <input    type=text    name=username><input    type=submit
name=submit
value=Submit>
</form>
```

Submit.php

```
<?
$username = $_POST['username'];
echo "Nama user Anda adalah: $username<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>
```

Ha1.php

```
<?
echo "Ini adalah halaman 1<br>";
echo "Nama user Anda adalah: $username<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>
```

Hal2.php

```
<?
echo "Ini adalah halaman 2<br>";
echo "Nama user Anda adalah: $username<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>
```

Hal3.php

```
<?
echo "Ini adalah halaman 3<br>";
echo "Nama user Anda adalah: $username<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>
```

Dengan file-file di atas, user diminta melakukan login terlebih dahulu (memasukkan nama user) melalui form.php. Selanjutnya akan muncul submit.php yang menampilkan nama user yang dimasukkan tadi dan beberapa link ke halaman lain. Begitu user mengakses 3 buah halaman web yang ada tersebut, diharapkan nama user yang telah dimasukkan dalam form sebelumnya tetap ditampilkan dalam setiap halaman web yang diakses. Akan tetapi apa yang terjadi? Ternyata untuk ketiga halaman web yang diakses tersebut tidak menampilkan nama user. Apa penyebabnya? Hal ini dikarenakan hilangnya data/informasi dari nama user yang dimasukkan sebelumnya.

Di sinilah letak perlunya digunakan session. Dengan session, data dapat disimpan dan selanjutnya dapat diakses di beberapa halaman web. Penggunaan session sering diterapkan pada aplikasi web yang bersifat multiuser, seperti online shopping, web based mail, e-banking, dll.

Data yang tersimpan dalam session bersifat temporary/ sementara. Biasanya akan terhapus secara otomatis begitu user menutup browser, atau melakukan logout.

Memulai PHP – Session

Sebelum Anda menyimpan data dalam session, terlebih dahulu harus memulai session. Untuk memulai session, perintahnya adalah:

```
session_start()
```

Menyimpan Data ke dalam Session

Untuk menyimpan data ke dalam session, digunakan perintah

```
$_SESSION['nama_session'] = data;
```

Contoh:

Berikut ini adalah code dalam file submit.php (contoh sebelumnya) yang telah dimodifikasi.

Nama user akan disimpan dalam session.

Menyimpan session setelah mendapat input form dari halaman berbeda,

```
<?
session_start();
$username = $_POST['username'];
$_SESSION['namauser'] = $username;
echo "Nama user Anda adalah: $username<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>
```

Memanggil Data yang Tersimpan dalam Session

Setelah data disimpan dalam session, selanjutnya dapat dipanggil kembali apabila diperlukan. Untuk memanggil data dalam session, caranya cukup menuliskan

```
$_SESSION['nama_session'].
```

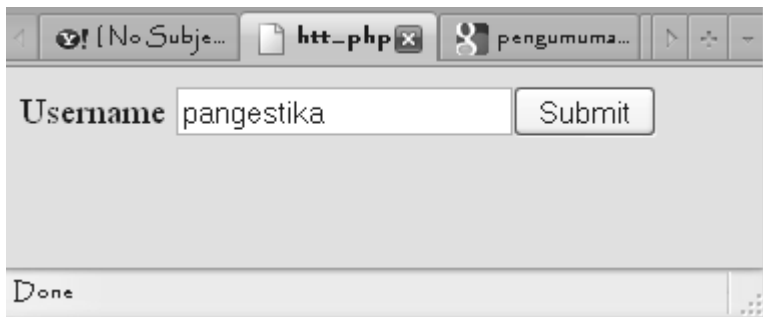
Contoh :

Berikut ini adalah code dari file hal1.php, hal2.php, dan hal3.php sehingga dapat menampilkan nama user yang telah disimpan dalam session. Dengan inputan form dari submit.php

form.php

```
<form action=submit.php method=post>
Username <input type=text name=username><input type=submit
name=submit
value=Submit>
</form>
```

Tampilan browsernya



Kemudian inputan tersebut diproses dan disimpan kedalam variable session

Submit.php

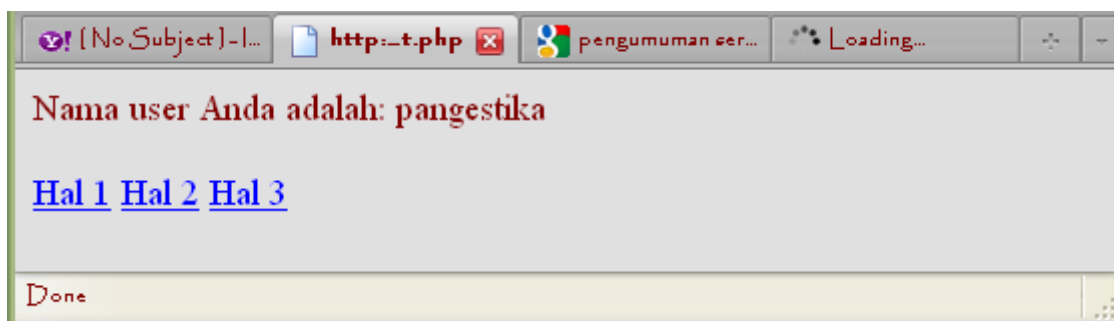
```

<?php
session_start();
$username = $_POST['username'];
$_SESSION['namauser'] = $username;
echo "Nama user Anda adalah: $username<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>

```

Hasil inputan dari form.php (yaitu "pangestika") diproses didalam submit.php dan pada halaman ini dimulai proses session. Inputan dengan variable luar \$username disimpan dalam session dengan variable penyimpan 'namauser'.

Tampilan browsernya



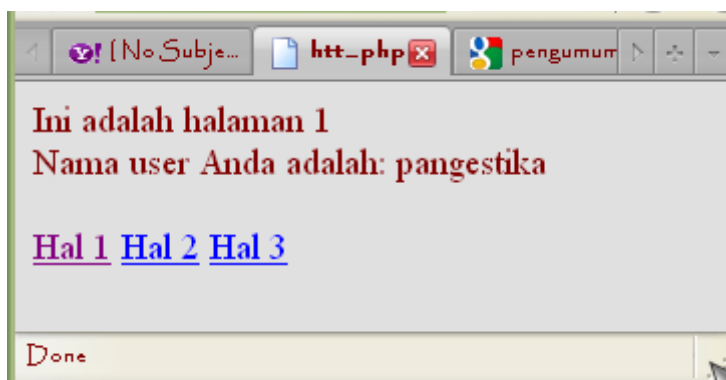
Hal1.php

```

<?
session_start();
echo "Ini adalah halaman 1<br>";
echo "Nama user Anda adalah: " . $_SESSION['namauser'] . "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>

```

Setelah inputan dari form.php disimpan di session pada submit.php, maka pada hal1, hal2, dan hal3, kita tinggal memanggil session dengan \$_SESSION['namauser']



Hal2.php

```

<?
session_start();

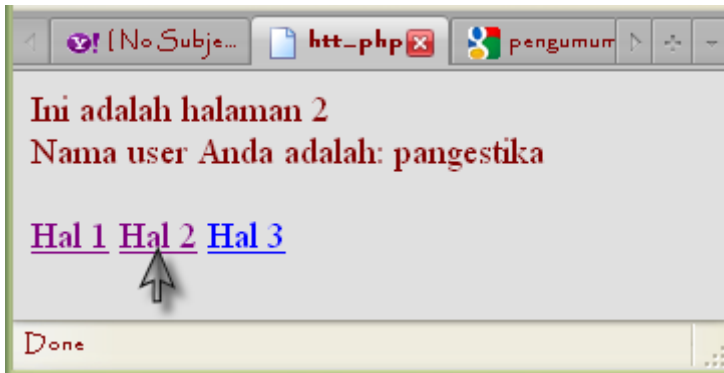
```

```

echo "Ini adalah halaman 2<br>";
echo "Nama user Anda adalah: ".$_SESSION['namauser']. "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>

```

Tampilan browsernya



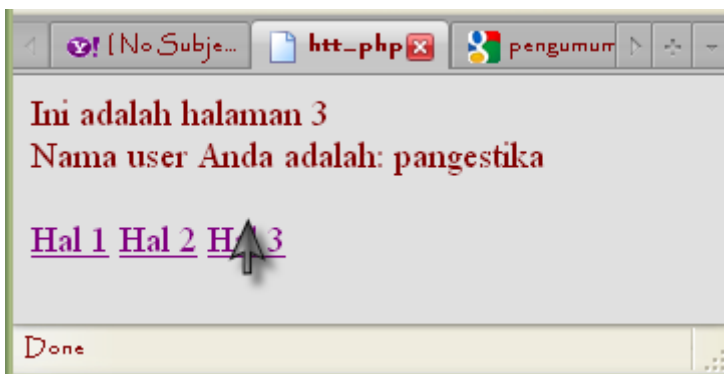
Hal3.php

```

<?
session_start();
echo "Ini adalah halaman 3<br>";
echo "Nama user Anda adalah: ".$_SESSION['namauser']. "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>

```

Tampilan browsernya



Menghapus Data Session

Untuk menghapus data yang tersimpan dalam suatu session, digunakan perintah `unset($_SESSION['nama_session']);`

Misalkan terdapat lebih dari satu session dan kita ingin menghapus semuanya, maka caranya dengan memberikan perintah

```
session_destroy();
```

Biasanya menghapus data session yang terkait dengan username diterapkan pada proses logout. Dengan proses logout, maka username yang telah tersimpan dalam session akan terhapus.

Contoh:

Akan dibuat script untuk proses logout dari kasus di atas.

```
<?  
session_start();  
unset($_SESSION['namauser']);  
?>
```

Penerapan Session untuk Security

Pada contoh kasus di atas, misalkan kita buat aturan bahwa untuk dapat mengakses halaman 1, 2, dan 3, user harus terlebih dahulu melakukan login. Dari script yang kita buat di atas, seorang user bisa saja langsung melakukan by pass ke tiga halaman tersebut tanpa login terlebih dahulu. Tentu saja hal ini bisa berbahaya untuk aplikasi multiuser yang harus menjamin keamanan data dari para usernya.

Untuk mencegah proses by pass tersebut, dapat kita akali dengan menggunakan session. Logikanya adalah, apabila user melakukan login, maka nama user yang dia masukkan akan disimpan dalam session. Sedangkan apabila ada seorang user yang mencoba mem by pass, maka dengan kata lain username tidak akan disimpan dalam session (session untuk user masih kosong), dengan catatan bahwa user lain yang sebelumnya login harus sudah me-logout. Dari hal ini, kita dapat melakukan cek apakah seorang user sudah melakukan login atau belum dengan melihat session, masih kosong atau tidak. Untuk melihat sebuah session masih kosong atau tidak dengan menggunakan perintah

```
isset($_SESSION['nama_session'])
```

Perintah di atas akan menghasilkan nilai TRUE apabila session sudah tidak kosong, dan akan menghasilkan nilai FALSE apabila session masih kosong.

Contoh:

Kita akan membuat script untuk mencegah by pass dalam kasus sebelumnya. Script ini selanjutnya akan disisipkan pada setiap halaman yang diinginkan, dalam hal ini adalah halaman 1, 2, dan 3.

Cek.php

```
<?
```

```

session_start();
if (!isset($_SESSION['namauser']))
{
    echo "Anda belum login";
    exit;
}
?>

```

Dan berikut adalah script hal1.php, hal2.php, dan hal3.php yang sudah dimodifikasi

Hal1.php

```

<?
session_start();
include "cek.php";
echo "Ini adalah halaman 1<br>";
echo "Nama user Anda adalah: " . $_SESSION['namauser'] . "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>

```

Hal2.php

```

<?
session_start();
include "cek.php";
echo "Ini adalah halaman 2<br>";
echo "Nama user Anda adalah: " . $_SESSION['namauser'] . "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>

```

Hal3.php

```

<?
session_start();
include "cek.php";
echo "Ini adalah halaman 3<br>";
echo "Nama user Anda adalah: " . $_SESSION['namauser'] . "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a href=hal2.php>Hal 2</a> <a
href=hal3.php>Hal 3</a>";
?>

```

Cookies

Cookies adalah sarana yang digunakan oleh administrator web untuk menyimpan informasi user yang telah mengunjungi website mereka dan cookies ini tersimpan di dalam komputer user tersebut.

Membuat Cookies

Untuk membuat cookies, caranya dengan menggunakan perintah `setcookie(name, value, expiration)`. Perintah ini memiliki 3 buah argumen, yaitu

1. `name`, merupakan nama dari cookie. Nama cookie ini dapat dipanggil sewaktu-waktu untuk mendapatkan informasi. Jadi jangan sampai nama cookie ini lupa.
2. `value`, merupakan informasi atau data yang disimpan dalam cookie. Biasanya `value` ini berupa username atau tanggal pengaksesan suatu halaman web.
3. `expiration`, merupakan batas waktu penyimpanan cookie (dalam detik timestamp). Apabila lama penyimpanan sebuah cookie melebihi batas waktu ini, maka secara otomatis cookie tersebut akan terhapus.

Contoh:

Pada contoh ini akan dibuat perintah PHP untuk membuat cookie yang digunakan untuk menyimpan data waktu kunjungan terakhir seorang user yang mengakses suatu halaman web.

Cookie ini diberi batas waktu sampai 2 jam penyimpanan.

```
<?php
$duaJamLagi = time() + 2 * 3600;
setcookie(KunjunganTerakhir, date("G:i - m/d/y"), $duaJamLagi);
?>
```

Mengambil Informasi dari Cookie

Apabila cookie belum terhapus, maka kita dapat mengambil informasi dari cookie. Untuk mengecek apakah suatu cookie sudah terhapus atau belum menggunakan perintah

```
isset($_COOKIE['nama_cookie'])
```

Apabila cookie masih ada, maka perintah di atas menghasilkan TRUE. Sedangkan apabila sudah terhapus, akan menghasilkan nilai FALSE.

Sedangkan untuk mengambil informasi dari cookie, menggunakan perintah


```
$_COOKIE['nama_cookie']
```

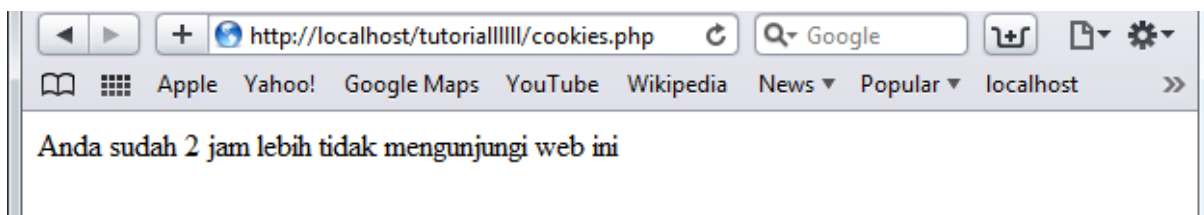
Contoh:

Berikut ini contoh perintah PHP untuk menampilkan tanggal kunjungan terakhir user yang mengunjungi halaman web.

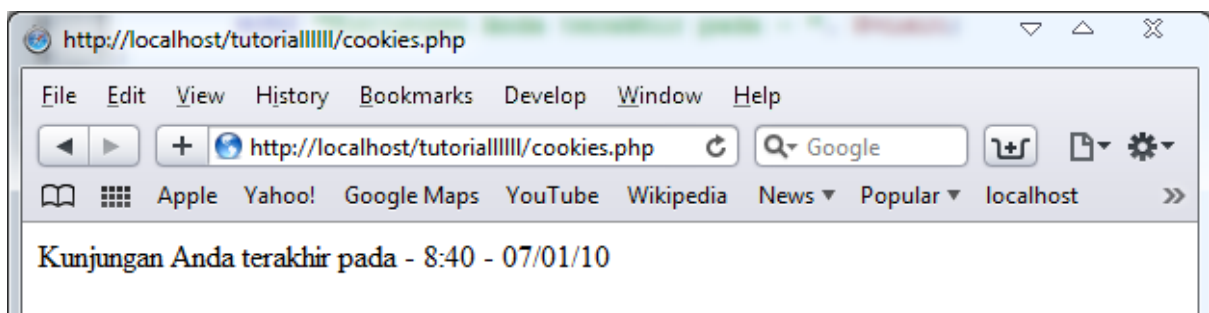
```
<?php
if(isset($_COOKIE['KunjunganTerakhir']))
{
    $visit = $_COOKIE['KunjunganTerakhir'];
    echo "Kunjungan Anda terakhir pada - ". $visit;
}
else
    echo "Anda sudah 2 jam lebih tidak mengunjungi web ini";

?>
```

Hasil tampilan dari perintah diatas (saat cookies tidak aktif) adalah :



Sedangkan apabila cookies masih aktif, maka akan ditampilkan :



Latihan

Dengan menggunakan table user yang ada pada database gunakan username dan password pada session user untuk keamanan halaman web dan sertakan pula cookies untuk berakhirnya session user.