

1. Tujuan

- Membuat program menggunakan Include dan Forward
- Mengetahui pengertian dari Filter dan penggunaannya

2. Latar Belakang

Sebuah filter adalah sebuah object yang melaksanakan tugas penfilteran pada saat me-request resource (servlet atau static content), atau juga saat me-respon dari resource, atau juga kedua-duanya.

Filter melaksanakan penfilteran dalam fungsi **doFilter**. Setiap filter memiliki akses ke objek *FilterConfig* dari mana ini dapat memperoleh parameter-parameter inisialisasi, sebuah referensi ke *ServletContext* yang mana bisa digunakan, sebagai contoh, untuk me-load sumber daya dibutuhkan tugas-tugas pen-filteran.

Filter dikonfigurasi dalam *deployment descriptor* aplikasi web



Contoh-contoh yang sudah diidentifikasi untuk desain ini antara lain:

- 1) Authentication Filters
- 2) Logging and Auditing Filters
- 3) Image conversion Filters
- 4) Data compression Filters
- 5) Encryption Filters
- 6) Tokenizing Filters
- 7) Filters that trigger resource access events
- 8) XSL/T filters
- 9) Mime-type chain Filter

Keuntungan besar dari filter adalah kemungkinan untuk memproses http request sebelum mencapai aplikasi utama, sehingga kita dapat meninggalkan Hello World kita atau aplikasi apapun dan menambahkan layer lainnya mengelilinginya, dalam hal ini, sebuah auth layer.


3. Percobaan

Percobaan 1 – Penanganan Response Redirection:

 Info	Tujuan section ini adalah menunjukkan bagaimana cara menangani response redirection. Panganannya meliputi menyertakan file lain ke dalam halaman web dan mengalihkan response ke halaman lain.
 Info	Skenario: Form login berisi username dan password. Ketika tombol submit ditekan akan dilakukan validasi proses login. Jika login gagal akan ditampilkan form login kembali. Jika login berhasil akan alihkan ke halaman success.

Langkah-langkahnya:

- Buat project web baru dengan nama **advancedServlet**.

 Tip	File → New Project, pilih category Web → Web Application, klik Next Ketikkan nama project, klik Finish
--------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------

- Pada Source Packages, buat package dengan nama **jeni3.servlet**.
- Pada package jeni3.servlet, buat class Servlet dengan nama **LoginFormServlet**.
Berikut kodenya:

```
package jeni3.servlet;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
 * @author mee_andto@yahoo.com
 * @version 1.1
 */
public class LoginFormServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
```

```

        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Login Form Page</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Login Page</h1>");
        out.println("<form action='login' method='POST'>");
        out.println("<table border='0'>");
        out.println("<tr>");
        out.println("<td>Username</td>");
        out.println("<td>: <input type='text' name='username' value='' /></td>");
        out.println("</tr>");
        out.println("<tr>");
        out.println("<td>Password</td>");
        out.println("<td>: <input type='password' name='password' value='' /></td>");
        out.println("</tr>");
        out.println("<tr>");
        out.println("<td><input type='reset' name='reset' value='Reset' /></td>");
        out.println("<td><input type='submit' name='login' value='Login' /></td>");
        out.println("</tr>");
        out.println("</table>");
        out.println("</form>");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }

    /** Handles the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
}

```



Action pada form di atas, biasanya merupakan url-pattern pada mapping servlet. Pastikan saat mapping servlet menggunakan nama yang sama dengan action form.

- Pastikan servlet mapping-nya seperti berikut:

```
<servlet>
  <servlet-name>LoginFormServlet</servlet-name>
  <servlet-class>jeni3.servlet.LoginFormServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>LoginFormServlet</servlet-name>
  <url-pattern>/loginForm</url-pattern>
</servlet-mapping>
```

- Untuk menangani action form, pada package **jeni3.servlet**, buat class Servlet dengan nama **LoginServlet**. Berikut kode-nya:

```
package jeni3.servlet;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
 * @author mee_andto@yahoo.com
 * @version 1.1
 */
public class LoginServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //initializing...
        RequestDispatcher rd = null;

        String userName = request.getParameter("username");
        String password = request.getParameter("password");
        if (authenticate(userName, password)){
            //success handler
            rd = request.getRequestDispatcher("/success");
            rd.forward(request, response);
        }else{
            //failed handler
            response.setContentType("text/html;charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("User does not exist with given username and/or password");

            //return to login page
            rd = request.getRequestDispatcher("/loginForm");
            rd.include(request, response);
            out.close();
        }
    }

    /**login handler
     * @param username String
     * @param password String
     * @return boolean
     */
}
```

```
public boolean authenticate(String username, String password){  
    //use hardcode username and password. (jeni and indonesia)  
    return (username.equals("jeni") && password.equals("indonesia"));  
}
```

**Perhatian**

Pengambilan URL oleh RequestDispatcher terhadap resource file dituliskan lengkap dengan ekstensinya (contoh: /login.html), sementara untuk resource URL servlet dituliskan url-pattern-nya (contoh: /success).

- Pastikan servlet mapping-nya seperti berikut:

```
<servlet>  
    <servlet-name>LoginServlet</servlet-name>  
    <servlet-class>jeni3.servlet.LoginServlet</servlet-class>  
</servlet>  
<servlet-mapping>  
    <servlet-name>LoginServlet</servlet-name>  
    <url-pattern>/login</url-pattern>  
</servlet-mapping>
```

- Untuk menangani login success, pada package **jeni3.servlet** buat class Servlet lagi dengan nama **SuccessServlet**. Berikut kodenya:

```
package jeni3.servlet;  
  
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
/**  
 * @author mee_andto@yahoo.com  
 * @version 1.1  
 */  
public class SuccessServlet extends HttpServlet {  
  
    protected void processRequest(HttpServletRequest request,  
    HttpServletResponse response)  
        throws ServletException, IOException {  
        RequestDispatcher rd = null;  
  
        response.setContentType("text/html; charset=UTF-8");  
  
        rd = request.getRequestDispatcher("temp/header.html");  
        rd.include(request, response);  
    }  
}
```

```
rd = request.getRequestDispatcher("temp/mainContent.html");
rd.include(request, response);

rd = request.getRequestDispatcher("temp/footer.html");
rd.include(request, response);
}

/** Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 */
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}
}
```

- Pastikan servlet-mapping untuk class SuccessServlet seperti berikut:

```
<servlet>
    <servlet-name>SuccessServlet</servlet-name>
    <servlet-class>jeni3.servlet.SuccessServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>SuccessServlet</servlet-name>
    <url-pattern>/success</url-pattern>
</servlet-mapping>
```

- Pada Web pages, buat folder **temp**.
- Untuk melengkapi class SuccessServlet, dalam folder **temp** buat file HTML dengan nama header.html, mainContent.html dan footer.html dengan kode seperti berikut:

header.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
    <head>
        <title>Login Success</title>
    </head>
    <body>
```

mainContent.html

```
<h1>Congratulation!! You are logged in.</h1>
```

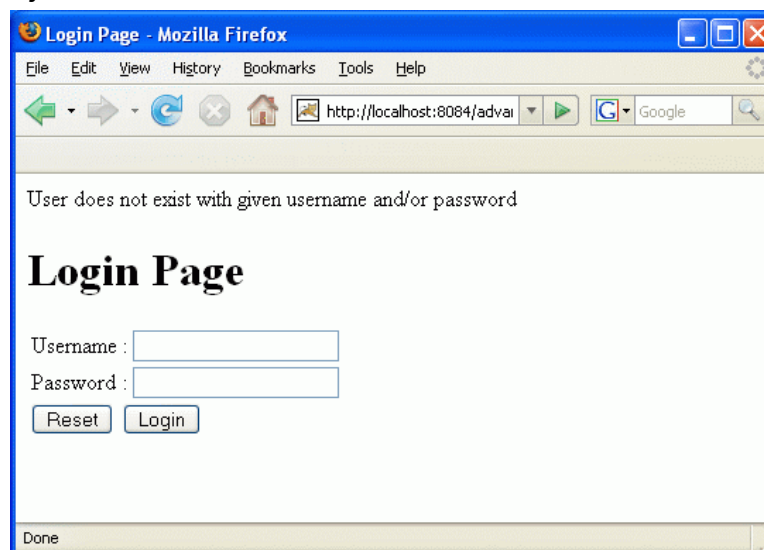
footer.html

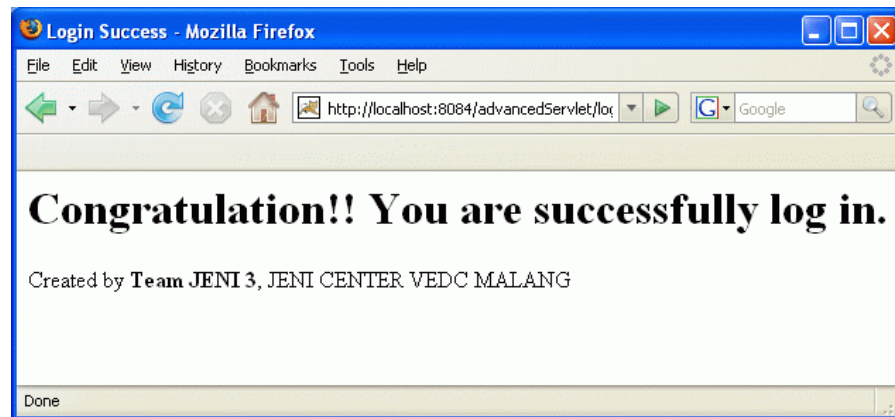
```
Created by <b>Team JENI 3</b>, JENI CENTER VEDC MALANG  
</body>  
</html>
```

- Tambahkan link pada file index.jsp untuk mengakses login.html:


```
<br/><a href="./loginForm">Login Page</a>
```


- Run → Run Project





Percobaan 2 – Session Tracking dan Manajemen :

 Info	Tujuan section ini adalah menunjukkan bagaimana cara menyimpan session user dan mengakses informasi session itu pada halaman lain.
-------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------

 Info	Section ini akan menggunakan hasil latihan pada section sebelumnya (Percobaan 1).
-------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------

Pada section sebelumnya class `LoginFormServlet`, `LoginServlet` dan `SuccessServlet` belum memiliki validasi untuk mengecek apakah user sudah login atau belum, karena memang informasi user belum disimpan dalam object session. Sehingga ketika url `loginForm` diakses masih memunculkan form login padahal user sudah login. Latihan berikutnya informasi user akan kita simpan dalam object session kemudian ditampilkan informasi itu di halaman lain.

 Info	Kode-kode yang tercetak tebal adalah perubahan/ tambahan.
---------------------------------------------------------------------------------------------	------------------------------------------------------------------

Langkah-langkahnya:

- Buka class `LoginFormServlet`, lakukan modifikasi untuk menangani form ditampilkan untuk user yang belum login, sehingga kodenya seperti berikut:

```
package jeni3.servlet;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
 * @author mee_andto@yahoo.com
 * @version 1.1
 */
public class LoginFormServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
    }
}
```

```
RequestDispatcher rd = null;
HttpSession session = request.getSession();
if (session.getAttribute("user") != null){
    rd = request.getRequestDispatcher("/success");
    rd.forward(request, response);
}

out.println("<html>");
out.println("<head>");
out.println("<title>Login Form Page</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Login Page</h1>");
out.println("<form action='login' method='POST'>");
out.println("<table border='0'>");
out.println("<tr>");
out.println("<td>Username</td>");
out.println("<td>: <input type='text' name='username' value='' /></td>");
out.println("</tr>");
out.println("<tr>");
out.println("<td>Password</td>");
out.println("<td>: <input type='password' name='password' value='' /></td>");
out.println("</tr>");
out.println("<tr>");
out.println("<td><input type='reset' name='reset' value='Reset' /></td>");
out.println("<td><input type='submit' name='login' value='Login' /></td>");
out.println("</tr>");
out.println("</table>");
out.println("</form>");
out.println("</body>");
out.println("</html>");
out.close();
}

/** Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

/** Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}
}
```

- Buka class LoginServlet, lakukan modifikasi untuk login success sehingga kodenya menjadi seperti berikut:

```
package jeni3.servlet;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
 * @author mee_andto@yahoo.com
 * @version 1.1
 */
public class LoginServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        //initializing...
        RequestDispatcher rd = null;
        HttpSession session = request.getSession();
        if (session.getAttribute("user") != null){
            rd = request.getRequestDispatcher("/success");
            rd.forward(request, response);
        }
        String userName = request.getParameter("username");
        String password = request.getParameter("password");
        if (authenticate(userName, password)){
            //success handler
            //session handler
            session.setAttribute("user",userName);
            rd = request.getRequestDispatcher("/success");
            rd.forward(request, response);
        }else{
            //failed handler
            response.setContentType("text/html;charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("User does not exist with given username and/or
password");

            //return to login page
            rd = request.getRequestDispatcher("/login.html");
            rd.include(request, response);
            out.close();
        }
    }
}
```

```
    }  
}  
/**login handler  
 * @param username String  
 * @param password String  
 * @return boolean  
 */  
public boolean authenticate(String username, String password){  
    //use hardcode username and password. (jeni and indonesia)  
    return (username.equals("jeni") && password.equals("indonesia"));  
}  
}
```

- Kemudian modifikasi juga class SuccessServlet untuk menampilkan informasi user, sehingga kodenya menjadi seperti berikut:

```
package jeni3.servlet;  
  
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
/**  
 * @author mee_andto@yahoo.com  
 * @version 1.1  
 */  
public class SuccessServlet extends HttpServlet {  
  
    protected void processRequest(HttpServletRequest request,  
        HttpServletResponse response)  
        throws ServletException, IOException {  
        HttpSession session = request.getSession();  
        RequestDispatcher rd = null;  
        response.setContentType("text/html;charset=UTF-8");  
        PrintWriter out = response.getWriter();  
  
        if (session.getAttribute("user") == null){  
            out.println("<font color='red'>You are not log in. Please login  
here</font>");  
            rd = request.getRequestDispatcher("/login.html");  
            rd.include(request, response);  
        }else{  
            rd = request.getRequestDispatcher("temp/header.html");  
            rd.include(request, response);  
  
            rd = request.getRequestDispatcher("temp/mainContent.html");  
            rd.include(request, response);  
        }  
    }  
}
```

```
        out.println("<h2>Hello <font  
color='red'>" + session.getAttribute("user") + "</font> <a  
href='./logout'>Logout</a></h2>");  
        rd = request.getRequestDispatcher("temp/footer.html");  
        rd.include(request, response);  
    }  
    out.close();  
}  
  
/** Handles the HTTP <code>GET</code> method.  
 * @param request servlet request  
 * @param response servlet response  
 */  
protected void doGet(HttpServletRequest request, HttpServletResponse  
response)  
    throws ServletException, IOException {  
    processRequest(request, response);  
}  
protected void doPost(HttpServletRequest request, HttpServletResponse  
response)  
    throws ServletException, IOException {  
    processRequest(request, response);  
}  
}
```

- Kemudian buat class Servlet baru dalam package jeni3.servlet dengan nama **LogoutServlet**.

```
package jeni3.servlet;  
  
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
  
/**  
 * @author mee_andto@yahoo.com  
 * @version 1.1  
 */  
public class LogoutServlet extends HttpServlet {  
  
    protected void processRequest(HttpServletRequest request,  
HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html; charset=UTF-8");  
        PrintWriter out = response.getWriter();  
        RequestDispatcher rd = null;  
        HttpSession session = request.getSession();  
        if (session.getAttribute("user") != null) {
```

```
        session.removeAttribute("user");
        session.invalidate();
        out.println("You have logged out successfully");
    }
    rd = request.getRequestDispatcher("/loginForm");
    rd.forward(request, response);
    out.close();
}

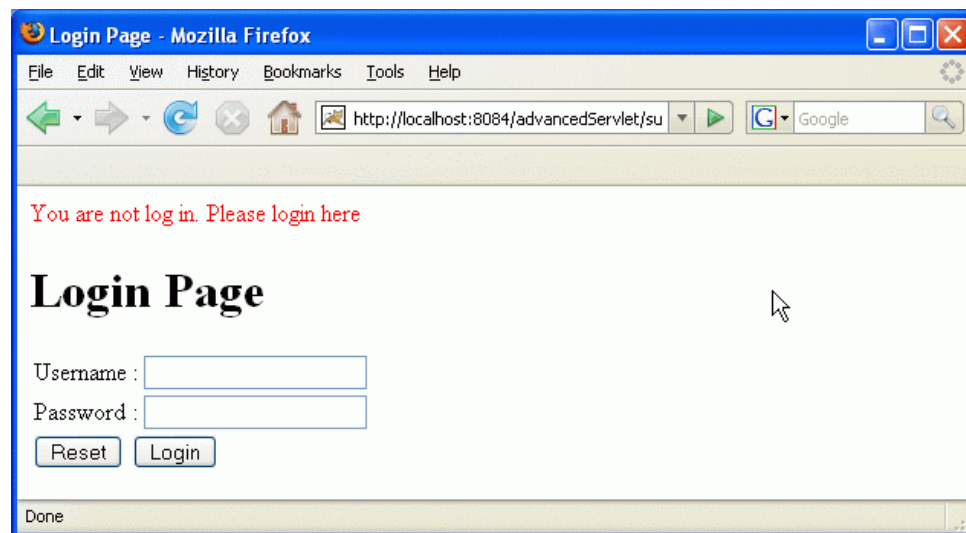
/** Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 */
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/** Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 */
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}
}
```

- Pastikan servlet mapping-nya seperti berikut:

```
<servlet>
  <servlet-name>LogoutServlet</servlet-name>
  <servlet-class>jeni3.servlet.LogoutServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>LogoutServlet</servlet-name>
  <url-pattern>/logout</url-pattern>
</servlet-mapping>
```

- Run → Run Project

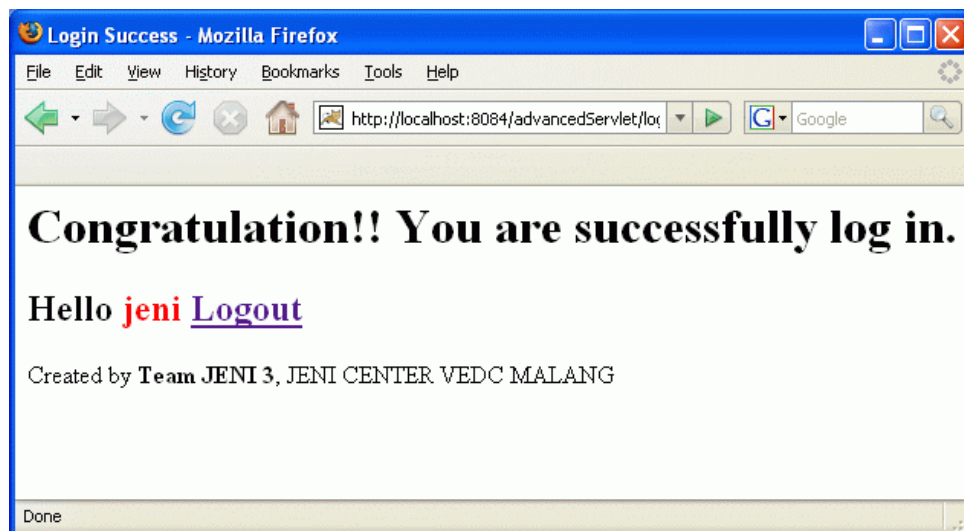




Info

Jadi,

1. Untuk menyimpan session digunakan sintak `session.setAttribute(String, Object)`
2. Untuk mengambil data session digunakan sintak `session.getAttribute(String)`
3. Untuk menghilangkan session digunakan sintak `session.removeAttribute(String)` atau `session.invalidate()`



Percobaan 3 – Membuat Filter Authentikasi :



Tujuan section ini adalah menunjukkan bagaimana membuat Filter Authentikasi sekaligus mengetahui bagaimana Filter itu bekerja.

Sebuah filter adalah sebuah object yang melaksanakan tugas penfilteran pada saat me-request resource (servlet atau static content), atau juga saat me-respon dari resource, atau juga kedua-duanya.

Filter melaksanakan penfilteran dalam fungsi **doFilter**. Setiap filter memiliki akses ke objek *FilterConfig* dari mana ini dapat memperoleh parameter-parameter inisialisasi, sebuah referensi ke *ServletContext* yang mana bisa digunakan, sebagai contoh, untuk me-load sumber daya dibutuhkan tugas-tugas pen-filteran.

Filter dikonfigurasi dalam *deployment descriptor* aplikasi web

Contoh-contoh yang sudah diidentifikasi untuk desain ini antara lain:

- 1) Authentication Filters
- 2) Logging and Auditing Filters
- 3) Image conversion Filters
- 4) Data compression Filters
- 5) Encryption Filters
- 6) Tokenizing Filters
- 7) Filters that trigger resource access events
- 8) XSL/T filters
- 9) Mime-type chain Filter



Skenario:

Dari aplikasi login sebelumnya, akan kita modifikasi sedikit. Setelah user berhasil login, maka user akan mampu mengakses URL /admin/index.jsp misalnya. Jika belum login maka tidak diperbolehkan mengakses URL /admin. Proses ini akan ditangani oleh Filter Authentikasi.

Langkah-langkah pembuatan Filter Authentikasi:

- Buat package **jeni3.servlet.filter**
- Dalam package itu buat class Filter dengan nama **AuthenticationFilter** yang mengextends interface Filter. Berikut kode lengkapnya:

```
package jeni3.servlet.filter;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
 * @author mee_andto@yahoo.com
 */
public class AuthenticationFilter implements Filter{
    private FilterConfig config;

    public void init(FilterConfig filterConfig) throws ServletException {
        this.config = filterConfig;
    }

    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws IOException, ServletException {
        response.setContentType("text/html;enCoding=UTF-8");
        PrintWriter out = response.getWriter();

        HttpSession session =
        ((HttpServletRequest)request).getSession(false);
        if (session.getAttribute("user")!= null){
            chain.doFilter(request, response);
        }else{
            out.println("<font color='red'>You are not allowed to access this
page. Please <b>log in</b> <a href='../login.html'>here</a></a>");
            out.close();
        }
    }

    public void destroy() {
        this.config = null;
    }
}
```

- Kemudian definisikan Filter mappingnya dalam file **web.xml**, ke URL mana filter ini akan diterapkan. Pastikan meletakkan mapping filter **sebelum** mapping servlet. Berikut kode mapping-nya:

```
<filter>
  <filter-name>AuthenticationFilter</filter-name>
  <filter-class>jeni3.servlet.filter.AuthenticationFilter</filter-class>
</filter>
<filter-mapping>
```

```
<filter-name>AuthenticationFilter</filter-name>
<url-pattern>/admin/*</url-pattern>
</filter-mapping>
```

Sesuai mapping di atas bahwa Filter Authentication ini akan diterapkan ke semua URL /admin. Artinya bahwa user yang sudah login saja yang boleh mengakses halaman /admin/* itu.

- Selanjutnya pada **Web Pages**, buat folder dengan nama **admin**.
- Dalam folder **admin** itu buat file **index.jsp** dengan kode seperti berikut:

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Admin Page</title>
  </head>
  <body>

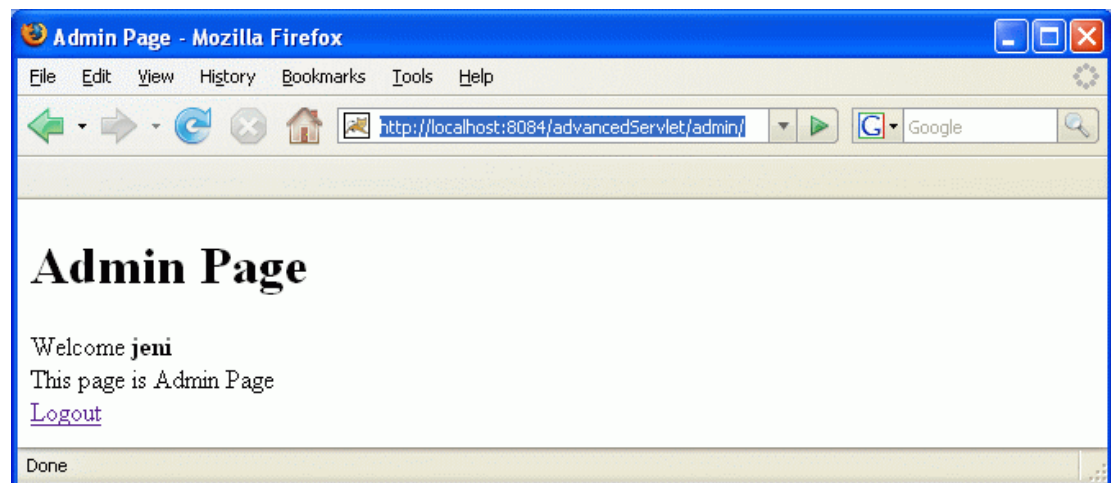
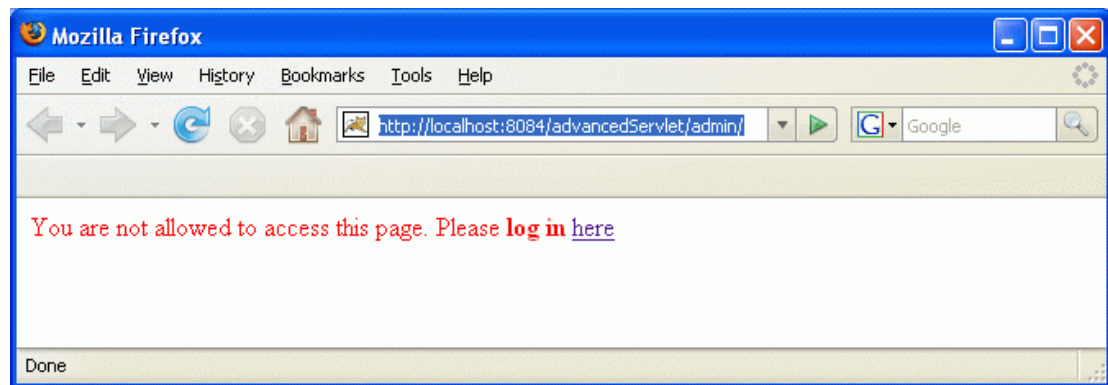
    <h1>Admin Page</h1>
    <%
      if (session.getAttribute("user") != null)
        out.println("Welcome
    <b>"+session.getAttribute("user")+"</b><br/>");
    %>
    This page is Admin Page
    <br/>
    <a href=" ../logout">Logout</a>
  </body>
</html>
```

- Untuk memudahkan akses ke /admin, tambahkan link dalam file **Web Pages/index.jsp**.

```
<br/>
<a href="/admin">Admin Page</a>
```

- Run – Run Project

Sebelum Login



Sesudah Login



> > > Java Education Network Indonesia