

LAPORAN PRAKTIKUM PEMROGRAMAN WEB

PRAKTIKUM 10



Disusun oleh:

Nama : Afif Imam Rahadi

Nim : L0122006

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

1. Screenshot Source Code

connection.php

```
1  <?php
2      $servername = "localhost";
3      $username = "root";
4      $password = "";
5      $database = "user";
6      $connection = new mysqli($servername, $username, $password);
7
8      if ($connection->connect_error)
9      {
10         die("Connection failed : " . $connection->connect_error);
11     }
12
13     $createDatabase = "CREATE DATABASE IF NOT EXISTS user";
14     if(!$connection->query($createDatabase) === TRUE)
15     {
16         echo "Error create database : " . $connection->error;
17     }
18
19     if (!$connection->select_db($database) === TRUE) {
20         echo "Error select database : " . $connection->error;
21     }
22
23     $createTableQuery = "CREATE TABLE IF NOT EXISTS data_user (
24         id INT(8) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
25         username VARCHAR(32) NOT NULL,
26         password VARCHAR(32) NOT NULL
27     )";
28     if (!$connection->query($createTableQuery) === TRUE) {
29         echo "Error create table : " . $connection->error;
30     }
31     ?>
```

Gambar di atas adalah gambar pembuatan connection untuk databasenya. Seperti biasa, connection diawali dengan \$servername dan lainnya. Yang perlu dipastikan adalah bagian \$database karena ini harus sesuai dengan database yang akan dibuat. Selanjutnya, database dibuat dengan OOP. Berikutnya, jika koneksi gagal maka akan ada pesan error melalui connect_error.

Setelah koneksi berhasil, maka terdapat query untuk pembuatan database user dengan perintah SQL CREATE DATABASE. Lalu terdapat pengecekan error untuk pembuatan database. Setelah database terbuat, maka program akan memilih database tersebut untuk mengisi datanya dengan membuat table terlebih dahulu. Table dibuat dengan perintah sql CREATE TABLE data_user. Table ini berisi id dengan tipe data int, username dengan tipe data varchar, dan password dengan tipe data varchar. Terakhir, terdapat pengecekan untuk pembuatan table.

Register.php

```
1 <?php
2 include 'connection.php';
3
4 if (!empty($_POST['username']) && !empty($_POST['password'])) {
5     $username = $_POST['username'];
6     $password = $_POST['password'];
7     $hashed_password = password_hash($password, PASSWORD_DEFAULT);
8
9     $sql = "INSERT INTO data_user (username, password) VALUES ('$username', '$hashed_password')";
10    if ($connection->query($sql) === TRUE) {
11        header("Location: login.php?register=success");
12        exit();
13    } else {
14        echo "Error: " . $sql . "<br>" . $connection->error;
15    }
16 }
17 ?>
18
19 <!DOCTYPE html>
20 <html>
21 <head>
22     <title>Register</title>
23 </head>
24 <body>
25     <h2>Register</h2>
26     <form method="post" action="register.php">
27         Username: <input type="text" name="username" required><br><br>
28         Password: <input type="password" name="password" required><br><br>
29         <input type="submit" value="Register">
30     </form>
31     <p>Sudah punya akun? <a href="login.php">Login di sini</a></p>
32 </body>
33 </html>
34
```

Code di atas adalah code untuk bagian register. Code dimulai dengan include connection agar halaman ini dapat melakukan koneksi ke database yang sudah dibuat. Setelah itu, terdapat pengecekan untuk `$_POST['username']` dan `$_POST['password']` tidak kosong. Jika kedua input tersebut tidak kosong, variabel `$username` dan `$password` akan diisi dengan nilai yang dikirim melalui formulir. Password kemudian di-hash menggunakan `password_hash()` dengan algoritma `PASSWORD_DEFAULT` untuk meningkatkan keamanan. Selanjutnya, terdapat query untuk menginput data ke tabel `data_user` dengan username dan password yang telah di-hash. Setelah itu terdapat pengecekan kondisi jika perintah SQL berhasil dijalankan, pengguna akan diarahkan ke halaman `login.php` dengan parameter query `register=success` untuk menunjukkan bahwa registrasi berhasil.

Dan pada bagian bawah, terdapat html untuk user melakukan register form. Register ini hanya berisi username dan password. Pada bagian akhir, terdapat link jika user sudah memiliki akun, maka bisa langsung melakukan login.

Login.php

```
1 <?php
2 include 'connection.php';
3
4 if (!empty($_POST['username']) && !empty($_POST['password'])) {
5     $username = $_POST['username'];
6     $password = $_POST['password'];
7
8     $sql = "SELECT * FROM data_user WHERE username = '$username'";
9     $result = $connection->query($sql);
10
11     if ($result->num_rows > 0) {
12         $row = $result->fetch_assoc();
13
14         if (password_verify($password, $row["password"])) {
15             session_start();
16             $_SESSION["username"] = $username;
17             if (isset($_POST["remember"])) {
18                 setcookie("username", $username, time() + (86400 * 30), "/");
19             }
20             header("Location: dashboard.php");
21             exit();
22         } else {
23             $login_error = "Invalid password";
24         }
25     } else {
26         $login_error = "No user found with that username";
27     }
28 }
29
30
31 <!DOCTYPE html>
32 <html>
33 <head>
34 <title>Login</title>
35 </head>
36 <body>
37 <h2>Login</h2>
38 <?php if (isset($login_error)) { echo "<p style='color:red;'>$login_error</p>"; } ?>
39 <form method="post" action="login.php">
40     Username: <input type="text" name="username" required><br><br>
41     Password: <input type="password" name="password" required><br><br>
42     Remember me: <input type="checkbox" name="remember"><br><br>
43     <input type="submit" value="Login">
44 </form>
45 <p>Tidak punya akun? <a href="register.php">Register di sini</a></p>
46 </body>
47 </html>
48
```

Ini adalah code untuk bagian login. Sama pada awalnya terdapat include connection untuk database. Lalu diikuti dengan pengecekan kondisi apakah username dan password tidak kosong. Selanjutnya, terdapat query sql yaitu `SELECT * FROM data_user` untuk mengambil data dari table tersebut. Hasil query disimpan dalam variabel `$result`. Jika jumlah baris hasil query lebih dari nol, artinya pengguna dengan username tersebut ditemukan dalam basis data, data pengguna tersebut diambil dengan `fetch_assoc()` dan disimpan dalam variabel `$row`.

Selanjutnya menggunakan `password_verify()` untuk memverifikasi apakah password yang dimasukkan sesuai dengan password yang di-hash dan disimpan dalam basis data. Jika verifikasi berhasil, sesi PHP dimulai dengan `session_start()` dan username pengguna disimpan dalam variabel sesi `$_SESSION["username"]`. Jika kotak "Remember me" dicentang, sebuah cookie dengan nama username dan nilai username diset, yang akan berlaku selama 30 hari. Pengguna kemudian diarahkan ke halaman dashboard dan proses dihentikan dengan `exit()`. Jika password tidak valid, atau username tidak ditemukan dalam basis data, pesan kesalahan yang sesuai disimpan dalam variabel `$login_error`.

Dan pada bagian bawah, terdapat html untuk form login yang terdiri dari username dan password. Dan terdapat juga checkbox untuk remember me agar cookies di simpan sesuai yang sudah ditentukan.

Dashboard.php

```
1 <?php
2 session_start();
3 if (!isset($_SESSION["username"]) && !isset($_COOKIE["username"])) {
4     header("Location: login.php");
5     exit();
6 }
7
8 $username = isset($_SESSION["username"]) ? $_SESSION["username"] : $_COOKIE["username"];
9 ?>
10
11 <!DOCTYPE html>
12 <html>
13 <head>
14     <title>Dashboard</title>
15 </head>
16 <body>
17     <h2>Welcome, <?php echo $username; ?></h2>
18     <a href="logout.php">Logout</a>
19 </body>
20 </html>
21
```

Code di atas adalah code untuk halaman dashboard. Pertama, terdapat `session_start()` untuk memulai sesi, setelah login. Selanjutnya, memeriksa apakah variabel sesi `$_SESSION["username"]` atau cookie `$_COOKIE["username"]` telah di-set. Jika tidak ada variabel sesi maupun cookie yang di-set, user dianggap belum login, dan diarahkan ke halaman `login.php`.

Selanjutnya, variabel `$username` diisi dengan nilai dari variabel sesi `$_SESSION["username"]` jika ada, atau dari cookie `$_COOKIE["username"]` jika tidak ada variabel sesi. Ini memungkinkan pengguna tetap terautentikasi baik melalui sesi maupun cookie. Dan terakhir, pada bagian bawah terdapat html untuk menampilkan bahwa user sudah masuk ke dalam dashboard. Di dalamnya, terdapat link untuk melakukan logout.

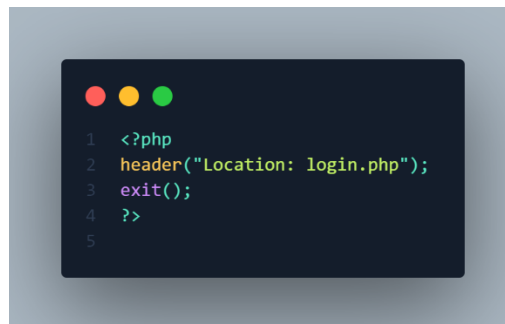
Logout.php

```
1 <?php
2 session_start();
3 session_unset();
4 session_destroy();
5
6 if (isset($_COOKIE["username"])) {
7     setcookie("username", "", time() - 3600, "/");
8 }
9
10 header("Location: login.php");
11 exit();
12 ?>
13
```

Code di atas adalah untuk proses logout user. Pertama, terdapat `session_start()` digunakan untuk memulai sesi yang sedang berlangsung. Selanjutnya, `session_unset()` yang digunakan untuk menghapus semua variabel data sesi saat ini. Setelah itu, `session_destroy()` dipanggil untuk menghancurkan sesi saat ini secara keseluruhan, menghapus semua data yang terkait dengan sesi tersebut di server.

Setelah sesi dihancurkan, skrip memeriksa apakah cookie dengan nama username ada menggunakan `isset($_COOKIE["username"])`. Jika cookie tersebut ada, `setcookie("username", "", time() - 3600, "/")` digunakan untuk menghapus cookie dengan mengatur waktu kedaluwarsa cookie ke waktu satu jam yang lalu (nilai negatif). Parameter `"/"` memastikan bahwa cookie dihapus untuk seluruh domain. Terakhir, user diarahkan kembali ke halaman `login.php` dan proses dihentikan dengan `exit()` untuk memastikan tidak ada kode tambahan yang dieksekusi setelah pengalihan.

Index.php



```
1 <?php
2 header("Location: login.php");
3 exit();
4 ?>
5
```

Ini adalah code untuk halaman utama atau index. Halaman utama ini akan langsung masuk ke halaman login karena menggunakan perintah `header`. Dan terakhir terdapat `exit` untuk menghentikan code php.

2. Screenshot Ouput

Register

Register

Username:

Password:

Sudah punya akun? [Login di sini](#)

Gambar di atas adalah halaman web dari register yang berupa form username dan password. Dan pada bagian bawah terdapat button register. Di bawah button, terdapat pesan jika sudah memiliki akun, maka dapat masuk ke halaman login.

Login

Login

Username:

Password:

Remember me: ☐

Tidak punya akun? [Register di sini](#)

Ini adalah halaman login. Tidak berbeda dengan register, hanya berbeda pada action setelah memencet button. Button di sini memiliki value Login. Dan pesan di bawahnya adalah pesan untuk user yang belum memiliki akun dapat melakukan register terlebih dahulu.

Dashboard

Welcome, afifrahadi

[Logout](#)

Ini adalah tampilan dari halaman dashboard yang berisi tulisan welcome dan nama username. Username ini diambil dari database username. Dan di bawah terdapat link untuk melakukan logout.