

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

## 1. Tujuan

- Membuat project menggunakan editor Netbeans
- Membuat aplikasi login
- Menggunakan tag HTML- link
- Menggunakan tag HTML- select
- Menggunakan bean tag library

## 2. Latar Belakang

### Pola MVC

Model-View-Controller adalah pola desain software yang membantu anda secara logik memisahkan kode anda, membuatnya lebih *reusable*, mudah dikelola (maintainable), dan membuat lebih baik pada umumnya. Model View Controller dijelaskan pertama kali oleh grup penulis Gang of Four. Dean Helman menulis (intisari dari white paper Objective Toolkit Pro):

"Paradigma MVC adalah suatu cara memecah aplikasi atau cukup beberapa bagian antar muka aplikasi ke dalam 3 bagian: model, view dan controller. MVC pada dasarnya dibangun untuk memetakan aturan tradisional yaitu input, proses dan output ke dalam GUI yang nyata."

|            |           |           |
|------------|-----------|-----------|
| Input      | -> Proses | -> Output |
| Controller | -> Model  | -> View   |

"Input user, pemodelan terhadap dunia luar/nyata, dan tampilan visual ke user dipisahkan dan dihandel oleh objek model, objek view dan objek controller. Controller mengipretesaikan input mouse dan keyboard dari user dan memetakan aksi user ke dalam perintah-perintah yang dikirim ke objek model dan/atau objek view untuk memberi efek terhadap perubahan yang sesuai. Model mengatur satu atau lebih elemen data, merespon queri-queri dan merespon instruksi-instruksi perubahan status. Bagian view mengatur area *rectangular* dari tampilan dan bertanggung jawab untuk menyajikan data ke user melalui kombinasi antara grafik dan text."

Setelah anda mengetahui konsep MVC, untuk lebih mendalami penerapannya pada proses development, kita akan membuat aplikasi mulai dari awal sekali. Latihan ini dibuat berkesinambungan antara satu latihan dengan latihan yang lain.

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

- **Download Struts**

Struts MVC Framework bisa didownload di <http://struts.apache.org/downloads.html>. Dalam distribusi Struts terdapat file struts-core.jar, dependencinya, contoh aplikasi dan dokumentasi dalam HTML format.

Saat ini Struts memiliki 2 versi Framework yaitu Struts1 dan Struts2. Struts1 sebagai Struts murni sampai saat ini mencapai versi 1.3.9. Sementara Struts2 sebagai versi terbaru Struts didalamnya menggunakan ActionController XWork dari Open Symphony. Saat ini masih versi 2.0.1.

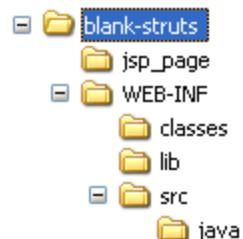
Untuk aplikasi yang dibahas pada latihan kali ini kita akan menggunakan Struts versi 1.2.9 built in pada IDE Netbeans.

- **Mempersiapkan Aplikasi**

Dalam distribusi struts sudah disertakan contoh aplikasi "blank.war". Anda bisa menggunakannya sebagai start awal anda memulai menggunakan struts. Cukup dengan mengekstraknya ke dalam direktori webapps Tomcat.

Aplikasi J2EE memiliki karakteristik dalam susunan foldernya, apalagi kalau sudah bekerja dengan framework. Di dalam document root aplikasi web anda harus terdapat folder WEB-INF, dimana didalamnya untuk menyimpan file-file konfigurasi, librari-librari lain yang dibutuhkan dan class-class java. Sementara resource dan file-file HTML, JSP atau template engine lainnya bisa anda atur sesuka anda.

Karena kita akan menggunakan framework struts, minimal kita hanya butuh file struts2-core.jar dalam daftar librari yang kita pakai, biasanya diletakkan di folder WEB-INF/lib. Namun, karena kita akan melakukan tugas-tugas lain yang tidak tersedia di file struts2-core.jar maka librari lain yang berhubungan bisa kita tambahkan ke dalam folder lib tersebut, misalnya commons-logging-1.1.jar untuk penanganan log, freemaker-2.3.8.jar untuk templating, dll.



## JENI 3 – 07 Modul Praktikum Pengenalan MVC

Sehingga minimal, aplikasi dengan menggunakan struts memiliki susunan folder:  
Berikut penjelasan susunan foldernya:

- o **jsp\_page**: untuk meletakkan file-file jsp. Bisa anda rename, menambah folder lagi, dll.
- o **WEB-INF**: harus ada di setiap aplikasi J2EE.
- o **classes**: tempat meletakkan class-class Java hasil kompilasi source code, dan juga file-file konfigurasi seperti struts.xml
- o **lib**: tempat meletakkan librari-librari tambahan dalam bentuk file jar
- o **src**: folder yang berisi semua file-file mentah aplikasi
- o **java**: folder tempat source code java diletakkan

### Mengenali pola MVC dalam Framework aplikasi berbasis web

Ada 3 hal yang selalu dilakukan dalam membuat aplikasi dengan struts (mvc):

- Membuat halaman JSP atau lainnya (HTML, Velocity, Framemaker,dll) untuk menampilkan pesan atau tampilan ke user (View)
- Membuat ActionForm class untuk menangani logika aplikasi dan permintaan client (Model). Komponen Model meliputi JavaBeans dan Scope, ActionForm Beans, System State Beans dan Business Logic Beans.
- Membuat pemetaan (mapping) untuk mengarahkan antara action dengan halaman view atau action lain (Controller). Komponen Controller meliputi ActionServlet, Request Processor, Action Class, Action Mapping.

### Penggunaan HTML Tag Library

Tag Library dalam struts bisa dibedakan ke dalam 3 macam jenis, yaitu HTML TagLib, Bean TagLib, Logic TagLib, Nested TagLib, Template TagLib, dan Tiles TagLib. Kali ini kita akan menggunakan HTML TagLib.

## 3. Percobaan

### Percobaan 1 : Mengenal Arsitektur aplikasi berbasis MVC



Info

Tujuan section ini adalah mengenalkan arsitektur aplikasi web berbasis MVC yang meliputi penggunaan layer-layer dalam pengaksesan class dan desain sistem dalam UML (use case dan class diagram).

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

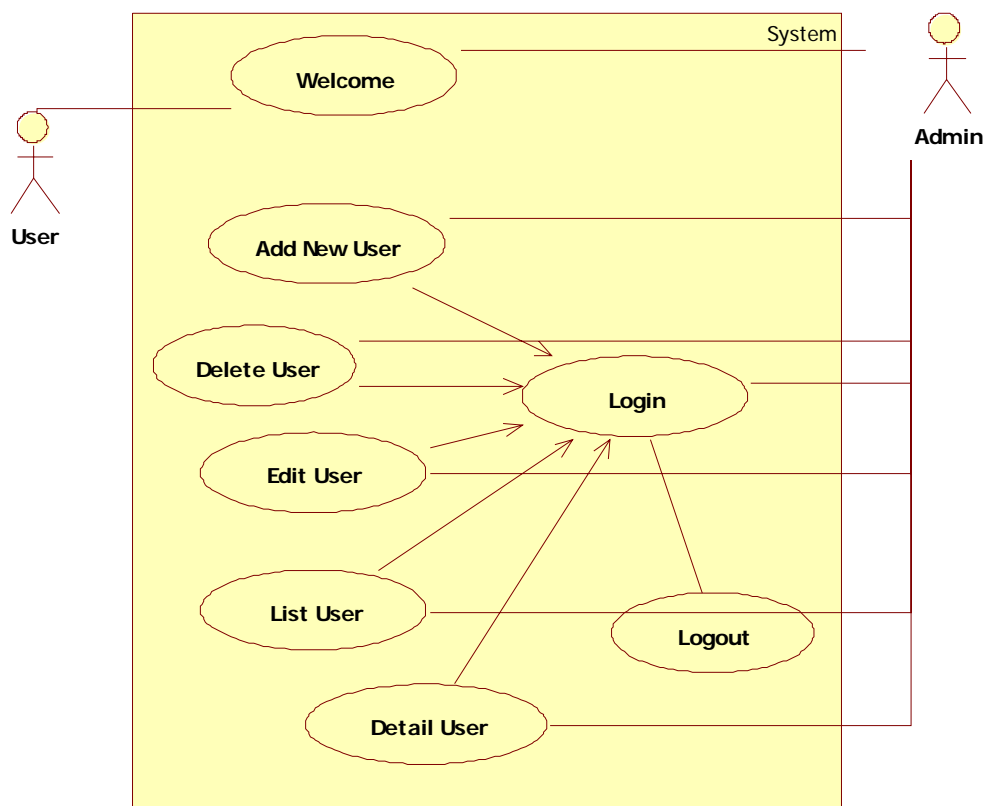


Info

Case: User Management

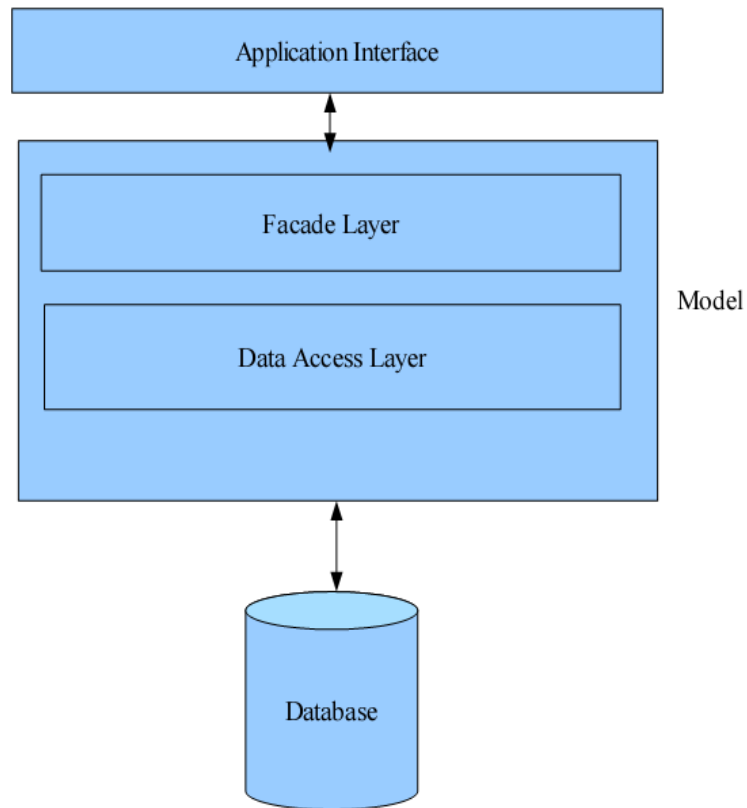
Aplikasi user management dilakukan oleh user dengan role Admin. Seorang admin dapat melihat/list seluruh user, melihat detail informasi masing-masing user, menambah user baru, mengedit user dan menghapus user.

Use Case Diagram aplikasi



Pembagian Layer Model pada aplikasi berbasis MVC:

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

**Application Interface**

Menyediakan class-class Java yang langsung diakses oleh object View aplikasi.

**Facade Layer**

Menyediakan class-class Java yang menangani pemrosesan bussiness logic ke obect DAO.

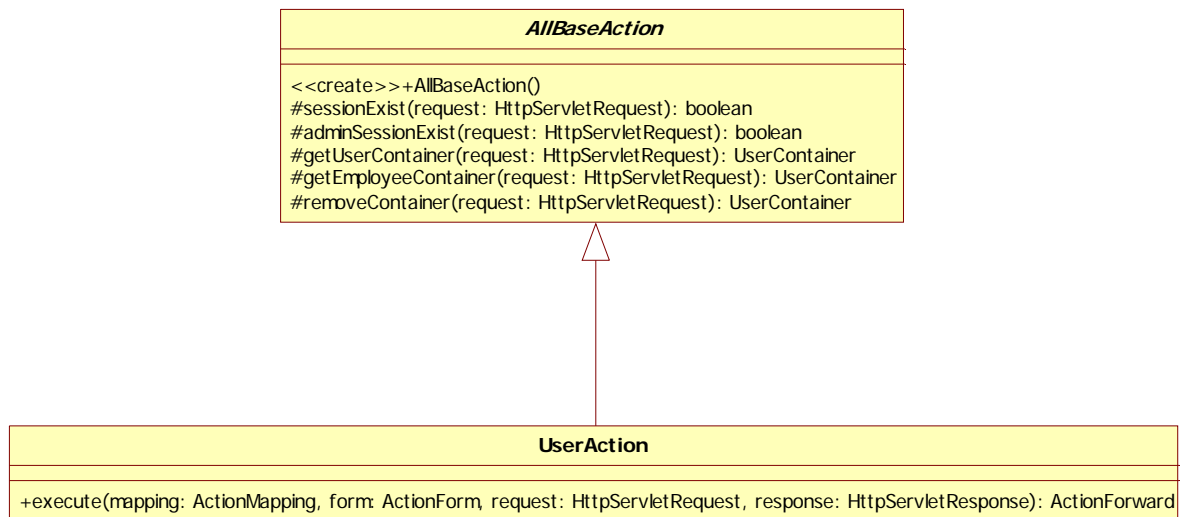
**Data Access Layer**

Menangani koneksi ke database dan query data.

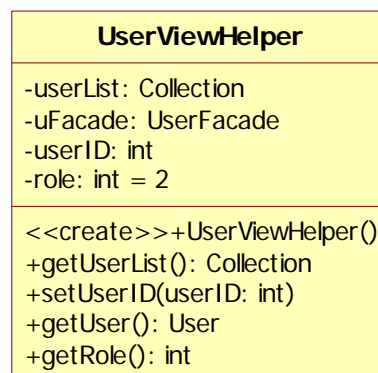
sehingga Class Diagramnya daat dibuat seperti berikut:

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

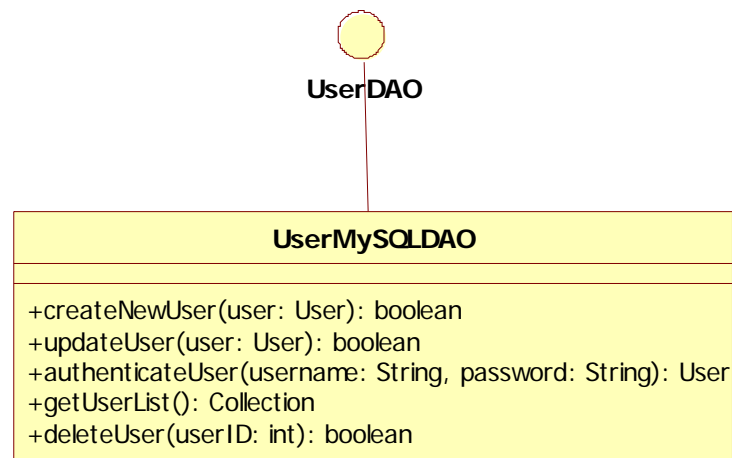
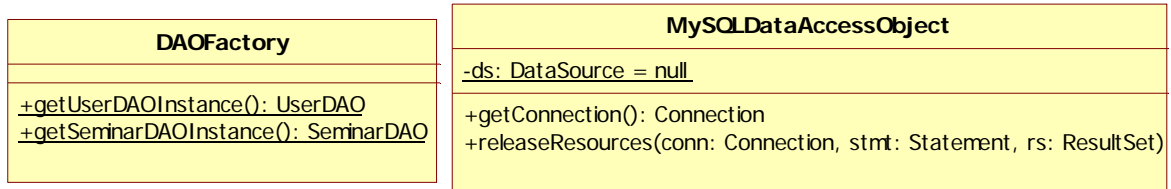
### Application Interface



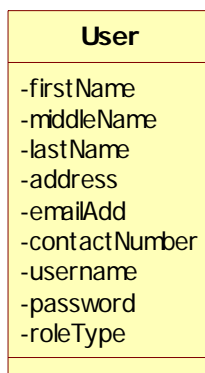
### Facade Layer



## JENI 3 – 07 Modul Praktikum Pengenalan MVC




### Data Access Layer



## JENI 3 – 07 Modul Praktikum Pengenalan MVC



### Percobaan 2 : Menyiapkan aplikasi project MVC

|   |   |
|---|---|
| <br>Info | Tujuan section ini adalah menunjukkan cara membuat project aplikasi web berbasis MVC.<br><br>Untuk aplikasi yang dibahas pada latihan kali ini kita akan menggunakan Struts versi 1.2.9 built in pada IDE Netbeans. |
|---|---|

### Membuat project dengan Netbeans IDE

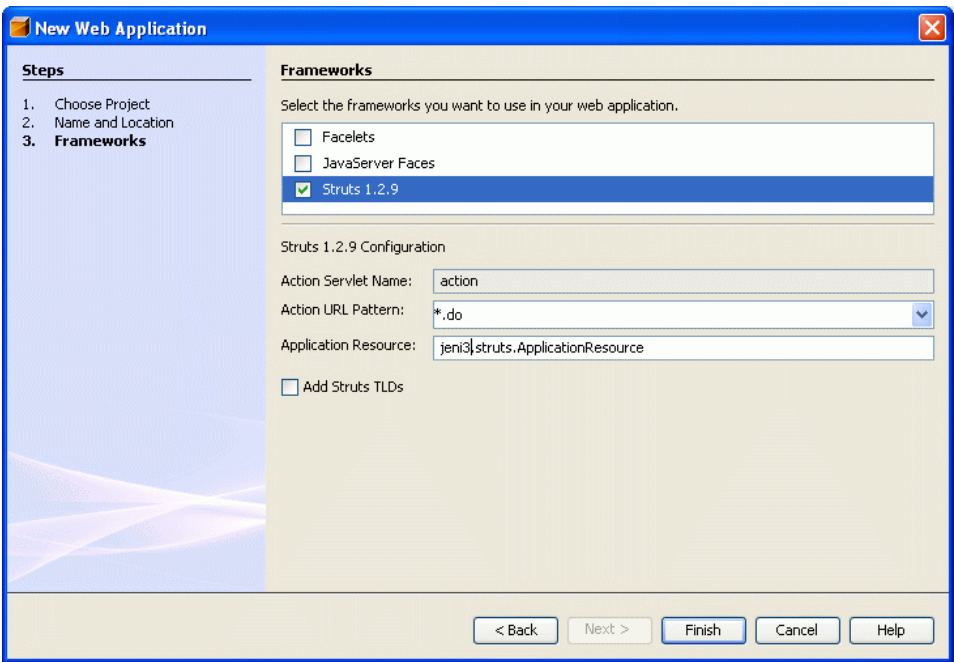
#### Langah-langkahnya:

- Buat project web baru dengan nama **jeni\_struts**

|  |   |
|--|---|
| <br>Tip  | File → New Project, pilih category Web → Web Application, klik Next<br>Ketikkan nama project, klik Next<br>Pilih Framework Struts 1.2.9, lakukan konfigurasi secukupnya.<br>Klik Finish |
| <br>Tip | Konfigurasi pada Struts:<br>Dengan Netbeans IDE:  |



## JENI 3 – 07 Modul Praktikum Pengenalan MVC



Action URL Pattern = ekstensi fiktif aplikasi  
Application resource = lokasi dimana Resource Bundle berada

### Secara Manual:

#### 1. Download Struts

Struts MVC Framework bisa didownload di <http://struts.apache.org/downloads.html> . Dalam distribusi Struts terdapat file struts-core.jar, dependencinya, contoh aplikasi dan dokumentasi dalam HTML format.

Saat ini Struts memiliki 2 versi Framework yaitu Struts1 dan Struts2. Struts1 sebagai Struts murni sampai saat ini mencapai versi 1.3.9. Sementara Struts2 sebagai versi terbaru Struts didalamnya menggunakan ActionController XWork dari Open Symphony. Saat ini masih versi 2.0.1.



2. Menambahkan librari struts dan dependencinya (struts.jar, antlr.jar, common-beanutils.jar, common-digester.jar, common-logging.jar, common-validator.jar, jakarta-oro.jar) ke dalam folder WEB-INF/lib (CLASSPATH aplikasi)
3. Membuat file web.xml dalam folder WEB-INF dan menambahkan kode berikut:

```
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-
class>org.apache.struts.action.ActionServlet</servlet-class>
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
<init-param>
  <param-name>config</param-name>
  <param-value>/WEB-INF/struts-config.xml</param-value>
</init-param>
<init-param>
  <param-name>debug</param-name>
  <param-value>2</param-value>
</init-param>
<init-param>
  <param-name>detail</param-name>
  <param-value>2</param-value>
</init-param>
<load-on-startup>2</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>
    30
  </session-timeout>
</session-config>
<welcome-file-list>
<welcome-file>
  index.jsp
</welcome-file>
</welcome-file-list>
```

### Percobaan 3 : Membuat halaman Login

|   |  |
|---|--|
| <br>Info | Tujuan section ini adalah membuat halaman Login dalam aplikasi MVC.                        |
| <br>Info | Section ini melanjutkan proejct pada section sebelumnya yaitu project <b>jeni_struts</b> . |

### Langkah-langkahnya:

1. Tambahkan link login dalam file index.jsp, seperti berikut:

```
<br>
<a href="./Login.do">Login</a>
```

4. Buatlah file JSP dengan nama **LoginForm.jsp** dan simpan di dalam direktori **WEB-INF/view\_pages**. Jika direktori view\_pages belum ada silahkan dibuat.

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

Isikan kode berikut:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Login</title>
</head>
<body>
<h1>Login Form</h1>
<form name="loginForm" action="Logon.do" method="POST">
  Username      : <input type="text" name="username" value="" size="30" /><br>
  Password      : <input type="password" name="password" value="" size="30" /><br>
  <input type="reset" value="Cancel" name="cancel" />
  <input type="submit" value="Login" name="login" />
</form>
</body>
</html>
```

5. Buat package dengan nama jeni3.struts.user.actionform
6. Dalam package jeni3.struts.user.actionform kemudian buatlah class Action Form dengan nama LoginForm.



Tip

File → New File atau klik kanan SourcePackage - New File/Folder,  
pilih category Web, pilih Struts ActionForm Bean, klik Next  
Ketikkan nama class-nya, pilih Superclass, klik Finish  
**atau**  
klik kanan SourcePackage - New Java Class,  
masukkan nama class-nya, klik Finish

Berikut kode pada class LoginForm:

```
package jeni3.struts.user.actionform;

import org.apache.struts.action.ActionForm;

/**
 * @author mee_andto@yahoo.com
 * @version 0.5
 */
public class LoginForm extends ActionForm{
  private String username;
  private String password;

  public String getUsername(){
    return username;
  }
  public void setUsername(String username){
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
this.username = username;
}
public String getPassword(){
    return password;
}
public void setPassword(String password){
    this.password = password;
}
}
```

7. Tambahkan mapping untuk Form Bean dalam file struts-config.xml, seperti kode berikut:

```
<form-beans>
    <form-bean name="loginForm" type="jeni3.struts.user.actionform.LoginForm" />
</form-beans>
```

8. Buat package dengan nama **jeni3.struts.util**
9. Dalam package **jeni3.struts.util** itu, buat class Java dengan nama **JENIConstants** dengan isi kode seperti berikut:

```
package jeni3.struts.util;

/**
 * @author mee_andto@yahoo.com
 * @version 0.5
 */
public class JENIConstants {
    public final static String RETURN_SUCCESS = "success";
    public final static String RETURN_INPUT = "input";
    public final static String RETURN_FAILURE = "failure";

    public static String USER_SESSION = "UserContainer";
}
```



Class **JENIConstants** adalah class yang berisi konstanta dalam aplikasi kita ini.

10. Buat package dengan nama **jeni3.struts.user.action**.
11. Dalam package **jeni3.struts.user.action**, kemudian buatlah class Action dengan nama **LoginAction**.

## JENI 3 – 07 Modul Praktikum Pengenalan MVC



## Tip

File → New File atau klik kanan SourcePackage - New File/Folder,  
pilih category Web, pilih Struts Action, klik Next  
Ketikkan nama class-nya, pilih location, Superclass, tentukan  
actionPath-nya, klik Next.  
Kalau checkbox pada wizard "Use ActionForm Bean" maka akan  
ditambahkan otomatis ke struts-config.xml  
**atau**  
klik kanan SourcePackage - New Java Class,  
masukkan nama class-nya, klik Finish. Kalau yang secara  
manual, ketikkan kode-kode berikut:

Berikut kode pada class LoginAction:

```
package jeni3.struts.user.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import jeni3.struts.user.actionform.LoginForm;
import jeni3.struts.util.JENIConstants;
import org.apache.struts.action.*;

/**
 * @author mee\_andto@yahoo.com
 * @version 0.5
 */
public class LoginAction extends Action {
    public ActionForward execute(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {

        LoginForm lf = (LoginForm)form;
        if (isValid(lf.getUsername())) return
(mapping.findForward(JENIConstants.RETURN_INPUT));
        if (isValid(lf.getPassword())) return
(mapping.findForward(JENIConstants.RETURN_INPUT));
    }
}
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```

        return (mapping.findForward(JENIConstants.RETURN_SUCCESS));
    }

    public boolean isValid(String value){
        return (value == null || value.length() == 0);
    }
}

```

12. Tambahkan action mapping dalam file struts-config.xml, seperti kode berikut:

```

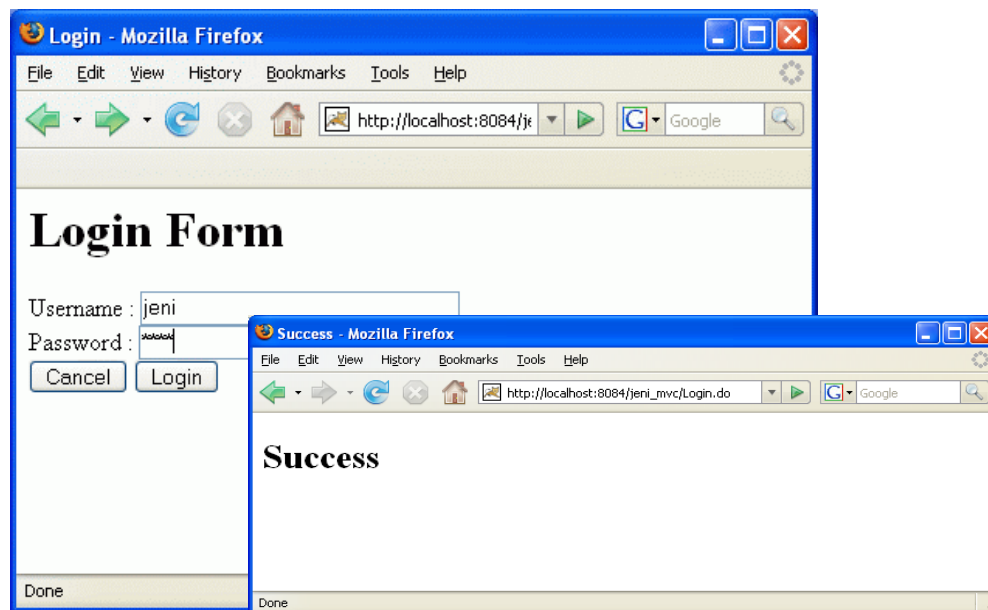
<action path="/Login" forward="/WEB-INF/view_pages/LoginForm.jsp"/>
<action name="loginForm"
        path="/Login"
        scope="request"
        type="jeni3.struts.user.action.LoginAction">
    <forward name="success" path="/WEB-INF/view_pages/Success.jsp"/>
    <forward name="input" path="/WEB-INF/view_pages/LoginForm.jsp"/>
</action>

```

13. Dalam folder **view\_pages**, buat file JSP dengan nama **Success.jsp**. Isikan tulisan `<h1>Success</h1>`.




14. Sampai disini, jalankan aplikasi anda, Run → Run (Main) Project

Demikian langkah-langkah penerapan pola MVC dalam pengembangan aplikasi web. Cukup mudah bukan. Semoga bisa dipahami langkah-langkahnya.



## JENI 3 – 07 Modul Praktikum Pengenalan MVC

### Percobaan 4 : Penggunaan HTML Tag Library

|   |   |
|---|---|
| <br>Info | Tujuan section ini adalah menggunakan HTML Tag Library dalam form.  |
| <br>Info | Section ini masih melanjutkan project pada section sebelumnya yaitu project <b>jeni_struts</b> .  |
| <br>Info | Untuk menggunakan HTML TagLib terlebih dahulu dalam setiap file JSP yang menggunakan HTML TagLib harus mendefinisikan taglib struts-html.tld.<br><b><code>&lt;%@taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%&gt;</code></b> |

#### Langkah-langkahnya:

1. Buka kembali file LoginForm.jsp dan lakukan modifikasi kode dengan memasukkan HTML TagLib, sehingga kodenya menjadi seperti berikut:





```
<%@taglib uri="http://jakarta.apache.org/struts/tags-html"
prefix="html"%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <title>Login</title>
  </head>
  <body>
<h1>Login</h1>
<html:form action="Logon.do" method="POST">
  Username : <html:text property="username"/> <br/>
  Password  : <html:password property="password"/> <br/>
  <html:reset/>
  <html:submit />
  </html:form>
</body>
</html>
```

#### 15. Run – Run Project



## JENI 3 – 07 Modul Praktikum Pengenalan MVC

### Percobaan 5 : Penggunaan Bean Tag Library

|   |  |
|---|--|
| <br>Info   | Tujuan section ini adalah menggunakan Bean Tag Library dalam menampilkan data dari Java Bean.  |
| <br>Info   | Section ini masih melanjutkan project pada section sebelumnya yaitu project <b>jeni_struts</b> .   |
| <br>Info   | <p>Bean TagLib adalah suatu tag yang memproses atribut pada class bean dan menampilkannya ke view aplikasi.</p> <p>Untuk menggunakan Bean TagLib terlebih dahulu dalam setiap file JSP yang menggunakan Bean TagLib harus mendefinisikan taglib struts-bean.tld.</p> <pre>&lt;%@taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%&gt;</pre> |
| <br>Info | <p>Skenario:</p> <p>Jika user berhasil login maka data username akan disimpan dalam session, kemudian data ini akan dipanggil dengan bean taglib dan ditampilkan.</p>  |

### Langkah-langkahnya:

1. Dalam Source Packages, buat package **jeni3.struts.bean**
2. Dalam package **jeni3.struts.bean** buat class Java Bean dengan nama **User**

```
package jeni3.struts.bean;

/**
 * @author mee_andto@yahoo.com
 * @version 0.5
 */
public class User {

    private int userID;
    private String firstName;
    private String middleName;
    private String lastName;
    private String address;
    private String emailAdd;
    private String contactNumber;
```



## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
private String username;  
private String password;  
private String roleType;  
  
//getter and setter  
}
```



Tip

Menggenerate getter dan setter secara otomatis:  
**Refactor** → **Encapsulate Fields**, klik Next  
(kiri bawah) klik **Do Refactoring**

3. Buat package **jeni3.struts.util.container**
4. Membuat class Container dengan nama **UserContainer**

```
package jeni3.struts.util.container;  
  
import javax.servlet.http.HttpSessionBindingEvent;  
import javax.servlet.http.HttpSessionBindingListener;  
import jeni3.struts.bean.User;  
  
/**  
 * @author mee\_andto@yahoo.com  
 * @version 0.5  
 */  
public class UserContainer implements HttpSessionBindingListener{  
    private User user = null;  
  
    public UserContainer() {  
        super();  
    }  
  
    public void valueBound(HttpSessionBindingEvent event) {  
    }  
  
    public void valueUnbound(HttpSessionBindingEvent event) {  
        cleanup();  
    }  
  
    public void cleanup() {  
        setUser(null);  
    }  
  
    //Add more getter and setter  
  
    public User getUser() {  
        return user;  
    }  
}
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
public void setUser(User user) {  
    this.user = user;  
}  
}
```



Info

Class UserContainer digunakan untuk menampung dan merawat object User dalam session nantinya.

5. Dalam package jeni3.struts.util.Container buat class dengan nama ContainerHandler. Berikut kodenya:

```
package jeni3.struts.util.container;  
  
import javax.servlet.http.HttpServletRequest;  
import jeni3.struts.util.*;  
  
/**  
 * @author mee\_andto@yahoo.com  
 * @version 0.5  
 */  
public class ContainerHandler {  
  
    /** Creates a new instance of ContainerHandler */  
    public ContainerHandler() {  
    }  
  
    public static UserContainer getUserContainer(HttpServletRequest request){  
        UserContainer container = (UserContainer)  
request.getSession().getAttribute(JENIConstants.USER_SESSION);  
        if (container == null){  
            container = new UserContainer();  
request.getSession().setAttribute(JENIConstants.USER_SESSION,  
container);  
        }  
        return container;  
    }  
}
```

getUserCotainer di sini untuk mendapatkan nilai session user.

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

6. Kita akan menambahkan kode pada LoginAction, agar ketika login informasi user tersimpan dalam session. Buka class LoginAction dan lengkapi kodenya menjadi seperti berikut:

```
package jeni3.struts.user.action;

import javax.servlet.http.*;
import jeni3.struts.bean.User;
import jeni3.struts.user.actionform.LoginForm;
import jeni3.struts.util.*;
import jeni3.struts.util.container.*;
import org.apache.struts.action.*;

/**
 * @author mee\_andto@yahoo.com
 * @version 0.5
 */
public class LoginAction extends Action{
    public ActionForward execute(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {

        LoginForm lf = (LoginForm)form;
        if (isValid(lf.getUsername())) return
(mapping.findForward(JENIConstants.RETURN_INPUT));
        if (isValid(lf.getPassword())) return
(mapping.findForward(JENIConstants.RETURN_INPUT));
        if (!lf.getUsername().equals("jeni"))
            return (mapping.findForward(JENIConstants.RETURN_INPUT));
        User user = new User();
        user.setUsername(lf.getUsername());
        UserContainer existingContainer =
ContainerHandler.getUserContainer(request);
        existingContainer.setUser(user);

        return (mapping.findForward(JENIConstants.RETURN_SUCCESS));
    }

    public boolean isValid(String value){
        return (value == null || value.length() == 0);
    }
}
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

7. Buka kembali file **Success.jsp** dan lakukan modifikasi kode dengan memasukkan Bean TagLib, sehingga kodenya menjadi seperti berikut:

```
<%@taglib uri="http://jakarta.apache.org/struts/tags-bean"
prefix="bean"%>
<%@taglib uri="http://jakarta.apache.org/struts/tags-html"
prefix="html"%>
<html>
<head><title>Menu Anda</title></head>
<body>
<h1>Success Login Page</h1>
<b>Hello <bean:write name="UserContainer" property="user.username"
scope="session"/></b>,
Selamat Datang di Negeri Dongeng<br/><br/>
<html:link page="./Logout.do">Logout</html:link><br/>
Menu Anda:<br/>
Administrator
</body>
</html>
```

**Action Logout**

8. Buat Action Mappingnya

```
<action
    path="/Logout"
    name="loginForm"
    type="jeni3.struts.user.action.LogoutAction"
    scope="request">
    <forward name="success" path="/WEB-
INF/view_pages/Logout.jsp"/>
</action>
```

9. Dalam package jeni3.struts.user.action, buat class LogoutAction

```
package jeni3.struts.user.action;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import jeni3.struts.util.JENIConstants;
import jeni3.struts.util.container.ContainerHandler;
import org.apache.struts.action.*;

/**
 * @author mee_andto@yahoo.com
 * @version 0.5
 */
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
*/  
  
public class LogoutAction extends Action{  
    public ActionForward execute(ActionMapping mapping, ActionForm  
form,  
        HttpServletRequest request, HttpServletResponse  
response)  
        throws Exception {  
        HttpSession session = request.getSession(false);  
        if(session != null) {  
            if(ContainerHandler.getUserContainer(request) != null) {  
                ContainerHandler.removeContainer(request);  
            }  
        }  
        return (mapping.findForward(JENIConstants.RETURN_SUCCESS));  
    }  
}
```

10. Pada class **ContainerHandler** tambahkan kode berikut:

**Logika:** Hanya user yang memiliki session saja yang akan dihilangkan (destroy) sessionnya.

```
public static UserContainer removeContainer(HttpServletRequest  
request) {  
    UserContainer container = (UserContainer)  
request.getSession().getAttribute(JENIConstants.USER_SESSION);  
    if (container != null){  
  
request.getSession().removeAttribute(JENIConstants.USER_SESSION);  
        container.cleanup();  
    }  
    return container;  
}
```




11. Buat file JSP **Logout.jsp** dalam folder /WEB-INF/view\_pages.

```
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html"  
prefix="html" %>  
You are logged out successfully. Go to <html:link  
action="./Login.do">Login</html:link> page
```

12. Run → Run Project

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

### Percobaan 6 : Penggunaan Logic Tag Library

|   |   |
|---|---|
| <br>Info | Tujuan section ini adalah menggunakan Logic Tag Library dalam menangani logika presentasi.  |
| <br>Info | Section ini masih melanjutkan project pada section sebelumnya yaitu project <b>jeni_struts</b> .  |
| <br>Info | Untuk menggunakan Logic TagLib terlebih dahulu dalam setiap file JSP yang menggunakan Logic TagLib harus mendefinisikan taglib struts-logic.tld.<br><pre>&lt;%@taglib uri="http://jakarta.apache.org/struts/tags-logic" prefix="logic"%&gt;</pre> |

### Langkah-langkahnya:

1. Tambahkan link pada file index.jsp, seperti berikut:

```
<br/>
<a href="./Menu.do">Menu Page</a>
```

2. Tambahkan mapping action pada file struts-config.xml, seperti berikut:

```
<action path="/Menu" forward="/WEB-INF/view_pages/Success.jsp"/>
```

3. Buka kembali file **Success.jsp** dan lakukan modifikasi kode dengan memasukkan Logic TagLib, sehingga kodenya menjadi seperti berikut:

```
<%@taglib uri="http://jakarta.apache.org/struts/tags-logic"
prefix="logic"%>
<%@taglib uri="http://jakarta.apache.org/struts/tags-bean"
prefix="bean"%>
<%@taglib uri="http://jakarta.apache.org/struts/tags-html"
prefix="html"%>
<html>
<head><title>Menu Anda</title></head>
<body>
<logic:present name="UserContainer" property="user" scope="session">
<h1>Success Login Page</h1>
<b>Hello <bean:write name="UserContainer" property="user.username"
scope="session"/></b>,
Selamat Datang di Negeri Dongeng<br/><br/>
<html:link page="/Logout.do">Logout</html:link><br/>
</logic:present>
<logic:notPresent name="UserContainer" property="user"
scope="session">
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```



Anda belum Login. Session anda tidak ada. Silahkan login disini
<html:link page="/Login.do">Login</html:link><br/>
</logic:notPresent>

Menu Anda:<br/>
<logic:present name="UserContainer" property="user" scope="session">
Administrator
</logic:present>
<logic:notPresent name="UserContainer" property="user"
scope="session">
No Menu
</logic:notPresent>
</body>
</html>

```

### 4. Run – Run Project

## Percobaan 7 : Struts dan koneksi ke database

|   |  |
|---|--|
| <br>Info  | <p>Tujuan section ini adalah memberikan contoh aplikasi MVC yang terkoneksi ke database.</p>   |
| <br>Info | <p>Section ini masih melanjutkan project pada section sebelumnya yaitu project <b>jeni_struts</b>.<br/>Section ini merupakan implementasi dari diagram yang ada pada Percobaan 1 di atas.<br/>Database yang digunakan pada section ini adalah MySQL.</p> |

### Langkah-langkahnya:

1. Buat package jeni3.struts.dao
2. Buat class MySQLDataAccessObject

```

package jeni3.struts.dao;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sql.DataSource;

/**

```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
* @author mee\_andto@yahoo.com
* @version 0.5
*/
public class MySQLDataAccessObject {
    private static DataSource ds = null;

    static{
        try {
            Context ctx = new InitialContext();
            Context envCtx ;

            envCtx = (Context) ctx.lookup("java:comp/env");
            ds = (DataSource)envCtx.lookup("jdbc/jeni_struts");
        } catch (NamingException ne) {
            System.out.println("Naming Exception" + ne);
        }
    }

    public Connection getConnection() throws SQLException{
        if (ds == null){
            return null;
        }
        return ds.getConnection();
    }

    public void releaseResources(Connection conn, Statement stmt,
ResultSet rs){
        //try closing RsultSet operation
        try {
            if(rs !=null){
                rs.close();
            }
        } catch (SQLException se) {
            System.out.println("Error occured while closing result
set");
            se.printStackTrace();
            //other error handling code here
        }

        //try closing Statement operation
        try {
            if (stmt != null){
                stmt.close();
            }
        } catch (SQLException se) {
            System.out.println("Error occured while closing
statement");
            se.printStackTrace();
            //other error handling code here
        }

        //try closing Connection operation
```



## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
try {
    if (conn != null){
        conn.close();
    }
} catch (SQLException se) {
    System.out.println("Error occured while closing
connection");
    se.printStackTrace();
    //other error handling code here
}
}
```

### 3. Koneksi via JNDI

Pada kode diatas terdapat kode dimana DataSource diperoleh dengan mencari Context yang bernama "**jdbc/jeni\_struts**". Context **jdbc/jeni\_struts** diperoleh dari Context yang berada dalam *active directory*-nya Web Container. Untuk dapat menyediakan Context tersebut perlu dilakukan konfigurasi terhadap Web Container dan aplikasi struts ini. Berikut caranya:

- Tambahkan kode deklarasi Context Resource berikut ke dalam file server.xml. Terletak di direktori CATALINA\_HOME/conf.

```
<Host .... >
    <Context crossContext="true"
docBase="C:/programme/app/jeni_struts/build/web"
path="/jen_i_struts">
        <Resource auth="Container"
driverClassName="com.mysql.jdbc.Driver"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
name="jdbc/jeni_struts" password="password" scope="Shareable"
type="javax.sql.DataSource"
url="jdbc:mysql://localhost:3306/jeni_struts?autoReconnect=true"
username="root"/>
    </Context>
</Host>
```



Tip

Gunakan menu pada Netbeans. Pada tab Runtime, pilih Servers – klik kanan pada Bundled Tomcat (5.5.17) – pilih edit server.xml. atau  
Gunakan Tomcat Administration Tool  
(<http://localhost:8084/admin>)

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

- Buka file **web.xml** pada aplikasi anda dan tambahkan kode berikut:

```
<resource-ref>
  <res-ref-name>jdbc/jeni_struts</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- Buka file **Context.xml** pada aplikasi anda (dalam folder META-INF) dan tambahkan kode berikut:

```
<ResourceLink global="jdbc/jeni_struts" name="jdbc/jeni_struts"
type="javax.sql.DataSource"/>
```

- Pastikan dalam CLASSPATH (folder WEB-INF/lib) anda terdapat librari-librari yang dibutuhkan oleh jdbc, yaitu mysql-connector-java.jar versi 3.xx, common-collections-1.8.jar, common-dbcp.jar, common-pool.jar.
- Restart Server Tomcat

### Membuat Object DAO

1. Buat package jeni3.struts.dao.user
2. Dalam package itu buat interface UserDao

```
package jeni3.struts.dao.user;

import java.util.Collection;
import jeni3.struts.bean.User;

/**
 * @author mee\_andto@yahoo.com
 * @version 0.5
 */
public interface UserDao {
    public boolean createNewUser(User user);
    public boolean updateUser(User user);
    public boolean deleteUser(int userID);

    public User authenticateUser(String username, String password);
}
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

### 3. Dalam package yang sama, buat class lagi dengan nama **UserMySQLDAO**

```
package jeni3.struts.dao.user;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import jeni3.struts.bean.User;
import jeni3.struts.dao.MySQLDataAccessObject;

/**
 * @author mee_andto@yahoo.com
 * @version 0.5
 */
public class UserMySQLDAO extends MySQLDataAccessObject implements
UserDAO{
    public boolean createNewUser(User user){
        return false;
    }
    public boolean updateUser(User user){
        return false;
    }
    public boolean deleteUser(int userID){
        return false;
    }
    public User authenticateUser(String username, String password){
        User user = null;
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String sql = "SELECT * FROM users WHERE
username='"+username+"' AND password = '"+password+"'";
        try {
            conn = getConnection();
            stmt = conn.createStatement();
            rs = stmt.executeQuery(sql);
            user = new User();
            int iCount = 0;
            while(rs.next()){
                user.setUserID(rs.getInt("userid"));
                user.setFirstName(rs.getString("firstname"));
                user.setMiddleName(rs.getString("middlename"));
                user.setLastName(rs.getString("lastname"));
                iCount++;
            }
            if (iCount == 0){
                System.out.println("No Results Found");
                return null;
            }else{
                return user;
            }
        }
    }
}
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
    }  
    } catch (SQLException e) {  
        System.out.println("Error while authenticate process");  
        e.printStackTrace();  
        return null;  
    } finally {  
        releaseResources(conn, stmt, rs);  
        //return user;  
    }  
}  
}
```

### 4. Dalam package `jeni3.struts.dao`, buat class **DAOFactory**

```
package jeni3.struts.dao;  
  
import jeni3.struts.dao.user.UserDAO;  
import jeni3.struts.dao.user.UserMySQLDAO;  
  
/**  
 * @author mee\_andto@yahoo.com  
 * @version 0.5  
 */  
public class DAOFactory {  
    public static UserDAO getUserDAOInstance(){  
        return new UserMySQLDAO();  
    }  
}
```

### Membuat class pada Facade layer

Seperti terlihat pada diagram aplikasi di atas bahwa fungsi Facade Layer disini akan menjembatani antara Data Access Object (DAO) dan View ActionHandler.

1. Buat package **jeni3.struts.facade**
2. Dalam package itu buat class **UserFacade**

```
package jeni3.struts.facade;  
  
import jeni3.struts.bean.User;  
import jeni3.struts.dao.DAOFactory;  
import jeni3.struts.dao.user.UserDAO;  
  
/**  
 * @author mee\_andto@yahoo.com  
 * @version 0.5  
 */
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
public class UserFacade {
    public User authenticateUser(String username, String password){
        UserDao dao = DAOFactory.getUserDAOInstance();

        return dao.authenticateUser(username,password);
    }
}
```

3. Kembali ke LoginAction, edit kode LoginAction menjadi seperti berikut:

```
package jeni3.struts.user.action;

import javax.servlet.http.*;
import jeni3.struts.bean.User;
import jeni3.struts.facade.UserFacade;
import jeni3.struts.user.actionform.LoginForm;
import jeni3.struts.util.*;
import jeni3.struts.util.container.*;
import org.apache.struts.action.*;

/**
 * @author mee\_andto@yahoo.com
 * @version 0.5
 */
public class LoginAction extends Action{
    public ActionForward execute(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {

        LoginForm lf = (LoginForm)form;
        if (isValid(lf.getUsername())) return
(mapping.findForward(JENIConstants.RETURN_INPUT));
        if (isValid(lf.getPassword())) return
(mapping.findForward(JENIConstants.RETURN_INPUT));

        UserFacade facade = new UserFacade();
        User user = new User();
        user = facade.authenticateUser(lf.getUsername(),
lf.getPassword());
        if (user != null){
            UserContainer existingContainer =
ContainerHandler.getUserContainer(request);
            existingContainer.setUser(user);

            return
(mapping.findForward(JENIConstants.RETURN_SUCCESS));
        }
        return (mapping.findForward(JENIConstants.RETURN_INPUT));
    }
}
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
public boolean isValid(String value){
    return (value == null || value.length() == 0);
}
```

### 4. Buat database jeni\_struts (MySQL Database)

```
CREATE DATABASE jeni_struts;
USE jeni_struts;

CREATE TABLE `users` (
  `userID` int(10) unsigned NOT NULL auto_increment,
  `firstName` varchar(45) NOT NULL,
  `middleName` varchar(45) NOT NULL,
  `lastName` varchar(45) NOT NULL,
  `address` varchar(100) NOT NULL,
  `emailAdd` varchar(45) NOT NULL,
  `contactNumber` varchar(45) NOT NULL,
  `username` varchar(45) NOT NULL,
  `password` varchar(45) NOT NULL,
  PRIMARY KEY (`userID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO users VALUES(1, 'JENI', 'Education Network',
'Indonesia', 'Malang', 'jeni@jardiknas.org', '', 'jeni', 'indonesia');
```

### 5. Run – Run Project

### 6. Login dengan username **jeni** dan password **indonesia**

Pola pengembangan software berbasis Struts

Ada 3 hal yang selalu dilakukan dalam membuat aplikasi dengan struts (mvc):

- Membuat halaman JSP atau lainnya (HTML, Velocity, Framemaker,dll) untuk menampilkan pesan atau tampilan ke user (View)
- Membuat ActionForm class untuk menangani logika aplikasi dan permintaan client (Model). Komponen Model meliputi JavaBeans dan Scope, ActionForm Beans, System State Beans dan Business Logic Beans.
- Membuat pemetaan (mapping) untuk mengarahkan antara action dengan halaman view atau action lain (Controller). Komponen Controller meliputi ActionServlet, Request Processor, Action Class, Action Mapping.

## EKSTRA

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

### Penggunaan html-link lebih detail

Kasus: Penggunaan tag html-link

Langkah-langkahnya:

- Buka kembali Project jeni\_mvc anda.
- Tambahkan link berikut di file **index.jsp**  
`<a href="./html-link.do">Contoh Penggunaan HTML Tag Library</a>`
- Buat sebuah Action class dengan nama **HTMLAction**, dengan kode seperti berikut:

```
package edu.jeni.struts;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.*;

public class HtmlAction extends Action {
    public ActionForward execute(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws Exception {

        if (isCancelled(request))
            return (mapping.findForward("index"));
        else
            return (mapping.findForward("input"));
    }
}
```

- Buat JavaBeans class dengan nama TestBean, seperti kode berikut:

```
package edu.jeni.struts;

import org.apache.struts.action.ActionForm;
/**
 *
 * @author User
 */
public class TestBean extends ActionForm{
    private boolean booleanProperty = true;

    public boolean getBooleanProperty() {
        return (booleanProperty);
    }

    public void setBooleanProperty(boolean booleanProperty) {
        this.booleanProperty = booleanProperty;
    }
}
```



## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
private int intProperty = 123;
private float floatProperty = (float) 123.0;
private String stringArray[] =
{ "String 0", "String 1", "String 2", "String 3", "String 4" };
private String stringProperty = "This is a string";

//Add More getter and setter here

private TestBean nested = null;
public TestBean getNested() {
    if (nested == null)
        nested = new TestBean();
    return (nested);
}
```

Jangan lupa melengkapi kode diatas dengan menambahkan getter dan setter untuk masing-masing variabel private. Gunakan menu **Refactor – Encapsulate Fields** pada Netbeans.

- Tambahkan ActionMapping dalam file konfigurasi seperti berikut:

```
<action      path="/html-link"
              type="edu.jeni.struts.HtmlAction"
              name="testbean"
              scope="session"
              validate="false">
    <forward name="input" path="/view_pages/html-link.jsp"/>
</action>
```

Tambahkan juga FormBean seperti berikut:

```
<form-bean name="testbean" type="edu.jeni.struts.TestBean" />
```

- Buat file JSP dengan nama **html-link.jsp** dalam direktori view\_pages dan berikut isinya berupa contoh-contoh kode penggunaan HTML Tag Library:

```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%
    String newValue = "New string value";
    pageContext.setAttribute("newValue", newValue);
    java.util.HashMap newValues = new java.util.HashMap();
    newValues.put("floatProperty", new Float(444.0));
    newValues.put("intProperty", new Integer(555));
    newValues.put("stringArray", new String[]
    { "Value 1", "Value 2", "Value 3" });
    pageContext.setAttribute("newValues", newValues);
%>
<html:html>
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
<head>
<title>Contoh html:link Tag</title>
</head>
<body bgcolor="white">
<div align="center">
<h1>Contoh Penggunaan struts-html Link Tag</h1>
</div>
<html:form action="html-link.do" method="POST">
<table border="0" width="100%">
  <tr>
    <th align="right">booleanProperty</th>
    <td align="left">
      <html:checkbox property="booleanProperty" />
    </td>
    <th align="right">intProperty</th>
    <td align="left">
      <html:text property="intProperty" size="16" />
    </td>
  </tr>
  <tr>
    <th align="right">floatProperty</th>
    <td align="left">
      <html:text property="floatProperty" size="16" />
    </td>
    <th align="right">stringProperty</th>
    <td align="left">
      <html:text property="stringProperty" size="16" />
    </td>
  </tr>
  <tr>
    <th align="right">stringArray</th>
    <td align="left" colspan="3">
      <html:text property="stringArray[0]" size="16" />
      <html:text property="stringArray[1]" size="16" />
    </td>
  </tr>
  <tr>
    <th colspan="4" align="center">
      Hyperlinks yang menggunakan atribut "page"
    </th>
  </tr>
  <tr>
    <td colspan="4" align="center">
      <html:link page="/html-link.do">
        Tanpa parameter
      </html:link>
    </td>
  </tr>
  <tr>
    <td colspan="4" align="center">
      <html:link page="/html-link.do?doubleProperty=321.321&longProperty=321321">

```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```

        Parameter Link yang berupa nilai Double dan long yang
        didefinisikan secara hard coded
        </html:link>
    </td>
</tr>
<tr>
    <td colspan="4" align="center">
        <html:link page="/html-link.do"
            paramId="stringProperty" paramName="newValue">
            Parameter Link yang berupa nilai String yang didefinisikan
            melalui parameter paramId dan paramName
        </html:link>
    </td>
</tr>
<tr>
    <td colspan="4" align="center">
        <html:link page="/html-link.do"
            paramId="booleanProperty"
            paramName="testbean" paramProperty="nested.booleanProperty">
            Parameter Link yang berupa nilai Boolean yang didefinisikan
            melalui parameter paramId, paramName dan paramValue
        </html:link>
    </td>
</tr>
<tr>
    <td colspan="4" align="center">
        <html:link page="/html-link.do"
            name="newValues">
            Parameter Link yang berupa nilai Float, int, dan stringArray
            yang didefinisikan melalui parameter name (Map)
        </html:link>
    </td>
</tr>
<tr>
    <th colspan="4" align="center">
        Hyperlinks yang menggunakan atribut "action"
    </th>
</tr>
<tr>
    <td colspan="4" align="center">
        <html:link action="/html-link">
            Tanpa Parameter
        </html:link>
    </td>
</tr>
<tr>
    <td colspan="4" align="center">
        <html:link action="/html-
link?doubleProperty=321.321&longProperty=321321">
            Parameter Link yang berupa nilai Double dan long yang
            didefinisikan secara hard coded
        </html:link>

```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```

        </td>
    </tr>
    <tr>
        <td colspan="4" align="center">
            <html:link action="/html-link"
                paramId="stringProperty" paramName="newValue">
                Parameter Link yang berupa nilai String yang didefinisikan
                melalui parameter paramId dan paramName
            </html:link>
        </td>
    </tr>
    <tr>
        <td colspan="4" align="center">
            <html:link action="/html-link"
                paramId="booleanProperty"
                paramName="testbean" paramProperty="nested.booleanProperty">
                Parameter Link yang berupa nilai Boolean yang didefinisikan
                melalui parameter paramId, paramName dan paramValue
            </html:link>
        </td>
    </tr>
    <tr>
        <td colspan="4" align="center">
            <html:link action="/html-link"
                name="newValues">
                Parameter Link yang berupa nilai Float, int, dan stringArray
                yang didefinisikan melalui parameter name (Map)
            </html:link>
        </td>
    </tr>
    <tr>
        <th colspan="4" align="center">Tombol Reset dan Cancel</th>
    </tr>
    <tr>
        <td colspan="4" align="center">
            <html:reset>Reset</html:reset>
            <html:cancel>Cancel</html:cancel>
        </td>
    </tr>
</table>
</html:form>

<p>
Penggunaan link: relative, context-relative, dan absolute.
</p>
<ul>
<li><html:link forward="relative">index page</html:link></li>
<li><html:link forward="context-relative">index page,
again</html:link></li>
<li><html:link forward="absolute">jakarta.apache.org</html:link></li>
</ul>
</html>

```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

Terlihat dalam kode JSP terdapat pendefinisian variabel/ property. Property atau variabel yang diparsing ke object View bisa dideklarasikan dan diinisialisasi baik melalui Java class maupun dalam file JSP sendiri.

- Karena dalam file jsp terdapat action forward maka perlu untuk menambahkan kode berikut dalam file konfigurasi:

```
<global-forwards>
  <forward name="absolute" path="http://jakarta.apache.org/" />
  <forward name="relative" path="/index.jsp" />
  <forward name="context-relative" path="/index.jsp"
contextRelative="true" />
  <forward name="index" path="/index.jsp" />
</global-forwards>
```

- Anda jalankan dan pelajari penggunaannya.

### Penggunaan tag html-select lebih detail

Sekarang kita coba latihan html TagLib lainnya, yaitu html-select. Dengan menggunakan project sebelumnya kita akan menambahkan action untuk html-select. Langkah-langkahnya:

1. Tambahkan kode berikut ke dalam ActionForm yaitu class **TestBean**:

```
private String singleSelect = "Single 5";
private String[] multipleSelect = { "Multiple 3", "Multiple 5",
                                     "Multiple 7" };
private String[] collectionSelect = { "Value 2", "Value 4",
                                     "Value 6" };
private String[] beanCollectionSelect = { "Value 1", "Value 3",
                                           "Value 5" };
private String resourcesSelect = "Resources 2";
private String withNulls = null;

private Collection beanCollection = null;

public Collection getBeanCollection() {
    if (beanCollection == null) {
        Vector entries = new Vector(10);

        entries.add(new LabelValueBean("Label 0", "Value 0"));
        entries.add(new LabelValueBean("Label 1", "Value 1"));
        entries.add(new LabelValueBean("Label 2", "Value 2"));
        entries.add(new LabelValueBean("Label 3", "Value 3"));
        entries.add(new LabelValueBean("Label 4", "Value 4"));
        entries.add(new LabelValueBean("Label 5", "Value 5"));
        entries.add(new LabelValueBean("Label 6", "Value 6"));
        entries.add(new LabelValueBean("Label 7", "Value 7"));
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
        entries.add(new LabelValueBean("Label 8", "Value 8"));
        entries.add(new LabelValueBean("Label 9", "Value 9"));

        beanCollection = entries;
    }

    return (beanCollection);
}

public void setBeanCollection(Collection beanCollection) {
    this.beanCollection = beanCollection;
}

//More getter and setter add here
```

**Jangan lupa** untuk menambahkan **getter dan setter sendiri** pada kode diatas untuk masing-masing variabel. Action class untuk html-select tetap memakai Action class yang telah kita buat sebelumnya.

2. Tambahkan ActionMapping ke dalam file konfigurasi, seperti kode berikut:

```
<action    path="/html-select"
          type="edu.jeni.struts.HtmlAction"
          name="testbean"
          scope="session"
          validate="false">
    <forward name="input" path="/view_pages/html-select.jsp"/>
</action>
```

3. Buat file JSP dengan nama **html-select.jsp** dalam direktori **view\_pages** dengan kode seperti berikut (**web/view\_pages/html-select.jsp**):

```
<%@ page language="java" import="java.util.*,
org.apache.struts.util.*"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>

<html:html>
<head>
<title>Test html:select Tag</title>
<%
    String multipleValues[] =
    { "Multiple 0", "Multiple 1", "Multiple 2", "Multiple 3", "Multiple
4",
      "Multiple 5", "Multiple 6", "Multiple 7", "Multiple 8", "Multiple
9" };
    pageContext.setAttribute("multipleValues", multipleValues);

    Vector options = new Vector();
    options.add(new LabelValueBean("Label 0", "Value 0"));
    options.add(new LabelValueBean("Label 1", "Value 1"));
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
options.add(new LabelValueBean("Label 2", "Value 2"));
options.add(new LabelValueBean("Label 3", "Value 3"));
options.add(new LabelValueBean("Label 4", "Value 4"));
options.add(new LabelValueBean("Label 5", "Value 5"));
options.add(new LabelValueBean("Label 6", "Value 6"));
options.add(new LabelValueBean("Label 7", "Value 7"));
options.add(new LabelValueBean("Label 8", "Value 8"));
options.add(new LabelValueBean("Label 9", "Value 9"));
pageContext.setAttribute("options", options);

String withNulls[] =
    { "String 0", null, "String 2" };
pageContext.setAttribute("withNulls", withNulls);

%>
</head>
<body bgcolor="white">

<div align="center">
<h1>Test struts-html Select Tag</h1>
</div>
<jsp:useBean id="testbean" scope="session" class="<jsp:useBean
id="testbean" scope="session" class="edu.jeni.struts.TestBean"/>

<html:form action="html-select.do" method="POST">
<table border="0" width="100%">

<tr>
<th align="right">Single Select Allowed:</th>
<td align="left">
<html:select property="singleSelect" size="10">
<html:option value="Single 0">Single 0</html:option>
<html:option value="Single 1">Single 1</html:option>
<html:option value="Single 2">Single 2</html:option>
<html:option value="Single 3">Single 3</html:option>
<html:option value="Single 4">Single 4</html:option>
<html:option value="Single 5">Single 5</html:option>
<html:option value="Single 6">Single 6</html:option>
<html:option value="Single 7">Single 7</html:option>
<html:option value="Single 8">Single 8</html:option>
<html:option value="Single 9">Single 9</html:option>
</html:select>
</td>
</tr>

<tr>
<th align="right">Multiple Select Allowed:</th>
<td align="left">
<html:select property="multipleSelect" size="10" multiple="true">
<html:options name="multipleValues" labelName="multipleValues"/>
</html:select>
</td>
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
</tr>

<tr>
  <th align="right">Multiple Select From A Collection (Using
<html:options>):</th>
  <td align="left">
    <html:select property="collectionSelect" size="10"
multiple="true">
      <html:options collection="options" property="value"
labelProperty="label"/>
    </html:select>
  </td>
</tr>

<tr>
  <th align="right">Multiple Select From A Collection (Using
<html:optionsCollection>):</th>
  <td align="left">
    <html:select property="beanCollectionSelect" size="10"
multiple="true">
      <html:optionsCollection name="testbean"
property="beanCollection"/>
    </html:select>
  </td>
</tr>

<tr>
  <th align="right">Select With Labels From Resources:</th>
  <td align="left">
    <html:select property="resourcesSelect" size="3">
      <html:option value="Resources 0" key="resources0"/>
      <html:option value="Resources 1" key="resources1"/>
      <html:option value="Resources 2" key="resources2"/>
    </html:select>
  </td>
</tr>

<tr>
  <th align="right">
    Collection with null labels and values:
  </th>
  <td align="left">
    <html:select property="withNulls" size="3">
      <html:options name="withNulls" labelName="withNulls"/>
    </html:select>
  </td>
</tr>

<tr>
  <td align="right">
    <html:submit>Save</html:submit>
  </td>
```



## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
<td align="left">
  <html:reset>Reset</html:reset>
  <html:cancel>Cancel</html:cancel>
</td>
</tr>

</table>
</html:form>
</html:html>
```

## 4. Run-Run Main Project

**Bean Tag Library**

Bean TagLib adalah suatu tag yang memproses atribut pada class bean dan menampilkannya ke view aplikasi.

Untuk menggunakan Bean TagLib terlebih dahulu dalam setiap file JSP yang menggunakan Bean TagLib harus mendefinisikan taglib struts-bean.tld.

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
```

Berikut contoh penggunaan bean TagLib:

**<bean:define />**

- Buat file JSP dengan nama **bean-define.jsp** dalam folder **web/view\_pages**. Berikut kodenya:

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html>
<head>
<title>Test struts-bean:define Tag</title>
</head>
<body bgcolor="white">
<div align="center">
<h1>Test struts-bean:define Tag</h1>
</div>

<jsp:useBean id="test1" scope="page" class="edu.jeni.struts.TestBean"/>

<bean:define id="test1_boolean" name="test1"
property="booleanProperty"/>
<bean:define id="test1_float" name="test1" property="floatProperty"/>
<bean:define id="test1_string" name="test1" property="stringProperty"/>
<bean:define id="test1_int" name="test1" property="intProperty"/>
<bean:define id="test1_ngetest" value="Testing ABCDE"/>

<table border="1">
<tr>
<th>Data Type</th>
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
<th>Correct Value</th>
<th>Test Result</th>
</tr>
<tr>
<td>boolean</td>
<td><jsp:getProperty name="test1" property="booleanProperty"/></td>
<td><%= test1_boolean %></td>
</tr>
<tr>
<td>float</td>
<td><jsp:getProperty name="test1" property="floatProperty"/></td>
<td><%= test1_float %></td>
</tr>
<tr>
<td>int</td>
<td><jsp:getProperty name="test1" property="intProperty"/></td>
<td><%= test1_int %></td>
</tr>
<tr>
<td>String</td>
<td><jsp:getProperty name="test1" property="stringProperty"/></td>
<td><%= test1_string %></td>
</tr>
<tr>
<td>Test value</td>
<td><bean:write name="test1_ngetest"/></td>
<td><%= test1_ngetest %></td>
</tr>
</table>
</body>
</html>
```

- Buat ActionMapping dalam file struts-config.xml:

```
<action path="/bean-define" forward="/view_pages/bean-define.jsp">
</action>
```

- Tambahkan link di file index.jsp:

```
<br>
<a href="/bean-define.do">Contoh Penggunaan Bean TagLib
bean:define</a>
```

- Run-Run Main Project

**<bean:include />**

- Buat file JSP dengan nama **bean-include.jsp** dalam folder web/view\_pages. Berikut kodenya:

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html>
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
<head>
<title>Test struts-bean:include Tag</title>
</head>
<body bgcolor="white">
<div align="center">
<h1>Test struts-bean:include Tag</h1>
</div>

<bean:include id="index" page="/index.jsp"/>

<p>Display the contents returned by invoking <code>/index.jsp</code>
directly, with no filtering.</p>
<hr>
<pre>
<%= index %>
</pre>
<hr>

<p>Display the contents returned by invoking <code>/index.jsp</code>
directly, with filtering.</p>
<hr>
<pre>
<bean:write name="index" filter="true"/>
</pre>
<hr>

</body>
</html>
```

- Buat ActionMapping dalam file struts-config.xml:

```
<action path="/bean-include" forward="/view_pages/bean-
include.jsp">
</action>
```

- Tambahkan link di file index.jsp:

```
<br>
<a href="/bean-include.do">Contoh Penggunaan Bean TagLib
bean:include</a>
```

- Run-Run Main Project

**<bean:parameter/>**

- Buat file JSP dengan nama **bean-parameter.jsp** dalam folder web/view\_pages. Berikut kodenya:

```
<%@ page language="java" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html>
<head>
<title>Test struts-bean:parameter Tag</title>
</head>
<body bgcolor="white">
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

```
<div align="center">
<h1>Test struts-bean:parameter Tag</h1>
</div>

<p>If called from the <code>index.jsp</code> page, two request
parameters will
be included and their values displayed below. If you call this page
without
including the appropriate request parameters, you will receive a JSP
runtime
error instead.</p>

<bean:parameter id="param1" name="param1"/>
<bean:parameter id="param2" name="param2"/>
<bean:parameter id="param3" name="param3" value="UNKNOWN VALUE"/>

<table border="1">
  <tr>
    <th>Parameter Name</th>
    <th>Correct Value</th>
    <th>Test Result</th>
  </tr>
  <tr>
    <td>param1</td>
    <td>value1</td>
    <td><%= param1 %></td>
  </tr>
  <tr>
    <td>param2</td>
    <td>value2</td>
    <td><%= param2 %></td>
  </tr>
  <tr>
    <td>param3</td>
    <td>UNKNOWN VALUE</td>
    <td><%= param3 %></td>
  </tr>
</table>

</body>
</html>
```

- Buat ActionMapping dalam file struts-config.xml:

```
<action path="/bean-parameter" forward="/view_pages/bean-parameter.jsp">
</action>
```

- Tambahkan link di file index.jsp:

```
<br>
<a href="/bean-parameter.do?param1=value1&param2=value2">Contoh
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

Penggunaan Bean TagLib bean:parameter&lt;/a&gt;

- Run-Run Main Project

## &lt;bean:write /&gt;

- Buat file JSP dengan nama **bean-write.jsp** dalam folder web/view\_pages. Berikut kodenya:

```
<%@ page language="java" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html>
<head>
<title>Test struts-bean:write Tag</title>
</head>
<body bgcolor="white">

<div align="center">
<h1>Test struts-bean:write Tag</h1>
</div>
<jsp:useBean id="test" scope="page" class="edu.jeni.struts.TestBean"/>

<table border="1">
<tr>
<th>Data Type</th>
<th>Correct Value</th>
<th>Test Result</th>
</tr>
<tr>
<td>boolean</td>
<td><jsp:getProperty name="test" property="booleanProperty"/></td>
<td><bean:write name="test" property="booleanProperty"/></td>
</tr>
<tr>
<td>float</td>
<td><jsp:getProperty name="test" property="floatProperty"/></td>
<td><bean:write name="test" property="floatProperty"/></td>
</tr>
<tr>
<td>int</td>
<td><jsp:getProperty name="test" property="intProperty"/></td>
<td><bean:write name="test" property="intProperty"/></td>
</tr>
<tr>
<td>int</td>
<td><jsp:getProperty name="test" property="stringProperty"/></td>
<td><bean:write name="test" property="stringProperty"/></td>
</tr>
</table>
</body>
</html>
```

## JENI 3 – 07 Modul Praktikum Pengenalan MVC

- Tambahkan link di file index.jsp:

```
<br>
<a href="./bean-write.do">Contoh Penggunaan Bean TagLib
bean:write</a>
```

- Buat ActionMapping dalam file struts-config.xml:

```
<action path="/bean-write" forward="/view_pages/bean-write.jsp">
</action>
```

- Run-Run Main Project