

# PRAKTIKUM #01

Wisnu Handito S.Kom  
TEKNOLOGI WEB (JAVA SERVER PAGES)

# Pengenalan JSP

## **POKOK BAHASAN:**

- Dasar JSP
- Daur hidup JSP
- Yang diperlukan untuk menjalankan JSP
- Membuat dan mendeploy halaman JSP

## **TUJUAN BELAJAR:**

Setelah mempelajari bab ini, siswa diharapkan mampu:

1. Mengetahui JSP
2. Mengetahui web server
3. Mengetahui daur hidup JSP
4. Mengetahui yang diperlukan untuk menjalankan JSP
5. Membuat dan mendeploy halaman JSP

## **Dasar JSP**

JSP adalah suatu teknologi web berbasis bahasa pemrograman Java dan berjalan di Platform Java, serta merupakan bagian teknologi J2EE (Java 2 Enterprise Edition). JSP sangat sesuai dan tangguh untuk menangani presentasi di web. Sedangkan J2EE merupakan platform Java untuk pengembangan sistem aplikasi enterprise dengan dukungan API (Application Programming Inteface) yang lengkap dan portabilitas serta memberikan sarana untuk membuat suatu aplikasi yang memisahkan antara business logic (sistem), presentasi dan data.

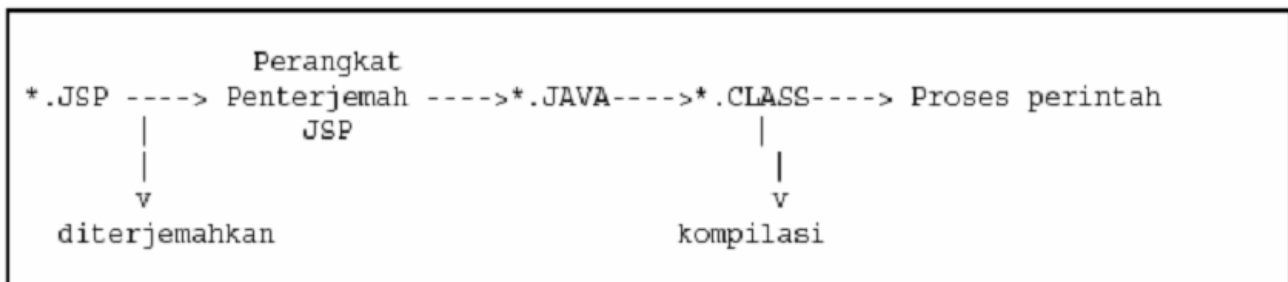
JSP merupakan bagian dari J2EE dan khususnya merupakan komponen web dari aplikasi J2EE secara keseluruhan. JSP juga memerlukan JVM (Java Virtual Machine) supaya dapat berjalan, yang berarti juga mengisyaratkan keharusan menginstal Java Virtual Machine di server, dimana JSP akan dijalankan. Selain JVM, JSP juga memerlukan server yang disebut dengan Web Container.

Teknologi JSP menyediakan cara yang lebih mudah dan cepat untuk membuat halaman-halaman web yang menampilkan isi secara dinamik. Teknologi JSP didesain untuk membuat lebih mudah dan cepat dalam membuat aplikasi berbasis web yang bekerja dengan berbagai macam web server, application server, browser dan development tool.

Java Server Pages (JSP) adalah 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. Berbeda dengan Servlet yang harus dikompilasi oleh USER menjadi class sebelum dijalankan, JSP tidak perlu dikompilasi oleh USER tapi SERVER yang akan melakukan tugas tersebut. Makanya pada saat user membuat pertama kali atau melakukan modifikasi halaman dan mengeksekusinya pada web browser akan memakan sedikit waktu sebelum ditampilkan.

## Daur Hidup JSP

Sebagai gambaran bagaimana JSP melalui masa hidupnya bisa dilihat pada gambar berikut :



Gambar 1.1 Daur hidup JSP

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:

```

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

Seperti halnya skrip-skrip server side yang lain, JSP pun memerlukan Web server. Skrip ASP memerlukan IIS sebagai web server, PHP memerlukan IIS atau Apache, sedangkan JSP bisa menggunakan Apache Tomcat sebagai salah satu web server yang mendukungnya. Agar bisa menjalankan file-file JSP yang berbasis Java, diperlukan web server yang mampu memproses Java, atau minimal JSP engine yang dapat terintegrasi dengan web server.

## Web Container

Menurut spesifikasi J2EE, dikenal EJB Container, Web Container dan Application Server. Web Container adalah services yang dijalankan oleh suatu Java Application Server khususnya untuk services yang compliance/kompatibel dengan Servlet dan JSP. Selain menjadi services oleh Java Application

Server, Web Container dapat berdiri sendiri. Contoh Web Container adalah Tomcat, ServletExec, Resin, Jrun, Blazix. Web Container juga dapat bekerja sama dengan web server, misalnya Tomcat dengan Apache, Jrun dengan IIS.

Web Server adalah software untuk server yang menangani request melalui protokol HTTP yang digunakan oleh situs-situs web saat ini dalam menangani request file statik HTML, seperti Apache dan Microsoft IIS. Web server sekarang sering "dibungkus" oleh Java Application Server sebagai HTTP Server.

Java Application Server adalah Server yang terdiri atas HTTP Server (Web Server), EJB Container maupun Web Container. Contoh Java Application Server: Sun J2EE RI 1.2/1.3, Borland AppServer 4.5/Enterprise Server 5.0, Oracle9i Application Server dan lainnya.

### **Apache Tomcat**

Tomcat adalah web application server, yang mempunyai kemampuan sebagai Servlet container dan JSP container di mana Anda bisa mendeploy Servlet dan JSP. Di atas Tomcat, Servlet dan JSP akan bekerja melayani request dari client, yang lumrahnya adalah berupa browser. Untuk bisa menjalankan Tomcat, Anda membutuhkan Java Development Kit (JDK). Untuk instalasi Jakarta Tomcat, Anda bias mendownload binary dari <http://apache.org> , dalam format .zip, .tar.gz. Yang Anda perlu lakukan hanyalah mendecompress file tersebut.

Dalam bekerja dengan Tomcat, Anda mempunyai sebuah directory yang dikenal sebagai TOMCAT\_HOME. TOMCAT\_HOME adalah directory di mana Jakarta Tomcat diinstall. Selanjutnya di bawah TOMCAT\_HOME Anda akan menemukan beberapa subdirectory, diantaranya bin/, conf/, logs/ dan webapp/. Di dalam subdirectory bin/ terdapat file-file executable terutama untuk menjalankan dan menghentikan Tomcat. Di dalam subdirectory conf/ terdapat file-file untuk configuration. Di dalam subdirectory logs/ terdapat file-file log. Dan subdirectory webapp/ adalah di mana Anda bisa meletakkan aplikasi Web yang Anda bangun dengan Servlet dan JSP. Di bawah subdirectory webapp/ Anda bisa mengcreate subdirectory. Sub directory ini akan dijadikan sebagai Context oleh Tomcat.

Anda menjalankan Tomcat dengan mengexecute startup.sh di subdirectory bin/. Sedangkan untuk menghentikan Tomcat Anda mengexecute shutdown.sh di sub directory bin/ juga. Secara default Tomcat siap melayani request dari client melalui port 8080. Melalui Web browser, Anda bisa menghubungi <http://localhost:8080>

## Elemen Dasar Pemrograman JSP

Elemen dasar JSP sama dengan Elemen dasar yang ada pada aplikasi desktop java, jadi anda hanya tinggal mengingat ulang apa yang sudah anda pelajari di dasar dasar pemrograman Java dan pemrograman berorientasi objek Java. Oleh karena itu tidak dibahas secara detail di modul ini. Akan tetapi langsung kepada penggunaan elemen-elemen tersebut di dalam JSP.

### Komentar

Seperti bahasa pemrograman lain, JSP memiliki sintaks khusus untuk memberikan komentar pada kode program, yaitu "<%-- komentar disini --%>". Komentar ini berguna untuk memudahkan kita dalam pembacaan program jika suatu saat kode program dibaca oleh orang lain.

```
<%-- ini adalah hello world --%>
<%
    Out.println("hello world");
%>
```

### Deklarasi

```
<%!
    public int fungsiTambah(int bilangan1, int bilangan2)
    {
        int jumlah = 0;
        jumlah = bilangan1 + bilangan2;
        return jumlah;
    }
%>
```

Deklarasi berguna untuk mendeklarasikan atau menginisiasi suatu variabel dan method. Suatu variabel atau method tidak akan bisa digunakan jika tidak dideklarasikan. Untuk memberikan deklarasi pada kode program diberikan sintaks "<%! %>".

### Ekspresi

```
<%! String str = "hello world"; %>
<%= str %>
```

Berguna untuk menampilkan nilai suatu variabel atau method. Dengan adanya ekspresi ini kode program bisa menjadi lebih indah dan nyaman untuk dibaca. Selain itu dengan adanya ekspresi ini, membuat konsep MVC bisa dilakukan secara optimal.

## Scriptlet

Scriptlet berisi kode-kode program untuk pemrosesan. Di scriptlet ini tidak boleh terdapat ekspresi ataupun deklarasi.

```
<%  
    for (int i=0;i<10;i++)  
    {  
        out.println("bilangan ke : " + i);  
    }  
%>
```

## Directive

Directive adalah media yang digunakan JSP untuk mengirimkan "pesan" ke JSP container. Directive berguna untuk melakukan setting nilai global seperti deklarasi class atau method. Setting yang dilakukan oleh directive berlaku pada seluruh halaman (hanya halaman itu saja).

Berikut sintaks directive :

```
<%@ nama_directive atribut1="nilai1" atribut2="nilai2" . . . . %>
```

Ada 3 jenis directive dalam jsp yang terbagi sesuai dengan fungsinya. Ketiga directive tersebut adalah :

### 1. page directive

Dengan page directive ini, suatu halaman JSP bisa diberikan informasi tentang package package apa yang akan diimport, halaman mana yang ditampilkan ketika terjadi error, dll. Berikut ini beberapa atribut dari page directive untuk memberikan informasi pada suatu halaman :

#### a. Atribut language

Atribut ini digunakan untuk memberikan informasi bahasa apa yang digunakan. Memang untuk saat ini bahasa yang digunakan oleh JSP hanya java saja. Akan tetapi atribut ini dibuat untuk mensupport agar kedepannya apabila JSP engine mendukung bahasa pemrograman lain.

```
<%@ page language="java" %>
```

#### b. Atribut impor

Atribut ini digunakan untuk memberikan informasi package apa saja yang dibutuhkan dalam suatu halaman jsp. Apabila lebih dari satu package yang diimport maka dipisahkan tanda koma (,).

```
<%@ page import="java.io.*, java.sql.*" %>
%>
```

### c. Atribut info

Atribut ini digunakan hanya untuk memberikan informasi tentang suatu halaman. Misalnya kita ingin memberikan info bahwa halaman jsp yang telah kita buat akan kita cantumkan nama kita.

```
<%@ page info="buatan indonesia asli" %>
%>
```

### d. Atribut errorPage

Atribut ini menentukan halaman mana yang akan di-redirect ketika terjadi eror pada halaman jsp. Dengan adanya atribut ini maka jika halaman jsp error akan diredirect ke halaman nilai atribut tersebut sehingga akan mengurangi terjadinya hal-hal yang tidak diinginkan semisal, hacking yang umumnya terjadi karena adanya error dalam web.

```
<%@ page errorPage="error.jsp" %>
%>
```

### e. Atribut contentType

Suatu halaman web dapat diketahui isinya dengan variable contentType dalam protocol HTTP. Dengan atribut ini pulalah kita juga akan merepresentasikan isi dari halaman web kita, bias berupa gambar, pdf, text, dll. Nilai defaultnya adalah text/html

### f. Atribut session

Apabila suatu halaman ingin menggunakan session sebagai variable bagi user, maka session ini diisi dengan nilai "true"

```
<%@ page session="true" %>
%>
```

Untuk lebih mengefektifkan kode program, penggunaan atribut bisa dilakukan sekaligus. contoh :

```
<%@ page language="java" import="java.sql.*, java.io.*"
session="true" info="buatan Indonesia asli" errorPage="error.jsp" %>
```

## 2. include directive

include directive ini berguna jika kita ingin menyisipkan isi suatu halaman dalam kode kita. Tidak seperti page directive yang memiliki banyak atribut, include directive hanya mempunyai

satu atribut yaitu "file" yang nilainya merupakan Ike yang isinya akan disisipkan dalam kode.

Berikut contoh sintaks yang digunakan :

```
<%@ include file="/namafile_yang disisipkan" %>
```

URL file yang akan disisipkan harus diawali tanda "/".

## Tipe Data dan Variabel

Berikut contoh operasi tipe data konversi dengan arah konversi string ke integer, anda simpan dengan nama konversi.jsp:

```
<FORM method="post" action="konversi.jsp">
    Masukkan sebuah bilangan bulat:<BR>
    <INPUT TYPE=TEXT NAME=bil><BR>
    <INPUT TYPE=SUBMIT VALUE="Proses">
</FORM>
<%
    String strBilangan = request.getParameter("bil");
    if (strBilangan != null) {
        int bilangan = Integer.valueOf(strBilangan).intValue();
        out.println("110% dari " + bilangan + " = " + (1.1 * bilangan));
    }
%>
```

## Operator

Berikut contoh operator aritmetika :

```
<%
    out.println("1 + 2 * 3 = " + (1 + 2 * 3));
    out.println("<BR>");
    out.println("13 % 5 = " + (13 % 5));
    out.println("<BR>");
    out.println("13 % 5.2 = " + (13 % 5.2));
    out.println("<BR>");
    out.println("13 / 5 = " + (13 / 5));
    out.println("<BR>");
    out.println("13 / 5.2 = " + (13 / 5.2));
%>
```



### for.jsp

```
<html>
<head>
<title>For loop di JSP</title>
</head>

<body>
<%
for(int i=0;i<=10;i++)
{
    out.print("Loop through JSP count :"+i+"<br/>");
}
%>
</body>
</html>
```

### while.jsp

```
<html>
<head>
<title>while loop in JSP</title>
</head>

<body>
<%
int i=0;
while(i<=10)
{
    out.print("While Loop through JSP count :"+i+"<br/>");
    i++;
}
%>
</body>
</html>
```

### dowhile.jsp

```
<html>
<head>
<title>do-while loop in JSP</title>
</head>

<body>
<%
int i=0;
do{
    out.print("Do While Loop through JSP count :"+i+"<br/>");
    i++;
}
while(i<=10);
%>
</body>
</html>
```

### ifelse.jsp

```
<html>
<head>
<title>If else di JSP</title>
</head>

<body>
<%
String sName="wawan";
String sSecondName="irwan";

    if(sName.equals("wawan")){
        out.print("if condition check satisfied JSP count :"+sName+"<br>");
    }

    if(sName.equals("wawan") && sSecondName.equals("wawan"))
    {
        out.print("if condition check if Block <br>");
    }
    else
    {
        out.print("if condition check else Block <br>");
    }
%>
</body>
</html>
```

### ifelse2.jsp

```
<FORM method="post" action="controll.jsp">
    Masukkan nilai Ujian (0-100):<BR>
    <INPUT TYPE=TEXT NAME=nilai><BR>
    <INPUT TYPE=SUBMIT VALUE="Proses">
</FORM>
<%
String strNilai = request.getParameter("nilai");
if (strNilai == null)
    strNilai = "0";
int nilaiUjian = Integer.valueOf(strNilai).intValue();
char skor;

if (nilaiUjian >= 90)
    skor = 'A';
else if (nilaiUjian >= 80)
    skor = 'B';
else if (nilaiUjian >= 60)
    skor = 'C';
else if (nilaiUjian >= 50)
    skor = 'D';
else
    skor = 'E';
```

```
out.println("Nilai: " + nilaiUjian + "<BR>");  
out.println("Skor: " + skor);
```

```
%>
```

### ifelse2.jsp

```
<FORM method="post" action="control2.jsp">  
    1. Tour ke Candi Borobudur<BR>  
    2. Tour ke Danau Toba<BR>  
    3. Tour ke Gunung Bromo<BR>  
        Pilihan salah satu <BR>  
<INPUT TYPE=TEXT NAME=pilihan><BR>  
<INPUT TYPE=SUBMIT VALUE="Proses">  
</FORM>  
<%  
    String strPilihan = request.getParameter("pilihan");  
    if (strPilihan == null)  
        strPilihan = "0";  
    int pilihan = Integer.valueOf(strPilihan).intValue();  
    switch (pilihan) {  
    case 1:  
        out.println("Pilihan Anda Candi Borobudur<BR>");  
        break;  
    case 2:  
        out.println("Tour ke Danau Toba<BR>");  
        break;  
    case 3:  
        out.println("Tour ke Gunung Bromo<BR>");  
        break;  
    default:  
        out.println("Silakan memilih 1, 2, atau 3");  
    }  
%>
```

### Latihan Praktikum :

1. Buatlah 1 buah aplikasi untuk menghitung volume dari bangun ruang.
2. Buatlah 2 buah aplikasi untuk menghitung **luas dan keliling** dari bangun datar.
3. Buatlah 1 buah form buku tamu sederhana yang terdiri dari Nama(Textbox), Email(Textbox), dan Komentar(Textarea)