

MODUL 6- JOBSHEET (Kegiatan Belajar 9)**Penggunaan Session untuk Autentifikasi User Level****PETUNJUK PENGGUNAAN MODUL**

1. Baca dan pahami **Dasar Teori** yang telah diberikan oleh guru pengajar
2. Kerjakan **Latihan-latihan** dan **Tugas Mandiri** yang ada di modul ini
3. Jawablah **Latihan-latihan** dan **Tugas Mandiri** pada **Lembar Jawaban**

KOMPETENSI DASAR

- 3.19 Menerapkan aplikasi web statefull
- 4.19 Membuat aplikasi web statefull
- 3.20 Mengevaluasi pengamanan data pada aplikasi web
- 4.20 Merevisi pengamanan data pada aplikasi web

TUJUAN

- ✓ Mampu menjelaskan konsep antar session dalam aplikasi web.
- ✓ Mampu menerapkan Penggunaan session dalam program aplikasi web.
- ✓ Mampu menerapkan Konsep kriptografi, plain text, chiper text, encrypted text
- ✓ Mampu menerapkan Konsep sistem pengamanan akses data

DASAR TEORI

Autentifikasi user adalah suatu mekanisme untuk memastikan apakah suatu user itu berhak masuk ke dalam sistem atau tidak. Dalam sebuah web aplikasi Autentifikasi user ini sangat penting untuk memberi batasan terhadap user dalam mengakses halaman/menu tertentu. Implementasinya adalah berupa login.

Dalam hal ini user yang berhak mengakses akan diberikan nama user tertentu beserta password. Nama user dan password akan disimpan dalam database. Bila terdapat nama user dan password yang tidakterdaftardalamdatabase,makausertersebuttidakberhakmengakses.

Mekanisme autentifikasi ini dimulai dengan proses pendaftaran user.

Pendaftaran User

1. User mengisi form pendaftaran sebagai legal user (user mengisi username dan password)
2. Sistem akan mengecek apakah username yang didaftarkan sudah ada yang memiliki atau belum
3. Jika sudah ada, user diminta mengisi kembali username yang lain beserta passwordnya. Sedangkan jika belum ada, data user ini akan disimpan dalam database, dengan password terenkripsi.
4. Level user akan menentukan hak akses terhadap halaman/menu tertentu

Proses Autentifikasi

1. User yang akan mengakses sistem diminta memasukkan username dan password (asli)
2. Sistem akan mencari password terenkripsi yang tersimpan dalam database berdasarkan username yang terdaftar
3. Sistem akan mencocokkan antara password asli terenkripsi yang diisikan user melalui form login, dengan password terenkripsi yang tersimpan dalam database
4. Jika password asli terenkripsi yang dikirim via form login ini sama dengan password terenkripsi yang ada dalam database, maka user tadi bisa masuk ke dalam sistem. Jika tidak sama, maka user tadi tidak berhak masuk ke sistem.

Proses autentifikasi user ini juga berfungsi untuk mengantisipasi terjadinya by pass dari user yang tidak berhak.

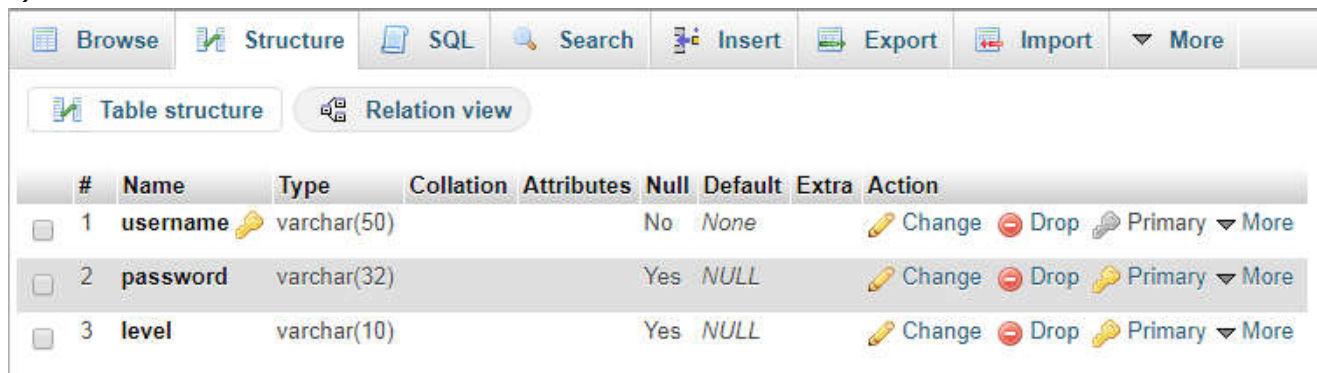
Dalam latihan ini juga kita gunakan proses pengamanan password dengan metode enkripsi MD5 dengan dikombinasi dengan pengacak. Mengapa kita tambahkan pengacak???. Dengan hanya menggunakan enkripsi MD5 saja kurang aman, sehingga perlu ditambahkan pengacak untuk memperoleh kombinasi pengacak yang berbeda dengan MD5.

LATIHAN :

Ikuti langkah-langkah latihan berikut ini dengan urut dan teliti :

Pertama, buat database **db_latihan** (kalau sudah ada tidak perlu dibuat lagi) Kemudian kita siapkan table **tb_user** terlebih dahulu.

```
CREATE TABLE tb_user (  
    username varchar(50),  
    password varchar(32),  
    level varchar(10),  
    PRIMARYKEY(username)  
)
```



The screenshot shows a database management interface with a toolbar at the top containing icons for Browse, Structure, SQL, Search, Insert, Export, Import, and More. Below the toolbar, there are two tabs: 'Table structure' (selected) and 'Relation view'. The main area displays a table structure for 'tb_user' with the following columns: #, Name, Type, Collation, Attributes, Null, Default, Extra, and Action. The table has three rows: 1. 'username' (varchar(50), No Null, None Default, Primary Key), 2. 'password' (varchar(32), Yes Null, NULL Default, Primary Key), and 3. 'level' (varchar(10), Yes Null, NULL Default, Primary Key). Each row has a 'Change' icon, a 'Drop' icon, and a 'Primary' icon.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	username	varchar(50)			No	None		Change Drop Primary More
2	password	varchar(32)			Yes	NULL		Change Drop Primary More
3	level	varchar(10)			Yes	NULL		Change Drop Primary More

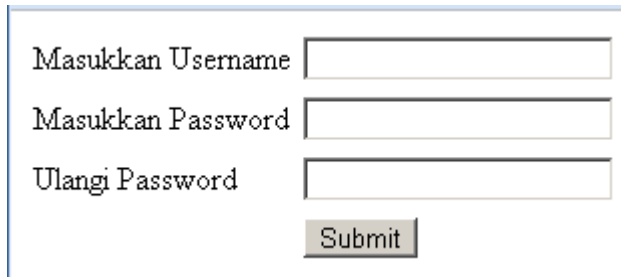
Field username kita gunakan *primary key*, karena username harus unik, tidak boleh ada username yang sama. Sedangkan password kita set tipe datanya varchar dengan panjang field minimal 32, karena password yang akan disimpan di sini adalah terenkripsi. Untuk enkripsi kita akan gunakan MD5() termodifikasi dengan menambahkan pengacak. hasil md5() ini adalah karakter dengan panjang 32.

Langkah kedua buat file **koneksi.php**

```
1 <?php  
2 //koneksi database  
3 $host="localhost";  
4 $user="root";  
5 $password="";  
6 $database="db_latihan";  
7 $db_koneksi=mysqli_connect($host,$user,$password,$database);  
8 if(!$conn){  
9     echo "KONEKSI DATABASE GAGAL!!!";  
10 }  
11 ?>
```

Langkah ketiga kita siapkan form pendaftaran usernya
register.php

```
1 <form method="post" action="submit_register.php">
2 <table border="0">
3 <tr>
4 <td>Masukkan Username </td>
5 <td><input name="username" type="text"></td>
6 </tr>
7 <tr>
8 <td>Masukkan Password </td>
9 <td><input name="pass1" type="password"></td>
10 </tr>
11 <tr>
12 <td>Ulangi Password </td>
13 <td><input name="pass2" type="password"></td>
14 </tr>
15 <tr>
16 <td>&nbsp;</td>
17 <td><input type="submit" name="Submit" value="Submit"></td>
18 </tr>
19 </table>
20 </form>
```



<form method="post" action="submit_register.php">

Pada form di atas terdapat dua isian untuk password. Isian password dilakukan dua kali untuk memastikan bahwa password yang dimasukkan adalah benar dari sisi ejaannya.

Selanjutnya kita buat script untuk pemrosesan registrasi user ini.
submit_register.php

```

1  <?php
2  $username = $_POST['username'];
3  $password1 = $_POST['pass1'];
4  $password2 = $_POST['pass2'];
5  $level = "user";//otomatis setiap register dianggap level user
6
7  // cek kesamaan password
8  if ($password1 == $password2)
9  {
10     include "koneksi.php";
11
12     // perlu dibuat sebarang pengacak
13     $pengacak = "p3ng4c4k";
14
15     // mengenkripsi password dengan md5() dan pengacak
16     $passmd = md5($pengacak . md5($password1));
17
18     // menyimpan username dan password terenkripsi ke database
19     $query = "INSERT INTO tb_user VALUES('$username', '$passmd', '$level')";
20     $hasil = mysqli_query($db_koneksi,$query);
21
22     // menampilkan status pendaftaran
23     if ($hasil) echo "User sudah berhasil terdaftar";
24     else echo "Username sudah ada yang memiliki";
25
26 }
27 else echo "Password yang dimasukkan tidak sama";
28 ?>

```

Anda boleh membuat pengacak yang lain, atau mengkombinasikan sendiri bentuknya, misalnya :

```
$password1=md5($pengacak.md5($pengacak.$password1.$pengacak).$pengacak)
```

yang jelas jangan menggunakan `$password1 = md5($password1);`

Isi form registrasi user dengan 2 nama (user), sehingga di database phpmyadmin terisi seperti ini :

Username : Adi → Password : admin

Username : Bayu → Password : user

	username	password	level
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Adi	d65c81cf756151229c67d150eac28fd6	admin
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Bayu	030c5f71c737e8c6af920ec09f82126d	user

Ganti level username **Adi** menjadi "**admin**" → melalui phpmyadmin

	username	password	level
  	Adi	d65c81cf756151229c67d150eac28fd6	admin
  	Bayu	030c5f71c737e8c6af920ec09f82126d	user

Buat skenario bahwa terdapat dua buah halaman :

1. Halaman **admin.php** → hanya dapat diakses oleh level admin
2. Halaman **user.php** → hanya dapat diakses oleh level user

Untuk dapat masuk ke dalam system membutuhkan autentifikasi user dan level.

Untuk proses autentifikasi, kita buat form loginnya.

login.php

```

1 <h3>Login User :</h3>
2 <form method="post" action="submit_login.php">
3   <table border="0">
4     <tr>
5       <td>Masukkan Username </td>
6       <td><input name="username" type="text"></td>
7     </tr>
8     <tr>
9       <td>Masukkan Password </td>
10      <td><input name="pass" type="password"></td>
11    </tr>
12    <tr>
13      <td>&nbsp;</td>
14      <td><input type="submit" name="Submit" value="Submit"></td>
15    </tr>
16  </table>
17 </form>

```

Login User :

Masukkan Username

Masukkan Password

<form method="post" action="submit_login.php">

submit_login.php

```
1 <?php
2 // menjalankan session : selalu diletakkan di awal
3 session_start();
4 include "koneksi.php";
5 $username = $_POST['username'];
6 $password = $_POST['pass'];
7
8 // mencari password terenkripsi berdasarkan username
9 $query = "SELECT * FROM tb_user WHERE username = '$username'";
10 $hasil = mysqli_query($db_koneksi,$query);
11 $data = mysqli_fetch_array($hasil);
12
13 //format pengacak harus sama dengan format di submit_register.php
14 $pengacak = "p3ng4c4k";
15
16 // cek kesesuaian password terenkripsi dari form login
17 // dengan password terenkripsi dari database
18 $passmd = md5($pengacak . md5($password));
19 if ($passmd == $data['password'])
20 {
21     // menyimpan username dan level ke dalam session
22     $_SESSION['level'] = $data['level'];
23     $_SESSION['username'] = $data['username'];
24
25     // menampilkan menu ke halaman akses
26     echo "<h2>Login sukses</h2>";
27     header('location: admin.php');
28 }
29 else echo "<h2>Login Gagal</h2>";
30 ?>
```

Pada saat cek kesesuaian password terenkripsi dari form dan dari database, harus menggunakan pengacak dan rule yang sama dengan proses mengenkripsi password sebelum disimpan ke database (**lihat kembali script submit_register.php**).

Selanjutnya, buat halaman admin.php

admin.php

```
1 <?php
2 // menjalankan session : selalu diletakkan di awal
3 session_start();
4 if (isset($_SESSION['level']) && isset($_SESSION['username']))
5 {
6     // jika level admin akan masuk ke halaman admin.php
7     if ($_SESSION['level'] == "admin")
8     {
9         echo "<h3>Ini Halaman Admin</h3>";
10        echo "Selamat Datang : <b>".$_SESSION['username']."</b><br>";
11        echo "Level Anda sebagai : <b>".$_SESSION['level']."</b><br>";
12        echo "ISI MENU-MENU ADMIN DISINI</b><br>";
13        echo "<a href='logout.php'>Logout</a> ";
14    }
15    // jika kondisi level user maka akan diarahkan ke halaman user.php
16    else if ($_SESSION['level'] == "user")
17    {
18        header('location:user.php');
19    }
20 }
21 // jika user belum terdaftar maka akan diarahkan ke halaman register.php
22 if (!isset($_SESSION['level']))
23 {
24     echo "Anda tidak boleh mengakses halaman ini tanpa : ";
25     echo "<a href='login.php'>Login</a><br>";
26     echo "<a href='register.php'>Belum punya User?</a>";
27 }
28 ?>
```

← → ↺ ↻ ⓘ localhost/aut/admin.php

Ini Halaman Admin

Selamat Datang : **Adi**
Level Anda sebagai : **admin**
ISI MENU-MENU ADMIN DISINI
[Logout](#)

Ket : hak akses terhadap halaman ditentukan oleh level user

- ✓ jika level admin akan masuk ke halaman **admin.php**
- ✓ jika kondisi level user maka akan diarahkan ke halaman **user.php**
- ✓ jika user belum terdaftar maka akan diarahkan ke halaman **register.php**

Buat halaman user.php

User.php

```
1 <?php
2 // menjalankan session : selalu diletakkan di awal
3 session_start();
4 if (isset($_SESSION['level']) && isset($_SESSION['username']))
5 {
6     // ini adalah halaman dengan level user
7     if ($_SESSION['level'] == "user")
8     {
9         echo "<h3>Ini Halaman User</h3>";
10        echo "Selamat Datang : <b>".$_SESSION['username']."</b><br>";
11        echo "Level Anda sebagai : <b>".$_SESSION['level']."</b><br>";
12        echo "ISI MENU-MENU USER DISINI</b><br>";
13        echo "<a href='logout.php'>Logout</a> ";
14    }else{
15        echo "Anda belum terdaftar sebagai User";
16    }
17 }
18 // jika user belum terdaftar maka akan diarahkan ke halaman register.php
19 if (!isset($_SESSION['level']))
20 {
21     echo "Anda tidak boleh mengakses halaman ini tanpa : ";
22     echo "<a href='login.php'>Login</a><br>";
23     echo "<a href='register.php'>Belum punya User?</a>";
24 }
25 ?>
```

← → ↻ 🏠 ⓘ localhost/aut/user.php

Ini Halaman User

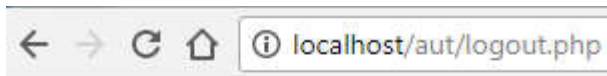
Selamat Datang : **Bayu**
Level Anda sebagai : **user**
ISI MENU-MENU USER DISINI
[Logout](#)

Perintah **isset()** digunakan untuk mengecek keberadaan suatu variabel (dalam hal ini variabel session untuk username). Perintah ini akan menghasilkan nilai **TRUE** jika variable yang di cek ada, dan **FALSE** jika variabel tidak ada.

Jangan lupa membuat script untuk **logout**. Konsep logout dalam autentifikasi adalah menghapus variabel session untuk **username** dan **level**.

logout.php

```
1 <?php
2 // menjalankan session : selalu diletakkan di awal
3 session_start();
4 // menghapus session username dan level
5 unset($_SESSION['username']);
6 unset($_SESSION['level']);
7 // menghapus semua session dari browser
8 session_destroy();
9 echo "<h1>Anda sudah logout</h1>";
10 echo "<p><a href='login.php'>Login Kembali</a></p>";
11 ?>
```



Anda sudah logout

[Login Kembali](#)

TUGAS MANDIRI :

Buatlah halaman web dengan **Tema Pendaftaran Ekstrakurikuler (04. KB 4 - Pertemuan 4)** atau **Tema Bebas** dengan menerapkan konsep session untuk autentifikasi user

Halaman Register User : untuk mendaftarkan user baru

Halaman Login User

Level admin : dapat melakukan/mengakses halaman insert, update, delete, dan view data

Level user : hanya dapat mengakses halaman view/melihat saja

Logout session destroy dan kembali ke halaman login lagi

Gunakan Layout css yang menarik dan bagus (boleh menggunakan bootstrap)

Catatan : Jika belum faham tentang penggunaan session untuk autentifikasi user dan penggunaan bootstrap, carilah literatur/tutorial di internet.