



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*:

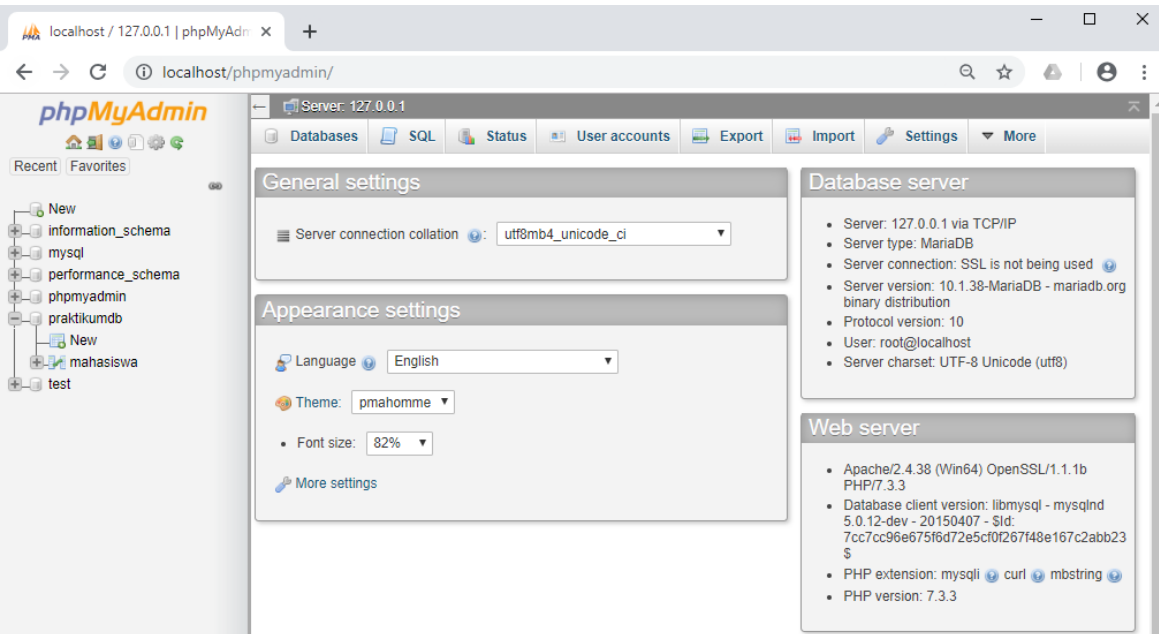
praktikumbd user	
id	int(11)
username	varchar(50)
password	varchar(50)

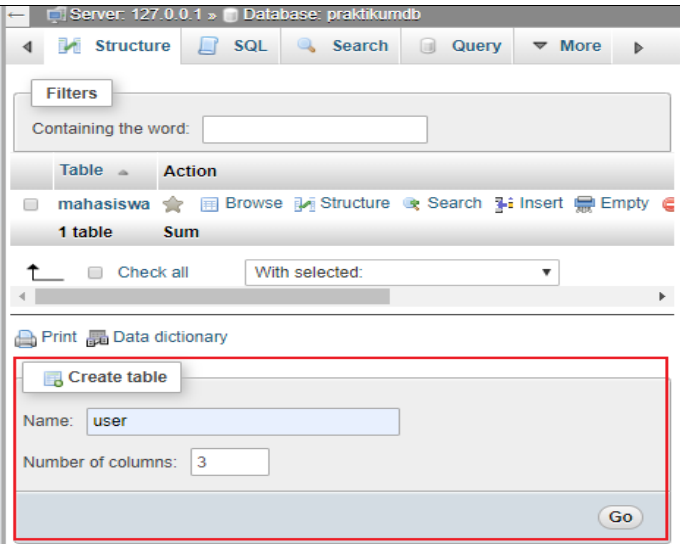
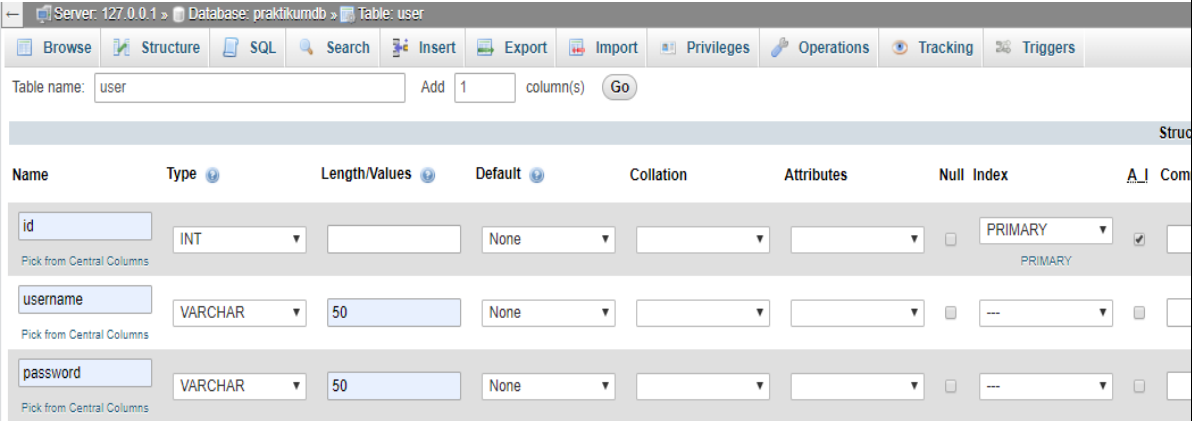
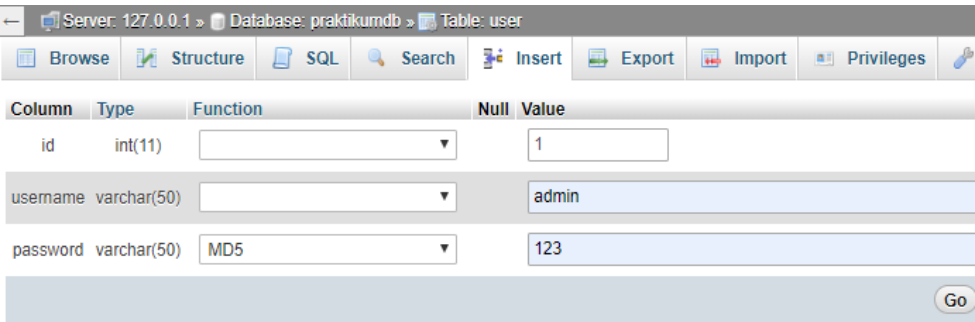
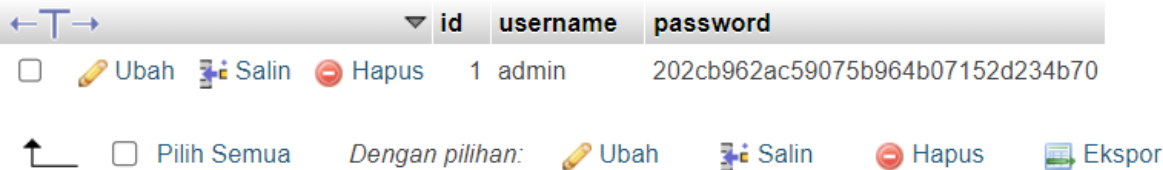
- tabel bernama *user*
- kolom tabel terdiri dari *id*, *username*, dan *password*
- *id* memiliki tipe data integer dan di atur sebagai *primary key*
- *username* dan *password* memiliki tipe data *varchar*
- *password* akan dienkripsi menggunakan fungsi *md5*

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	<p>Berikut ini adalah kolom-kolom dalam tabel user</p> 
7	<p>Masukkan data sebagai berikut ke dalam tabel user</p> 
8	<p>Tunjukkan hasil <i>screenshot</i> isi tabel user (soal no 1)</p> <p>Jawab:</p> 
9	<p>Tuliskan <i>query</i> untuk membuat tabel user menggunakan fungsi <code>mysql_query()</code> (soal no 2)</p> <p>Jawab:</p>

Tabel 'user' berhasil dibuat.

```
<?php

$host = "127.0.0.1:3308";
$username = "root";
$password = "123";
$database = "prakwebdb";

$conn = mysqli_connect($host, $username, $password, $database);

if (!$conn) {
    die("koneksi gagal: " . mysqli_connect_error());
}else{
    echo "koneksi berhasil";
}

$query = "CREATE TABLE user (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL)";

if (mysqli_query($conn, $query)) {
    echo "Tabel 'user' berhasil dibuat.";
} else {
    echo "Error: " . mysqli_error($conn);
}
```

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

Jawab:

Data berhasil dimasukkan ke dalam tabel 'user'.

```

$newUserName = "admin";
$newPassword = "123";
$query = "INSERT INTO user (username, password) VALUES ('$newUserName', '$newPassword')";

// Menjalankan query dengan mysqli_query()
if (mysqli_query($conn, $query)) {
    echo "Data berhasil dimasukkan ke dalam tabel 'user'.";
} else {
    echo "Error: " . mysqli_error($conn);
}

Tutup koneksi
mysqli_close($conn);
?>

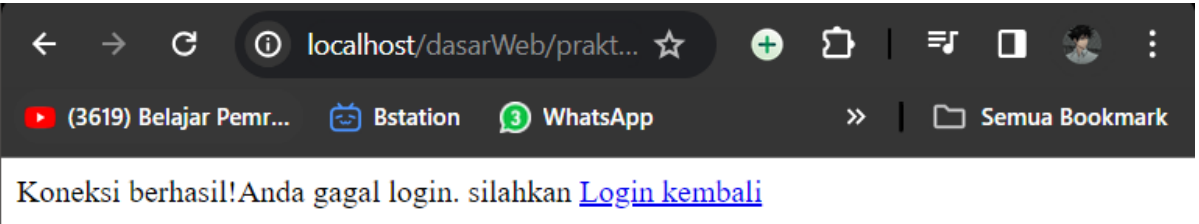
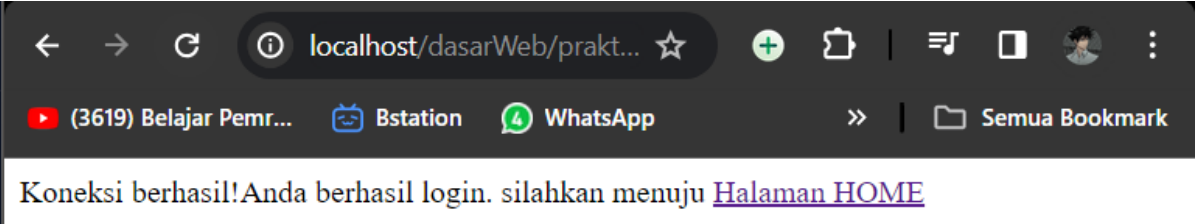
```

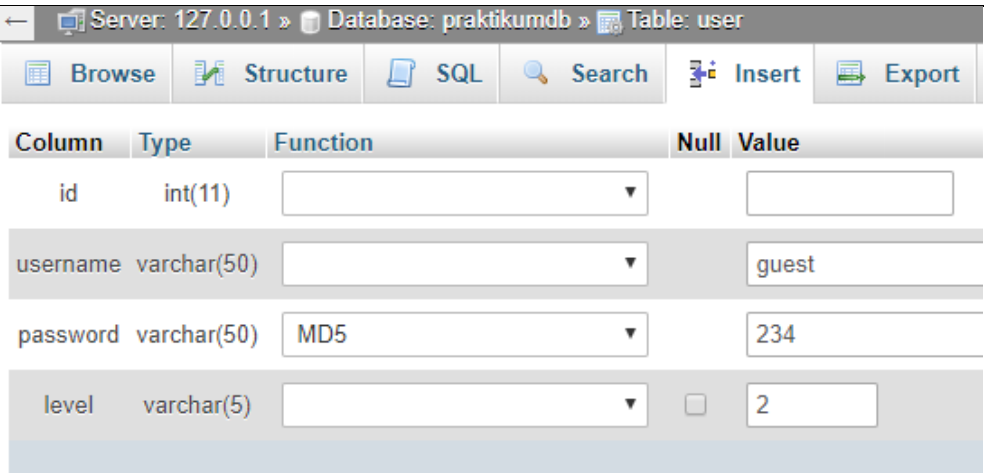
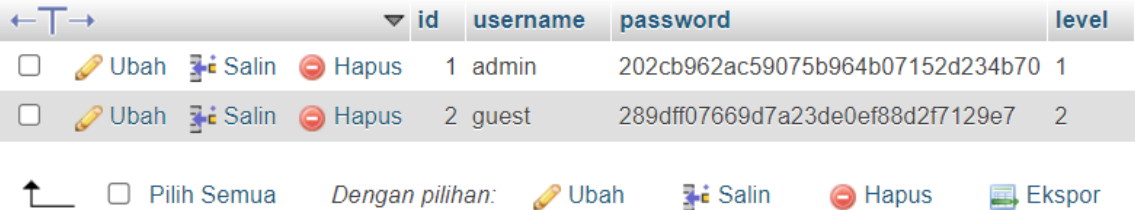
Praktikum Bagian 2. Login Single User

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 } 15 }else{ 16 echo "Anda gagal login. silahkan "; ?> 17 Login kembali 18 } 19 echo mysqli_error(\$connect); 20 } 21 ?> </pre>
4	<p>Buat file dengan nama homeAdmin.html di dalam direktori praktik_php, kemudian ketikkan kode program berikut ini:</p> <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) Jawab:</p> 
7	<p>Lakukan login dengan mengetikkan username dan password yang datanya sudah tersimpan di database. Jelaskan hasil pengamatanmu (soal no 5) Jawab:</p> 

	
4	<p>Amati hasil yang ditampilkan dan jelaskan hasil pengamatanmu (soal no 6)</p> <p>Jawab:</p> <p>Akan menambah 1 baris dan isi pada kolom username, password, level setelah kita memerintah untuk mengisi database tersebut</p>
5	<p>Tunjukkan hasil <i>screenshot</i> isi tabel user (soal no 7)</p> <p>Jawab:</p> 

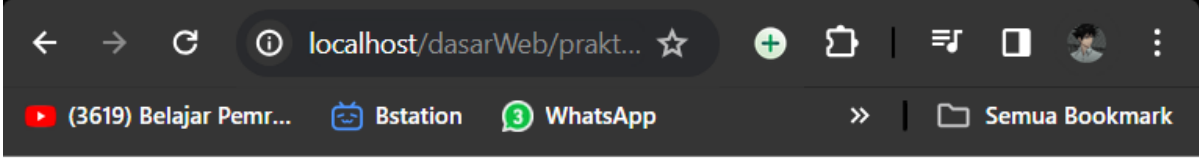
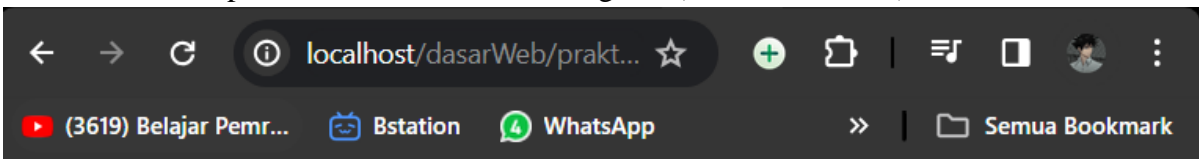
Praktikum Bagian 4. Login *Multiuser*

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	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:

	<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 homeGuest.html di dalam direktori praktik_php, 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>Jawab:</p> <p>Pengguna memasukkan username dan password milik admin yang memiliki level 1 sehingga web akan menampilkan halaman Home untuk admin (homeAdmin.html)</p>

	 <p>Koneksi berhasil! Anda berhasil login. silahkan menuju Halaman HOME</p>  <p>Ini adalah halaman admin</p>
6	<p>Lakukan login dengan mengetikkan data guest. Jelaskan hasil pengamatanmu (soal no 9)</p> <p>Jawab:</p> <p>Pengguna memasukkan username dan password milik guest yang memiliki level 2 sehingga web akan menampilkan halaman Home untuk guest (homeGuest.html)</p>  <p>Koneksi berhasil! Anda berhasil login. silahkan menuju Halaman HOME</p>  <p>Ini adalah halaman guest</p>
7	<p>Lakukan login dengan mengetikkan username dan password yang tidak tersimpan di database. Jelaskan hasil pengamatanmu (soal no 10)</p> <p>Jawab:</p> <p>Pengguna memasukkan username dan password yang belum terdaftar di database. Hasilnya koneksi dapat tersambung namun terdapat error karena data level yang sedang di cek pada percabangan yang ada pada file loginMultiProses bernilai null sehingga program tidak dapat mendeteksi, kemudian pengguna diminta untuk login kembali</p>  <p>Koneksi berhasil!</p> <p>Warning: Trying to access array offset on value of type null in C:\xampp\htdocs\dasarWeb\praktik_php\loginMultiProses.php on line 9</p> <p>Warning: Trying to access array offset on value of type null in C:\xampp\htdocs\dasarWeb\praktik_php\loginMultiProses.php on line 13</p> <p>Anda gagal login. silahkan Login kembali</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*

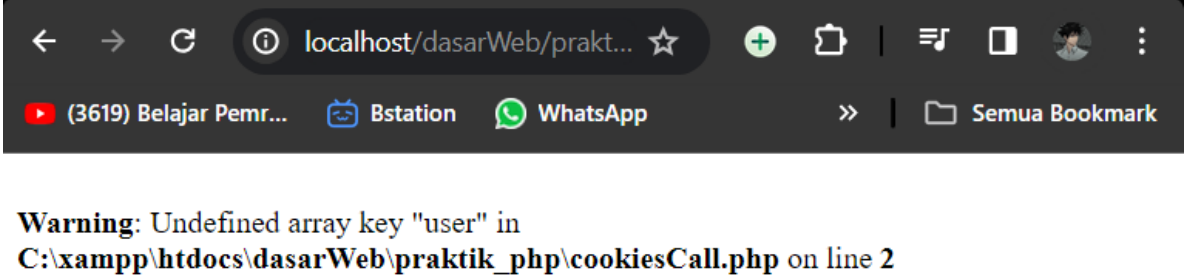
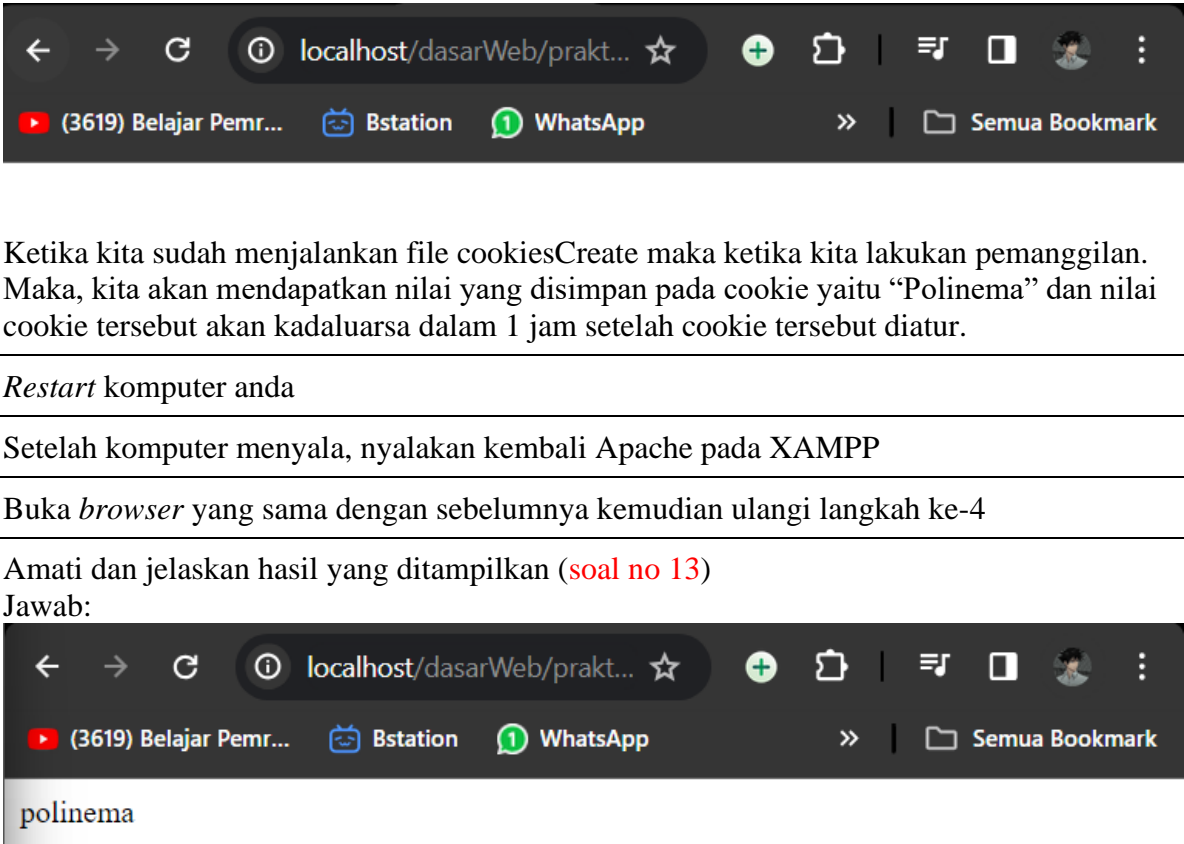
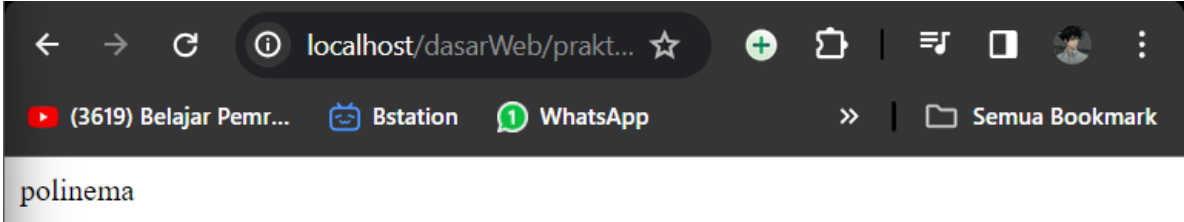
expire, adalah jangka waktu *cookies* akan disimpan di dalam komputer

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: <pre>1 <?php 2 setcookie("user", "Polinema", time()+3600); 3 ?></pre>
3	Buat file baru dengan nama <code>cookiesCall.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut: <pre>1 <?php 2 echo \$_COOKIE['user']; 3 ?></pre>
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) Jawab: Dari hasil diatas kita dapatkan karena kita belum membuat <i>cookies</i> nya terlebih dahulu. Sehingga data yang memang kosong. Kita ingin akses akan menghasilkan <code>undefined</code> . Kita

	<p>harus membuat cookiesnya terlebih dahulu baru kita lakukan pemanggilan.</p> 
6	Buka <i>browser</i> dan jalankan kode program langkah ke-2 dengan mengetikkan localhost/dasarWeb/praktik_php/cookiesCreate.php
7	Ulangi langkah ke-4
8	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 12)</p> <p>Jawab:</p>  <p>Ketika kita sudah menjalankan file cookiesCreate maka ketika kita lakukan pemanggilan. Maka, kita akan mendapatkan nilai yang disimpan pada cookie yaitu “Polinema” dan nilai cookie tersebut akan kadaluarsa dalam 1 jam setelah cookie tersebut diatur.</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>Jawab:</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:

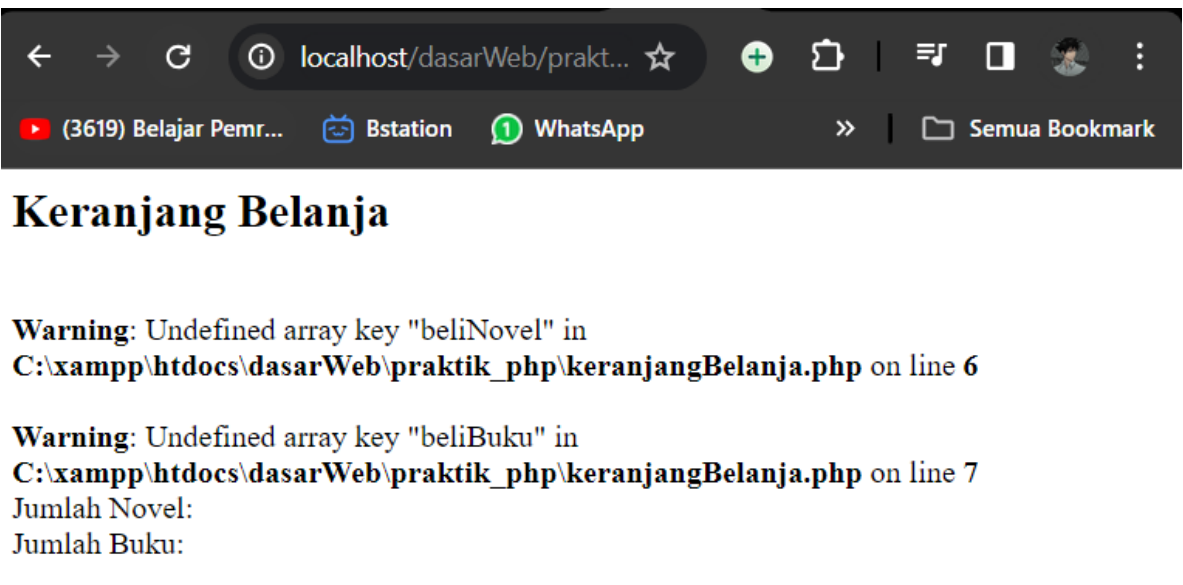
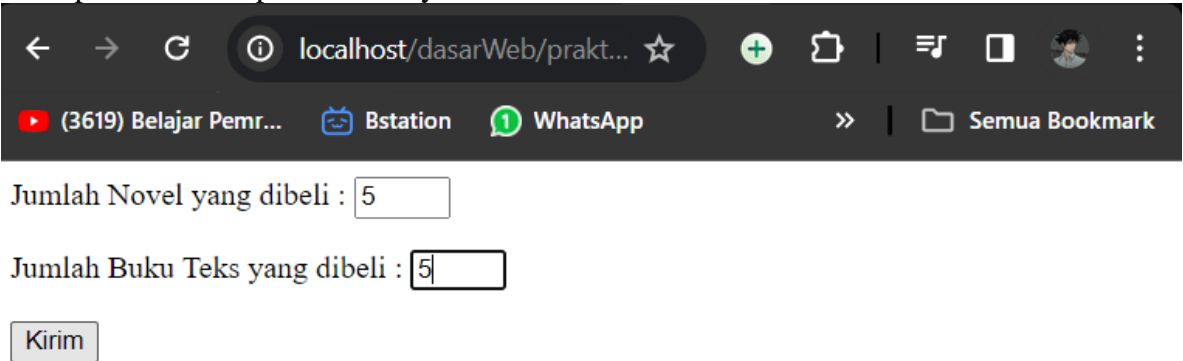
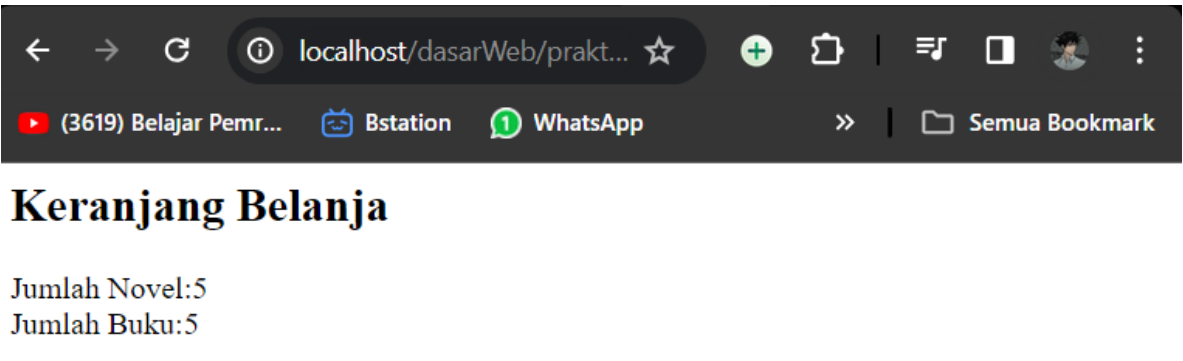
Langkah	Keterangan
1	Buat file baru dengan nama <code>cookiesDel.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut: <pre>1 <?php 2 setcookie("user", "Polinema", time()-3600); 3 ?></pre>
2	Buka <i>browser</i> dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cookiesDel.php</code>
3	Buka <i>browser</i> dan jalankan kode program dari praktikum bagian 1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/cookiesCall.php</code>
4	Amati dan jelaskan hasil yang dihasilkan dari langkah 2 dan 3, kemudian beri kesimpulan (soal no 14) Jawab: Ketika dijalankan maka cookie akan terhapus. Untuk pembuktian kita lakukan pemanggilan pada file <code>cookieCall</code> . Yang terjadi adalah akan terjadi error karena memang cookie nya tidak ada dan dipaksa dilakukan pemanggilan sehingga terjadi error. <pre>Warning: Undefined array key "user" in D:\Xampp\htdocs\pertemuan9\cookiesCall.php on line 2</pre>

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	Buat file baru dengan nama <code>formBeli.html</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut:

	<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 prosesBeli.php di dalam direktori praktik_php, 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 prosesBeli.php di dalam direktori praktik_php, 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 localhost/dasarWeb/praktik_php/keranjangBelanja.php</p>
5	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 15)</p> <p>Jawb:</p> <p>Yang terjadi adalah error karena cookie yang belum ada ingin kita akses. Seharusnya kita mengisikan form terlebih dahulu agar mendapatkan datanya. Lalu, kita panggil kembali agar tidak terjadi error.</p>

	 <p>Keranjang Belanja</p> <p>Warning: Undefined array key "beliNovel" in C:\xampp\htdocs\dasarWeb\praktik_php\keranjangBelanja.php on line 6</p> <p>Warning: Undefined array key "beliBuku" in C:\xampp\htdocs\dasarWeb\praktik_php\keranjangBelanja.php on line 7</p> <p>Jumlah Novel: Jumlah Buku:</p>
6	Jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik_php/formBeli.html
7	Isikan jumlah novel dan buku teks yang akan dibeli kemudian klik tombol “submit”
8	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 16)</p> <p>Jawab:</p> <p>Dari hasil inputan dan setelah mengklik tombol kirim maka akan diarahkan ke halaman selanjutnya dan mengeluarkan output apa yang kita masukkan.</p> <p>Ketika kita mengisi data pada form lalu menekan tombol submit maka akan langsung dihubungkan dengan file lainnya yaitu keranjangBelanja.php untuk ditampilkan hasil inputan form nya.</p>  <p>Jumlah Novel yang dibeli : <input type="text" value="5"/></p> <p>Jumlah Buku Teks yang dibeli : <input type="text" value="5"/></p> <p><input type="button" value="Kirim"/></p>  <p>Keranjang Belanja</p> <p>Jumlah Novel:5 Jumlah Buku:5</p>
9	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
10	Amati dan jelaskan hasil yang ditampilkan (soal no 17)

	<p>Jawab:</p> <p>ketika kita langsung mengakses pada file keranjangBelanja.php tanpa kita mengisi form tersebut website tersebut dapat dijalankan. Karena, sebelumnya kita telah mengisi data pada form terkait dan data tersebut disimpan ke dalam cookie sehingga masih dapat tertampilkan hasil pada file tersebut.</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();
?>
```

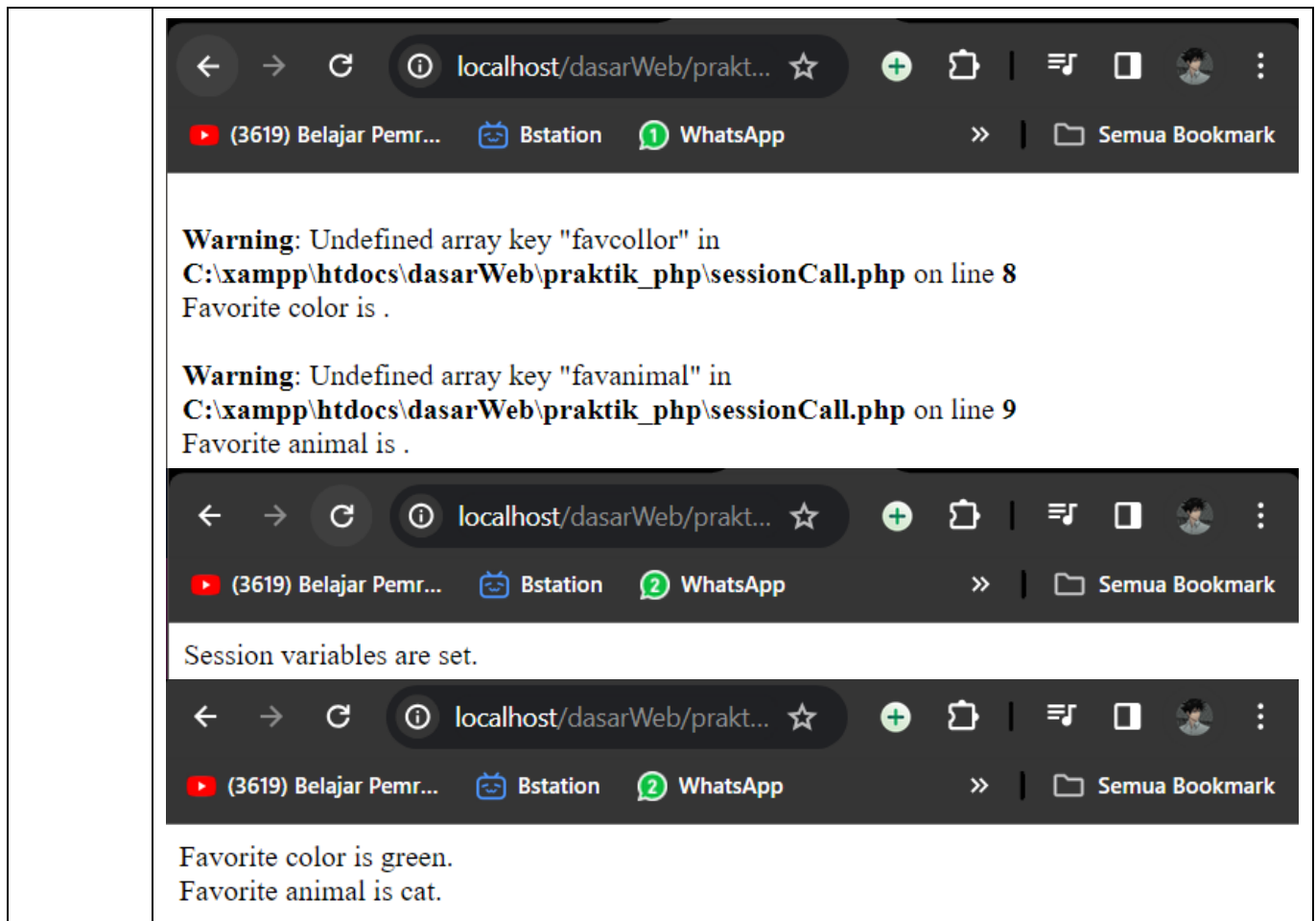
Setelah *session* dimulai, maka variabel *session* sudah dapat mulai digunakan. Variabel yang dimaksud adalah PHP global variable `$_SESSION`. Berikut adalah sintaks menggunakan variable global `$_SESSION`:

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

Ikuti langkah berikut untuk memahami cara kerja *session*:

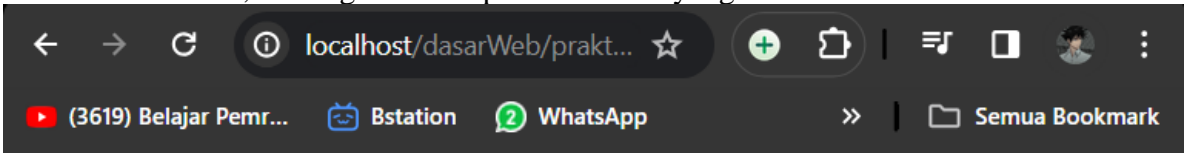
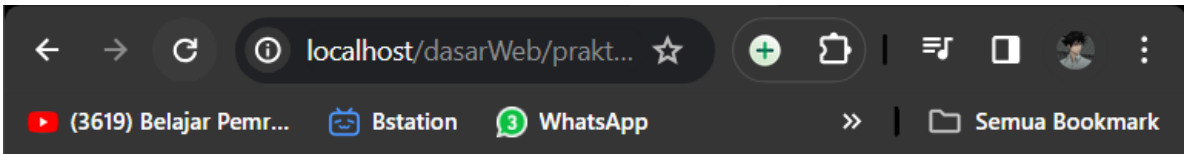
Langkah	Keterangan
1	Buat file baru dengan nama <code>sessionCreate.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut:

	<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 sessionCall.php di dalam direktori praktik_php, 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 localhost/dasarWeb/praktik_php/sessionCall.php</p>
4	<p>Buka <i>browser</i> dan jalankan kode program langkah ke-1 dengan mengetikkan localhost/dasarWeb/praktik_php/sessionCreate.php</p>
5	<p>Ulangi langkah ke-3</p>
6	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 18)</p> <p>Jawab:</p> <p>Akan terjadi error ketika kita memanggil session Call terlebih dahulu dikarenakan variable belum di set, oleh karena itu kita harus melakukan pemanggilan sessionCreate, setelah variable di set maka panggil lagi sessionCall nya akan mengeluarkan output yang sesuai</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	Buka <i>browser</i> dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionDel.php</code>
3	Buka <i>browser</i> dan jalankan kode program dari praktikum bagian 1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionCall.php</code>
4	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 19)</p> <p>Jawab:</p> <p>Dari hasil pengamatan ketika kita menjalankan file <code>sessionDel.php</code> kita akan menghapus sesi sehingga ketika kita memanggil atau menjalankan kembali pada file <code>sessionCall.php</code> hasilnya akan error. Karena, kita ingin menampilkan sesuatu yang tidak ada.</p>  <p>All session are now removed, and the session is destroyed.</p>  <p>Warning: Undefined array key "favcolor" in C:\xampp\htdocs\dasarWeb\praktik_php\sessionCall.php on line 8 Favorite color is .</p> <p>Warning: Undefined array key "favourite" in C:\xampp\htdocs\dasarWeb\praktik_php\sessionCall.php on line 9 Favorite animal is .</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	Buat file baru dengan nama homeSession.php di dalam direktori praktik_php, kemudian ketikkan kode berikut:

	<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) Jawab: Dari hasil pengamatan ketika kita langsung memanggil atau menjalankan file sessionCall.php tanpa kita create terlebih dahulu akan terjadi error. Karena, kita belum ada data yang ingin kita tampilkan atau kita inputkan. Sehingga ketika pertama kali mengakses akan terjadi error undefined. Kita harus mengcreate terlebih dahulu dengan menjalankan sessionCreate.php lalu kita jalankan kembali sessionCall.php sehingga akan tertampil seperti pada gambar terakhir.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>Username <input type="text" value="admin"/></p> <p>Password <input type="password" value="..."/></p> <p style="text-align: center;"><input type="button" value="Log in"/></p> </div> <p style="text-align: center;">Anda Berhasil Login, silahkan menuju Halaman HOME</p>
10	<p>Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (soal no 21) Jawab:</p>

	<p>Urutan proses</p> <p>1.. sessionLoginForm.html : disini untuk mengisi form login..</p> <p>2.. sessionLoginProses.php : disini kita cek apakah data yang dimasukkan cocok dan ada di database atau tidak..</p> <p>3.. homeSession.php : disini adalah tampilan awal atau homepage ketika berhasil login..</p> <p>4.. sessionLogout.php : disini adalah proses untuk melakukan logout dengan menghapus sesi atau jejak akun ketika melakukan login..</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