

NAMA : ZEEFANIA YIZREEL NEFRIT SINDHOE

KELAS : SIB 2C

NIM : 2241760117

JOBSHEET 9

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

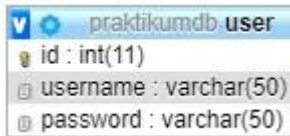
Session adalah salah satu fasilitas yang ada pada PHP yang digunakan untuk menyimpan data sementara ke dalam variabel, sehingga data tadi dapat diakses oleh pengguna selama variabel *session* tersebut tidak dikosongkan atau dihilangkan. *Session* pada prinsipnya hampir sama dengan *cookies*, yang membedakan 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
Cookies	Ya	Client	Ya	Ya	Ya	Ya	Ya
Session	Ya	Server	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*:

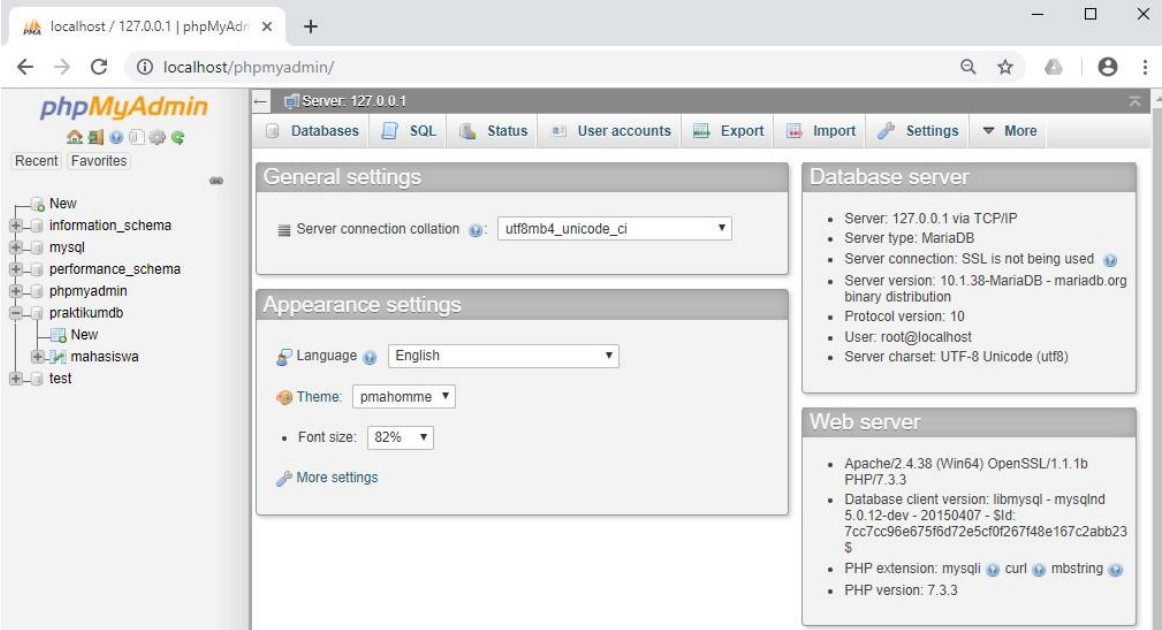


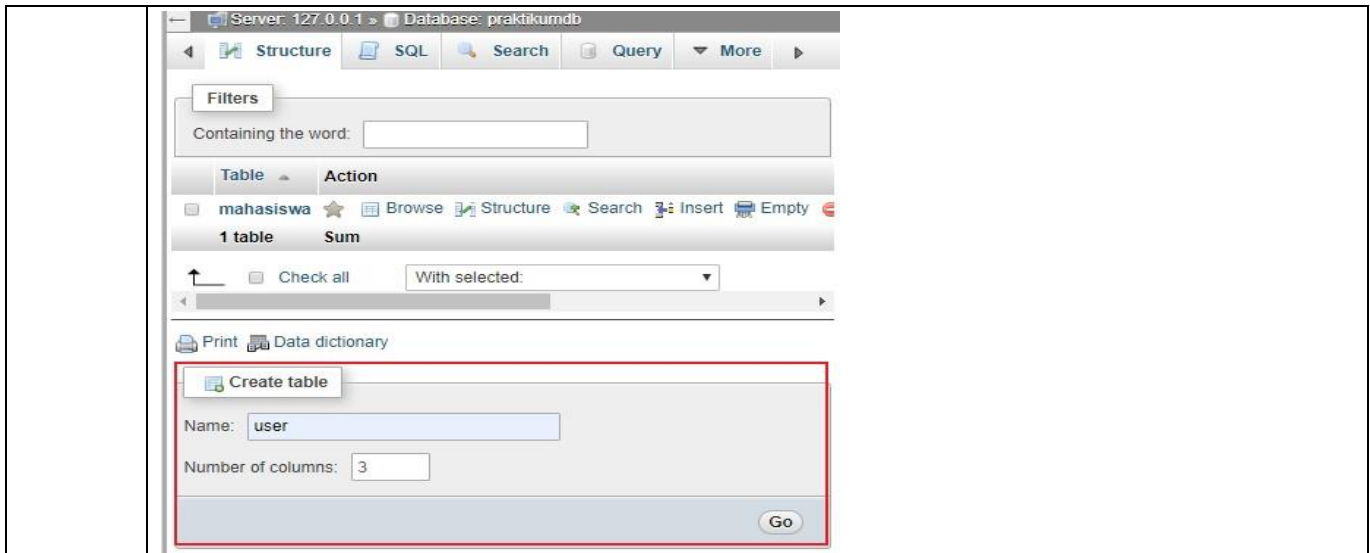
- tabel bernama *user*
- kolom tabel terdiri dari *id*, *username*, dan *password*
- *id* memiliki tipe data integer dan di atur sebagai *primary key*
- *username* dan *password* memiliki tipe data *varchar*
- *password* akan dienkripsi menggunakan fungsi *md5*

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:

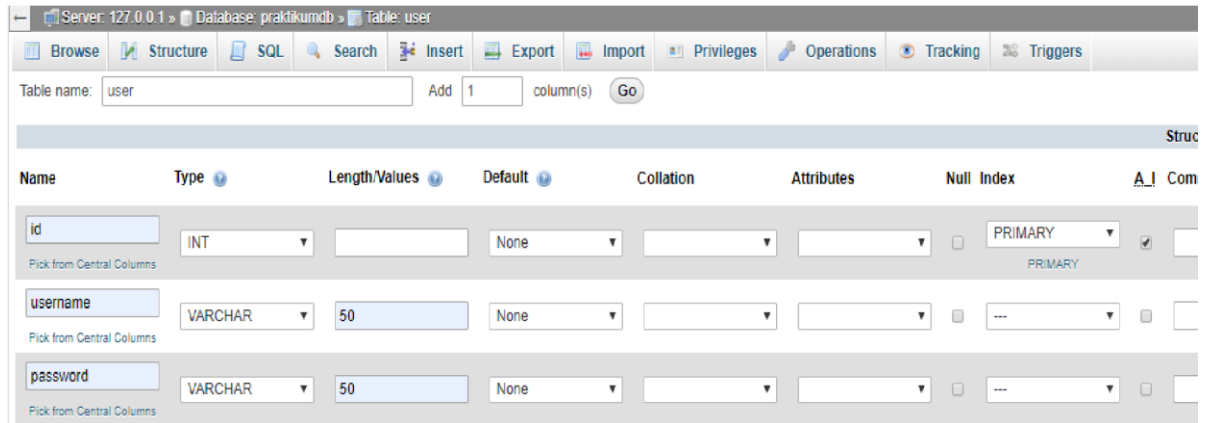
```
<?php    echo md5('string');  
?>
```

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

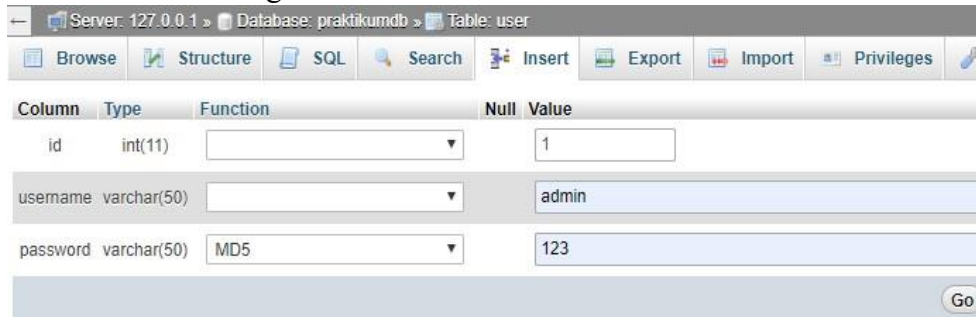
Langkah	Keterangan
1	Buka XAMPP kemudian jalankan web server dan database dengan mengklik tombol start pada modul Apache dan MySQL
2	Buka browser kemudian ketikkan localhost/phpmyadmin
3	Berikut adalah tampilan halaman phpmyadmin 
4	Buka database “prakwebdb” kemudian buat tabel baru bernama “user” dengan kolom “3”



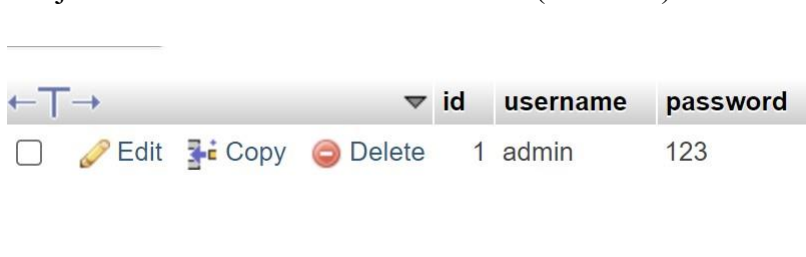
6 Berikut ini adalah kolom-kolom dalam tabel user



7 Masukkan data sebagai berikut ke dalam tabel user



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



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

Tabel user berhasil dibuat atau sudah ada sebelumnya.

```
<?php
// Soal: create table with mysqli_query()
```

```

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "prakwebdb";

// Buat koneksi ke database
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Periksa koneksi
if (!$conn) {
    die("Koneksi gagal: " . mysqli_connect_error());
}

// Query untuk membuat tabel
$query = "CREATE TABLE IF NOT EXISTS user (
id INT AUTO_INCREMENT PRIMARY KEY,
username VARCHAR(255) NOT NULL,
password VARCHAR(255) NOT NULL
)";

// Mengeksekusi query
if (mysqli_query($conn, $query)) {
    echo "Tabel user berhasil dibuat atau sudah ada sebelumnya.";
} else {
    echo "Error saat membuat tabel: " . mysqli_error($conn);
}

// Tutup koneksi
mysqli_close($conn);
?>

```

10

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

Data berhasil dimasukkan ke tabel user.

```

<?php
// Soal: insert data with mysqli_connect()

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "prakwebdb";

// koneksi ke database

```

```

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Periksa koneksi
if (!$conn) {
    die("Koneksi gagal: " . mysqli_connect_error());
}

// Data yang akan dimasukkan
$nama_pengguna = "username";
$kata_sandi = md5("kata_sandi");

// Query untuk memasukkan data ke dalam tabel user
$query = "INSERT INTO user (username, password) VALUES ('$nama_pengguna',
$kata_sandi)";

// Mengeksekusi query
if (mysqli_query($conn, $query)) {
    echo "Data berhasil dimasukkan ke tabel user.";
} else {
    echo "Error saat memasukkan data: " . mysqli_error($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 <code>koneksi.php</code> dari jobsheet sebelumnya, tersimpan di dalam direktori <code>praktik_php</code> .
2	<p>Buat file dengan nama <code>loginForm.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 <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	Buat file dengan nama <code>loginProses.html</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode program berikut ini:

	<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 <code>homeAdmin.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 admin </h2> 6 </body> 7 </html> </pre>
5	<p>Simpan file dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/loginForm.php</code></p>
6	<p>Lakukan login dengan mengetikkan username dan password yang datanya belum tersimpan di database. Jelaskan hasil pengamatanmu (soal no 4)</p> <div data-bbox="316 1249 863 1420"> <p>Username <input type="text"/></p> <p>Password <input type="password"/></p> <p><input type="button" value="Proses"/></p> </div> <hr/> <p>Koneksi berhasil!Anda gagal login. silahkan Login kembali</p> <ul style="list-style-type: none"> - ini gagal akibat username dan password yg tidak ada di database <p>Koneksi berhasil!Anda berhasil login. Silahkan menuju Halaman HOME</p> <ul style="list-style-type: none"> - berhasil karena memauskkan username dan pass sesuai dengan database <p>Dari hasil pengamatan, ketika kita memasukkan data yang memang tidak tersimpan di dalam database nya. Maka, kita tidak akan bisa masuk ke dalam homepage nya. Dikarenakan, ketika melakukan klik pada tombol proses. Selanjutnya akan ada proses pengecekan ke dalam database dan akan melihat apakah data yang diinputkan tersebut memang ada atau tidak. Jika, tidak ada maka user akan dipaksa login kembali sampai datanya tersebut memang benar-benar terdaftar di dalam 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>Koneksi berhasil! Anda berhasil login. Silahkan menuju Halaman HOME</p> <p>Dari hasil pengamatan, ketika kita memasukkan data yang benar. Maka, kita akan bisa masuk ke dalam homepage. Dikarenakan ketika dilakukan pengecekan ke dalam database. Data yang diinputkan terdaftar atau tersedia. Sehingga nanti akan dikirimkan informasi bahwa benarbenar terdaftar dan akan mempersilahkan untuk bisa melanjutkan ke tahap selanjutnya. Yaitu masuk ke dalam homepage.</p>
---	---

Praktikum Bagian 3. Menambahkan Kolom pada Tabel

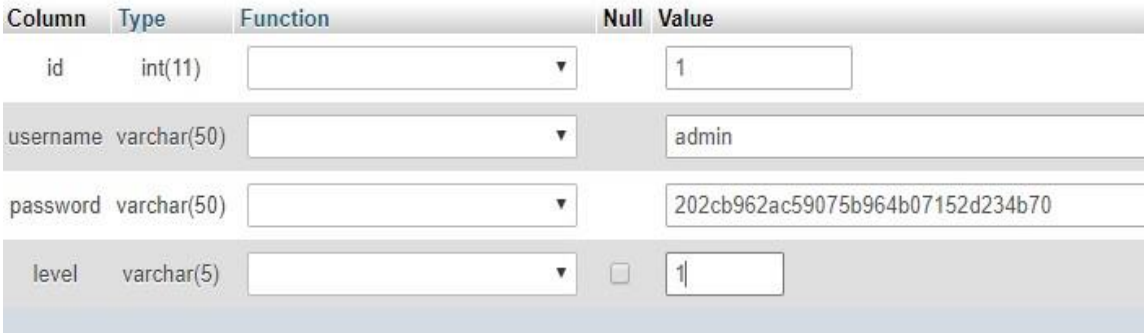
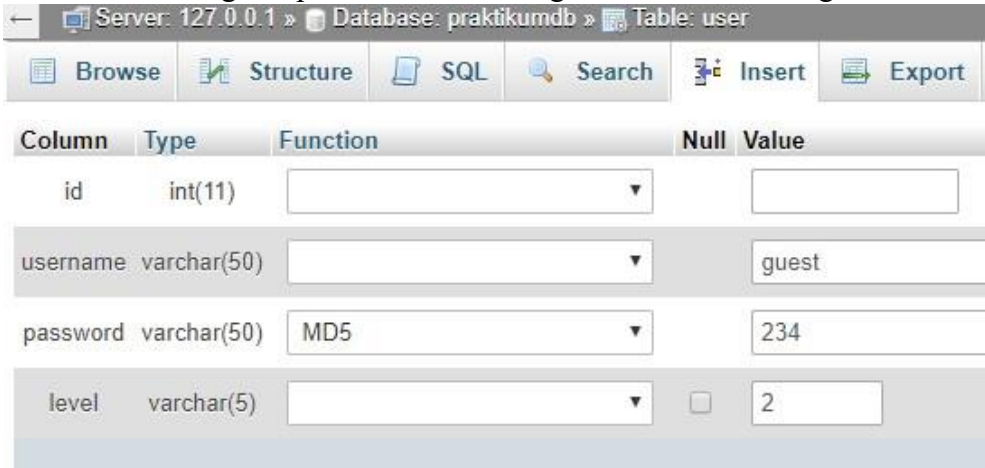

Setelah berhasil membuat login untuk *single user*, selanjutnya akan dibahas langkah-langkah untuk membuat login multiuser. Pertama yang perlu dipersiapkan adalah memberikan level akses di dalam *database*. Sebelumnya kita hanya memiliki kolom *id*, *username*, dan *password* di dalam tabel *user*. Untuk 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	Pada tabel <i>user</i> , ketikkan perintah SQL seperti berikut ini:

	 <p>The screenshot shows a web-based database management tool. At the top, there's a breadcrumb trail: "Server: 127.0.0.1 » Database: praktikumdb » Table: user". Below this is a navigation bar with buttons for "Browse", "Structure", "SQL", and "Search". The "SQL" button is active. A text box titled "Run SQL query/queries on table praktikumdb.user:" contains the following SQL commands:</p> <pre>1 ALTER TABLE user 2 ADD level VARCHAR(5) NOT NULL;</pre>
--	--

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>Jawab : Dari hasil pada proses diatas. Kita menambahkan data baru yaitu dengan username guest dan kita juga menambahkan data lainnya yaitu level dan ditambahkan pada admin dan guestnya.</p>
5	<p>Tunjukkan hasil <i>screenshot</i> isi tabel user (soal no 7)</p> 

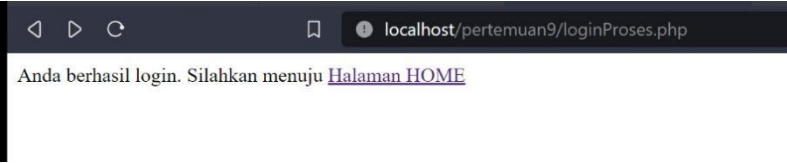
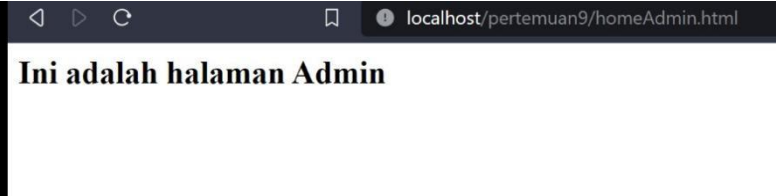
Praktikum Bagian 4. Login *Multiuser*

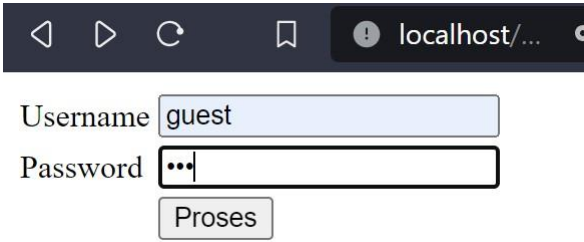
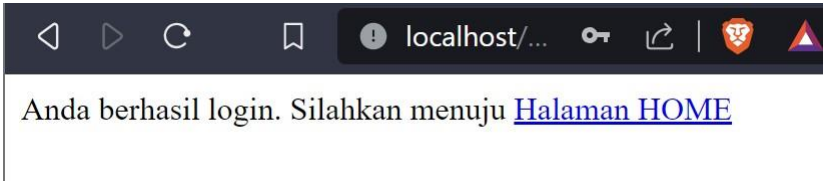

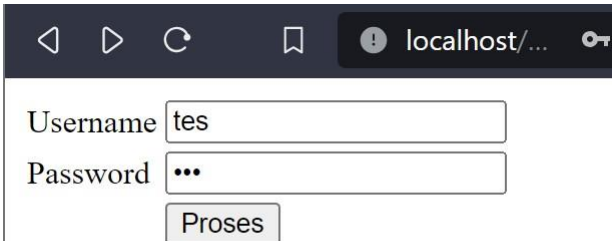

Pada praktikum ini, akan dijelaskan langkah-langkah untuk membuat login *multiuser*. Berbeda dengan login sebelumnya, selain melakukan pencocokan *username* dan *password* yang diinputkan melalui form dengan data *username* dan *password* yang disimpan di *database*, login *multiuser* akan mengenali *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	Buat file dengan nama loginMultiProses.php di dalam direktori praktik_php, kemudian ketikkan kode program berikut ini:

<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 } 15 }else if(\$row['level'] == 2){ 16 echo "Anda berhasil login. silahkan menuju "; ?> 17 Halaman HOME 18 } 19 }else{ 20 echo "Anda gagal login. silahkan "; ?> 21 Login kembali 22 } 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>Berdasarkan hasil nya ketika kita melakukan login menggunakan akun admin. Maka, akan ada pemfilteran untuk mendeteksi admin agar memiliki tampilan berbeda dengan menggunakan level. Namun, yang sudah pasti adalah menggunakan username dan password pada pemfilteran pertama. Lalu, dilanjutkan dengan level.</p>

6	<p>Lakukan login dengan mengetikkan data guest. Jelaskan hasil pengamatanmu (soal no 9)</p>    <p>Sama dengan yang admin. Untuk user lainnya atau guest memiliki tampilannya sendiri. Hal tersebut terjadi karena pemfilteran pertama dengan menggunakan username dan password untuk melihat apakah terdaftar atau tidak. Lalu, untuk penentuan halaman yang ditampilkan berdasarkan level. Jadi, nanti halaman yang ditampilkan berdasarkan level user itu sendiri.</p>
7	<p>Lakukan login dengan mengetikkan username dan password yang tidak tersimpan di database. Jelaskan hasil pengamatanmu (soal no 10)</p>  

	<div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <div style="border-bottom: 1px solid black; margin-bottom: 5px; width: 100%;"></div> <div style="display: flex; justify-content: space-between; align-items: center;"> Username <input style="width: 80%;" type="text"/> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> Password <input style="width: 80%;" type="password"/> </div> <div style="text-align: center; margin-top: 5px;"> <input type="button" value="Proses"/> </div> </div> <p>Ketika kita mengisi data yang memang tidak terdaftar maka, sistem akan mendeteksi jika terdaftar akan dilemparkan kembali ke form login. Untuk benar-benar mengisi username dan password yang benar agar bisa mengakses ke halaman selanjutnya.</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);
?>
```

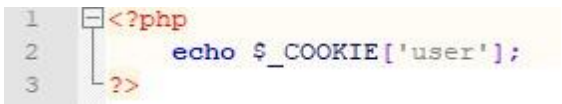
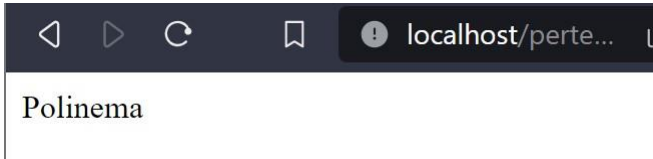
name, adalah nama *cookies* atau variabel yang menjadi pengenal *cookies* *value*, berisi nilai yang tersimpan dalam *cookies*
expire, adalah jangka waktu *cookies* akan disimpan di dalam komputer


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: <div style="background-color: #f0f0f0; padding: 10px; margin-top: 5px;"> <pre> 1 <?php 2 setcookie("user", "Polinema", time()+3600); 3 ?></pre> </div>
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 localhost/dasarWeb/praktik_php/cookiesCall.php
5	<p>Amati dan jelaskan hasil pengamatanmu (soal no 11)</p> <div> <div>Warning: Undefined array key "user" in D:\Xampp\htdocs\pertemuan9\cookiesCall.php on line 2</div> </div> <p>Dari hasil diatas kita dapatkan karena kita belum membuat cookies nya terlebih dahulu. Sehingga data yang memang kosong. Kita ingin akses akan menghasilkan undefined. Kita 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> <div>  </div> <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>Polinema</p> <p>Ketika di refresh dan dijalankan kembali hasil yang dihasilkan adalah tetap sama. Dikarenakan cookie masih dalam waktu 1 jam. Sehingga tidak ada perubahan.</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	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	<p>Amati dan jelaskan hasil yang dihasilkan dari langkah 2 dan 3, kemudian beri kesimpulan (soal no 14)</p> <p>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.</p> <p>Warning: Undefined array key "user" in <code>D:\Xampp\htdocs\pertemuan9\cookiesCall.php</code> on line 2</p>

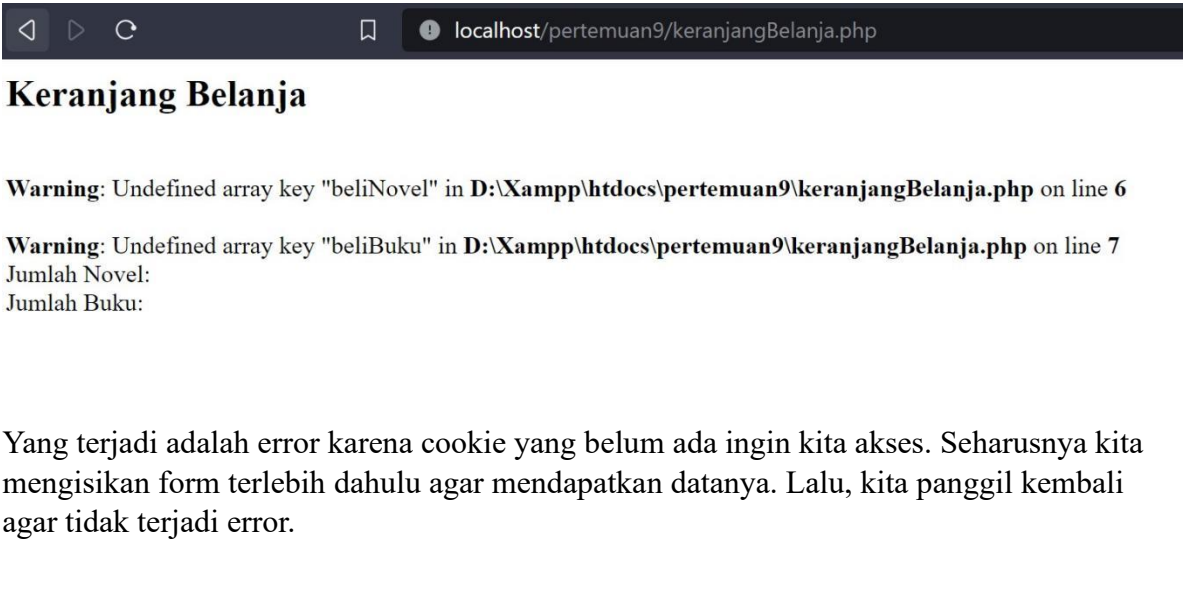

Praktikum Bagian 7. Penerapan *Cookies* pada Fitur Keranjang Belanja

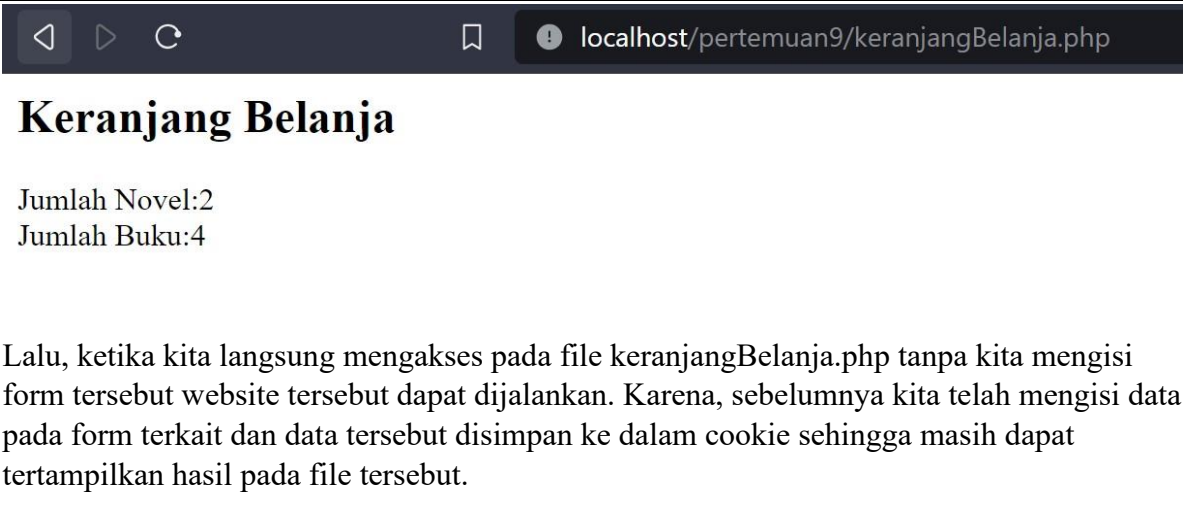
Salah satu contoh pemanfaatan *cookies* adalah pada fitur “keranjang belanja” pada aplikasi web toko online. Keranjang belanja berisi barang-barang yang akan dibeli oleh user. *Cookies* digunakan untuk

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	Amati dan jelaskan hasil yang ditampilkan (soal no 15) (dibawah)
---	--

	 <p>The screenshot shows a web browser at the URL <code>localhost/pertemuan9/keranjangBelanja.php</code>. The page title is "Keranjang Belanja". There are two PHP warnings displayed: "Warning: Undefined array key 'beliNovel' in D:\Xampp\htdocs\pertemuan9\keranjangBelanja.php on line 6" and "Warning: Undefined array key 'beliBuku' in D:\Xampp\htdocs\pertemuan9\keranjangBelanja.php on line 7". Below the warnings, the text "Jumlah Novel:" and "Jumlah Buku:" is visible. A paragraph explains that the error occurs because a cookie is missing, and a form should be filled out first to retrieve the data.</p>
6	Jalankan kode program langkah ke-1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/formBeli.html</code>
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>The screenshot shows two parts of a web application. The top part is a form at <code>localhost/pertemuan9/formBeli.html</code> with input fields for "Jumlah Novel yang dibeli" (containing the value 2) and "Jumlah Buku Teks yang dibeli" (containing the value 4), and a "Submit" button. The bottom part is the "Keranjang Belanja" page at <code>localhost/pertemuan9/keranjangBelanja.php</code>, which displays "Jumlah Novel:2" and "Jumlah Buku:4". A paragraph explains that when data is entered and the submit button is pressed, the application connects to another file to display the form's input.</p>

9	Tutup <i>browser</i> kemudian buka kembali <i>browser</i> kemudian jalankan kembali kode program langkah ke-3 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/keranjangBelanja.php</code>
10	Amati dan jelaskan hasil yang ditampilkan (soal no 17)
	 <p>Keranjang Belanja</p> <p>Jumlah Novel:2 Jumlah Buku:4</p> <p>Lalu, ketika kita langsung mengakses pada file <code>keranjangBelanja.php</code> 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*:

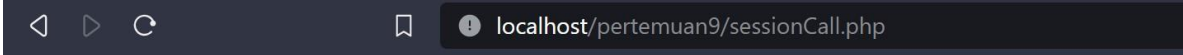
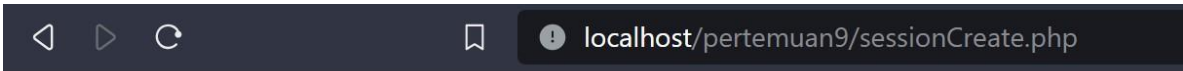
```
<?php
    session_start();
?>
```

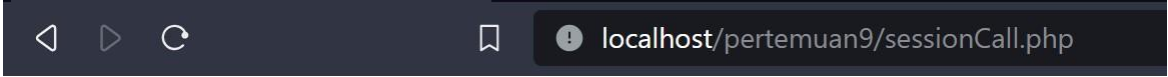
Setelah *session* dimulai, maka variabel *session* sudah dapat mulai digunakan. Variabel yang dimaksud adalah PHP global variable `$_SESSION`. Berikut adalah sintaks penggunaan variable global `$_SESSION`:

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

Ikuti langkah berikut untuk memahami cara kerja *session*:

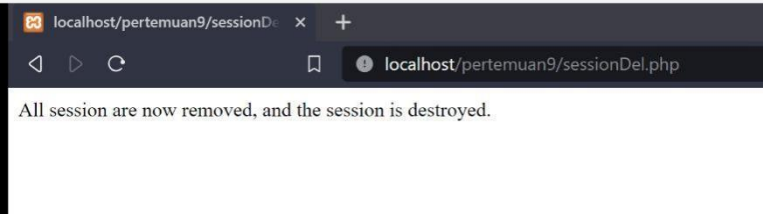
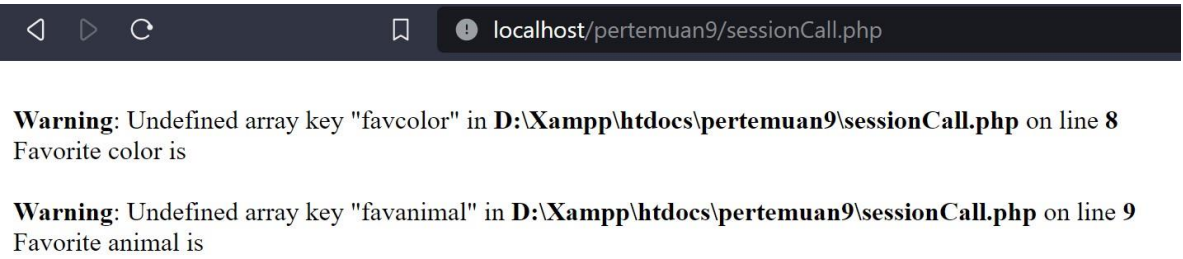
Langkah	Keterangan
1	Buat file baru dengan nama <code>sessionCreate.php</code> di dalam direktori <code>praktik_php</code> , kemudian ketikkan kode berikut:

	<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 <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>
4	<p>Buka <i>browser</i> dan jalankan kode program langkah ke-1 dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionCreate.php</code></p>
5	<p>Ulangi langkah ke-3</p>
6	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 18)</p> <div>  <p>Warning: Undefined array key "favcolor" in D:\Xampp\htdocs\pertemuan9\sessionCall.php on line 8 Favorite color is</p> <p>Warning: Undefined array key "favanimal" in D:\Xampp\htdocs\pertemuan9\sessionCall.php on line 9 Favorite animal is</p>  <p>Session variables are set.</p> </div>

	 <p>Favorite color is green Favorite animal is cat</p> <p>Dari hasil pengamatan ketika kita langsung memanggil atau menjalankan file <code>sessionCall.php</code> tanpa kita <code>create</code> 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 <code>undefined</code>. Kita harus <code>mengcreate</code> terlebih dahulu dengan menjalankan <code>sessionCreate.php</code> lalu kita jalankan kembali <code>sessionCall.php</code> sehingga akan tertampil seperti pada gambar terakhir.</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>
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>
4	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 19)</p>   <p>Warning: Undefined array key "favcolor" in <code>D:\Xampp\htdocs\pertemuan9\sessionCall.php</code> on line 8 Favorite color is</p> <p>Warning: Undefined array key "favanimal" in <code>D:\Xampp\htdocs\pertemuan9\sessionCall.php</code> on line 9 Favorite animal is</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>

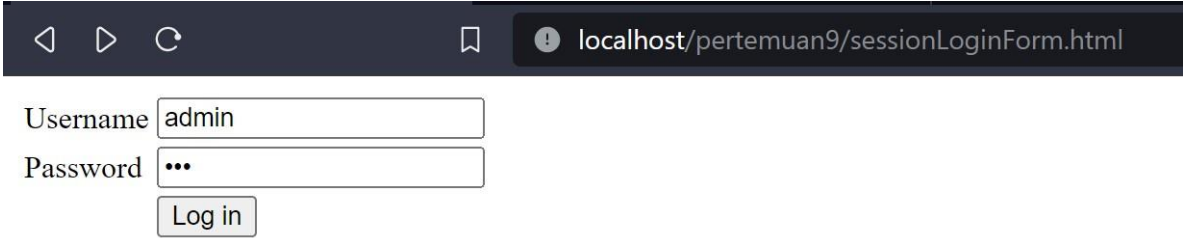
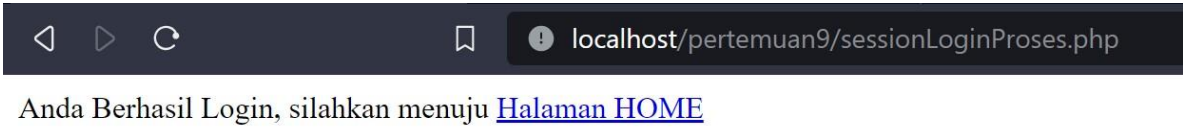
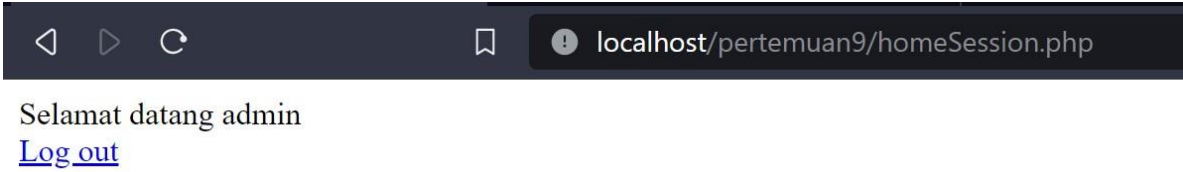
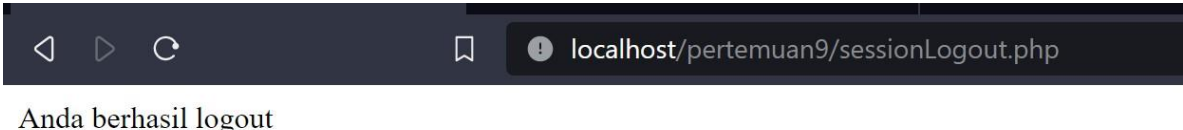
Praktikum Bagian 10. Penerapan *Session* pada Fitur Login

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

Langkah	Keterangan
1	Jalankan MySQL pada XAMPP
2	Pastikan file <code>koneksi.php</code> yang telah dibuat pada praktikum jobsheet sebelumnya ada di dalam direktori <code>praktik_php</code>
3	<p>Buat file baru dengan nama <code>sessionLoginForm.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="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 <code>sessionLoginProses.php</code> di dalam direktori <code>praktik_php</code>, 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 <code>homeSession.php</code> di dalam direktori <code>praktik_php</code>, 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 } 13 <?php 14 }else{ 15 echo "anda belum login. silahkan "; 16 Log In 17 } 18 <?php 19 ?> 20 </body> 21 </html> </pre>
6	<p>Buat file baru dengan nama <code>sessionLogout.php</code> di dalam direktori <code>praktik_php</code>, kemudian ketikkan kode berikut:</p> <pre> 1 <?php 2 session_start(); 3 session_destroy(); 4 5 echo "Anda berhasil logout"; 6 ?> </pre>

7	Buka <i>browser</i> dan jalankan kode program dengan mengetikkan <code>localhost/dasarWeb/praktik_php/sessionLoginForm.html</code>
8	Login menggunakan username admin.
9	<p>Amati dan jelaskan hasil yang ditampilkan (soal no 20)</p>   
	 <p>Dari hasil yang dijalankan diketahui ketika kita mengisi form untuk login lalu menekan tombol login. Maka, akan terjadi pengecekan pada file <code>sessionLoginProses.php</code> untuk melihat apakah data yang dimasukkan benar-benar ada atau tidak. Ketika benar ada pada database nya. Maka, akan dilanjutkan dengan informasi bahwa Anda sudah login dan diarahkan untuk ke link website selanjutnya. Ketika sudah di website tersebut, terdapat pilihan untuk melakukan logout dengan menekan link yang tersedia, link tersebut akan menjalankan file pada <code>sessionLogout.php</code> untuk melakukan proses penghapusan sesi pada sesi awal ketika login pada form nya.</p>

10	<p>Jelaskan urutan proses dari login hingga logout (sebutkan juga urutan file yang diproses) (soal no 21)</p> <p>Urutan proses</p> <ol style="list-style-type: none"> 1. sessionLoginForm.html : disini untuk melakukan pengisian form login. 2. sessionLoginProses.php : disini dilakukan pengecekan apakah data yang diinput sesuai dan ada pada database. 3. homeSession.php : disini adalah tampilan awal atau homepage ketika berhasil login. 4. sessionLogout.php : disini adalah proses untuk melakukan logout dengan menghapus sesi atau jejak akun ketika melakukan login.
----	---

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