

LAPORAN PRAKTIKUM PBP KE-4

KONEKSI PHP DAN MYSQL



Disusun Oleh :

Ahmad Alvin Griffin

2406121140106

Lab D2

DEPARTEMEN ILMU KOMPUTER/INFORMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO 2023

BAB I

PENDAHULUAN

1.1 Tujuan

- a. Melakukan koneksi ke basis data MySQL dan menampilkan hasil query select ke halaman web.
- b. Membuat halaman untuk CRUD (create, read, update, delete) data dari/ ke basis data.
- c. Membuat file untuk login dan logout serta menambahkan penanganan pada file yang hanya dapat dibuka setelah login terlebih dahulu.
- d. Menerapkan penggunaan client-state management (session) untuk membuat shopping cart

1.2 Pertanyaan

1. Buatlah file untuk CRUD data berikut:

- view_customer.php, untuk menampilkan data pelanggan
- edit_customer.php, untuk edit data pelanggan
- add_customer.php, untuk menambah data customer baru
- delete_customer.php, untuk menghapus data customer tertentu.

2. Buatlah file untuk login seperti berikut:

- login.php seperti pada contoh di modul ini, jika berhasil redirect ke halaman view_customer.php.
- Tambahkan penanganan pada file view_customer.php sehingga file tersebut hanya dibuka oleh admin setelah berhasil login.
- Tambahkan link logout pada file view_customer.php untuk melakukan logout, lalu buatlah file logout.php seperti pada contoh di modul.

3. Buatlah file-file berikut untuk memproses shopping cart seperti yang dicontohkan di modul:

- view_books.php
- show_cart.php
- delete_cart.php

BAB II

DASAR TEORI

2.1 Koneksi ke Basis Data MySQL

2.1.1 Pengenalan MySQL

MySQL adalah sistem basis data relasional yang populer dan sering digunakan dalam pengembangan web. Ia menyediakan berbagai fitur untuk mengelola dan mengakses data.

2.1.2 Koneksi ke MySQL menggunakan PHP

Untuk menghubungkan PHP dengan MySQL, kita menggunakan ekstensi MySQLi (MySQL Improved). Dengan ekstensi ini, kita dapat melakukan operasi basis data seperti query, insert, update, dan delete.

Contoh Koneksi ke MySQL:

```
<?php
$db_host = 'localhost';
$db_database = 'bookorama';
$db_username = 'root';
$db_password = "";

// connect database
$db = new mysqli($db_host, $db_username, $db_password, $db_database);
if ($db->connect_errno){
    die ("Tidak menghubungkan database: <br />". $db->connect_error);
}

function test_input($data){
    $data = trim($data);
    $data = stripslashes($data);
```

```
$data = htmlspecialchars($data);  
return $data;  
}  
?>
```

2.2 CRUD Data dari/ke Basis Data

2.2.1 Create (Tambah Data)

Untuk menambah data ke dalam basis data MySQL, kita menggunakan perintah SQL INSERT. Kita juga dapat menggunakan PHP untuk memproses input dari formulir dan menambahkannya ke basis data.

2.2.2 Read (Baca Data)

Untuk membaca data dari basis data MySQL, kita menggunakan perintah SQL SELECT. Hasil query dapat ditampilkan dalam bentuk tabel atau format lainnya di halaman web.

2.2.3 Update (Edit Data)

Untuk mengedit data di basis data MySQL, kita menggunakan perintah SQL UPDATE. Kita juga dapat menggunakan PHP untuk memproses input dari formulir dan mengupdate data.

2.2.4 Delete (Hapus Data)

Untuk menghapus data dari basis data MySQL, kita menggunakan perintah SQL DELETE. Kita dapat menggunakan PHP untuk memproses permintaan penghapusan data.

2.3 Penggunaan Client-State Management (Session)

2.3.1 Pengenalan Session

Session adalah cara untuk menyimpan informasi di server untuk digunakan di berbagai halaman. Ini memungkinkan kita untuk menyimpan data pengguna selama mereka masih berada di situs web.

2.3.2 Implementasi Session dalam PHP

```
<?php
session_start(); // Memulai session

$_SESSION['username'] = 'admin'; // Menyimpan data dalam session
$username = $_SESSION['username']; // Mengakses data dari session
?>
```

2.4 Shopping Cart

2.4.1 Pengenalan Shopping Cart

Shopping cart adalah fitur yang memungkinkan pengguna untuk menambahkan item ke keranjang belanja dan melakukan pembelian di situs web.

2.4.2 Implementasi Shopping Cart

Shopping cart dapat diimplementasikan dengan menggunakan session untuk menyimpan item yang ditambahkan ke keranjang belanja. Item dapat ditambahkan, dihapus, atau diperbarui di dalam keranjang belanja.

BAB III

PEMBAHASAN

3.1 Source Code Dan Penjelasan File CRUD

3.1.1 view_customer.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstra
p.min.css" integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
    <title>Document</title>
  </head>
  <body>
    <div class="card">
      <div class="card-header">Customer Data</div>
      <div class="card-body">
        <br>
        <a class="btn btn-primary" href="add_customer.php">+ Add
Customer Data</a><br /><br />
        <table class="table table-striped">
          <tr>
            <th>No</th>
            <th>Name</th>
            <th>Address</th>
            <th>City</th>
            <th>Action</th>
          </tr>

          <?php
            session_start(); //inisialisasi session
            if (!isset($_SESSION['username'])) {
```

```

        header('Location: login.php');
    }
    // Include our login information
    require_once('./db_login.php');
    // Execute the query
    $query = "SELECT * FROM customers ORDER BY
customerid";

    $result = $db->query($query);
    if (!$result) {
        die("Could not query the database: <br />" .
$db->error . "<br/>Query: " . $query);
    }
    // Fetch and display the results
    $i = 1;
    while ($row = $result->fetch_object()) {
        echo '<tr>';
        echo '<td>' . $i . '</td>';
        echo '<td>' . $row->name . '</td>';
        echo '<td>' . $row->address . '</td>';
        echo '<td>' . $row->city . '</td>';
        echo '<td><a class="btn btn-warning btn-sm"
href="edit_customer.php?id=' . $row->customerid .
'">Edit<a/>&nbsp;&nbsp; ';
        echo '<a class="btn btn-danger btn-sm"
href="confirm_delete_customer.php?id=' . $row->customerid .
'">Delete<a/>';
        echo '</td>';
        echo '</tr>';
        $i++;
    }
    echo '</table>';
    echo '<br />';
    echo 'Total Rows = ' . $result->num_rows;
    $result->free();
    $db->close();
?>
<br><br>
<a class="btn btn-primary"
href="logout.php">Logout<a/>

```



```
        </table>
    </div>
</div>
</body>
</html>
```

Penjelasan :

view_customer.php adalah sebuah file PHP yang bertanggung jawab untuk menampilkan data pelanggan dari database. File ini juga mengimplementasikan beberapa fitur tambahan untuk memperkaya pengalaman pengguna. Pertama, dalam bagian <head>, terdapat definisi karakter set dan pengaturan viewport untuk memastikan tampilan yang konsisten di berbagai perangkat. Selain itu, file ini memuat file Bootstrap dari content delivery network (CDN) untuk memastikan tata letak dan elemen desain terintegrasi dengan baik.

Bagian utama dari file ini adalah penggunaan komponen Bootstrap seperti card untuk mempresentasikan data pelanggan. Card ini memiliki header dengan teks "Customer Data". Di dalam card body, terdapat beberapa elemen, seperti tombol untuk menambahkan data pelanggan baru (+ Add Customer Data) yang mengarahkan ke halaman add_customer.php. Selain itu, terdapat sebuah tabel yang menggunakan kelas Bootstrap untuk memberikan tata letak yang rapi dan responsif.

File ini juga menerapkan pengelolaan sesi, yang memungkinkan hanya pengguna yang telah login (diperiksa melalui variabel \$_SESSION['username']) yang dapat mengakses halaman ini. Jika pengguna belum login, maka akan diarahkan ke halaman login.php.

Selain itu, file ini memuat file db_login.php yang berisi informasi login ke basis data MySQL. Kemudian, sebuah query SQL dieksekusi untuk mengambil data pelanggan dari database, dan hasilnya ditampilkan dalam tabel menggunakan loop while.

Setiap baris dari data pelanggan termasuk nomor, nama, alamat, dan kota ditampilkan dalam baris terpisah dalam tabel. Selain itu, setiap baris juga memiliki dua tombol aksi, yaitu "Edit" dan "Delete". Tombol "Edit" mengarahkan pengguna ke halaman edit_customer.php dengan menyertakan parameter id untuk mengidentifikasi pelanggan yang akan diubah.

Di bagian akhir, terdapat informasi total jumlah baris yang ditampilkan dalam tabel. Terdapat juga tombol "Logout" yang mengarahkan pengguna ke halaman logout.php.

3.1.2 edit_customer.php

```
<?php
//File      : edit_customer.php
//Deskripsi : menampilkan form edit data customer dan mengupdate ke
database

require_once('db_login.php');
$id = $_GET['id']; //mendapatkan customerid yang dilewatkan ke
url

// mengecek apakah user belum menekan tombol submit
if (!isset($_POST["submit"])){
    $query = " SELECT * FROM customers WHERE customerid=" . $id.
    " ";

    //execute the query
    $result = $db->query($query);
    if (!$result){
        die ("Could not the query database: <br />" . $db->error);
    } else {
        while ($row = $result->fetch_object()){
            $name = $row->name;
            $address = $row->address;
            $city = $row->city;
        }
    }
} else {
    $valid = TRUE; //flag validasi
    $name = test_input($_POST['name']);
    if ($name == ''){
        $error_name = "Name is required";
        $valid = FALSE;
    } else if (!preg_match("/^[a-zA-Z ]*$/", $name)){
        $error_name = "Only letters and white space allowed";
    }
}
```

```

        $valid = FALSE;
    }

    $address = test_input($_POST['address']);
    if ($address == ''){
        $error_address = "Address is required";
        $valid = FALSE;
    }

    $city = $_POST['city'];
    if ($city == '' || $city == 'none'){
        $error_city = "City is required";
        $valid = FALSE;
    }

    //update data into database
    if ($valid){
        //escape inputs data
        $address = $db->real_escape_string($address);
//menghapus tanda petik
        //assign a query
        $query = " UPDATE customers SET name='".$name."',
address='".$address."', city='".$city.'" WHERE customerid='".$id.'" ";
        //execute the query
        $result = $db->query($query);
        if (!$result){
            die ("Could not the query the database: <br />" .
$db->error . '<br>Query:' . $query);
        } else{
            //ketika sudah di submit , maka akan langsung pindah
ke halaman view_customer.php
            $db->close();
            header('Location: view_customer.php');
        }
    }
}

?>

<!doctype html>

```

```

<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstra
p.min.css" rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
  </head>
  <body>
    <br>
    <div class="container">
      <div class="card">
        <div class="card-header">Edit Customers Data</div>
        <div class="card-body">
          <form method="POST" autocomplete="on"
action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]).'?id='
.$id; ?>">
            <div class="form-group">
              <label for="name">Nama:</label>
              <input type="text" class="form-control"
id="name" name="name" value="<?php echo $name;?>">
              <!-- Fungsi isset() memeriksa apakah suatu
variabel disetel, yang berarti variabel tersebut harus
dideklarasikan dan bukan NULL. -->
              <div class="text-danger"><?php if
(isset($error_name)) echo $error_name;?></div>
            </div>
            <div class="form-group">
              <label for="address">Address:</label>
              <textarea class="form-control" id="address"
name="address" rows="5"><?php echo $address;?></textarea>
              <div class="text-danger"><?php if
(isset($error_address)) echo $error_address;?></div>
            </div>

```

```

        <div class="form-group">
            <label for="city">City:</label>
            <select name="city" id="city" class="form-
control" required>
                <option value="none" <?php if
(isset($city)) echo 'selected="true"';?>>--Select a city--</option>
                <option value="Airport West" <?php if
(isset($city) && $city=="Airport West") echo
'selected="true"';?>>Airport West</option>
                <option value="Box Hill" <?php if
(isset($city) && $city=="Box Hill") echo 'selected="true"';?>>Box
Hill</option>
                <option value="Yarraville" <?php if
(isset($city) && $city=="Yarraville") echo
'selected="true"';?>>Yarraville</option>
            </select>
            <div class="text-danger"><?php
if(isset($error_city)) echo $error_city;?></div>
        </div>
        <br>
        <button type="submit" class="btn btn-primary"
name="submit" value="submit">Submit</button>
        <a href="view_customer.php" class="btn btn-
secondary">Cancel</a>
    </form>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.
bundle.min.js" integrity="sha384-
u10knCvxxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
</body>
</html>

<?php $db->close(); ?>

```

Penjelasan :

edit_customer.php adalah sebuah file PHP yang bertanggung jawab untuk menampilkan formulir pengeditan data pelanggan serta mengelola pembaruan data ke basis data MySQL. Pertama-tama, file ini memuat file db_login.php yang berisi informasi login ke basis data MySQL.

Kemudian, file ini mengambil customerid dari URL menggunakan \$_GET['id'] untuk menentukan pelanggan mana yang akan diubah. Ini memungkinkan file untuk mengenali dan memuat data pelanggan yang tepat untuk diedit.

Selanjutnya, file ini memiliki mekanisme validasi yang dilakukan sebelum data diperbarui. Jika pengguna belum menekan tombol "Submit", maka file akan mengeksekusi query SQL untuk mengambil data pelanggan dari database. Data tersebut akan ditampilkan di formulir untuk memungkinkan pengguna mengeditnya. Informasi nama, alamat, dan kota akan diisi otomatis sesuai dengan data pelanggan yang ada.

Saat pengguna menekan tombol "Submit", file ini akan memproses data yang dikirimkan melalui metode POST. Data tersebut akan diperiksa untuk memastikan validitasnya. Misalnya, nama harus diisi dan hanya diperbolehkan berisi huruf dan spasi.

Jika validasi berhasil, data akan diperbarui di basis data menggunakan query SQL UPDATE. Sebelum query dieksekusi, data dari formulir akan di-escape untuk mencegah serangan SQL injection menggunakan `real_escape_string()`.

Jika pembaruan data berhasil, pengguna akan diarahkan ke halaman view_customer.php menggunakan `header('Location: view_customer.php')`. Hal ini memberikan umpan balik kepada pengguna bahwa data telah berhasil diperbarui dan mereka akan melihat tampilan terbaru dari data pelanggan.

Jika terdapat kesalahan selama proses validasi atau pembaruan, pesan kesalahan akan ditampilkan di bawah masing-masing kolom formulir yang terkait. Ini memberikan pengguna informasi tentang apa yang harus diperbaiki sebelum melanjutkan.

Terakhir, edit_customer.php juga memastikan untuk membebaskan sumber daya dan menutup koneksi ke basis data setelah proses selesai untuk memastikan efisiensi dan menghindari kebocoran sumber daya.

3.1.3 delete_customer.php

```
<?php
    $id = $_GET['id'];
    // Include our login information
    require_once('./db_login.php');
    //Assign a query
    $query = " DELETE FROM customers WHERE customerid=" . $id . " ";
    // Execute the query
    $result = $db->query($query);
    if (!$result) {
        die("Could not query the database: <br />" . $db->error);
    } else {
        $db->close();
        header('Location: view_customer.php');
    }
    //close db connection
    $db->close();
?>
```

Penjelasan :

File delete_customer.php bertanggung jawab untuk menghapus data pelanggan dari basis data MySQL. Pertama-tama, file ini mengambil customerid dari URL menggunakan \$_GET['id']. Ini memungkinkan file untuk menentukan pelanggan mana yang akan dihapus.

Selanjutnya, file ini memuat file db_login.php yang berisi informasi login ke basis data MySQL. Setelah itu, file ini membangun dan mengeksekusi query SQL DELETE untuk menghapus data pelanggan dari tabel customers berdasarkan customerid.

Jika query berhasil dieksekusi, pengguna akan diarahkan kembali ke halaman view_customer.php menggunakan header('Location: view_customer.php'). Hal ini memberikan umpan balik kepada pengguna bahwa data telah berhasil dihapus dan mereka akan melihat tampilan terbaru dari data pelanggan.

Jika terdapat kesalahan selama proses penghapusan data, pesan kesalahan akan ditampilkan. Ini memberikan pengguna informasi tentang apa yang mungkin telah terjadi dan memungkinkan mereka untuk mengatasi masalah tersebut.

Terakhir, file ini memastikan untuk membebaskan sumber daya dan menutup koneksi ke basis data setelah proses selesai untuk memastikan efisiensi dan menghindari kebocoran sumber daya.

3.2 Source Code File Login

3.2.1 login.php

```
<?php
//File           : login.php
//Deskripsi : menampilkan form login dan melakukan login ke halaman
admin.php

session_start(); //inisialisasi session
require_once('./db_login.php');

//cek apakah user sudah submit form
if (isset($_POST["submit"])) {
    $valid = TRUE; //flag validasi
    //cek validasi email
    $email = test_input($_POST['email']);
    if ($email == '') {
        $error_email = "Email is required";
        $valid = FALSE;
    } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $error_email = "Invalid email format";
        $valid = FALSE;
    }
    //cek validasi password
    $password = test_input($_POST['password']);
    if ($password == '') {
        $error_password = "Password is required";
        $valid = FALSE;
    }
}
```



```

        //cek validasi
        if ($valid) {
            //Assign a query
            $query = " SELECT * FROM admin WHERE email='" . $email . "'
AND password='" . md5($password) . "' ";
            // Execute the query
            $result = $db->query($query);
            if (!$result) {
                die("Could not query the database: <br />" . $db->error);
            } else {
                if ($result->num_rows > 0) { //login berhasil
                    $_SESSION['username'] = $email;
                    $_SESSION['kategori'] = 'admin';
                    header('Location: view_customer.php');
                    exit;
                } else { //login gagal
                    echo '<span class="error">Combination of username and
password are not correct.</span>';
                }
            }
            //close db connection
            $db->close();
        }
    }
?>
<?php include('./header.html') ?>
<br>
<div class="card">
    <div class="card-header">Login Form</div>
    <div class="card-body">
        <form method="POST" autocomplete="on" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
            <div class="form-group">
                <label for="email">Email</label>
                <input type="email" class="form-control" id="email"
name="email" size="30" value="<?php if (isset($_POST['submit'])) echo
$email; ?>">
            </div>
        </form>
    </div>
</div>

```

```

        <div class="error"><?php if (isset($error_email)) echo
$error_email; ?></div>
    </div>
    <div class="form-group">
        <label for="password">Password</label>
        <input type="password" class="form-control"
id="password" name="password" value="">
        <div class="error"><?php if (isset($error_password))
echo $error_password; ?></div>
    </div>
    <button type="submit" class="btn btn-primary"
name="submit" value="submit">Login</button>
    </form>
</div>
</div>
<?php include('../footer.html') ?>

```

Penjelasan :

File login.php ini jika berhasil akan di redirect ke halaman view_customer

File login.php bertanggung jawab untuk menangani proses login ke halaman admin.php.

Pertama, file ini memulai atau menginisialisasi session menggunakan session_start(), yang memungkinkan pengguna untuk tetap terautentikasi selama sesi berlangsung.

Kemudian, file ini memuat file db_login.php yang berisi informasi login ke basis data MySQL.

Selanjutnya, file ini memeriksa apakah pengguna telah mengirimkan formulir login dengan menggunakan kondisi if (isset(\$_POST["submit"])). Jika formulir telah dikirim, maka validasi akan dilakukan.

Validasi pertama adalah untuk memeriksa apakah alamat email yang dimasukkan oleh pengguna sesuai dengan format email yang benar. Jika tidak, pesan kesalahan akan ditetapkan dan flag validasi akan diubah menjadi FALSE.

Validasi kedua adalah untuk memeriksa apakah kolom password tidak kosong. Jika kosong, pesan kesalahan akan ditetapkan dan flag validasi akan diubah menjadi FALSE.

Setelah validasi, jika semua data masukan valid, maka file ini akan membangun dan menjalankan query SQL untuk memeriksa kecocokan kombinasi alamat email dan kata sandi di tabel admin. Jika query berhasil dieksekusi, dan jika ada baris yang cocok, maka login berhasil. Pengguna akan diarahkan ke halaman view_customer.php dan informasi sesi seperti nama pengguna dan kategori (dalam hal ini, "admin") akan disimpan dalam variabel \$_SESSION.

Jika tidak ada kombinasi yang cocok atau ada kesalahan selama proses query, maka pesan kesalahan akan ditampilkan kepada pengguna.

Terakhir, file ini memuat formulir login yang berisi kolom untuk memasukkan email dan kata sandi. Jika formulir telah dikirim sebelumnya, nilai yang dimasukkan oleh pengguna akan tetap ada di dalam formulir. Pesan kesalahan (jika ada) juga akan ditampilkan di bawah kolom terkait.

3.2.2 view_customer 2

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstra
p.min.css"          rel="stylesheet"          integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaIlGyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
  </head>
  <body>
    <div class="container">
      <div class="card mt-4">
        <div class="card-header">Customers Data 2</div>
        <div class="card-body">
          <br>
          <a          class="btn          btn-primary"
href="add_customer.php">+ Add Customer Data</a><br /><br />
```

```

<table class="table table-striped">
    <tr>
        <th>No</th>
        <th>Name</th>
        <th>Address</th>
        <th>City</th>
        <th>Action</th>
    </tr>

    <?php
        // include our login information
        require_once('db_login.php');

        // execute the query
        // yang berbeda dengan sebelumnya yaitu
pada view_customer untuk nama panggilan label pada fetch object harus
sama dengan di database

        // tetapi kalau view_customer2 pada
pemanggilan querynya dirubah untuk tiap label sesuai dengan keinginan
masing masing seperti dibawah

        // dan pada fetch object menyesuaikan
dengan pemanggilan query yang telah diubah

        $query = " SELECT customerid AS ID, name
AS Nama, address AS Alamat, city AS Kota FROM customers ORDER BY
customerid ";

        $result = $db->query($query);
        if (!$result) {
            die("Could not the query the database:
<br />" . $db->error . "<br>Query: " . $query);
        }

        // fetch and display the results
        $i = 1;
        while ($row = $result->fetch_object()) {
            echo '<tr>';
            echo '<td>' . $i . '</td>';
            echo '<td>' . $row->Nama . '</td>';
            echo '<td>' . $row->Alamat . '</td>';
            echo '<td>' . $row->Kota . '</td>';

```


Penting untuk dicatat bahwa sebelum menampilkan data pelanggan, file ini melakukan beberapa operasi penting. Pertama, file ini memastikan bahwa pengguna yang mengakses halaman ini adalah admin yang telah berhasil login. Ini dilakukan dengan memeriksa apakah sesi `$_SESSION['kategori']` telah diatur dan bernilai 'admin'. Jika tidak, pengguna akan diarahkan kembali ke halaman login.

Setelah memastikan bahwa pengguna adalah admin yang berhasil login, file ini memuat file `db_login.php` untuk mendapatkan akses ke database MySQL.

Kemudian, file ini menjalankan query SQL untuk mengambil data pelanggan dari tabel `customers`. Pada query ini, label untuk kolom-kolom seperti ID, Nama, Alamat, dan Kota telah diubah menggunakan alias. Hal ini memudahkan dalam pengolahan data selanjutnya.

Hasil dari query kemudian ditampilkan dalam bentuk tabel menggunakan perulangan `while`. Setiap baris dari hasil query direpresentasikan sebagai satu baris dalam tabel HTML. Data ditampilkan dalam kolom-kolom yang sesuai dengan label yang telah ditentukan sebelumnya.

Setiap baris juga menyertakan dua tombol aksi: "Edit" dan "Delete". Tombol "Edit" mengarahkan pengguna ke halaman `edit_customer.php` dengan menyertakan ID pelanggan sebagai parameter GET. Tombol "Delete" mengarahkan pengguna ke halaman `confirm_delete_customer.php` dengan menyertakan ID pelanggan sebagai parameter GET.

Di bagian bawah tabel, ditampilkan juga jumlah total baris yang dihasilkan oleh query.

Terakhir, file ini memuat file JavaScript dari Bootstrap untuk menambahkan fungsionalitas tata letak halaman.

3.2.3 `logout.php` dan link `view_customer.php`

```
<?php
//File           : logout.php
//Deskripsi     : untuk logout (menghapus session yang dibuat saat login)
```

```
session_start();  
if (isset($_SESSION['username'])) {  
    unset($_SESSION['username']);  
    session_destroy();  
}  
header('Location: login.php');  
?>
```

Penjelasan :

File `view_customer2.php` adalah halaman yang menampilkan data pelanggan. Dalam halaman ini, terdapat tabel yang menampilkan informasi pelanggan seperti ID, Nama, Alamat, dan Kota. Di bagian atas halaman, terdapat tombol "Add Customer Data" yang mengarahkan pengguna ke halaman `add_customer.php` untuk menambahkan data pelanggan baru.

Penting untuk dicatat bahwa sebelum menampilkan data pelanggan, file ini melakukan beberapa operasi penting. Pertama, file ini memastikan bahwa pengguna yang mengakses halaman ini adalah admin yang telah berhasil login. Ini dilakukan dengan memeriksa apakah sesi `$_SESSION['kategori']` telah diatur dan bernilai 'admin'. Jika tidak, pengguna akan diarahkan kembali ke halaman login.

Setelah memastikan bahwa pengguna adalah admin yang berhasil login, file ini memuat file `db_login.php` untuk mendapatkan akses ke database MySQL.

Kemudian, file ini menjalankan query SQL untuk mengambil data pelanggan dari tabel `customers`. Pada query ini, label untuk kolom-kolom seperti ID, Nama, Alamat, dan Kota telah diubah menggunakan alias. Hal ini memudahkan dalam pengolahan data selanjutnya.

Hasil dari query kemudian ditampilkan dalam bentuk tabel menggunakan perulangan `while`. Setiap baris dari hasil query direpresentasikan sebagai satu baris dalam tabel HTML. Data ditampilkan dalam kolom-kolom yang sesuai dengan label yang telah ditentukan sebelumnya.

Setiap baris juga menyertakan dua tombol aksi: "Edit" dan "Delete". Tombol "Edit" mengarahkan pengguna ke halaman `edit_customer.php` dengan menyertakan ID pelanggan sebagai parameter GET. Tombol "Delete" mengarahkan pengguna ke halaman `confirm_delete_customer.php` dengan menyertakan ID pelanggan sebagai parameter GET.

Di bagian bawah tabel, ditampilkan juga jumlah total baris yang dihasilkan oleh query.

Terakhir, file ini memuat file JavaScript dari Bootstrap untuk menambahkan fungsionalitas tata letak halaman.

```
<br><br>
<a class="btn btn-primary" href="logout.php">Logout</a>
</table>
</div>
</div>
```

Tempatkan kode di atas di bawah baris `<table>` sebelumnya. Ini akan menambahkan tombol "Logout" yang mengarahkan pengguna ke halaman `logout.php` ketika diklik.

3.3 Source Code File Shopping Cart

3.3.1 view_books.php

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstra
p.min.css"          rel="stylesheet"          integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaIlGyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
  </head>
  <body>
    <div class="container">
```



```

<div class="card mt-4">
    <div class="card-header">Books Data</div>
    <div class="card-body">
        <br>
        <table class="table table-striped">
            <tr>
                <th>ISBN</th>
                <th>Author</th>
                <th>Title</th>
                <th>Price</th>
                <th>Action</th>
            </tr>

            <?php
            // include our login information
            require_once('db_login.php');

            // execute the query
            $query = " SELECT * FROM books ORDER BY
isbn ";

            $result = $db->query($query);
            if (!$result){
                die ("Could not the query the
database: <br />" . $db->error . "<br>Query: " . $query);
            }

            // fetch and display the results
            while ($row = $result->fetch_object()){
                echo '<tr>';
                echo '<td>' . $row->isbn. '</td>';
                echo '<td>' . $row->author. '</td>';
                echo '<td>' . $row->title. '</td>';
                echo '<td> $' . $row->price. '</td>';
                echo '<td><a class="btn btn-primary"
href="show_cart.php?id=' . $row->isbn. '">Add to Cart</a></td>';
                echo '</tr>';
            }
            echo '</table>';
            echo '<br />';

```

```

        echo 'Total Rows = ' . $result->num_rows;
        $result->free();
        $db->close();
    ?>

</div>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.
bundle.min.js"
integrity="sha384-
u10knCvxxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
</body>
</html>

```

Penjelasan :

Kode `view_books.php` merupakan sebuah file HTML yang digunakan untuk menampilkan data buku dalam bentuk tabel. Pertama, file ini mendefinisikan struktur dokumen HTML dengan menggunakan tag `<html>`, `<head>`, dan `<body>`. Di bagian `<head>`, didefinisikan metadata dokumen seperti pengaturan karakter dan viewport untuk tampilan responsif pada berbagai perangkat.

Judul halaman ditetapkan dengan menggunakan tag `<title>` dalam bagian `<head>`, yang akan terlihat pada judul tab browser.

File ini menggunakan Bootstrap, sebuah kerangka kerja CSS populer, untuk memperindah tata letak dan gaya elemen halaman. Bootstrap diimpor dari CDN (Content Delivery Network) untuk memastikan bahwa versi terbaru dari Bootstrap digunakan dalam halaman.

Di bagian `<body>`, terdapat elemen `<div>` dengan kelas "container". Ini digunakan untuk mengelompokkan konten dan memberikan sedikit pola desain. Di dalamnya, terdapat sebuah kartu (`<div>` dengan kelas "card") yang berisi data buku.

Kartu ini memiliki header dengan teks "Books Data" yang ditampilkan di bagian atas. Di dalam tubuh kartu (<div> dengan kelas "card-body"), terdapat sebuah tabel (<table>) untuk menampilkan data buku.

Pada tabel, didefinisikan baris pertama sebagai header tabel (<tr>), yang berisi judul kolom seperti ISBN, Author, Title, dan Price. Kemudian, kode PHP digunakan untuk mengakses basis data dan mengambil data buku.

Perulangan while digunakan untuk mengambil setiap baris hasil query dan menampilkannya dalam bentuk baris tabel. Data dari masing-masing kolom (ISBN, Author, Title, dan Price) diambil dari objek hasil query dan dimasukkan ke dalam sel tabel (<td>).

Setiap baris juga memiliki sebuah tombol "Add to Cart" yang mengarahkan ke halaman show_cart.php dengan mengirimkan ID buku sebagai parameter melalui URL.

Pada akhir dokumen, terdapat pautan ke Bootstrap JavaScript untuk memastikan interaktivitas yang tepat pada halaman.

3.3.2 show_cart.php

```
<?php
//File      : show_cart.php
//Deskripsi : untuk menambahkan item ke shopping cart dan menampilkan
isi shopping cart
    session_start();
    $id = $_GET['id'];
    if($id != ""){
        //jika $_SESSION['cart'] belum ada, maka inisialisasi dengan
array kosong
        //$_SESSION['cart'] merupakan array asosiatif
        //indeksnya berisi isbn buku yang dipilih
        //value-nya berisi jumlah (qty) dari buku dengan isbn tersebut
        if (!isset($_SESSION['cart'])){
            $_SESSION['cart'] = array();
        }
        //jika $_SESSION['cart'] dengan indeks isbn buku yang dipilih
sudah ada, tambahkan value-nya
```

```
// dengan 1, jika belum ada, buat elemen baru dengan indeks
tersebut dan set nilainya dengan 1
    if (isset($_SESSION['cart'][$id])){
        $_SESSION['cart'][$id]++;
    } else{
        $_SESSION['cart'][$id] = 1;
    }
}
?>

<!--Menampilkan isi shopping cart-->
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <title>Bootstrap demo</title>
        <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.1/dist/css/bootstra
p.min.css"
rel="stylesheet"
integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaIlGyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
    </head>
    <body>
        <br>
        <div class="container">
            <div class="card">
                <div class="card-header">Shopping Cart</div>
                <div class="card-body">
                    <br>
                    <table class="table table-striped">
                        <tr>
                            <th>ISBN</th>
                            <th>Author</th>
                            <th>Title</th>
                            <th>Price</th>
                            <th>Qty</th>
                            <th>Price * Qty</th>
                        </tr>
                    </table>
                </div>
            </div>
        </div>
    </body>
</html>
```

```

        </tr>
        <?php
        //include our login information
        require_once('db_login.php');
        $sum_qty = 0; //inisialisasi total item di
shopping cart
        $sum_price = 0; //inisialisasi total price di
shopping cart
        if (is_array($_SESSION['cart'])) {
            foreach ($_SESSION['cart'] as $id =>
$quantity) {
                $query = " SELECT * FROM books WHERE
isbn='".$id."'";
                $result = $db->query($query);
                if (!$result) {
                    die("Could not query the
database: <br>".$db->error."<br>Query: " . $query);
                }
                while ($row = $result-
>fetch_object()) {
                    echo '<tr>';
                    echo '<td>'.$row->isbn.'</td>';
                    echo '<td>'.$row->author.'</td>';
                    echo '<td>'.$row->title.'</td>';
                    echo '<td>'.$row->price.'</td>';
                    echo '<td>'.$quantity.'</td>';
                    $price = $row->price;
                    echo
'<td>'.$price*$quantity.'</td>';
                    $sum_price = $sum_price + ($row-
>price * $quantity);
                    $sum_qty = $sum_qty + $quantity;
                }
            }
            echo
'<tr><td></td><td></td><td></td><td></td><td>'.$sum_qty.'</td><td>'.
$sum_price.'</td></tr>';
            $result->free();
            $db->close();

```

```

        } else{
            echo '
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| There is no item in shopping cart | | | | | |

';
        }
    }
}
?>
</table>
Total items = <?php echo $sum_qty; ?><br><br>
<a class="btn btn-primary" href="view_books.php">Continue Shopping</a>
<a class="btn btn-danger" href="delete_cart.php">Empty Cart</a><br /><br />
</div>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.
bundle.min.js" integrity="sha384-
u10knCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
</body>
</html>

```

Penjelasan :

Kode `show_cart.php` bertujuan untuk menangani penambahan item ke dalam shopping cart dan menampilkan isi dari shopping cart. Pertama, file ini memulai atau memulihkan sesi dengan menggunakan `session_start()`. Selanjutnya, ia mengambil ID buku yang dikirimkan melalui parameter URL dengan menggunakan `$_GET['id']` dan menyimpannya dalam variabel `$id`.

Kemudian, kode melakukan pengecekan apakah `$id` tidak kosong. Jika tidak kosong, maka akan memproses penambahan item ke dalam shopping cart. Awalnya, kode memeriksa apakah `$_SESSION['cart']` sudah ada atau belum. Jika belum, maka inisialisasi `$_SESSION['cart']` dengan sebuah array kosong.

Shopping cart diimplementasikan sebagai array asosiatif di `$_SESSION['cart']`, di mana kunci (key) dari array adalah ISBN buku, dan nilai (value) dari array adalah jumlah (quantity) dari buku tersebut dalam shopping cart.

Selanjutnya, kode memeriksa apakah buku dengan ISBN yang dipilih sudah ada di dalam shopping cart (`if (isset($_SESSION['cart'][$id]))`). Jika sudah ada, maka jumlahnya akan ditambahkan satu, dan jika belum ada, maka akan ditambahkan entri baru dengan jumlah satu.

Setelah menangani proses penambahan item ke dalam shopping cart, kode melanjutkan dengan menampilkan isi dari shopping cart dalam bentuk HTML. Terdapat sebuah formulir dengan metode POST yang memungkinkan pengguna untuk melakukan aksi tertentu terhadap item dalam shopping cart.

Shopping cart ditampilkan dalam sebuah kartu (`<div>` dengan kelas "card") dengan judul "Shopping Cart". Di dalam tubuh kartu (`<div>` dengan kelas "card-body"), terdapat sebuah tabel (`<table>`) yang menampilkan informasi tentang buku-buku dalam shopping cart.

Dalam tabel, setiap baris mewakili satu buku dalam shopping cart. Informasi buku termasuk ISBN, Author, Title, Price, Qty (jumlah), dan Price * Qty (harga kali jumlah). Seluruh item dari shopping cart ditampilkan dengan menggunakan perulangan `foreach` untuk setiap elemen di `$_SESSION['cart']`.

Selama perulangan, kode juga menghitung total jumlah item (`$sum_qty`) dan total harga (`$sum_price`) dalam shopping cart.

Di akhir tabel, ada satu baris tambahan yang menampilkan total keseluruhan jumlah item dan harga dari shopping cart.

Pada akhirnya, terdapat pautan untuk melanjutkan belanja (`<a>` dengan kelas "btn btn-primary") dan mengosongkan shopping cart (`<a>` dengan kelas "btn btn-danger").

3.3.3 delete_cart.php

```
<?php
//File: delete_cart.php
//Deskripsi: untuk menghapus session
```

```
session_start();  
if(isset($_SESSION['cart'])) {  
    //Fungsi unset() dapat digunakan untuk membatalkan setel  
variabel  
    unset($_SESSION['cart']);  
}  
header('Location: view_books.php');  
?>
```

Penjelasan :

File delete_cart.php merupakan sebuah skrip PHP yang bertujuan untuk menghapus session yang disimpan dalam variabel `$_SESSION['cart']`.

Pada awalnya, dilakukan inisialisasi sesi dengan memanggil `session_start()`, yang memungkinkan untuk memanipulasi variabel sesi dalam skrip ini.

Kemudian, dilakukan pengecekan apakah variabel sesi `$_SESSION['cart']` sudah di-set atau belum dengan menggunakan `isset()`. Jika variabel sesi tersebut sudah di-set, maka fungsi `unset($_SESSION['cart'])` akan dijalankan. Fungsi `unset()` berfungsi untuk menghapus variabel yang diberikan, dalam hal ini, variabel sesi 'cart' dihapus sehingga data keranjang belanja dihapus dari sesi.

Setelah itu, pengguna akan diarahkan kembali ke halaman `view_books.php` menggunakan perintah `header('Location: view_books.php')`

Hasil Screenshot

localhost/tugas_praktikum_4/view_cust...

Customer Data

+ Add Customer Data

No	Name	Address	City	Action
1	Michelle Arthur	357 North Road A	Yarraville	<div>EditDelete</div>
2	Anne	Boulevard ' Street	Box Hill	<div>EditDelete</div>
3	Melly	Vicoria Street	Airport West	<div>EditDelete</div>
4	Rose	Queen Street	Airport West	<div>EditDelete</div>
5	Alan Wonga	Buanglow Street	Airport West	<div>EditDelete</div>
6	Dionysius Mario	Jl. Nangka IV No. 39, RT 04/RW 01, Cengkareng Barat	Box Hill	<div>EditDelete</div>

Total Rows = 6

Logout

localhost/tugas_praktikum_4/edit_cust...

Edit Customers Data

Nama:

Alvin

Address:

Mentari

City:

--Select a city--

Submit

Cancel

City:

--Select a city--

--Select a city--

Airport West

Box Hill

Yarraville

Confirm Delete Customers Data

Name	:	Michelle Arthur
Address	:	357 North Road A
City	:	Yarraville

Are you sure want to delete this item?

Login Form

Email

Password

Combination of username and password are not correct.

Login Form

Email

← → ↻ ⓘ localhost/tugas_praktikum_4/view_cust... 🔍 📄 ⌕ ☆ 🗨️ ⌚ ⚙️ 🖨️ 🌐 ⋮

Customers Data 2

+ Add Customer Data

No	Name	Address	City	Action
1	Anne	Boulevard ' Street	Box Hill	<a>Edit <a>Delete
2	Melly	Vicoria Street	Airport West	<a>Edit <a>Delete
3	Rose	Queen Street	Airport West	<a>Edit <a>Delete
4	Alan Wonga	Buanglow Street	Airport West	<a>Edit <a>Delete
5	Dionysius Mario	Jl. Nangka IV No. 39, RT 04/RW 01, Cengkareng Barat	Box Hill	<a>Edit <a>Delete

Total Rows = 5

← → ↻ ⓘ localhost/tugas_praktikum_4/view_boo... 🔍 📄 ⌕ ☆ 🗨️ ⌚ ⚙️ 🖨️ 🌐 ⋮

Books Data

ISBN	Author	Title	Price	Action
0-672-31281-1	Clarita Michelle	Jakarta dan Debu Adalah Kawan	\$66.99	<a>Add to Cart
0-672-31509-2	Pruitt, et al.	Teach Yourself GIMP in 24 Hours	\$24.99	<a>Add to Cart
0-672-31697-8	Michael Morgan	Java 2 for Professional Developers	\$34.99	<a>Add to Cart
0-672-31745-1	Thomas Down	Installing Debian GNU/Linux	\$24.99	<a>Add to Cart
0-672-31769-9	Thomas Schenk	Caldera OpenLinux System Administration Unleashed	\$49.99	<a>Add to Cart

Total Rows = 5

← → ↻ ⓘ localhost/tugas_praktikum_4/show_car... 🔍 📄 ⌕ ☆ 🗨️ ⌚ ⚙️ 🖨️ 🌐 ⋮

Shopping Cart

ISBN	Author	Title	Price	Qty	Price * Qty
0-672-31281-1	Clarita Michelle	Jakarta dan Debu Adalah Kawan	66.99	1	66.99
				1	66.99

Total items = 1

Continue Shopping Empty Cart

BAB IV

PENUTUP

4.1 Kesimpulan

Dalam pengembangan aplikasi ini, berbagai konsep dasar pemrograman web telah diimplementasikan dengan baik. Koneksi ke basis data MySQL berhasil dilakukan, dan data berhasil ditampilkan dengan menggunakan bahasa pemrograman PHP. Pembuatan halaman untuk melakukan operasi CRUD (Create, Read, Update, Delete) juga berhasil diimplementasikan dengan baik. Selain itu, mekanisme login dan logout telah diintegrasikan dengan baik untuk memastikan akses terbatas hanya kepada pengguna yang telah login.

Penggunaan session untuk manajemen state di sisi klien juga telah diterapkan dengan efektif. Hal ini memungkinkan informasi keranjang belanja tetap tersimpan selama pengguna berinteraksi dengan aplikasi.

4.2 Saran

Kita disarankan untuk terus memperdalam pemahaman terhadap konsep dasar database, seperti struktur tabel, relasi antar tabel, dan bahasa SQL. Hal ini akan membantu dalam mengembangkan aplikasi web yang membutuhkan interaksi dengan basis data.

DAFTAR PUSTAKA

Modul Praktikum 4 – Koneksi Basis Data

Connolly, T., & Begg, C. (2014). *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson.

Ullman, J. D., & Widom, J. (2013). *A First Course in Database Systems*. Pearson.

Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development*. Addison-Wesley.

Freeman, E., & Robson, E. (2018). *Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages*. O'Reilly Media.