

JENI 3 – 02 Modul Praktikum Servlet

1. Tujuan

- Membuat Servlet Hello World
- Menangani *Request* dan *Response*



2. Latar Belakang

Pada modul praktikum ini, anda akan mulai mencoba membuat Servlet Hello World. Selain sebuah web server, pada bagian ini, anda juga membutuhkan web browser (Internet Explorer, Mozilla Firefox, dll). Seperti halnya pada saat pertama anda mencoba bahasa pemrograman selain JAVA, disini anda akan dituntun bagaimana cara menampilkan tulisan Hello World pada Web Browser yang sudah terinstal di komputer anda, mulai dari file-file yang harus dikonfigurasi terlebih dahulu, hingga program yang anda harus tulis.

Pada saat Hello World telah sukses ditampilkan, selanjutnya anda akan mencoba bagaimana penanganan dari proses Request dan Response.

3. Percobaan

Percobaan 1 – Menyiapkan Aplikasi Web Java(Servlet) tanpa IDE :

 Info	Section ini merupakan kelanjutan dari Modul Praktikum 01 Percobaan 5 tentang Menyiapkan aplikasi web minimal.
 Tip	Anda bisa mengulangi kembali Modul Praktikum 01 Percobaan 5 atau menggunakan hasil Modul Praktikum tersebut untuk aktivitas ini.

Langkah berikutnya:

Kemudian untuk melengkapi definisi task pada file **build.xml** agar Ant bisa membantu developer dalam proses kompilasi dan pembuatan distribusi, maka kita tambahkan konfigurasi dan target-target lainnya.

JENI 3 – 02 Modul Praktikum Servlet

- Masih dalam folder **webStandard**, buat file konfigurasi **tomcat.properties**. Dalam file ini berisi konfigurasi untuk URL Tomcat Server.

```
# Properties for Tomcat Server
tomcat.server=localhost
tomcat.manager.url=http://${tomcat.server}:8080/manager
tomcat.username=root
tomcat.password=password
```

- Buat file konfigurasi **tomcatTasks.properties**. Dalam file ini berisi definisi kelas pada catalina-ant.jar untuk task spesifik.

```
deploy=org.apache.catalina.ant.DeployTask
install=org.apache.catalina.ant.InstallTask
list=org.apache.catalina.ant.ListTask
reload=org.apache.catalina.ant.ReloadTask
remove=org.apache.catalina.ant.RemoveTask
resources=org.apache.catalina.ant.ResourcesTask
roles=org.apache.catalina.ant.RolesTask
start=org.apache.catalina.ant.StartTask
stop=org.apache.catalina.ant.StopTask
undeploy=org.apache.catalina.ant.UndeployTask
```

- Modifikasi file **build.xml** dengan menambahkan target-target baru, sehingga isi file build.xml menjadi seperti berikut:

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="antTasks" default="init" basedir=".">
  <description>Compile, Test and Deploy a Project.</description>
  <property file="tomcat.properties"/>
  <property file="build.properties"/>

  <property name="locale.default" value="en"/>
  <property name="debug" value="on"/>
  <property name="optimize" value="on"/>
  <property name="deprecation" value="off"/>
  <!-- ===== -->
  <!-- Builds Classpath -->
  <!-- ===== -->
  <path id="classpath">
    <fileset dir="${webapp.lib.dir}">
      <include name="*.jar"/>
    </fileset>

    <fileset dir="${tomcat.home}/lib">
      <include name="*.jar"/>
    </fileset>
  </path>

  <taskdef file="tomcatTasks.properties">
```

JENI 3 – 02 Modul Praktikum Servlet

```
<classpath>
    <pathelement path="${tomcat.home}/lib/catalina-ant.jar"/>
</classpath>
</taskdef>

<target name="init" description="Initialize ....">
    <tstamp/>
    <echo message="Initialize ..."/>
    <echo message="Tomcat Home = ${tomcat.home}"/>
    <echo message="Build Directory = ${webapp.build.dir}"/>
    <echo message="Source Java Directory = ${webapp.src.java.dir}"/>
</target>

<target name="create-web-standard-folder" description="Create a standard
web directory for Java application">
    <echo message="Creating standard directory for Web Based Java
Application..."/>
    <mkdir dir="${webapp.build.web.lib.dir}"/>
    <mkdir dir="${webapp.build.classes.dir}"/>
    <mkdir dir="${webapp.dist.dir}"/>
    <mkdir dir="${webapp.src.java.dir}"/>
    <mkdir dir="${webapp.src.web.dir}"/>
    <mkdir dir="${webapp.lib.dir}"/>
</target>

<target name="prepare-dir" depends="init">
    <echo message="Creating directory ..."/>
    <mkdir dir="${webapp.build.dir}"/>
    <mkdir dir="${webapp.build.web.dir}"/>
    <mkdir dir="${webapp.build.web.lib.dir}"/>
    <mkdir dir="${webapp.build.classes.dir}"/>
    <mkdir dir="${webapp.dist.dir}"/>
</target>

<target name="compile" depends="prepare-dir" description="Compiling all the
code ....">
    <echo message="Compiling ..."/>
    <javac
        classpathref="classpath"
        srcdir="${webapp.src.java.dir}/"
        destdir="${webapp.build.classes.dir}"
        debug="${debug}"
        deprecation="${deprecation}"
        optimize="${optimize}"
    </javac>
</target>

<!-- ===== -->
<!-- Cleans up the distribution -->
<!-- ===== -->
<target name="clean" description="Cleans up the distribution" >
    <echo message="Clean out the output directories"/>
```

JENI 3 – 02 Modul Praktikum Servlet

```
<delete quiet="true" includeEmptyDirs="true" failonerror="false">
  <fileset dir="${webapp.build.dir}" />
  <fileset dir="${webapp.dist.dir}" />
</delete>
</target>

<target name="deploy" description="Install application in Tomcat"
depends="create-war">
  <deploy url="${tomcat.manager.url}"
    username="${tomcat.username}"
    password="${tomcat.password}"
    path="/${webapp.name}"
    war="file:///${webapp.dist.dir}/${webapp.war}"/>
</target>

<target name="deploy-local" description="Install application in Tomcat
webbapp" depends="build">
  <copy todir="${tomcat.home}/webapps/${project.name}">
    <fileset dir="${webapp.build.web.dir}" />
  </copy>
<!--
  http://localhost:8080/manager/html/deploy?deployPath=&deployConfig=&dep
  loyWar=
  -->
</target>

<target name="undeploy" description="Remove application in Tomcat">
  <undeploy url="${tomcat.manager.url}"
    username="${tomcat.username}"
    password="${tomcat.password}"
    path="/${webapp.name}"/>
</target>

<target name="reload" description="Reload application in Tomcat">
  <reload url="${tomcat.manager.url}"
    username="${tomcat.username}"
    password="${tomcat.password}"
    path="/${webapp.name}"/>
</target>

<target name="start" description="Start Tomcat application">
  <!--<start url="${tomcat.manager.url}"
    username="${tomcat.username}"
    password="${tomcat.password}"
    path="/${webapp.name}"/>
  -->
  <java jar="${tomcat.home}/bin/bootstrap.jar" fork="true">
    <jvmarg value="-Dcatalina.home=${tomcat.home}"/>
  </java>
</target>
```

JENI 3 – 02 Modul Praktikum Servlet

```
<target name="stop" description="Stop Tomcat application">
  <!--      <stop url="${tomcat.manager.url}"
            username="${tomcat.username}"
            password="${tomcat.password}"
            path="/${webapp.name}"/>
        -->
  <java jar="${tomcat.home}/bin/bootstrap.jar" fork="true">
    <jvmarg value="-Dcatalina.home=${tomcat.home}"/>
    <arg line="stop"/>
  </java>
</target>

<target name="list" description="List Tomcat applications">
  <list url="${tomcat.manager.url}"
        username="${tomcat.username}"
        password="${tomcat.password}"/>
</target>

<target name="build" description="Building application" depends="compile,
copy-web">
  <echo message="Build the application"/>
</target>

<target name="copy-web" description="Building application"
depends="prepare-dir">
  <copy todir="${webapp.build.web.dir}">
    <fileset dir="${webapp.src.web.dir}"/>
  </copy>
</target>

<target name="create-jar" description="Create JAR" depends="build">
  <!-- Create Jar -->
  <jar jarfile="${webapp.build.dir}/${project.name}-
${project.version}.jar"
      basedir="${webapp.build.dir}"/>
</target>

<target name="create-war" description="Packaging WAR archive"
depends="build">
  <echo message="Packaging WAR archive"/>
  <war destfile="${webapp.dist.dir}/${webapp.war}"
      webxml="${webapp.build.web.dir}/WEB-INF/web.xml"
      basedir="${webapp.build.web.dir}">
  </war>
  <!--
  <war destfile="login.war" webxml="${src}/login.xml">
    <fileset dir="${src}/html"/>
    <classes dir="${bin}"/>
  </war>
  -->
</target>
</project>
```

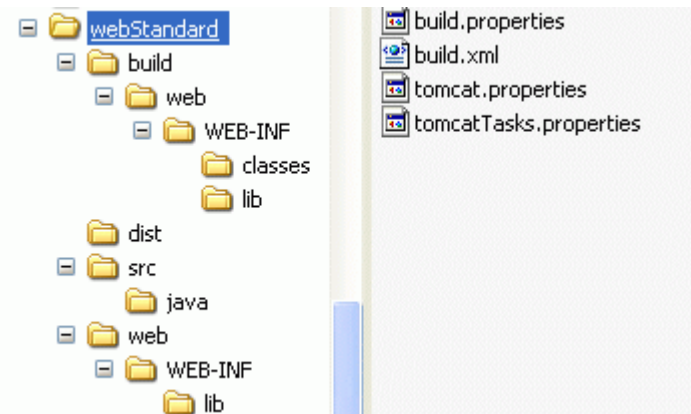
JENI 3 – 02 Modul Praktikum Servlet

- Buat file web.xml dan letakkan dalam folder web/WEB-INF yang berisi kode berikut:

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  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"
  version="2.4">
</web-app>
```

- Sehingga susunan akhir folder webStandard terlihat seperti berikut:



Catatan: folder **build** dan **dist** terbuat secara otomatis oleh Ant.

Sampai disini, Ant bisa digunakan untuk membuild project dan manajemen Tomcat.

Penggunaannya Build Tool Apache Ant dalam Project:

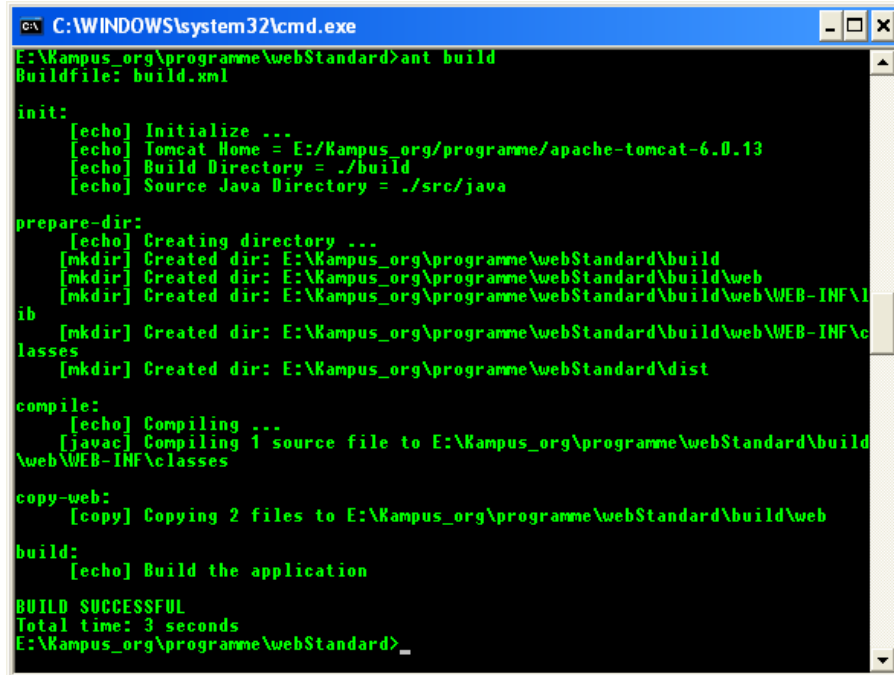
1. Mem-build aplikasi

Target build akan menjalankan beberapa target lainnya yaitu compile dan copy-web. Target ini akan melakukan kompilasi kode Java menjadi class dan mereplikasi resource web ke dalam folder build. Perintahnya: **ant build**

JENI 3 – 02 Modul Praktikum Servlet

2. Membuat distribusi/ release aplikasi

Setelah proses build dijalankan, developer yang ingin menjadikan aplikasinya menjadi file WAR bisa menjalankan target create-war. Hal ini akan membungkus file-file yang ada di folder build/web ke dalam file WAR dan meletakkan file WAR di dalam folder dist. Perintahnya: **ant create-war**



```
C:\WINDOWS\system32\cmd.exe
E:\Kampus_org\programme\webStandard>ant build
Buildfile: build.xml

init:
[echo] Initialize ...
[echo] Tomcat Home = E:/Kampus_org/programme/apache-tomcat-6.0.13
[echo] Build Directory = ./build
[echo] Source Java Directory = ./src/java

prepare-dir:
[echo] Creating directory ...
[mkdir] Created dir: E:\Kampus_org\programme\webStandard\build
[mkdir] Created dir: E:\Kampus_org\programme\webStandard\build\web
[mkdir] Created dir: E:\Kampus_org\programme\webStandard\build\web\WEB-INF\classes
[echo] Build the application

BUILD SUCCESSFUL
Total time: 3 seconds
E:\Kampus_org\programme\webStandard>
```

3. Mendeploy aplikasi

Setelah proses build dan pembuatan distribusi aplikasi, selanjutnya aplikasi bisa dideploy ke server container.

Melalui web dapat dilakukan dengan cara masuk ke Tomcat Manager dan gunakan Form Upload War file atau Form Deploy aplikasi pada tomcat.

Dengan Apache ant dapat pula dilakukan dengan mengintegrasikan apache ant dengan tomcat task. Kemudian membuat target. Pada contoh file build.xml di atas, kita bisa mendeploy aplikasi dengan perintah **ant deploy**.

4. Meng-undeploy aplikasi

Seperti halnya mendeploy aplikasi, undeploy bisa dilakukan melalui Tomcat Manager ataupun Apache Ant task (**ant undeploy**).

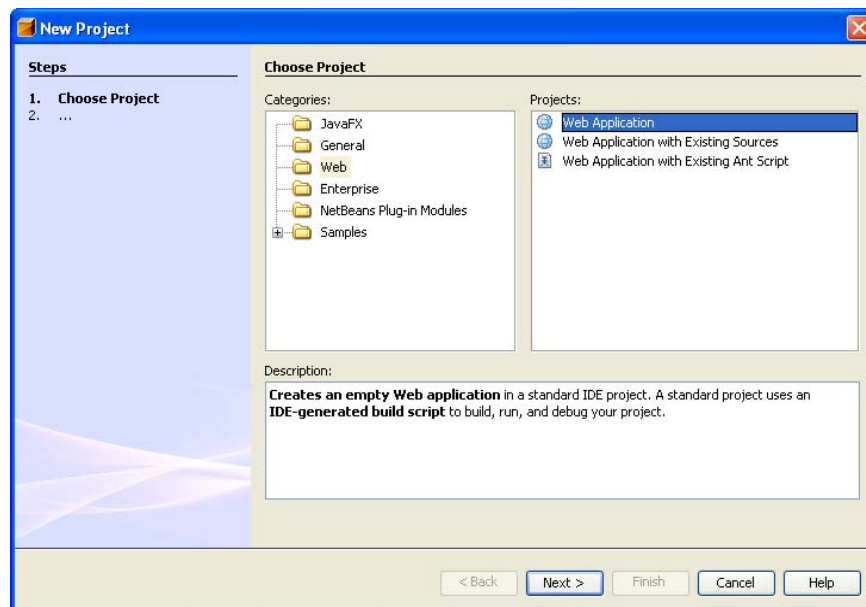
JENI 3 – 02 Modul Praktikum Servlet

Percobaan 2 – Menyiapkan Aplikasi Web Java(Servlet) dengan Netbeans IDE :

Dengan menggunakan Netbeans IDE, developer tidak perlu susah-susah menyiapkan file build.xml dan file-file lainnya, karena Netbeans telah menyederhanakan proses ini atau bahkan menghilangkan proses ini. Developer cukup dengan membuat project web, maka langkah-langkah pada Modul Praktikum 01 Percobaan 5 dan Modul Praktikum 02 Percobaan 1 sudah tercover dengan baik.

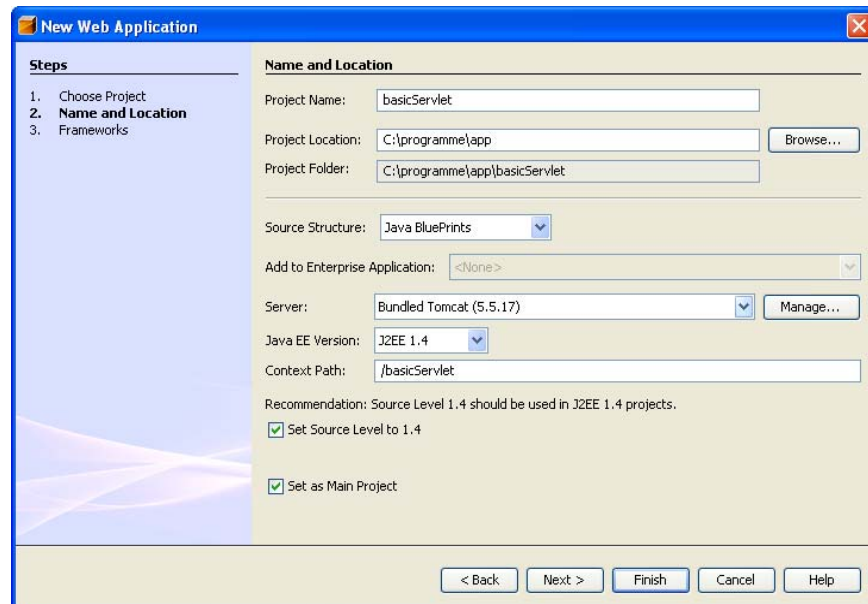
Langkah-langkahnya:

- Buat Project Baru, klik File → New Project

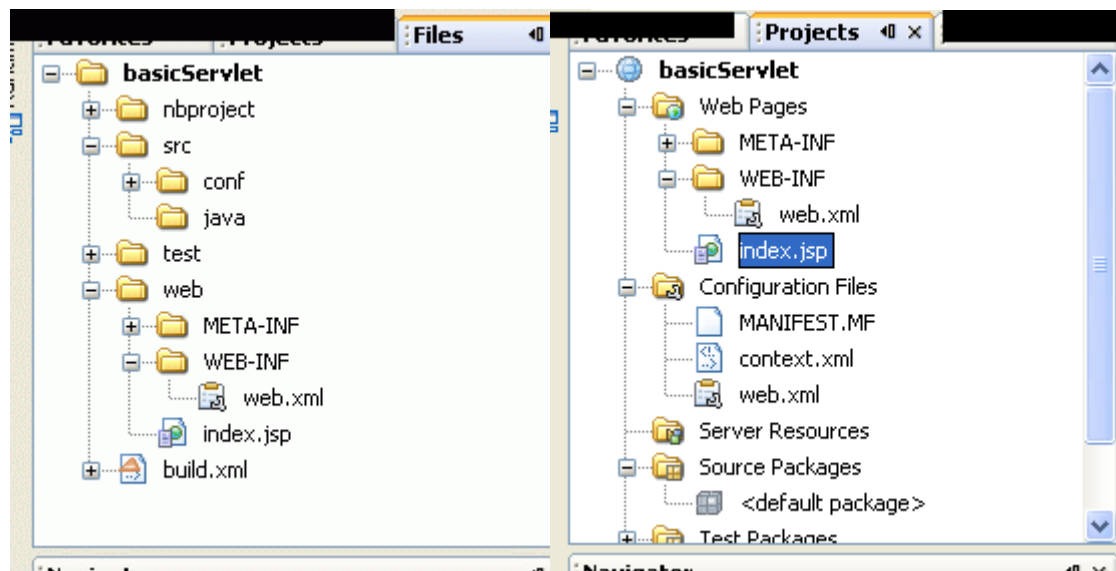


- Pilih category web → Web Application, klik Next
- Ketikkan nama project, misal **basicServlet**. Biarkan semuanya default dan klik Finish

JENI 3 – 02 Modul Praktikum Servlet



Maka Netbeans akan membuat susunan directori seperti gambar berikut (dilihat melalui tab **Files** dan tab **Projects**):



Pada susunan directori project Netbeans, terdapat pula file **build.xml** dan **build-impl.xml** yang siap digunakan untuk mem-*build* project.

JENI 3 – 02 Modul Praktikum Servlet



Tip

Selalu gunakan tab Projects selama *development*, untuk memudahkan pengaturan file dan konfigurasi.



Info

Untuk latihan-latihan berikutnya kita menggunakan Netbeans IDE

JENI 3 – 02 Modul Praktikum Servlet

Percobaan 3 – Membuat Servlet Pertama :



Info

Untuk efisiensi, kita akan menggunakan project sebelumnya "basicServlet" (Percobaan 2)



Info

Tujuan dari section ini adalah menunjukkan bagaimana cara membuat servlet sederhana. Untuk servlet pertama ini kita buat dari class Java kosong kemudian kita tambahkan syarat-syarat class Java sehingga bisa disebut sebagai Servlet.

Langkah-langkahnya:

- Buat package **jeni3.servlet**.



Tip

Klik kanan Source Packages → pilih New → pilih Java Package.

- Pada package **jeni3.servlet**, buat class Java dengan nama **HelloServlet**.



Tip

Pada Source Packages klik kanan package **jeni3.servlet** → pilih New → pilih Java Class.

- Lengkapi kode class HelloServlet menjadi seperti berikut:

```
package jeni3.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * @author mee_andto@yahoo.com
 * @version 0.5
 */
public class HelloServlet extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse response)
```

JENI 3 – 02 Modul Praktikum Servlet

```
throws ServletException, IOException
{
    // "request" is used for reading incoming HTTP headers
    // HTML form data (e.g. data the user entered and submitted)
    // and other data that can be inferred from the client request.

    // "response" is for specifying the HTTP response line and
    // headers(e.g. specifying the content type, setting cookies).
    // It also contains methods that allow the servlet to generate
    // responses for the client.

    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Hello Servlet</title></head>");
    out.println("<body>");
    out.println("<h1>Hello Servlet!!!</h1>");
    out.println("</body>");
    out.println("</html>");
}
}
```

- Buka file **web.xml** dan tambahkan mapping servlet berikut:

```
<servlet>
    <servlet-name>helloServlet</servlet-name>
    <servlet-class>jeni3.servlet.HelloServlet</servlet-class>
</servlet>

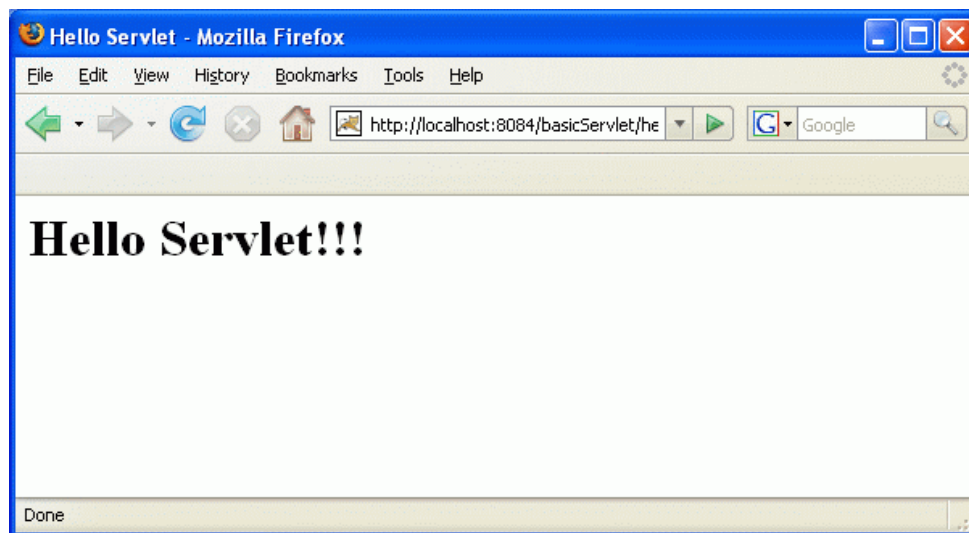
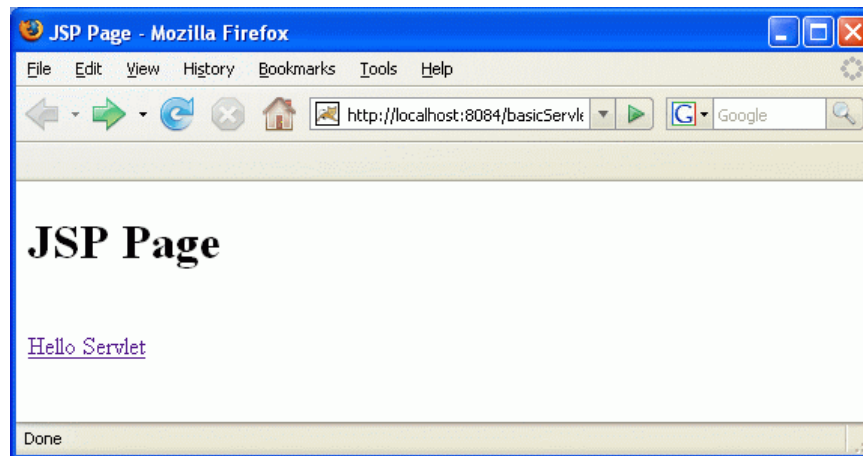
<servlet-mapping>
    <servlet-name>helloServlet</servlet-name>
    <url-pattern>/hello</url-pattern>
</servlet-mapping>
```

- Buka file **index.jsp** dan tambahkan link berikut:

```
<br/>
<a href="./hello">Hello Servlet</a>
```

JENI 3 – 02 Modul Praktikum Servlet

- Klik kanan nama project (basicServlet), pilih Run → Run Project.



Info

Syarat-syarat class Java disebut sebagai Servlet adalah:

Turunan dari class **javax.servlet.http.HttpServlet**

Memiliki fungsi penanganan request dengan parameter object dari class **javax.servlet.http.HttpServletRequest** dan object dari class **javax.servlett.http.HttpServletResponse**.

Fungsi-fungsi servlet antara lain init, doGet, doPost, doPut, doTrace, service, destroy.

JENI 3 – 02 Modul Praktikum Servlet

Percobaan 4 – Membuat Servlet Kedua :



Info

Melanjutkan section sebelumnya, untuk membuat servlet melalui menu Servlet pada Netbeans IDE



Info

Tujuan dari section ini adalah menunjukkan bagaimana cara membuat servlet sederhana menggunakan Netbeans IDE. Untuk servlet kedua ini kita buat dengan menggunakan menu Servlet pada Netbeans, sehingga Servlet telah memiliki syarat-syarat sebagai Servlet sesuai pada section sebelumnya.

Langkah-langkahnya:

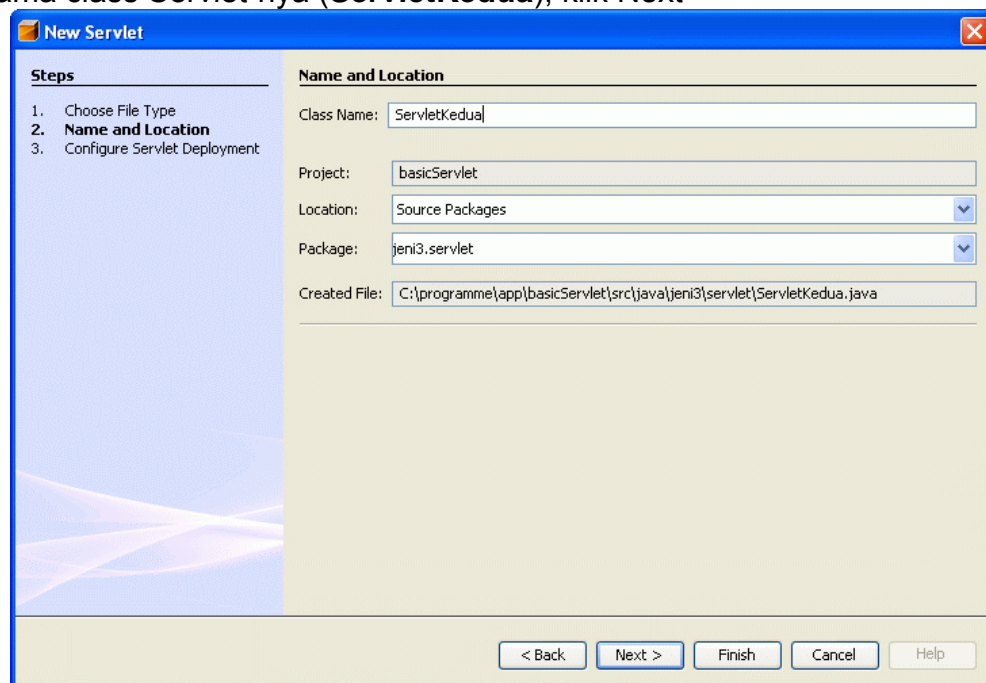
- Pada package **jeni3.servlet** buat Servlet dengan nama **ServletKedua**



Tip

Pada Source Packages klik kanan package **jeni3.servlet** → pilih New → pilih Servlet.

- Tuliskan nama class Servlet-nya (**ServletKedua**), klik Next



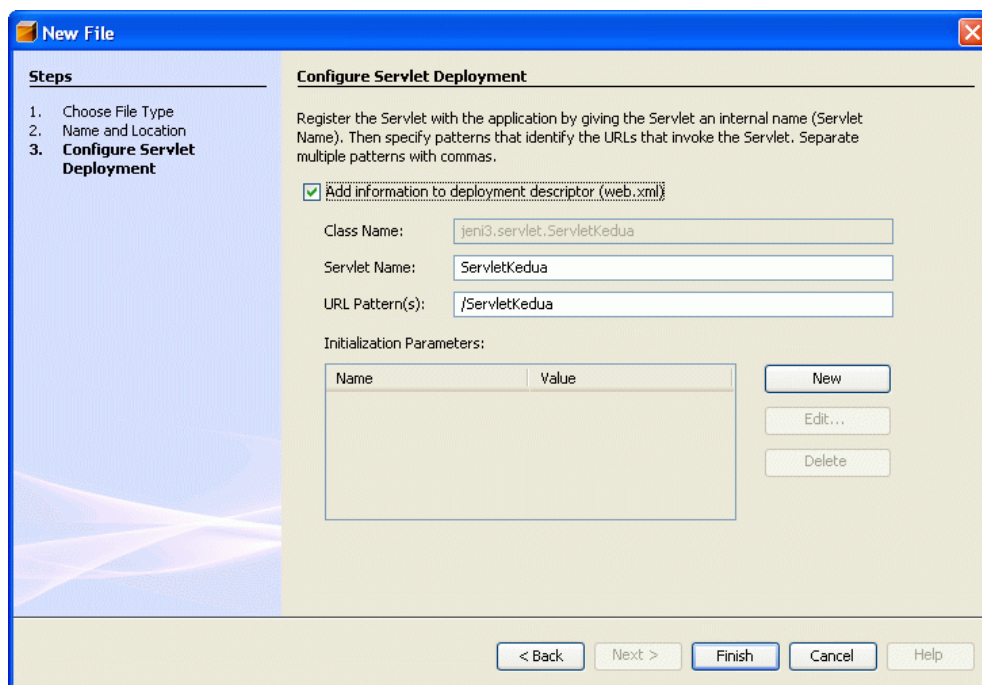
JENI 3 – 02 Modul Praktikum Servlet

Jika anda men-check option "Add information to deployment descriptor (web.xml)", maka ketika di-klik Finish, Netbeans secara otomatis akan menambahkan servlet mapping ke dalam file web.xml sesuai dengan isian pada wizard.

Jika option itu tidak di-check, maka developer harus menambahkan secara manual servlet mapping seperti berikut:

```
<servlet>
    <servlet-name>ServletKedua</servlet-name>
    <servlet-class>jeni3.servlet.ServletKedua</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>ServletKedua</servlet-name>
    <url-pattern>/ServletKedua</url-pattern>
</servlet-mapping>
```



Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name:

Servlet Name:

URL Pattern(s):

Initialization Parameters:

Name	Value
------	-------

New Edit... Delete

< Back Next > Finish Cancel Help

- Hilangkan komentar pada class ServletKedua pada fungsi processRequest:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    /** TODO output your page here
    out.println("<html>");
    out.println("<head>");
```


JENI 3 – 02 Modul Praktikum Servlet

```

out.println("<title>Servlet ServletKedua</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Servlet ServletKedua at " + request.getContextPath
() + "</h1>");
out.println("</body>");
out.println("</html>");
/**/
out.close();
}

```



Info

Pembuatan servlet melalui menu new → Servlet, secara default class itu menyediakan 2 fungsi penanganan request yaitu doGet dan doPost. Kemudian masing-masing fungsi tersebut memanggil fungsi processRequest. Dalam contoh di atas kita meng-override fungsi processRequest.

doGet digunakan untuk menangani request dengan metode GET. doPost digunakan untuk menangani request dengan metode POST.

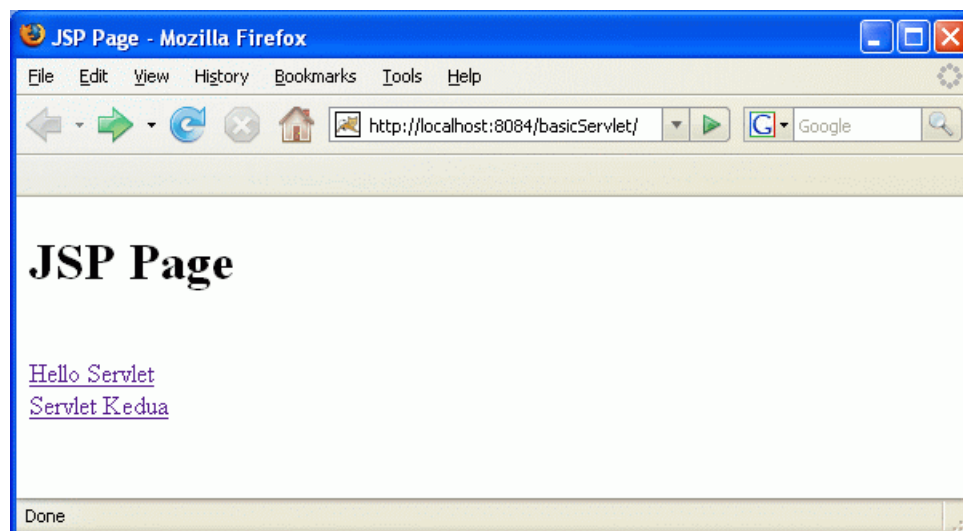
- Tambahkan link berikut di file index.jsp:

```

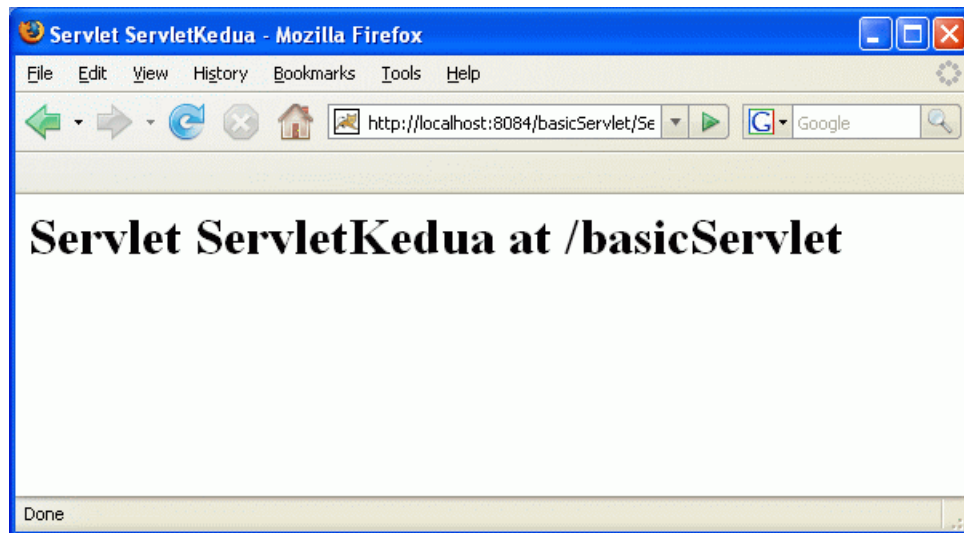
<br/>
<a href="./ServletKedua">Servlet Kedua</a>

```


- Run → Run Project




JENI 3 – 02 Modul Praktikum Servlet



Percobaan 5 – Penanganan Request dan Response :

 Info	Section ini merupakan kelanjutan dari Modul Praktikum 02 Percobaan 4 (sebelumnya). Sehingga kita bekerja masih dalam project "basicServlet"
---	---

Fungsi **GetParameter.**

 Info	Tujuan dari section ini adalah menunjukkan bagaimana cara menangani pengambilan sebuah data/ variabel dari sebuah request.
---	---

Langkah-langkahnya:

- Pada Web Pages buat file HTML dengan nama **form-data.html** dengan kode seperti berikut:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title></title>
  </head>
  <body>
    <form action="parameter" method="POST">
      Enter your name = <input type="text" name="username" value="" />
      <input type="submit" value="Send" name="send" />
    </form>
  </body>
</html>

```

JENI 3 – 02 Modul Praktikum Servlet

```
</form>
</body>
</html>
```



Perhatian

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

- Pada Source Packages pada package **jeni3.servlet** buat servlet dengan nama **GetParameterServlet** dengan kode seperti berikut:

```
package jeni3.servlet;

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

/**
 * @author mee_andto@yahoo.com
 * @version 0.5
 */
public class GetParameterServlet extends HttpServlet {
    /** 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 {
        // retrieve the value supplied by the user
        String userName = request.getParameter("username");

        // retrieve the PrintWriter object and use it to output the greeting
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>GetParameter Servlet</title></head>");
        out.println("<body>");
        out.println("<h1>");
        out.println("HELLO AND WELCOME, " + userName + "!");
        out.println("</h1>");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
}
```

JENI 3 – 02 Modul Praktikum Servlet

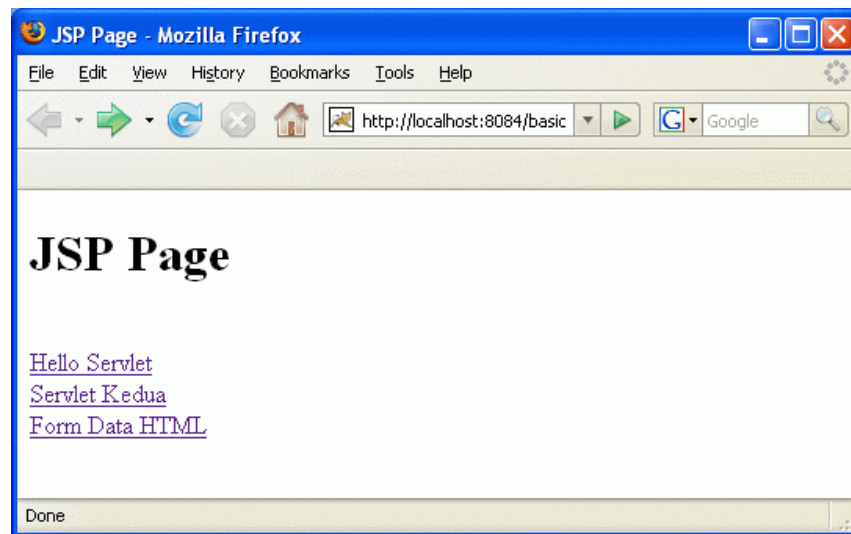
- Pastikan mapping untuk servlet GetParameterServlet terlihat seperti berikut:

```
<servlet>
  <servlet-name>GetParameterServlet</servlet-name>
  <servlet-class>jeni3.servlet.GetParameterServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>GetParameterServlet</servlet-name>
  <url-pattern>/parameter</url-pattern>
</servlet-mapping>
```

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

```
<br/>
<a href="./form-data.html">Form Data HTML</a>
```

- Run → Run Project



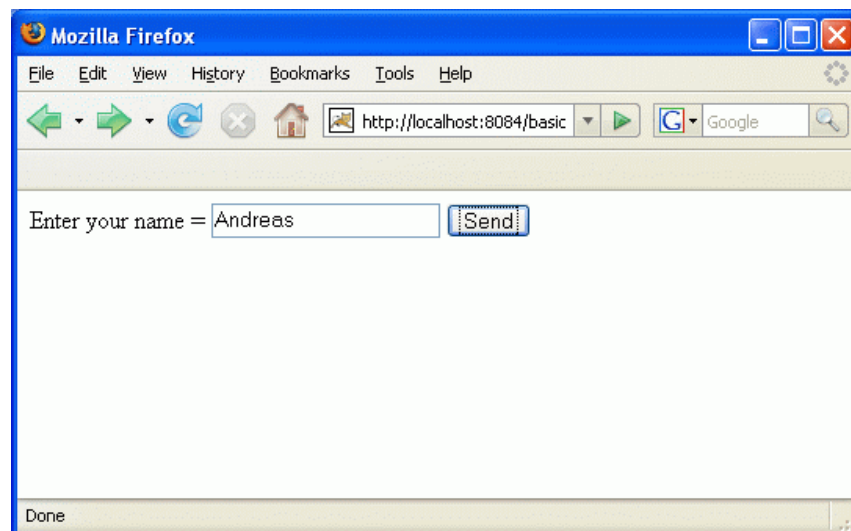
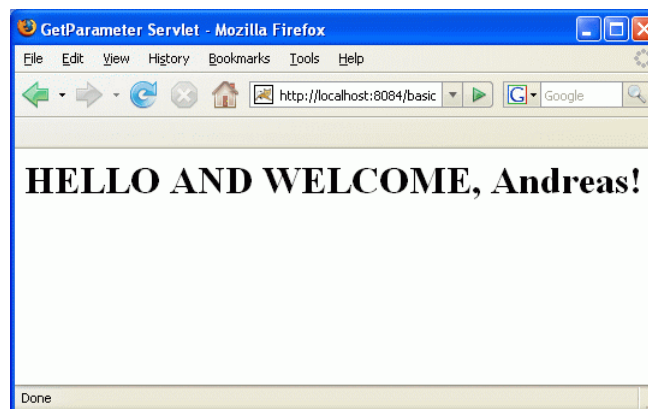
JENI 3 – 02 Modul Praktikum Servlet



Info

Jadi, untuk mengambil nilai dari suatu variabel dalam request digunakan fungsi `getParameter()`.

Inisialisasi isi response harus dilakukan sebelum pembuatan instance dari class `PrintWriter`.

Fungsi **getParameterValues**

JENI 3 – 02 Modul Praktikum Servlet



Info

Tujuan dari section ini adalah menunjukkan bagaimana cara menangani pengambilan data/ **variabel array** dari sebuah request/ form.

Langkah-langkahnya:

- Buat file HTML dengan nama **form-sports.html**, dengan kode seperti berikut:

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

<html>
  <head>
    <title>GetParameterValues Servlet</title>
  </head>
  <body>
    <h1>What sport activities do you perform?</h1>
    <form action="getParameterValuesServlet" method="POST">
      <input type="checkbox" name="sports" value="Biking" />Biking
    <br/>
      <input type="checkbox" name="sports" value="Table Tennis" />Table
      Tennis <br/>
      <input type="checkbox" name="sports" value="Swimming" />Swimming
    <br/>
      <input type="checkbox" name="sports" value="Basketball"
    />Basketball <br/>
      <input type="checkbox" name="sports" value="Others" />Others
    <br/>
      <input type="submit" value="Submit" name="submit" />
    </form>
  </body>
</html>
```



Perhatian

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

- Dalam package **jeni3.servlet**, buat servlet dengan nama **GetParameterValuesServlet** dengan kode seperti berikut:

```
package jeni3.servlet;

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

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

JENI 3 – 02 Modul Praktikum Servlet

```
* @version 0.5
*/
public class GetParameterValuesServlet extends HttpServlet {
    /** 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 {
        String paramValues[] = request.getParameterValues("sports");
        StringBuffer myResponse = new StringBuffer();
        myResponse.append("<html>");
        myResponse.append("<head><title>Your Choices</title></head>");
        myResponse.append("<body>");
        myResponse.append("<h1>Your choices were: </h1>");

        myResponse.append("<ul>");
        for (int i = 0; i < paramValues.length; i++) {
            myResponse.append("<li>" + paramValues[i] + "</li>");
        }
        myResponse.append("</ul>");
        myResponse.append("</body>");
        myResponse.append("</html>");

        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println(myResponse.toString());
    }
}
```

- Pastikan mapping servlet:

```
<servlet>
    <servlet-name>GetParameterValuesServlet</servlet-name>
    <servlet-class>jeni3.servlet.GetParameterValuesServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>GetParameterValuesServlet</servlet-name>
    <url-pattern>/getParameterValuesServlet</url-pattern>
</servlet-mapping>
```

- Tambahkan link dalam file index.jsp untuk mengakses form-sports.html:

```
<br/>
<a href="./form-sports.html">Form Sports HTML</a>
```

JENI 3 – 02 Modul Praktikum Servlet

- Run → Run Project

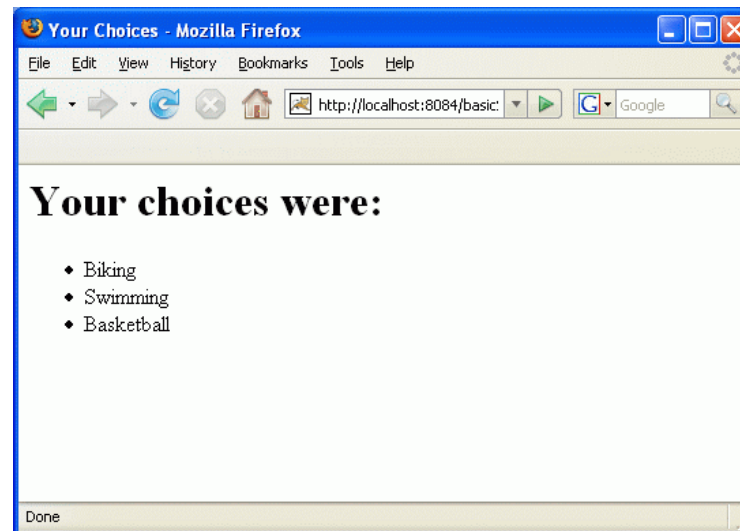


Info

Jadi, untuk mengambil nilai dari suatu variabel array dalam request digunakan fungsi `getParameterValues()`.



JENI 3 – 02 Modul Praktikum Servlet



JENI 3 – 02 Modul Praktikum Servlet

Percobaan 6 – Packaging Aplikasi Web :



Info

Section ini masih menggunakan project **basicServlet** sebelumnya.



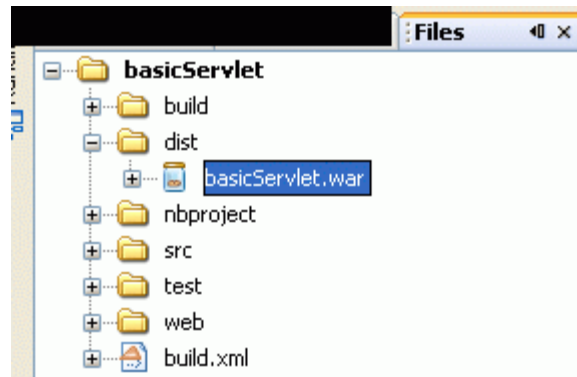
Info

Tujuan dari section ini adalah menunjukkan bagaimana membuat paket dari aplikasi yang telah kita. Tujuan dibuat paket adalah untuk memudahkan dalam instalasi ke server lain.



Info

Netbeans IDE telah mem-*package* aplikasi anda ketika anda membuild project (menu Build → Build Main Project atau klik kanan namaProject → Build Project). Hasilnya disimpan dalam folder **dist** dengan nama **namaProject.war**. Dalam section ini bernama **basicServlet.war**.



Untuk *packaging* secara manual, anda baca kembali Percobaan 1 pada Modul Praktikum ini pada subsection tentang **Penggunaan Build Tool Apache Ant dalam project**.



Info

Setelah di-*package*, aplikasi anda tinggal meng-*upload*-nya ke server lain. Untuk proses peng-*upload*-an yang menggunakan Tomcat sebagai web Container anda bisa mengikuti langkah-langkah pada Modul Praktikum 01 Percobaan 2: Jakarta Apache Tomcat pada subsection tentang **Tomcat Manager Tool**.