

6

Java Server Page

Objektif :

- Mengetahui apa itu JSP
 - Mengetahui kegunaan dan cara kerja JSP
 - Mampu membuat program JSP sederhana
-

6.1. Definisi JSP

Pada bab sebelumnya kita telah melihat bagaimana kekurangan serta kelebihan dari servlet serta penggunaannya. Namun seperti yang anda lihat pada pengimplentasiannya, cara tersebut memang bukan cara yang baik dan bagus dalam menggunakan servlet. Pada java untuk membuat tampilan ke user dengan konten dinamis umumnya menggunakan sebuah teknologi lain dan tersendiri. Teknologi tersebut bernama Java Server Page atau umum dikenal dan disingkat dengan JSP. Penggunaan dari servlet yang benar adalah menggunakannya sebagai controller dari request yang hendak dibuat. Untuk penerapan lebih jelasnya akan dibahas tersendiri pada bab terakhir.

JSP merupakan suatu teknologi web berbasis bahasa pemrograman Java dan berjalan di Platform Java, dan juga merupakan bagian teknologi J2EE (Java 2 Enterprise Edition). JSP merupakan bahasa scripting untuk web programming yang bersifat server side seperti halnya PHP dan ASP. JSP dapat berupa gabungan antara baris HTML dan fungsi-fungsi dari JSP itu sendiri. JSP pada pengolahannya di server akan diubah menjadi servlet kembali dan dikompilasi. Hal ini membuat pengoprasian JSP pada saat pertama kali akan sedikit lama, tetapi setelah kedua dan seterusnya akan lebih cepat dan memiliki performa yang lebih baik dibanding bahasa scripting seperti PHP. Seperti tipe aplikasi java lainnya (Servlet, Applet, Midlet dll), JSP juga bertipe Strong Type artinya penggunaan variable pada halaman tersebut harus dideklarasikan terlebih dahulu. Misalnya pada sintaks pengulangan berikut:

```
int i;
for (i=1; i<13; i++)
{
    // statement
}
```

Atau

```
for (int i=1; i<13; i++)
{
    // statement
}
```

Hal ini harap diperhatikan bagi para developer yang terbiasa dengan PHP yang tidak memerlukan deklarasi variable yang akan digunakan.

Keunggulan JSP

JSP merupakan dokumen text seperti HTML, yang menghindari format dan manipulasi yang memungkinkan String yang sangat panjang untuk menghasilkan output. Berikut merupakan beberapa kelebihan dari teknologi JSP:

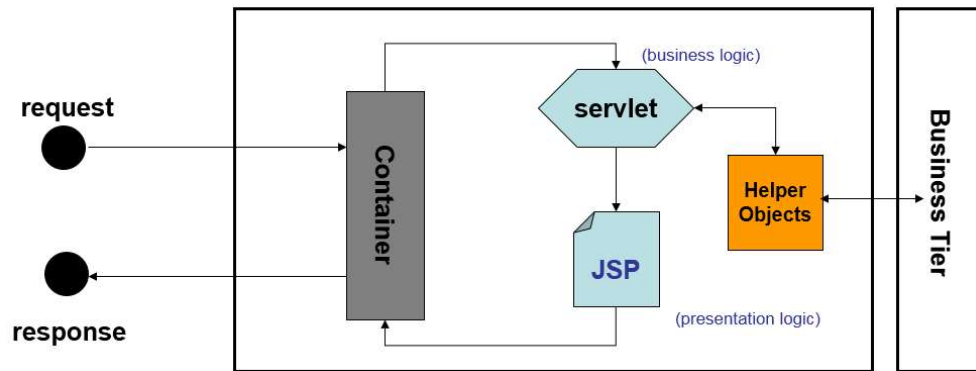
1. JSP lebih dikenal oleh semua orang dengan pengetahuan dari HTML, untuk memudahkan dalam pengembangan web page.
2. JSP memiliki built-in yang mendukung untuk penggunaan komponen software yang dapat digunakan kembali (JavaBeans). Hal ini tidak hanya membiarkan para pengembang menghindari kemungkinan menemukan kembali inti/kemudi dari tiap aplikasi, mempunyai software pendukung untuk memisahkan komponen software untuk menghandle logic promotes separation dari presentasi dan business logic
3. JSP merupakan bagian solusi dari Java untuk pengembang aplikasi web, merupakan multi-platform yang tak terpisahkan dan dapat dijalankan pada berbagai container servlet yang compatible, dengan mengabaikan vendor atau sistem operasinya

6.2. Arsitektur JSP

Untuk arsitektur dari JSP sendiri tidak terdapat perbedaan yang signifikan jika dibandingkan oleh servlet. Lokasi dan level eksekusinya nyaris sama, hanya saja berbeda dalam beberapa hal. Salah satu perbedaannya adalah Servlet merupakan business layer dari J2EE sedangkan JSP adalah presentation layer. Yang dimaksud business layer adalah servlet bekerja dibalik layar dan merupakan tempat dimana logika yang lebih rumit dan kompleks ditempatkan. Sedangkan JSP bertindak sebagai layer presentasi yang merepresentasikan sebuah servlet.

6.3. Cara Kerja JSP

Cara kerja sebuah JSP juga tidak begitu berbeda dengan servlet. Perbedaannya hanya pada sebuah file JSP terlebih dahulu diubah menjadi sebuah servlet. Hal ini akan dilakukan oleh container. Container bertanggung jawab terhadap siklus hidup dari JSP dan manajemen *threading*-nya.



Gambar 6.1 - Ilustrasi cara kerja JSP

6.4. Implementasi JSP

Untuk membuat sebuah program JSP anda dapat membuat file dengan ekstensi berupa .JSP. Sebuah kode JSP diawali dengan menggunakan tanda `<%` dan diakhiri dengan tanda `%>`. Dalam tag JSP tersebut kita dapat mengetikkan koding java seperti biasanya. Sedang untuk penulisan komentar pada file JSP terdapat beberapa cara yang dapat dilakukan. Berikut merupakan cara penulisan comment pada JSP:

Tabel 6.1. Penulisan komentar di JSP

Jenis komentar	Penulisan komentar	Kegunaan
JSP Comments	<code><%-- {CODE HERE} --%></code>	Hanya bisa digunakan diluar JSP element
HTML Comments	<code><!-- {CODE HERE} --></code>	Hanya bisa digunakan diluar JSP element
Single Line Comments	<code>//{CODE HERE}</code>	Hanya bisa digunakan di dalam deklarasi atau di luar JSP element

Untuk lebih memahami bagaimana struktur dasar dari JSP anda dapat mencoba program sederhana HelloName.jsp dibawah ini.

HelloName.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

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

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  
```

```
<title>Insert title here</title>

</head>

<body>

    <% String name = "Merry"; %>

    <p>Hello <% out.println(name); %> </p>

</body>

</html>
```



7

Aplikasi Berbasis Web dengan J2EE

Objektif :

- Mengetahui peran dari Servlet dan JSP dalam pembuatan aplikasi berbasis web
 - Mampu membangun aplikasi web dinamis sederhana dengan J2EE
-

7.1. Kelebihan J2EE dalam Pembuatan Aplikasi Berbasis Web

Pada umum, aplikasi helper pada web server bukanlah sebuah program servlet, melainkan sebuah program yang pada umumnya disebut sebagai CGI (Common Gateway Interface). Namun terdapat beberapa perbedaan dalam pengolahan proses konten web dinamis dengan CGI maupun dengan menggunakan Servlet. Hal ini juga menimbulkan kelebihan dan kekurangan yang ditawarkan oleh masing-masing pihak. Jika kita bandingkan CGI dengan servlet terdapat beberapa keuntungan yang dapat diraih apabila menggunakan servlet. Berikut merupakan keuntungan menggunakan servlet dari pada CGI:

- Servlet memiliki performa yang lebih baik
- Lebih aman
- *Write once run anywhere!*
- Independen terhadap platform
- Dapat diperluas (extensibility)

7.2. Implementasi J2EE Web Tier

Pada implementasinya J2EE pada aplikasi berbasis web dua hal yang akan diterapkan adalah Servlet dan JSP. Pada bab sebelumnya, kita telah mempelajari mengenai bagaimana cara mengembangkan sebuah program sederhana dari servlet ataupun JSP. Pada bab ini kita akan mempelajari bagaimana mengkolaborasikan fitur dari kedua teknologi tersebut.

Pada pengembangan aplikasi dengan J2EE servlet berperan sebagai kontroler dari request. Yaitu sebuah program yang akan menangani setiap request yang terjadi. Jadi pada servlet kita tidak akan menggunakannya sebagai sebuah aplikasi yang akan menampilkan sebuah hasil pada web browser. Kita akan men-forward hasil olahan dari sebuah servlet kepada sebuah page halaman web lainnya. Dan untuk halaman yang menampilkan hasil tersebut adalah JSP. Pada JSP anda tidak dianjurkan untuk menaruh algoritma program anda yang rumit pada sebuah JSP file. Hal ini akan membantu anda agar menjaga script HTML yang anda ketik akan tetap bersih dan agar mudah digunakan ulang dalam pengembangan selanjutnya.

Untuk mempermudah anda membayangkan apa yang telah kita pelajari, mari kita coba langsung 2 buah program yang ada dibawah ini:

Contoh Program 1 – SayHey (Method GET)

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <form action="Saying" method="GET">
        Input your name: <input type="text" name="name"/>
        <input type="submit" value="Say Hi!"/>
    </form>
</body>
</html>
```

hello.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
</head>
<body>
    <%
        String name;
        name = (String) request.getAttribute("name");
    %>

    <h2>Hi <% out.println(name); %></h2>
</body>
</html>
```

Saying.java

```
package com.tutorial.say;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Saying
 */
@WebServlet("/Saying")
public class Saying extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        String name;

        if (request.getParameter("name") == null) {
            out.println("GET method kosong");
            return;
        }

        name = request.getParameter("name");
        request.setAttribute("name", name);

        request.getRequestDispatcher("/hello.jsp").forward(
            request, response);
    }
}
```

Contoh Program 2 – Calculator (Method POST)

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">

    <title>Insert title here</title>
    <!-- Code CSS untuk styling -->
    <style type="text/css">
        *{
            margin:0px;
            padding:0px;
        }
        fieldset{
            display:block;
            border:none;
        }
        label {
            width: 110px;
            display:inline-block;
        }
    </style>
</head>
<body>
    <form method="POST" action="Hitung">

        <fieldset>
            <label>Angka Pertama</label><input type="text"
name="firstnum"/>
        </fieldset>

        <fieldset>
            <label>Angka Kedua</label><input type="text"
name="secondnum"/>
        </fieldset>
```



```

<fieldset>
    <label>Jenis Operasi</label>
    <select name="operator">
        <option>+</option>
        <option>-</option>
        <option>*</option>
        <option>/</option>
    </select>
</fieldset>
<br/>
<input type="submit" value="Hitung!">
</form>
</body>
</html>

```

hasil.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
    <title>Insert title here</title>
</head>
<body>

<%
    Float firstnum, secondnum, result;
    String operator;

    firstnum    = (Float) request.getAttribute("firstnum");
    secondnum   = (Float) request.getAttribute("secondnum");
    result      = (Float) request.getAttribute("result");
    operator    = (String) request.getAttribute("operator");

    %>

<h2>

```

```

        Hasil dari <br/>
        <b><% out.println(firstnum); %></b> <% out.println(operator);
%>
        <b><% out.println(secondnum); %></b> adalah <b><%
out.println(result); %></b>
    </h2>
</body>
</html>

```

Hitung.java

```

package com.calculator.hitung;

import java.io.IOException;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Hitung
 */
@WebServlet("/Hitung")
public class Hitung extends HttpServlet {
    private static final long serialVersionUID = 1L;
    float firstnum, secondnum, result;
    String operator;

    public void init(ServletConfig config) throws ServletException {
        // TODO Auto-generated method stub
    }

    public void destroy() {
        // TODO Auto-generated method stub
    }

    protected void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

```

```

        //PrintWriter out = response.getWriter();
        // Jika kosong maka terminate proses hitung
        if(request.getParameter("operator") == null ||
request.getParameter("operator") == null|| request.getParameter("operator")
== null){

            request.getRequestDispatcher("/index.jsp").forward(request, response);

            return;
        } else {

            firstnum =
Float.parseFloat(request.getParameter("firstnum"));
            secondnum =
Float.parseFloat(request.getParameter("secondnum"));
            operator = request.getParameter("operator");

            // Proses Hitung
            if(operator.equals("+")){
                result = firstnum + secondnum;
            } else if(operator.equals("-")){
                result = firstnum - secondnum;
            } else if(operator.equals("*")){
                result = firstnum * secondnum;
            } else if(operator.equals("/")){
                result = firstnum / secondnum;
            }

            // settings attribute
            request.setAttribute("firstnum", firstnum);
            request.setAttribute("secondnum", secondnum);
            request.setAttribute("result", result);
            request.setAttribute("operator", operator);

            //out.println(firstnum+ " + " + secondnum + " adalah " +
result);

            request.getRequestDispatcher("/hasil.jsp").forward(request, response);
        }
    }
}

```