

### **Aturan Praktikum:**

- Keterlambatan maksimal 10 menit
- Hasil praktikum, selesai atau tidak selesai ditanggung mahasiswa
- Nilai praktikum adalah nilai perorangan, sebagai tambahan nilai test atau quiz jika nilai test atau quiz kurang mencukupi
- Mahasiswa login ke komputer
- Persiapan dan penjelasan praktikum 10 menit
- Praktikum dilakukan selama 70 menit terdiri dari praktikum sesuai modul
- Hasil praktikum di-zip dan dikumpulkan
- Sisa waktu 20 menit digunakan untuk tes kecil hasil praktikum, saat tes tampilan komputer adalah background desktop dan mahasiswa tidak diperkenankan menggunakan komputer dan melihat modul praktikum
- Mahasiswa logout dari komputer
- Praktikum selesai

## **JSP (JAVASERVER PAGES) (BEGINNER) (Minggu 2 – Praktikum II)**

### **A. Sekilas JSP**

*JavaServer Pages* atau JSP merupakan salah satu bagian dari J2EE (*Java 2 Enterprise Edition*). JSP dibangun di atas Servlet dan bertujuan untuk menambah efisiensi. Kelebihan dari JSP yaitu:

- JSP mengijikan pemakaian pustaka tag standar dan tag buatan sendiri
- JSP dikompilasi terlebih dahulu sehingga proses server lebih efisien
- JSP dapat dikombinasikan dengan Servlet
- JSP adalah bagian dari J2EE sehingga dapat digunakan dalam aplikasi *enterprise*

Pada saat klien mengirimkan *request* ke JSP maka kontainer web akan mengecek apakah JSP telah terkompilasi, jika belum maka akan dikompilasi, kemudian seperti halnya servlet, JSP diinisialisasi, memproses *request*, dan akan menjalankan *destroy* ketika kontainer web dimatikan.

Kelemahan Servlet adalah memasukkan kode HTML sebagai kode Java yaitu mengirim kode HTML dengan menggunakan `println()` sehingga proses pengiriman menjadi lebih lama dan jika sebuah kode servlet tanpa menggunakan *framework* ingin diubah kode HTML-nya maka harus dilakukan oleh *programmer* karena harus mengutak-atik kode servlet, padahal perubahan kode HTML sering dilakukan oleh *web designer*. Oleh karena itu Sun Microsystem membuat teknologi JSP sebagai solusinya.

Salah satu keuntungan dari JSP adalah pengkodeannya yang seperti HTML sehingga orang yang tidak terbiasa dengan Java dapat lebih mudah mempelajari JSP, tapi kelemahan dari JSP adalah bercampurnya kode HTML dengan kode tag JSP. JSP dapat langsung dijalankan hanya dengan membuat *file* .jsp saja sedangkan Servlet harus dibungkus menjadi sebuah paket (.jar) dengan berbagai atribut yang diperlukan agar sebuah Servlet dapat dijalankan.

## B. Sintaks JSP

JSP memiliki tag standar yang bisa disebut dengan *JSP Standard Tag Library (JSTL)*, tapi JSP juga mengizinkan tag-tag yang dapat dibuat sendiri sesuai kebutuhan (*custom element*). Elemen standar pada JSP dapat dikelompokkan menjadi tiga yaitu:

- elemen perintah (*directive element*)
- elemen skrip (*scripting element*)
- elemen aksi (*action element*)

### a. Elemen Perintah (*Directive Element*)

Elemen perintah (*directive element*) adalah elemen untuk menspesifikasikan informasi tentang halaman yang bersangkutan. Secara umum tag elemen perintah adalah sebagai berikut:

```
<%@ elemen_perintah (atribut="nilai")* %>
```

dimana tanda bintang (\*) berarti elemen dapat memiliki atribut yang diulang sebanyak nol atau beberapa kali, misalkan:

```
<%@ page info="halaman contoh" %>
<%@ page buffer="16384" %>
<%@ page info="akses bebas" %>
```

Tag-tag yang termasuk dalam deklarasi elemen perintah adalah sebagai berikut:

| Tag   | Keterangan   |         |            |   |  |  |  |                    |                                    |
|---|--|---------|------------|---|--|--|--|--------------------|------------------------------------|
| <pre>&lt;%@ page (atribut="nilai")* %&gt;</pre> <p>atau</p> <pre>&lt;%@ page atribut1="nilai1" atribut2="nilai2" .... %&gt;</pre> | <p>mendefinisikan atribut halaman seperti <i>error</i>, <i>session</i>, dan keperluan <i>buffering</i></p> <p>atribut yang dimiliki:</p> <table border="1"> <thead> <tr> <th>Atribut</th> <th>Keterangan</th> </tr> </thead> <tbody> <tr> <td> <p>language</p> <p>contoh:</p> <pre>&lt;%@ page language= "javascript" %&gt; &lt;%     var helloStr = 'Hello World!'; %&gt;</pre> </td> <td> <p>menspesifikasikan bahasa yang digunakan dengan nilai standar "java"</p> </td> </tr> <tr> <td> <p>info</p> <p>contoh:</p> <pre>&lt;%@ page info="JSP pertamaku" %&gt;</pre> </td> <td> <p>informasi mengenai JSP, nilai standarnya berupa <i>string</i> tergantung pada kontainer web</p> </td> </tr> <tr> <td> <p>contentType</p> </td> <td> <p>menspesifikasikan data yang</p> </td> </tr> </tbody> </table> | Atribut | Keterangan | <p>language</p> <p>contoh:</p> <pre>&lt;%@ page language= "javascript" %&gt; &lt;%     var helloStr = 'Hello World!'; %&gt;</pre> | <p>menspesifikasikan bahasa yang digunakan dengan nilai standar "java"</p> | <p>info</p> <p>contoh:</p> <pre>&lt;%@ page info="JSP pertamaku" %&gt;</pre> | <p>informasi mengenai JSP, nilai standarnya berupa <i>string</i> tergantung pada kontainer web</p> | <p>contentType</p> | <p>menspesifikasikan data yang</p> |
| Atribut   | Keterangan   |         |            |   |  |  |  |                    |                                    |
| <p>language</p> <p>contoh:</p> <pre>&lt;%@ page language= "javascript" %&gt; &lt;%     var helloStr = 'Hello World!'; %&gt;</pre> | <p>menspesifikasikan bahasa yang digunakan dengan nilai standar "java"</p>   |         |            |   |  |  |  |                    |                                    |
| <p>info</p> <p>contoh:</p> <pre>&lt;%@ page info="JSP pertamaku" %&gt;</pre>  | <p>informasi mengenai JSP, nilai standarnya berupa <i>string</i> tergantung pada kontainer web</p>   |         |            |   |  |  |  |                    |                                    |
| <p>contentType</p>  | <p>menspesifikasikan data yang</p>   |         |            |   |  |  |  |                    |                                    |

| Tag | Keterangan   |  |
|-----|--|--|
|     | <p>contoh:</p> <pre>&lt;%@ page contentType= "text/xml" %&gt;</pre>  | <p>dikembalikan ke <i>browser</i> dengan nilai standar "text/html;charset=ISO-8859-1"</p>                |
|     | <p>extends</p> <p>contoh:</p> <pre>&lt;%@ page extends= "usingjsp. JSPSuperclass" %&gt;</pre>  | <p>menspesifikasikan <i>superclass</i> JSP dengan nilai standar <i>none</i></p>                          |
|     | <p>import</p> <p>contoh:</p> <pre>&lt;%@ page import= "java.sql.*" %&gt; &lt;%@ page import= "java.util.*" %&gt;</pre> <p>atau</p> <pre>&lt;%@ page import= "java.sql.*" import= "java.util.*" %&gt;</pre> | <p>jika diperlukan mengimpor pustaka lain dengan nilai standar <i>none</i></p>                           |
|     | <p>buffer</p> <p>contoh:</p> <pre>&lt;%@ page buffer="none" %&gt;</pre>  | <p>besar penggunaan <i>buffer</i> dengan nilai standar 8192</p>  |
|     | <p>autoFlush</p>   | <p>metode <i>flush</i> untuk buffer dengan nilai standar "true"</p>                                      |
|     | <p>session</p> <p>contoh:</p> <pre>&lt;%@ page session="true" %&gt;</pre>  | <p>diisi dengan <i>true</i> jika memakai <i>session</i> dengan nilai standar "true"</p>                  |
|     | <p>isThreadSafe</p> <p>contoh:</p> <pre>&lt;%@ page isThreadSafe= "false" %&gt;</pre>  | <p>mengijinkan penggunaan banyak <i>thread</i> jika bernilai <i>true</i> dengan nilai standar "true"</p> |

| Tag                                | Keterangan  |   |
|------------------------------------|---|---|
|                                    | <p>errorPage</p> <p>contoh:</p> <pre>&lt;%@ page errorPage= "error.jsp" %&gt;</pre>   | <p>halaman yang menangani dan menampilkan kesalahan jika terjadi kesalahan dengan nilai standar <i>none</i></p> |
|                                    | <p>isErrorPage</p> <p>contoh:</p> <pre>&lt;%@ page isErrorPage= "true" %&gt;</pre>  | <p>berisi <i>true</i> jika JSP merupakan halaman yang menangani <i>error</i> dengan nilai standar "false"</p>   |
|                                    | <p>contoh:</p> <pre>&lt;%@ page buffer="16384" session="false" %&gt;</pre> <p>atau</p> <pre>&lt;%@ page buffer="16384" %&gt; &lt;%@ page session="false" %&gt;</pre>  |   |
| <p>&lt;%@ include ...</p>          | <p>menyatakan <i>file</i> yang di-include</p> <p>contoh:</p> <pre>&lt;%@ include file="URL" %&gt;</pre> <p>atau</p> <pre>&lt;%@ include file="includes/header.html" %&gt; &lt;%   out.println("&lt;BODY&gt;"); %&gt; &lt;%@ include file="includes/footer.html" %&gt;</pre> |   |
| <p>&lt;%@ taglib ...<br/>%&gt;</p> | <p>mendeklarasikan pustaka tag (<i>tag library</i>) berisi aksi <i>custom</i> yang digunakan pada halaman web (<i>custom tag</i>)</p>   |   |

Elemen perintah juga dapat ditulis dalam bentuk XML menjadi seperti berikut:

```
<%@ namaDirective atribut1="nilai1" atribut2="nilai2" %>
```

menjadi

```
<jsp:directive.namaDirective atribut1="nilai1" atribut2="nilai2" />
```

## b. Elemen Skrip (*Scripting Element*)

Elemen skrip (*scripting element*) mengizinkan *programmer* memasukkan kode Java pada kode JSP. Berikut adalah beberapa elemen skrip yang dapat digunakan pada JSP:

- Sintaks mendeklarasikan metode dan variabel adalah sebagai berikut:

```
<%!  
    deklarasi_variabel_dan_metode  
%>
```

misalnya

```
<%!  
    String nama;  
    String alamat;  
    String ttl;  
  
    public final String tulisNama(String n){  
        return "Nama Masukan : "+ n;  
    }  
%>
```

- Sintaks pemakaian metode dan variabel tanpa ditampilkan hasil atau nilainya adalah sebagai berikut:

```
<%  
    pemakaian_metode_dan_variabel  
%>
```

misalnya

```
<%  
    tulisNama(nama);  
%>
```

- Sintaks menampilkan variabel nilai adalah sebagai berikut:

```
<%=  
    variabel_nilai_atau_metode  
%>
```

misalnya

```
<%=  
    tulisNama("Gadiza")  
%>
```

hasil tampilannya seperti pada gambar berikut:

|                       |
|-----------------------|
| Nama Masukan : Gadiza |
|-----------------------|

- Sintaks penulisan komentar pada JSP adalah sebagai berikut:

```
<%  
    // komentar  
%>
```

atau

```
<% /* komentar */ %>
```

atau

```
<% out.println("<!--komentar-->"); %>
```

atau

```
<!--komentar-->
```

Elemen skrip dapat ditulis dalam bentuk XML seperti berikut:

- <%! deklarasi kode %> menjadi  
`<jsp:directive:namaDirective list_atribut />`
- <% skrip kode %> menjadi  
`<jsp:scriptlet> skrip kode </jsp:scriptlet>`
- <%= ekspresi %> menjadi  
`<jsp:expression> ekspresi </jsp:expression>`

atau menggunakan tag umum berikut:

```
<jsp:text> teks </jsp:text>
```

### c. Elemen Aksi Standar

Elemen aksi merespon aksi yang berdasarkan informasi yang dibutuhkan halaman JSP ketika diminta oleh *browser*. Tag-tag yang termasuk dalam deklarasi elemen aksi standar adalah sebagai berikut:

| Tag               | Keterangan   |
|-------------------|--|
| <jsp:useBean>     | membuat komponen JavaBeans tersedia di halaman web<br><br>contoh:<br><pre>&lt;jsp:useBean id="theBean"<br/>class="com.math.CalculatorBean"/&gt;<br/>&lt;%<br/>int i = 4;<br/>int j = theBean.doubleIt(i);<br/>out.print("2*4=" + j);<br/>%&gt;</pre> |
| <jsp:getProperty> | mengambil sebuah nilai properti dari sebuah komponen JavaBeans dan mengikutsertakannya dalam respon<br><br>contoh:   |

| Tag               | Keterangan  |
|-------------------|---|
|                   | <pre>&lt;jsp:setProperty name="theBean" property="memory" value="&lt;%= 100 * i %&gt;" /&gt;</pre>  |
| <jsp:setProperty> | <p>mengeset nilai properti komponen JavaBeans</p>   |
| <jsp:include>     | <p>mengikutsertakan respon dari servlet atau JSP pada proses <i>request</i> (hampir sama dengan elemen perintah <i>include</i>)</p> <p>contoh:</p> <pre>&lt;jsp:include page="URL" flush="true" /&gt;</pre> <p>atau jika ingin melewati informasi di dalamnya akan menjadi</p> <pre>&lt;jsp:include page="relativeURL" flush="true"&gt; ( &lt;jsp:param ... /&gt; )* &lt;/jsp:include&gt;</pre> |
| <jsp:forward>     | <p>mengirimkan lagi proses <i>request</i> ke servlet atau JSP (menterminasi eksekusi JSP)</p> <p>contoh:</p> <pre>&lt;jsp:forward page="URL" /&gt;</pre> <p>atau jika ingin melewati informasi di dalamnya akan menjadi</p> <pre>&lt;jsp:forward page="URL"&gt; ( &lt;jsp:param . . . /&gt; )* &lt;/jsp:forward&gt;</pre>   |
| <jsp:param>       | <p>menambahkan sebuah nilai parameter ke penanganan <i>request</i> servlet atau JSP lain</p>  |
| <jsp:plugin>      | <p>menggenerasi HTML yang berisi elemen <i>OBJECT</i> atau <i>EMBED</i> yang dibutuhkan untuk mengeksekusi sebuah applet dengan perangkat <i>plug-in</i> dari Java</p>  |
| <jsp:params>      | <p>memiliki fungsi sama dengan tag param pada kode HTML applet, hanya dapat digunakan di dalam tag &lt;jsp:plugin&gt;</p> <p>contoh:</p> <pre>&lt;jsp:plugin type="applet" code="MyApplet.class" width="475" height="350"&gt; &lt;jsp:params&gt; &lt;jsp:param name="PARAM1" value="VALUE1" /&gt;</pre>   |

| Tag                               | Keterangan   |
|-----------------------------------|--|
|                                   | <pre>&lt;jsp:param name="PARAM2" value="VALUE2" /&gt; &lt;/jsp:params&gt; &lt;/jsp:plugin&gt;</pre>  |
| <code>&lt;jsp:fallback&gt;</code> | <p>menyediakan teks alternatif jika <i>browser</i> tidak mendukung elemen <i>OBJECT</i> dan <i>EMBED</i> HTML, hanya dapat digunakan di dalam tag <code>&lt;jsp:plugin&gt;</code></p> <p>contoh:</p> <pre>&lt;jsp:plugin type="applet"             code="MyApplet.class"             width="475" height="350"&gt; &lt;jsp:fallback&gt;   &lt;B&gt;Error: tidak ada Java.&lt;/B&gt; &lt;/jsp:fallback&gt; &lt;/jsp:plugin&gt;</pre> |

### C. Praktikum

#### a. Persiapan

- Membuat direktori kerja dengan nama SI319-P2-2-Kelas-NIM misalnya SI319-P2-2-A-23507024
- Di dalam direktori di atas, buat direktori JSP untuk menyimpan file-file yang akan dibuat.

#### b. Membuat Hello World

Susunan file yang harus dibuat di dalam direktori JSP adalah sebagai berikut:

- Direktori common
  - File build.properties
  - File targets.xml
- Direktori commonweb
  - File build.properties
  - File targets.xml
- Direktori HelloJSP
  - Direktori src  
Berisi file-file *source code*
  - Direktori web  
Berisi file-file html, jsp, gambar, dll
  - File build.xml
  - File sun-web.xml
  - File web.xml



File-file pada direktori common dan commonweb digunakan untuk melakukan kompilasi terhadap *source code* java.

### **Nama file: build.properties (pada direktori common)**

```
j2ee.home=C:/Sun/AppServer
j2ee.tutorial.home="C:/Servlet"
sunone.home=${j2ee.home}
admin.password.file=${j2ee.tutorial.home}/common/admin-password.txt
admin.host=localhost
admin.user=admin
admin.port=4848
https.port=8181
domain.resources="domain.resources"
domain.resources.port=8080
db.root=${j2ee.home}/pointbase
db.driver=com.pointbase.jdbc.jdbcUniversalDriver
db.host=localhost
db.port=9092
db.sid=sun-appserv-samples
db.url=jdbc:pointbase:server://${db.host}:${db.port}/${db.sid}
db.user=pbpublic
db.pwd=pbpublic
url.prop=DatabaseName
ds.class=com.pointbase.jdbc.jdbcDataSource
db.jvmargs=-ms16m -mx32m
```

Yang harus diganti dari file di atas jika berbeda path adalah sebagai berikut:

```
j2ee.home=C:/Sun/AppServer [path dimana sun application server
diinstal]
j2ee.tutorial.home="C:/Servlet" [path direktori servlet]
```

### **Nama file: targets.xml (pada direktori common)**

```
<path id="classpath">
  <fileset dir="${j2ee.home}/lib">
    <include name="j2ee.jar"/>
  </fileset>
</path>

<target name="clean" >
  <delete dir="${build}" />
  <delete dir="${dist}" />
  <delete dir="${assemble}" />
</target>

<path id="db.classpath">
  <fileset dir="${db.root}/lib">
    <include name="*.jar"/>
  </fileset>
</path>

<target name="create-db_common" depends="init"
```

```
description="Create database tables and populate database." >
<java classname="com.pointbase.tools.toolsCommander" fork="yes" >
  <jvmarg line="${db.jvmargs}" />
  <arg line="${db.driver} ${db.url} ${sql.script} ${db.user}
${db.pwd}" />
  <classpath refid="db.classpath" />
</java>
</target>

<target name="admin_command_common">
  <echo message="Doing admin task ${admin.command}"/>
  <sun-appserv-admin
    command="${admin.command}"
    user="${admin.user}"
    passwordfile="${admin.password.file}"
    host="${admin.host}"
    port="${admin.port}"
    asinstalldir="${j2ee.home}" />
</target>

<target name="create-jdbc-resource_common">
  <antcall target="admin_command_common">
    <param name="admin.command"
      value="create-jdbc-resource
--connectionpoolid ${conpool.name} ${jdbc.resource.name}" />
  </antcall>
</target>

<target name="delete-jdbc-resource_common">
  <antcall target="admin_command_common">
    <param name="admin.command"
      value="delete-jdbc-resource ${jdbc.resource.name}" />
  </antcall>
</target>

<target name="deploy-war">
  <antcall target="admin_command_common">
    <param name="admin.command"
      value="deploy ${war.file}" />
  </antcall>
</target>

<target name="undeploy-war">
  <antcall target="admin_command_common">
    <param name="admin.command"
      value="undeploy ${example}" />
  </antcall>
</target>

<property environment="env" />

<target name="listprops"
  description="Displays values of some of the properties of this
build file">
  <property file="../../common/admin-password.txt" />

  <echo message="Path information" />
</target>
```

```
<echo message="j2ee.home = ${j2ee.home}" />
<echo message="j2ee.tutorial.home = ${j2ee.tutorial.home}" />
<echo message="j2ee.home = ${j2ee.home}" />
<echo message="env.Path = ${env.Path}" />
<echo message="env.PATH = ${env.PATH}" />
<echo message="" />
<echo message="Classpath information" />
<echo message="classpath = ${env.CLASSPATH}" />
<echo message="" />
<echo message="Admin information" />
<echo message="admin.password = ${AS_ADMIN_PASSWORD}" />
<echo message="admin.password.file = ${admin.password.file}" />
<echo message="admin.host = ${admin.host}" />
<echo message="admin.user = ${admin.user}" />
<echo message="admin.port = ${admin.port}" />
<echo message="https.port = ${https.port}" />
<echo message="" />
<echo message="Domain information" />
<echo message="domain.resources = ${domain.resources}" />
<echo message="domain.resources.port = ${domain.resources.port}"
/>

<echo message="" />
<echo message="Database information" />
<echo message="db.root = ${db.root}" />
<echo message="db.driver = ${db.driver}" />
<echo message="db.host = ${db.host}" />
<echo message="db.port = ${db.port}" />
<echo message="db.sid = ${db.sid}" />
<echo message="db.url = ${db.url}" />
<echo message="db.user = ${db.user}" />
<echo message="db.pwd = ${db.pwd}" />
<echo message="url.prop = ${url.prop}" />
<echo message="ds.class = ${ds.class}" />
<echo message="db.jvmargs = ${db.jvmargs}" />
</target>
```

### **Nama file: build.properties (pada direktori commonweb)**

```
#war
build=build
sql.script=books.sql
war.file=${example}.war
assemble=assemble
assemble.war=${assemble}/war
```

### **Nama file: targets.xml (pada direktori commonweb)**

```
<target name="prepare" depends="init"
  description="Create build directories.">
  <mkdir dir="${build}" />
</target>

<target name="copy" depends="prepare"
```

```
description="Copy HTML and JSP pages" >
<copy todir="${build}">
  <fileset dir="web">
    <include name="**/*.html" />
    <include name="**/*.jsp" />
    <include name="**/*.jspx" />
    <include name="**/*.jspx" />
    <include name="**/*.gif" />
    <include name="**/*.xml" />
    <include name="**/*.tld" />
    <include name="**/*.tag" />
    <include name="**/*.jpg" />
    <include name="**/*.css" />
  </fileset>
</copy>
</target>

<target name="create-war" depends="build"
  description="Packages the WAR file">
  <echo message="Creating the WAR...."/>
  <delete file="${assemble.war}/${war.file}" />
  <delete dir="${assemble.war}/WEB-INF" />
  <copy todir="${assemble.war}/WEB-INF">
    <fileset dir=".">
      <include name="*.xml" />
      <exclude name="build.xml" />
      <exclude name="web.xml" />
    </fileset>
  </copy>
  <copy todir="${assemble.war}/WEB-INF/classes/">
    <fileset dir="${build}">
      <include name="**/*.class" />
    </fileset>
  </copy>
  <copy todir="${assemble.war}/WEB-INF/tags">
    <fileset dir="${build}">
      <include name="*.tag" />
    </fileset>
  </copy>
  <copy todir="${assemble.war}/WEB-INF">
    <fileset dir="${build}">
      <include name="*.tld" />
    </fileset>
  </copy>
  <copy todir="${assemble.war}">
    <fileset dir="${build}">
      <include name="*.jsp" />
      <include name="*.gif" />
    </fileset>
  </copy>
  <war destfile="${assemble.war}/${war.file}"
    webxml="./web.xml" filesonly="true" >
    <fileset dir="${assemble.war}" includes="WEB-INF/**, *.jsp,
*.gif" />
  </war>
  <copy file="${assemble.war}/${war.file}" todir="." />
</target>
```

```
<target name="copy-clock" depends="build" if="clock.exists"
  description="Copies clock class for bookstore2">
  <copy file="{build}/clock/DigitalClock.class"
    todir="{assemble.war}/clock/" />
</target>

<target name="listprops-web" depends="init,listprops"
  description="List property values">
  <echo message="sql.script = {sql.script}"/>
  <echo message="conpool.name = {conpool.name}"/>
  <echo message="jdbc.resource.name = {jdbc.resource.name}"/>
</target>
```

### Nama file: build.xml (dalam direktori HelloJSP)

```
<!DOCTYPE project [
  <!ENTITY targets SYSTEM "../common/targets.xml">
  <!ENTITY webtargets SYSTEM "../commonweb/targets.xml">
]>

<project name="hello" default="build" basedir=".">
  <target name="init">
    <tstamp/>
  </target>

  <!-- Configure the context path for this application -->
  <property name="example" value="hello" />

  <!-- Configure properties -->
  <property file="../common/build.properties"/>
  <property file="../commonweb/build.properties"/>

  &targets;
  &webtargets;

  <target name="build" depends="copy"
    description="Compile app Java files" >
    <javac srcdir="src" destdir="{build}">
      <include name="**/*.java" />
      <classpath refid="classpath"/>
    </javac>
  </target>

</project>
```

### Nama file: web.xml (dalam direktori HelloJSP)

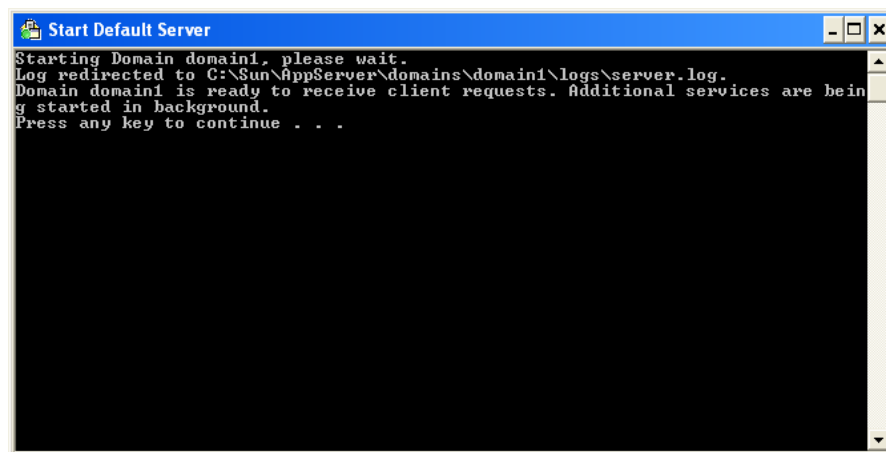
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" version="2.4"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>Hello</display-name>
</web-app>
```



```
</td>
</tr>
<tr>
<td colspan="2">
<input type="submit" name="enter" value="    Tampilkan    "><br/>
</form>
</td>
</tr>
</table>
<br/>
<p align='center'>
<b>
<jsp:useBean id="helloJSP" scope="application"
class="hello.jsp.HelloJSP"/>
<jsp:setProperty name="helloJSP" property="word" param="word"/>
<c:if test="${(not empty param.word)}">
    ${helloJSP.word}
</c:if>
</b>
</p>
</body>
</html>
```

### c. Menjalankan JSP

1. Buka Start->All Programs->Sun Microsystems->Application Server PE->Start Default Server sampai keluar command prompt



Tekan enter.

Jika gagal start maka start server secara manual, buka command prompt masuk ke c:/ dan ketik perintah:

C:/Sun/AppServer/domains/domain1/bin/startserv.bat [tekan enter]

Kemudian tekan ctrl+c, kemudian ketik y [tekan enter] setelah muncul pilihan.

Cek server jalan atau tidak di web browser dengan alamat:

`http://localhost:8080/`

Jika server sudah jalan akan ada tampilan web server running.

2. Buka jendela command prompt, masuk ke direktori HelloJSP
3. ketik perintah: `asant build`. Perintah `asant build` digunakan untuk mengkompilasi *source code*, jika *source code* telah lulus kompilasi maka akan muncul pesan sebagai berikut:

```
Buildfile: build.xml
```

```
init:
```

```
prepare:
```

```
copy:
```

```
build:
```

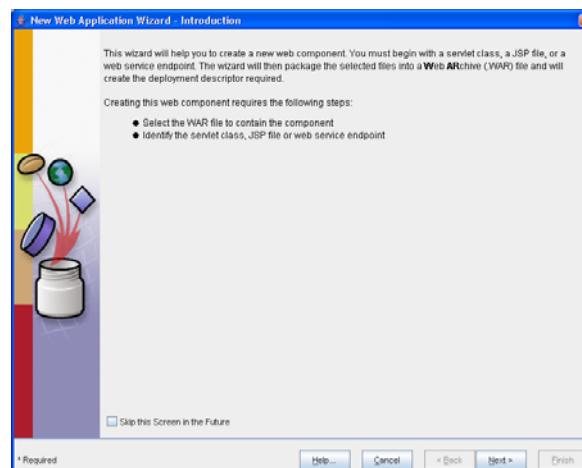
```
BUILD SUCCESSFUL
```

```
Total time: 2 seconds
```

Jika perintah `asant build` tidak dikenali, ketik perintah sebagai berikut:

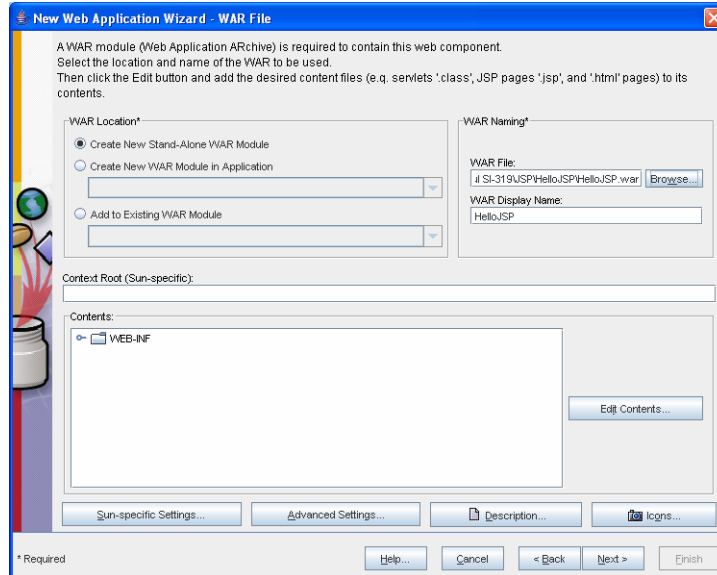
```
path=C:/Sun/AppServer/bin;%path%
```

4. Buka Deploytool dari Sun untuk melakukan *packaging* menjadi file WAR. Klik Start->All Program->Sun Microsystems->Application Server PE->Deploytool.
5. Klik File->New->Web Component... . Hingga muncul jendela seperti pada gambar berikut.

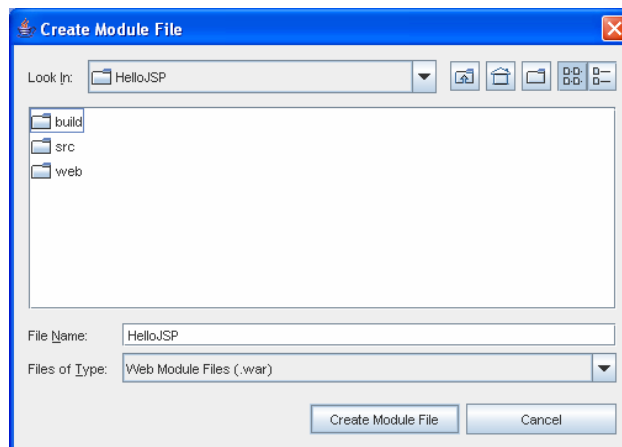




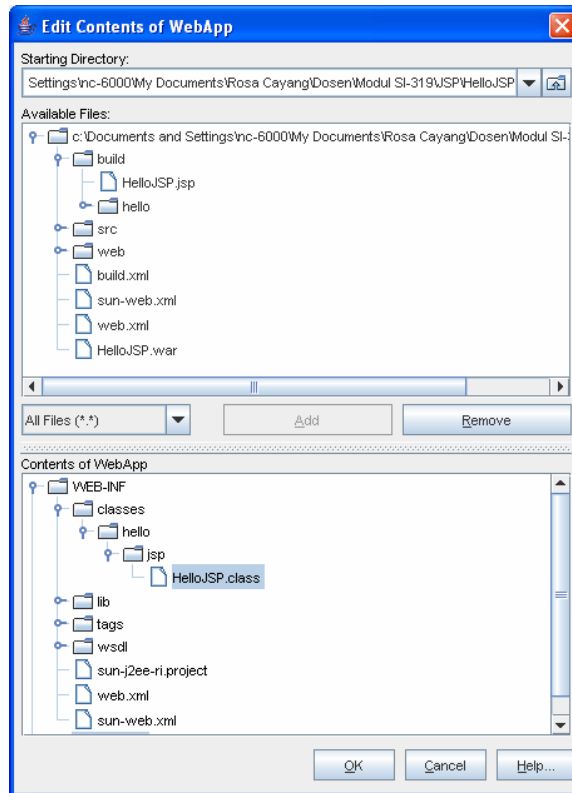
6. Klik Next. Hingga muncul jendela seperti pada gambar berikut.



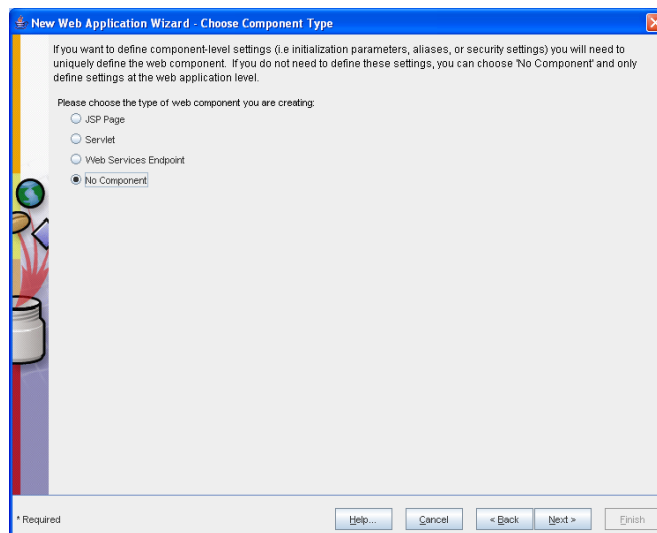
Pilih Create New Stand-Alone WAR Module, isi *field* WAR file dengan mengklik browse untuk mencari direktori HelloJSP, dan isi file name dengan HelloJSP seperti pada gambar berikut dan klik create module file.



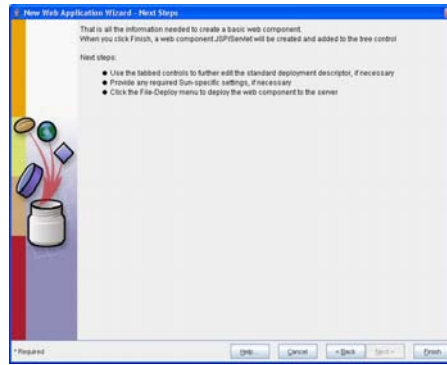
Klik edit contents (gambar sebelumnya) hingga muncul jendela seperti gambar berikut. Tambahkan direktori contoh pada available file dengan mengklik direktori build, kemudian klik direktori hello dan HelloJSP.jsp dan klik tombol add kemudian klik ok. Kemudian klik next.



7. Pada jendela seperti gambar berikut, pilih No Component. Klik next.

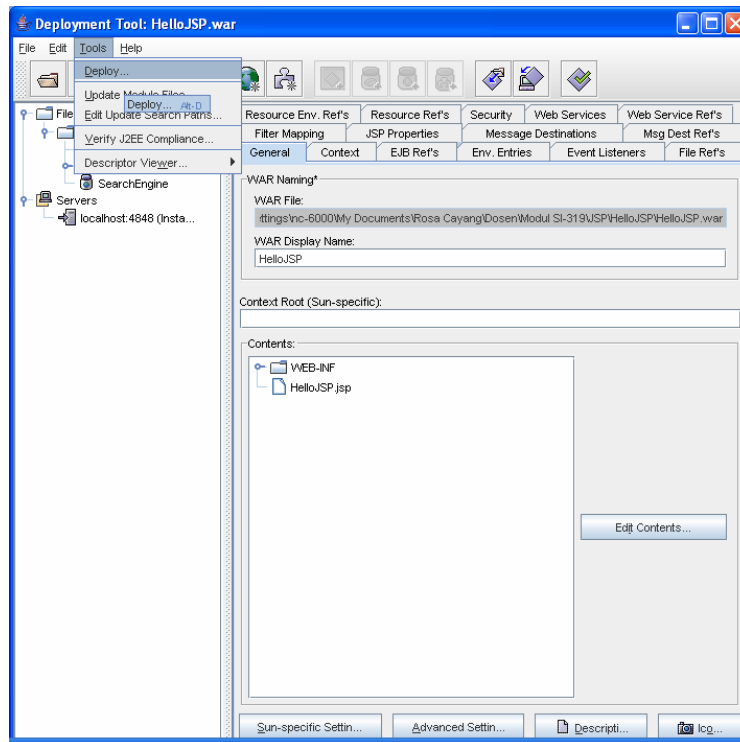


8. Setelah muncul jendela seperti pada gambar berikut, klik Finish.

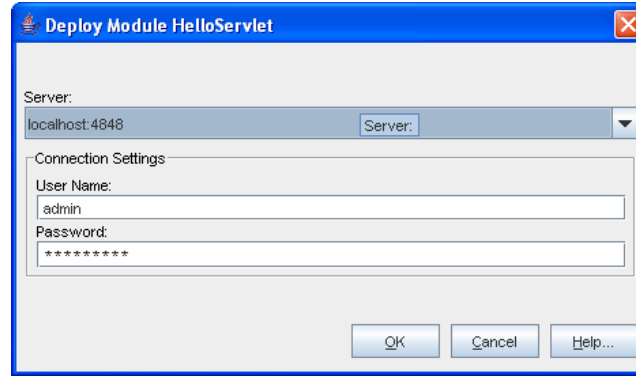


9. Setelah kembali ke jendela awal, klik tombol save yang bergambar disket, maka file WAR akan terbentuk di dalam direktori HelloJSP.

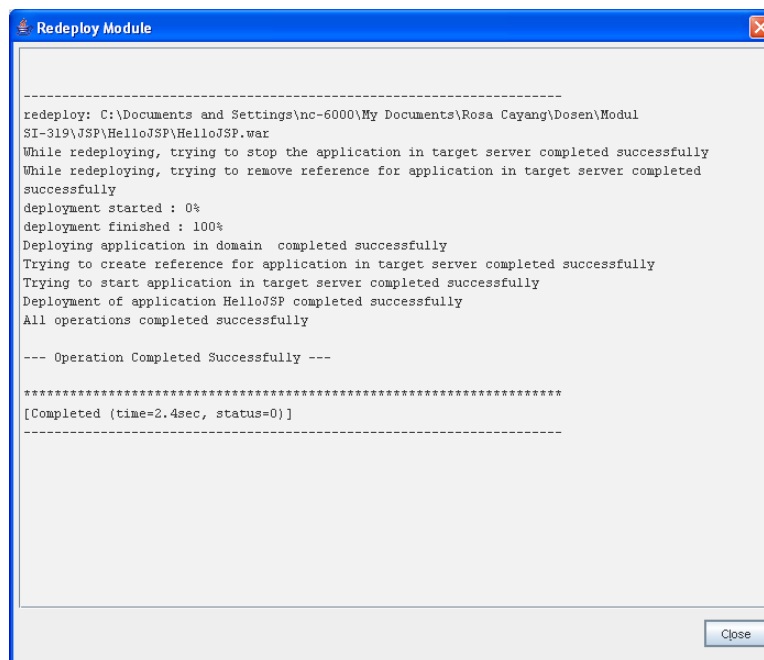
10. Pilih Tools->Deploy seperti pada gambar berikut:



11. Hingga muncul jendela berikut, lalu klik ok



12. Hingga muncul jendela berikut, klik close:



ketikkan `http://localhost:8080/HelloJSP/`

jika tampilan bukan halaman JSP maka cek port mungkin salah dengan membuka file `C:/Sun/AppServer/domains/domain1/config/domain.xml`

cek pada bagian file

```
<http-service>
  <http-listener acceptor-threads="5" address="0.0.0.0" blocking-
enabled="false" default-virtual-server="server" enabled="true"
family="inet" id="http-listener-1" port="1309" security-enabled="false"
server-name="" xpowered-by="true">
  </http-listener>
```

Untuk contoh bagian file di atas berarti port yang digunakan bukan 8080 tapi 1309

