

MODUL VIII

JSP & MySQL

A. Tujuan

Pada modul JSP & MySQL, Anda diharapkan untuk mampu:

1. Membuat koneksi ke database dengan JSP dan MySQL.
2. Mengirim query ke MySQL dengan JSP.
3. Mengelola data menggunakan query dari JSP.

B. Langkah Kerja

Buat sebuah file JSP bernama `readUsers.jsp`. File ini akan digunakan untuk menampilkan seluruh baris data di table `users`.

```
readUsers.jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Read Users</title>
  </head>
  <body>

  </body>
</html>
```

Di dalam elemen `body`, lakukan import terhadap seluruh class di package `java.sql` dan deklarasikan variable untuk menyimpan informasi koneksi ke basis data serta variable untuk menyimpan data yang diambil.

```
readUsers.jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Read Users</title>
  </head>
  <body>
    <%@page import = "java.sql.*" %>

    <%
      Connection connection = null;
      Statement statement = null;
      ResultSet resultSet = null;
    %>
  </body>
</html>
```

Kemudian hubungkan JSP dengan MySQL menggunakan class `DriverManager` dan method `getConnection()`. Berikan informasi URL dari basis data, username, dan password untuk mengakses basis data.

```
readUsers.jsp
```

```
...  
<body>  
<%@page import = "java.sql.*" %>  
  
<%  
    Connection connection = null;  
    Statement statement = null;  
    ResultSet resultSet = null;  
  
    try {  
        String connectionURL = "jdbc:mysql://localhost/company";  
        String userName = "root";  
        String passWord = "";  
        Class.forName("com.mysql.jdbc.Driver");  
        connection = DriverManager.getConnection(connectionURL, userName,  
passWord);  
        statement = connection.createStatement();  
    } catch (Exception e) {  
    }  
    %>  
</body>  
</html>
```

Setelah itu, deklarasikan sebuah query untuk mengakses seluruh baris data dari table users. Kemudian gunakan method `executeQuery()` untuk mengirimkan query ini ke MySQL, serta perulangan `while` untuk mengakses setiap baris data satu per satu dengan method `next()`. Tentukan data yang diakses berdasarkan kolom menggunakan method `getString()` kemudian tampilkan dengan method `println()`.

```
readUsers.jsp
```

```

<body>
<%@page import = "java.sql.*" %>

<%
...
try {
    ...
    String query = "SELECT * FROM users";
    resultSet = statement.executeQuery(query);

    while(resultSet.next()) {
%>

<% out.println(resultSet.getString("username")); %>
<% out.println(resultSet.getString("password")); %>

<%
    }
    resultSet.close();
    statement.close();
    connection.close();
} catch(Exception e) {
    out.println(e.getMessage());
}
%>
</body>
</html>

```

Selanjutnya buatlah sebuah file JSP bernama createUser.jsp. File ini akan digunakan untuk membuat user baru ke basis data. Tampilkan sebuah form untuk user menginputkan username dan password.

```

createUser.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Create User</title>
</head>
<body>
    <h1>Create User</h1>
    <form action="" method="post">
        <label>Username
            <input type="text" name="userName">
        </label>
        <label>Password
            <input type="password" name="passWord">
        </label>
        <input type="submit" name="createUser">
    </form>
</body>
</html>

```

Deklarasikan scriptlet sebelum tag penutup elemen html untuk melakukan import terhadap seluruh class di package java.sql. Kemudian deklarasikan variable untuk menerima input user melalui method getParameter().

```
createUser.jsp
```

```
...  
<body>  
...  
</body>  
<%@page import = "java.sql.*" %>  
<%@page import = "javax.swing.*" %>  
  
<%  
String userName = request.getParameter("userName");  
String passWord = request.getParameter("passWord");  
%>  
</html>
```

Deklarasikan variable untuk menyimpan URL dari basis data, status koneksi ke basis data, dan perintah SQL.

createUser.jsp

```
...  
<%  
...  
String connectionURL = "jdbc:mysql://localhost/company";  
Connection connection = null;  
PreparedStatement statement = null;  
Class.forName("com.mysql.jdbc.Driver").newInstance();  
  
int updatedQuery = 0;  
%>  
</html>
```

Buat koneksi ke basis data dengan method getConnection(). Siapkan query untuk menambahkan data baru dengan method prepareStatement(). Sisipkan data baru ke dalam query dengan method executeUpdate().

createUser.jsp

```

...
<%
...
if(userName != null && passWord != null) {
    try {
        connection = DriverManager.getConnection(connectionURL, "root", "");
        String query = "INSERT INTO users VALUES (?, ?, ?)";
        statement = connection.prepareStatement(query);

        int randomNum = (int) Math.random() * (99 - 1 + 1) + 1;

        statement.setInt(1, 99);
        statement.setString(2, userName);
        statement.setString(3, passWord);
        updatedQuery = statement.executeUpdate();

        if(updatedQuery != 0) {
            response.sendRedirect("readUsers.jsp");
        }
    } catch (Exception e) {
        response.sendRedirect("readUsers.jsp");
    } finally {
        statement.close();
        connection.close();
    }
}
%>
</html>
    
```

Kembali ke getUsers.jsp. Buat sebuah link untuk menghapus user. Link akan digunakan untuk mengirimkan id dari user yang akan dihapus.

readUsers.jsp

```

...
<body>
...
<%
...
<% out.println(resultSet.getString("username")); %>
<% out.println(resultSet.getString("password")); %>
<a
href="deleteUser.jsp?id=<%out.println(resultSet.getString("id"));%>">
Delete
</a>
...
%>
</body>
</html>
    
```

Kemudian buatlah sebuah file JSP bernama deleteUser.jsp. File ini akan menghapus salah satu data dari table users berdasarkan id user yang dikirimkan dari readUsers.jsp. Import package java.sql lalu deklarasikan variable id untuk menerima id user dari metode HTTP GET yang dikirimkan oleh readUsers.jsp. Deklarasikan variable dari class Connection dan Statement untuk menyimpan informasi koneksi ke basis data dan memproses query untuk menghapus data. Gunakan method executeUpdate() untuk mengirimkan query.

deleteUser.jsp

```

<%@page import = "java.sql.*" %>
<%
String id = request.getParameter("id");
try {
    Class.forName("com.mysql.jdbc.Driver");
    String connectionURL = "jdbc:mysql://localhost/company";

    Connection connection = DriverManager.getConnection(connectionURL,
"root", "");
    Statement statement = connection.createStatement();

    int i = statement.executeUpdate("DELETE FROM users WHERE id=" + id +
    "");
    response.sendRedirect("readUsers.jsp");
} catch (Exception e) {
    response.sendRedirect("readUsers.jsp");
}
%>
    
```

Kembali ke `readUsers.jsp`, sisipkan sebuah elemen hyperlink untuk mengubah data user. Hyperlink akan mengirimkan id dari user yang akan diubah.

```

readUsers.jsp
...
<body>
...
<%
...
<% out.println(resultSet.getString("username")); %>
<% out.println(resultSet.getString("password")); %>
<a
href="updateUser.jsp?id=<%out.println(resultSet.getString("id"));%>">
    Update
</a>
<a
href="deleteUser.jsp?id=<%out.println(resultSet.getString("id"));%>">
    Delete
</a>
...
%>
</body>
</html>
    
```

Setelah itu, buatlah file JSP bernama `updateUser.jsp` yang akan menampilkan data user yang akan diubah. Sisipkan sebuah form untuk user menginputkan data yang baru.

```
updateUser.jsp
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Update User</title>
</head>
<body>
  <form action="processUpdateUser.jsp">
    <label>Username
      <input type="text" name="userName">
    </label>
    <label>Password
      <input type="password" name="passWord" >
    </label>
    <input type="submit" name="updateUser">
  </form>
</body>
</html>
```

File updateUser akan menerima id dari readUsers.jsp dan menampilkan data user ke textfield untuk diubah. Attribute value di input control akan menampilkan data user menggunakan method getString().

```
updateUser.jsp
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>...</head>
<body>
  <%@page import = "java.sql.*" %>
  <h1>Update User</h1>
  <%
    String id = request.getParameter("id");
    Connection connection = null;
    Statement statement = null;
    ResultSet resultSet = null;

    try {
      String connectionURL = "jdbc:mysql://localhost/company";
      String userName = "root";
      String passWord = "";
      Class.forName("com.mysql.jdbc.Driver");
      connection = DriverManager.getConnection(connectionURL, userName,
passWord);
      statement = connection.createStatement();
      String query = "SELECT * FROM users WHERE id=" + id + "";
      resultSet = statement.executeQuery(query);

      while(resultSet.next()) {

    %>

      <form>
        <label>Username
        <input type="text" name="userName"
value="<%out.println(resultSet.getString("userName"));%>">
        </label>
        <label>Password
        <input type="password" name="passWord"
value="<%out.println(resultSet.getString("passWord"));%>">
        </label>
        <input type="submit" name="updateUser">
      </form>

      <%
    }
  } catch (Exception e) {
    out.println(e.getMessage());
  }
  %>
</body>
</html>
```

Kemudian atur agar form akan mengirimkan data ke processUpdateUser.jsp menggunakan attribute action. Sisipkan sebuah input control tersembunyi untuk mengirimkan id dari user yang diubah.

```
updateUser.jsp
```



```
...
    <form action="processUpdateUser.jsp">
        <input type="hidden" name="id"
value="<%out.println(resultSet.getString("id"));%>">

        <label>Username
            <input type="text" name="userName"
value="<%out.println(resultSet.getString("userName"));%>">
        </label>
        <label>Password
            <input type="password" name="passWord"
value="<%out.println(resultSet.getString("passWord"));%>">
        </label>
        <input type="submit" name="updateUser">
    </form>
...
</body>
</html>
```

Buatlah sebuah file `processUpdateUser.jsp` yang akan mengirimkan query untuk mengubah data user dari basis data.

```
processUpdateUser.jsp

<%@page import = "java.sql.*" %>

<%
String id = request.getParameter("id");
String userName = request.getParameter("userName");
String passWord = request.getParameter("passWord");

if(id != null) {

    String connectionURL = "jdbc:mysql://localhost/company";
    Connection connection = null;
    PreparedStatement statement = null;
    Class.forName("com.mysql.jdbc.Driver");

    try {
        Class.forName("com.mysql.jdbc.Driver");
        connection = DriverManager.getConnection(connectionURL, "root",
        "");
        String query = "UPDATE users SET id=?, username=?, password=? WHERE
id=" + id + "";
        statement = connection.prepareStatement(query);
        statement.setString(1, id);
        statement.setString(2, userName);
        statement.setString(3, passWord);
        int i = statement.executeUpdate();
        response.sendRedirect("readUsers.jsp");
    } catch (Exception e) {
        response.sendRedirect("readUsers.jsp");
    }
}
%>
```