

MODUL PEMROGRAMAN WEB LANJUT



SESSION & COOKIE

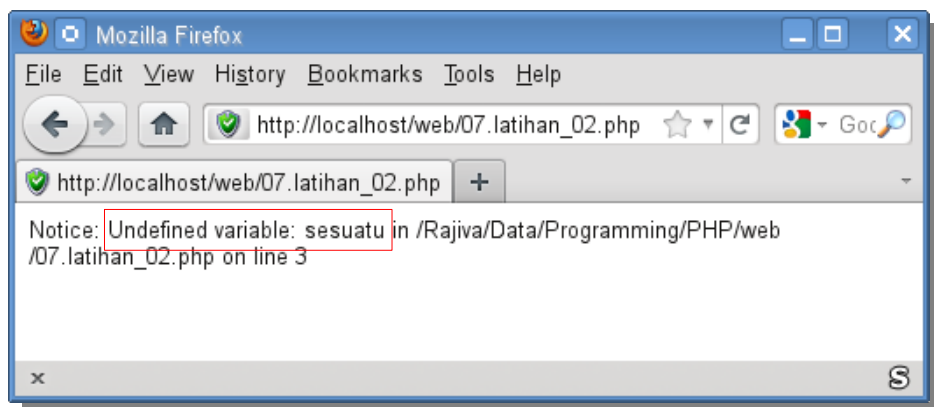
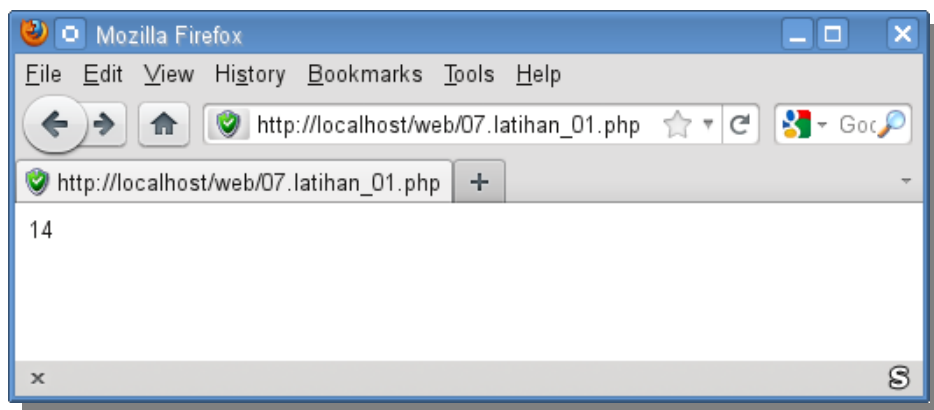
Koneksi protokol HTTP adalah koneksi *stateless*, yang artinya secara default protokol ini tidak memiliki fasilitas untuk menangani lebih dari satu transaksi. Sebagai contoh : jika suatu variabel dideklarasikan pada program A, maka variabel tersebut tidak akan dikenali pada program B. Perhatikan contoh program dibawah ini :

07.latihan_01.php

```
<?php  
  
$sesuatu = 14;  
echo $sesuatu;  
  
?>
```

07.latihan_02.php

```
<?php  
  
echo $sesuatu;  
  
?>
```



Dari kedua skrip latihan program diatas tampak bahwa variabel `$sesuatu` yang dideklarasikan pada program `07.latihan_01.php` tidak dikenali pada `07.latihan_02.php`. Oleh karena itu, diperlukan sebuah mekanisme agar sebuah variabel dapat dikenali dan dapat dioperasikan di program manapun.

Terdapat dua cara yang dapat digunakan untuk mengatasi permasalahan tersebut diatas, yaitu menggunakan session dan cookie. Terdapat banyak kasus yang dapat diselesaikan menggunakan session dan cookie, diantaranya yaitu : proses autentikasi, keranjang belanja pada aplikasi e-commerce, statistik, dll.

1. Session

`07.latihan_03.php`

```
<?php
session_start();

$_SESSION['sesuatu'] = 14;
echo $_SESSION['sesuatu'];

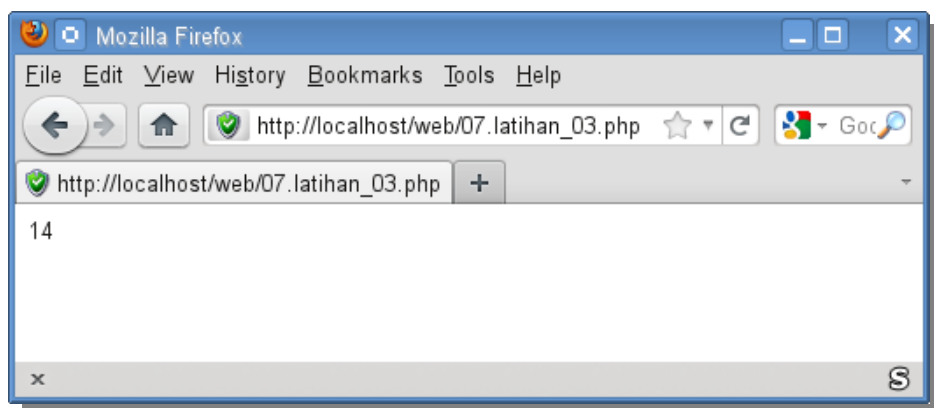
?>
```

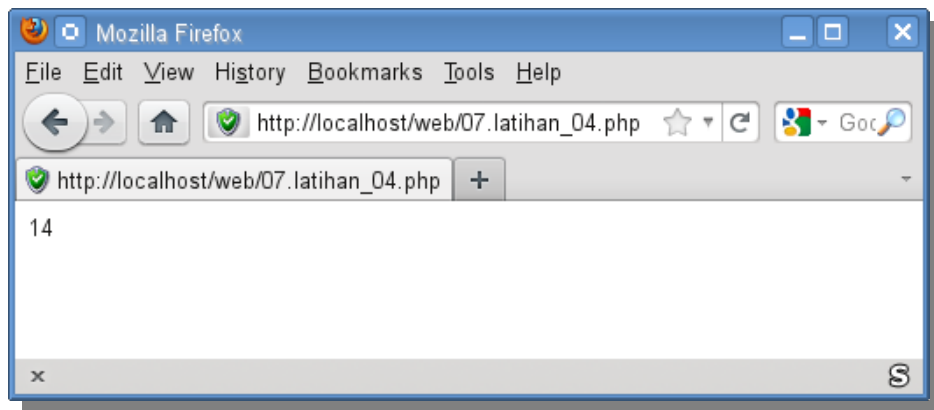
`07.latihan_04.php`

```
<?php
session_start();

echo $_SESSION['sesuatu'];

?>
```





Saat ini, variabel `$_SESSION['sesuatu']` dapat dikenali dan dioperasikan pada skrip program manapun melalui bantuan session. Sebelum digunakan, sebuah session terlebih dahulu harus melalui proses inisialisasi menggunakan fungsi `session_start()`. Fungsi ini akan memeriksa apakah seorang user saat ini telah memiliki sebuah session (masing-masing user memiliki session yang berbeda-beda).

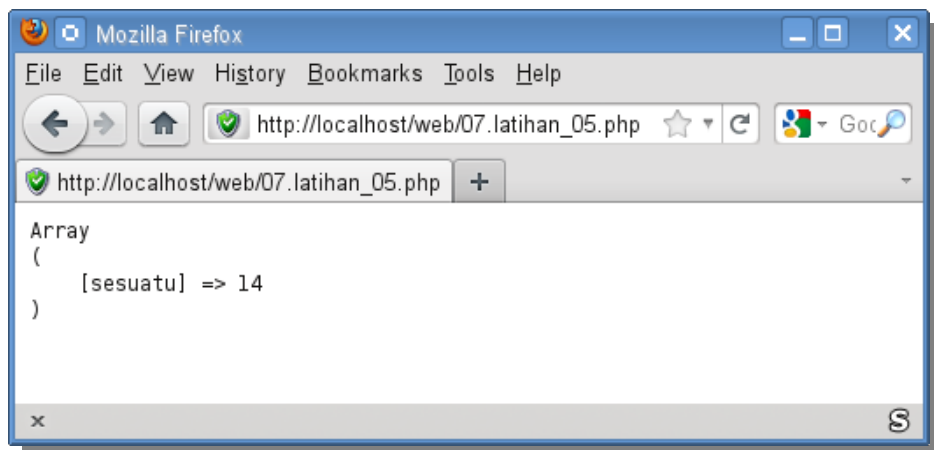
Jika sudah memiliki session, data pada session tersebut akan diimport ke program yang saat ini sedang dijalankan. Data pada session ini ditangani oleh PHP melalui array `$_SESSION`. Karena berbentuk array, kita dapat menggunakan fungsi `print_r()` untuk melihat deskripsi (*index* dan *value*) dari array tersebut.

07.latihan_05.php

```
<?php
session_start();

echo "<pre>";
print_r($_SESSION);
echo "</pre>";

?>
```



2. Cookie

Session dan cookie digunakan untuk tujuan yang sama, yaitu bagaimana agar sebuah variabel dapat dikenali dan dioperasikan pada semua program. Data yang tersimpan pada sebuah session disimpan di sisi server, sehingga user tidak dapat melihat data-data tersebut secara langsung.

Berbeda dengan session, data pada cookie diletakkan di sisi client, sehingga user dapat melihat bahkan memodifikasi dan menghapus cookie tersebut. Hal tersebut sering membuat penyimpanan data menggunakan cookie menjadi tidak efektif, apalagi user dapat menonaktifkan penggunaan cookie melalui *setting* browser yang mengakibatkan penggunaan cookie menjadi sia-sia.

07.latihan_06.php

```
<?php  
  
setcookie('sesuatu', 14);  
  
?>
```

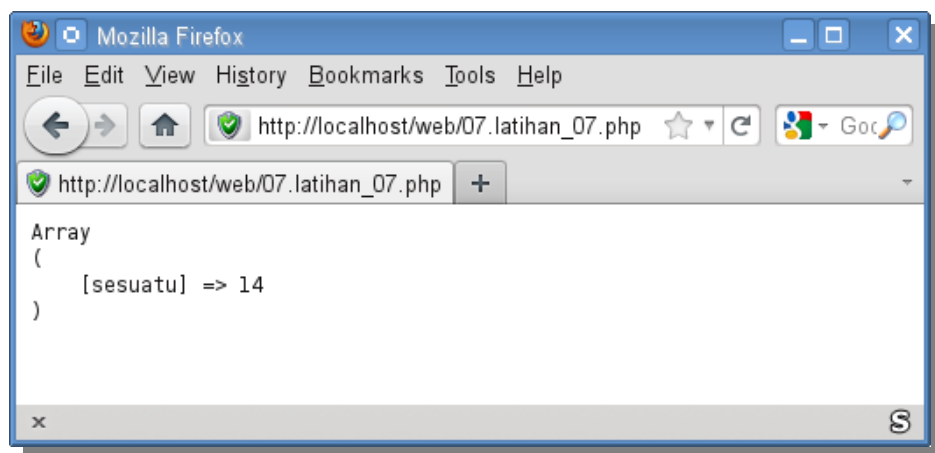
Setelah sebuah variabel cookie dideklarasikan, ia akan disimpan di sisi client dan selalu tersedia saat browser mengakses website (selama masa kadaluarsa cookie belum habis). Screenshot dibawah ini menunjukkan kepada kita mengenai informasi cookie yang tersimpan pada browser client (Firefox Preference → Privacy → remove individual cookies).



Data pada cookie ini ditangani oleh PHP melalui array `$_COOKIE`. Oleh karena berbentuk array, kita dapat menggunakan fungsi `print_r()` untuk melihat deskripsi (*index* dan *value*) dari array tersebut.

07.latihan_07.php

```
<?php  
  
echo "<pre>";  
print_r($_COOKIE);  
echo "</pre>";  
  
?>
```



3. Program Autentikasi

Salah satu program yang erat hubungannya dengan session dan cookie adalah aplikasi autentikasi. Pada contoh kali ini kita akan menggunakan session, karena dirasa lebih aman dan cocok untuk kasus autentikasi. Program autentikasi yang akan kita buat kali ini telah terintegrasi dengan basis data, sehingga nantinya contoh program ini dapat diterapkan secara langsung pada proyek aplikasi website. Pertama-tama kita persiapkan struktur basis datanya terlebih dahulu.

07.latihan_08.sql

```
CREATE TABLE user (  
    username VARCHAR (25) NOT NULL PRIMARY KEY,  
    password VARCHAR (35) NOT NULL  
);
```

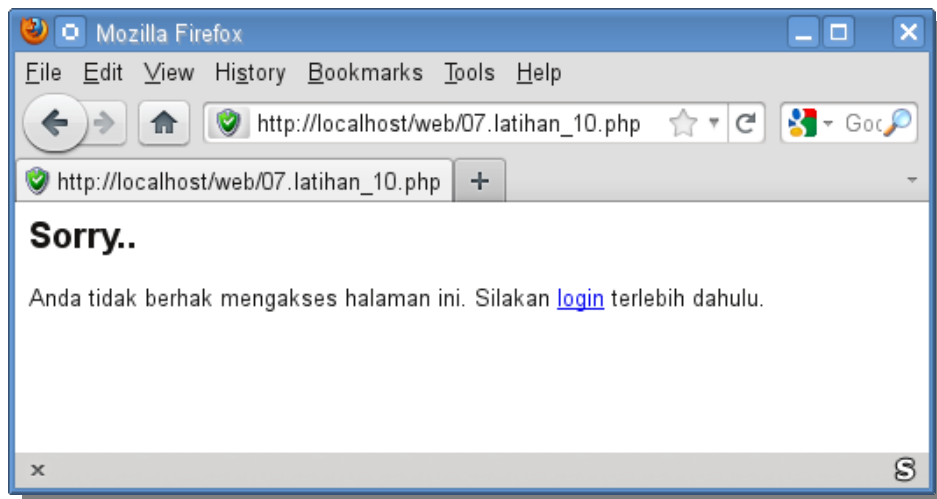
07.latihan_09.sql

```
INSERT INTO user (username,password) VALUES ('amikom','amikom0k3');
INSERT INTO user (username,password) VALUES ('juminten','adadeh');
INSERT INTO user (username,password) VALUES ('rajiva','123456');
```

07.latihan_10.php

```
<?php
session_start();

if(isset($_SESSION['username'])){ ?>
    <h2>Control Panel</h2>
    <p>
        Selamat datang user "<?php echo $_SESSION['username']; ?>".
        Klik <a href="07.latihan_13.php">disini</a> untuk logout.
    </p>
    <?php
} else { ?>
    <h2>Sorry..</h2>
    <p>
        Anda tidak berhak mengakses halaman ini.
        Silakan <a href="07.latihan_11.php">login</a> terlebih dahulu.
    </p>
    <?php
}
?>
```

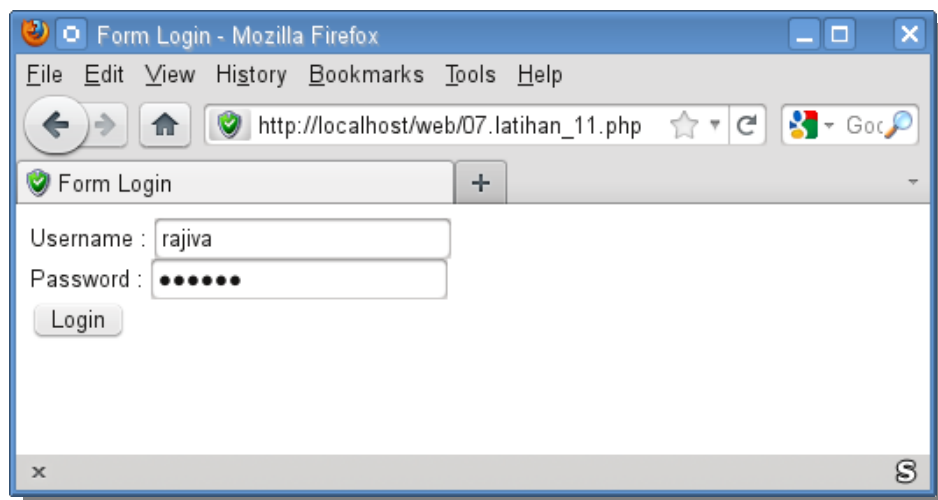


Pada skrip program 07.latihan_10.php diatas, akan dilakukan pemeriksaan apakah seorang user sudah ter-autentikasi melalui fungsi `isset($_SESSION['username'])`. Seperti telah diketahui sebelumnya, fungsi `isset()` digunakan untuk memeriksa apakah sebuah variabel sudah pernah dideklarasikan sebelumnya.

Dalam hal ini, variabel `$_SESSION['username']` akan dideklarasikan hanya jika user telah melakukan login dengan username dan password yang tepat. Jika belum terautentikasi, user akan dibawa ke halaman login.

07.latihan_11.php

```
<html>
  <head>
    <title>Form Login</title>
  </head>
  <body>
    <form action="07.latihan_12.php" method="post">
      Username : <input type="text" name="txtUsername" />
      <br />
      Password : <input type="password" name="txtPassword" />
      <br />
      <input type="submit" value="Login" />
    </form>
  </body>
</html>
```



Program diatas menyediakan form login yang terdiri dari input dengan tipe text, input bertipe password dan sebuah tombol submit. Data username dan password akan dikirimkan ke skrip 07.latihan_12.php menggunakan metode POST untuk diolah lebih lanjut.


```
<?php
session_start();

include("koneksi.php");

$sql = "SELECT username FROM user ";
$sql.= "WHERE username='". $_POST['txtUsername']."' ";
$sql.= "AND password='". $_POST['txtPassword']."' ";

$hasil = mysql_query($sql) or exit("Error query : <b>".$sql."</b>.");

if(mysql_num_rows($hasil)>0){

    $data = mysql_fetch_array($hasil);
    $_SESSION['username'] = $data['username'];

    header("Location:07.latihan_10.php");
    exit();
} else { ?>
    <h2>Sorry..</h2>
    <p>
        Username atau password salah.
        Klik <a href="07.latihan_11.php">disini</a> untuk kembali login.
    </p>
    <?php
}

?>
```

Pada skrip program diatas, data login yang diinputkan oleh user diolah sebagai parameter dari sebuah query. Query tersebut akan mencocokkan data login yang terdiri dari username dan password dengan data yang ada pada basis data. Jika data yang diinputkan cocok, maka variabel `$_SESSION['username']` akan segera dideklarasikan dan menampung data username tadi.

Fungsi `mysql_num_rows()` berguna untuk menghitung baris data yang dihasilkan dari sebuah query. Dalam kasus ini, fungsi tersebut akan mengembalikan nilai 0, jika data username dan password yang diinputkan tidak cocok dengan data manapun pada basis data.

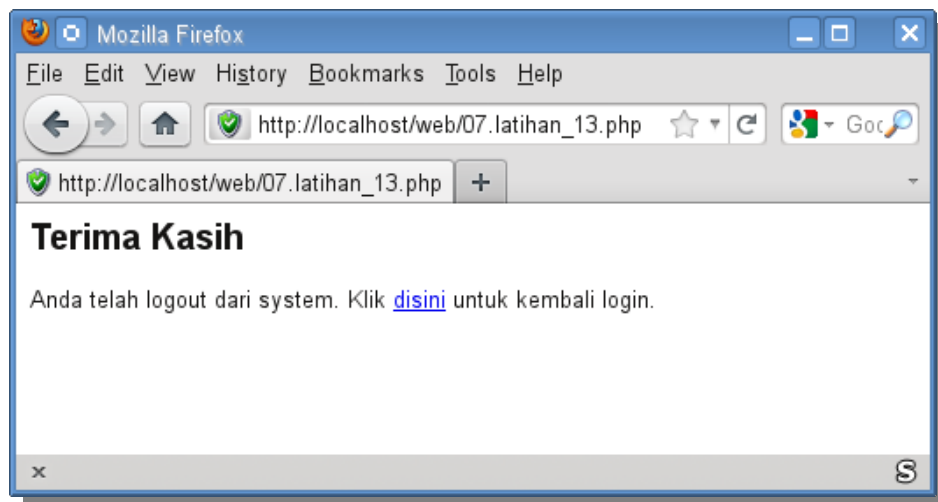
Jika proses autentikasi berhasil, selanjutnya user akan dibawa (*redirect*) ke halaman 07.latihan_10.php melalui statement `header("Location:07.latihan_10.php");`.

07.latihan_13.php

```
<?php
session_start();

unset($_SESSION['username']);

?>
<h2>Terima Kasih</h2>
<p>
    Anda telah logout dari system.
    Klik <a href="07.latihan_11.php">disini</a> untuk kembali login.
</p>
```



Skrip program diatas berguna untuk menghapus data session dari seorang user melalui fungsi `unset()`, fungsi ini akan menghapus variabel `$_SESSION['username']` dari web server.

Terdapat cara lain yang bisa digunakan untuk menghapus data session dari seorang user, yaitu menggunakan fungsi `session_destroy()`. Kekurangannya, jika seorang user memiliki banyak data session, ia akan menghapus semua data session dari user tersebut. Berbeda dengan fungsi `unset()` yang hanya menghapus data session tertentu saja.

DAFTAR PUSTAKA

1. Mulyana, Y.B. 2004. *Trik Membangun Situs Menggunakan PHP dan MySQL*. Jakarta: PT Elex Media Komputindo.
2. Buzton, Toby. 2002. *PHP By Example*. Indianapolis, Indiana: Que.
3. Muhardin, Endy. 2003. *PHP Programming Fundamental dan MySQL Fundamental*. IlmuKomputer.Com
4. Rasmus, Lerdorf., Dkk. 2003. *PHP 5 Manual*. PHP Documentation Group.