

## Fitur Login/Logout Halaman Web Dengan PHP

Banyak sekali aplikasi di internet yang menggunakan fitur login dan logout. Fungsinya tentu saja membatasi akses terhadap layanan tertentu yang disediakan pada situs tersebut. Hanya anggota yang telah terdaftar saja yang diijinkan mengakses layanan-layanan khusus tersebut. Kali ini PCplus akan membahas bagaimana membuat aplikasi yang memiliki fitur login/logout. Yang dibahas di sini adalah halaman login dan logoutnya saja, sedangkan isi aplikasi yang terletak di dalamnya bisa Anda kembangkan sendiri.

Dalam pembuatan aplikasi ini kita membutuhkan database yang akan menyimpan data-data dari para anggota yang telah terdaftar. Buatlah database dengan MySQL sebagai berikut:

```
mysql> create database daftar;
```

```
mysql> create table anggota(  
    -> uname varchar(20),  
    -> pswd varchar(32));
```

Database tersebut terdiri dari satu tabel bernama “anggota” dan tabel tersebut memiliki 2 buah field yaitu uname untuk menyimpan user name dan pswd untuk menyimpan password. Mengapa field pswd ditentukan untuk memiliki 32 karakter? Kita akan bahas nanti.

Sekarang mula-mula kita akan buat dahulu halaman pendaftaran anggota. Skripnya diberikan pada listing 1.

### Listing 1. Halaman pendaftaran

```
<HTML>  
<HEAD>  
<TITLE> Sign Up </TITLE>  
</HEAD>  
  
<BODY>  
Isilah username dan password yang Anda inginkan <BR>  
  
<PRE>  
<FORM NAME="frmSignUp" METHOD="post" ACTION="process.php">  
Username      : <INPUT TYPE="text" NAME="uname">  
Password      : <INPUT TYPE="password" NAME="pswd">  
  
<INPUT TYPE="submit" VALUE="Submit">  
</FORM>  
  
</BODY>  
</HTML>
```

Simpanlah skrip ini dengan nama signup.php. Diberi nama signup.htm pun boleh karena pada kenyataannya tidak ada skrip php pada file ini. Jika dijalankan pada browser hasilnya akan nampak seperti gambar 1.

Berikutnya kita akan membuat file process.php yang akan memproses pendaftaran kita tadi. Skripnya diberikan pada listing 2.

### Listing 2. Halaman proses

```

<HTML>
<HEAD>
<TITLE> Processing your Account </TITLE>
</HEAD>

<BODY>
<?
$uname = $_POST['uname'];
$pswd = md5($_POST['pswd']);
$host = "localhost";
$user = "root";
$pwd = "rahasia";
$db = "daftar";
$stop = 0;
$sql = "select uname from anggota";
$conn = @mysql_connect($host,$user,$pwd) or
        die("Koneksi gagal : " . mysql_error());
mysql_select_db($db);
$qry = mysql_query($sql) or
        die("Query salah : " . mysql_error());
while ($row = mysql_fetch_array($qry)) {
    if ($uname == $row['uname']) {
        echo "Username $uname was already chosen by someone else <BR>";
        echo "Go back and choose another username";
        $stop = 1;
        break;
    }
}

if ($stop==0) {
    $sql = "insert into anggota values ('$uname','$pswd')";
    $qry = mysql_query($sql) or
        die("Query salah : " . mysql_error());
    echo "Your account is successfully created <BR>";
    echo "<A HREF='login.php'> Click here </A> to login";
}
?>
</BODY>
</HTML>

```

Perhatikan bahwa pada baris kedua skrip php (setelah bagian <BODY>) terdapat fungsi md5. Fungsi md5 merupakan fungsi yang menghitung md5 hash dari suatu string. (Mengenai hal ini Anda dapat mendapatkan informasi lebih detil di <http://www.faqs.org/rfcs/rfc1321.html>). Md5 hash menghasilkan 32 karakter dalam format heksadesimal. Itulah sebabnya mengapa field pswd dibuat bertipe data karakter dengan panjang 32 karakter. Dengan md5 hash ini password Anda akan disimpan dalam bentuk yang sangat sulit ditebak.

Pada skrip ini mula-mula diperiksa apakah username yang dipilih telah ada di dalam database. Jika telah ada, maka php akan memberikan peringatan bahwa username tersebut telah ada dan Anda dipersilakan membuat username baru. Jika belum ada, maka informasi username dan password Anda akan didaftarkan ke dalam database dan Anda akan diberitahu bahwa account Anda telah dibuat. Bandingkan dengan gambar 2 dan gambar 3.

Setelah proses sign up berhasil, kini kita akan membuat halaman untuk login. Skripnya

diberikan pada listing 3.

Listing 3. Halaman login

```
<HTML>
<HEAD>
<TITLE> Login </TITLE>
</HEAD>

<BODY>
Registered user log in: <BR>

<PRE>
<FORM NAME="frmLogIn" METHOD="post" ACTION="main.php">
Username      : <INPUT TYPE="text" NAME="uname">
Password      : <INPUT TYPE="password" NAME="pswd">

<INPUT TYPE="submit" VALUE="Submit">
</FORM>
</PRE>
New user please register
<A HREF="signup.php"> here </A>
</BODY>
</HTML>
```

Simpanlah file ini dengan nama login.php atau login.htm. Jika dijalankan pada browser hasilnya akan nampak seperti gambar 4. Skrip ini akan diolah pada file main.php. Nah, skrip dari file main.php itu sendiri diberikan pada listing 4.

Listing 4. Pengolah login

```
<HTML>
<HEAD>
<TITLE> Main Page </TITLE>
</HEAD>

<BODY>
<?
$uname = $_POST['uname'];
$pswd = md5($_POST['pswd']);
$host = "localhost";
$user = "root";
$pwd = "rahasia";
$db = "daftar";
$stop = 0;
$sql = "select * from anggota where uname='$uname'";
$conn = @mysql_connect($host,$user,$pwd) or
        die("Koneksi gagal : " . mysql_error());
mysql_select_db($db);
$qry = mysql_query($sql) or
        die("Query salah : " . mysql_error());
$num = mysql_num_rows($qry);
```

```

$row = mysql_fetch_array($qry);

if ($num==0) {
    echo "Username not found <BR>";
    echo "Go back and try another username";
} else {
    if ($pswd!=$row['pswd']) {
        echo "Your password is incorrect <BR>";
        echo "Go back and type the correct password";
    } else {
        echo "You are successfully logged in";
    }
}
?>
</BODY>
</HTML>

```

Pada prinsipnya skrip main.php ini mengolah data-data dari halaman login. Langkah pertama yang dilakukan tentu saja adalah koneksi ke database. Setelah itu mencocokkan username yang ada di database, jika tidak cocok Anda diminta memasukkan username kembali. Jika username cocok dengan record yang ada pada database, diperiksa passwordnya. Jika passwordnya cocok, maka Anda akan dapat login, jika tidak Anda diminta untuk mengisi password yang tepat. Mudah bukan?

Namun sebenarnya skrip tersebut di atas belumlah selesai. Mengapa? Sebab skrip seperti di atas (sekali pun telah diberi “pagar” di sana-sini) tetap mudah ditembus oleh siapapun, bahkan Anda tidak perlu belajar menjadi hacker untuk dapat menembusnya.

Contoh mudah, isilah form login dengan data yang ada pada database Anda. Dengan sendirinya Anda akan login dan masuk ke halaman main.php yang bertuliskan “You are successfully logged in”. Sekarang tekan tombol “back” browser Anda, lalu setelah Anda tiba kembali ke halaman login, tekan tombol “forward”. Apa yang terjadi? Ya, Anda akan kembali masuk ke halaman main.php dan login dengan sukses. Itu artinya jika Anda lalai dan meninggalkan komputer Anda, orang lain dengan mudah akan masuk menggunakan account Anda.

Lalu bagaimana solusinya? Salah satu caranya adalah dengan menambahkan session ke dalam aplikasi tersebut. Dengan memeriksa apakah suatu variabel session pernah didaftarkan atau belum, maka aplikasi tersebut menjadi lebih aman. Dengan menggunakan solusi ini, fitur login/logout pada aplikasi kita akan terpecah menjadi lima file yang diberikan pada listing 5 hingga listing 9.

Listing 5. File login.php

```

<HTML>
<HEAD>
<TITLE> Login </TITLE>
</HEAD>

<BODY>
Registered user log in: <BR>
<PRE>
<FORM NAME="frmLogIn" METHOD="post" ACTION="cekpswd.php">
Username      : <INPUT TYPE="text" NAME="uname">
Password      : <INPUT TYPE="password" NAME="pswd">
<INPUT TYPE="submit" VALUE="Submit">
</FORM>
</PRE>

```

```
New user please register
<A HREF="signup.php"> here </A>

</BODY>
</HTML>
```

File login.php tersebut hanyalah form login biasa seperti gambar 4. Parameter ACTION dari tag <FORM> diisi dengan cekpswd.php, karena isian dari form ini akan diperiksa oleh file cekpswd.php.

#### Listing 6. File cekpswd.php

```
<?
session_start();
$uname = $_POST['uname'];
$pswd = md5($_POST['pswd']);
$host = "localhost";
$user = "root";
$pwd = "rahasia";
$db = "daftar";
$sql = "select * from anggota where uname='$uname'";
$conn = @mysql_connect($host,$user,$pwd) or
        die("Koneksi gagal : " . mysql_error());
mysql_select_db($db);
$qry = mysql_query($sql) or
        die("Query salah : " . mysql_error());
$num = mysql_num_rows($qry);
$row = mysql_fetch_array($qry);
if ($num==0 OR $pswd!=$row['pswd']) {
    header("Location: sorry.php");
} else {
    $_SESSION['login']=1;
    header("Location: main.php");
}
?>
```

File cekpasswd.php digunakan untuk memeriksa username dan password yang dimasukkan pada file login.php. Pada file ini mulai diinisiasi sebuah session. Langkah selanjutnya adalah mencocokkan username dan password dengan record yang terdapat pada database. Jika username dan password itu tidak cocok – bisa usernamenya tidak terdaftar atau passwordnya yang salah – maka skrip tersebut akan mengarahkan langkahnya menuju ke file sorry.php, tapi jika benar maka arahnya akan menuju ke file main.php dan sebuah variabel session diinisiasi. (Pada skrip kita variabel tersebut diberi nama \$\_SESSION['login']). Yang berfungsi sebagai “pengatur arah” adalah fungsi header(). Fungsi header() akan memberikan output berupa HTTP header yang harus diberikan sebelum ada tag HTML apapun. Itu sebabnya file cekpswd.php hanya terdiri dari skrip php saja dan tidak memiliki tag HTML. (Kalaupun toh ada, itu harus dibawah skrip php yang mengeluarkan HTTP header). Untuk informasi lebih lanjut mengenai HTTP header Anda dapat mengunjungi <http://www.w3.org/Protocols/rfc2616/rfc2616>. Yang jelas fungsi header() pada skrip kita di atas digunakan untuk redirection, yaitu mengarahkan browser untuk memanggil URL lain. Oleh karena output file cekpswd.php hanya HTTP header saja, maka file ini tidak memiliki “penampakan fisik” pada browser.

Listing 7. File main.php

```
<?
session_start();
if(!isset($_SESSION['login'])) {
    include("login.php");
} else {
?>

    <HTML>
    <HEAD>
    <TITLE> Main Page </TITLE>
    </HEAD>

    <BODY>
    You are successfully logged in <BR>
    You can access this application <BR> <BR>
    <A HREF="logout.php"> Log Out </A>
    </BODY>
    </HTML>

<?
}
?>
```

File main.php adalah file yang akan dipanggil apabila username dan password yang dimasukkan pada halaman login.php berhasil melewati “satpam” cekpswd.php. Lihat gambar 5. File inilah yang nantinya bisa dikembangkan untuk menyusun aplikasi-aplikasi web yang diperuntukkan bagi mereka yang login. Sebelumnya pada bagian awal dari file ini diperiksa terlebih dahulu apakah variabel \$\_SESSION['login'] sudah pernah diinisiasi dengan fungsi isset(). Seandainya belum, itu artinya user yang mengakses halaman main.php belum login. Jika demikian, maka orang tersebut akan “diangkut” menuju ke halaman login.php untuk login terlebih dahulu. Apakah mungkin user yang belum login dapat langsung menuju ke halaman main.php? Mungkin saja apabila ia langsung mengetikkan “http://bla..bla..bla/main.php” di bagian address browser. Oleh karena itu harus diberi “satpam” agar user tersebut tidak main selonong saja.

Jika seandainya aplikasi Anda terdiri dari beberapa halaman, maka pada prinsipnya setiap halaman harus diberi skrip sebagai berikut:

```
<?
session_start();
if(!isset($_SESSION['login'])) {
    include("login.php");
} else {
?>
```

Di bawah skrip ini baru diberikan tag-tag HTML atau skrip PHP lain yang menyusun aplikasi. Dengan demikian setiap halaman selalu memeriksa apakah variabel session \$\_SESSION['login'] telah diinisiasi. Jika belum, maka redirection akan beraksi dan “menendang” user untuk kembali ke halaman login.php.

Listing 8. File sorry.php

```
<HTML>
<HEAD>
<TITLE> Sorry </TITLE>
```

```
</HEAD>

<BODY>
Your user name or password is incorrect <BR>
Sorry, you are not allowed to access this page
</BODY>
</HTML>
```

File ini akan muncul apabila user salah memasukkan username atau password pada halaman login. Lihat gambar 6. User tersebut dapat mengklik tombol back di browser untuk kembali ke halaman login dan memasukkan username dan password yang benar.

#### Listing 9. File logout.php

```
<?
session_start();
unset($_SESSION['login']);
session_destroy();
header("Location: login.php");
?>
```

File ini digunakan untuk melakukan proses logout. Pada file ini variabel `$_SESSION['login']` dihilangkan sekaligus bersama dengan sessionnya. Langkah selanjutnya adalah mengarahkan aplikasi ke halaman login.php. Setelah session dihilangkan, maka user harus login lagi untuk dapat mengakses aplikasi tersebut.

Pada dasarnya fungsi `header()` tidak harus mengarahkan aplikasi ke file login.php. Bisa saja Anda membuat sebuah file lain yang bernama `goodbye.php` misalnya yang isinya adalah ucapan perpisahan.

Nah, selamat belajar mengembangkan aplikasi dengan menggunakan fitur login/logout.