



Nama: Muhammad Wildan Ramadhana

Kelas: SIB 2E

Absen: 17

Topik

- Pengenalan konsep login
- Multi user login
- Pengenalan konsep cookies dan session

Tujuan

Mahasiswa diharapkan mampu:

1. Memahami konsep login *single user* dan *multiuser*
2. Memahami konsep dasar *cookies* dan penggunaannya
3. Memahami konsep dasar *session* dan penggunaannya
4. Mengimplementasikan konsep PHP dan database MySQL ke dalam web dinamis

Pendahuluan

Pengenalan Login

Saat ini fitur login menjadi fitur yang banyak digunakan di dalam pengembangan aplikasi web modern. Login diartikan sebagai proses masuk ke jaringan komputer. Seorang *user* harus memasukkan identitas akun minimal terdiri dari *username*/akun pengguna dan *password* untuk mendapatkan hak akses. Kemudian sistem akan melakukan pencocokan *username* dan *password* yang diinput dengan data *username* dan *password* yang tersimpan di dalam *database*.

Di dalam praktikum ini akan dibahas mengenai login *single* dan *multiuser*. Login *multiuser* adalah proses login yang memiliki validasi level dari seorang user saat login. Pengaturan validasi level ini berkaitan dengan hak akses yang akan diberikan kepada user yang login ke dalam sistem. Di dalam login *multiuser* terdapat istilah level akses. Level inilah yang akan menentukan tampilan mana atau aksi apa saja yang bisa dilakukan oleh *user* tersebut.

Contoh penerapan login *multiuser* dapat kita lihat pada web lms.jti.polinema.ac.id. Terdapat beberapa level akses untuk *user* yang akan memberikan hak akses yang berbeda dan dapat melakukan aksi yang berbeda di dalam web tersebut. Berikut adalah level akses dan aksi yang dapat dilakukan setiap *user* di dalam web lms.jti.polinema.ac.id:

1. *Student*

Level akses *student* dapat melakukan beberapa aksi yaitu, mengedit profil, melihat mata kuliah, mengunduh (*download*) materi, mengumpulkan (*upload*) tugas, dan mengerjakan kuis.

2. *Teacher*

Level akses *teacher* dapat melakukan beberapa aksi yaitu, mengedit topik/bab pada mata kuliah, menambahkan aktivitas (*upload* materi, *jobsheet* praktikum, *submission* tugas, kuis), dan melakukan penilaian pada tugas.

3. Administrator

Level akses *administrator* dapat melakukan beberapa yaitu, mendaftarkan *user*, membuat dan mengatur mata kuliah, dan mengatur *grouping* tiap mata kuliah.

Pengenalan Cookies

Cookies adalah sebuah nilai yang dikirimkan dan ditanamkan *server* pada komputer *client*. *Cookies* berupa berkas berukuran kecil umumnya kurang dari 100 bytes, sehingga tidak akan mempengaruhi kecepatan *browsing*. *Cookies* menampung informasi berkaitan dengan pengguna. *Cookies* berguna untuk memudahkan pengguna, seperti "mengingat" pengguna setiap kali mengunjungi website yang sama.

Informasi-informasi yang disimpan dalam *cookies* adalah data-data sederhana tentang kebiasaan pengguna. Contoh sederhananya saat pengguna mengunjungi sebuah website yang menawarkan tampilan dengan versi bahasa Inggris dan bahasa Indonesia. Apabila pengguna memilih bahasa Indonesia, *cookies* akan secara otomatis menyimpan konfigurasi tersebut. Konfigurasi tersebut akan diingat, sehingga apabila pengguna mengunjungi kembali website tersebut, pengguna tidak perlu memilih jenis bahasa lagi.

Cookies mempunyai umur, maksudnya setiap data yang disimpan dalam komputer pengguna suatu saat bisa hilang atau musnah. Berdasarkan umurnya *cookies* dibagi menjadi dua jenis, yaitu *session cookies* dan *persistent cookies*. *Session cookies* adalah jenis *cookies* yang tersimpan sementara dan akan hilang saat pengguna menutup *browser*. *Session cookies* ini biasanya digunakan pada fitur "keranjang belanja" di situs belanja online. *Persistent cookies* adalah jenis *cookies* yang tersimpan secara permanen di dalam komputer sampai pengguna memutuskan untuk menghapusnya. *Persistent cookies* umumnya digunakan pada fitur "remember me" saat login.

Pengenalan Session

Session adalah salah satu fasilitas yang ada pada PHP yang digunakan untuk menyimpan data sementara ke dalam variabel, sehingga data tadi dapat diakses oleh pengguna selama variabel *session* tersebut tidak dikosongkan atau dihilangkan. *Session* pada prinsipnya hampir sama dengan *cookies*, yang membedakan adalah posisi penyimpanan data. Jika *cookies* menyimpan data pada *client*, data *session* akan disimpan pada sisi *server*. Jadi *session* relatif lebih aman digunakan untuk menyimpan variabel nilai yang bersifat rahasia seperti *username* dan *password* pada saat login. Contoh penggunaan *session* diantaranya untuk menyimpan informasi login yang berlaku hanya dalam satu sesi dan menyimpan catatan pemesanan barang dalam sistem *e-commerce*/transaksi online. Tabel berikut menjelaskan perbandingan *cookies* dan *session*:

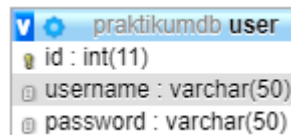
Tabel 1 Perbandingan *Cookies* dan *Session*

Jenis	Menyimpan Informasi	Lokasi Penyimpanan	Dapat dipanggil	Bersifat Global	Dapat dipanggil Lain Waktu	Dibatasi Browser	Dapat diedit User
<i>Cookies</i>	Ya	<i>Client</i>	Ya	Ya	Ya	Ya	Ya

<i>Session</i>	Ya	<i>Server</i>	Ya	Ya	Tidak	Tidak	Tidak
----------------	----	---------------	----	----	-------	-------	-------

Praktikum Bagian 1. Membuat Tabel User

Tahapan pertama dalam membuat fitur login adalah membuat database. Pada praktikum kali ini, kita akan menggunakan *database* yang telah dibuat pada praktikum sebelumnya, yaitu *database* yang bernama “prakwebdb”. Selanjutnya hal yang harus dilakukan adalah membuat tabel *user*. Tabel *user* minimal terdiri dari *username* dan *password*. Berikut adalah ketentuan *username* dan *password* yang dibuat di tabel *user*:



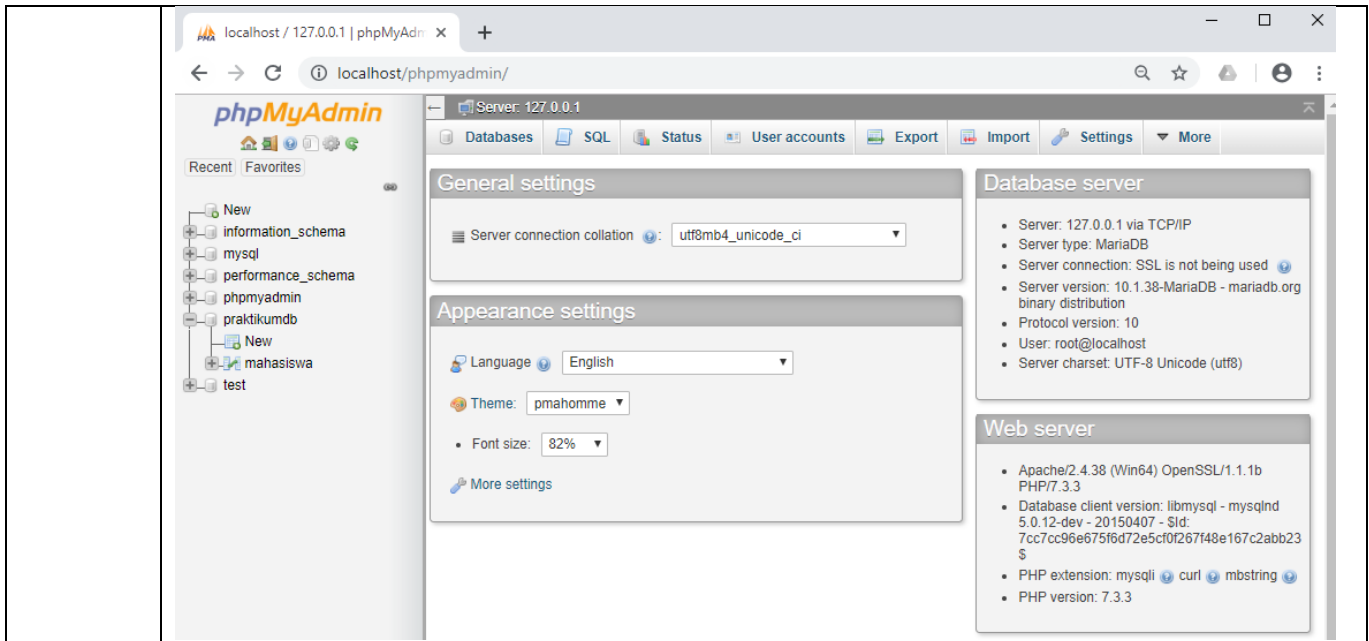
- tabel bernama *user*
- kolom tabel terdiri dari *id*, *username*, dan *password*
- *id* memiliki tipe data integer dan di atur sebagai *primary key*
- *username* dan *password* memiliki tipe data *varchar*
- *password* akan dienkripsi menggunakan fungsi *md5*

MD5 adalah sebuah algoritma fungsi hash dari kriptografi yang digunakan dengan hash value 128-bit. MD5 telah digunakan untuk berbagai macam keperluan dalam membangun keamanan sebuah aplikasi, salah satunya digunakan untuk validasi login pada sistem. MD5 digunakan untuk mengubah nilai asli menjadi serangkaian nilai yang terenkripsi atau acak. Sintak MD5 di dalam PHP adalah:

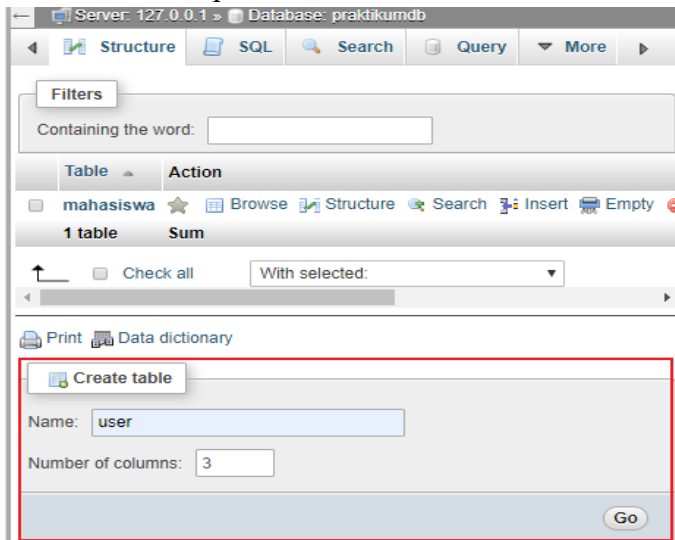
```
<?php
    echo md5('string');
?>
```

Ikuti langkah-langkah berikut untuk melakukan persiapan dalam pembuatan tabel *user*:

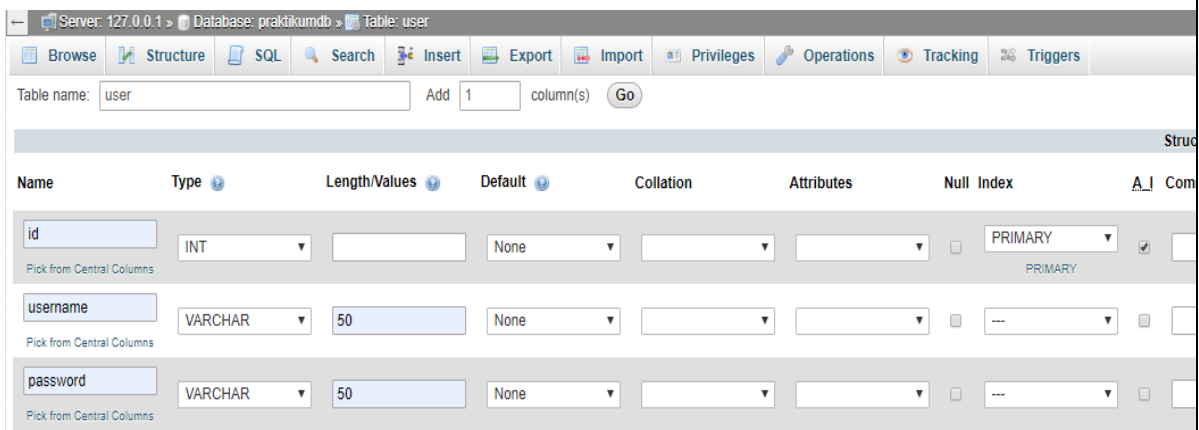
Langkah	Keterangan
1	Buka XAMPP kemudian jalankan web server dan database dengan mengklik tombol start pada modul Apache dan MySQL
2	Buka browser kemudian ketikkan localhost/phpmyadmin
3	Berikut adalah tampilan halaman phpmyadmin



4 Buka database “prakwebdb” kemudian buat tabel baru bernama “user” dengan kolom “3”



6 Berikut ini adalah kolom-kolom dalam tabel user



7 Masukkan data sebagai berikut ke dalam tabel user

Server: 127.0.0.1 > Database: praktikumdb > Table: user

Browse Structure SQL Search Insert Export Import Privileges

Column	Type	Function	Null	Value
id	int(11)			1
username	varchar(50)			admin
password	varchar(50)	MD5		123

Go

8 Tunjukkan hasil *screenshot* isi tabel user (soal no 1)

Showing rows 0 - 0 (1 total, Query took 0.0012 seconds.)

SELECT * FROM `user`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	id	username	password
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	admin	202cb962ac59075b964b07152d234b70

9 Tuliskan *query* untuk membuat tabel user menggunakan fungsi `mysqli_query()` (soal no 2)

Kode Program:

```

1  <?php
2
3  $servername = "localhost";
4  $username = "root";
5  $password = "";
6  $dbname = "prakwebdb";
7
8  // Buat koneksi ke database
9  $conn = mysqli_connect($servername, $username, $password, $dbname);
10
11 // Periksa koneksi
12 if (!$conn) {
13     die("Koneksi gagal: " . mysqli_connect_error());
14 }
15
16 // Query membuat tabel user
17 $query = "CREATE TABLE user (
18     id INT AUTO_INCREMENT PRIMARY KEY,
19     username VARCHAR(50) NOT NULL,
20     password VARCHAR(50) NOT NULL
21 )";
22
23 // Mengeksekusi query
24 if (mysqli_query($conn, $query)) {
25     echo "Tabel user berhasil dibuat";
26 } else {
27     echo "Error saat membuat tabel: " . mysqli_error($conn);
28 }
29
30 // Tutup koneksi
31 mysqli_close($conn);
32 ?>

```

Hasil:

Fatal error: Uncaught mysqli_sql_exception: Table 'user' already exists in F:\File xampp\htdocs\dasarWeb\Jobsheet 9\koneksi.php:24 Stack trace: #0 F:\File xampp\htdocs\dasarWeb\Jobsheet 9\koneksi.php(24): mysqli_query(Object(mysqli), 'CREATE TABLE us...') #1 {main} thrown in F:\File xampp\htdocs\dasarWeb\Jobsheet 9\koneksi.php on line 24

Muncul error karena tabel user sudah dibuat sebelumnya.

10 Tuliskan *query* untuk memasukkan dan ke tabel user menggunakan fungsi `mysqli_query()` (soal no 3)

Kode Program:

```
1 <?php
2
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "prakwebdb";
7
8 // Buat koneksi ke database
9 $conn = mysqli_connect($servername, $username, $password, $dbname);
10
11 // Periksa koneksi
12 if (!$conn) {
13     die("Koneksi gagal: " . mysqli_connect_error());
14 }
15
16 // Data yang akan dimasukkan
17 $nama_pengguna = "contoh_user";
18 $kata_sandi = "contoh_katasandi";
19
20 // Query memasukkan data ke tabel user
21 $query = "INSERT INTO user (username, password) VALUES ('$nama_pengguna', '$kata_sandi')";
22
23 // Mengeksekusi query
24 if (mysqli_query($conn, $query)) {
25     echo "Data berhasil dimasukkan ke tabel user";
26 } else {
27     echo "Error saat memasukkan data: " . mysqli_error($conn);
28 }
29
30 // Tutup koneksi
31 mysqli_close($conn);
32 ?>
```

Hasil:

localhost/dasarWeb/Jobsheet%209/koneksi.php

Data berhasil dimasukkan ke tabel user

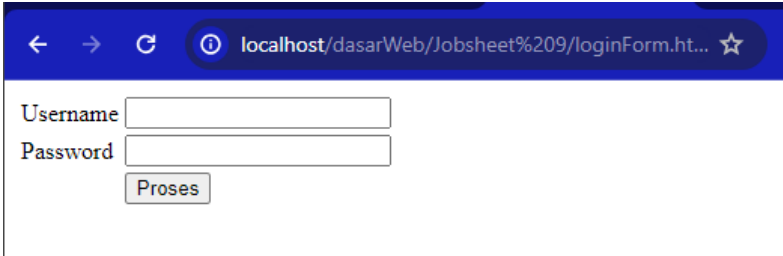
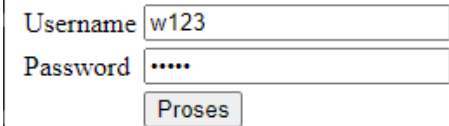
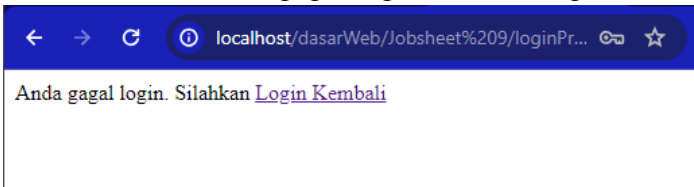
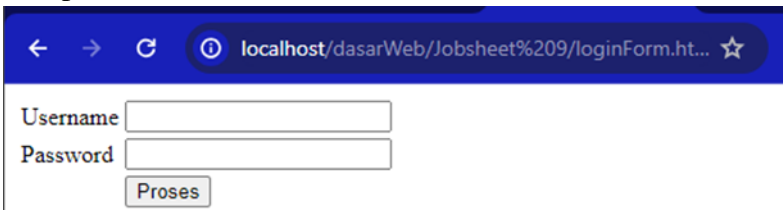
	id	username	password
<input type="checkbox"/>	1	admin	202cb962ac59075b964b07152d234b70
<input type="checkbox"/>	2	contoh_user	contoh_katasandi

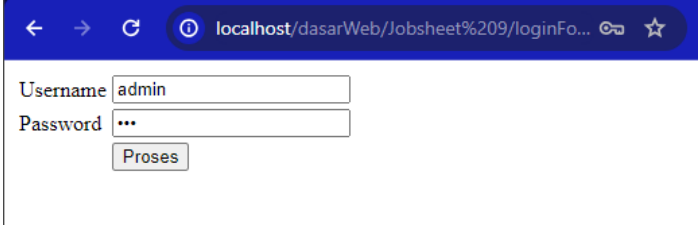
Data tersebut berhasil ditambahkan ke dalam tabel user.

Pada praktikum ini akan dijelaskan langkah-langkah untuk membuat form login single user. Cara kerja login single user adalah dengan mencocokkan *username* dan *password* yang diinputkan pada form login dengan data *username* dan *password* yang telah disimpan pada tabel user.

Ikuti langkah-langkah berikut untuk membuat login *single user*:

Langkah	Keterangan
1	Pastikan file koneksi.php dari jobsheet sebelumnya, tersimpan di dalam direktori praktik_php.
2	<p>Buat file dengan nama loginForm.html di dalam direktori praktik_php, kemudian ketikkan kode program berikut ini:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="loginProses.php" method="post"> 6 <table> 7 <tr> 8 <td>Username</td> 9 <td><input type="text" name="username" size="20"></td> 10 </tr> 11 <tr> 12 <td>Password</td> 13 <td><input type="password" name="password" size="20"></td> 14 </tr> 15 <tr> 16 <td>&nbsp;</td> 17 <td><input type="submit" name="Login" value="Proses"></td> 18 </tr> 19 </table> 20 </form> 21 </body> 22 </html> </pre>
3	<p>Buat file dengan nama loginProses.html di dalam direktori praktik_php, kemudian ketikkan kode program berikut ini:</p> <pre> 1 <?php 2 include "koneksi.php"; 3 4 \$username=\$_POST['username']; 5 \$password=md5(\$_POST['password']); 6 7 \$query="SELECT * FROM user WHERE username='\$username' and password='\$password'"; 8 \$result=mysqli_query(\$connect, \$query); 9 \$cek=mysqli_num_rows(\$result); 10 11 if(\$cek){ 12 echo "Anda berhasil login. silahkan menuju "; ?> 13 Halaman HOME 14 <?php 15 }else{ 16 echo "Anda gagal login. silahkan "; ?> 17 Login kembali 18 <?php 19 echo mysqli_error(\$connect); 20 } 21 ?> </pre>
4	Buat file dengan nama homeAdmin.html di dalam direktori praktik_php, kemudian ketikkan kode program berikut ini:

	<pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <h2> Ini adalah halaman admin </h2> 6 </body> 7 </html> </pre>
5	<p>Simpan file dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/loginForm.php</p>
6	<p>Lakukan login dengan mengetikkan username dan password yang datanya belum tersimpan di database. Jelaskan hasil pengamatanmu (soal no 4)</p> <p>Tampilan Awal:</p>  <p>Login dengan username dan password yg blum ada:</p>  <p>Akan muncul tulisan gagal login dan bisa login kembali atau login ulang.</p>  <p>Hasil yang tampak adalah tidak bisa login dikarenakan user dan password tidak ada di dalam tabel user (di database).</p>
7	<p>Lakukan login dengan mengetikkan username dan password yang datanya sudah tersimpan di database. Jelaskan hasil pengamatanmu (soal no 5)</p> <p>Tampilan Awal:</p>  <p>Login dengan username dan password yg sudah ada:</p>



Akan muncul tulisan berhasil login dan bisa menuju ke halaman home.




Proses login tersebut berhasil karena username dan password sudah tersimpan di dalam database sebelumnya.

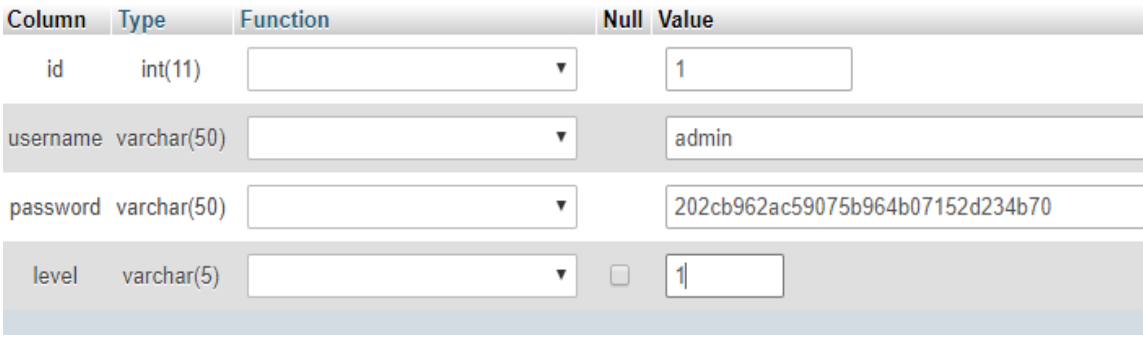
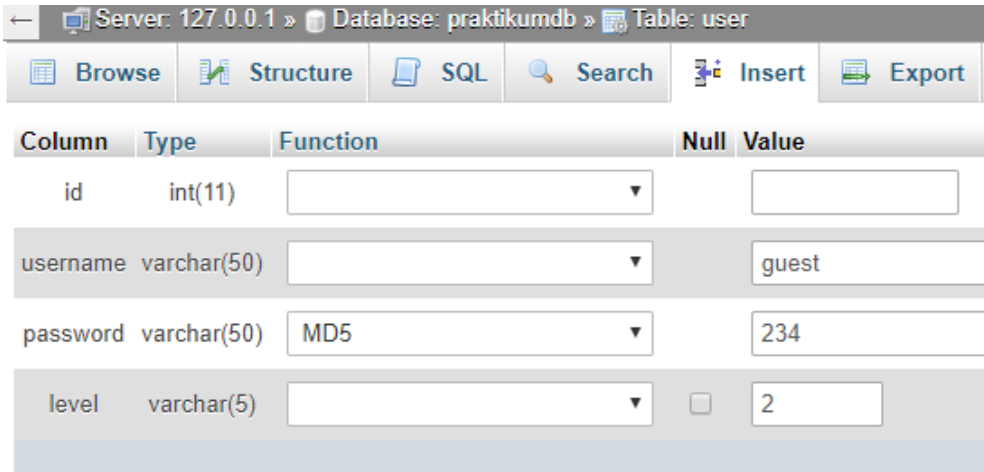
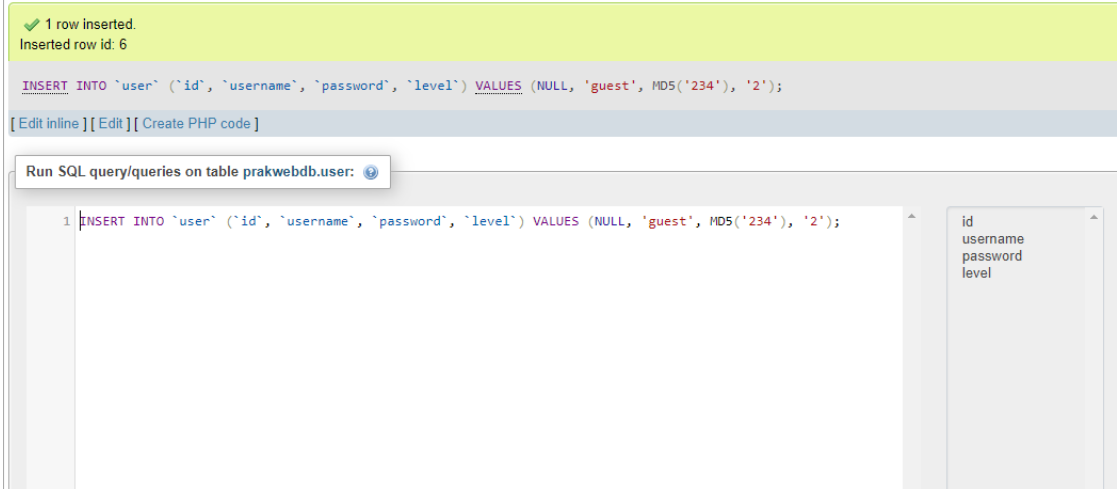
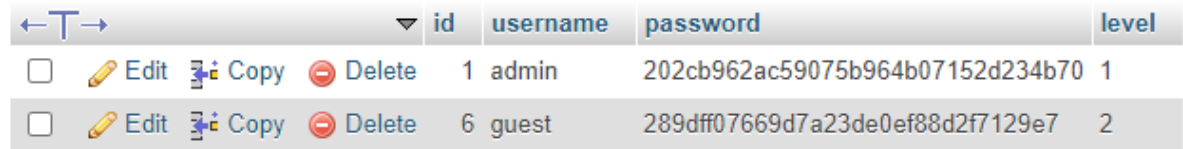
Praktikum Bagian 3. Menambahkan Kolom pada Tabel

Setelah berhasil membuat login untuk *single user*, selanjutnya akan dibahas langkah-langkah untuk membuat login multiuser. Pertama yang perlu dipersiapkan adalah memberikan level akses di dalam *database*. Sebelumnya kita hanya memiliki kolom *id*, *username*, dan *password* di dalam tabel *user*. Untuk membuat login *multiuser* perlu ditambahkan kolom level akses. Untuk menambahkan kolom pada tabel digunakan perintah **ALTER TABLE**. Berikut ini adalah sintak dasar **ALTER TABLE**:

```
ALTER TABLE namaTabel
ADD namaKolom tipeData
```

Ikuti langkah-langkah berikut untuk menambahkan kolom pada *database*

Langkah	Keterangan
1	<p>Pada tabel <i>user</i>, ketikkan perintah SQL seperti berikut ini:</p> 

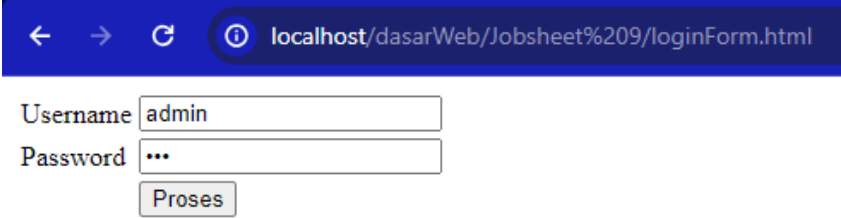


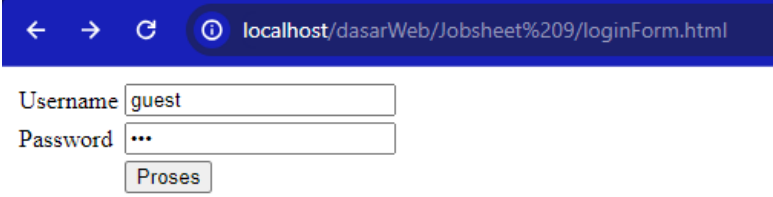
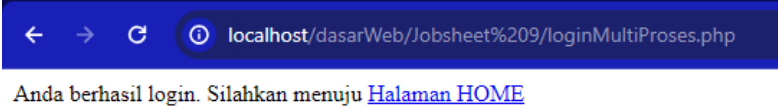
2	<p>Edit data admin, ubah level menjadi “1”</p> 
3	<p>Tambahkan data guest pada tabel user, dengan rincian data sebagai berikut:</p> 
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 6)</p>  <p>Data guest berhasil ditambahkan ke dalam tabel user.</p>
5	<p>Tunjukkan hasil <i>screenshot</i> isi tabel user (soal no 7)</p>  <p>Untuk id nya tidak bernilai 2 karena pada praktikum sebelumnya terdapat percobaan memasukkan data ke dalam tabel user melalui kode program php dengan fungsi mysqli_query sehingga id yang didapat untuk guest bernilai setelah berapa kali percobaan sebelumnya.</p>

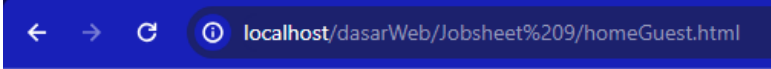

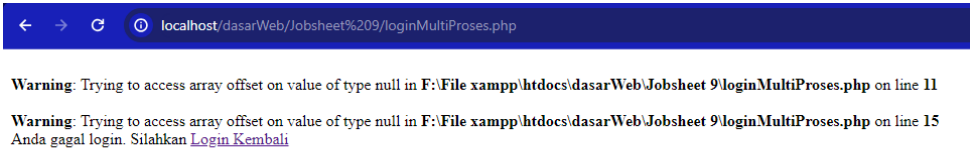

Praktikum Bagian 4. Login Multiuser

Pada praktikum ini, akan dijelaskan langkah-langkah untuk membuat login *multiuser*. Berbeda dengan login sebelumnya, selain melakukan pencocokan *username* dan *password* yang diinputkan melalui form dengan data *username* dan *password* yang disimpan di *database*, login *multiuser* akan mengenali *user* yang login dan menampilkan halaman yang sesuai dengan hak aksesnya. Pada praktikum ini terdapat 2 level login, yaitu level 1 untuk admin dan level 2 untuk guest.

Ikuti langkah-langkah berikut untuk memahami cara membuat login *multiuser*:

Langkah	Keterangan
1	Praktikum bagian 4 menggunakan file koneksi.php, sehingga pastikan file tersebut berada di dalam folder praktik_php.
2	<p>Buka file loginForm.html kemudian modifikasi kode program dengan mengubah nilai atribut action di baris ke-5 menjadi loginMultiProses.php, sehingga kode program menjadi seperti berikut ini:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="loginMultiProses.php" method="post"> 6 <table> 7 <tr> 8 <td>Username</td> 9 <td><input type="text" name="username" size="20"></td> 10 </tr> 11 <tr> 12 <td>Password</td> 13 <td><input type="password" name="password" size="20"></td> 14 </tr> 15 <tr> 16 <td>&nbsp;</td> 17 <td><input type="submit" name="Login" value="Proses"></td> 18 </tr> 19 </table> 20 </form> 21 </body> 22 </html> </pre>
3	<p>Buat file dengan nama loginMultiProses.php di dalam direktori praktik_php, kemudian ketikkan kode program berikut ini:</p> <pre> 1 <?php 2 include "koneksi.php"; 3 4 \$username=\$_POST['username']; 5 \$password=md5(\$_POST['password']); 6 7 \$query="SELECT * FROM user WHERE username='\$username' and password='\$password'"; 8 \$result=mysqli_query(\$connect, \$query); 9 \$row = mysqli_fetch_assoc(\$result); 10 11 if(\$row['level'] == 1){ 12 echo "Anda berhasil login. silahkan menuju "; ?> 13 Halaman HOME 14 <?php 15 }else if(\$row['level'] == 2){ 16 echo "Anda berhasil login. silahkan menuju "; ?> 17 Halaman HOME 18 <?php 19 }else{ 20 echo "Anda gagal login. silahkan "; ?> 21 Login kembali 22 <?php 23 echo mysqli_error(\$connect); 24 } 25 ?> </pre>

4	<p>Buat file dengan nama <code>homeGuest.html</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode program berikut ini:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <h2> Ini adalah halaman guest </h2> 6 </body> 7 </html> </pre>
5	<p>Lakukan login dengan mengetikkan data admin. Jelaskan hasil pengamatanmu (soal no 8)</p>    <p>Ketika login menggunakan data admin, maka proses login telah berhasil karena data tersebut sudah tersimpan di dalam tabel user (pada database). Selanjutnya, bisa menuju ke halaman home untuk admin.</p>
6	<p>Lakukan login dengan mengetikkan data guest. Jelaskan hasil pengamatanmu (soal no 9)</p>  

	 <p>Ini adalah halaman guest</p> <p>Proses login menggunakan data guest berhasil dilakukan karena data guest sudah tersimpan di dalam tabel user (pada database) sebelumnya. Lalu, bisa menuju ke halaman home untuk guest.</p>
7	<p>Lakukan login dengan mengetikkan username dan password yang tidak tersimpan di database. Jelaskan hasil pengamatanmu (soal no 10)</p>    <p>Proses login gagal karena data yang diinputkan tidak tersimpan di tabel user (pada database). Selain itu, muncul juga error karena kolom level yang tidak terdeteksi. Bisa login kembali atau login ulang.</p>

Praktikum Bagian 5. Membuat Cookies

PHP memiliki fungsi yang dapat digunakan untuk membuat dan mengambil nilai *cookies*. *Cookies* harus dideklarasikan sebelum halaman ditampilkan, yang artinya dituliskan sebelum tag `<html>`. *Cookies* dibuat menggunakan fungsi `setcookie()`. Berikut adalah sintaks dasar fungsi `setcookie()`:

```
<?php
    setcookie(name, value, expire);
?>
```

name, adalah nama *cookies* atau variabel yang menjadi pengenalan *cookies*

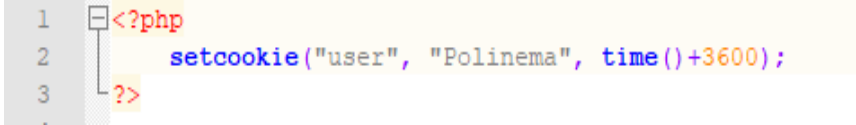
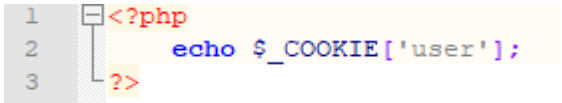
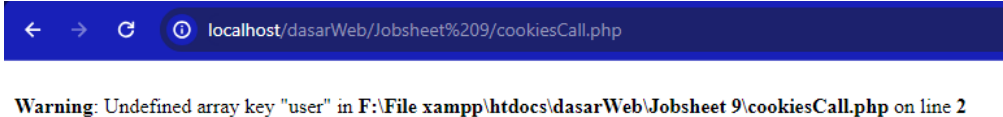

value, berisi nilai yang tersimpan dalam *cookies*

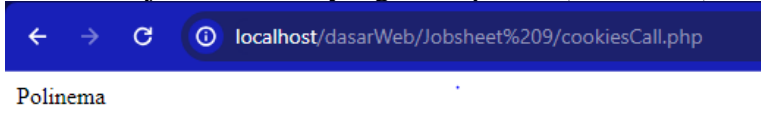

expire, adalah jangka waktu *cookies* akan disimpan di dalam komputer

Setelah mengetahui cara membuat *cookies*, selanjutnya nilai *cookies* yang telah di buat dapat diambil dengan menggunakan variabel `$_COOKIE`. Berikut ini adalah sintaks untuk mengambil nilai *cookies*:

```
<?php
    $_COOKIE['user'];
?>
```

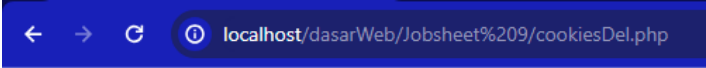
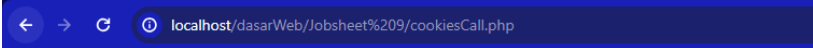
Ikuti langkah berikut untuk memahami cara kerja *cookies*:

Langkah	Keterangan
1	Jalankan Apache pada XAMPP.
2	Buat file baru dengan nama <code>cookiesCreate.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut: 
3	Buat file baru dengan nama <code>cookiesCall.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut: 
4	Buka <i>browser</i> dan jalankan kode program pada langkah ke-3 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cookiesCall.php</code>
5	Amati dan jelaskan hasil pengamatanmu (soal no 11)  Muncul error dan tidak dapat menghasilkan cookies karena pada program tersebut tidak ada perintah untuk membuat cookies, hanya ada echo untuk menampilkan cookies. Cookies belum dibuat sebelumnya jadi tidak ada/ ditemukan.
6	Buka <i>browser</i> dan jalankan kode program langkah ke-2 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cookiesCreate.php</code> 

	Tidak muncul apapun karena pada program ini hanya digunakan untuk membuat cookies.
7	Ulangi langkah ke-4
8	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 12)</p>  <p>Polinema</p> <p>Menghasilkan nilai atau tulisan 'Polinema' karena sebuah cookie dengan nama 'user' telah diatur dengan nilai 'Polinema' yang berlaku selama satu jam atau 3600 detik. Jika sudah lebih dari satu jam, maka cookie 'user' sudah tidak tersedia kembali sehingga di halaman web tidak muncul apa-apa.</p>
9	<i>Restart</i> komputer anda
10	Setelah komputer menyala, nyalakan kembali Apache pada XAMPP
11	Buka <i>browser</i> yang sama dengan sebelumnya kemudian ulangi langkah ke-4
12	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 13)</p>  <p>Polinema</p> <p>Untuk nilai atau tulisan tersebut masih ada karena belum melebihi batas maksimal waktunya, yaitu 1 jam.</p>

Praktikum Bagian 6. Menghapus Nilai *Cookies*

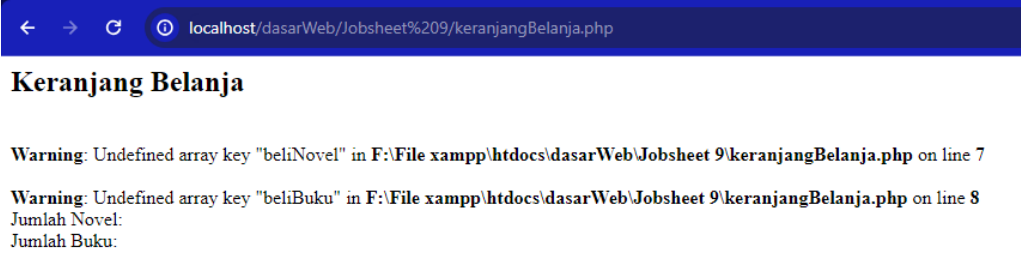
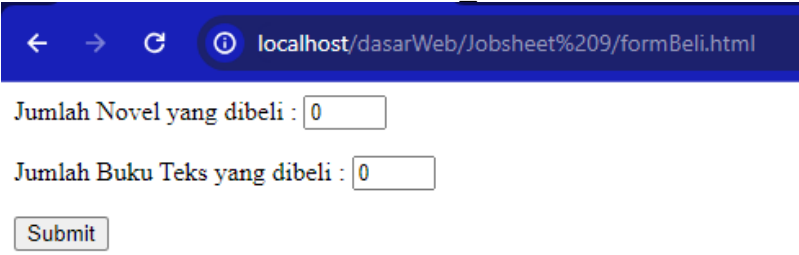
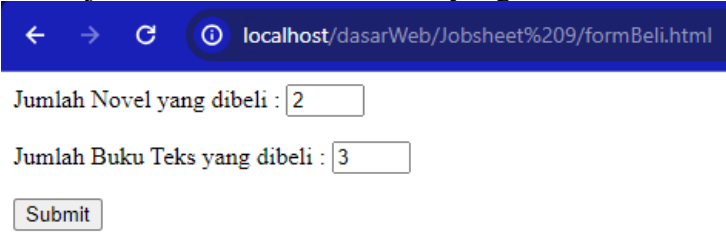
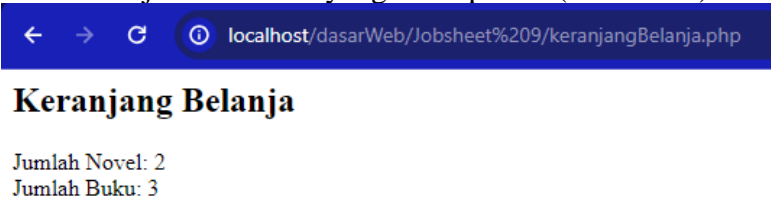
Pada praktikum ini akan dibahas mengenai cara menghapus nilai *cookies*. Jika pada praktikum bagian sebelumnya *cookies* di set dengan waktu *expire* `time()+3600`, maka untuk menghapus nilai *cookies* adalah dengan sebagai berikut:

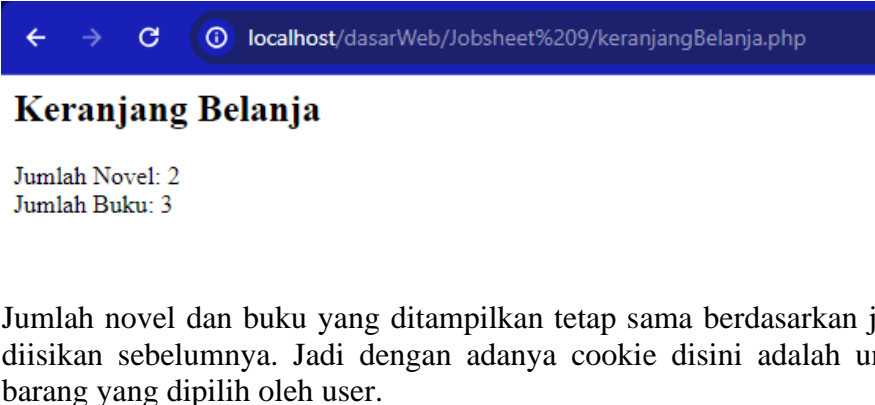
Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>cookiesDel.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre>1 <?php 2 setcookie("user", "Polinema", time()-3600); 3 ?></pre>
2	<p>Buka <i>browser</i> dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cookiesDel.php</code></p>
3	<p>Buka <i>browser</i> dan jalankan kode program dari praktikum bagian 1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cookiesCall.php</code></p>
4	<p>Amati dan jelaskan hasil yang dihasilkan dari langkah 2 dan 3, kemudian beri kesimpulan (soal no 14)</p>   <p>Warning: Undefined array key "user" in F:\xampp\htdocs\dasarWeb\Jobsheet 9\cookiesCall.php on line 2</p> <p>Ketika menjalankan langkah ke-2, tidak terlihat apa-apa di halaman web karena hanya menghapus cookies tanpa menampilkan output apapun. Ketika menjalankan langkah ke-3, muncul error yang disebabkan tidak ada cookies atau cookies tersebut sudah kadaluwarsa karena adanya <code>time()-3600</code>.</p>

Praktikum Bagian 7. Penerapan *Cookies* pada Fitur Keranjang Belanja

Salah satu contoh pemanfaatan *cookies* adalah pada fitur “keranjang belanja” pada aplikasi web toko online. Keranjang belanja berisi barang-barang yang akan dibeli oleh user. *Cookies* digunakan untuk mengingat jumlah barang yang dipilih user. Berikut adalah contoh penggunaan *cookies* pada fitur keranjang belanja:

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>formBeli.html</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="prosesBeli.php" method="POST"> 6 <p> Jumlah Novel yang dibeli : 7 <input type="text" name="beliNovel" value= "0" size="2"> </p> 8 <p> Jumlah Buku Teks yang dibeli : 9 <input type="text" name="beliBuku" value= "0" size="2"> </p> 10 <input type="submit"> 11 </form> 12 </body> 13 </html> </pre>
2	<p>Buat file baru dengan nama <code>prosesBeli.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 if(isset(\$_POST["beliNovel"]) && isset(\$_POST["beliBuku"])){ 3 setcookie("beliNovel", \$_POST["beliNovel"]); 4 setcookie("beliBuku", \$_POST["beliBuku"]); 5 header("location:keranjangBelanja.php"); 6 } 7 ?> </pre>
3	<p>Buat file baru dengan nama <code>prosesBeli.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <h2> Keranjang Belanja </h2> 6 7 <?php 8 \$beliNovel=\$_COOKIE['beliNovel']; 9 \$beliBuku = \$_COOKIE['beliBuku']; 10 11 echo "Jumlah Novel:" . \$beliNovel . "
"; 12 echo "Jumlah Buku :" . \$beliBuku ; 13 ?> 14 </body> 15 </html> </pre>
4	<p>Buka <i>browser</i> dan jalankan kode program langkah ke-3 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/keranjangBelanja.php</code></p>

5	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 15)</p>  <p>Warning: Undefined array key "beliNovel" in F:\File xampp\htdocs\dasarWeb\Jobsheet 9\keranjangBelanja.php on line 7</p> <p>Warning: Undefined array key "beliBuku" in F:\File xampp\htdocs\dasarWeb\Jobsheet 9\keranjangBelanja.php on line 8</p> <p>Jumlah Novel: Jumlah Buku:</p> <p>Hasil yang ditampilkan adalah tidak ada nilai dari jumlah novel dan buku beserta muncul error karena mencoba mengakses elemen array dari \$_COOKIE yang sebelumnya masih bernilai kosong atau null sehingga perlu mengisi jumlahnya terlebih dahulu pada bagian form.</p>
6	<p>Jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik_php/formBeli.html</p>  <p>Jumlah Novel yang dibeli : <input type="text" value="0"/></p> <p>Jumlah Buku Teks yang dibeli : <input type="text" value="0"/></p> <p><input type="button" value="Submit"/></p>
7	<p>Isikan jumlah novel dan buku teks yang akan dibeli kemudian klik tombol "submit"</p>  <p>Jumlah Novel yang dibeli : <input type="text" value="2"/></p> <p>Jumlah Buku Teks yang dibeli : <input type="text" value="3"/></p> <p><input type="button" value="Submit"/></p>
8	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 16)</p>  <p>Keranjang Belanja</p> <p>Jumlah Novel: 2 Jumlah Buku: 3</p> <p>Setelah menekan tombol submit, akan menampilkan halaman dari keranjang belanja yang mana menampung jumlah atau angka yang kita inputkan untuk jumlah novel dan buku pada file 'formBeli.html' yang sebelumnya juga diatur cookies tersebut pada file 'prosesBeli.php'.</p>
9	<p>Tutup <i>browser</i> kemudian buka kembali <i>browser</i> kemudian jalankan kembali kode program langkah ke-3 dengan mengetikkan localhost/dasarWeb/praktik_php/keranjangBelanja.php</p>
10	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 17)</p>

	
--	--

Praktikum Bagian 8. Membuat *Session*

Untuk membuat *session* dalam PHP digunakan fungsi `session_start()`. Berikut adalah sintaks dasar untuk memulai *session*:

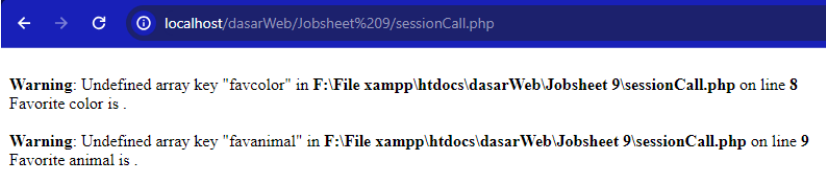
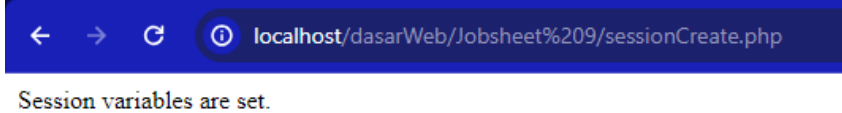
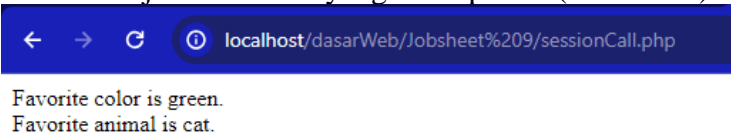
```
<?php
    session_start();
?>
```

Setelah *session* dimulai, maka variabel *session* sudah dapat mulai digunakan. Variabel yang dimaksud adalah PHP global variable `$_SESSION`. Berikut adalah sintaks menggunakan variable global `$_SESSION`:

```
<?php
    $_SESSION['namaVariabel'];
?>
```

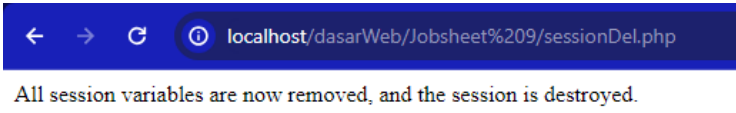
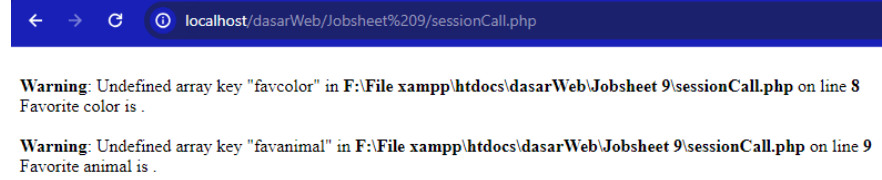
Ikuti langkah berikut untuk memahami cara kerja *session*:

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>sessionCreate.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 ?> 4 5 <!DOCTYPE html> 6 <html> 7 <body> 8 <?php 9 \$_SESSION["favcolor"] = "green"; 10 \$_SESSION["favanimal"] = "cat"; 11 echo "Session variables are set."; 12 ?> 13 </body> 14 </html></pre>

2	<p>Buat file baru dengan nama <code>sessionCall.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 ?> 4 <!DOCTYPE html> 5 <html> 6 <body> 7 <?php 8 echo "Favorite color is " . \$_SESSION["favcolor"] . "
"; 9 echo "Favorite animal is " . \$_SESSION["favanimal"] . "."; 10 ?> 11 </body> 12 </html> </pre>
3	<p>Buka <i>browser</i> dan jalankan kode program pada langkah ke-2 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionCall.php</code></p>  <p>Muncul error karena belum ada session yang tersedia atau hanya menampilkannya saja.</p>
4	<p>Buka <i>browser</i> dan jalankan kode program langkah ke-1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionCreate.php</code></p>  <p>Setelah eksekusi file <code>sessionCreate.php</code>, maka session yang abru terbentuk dengan nilai green dan cat.</p>
5	<p>Ulangi langkah ke-3</p>
6	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 18)</p>  <p>Ketika dieksekusi kembali file <code>sessionCall.php</code>, sudah muncul nilai dari session yang dibuat sebelumnya.</p>

Praktikum Bagian 9. Menghapus Nilai Session

Pada praktikum ini akan dibahas mengenai cara menghapus *session*. PHP menyediakan fungsi `session_destroy()` yang dapat digunakan untuk menghapus *session*. Berikut adalah langkah-langkah untuk menghapus *session*:

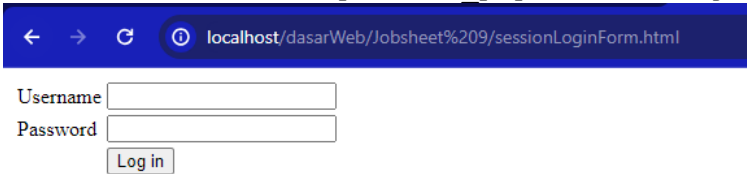
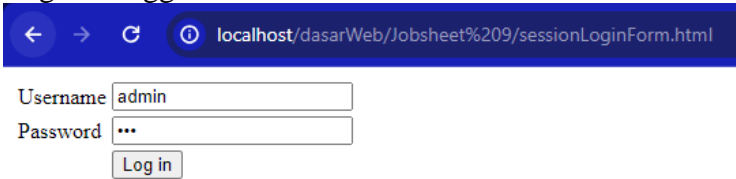

Langkah	Keterangan
1	<p>Buat file baru dengan nama <code>sessionDel.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 ?> 4 5 <!DOCTYPE html> 6 <html> 7 <body> 8 <?php 9 session_unset(); 10 session_destroy(); 11 12 echo "All session variables are now removed, and the session is destroyed." 13 ?> 14 </body> 15 </html> </pre>
2	<p>Buka <i>browser</i> dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionDel.php</code></p>  <p>All session variables are now removed, and the session is destroyed.</p>
3	<p>Buka <i>browser</i> dan jalankan kode program dari praktikum bagian 1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionCall.php</code></p>  <p>Warning: Undefined array key "favcolor" in F:\File xampp\htdocs\dasarWeb\Jobsheet 9\sessionCall.php on line 8 Favorite color is .</p> <p>Warning: Undefined array key "favanimal" in F:\File xampp\htdocs\dasarWeb\Jobsheet 9\sessionCall.php on line 9 Favorite animal is .</p>
4	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 19)</p> <p>Ketika menjalankan file <code>sessionDel.php</code>, maka semua session akan dihapus (nilai green dan cat yang sebelumnya ada) sehingga ketika file <code>sessionCall.php</code> dijalankan kembali, akan muncul error kembali karena sudah tidak ada session lagi.</p>

Praktikum Bagian 10. Penerapan *Session* pada Fitur Login

Ikuti langkah-langkah berikut untuk mengetahui penerapan *session* pada fitur login:

Langkah	Keterangan
1	Jalankan MySQL pada XAMPP

2	Pastikan file koneksi.php yang telah dibuat pada praktikum jobsheet sebelumnya ada di dalam direktori praktik_php
3	<p>Buat file baru dengan nama sessionLoginForm.html di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="sessionLoginProses.php" method="POST"> 6 <table> 7 <tr> 8 <td>Username</td> 9 <td><input type="text" name="username" size="20"></td> 10 </tr> 11 <tr> 12 <td>Password</td> 13 <td><input type="password" name="password" size="20"></td> 14 </tr> 15 <tr> 16 <td>&nbsp;</td> 17 <td><input type="submit" name="Login" value="Log in"></td> 18 </tr> 19 </table> 20 </form> 21 </body> 22 </html> </pre>
4	<p>Buat file baru dengan nama sessionLoginProses.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 include "koneksi.php"; 3 4 \$username = \$_POST['username']; 5 \$password = md5(\$_POST['password']); 6 7 \$sql = "SELECT * FROM user WHERE username='\$username' and password='\$password'"; 8 \$result = mysqli_query(\$connect, \$sql); 9 \$cek = mysqli_num_rows(\$result); 10 11 if(\$cek > 0){ 12 session_start(); 13 \$_SESSION['username'] = \$username; 14 \$_SESSION['status'] = 'login'; 15 } 16 Anda Berhasil Login, silahkan menuju 17 Halaman Home <?php 18 }else{ 19 ?> 20 Gagal login, silahkan login lagi 21 Halaman Login <?php 22 echo mysqli_error(\$connect); 23 } 24 ?> </pre>
5	<p>Buat file baru dengan nama homeSession.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p>

	<pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <?php 6 session_start(); 7 8 if(\$_SESSION['status'] == 'login'){ 9 echo "selamat datang " . \$_SESSION['username']; 10 ?> 11
 Log Out 12 <?php 13 }else{ 14 echo "anda belum login. silahkan "?> 15 Log In 16 <?php 17 } 18 ?> 19 </body> 20 </html> </pre>
6	<p>Buat file baru dengan nama sessionLogout.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 session_destroy(); 4 5 echo "Anda berhasil logout"; 6 ?> </pre>
7	<p>Buka <i>browser</i> dan jalankan kode program dengan mengetikkan localhost/dasarWeb/praktik_php/sessionLoginForm.html</p> 
8	<p>Login menggunakan username admin.</p> 
9	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 20)</p> 

	  <p>Ketika login menggunakan data admin, proses tersebut akan berhasil karena data admin sudah tersimpan di dalam database sebelumnya. Setelah berhasil login, bisa menuju ke halaman home. Di halaman home, admin bisa melakukan log out dengan klik link tersebut dan nantinya akan otomatis log out atau berhasil log out.</p>
10	<p>Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (soal no 21)</p> <p>Proses login dimulai dengan mengisi username dan password pada file 'sessionLoginForm.html'. Setelah klik Log in, maka pada file sessionLoginProses.php akan dicek apakah username dan password yang dimasukkan sesuai dengan isi dari tabel user yang terdapat di database. Jika tidak sesuai, maka akan diarahkan untuk login kembali atau login ulang. Jika sesuai, maka akan muncul tulisan berhasil login dan diarahkan untuk menuju ke halaman home melalui sebuah hyperlink. Pada bagian ini, juga sudah dibuatkan session yang baru untuk username dan status login. Ketika menekan halaman home, otomatis akan mengakses file homeSession.php yang di dalamnya bertuliskan 'selamat datang user'. User tersebut didapatkan dari session yang telah dibuat sebelumnya di file 'sessionLoginProses.php'. Jika anda belum login tetapi mengakses halaman homeSession.php, maka akan diarahkan untuk login terlebih dahulu. Selanjutnya, jika ingin log out bisa menekan hyperlink yang bertuliskan Log Out, maka otomatis akan terlog out dari form tersebut.</p>

Referensi :

- 1) Nixon, Robin. (2018). Learning PHP, MySQL, JavaScript, CSS & HTML: A Step-by-step Guide to Creating Dynamic Websites, 5th Edition. O'Reilly Media, Inc.
- 2) Forbes, Alan. (2012). The Joy of PHP: A Beginners's Guide to Programming Interactive Web Applications with PHP and MySQL, 5th Edition. Plum Island Publishing