

Bab 9 Session dan Cookies

- ❖ Session
- ❖ Cookies

Session

Session merupakan hal yang cukup penting dalam aplikasi berbasis web. Dengan session memungkinkan programmer menyimpan informasi user secara semi-permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di server, jadi tidak bisa diakses secara langsung oleh client.

Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login. Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman. Misalnya saja, untuk melihat halaman kotak surat pada email, kita harus login terlebih dahulu. Dalam proses login antara lain akan terjadi pembuatan suatu session yang akan dibawa oleh user di setiap halaman. Di halaman kotak surat, session tersebut akan diperiksa. Jika session benar maka user dipersilahkan membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta untuk login terlebih dahulu. Itulah sebabnya, user tidak bisa mengakses halaman kotak surat secara langsung tanpa melakukan login.

Dalam penanganan session terdapat beberapa proses yang perlu diperhatikan :

- Proses pembuatan session
- Proses pemeriksaan session
- Proses penghapusan session

Selanjutnya bagaimana session itu sendiri dijalankan? Agar proses penyimpanan dalam session berjalan, PHP melakukan beberapa hal berikut ini :

1. PHP meng-generate (membentuk) sebuah ID session.
ID session ini merupakan sejumlah deret angka random yang unik untuk setiap user dan hampir tidak mungkin bisa ditebak. ID session disimpan oleh PHP di dalam variabel sistem PHP dengan nama PHPSESSID
2. PHP menyimpan nilai yang akan Anda simpan dalam session di dalam file yang berada di server.
Nama file tempat penyimpanan session tersebut sesuai (sama) dengan ID session. File disimpan dalam suatu direktori yang ditunjukkan oleh session.save_path dalam file php.ini.
3. PHP melempar ID session ke setiap halaman.
4. PHP mengambil nilai session dari file session untuk setiap halaman session.

Program 9.1

Nama File : session01.php

Deskripsi : Program menciptakan session.

```

1 <?php
2 /*****
3 Nama file : session01.php
4 Halaman ini merupakan halaman contoh penciptaan session.
5 Perintah session_start() harus ditaruh di perintah pertama
6 tanpa spasi di depannya. Perintah session_start() harus ada
7 pada setiap halaman yang berhubungan dengan session
8 *****/
9 session_start();
10 if (isset($_POST['Login'])) {
11     $user = $_POST['user'];

```

```

12     $pass = $_POST['pass'];
13     //periksa login
14     if ($user == "achmatim" && $pass = "123") {
15         //menciptakan session
16         $_SESSION['login'] = $user;
17         //menuju ke halaman pemeriksaan session
18         echo "<h1>Anda berhasil LOGIN</h1>";
19         echo "<h2>Klik <a href='session02.php'>di sini
(session02.php)</a>
untuk menuju ke halaman pemeriksaan session";
20     }
21 } else {
22 }?>
23 ?>
24 <html>
25     <head>
26         <title>Login here...</title>
27     </head>
28     <body>
29         <form action="" method="post">
30             <h2>Login Here...</h2>
31             Username : <input type="text" name="user"><br>
32             Password : <input type="password" name="pass"><br>
33             <input type="submit" name="Login" value="Log In">
34         </form>
35     </body>
36 </html>
37 <? } ?>

```

Penjelasan Program 9.1.

Pada program di atas terdapat fungsi **session_start()** yang berada pada baris ke-9. Fungsi `session_start()` berfungsi untuk memulai sebuah session. Fungsi harus dipanggil pertama kali dalam suatu halaman PHP, artinya fungsi `session_start()` harus dipanggil sebelum sesuatu tampil di layar, walaupun hal itu hanya berupa spasi. Jika sebelum `session_start()` terdapat output yang ditampilkan maka akan terjadi error seperti pada gambar 9.1 berikut ini :



Gambar 9.1. Error yang terjadi karena sebelum `session_start()` sudah ada output

Selanjutnya, program 9.1 akan menampilkan sebuah form login sederhana di layar. Form terdiri dari form inputan username, password dan sebuah tombol login. Jika tombol login ditekan maka pemeriksaan kondisi pada baris 10 akan bernilai TRUE dan perintah yang ada di blok **if** akan dieksekusi. Baris 11 dan 12 merupakan perintah untuk mengambil nilai pada form inputan username dan password. Nilai username dan password tersebut diperiksa pada baris 14. Pemeriksaan username dan password ini, untuk aplikasi web yang sudah

menggunakan database, bisa diganti dengan pemeriksaan username dan password ke tabel database.

Pada baris 16, terdapat perintah sebagai berikut :

```
$_SESSION['login'] = $user;
```

Baris perintah tersebut merupakan perintah untuk membuat session baru (*create session*) dimana nama session adalah "login" dan isi dari session adalah \$user. \$_SESSION merupakan sebuah variabel array global yang didefinisikan oleh PHP, jadi variabel ini harus ditulis dengan huruf kapital (*upper-case*).

Selanjutnya, pada baris 18-20 akan ditampilkan pesan bahwa login berhasil dan juga sebuah link menuju ke halaman pemeriksaan session (session02.php, program 9.2).

Program 9.2

Nama File : session02.php

Deskripsi : Program pemeriksaan session.

```

1  <?php
2  /*****
3  Halaman ini merupakan contoh halaman pemeriksaan session. Peme-
4  riksaan session biasanya dilakukan jika suatu halaman memiliki
5  akses terbatas. Misalnya harus login terlebih dahulu.
6  *****/
7  session_start();
8
9  //pemeriksaan session
10 if (isset($_SESSION['login'])) { //jika sudah login
11     //menampilkan isi session
12     echo "<h1>Selamat Datang ". $_SESSION['login'] ."</h1>";
13     echo "<h2>Halaman ini hanya bisa diakses jika Anda sudah
login</h2>";
14     echo "<h2>Klik <a href='session03.php'>di sini
(session03.php)</a> untuk LOGOUT</h2>";
15 } else {
16     //session belum ada artinya belum login
17     die ("Anda belum login! Anda tidak berhak masuk ke halaman
ini.Silahkan login <a href='session01.php'>di sini</a>");
18 }
19 ?>

```

Penjelasan Program 9.2

Program 9.2 di atas diawali dengan pemanggilan fungsi **session_start()**. Hal ini berarti dalam halaman program ini terdapat operasi session. Pada baris 10 program di atas, terdapat pemeriksaan keberadaan variabel \$_SESSION['login'] dengan menggunakan fungsi **isset()**. Pemeriksaan kondisi ini akan bernilai TRUE jika \$_SESSION['login'] sudah terbentuk. Jika bernilai TRUE maka blok program baris 11-14 yang akan dieksekusi, dan sebaliknya, jika FALSE maka blok program baris 16-17 yang akan dijalankan. Blok program baris 11-14 hanya bisa diakses saat user sudah login dengan benar. Dengan kata lain, bagian tersebut merupakan bagian yang bersifat private (terbatas).

Untuk menampilkan isi variabel session dapat dilakukan seperti pada baris 12 program di atas. Cukup dengan meng-echo variabel \$_SESSION['login'].

Jika user belum login (tidak melewati halaman login) atau langsung mengakses halaman ini, maka akan ditampilkan pesan bahwa user tidak berhak masuk ke halaman ini. Fungsi **die()** pada baris 17 digunakan untuk menampilkan suatu pesan sekaligus men-*terminate* program, artinya perintah selain die() akan diabaikan.