Bab 3
Directive JSP

POKOK BAHASAN:

- Direktif JSP
- Jenis Direktif JSP
- Tag Deklaratif

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan mampu:

1. Mahasiswa mengenal directive
2. Mahasiswa dapat membuat directive JSP
3. Mahasiswa mengenal deklaratif
4. Mahasiswa dapat membuat deklaratif JSP

3.1 Mengenal Direktif JSP

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).

Sintaks Directive

Secara umum sintaks directive adalah sebagai berikut :

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

3.2 Jenis Directive

Directive pada JSP terdiri atas tiga jenis tentu saja dengan fungsi yang berbeda-beda.
JSP memiliki tiga buah direktif :

- Page : digunakan untuk mendefinisikan atribut-atribut yang terdapat pada halaman JSP
- Include : digunakan untuk menyisipkan suatu berkas atau mengimpor suatu kelas.
- Taglib : digunakan untuk mendefinisikan tag-tag buatan pemrogram.

page directive

Directive ini berfungsi untuk mendefinisikan atribut-atribut yang akan berlaku pada halaman tersebut. Sebagai contoh dengan menggunakan directive ini suatu halaman bisa diberikan informasi mengenai apa, meng-import package-package yang akan digunakan, menyatakan halaman tersebut terlibat dalam HTTP session, mendefinisikan URL yang akan ditampilkan apabila terjadi error pada halaman JSP tersebut dan lain-lain. Pada sebuah halaman JSP dapat berisi atas banyak page directive.

Tanda yang digunakan untuk directive ini adalah :

```
<%@ page atribut1 atribut2 . . . %>
```

Atribut untuk tipe directive ini dapat dilihat pada contoh berikut :

Atribut language

Atribut ini mendefinisikan bahasa pemrograman apa yang digunakan pada halaman tersebut. Atribut ini ada dikarenakan apabila dimasa yang akan datang JSP engine dapat men-support bahasa pemrograman lain. Berikut adalah contoh penggunaannya :

```
<%@ page language=”java” %>
```

Atribut import

Berikut adalah contoh penggunaannya :

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

Atribut info

Atribut ini hanya mendefinisikan informasi dari halaman. Dengan menggunakan atribut ini suatu aplikasi servlet dapat mengambil informasi tersebut dengan method Servlet.getServletInfo(). Berikut adalah contohnya :

```
<%@ page info=”Ini adalah halaman JSP-nya Chocolove” %>
```

Atribut errorPage

Atribut ini mendefinisikan URL yang akan ditampilkan apabila terjadi error pada halaman JSP tersebut.

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

Atribut contentType

Nilai default dari atribut ini adalah “text/html”.

Atribut session

Menyatakan halaman tersebut terlibat dalam HTTP session. Apabila halaman JSP menggunakan directive page dengan atribut ini artinya halaman tersebut nantinya akan digunakan untuk mengakses atau memberikan nilai pada variabel yang disimpan pada session.

```
<%@ page session=”true” %>
```

Atribut lai yang dimiliki oleh directive ini adalah : extends, buffer, autoFlush, isErrorPage dan isThreadSafe.
Untuk penggunaan atribut tentu saja tidak harus dituliskan satu-satu seperti contoh di atas, tapi dapat disatukan sekaligus, seperti contoh berikut :

```
<%@ page language=”java” import=”java.sql.*, java.io.*, java.util.*”
session=”true” buffer=”24kb” autoFlush=”true” info=”Contoh penggunaan
Directive” errorPage=”error.jsp” isErrorPage=”false” isTreadSafe=”false” %>
```

include directive

Directive ini berfungsi untuk menyisipkan isi dari suatu file dengan tipe teks pada suatu halaman JSP. Sintaks yang digunakan oleh directive ini adalah :

```
<%@ include file="/namafile_yang_akan_disisipkan" %>
```

Bisa dilihat pada directive ini mempunyai satu atribut yaitu file. URL dari file yang akan disisipkan harus diawali dengan tanda "/". Apabila yang akan disisipkan adalah file header.html maka penulisannya adalah :

```
<%@ include file="/header.html" %>
```

Sedangkan apabila file header.html berada dalam direktori "html-file" maka penulisannya menjadi sebagai berikut :

```
<%@ include file="/html-file/header.html" %>
```

taglib directive

Directive ini berfungsi untuk penggunaan tag-tag yang dibuat sendiri oleh user pada halaman JSP. Tag-tag tersebut biasanya disimpan dalam “tag library” dalam bentuk file yang dikompres (ZIP atau JAR). Dalam file yang dikompres tersebut terdapat class-class dalam suatu paket. Dan untuk memanggil atau mengoperasikan method atau properti dalam class tersebut digunakan directive ini.
Sintaks dari directive ini adalah :

```
<%@ taglib uri=”tag_library_URI” prefix=”tag_prefix” %>
```

Atribut uri (Uniform Resource Identifier) berfungsi sebagai “tag library descriptor”. Dan atribut prefix berfungsi sebagai ID yang akan mempermudah “JSP Compiler” menentukan tag-tag dari “external library”. Tag-tag yang telah dikenali oleh “JSP Compiler” adalah jsp, jspx, java, javax, servlet dan sunw.
Berikut adalah contoh tag dari “external library” :

```
<%@ taglib uri="http://jakarta.apache.org/taglibs/application-1.0" prefix="app" %>
<app:attribute name="test1"/>
<app:setAttribute name="test1">Isi Atribut</app:setAttribute>
```

Pada contoh di atas, secara default tidak dapat dijalankan dan akan menampilkan pesan error. Karena “tag library” ini belum diinstall pada sistem.

Beberapa atribut directive page :

Atribut	*Keterangan*
`language`	Mendefinisikan bahasa yang digunakan pada code fragmen
`extends`	Mendefinisikan kelas induk dari servlet yang dihasilkan
`import`	Mendefinisikan daftar kelas-kelas yang diimport
`session`	Menentukan apakah sebuah objek session harus didefinisikan
`buffer`	Mengontrol ukuran buffer keluaran response
`autoFlush`	Jika 'false', jika terjadi buffer overflow akan muncul exception, jika 'true', buffer akan di*flush* ke stream output
`isThreadSafe`	Jika 'false', container JSP hanya mengerjakan 1 request sat waktu.
`info`	Mendefinisikan suatu String yang dapat dipanggil dengan method getServletInfo()
`errorPage`	Mendefinisikan URL yang akan dikirimi sembarang objek Exception atau Error
`isErrorPage`	Jika 'true', page aktif adalah error page juga
`contentType`	Mendefinisikan nilai tipe header MIME

Gambar 3.1

3.3 Mengenal Tag Deklaratif

Semua bahasa pemrograman menyediakan variable yang berfungsi untuk menyimpan suati nilai dan nilai yang ada di dalamnya dapat diubah sewaktu-waktu. Begitu halnya JSP, JSP menyediakan tag yang secara khususu ditujukan untuk melakukan pendeklarasian variable yang berlevel halaman. Variabel seperti ini akan dikenali di sepanjang halaman. Tag yang dimaksud dinamakan tag deklaratif. Tag ini berbentuk sebagai berikut :

```
<%!.....%>
```

Contoh mendeklarasikan variabel bernama buku yang bertipe string, dan variabel bernama harga bertipe int.

```
<%!
        String buku;
        Int harga = 60000;
%>
```

3.4 Percobaan

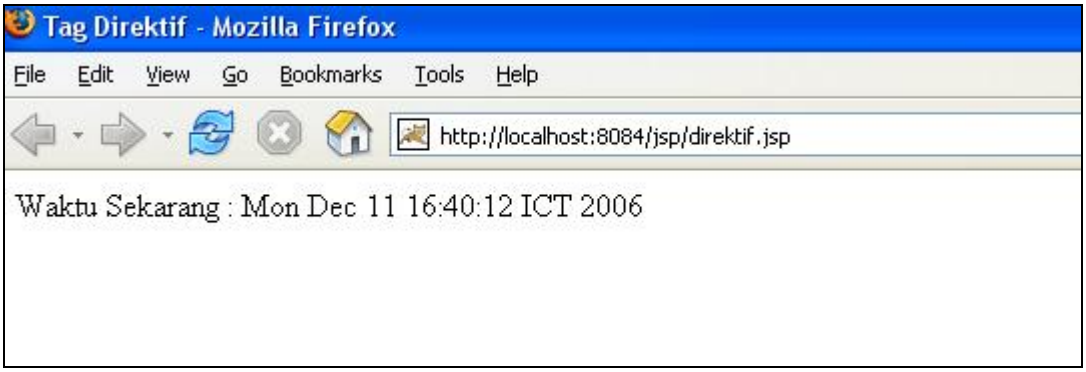
1. Membuat tag direktif JSP

Kode : direktif.jsp

```
<html>
   <head>
      <title>Tag Direktif</title>
   </head>
   <body>
   <%@ page import="java.util.Date"%>
   Waktu Sekarang : <%= new Date()%>
   </body>
</html>
```

Listing program 3.1

Hasil kode direktif.jsp



Gambar 3.2

Dengan adanya pengimporan java.util.Date secara eksplisit melalui :

<%@ page import="java.util.Date"%>

Maka penulisan :

<%= new java.util.Date() %>

Bisa digantikan dengan :

<%= new Date()%>

2. Membuat deklarasi JSP
   Kode : deklarasi.jsp

```
<html>
   <head>
      <title>Tag Deklarasi</title>
   </head>
   <body>
 <%!
   String buku;
   int harga = 60000;
 %>

 Setelah Deklarasi : <br>
 <%
   buku = "Pemrograman JSP";
   out.println("Judul Buku : " + buku + "<BR>");
   out.println("Harga BUku : " + harga + "<BR>");
 %>
   </body>
</html>
```

Listing program 3.2

Hasil di browser :

Gambar 3.3

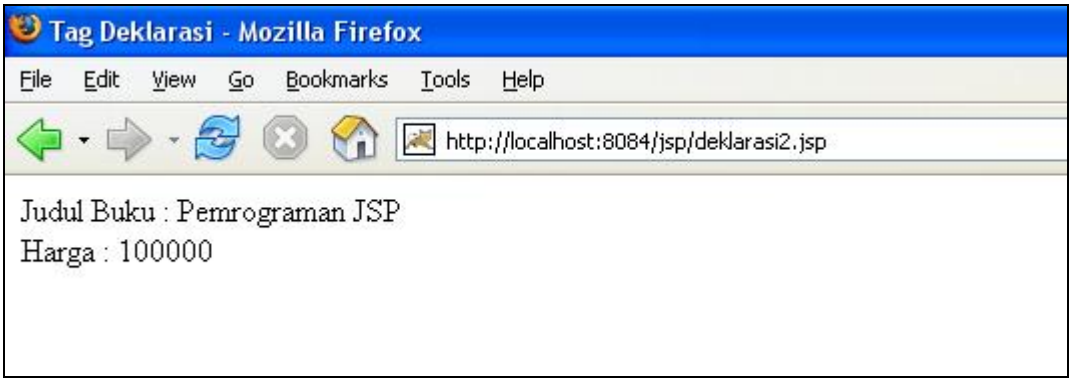
3. Deklarasi variabel dengan tag scriplet.
   Pendeklarasian variabel juga dapat dilakukan secara langsung pada tag scriplet (<% %>)

Kode : deklarasi2.jsp

```
<html>
   <head>
            <title>Tag Deklarasi</title>
   </head>
   <body>
    <%
       String buku;
       int harga = 100000;
       buku = "Pemrograman JSP";

       out.println("Judul Buku : " + buku + "<BR>");
       out.println("Harga : " + harga + "<br>");
    %>
    </body>
</html>
```

Listing program 3.3

Hasil di browser :



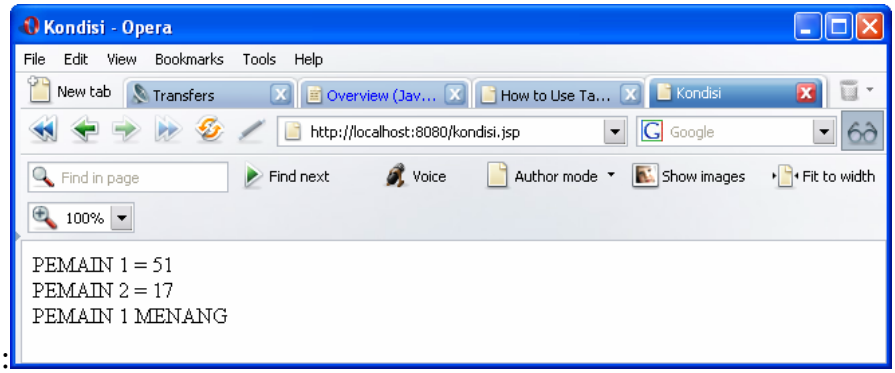
Gambar 3.4

Contoh Soal :

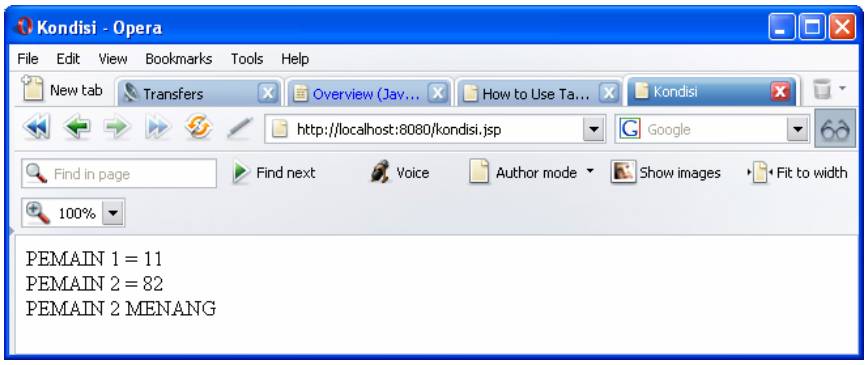
1. Terdapat dua pemain, masing-masing pemain mendapatkan nilai secara random, bilangan random yang dibangkitkan antara 0 – 99. Jika pemain 1 lebih besar dari pemain 2 maka pemain 1 yang menang, dan jika pemain 2 lebih besar dari pemain 1 maka pemain 2 yang menang. Simpan dengan nama kondisi.jsp.

```
<html>
<title> Kondisi </title>
<body>

<%
int pemain1 = (int) (Math.random()* 100);
int pemain2 = (int) (Math.random()* 100);

out.println("PEMAIN 1 = " + pemain1 );
out.println("<br>PEMAIN 2 = " + pemain2);

if (pemain1 > pemain2)
out.println("<br>PEMAIN 1 MENANG");
else
out.println("<br>PEMAIN 2 MENANG");

%>

</body>
</html>
```

Hasil di browser



Gambar 3.5

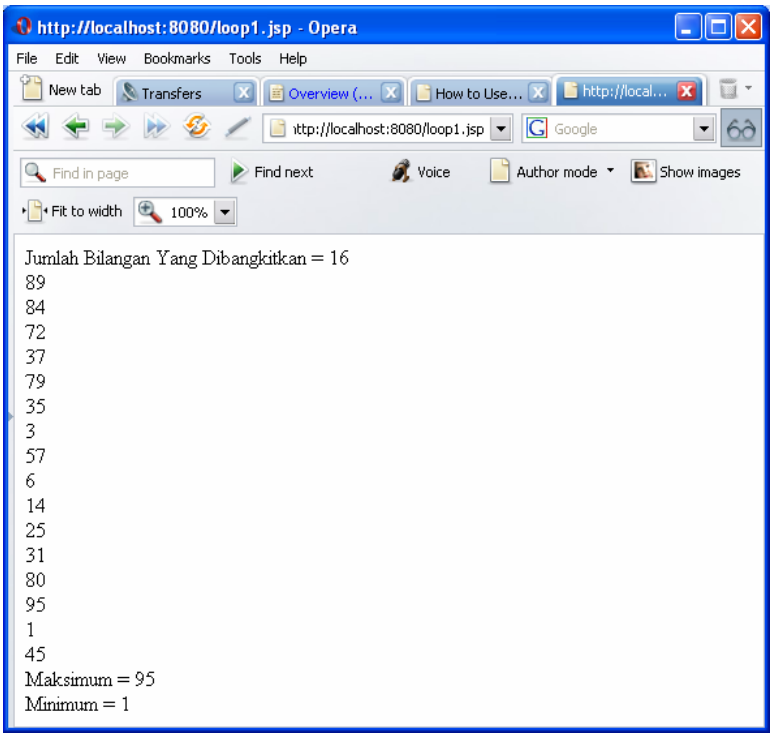


Gambar 3.6

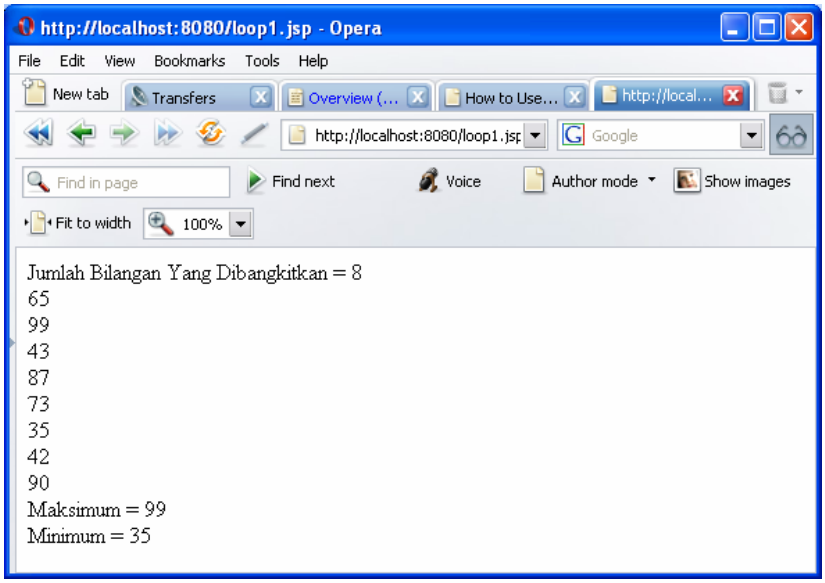
2. Tentukan sejumlah n bilangan. Besar n tentukan secara random, maksimal 20 bilangan. Setiap bilangan yang dibangkitkan antara 0 – 99. Dari semua bilangan tadi, tentukan nilai maksimum dan minimumnya. Simpan dengan nama loop1.jsp.

```
<html>
<body>

<%
int max = 0 ;
int min = 1000 ;
int jumlah = (int) (Math.random() * 20) ;
out.println("Jumlah Bilangan Yang Dibangkitkan = " + jumlah+"<br>");

int i = 0 ;
while(i<jumlah){
  int r = (int) (Math.random() * 100) ;
  max = Math.max(max,r) ;
  min = Math.min(min,r) ;
  out.println(r+"<br>");
  i++ ;
}

out.println("Maksimum = "+max);
out.println("<br>Minimum = "+min);
%>

</body>
</html>
```

Hasil di browser :



Gambar 3.7

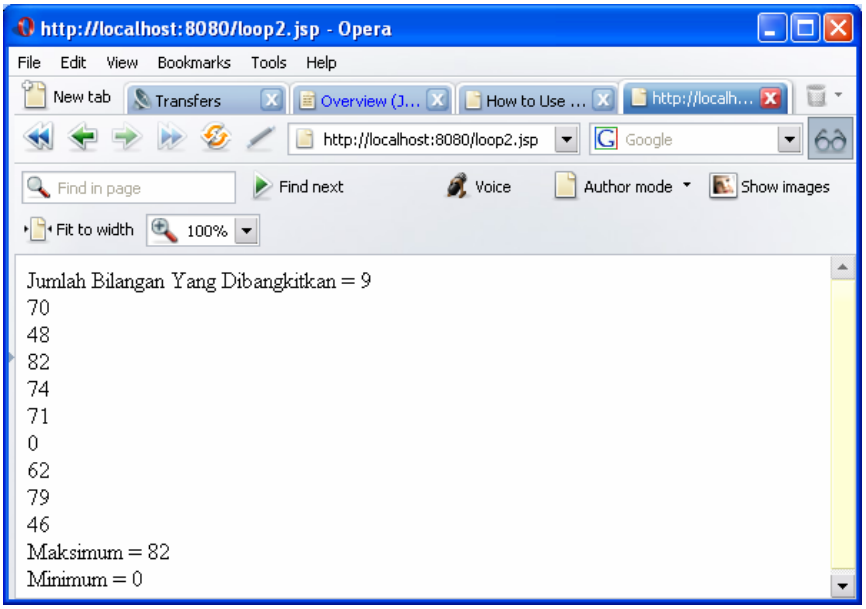


Gambar 3.8

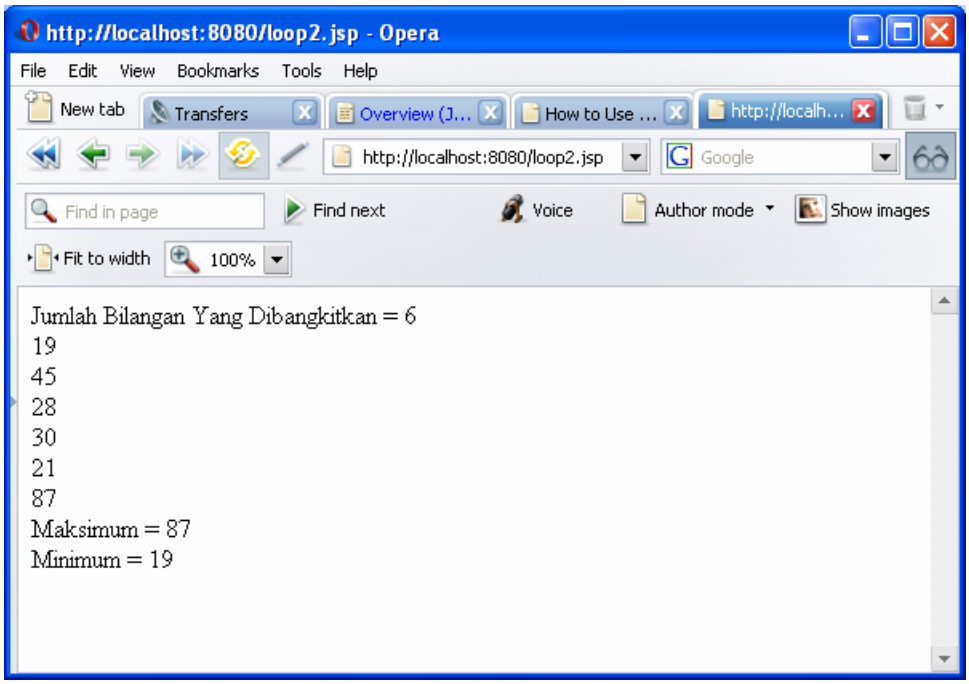
3. Tentukan sejumlah n bilangan. Besar n tentukan secara random, maksimal 20 bilangan. Simpan bilangan-bilangan tersebut dalam sebuah array. Setiap bilangan yang dibangkitkan antara 0 – 99. Dari semua bilangan tadi, tentukan nilai maksimum dan minimumnya. Simpan dengan nama loop2.jsp.

```
<html>
<body>

<%

int max = 0 ;
int min = 1000 ;

int jumlah = (int) (Math.random() * 20) ;
out.println("Jumlah Bilangan Yang Dibangkitkan = " +
jumlah+"<br>");
int A[] = new int[jumlah] ;

int i = 0 ;
while(i<jumlah){
  A[i] = (int) (Math.random() * 100) ;
  out.println(A[i]+"<br>");

  max = Math.max(max,A[i]) ;
  min = Math.min(min,A[i]) ;

  i++ ;
}

out.println("Maksimum = "+max);
out.println("<br>Minimum = "+min);

%>

</body>
</html>
```

Hasil di browser :



Gambar 3.9

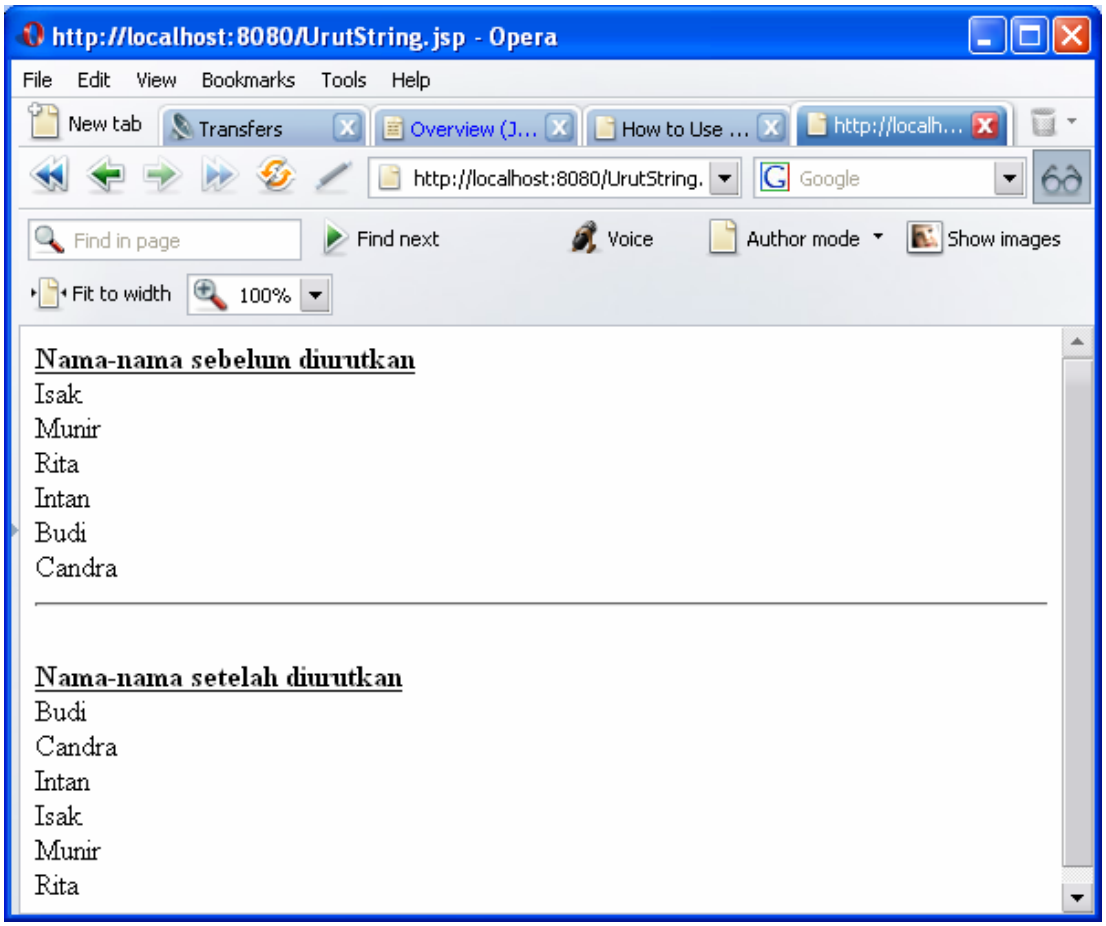


Gambar 3.10

4. Terdapat array String yang berisi nama-nama. Nama-nama tersebut diurutkan menggunakan Arrays.sor(). Simpan dalam file UrutString.jsp.

```
<html>
<body>

<%@ page import="java.util.Arrays" %>
<%

String nama[] =
{"Isak","Munir","Rita","Intan","Budi","Candra"} ;
out.println("<u><b>Nama-nama sebelum
diurutkan</b></u><br>");
for(int i=0;i<nama.length;i++)
  out.println(nama[i]+"<br>");

Arrays.sort(nama);

out.println("<hr><u><b><br>Nama-nama setelah
diurutkan</b></u><br>");
for(int i=0;i<nama.length;i++)
  out.println(nama[i]+"<br>");

%>

</body>
</html>
```

Hasil di browser :



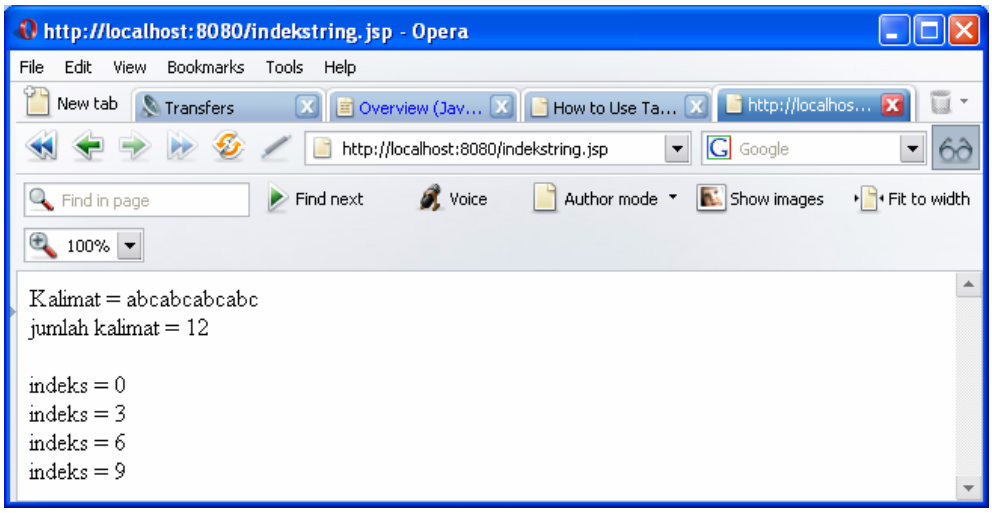
Gambar 3.11

5. Tentukan sebuah kalimat, misalkan kalimat tersebut adalah "abcabcabcabc", carilah huruf a diindek berapa saja. Untuk mendapatkan huruf a diindeks berapa saja gunakan fungsi indexOf(String str,int fromIndex). Fungsi ini akan mengembalikan nilai indeks pada String yang pertama kali yang memenuhi substring yang ditentukan, pencarian dimulai dari fromIndex. Simpan dengan nama indekstring.jsp.

```
String kalimat = "abcabcabcabc" ;
int i = indexOf(”a”,2) ; // i = 3
```

```
<html>
<body>

<%
   String kalimat = "abcabcabcabc" ;
   out.println("Kalimat = " + kalimat+"<br>");
   int i=0 ;
   out.println("jumlah kalimat = " + kalimat.length()+"<br>");

   while(i<kalimat.length()){
        i = kalimat.indexOf("a",i) ;

        if (i != -1)  {
            out.println("<br>indeks = " + i);
            i++;
        }
        else
            break ;
    }
%>
</body>
</html>
```

Hasil di browser :

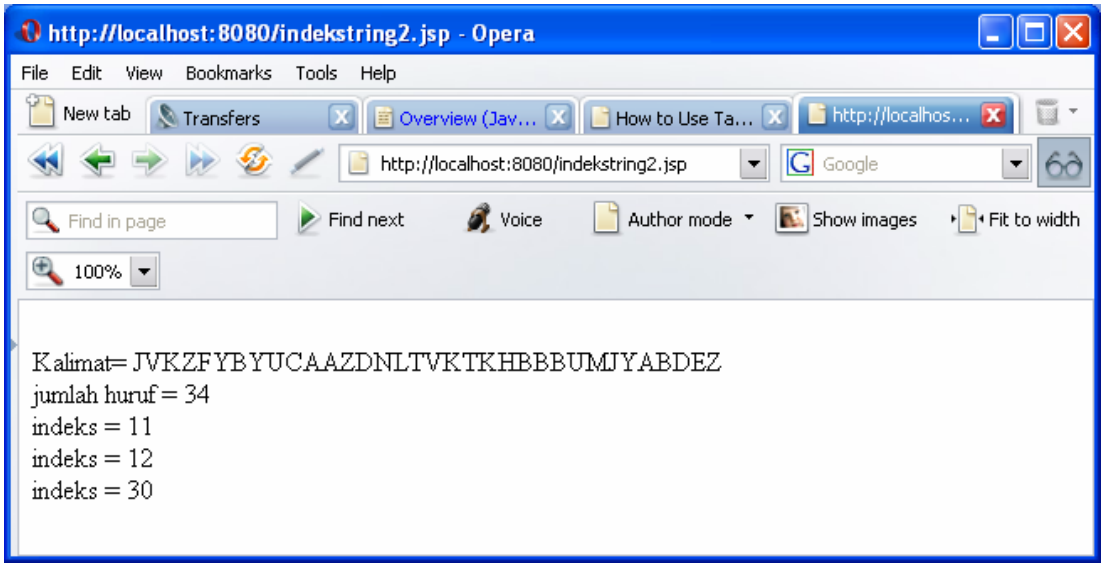


Gambar 3.12

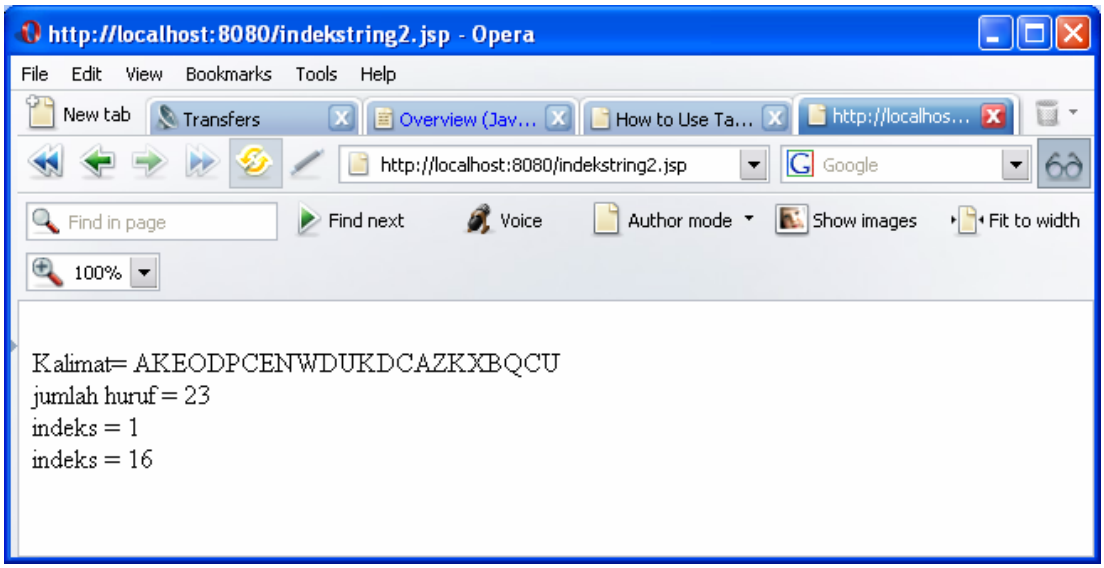
6. Bangkitkan kalimat secara random, carilah huruf A diindek berapa saja. Untuk mendapatkan huruf A diindeks berapa saja gunakan fungsi indexOf(String str,int fromIndex). Fungsi ini akan mengembalikan nilai indeks pada String yang pertama kali yang memenuhi substring yang ditentukan, pencarian dimulai dari fromIndex. Fungsi ini akan mengembalikan nilai -1 apabila tidak menemukan posisi subString yang dicari sehingga apa bila nilai mengembalikan -1 maka lakukan break. Simpan dengan nama indekstring2.jsp.

```
<html>
<body>
<%
   int jumKata = (int)(Math.random() * 50);
   char kata2[] = new char[jumKata] ;
   for(int k=0;k<jumKata;k++){
      int h = ((int)(Math.random() * 26)) + 65 ;
      kata2[k] = (char) h ;
   }

   String kalimat=null ;
   kalimat = kalimat.copyValueOf(kata2) ;

   out.println("<br>Kalimat= " + kalimat);
   int i=0 ;
   out.println("<br>jumlah huruf = " + kalimat.length());

   while(i<kalimat.length()){
        i = kalimat.indexOf("A",i) ;

        if (i != -1)  {
            i++;
            out.println("<br>indeks = " + i);
        }
        else
            break ;
    }
%>

</body>
</html>
```

Hasil di browser :



Gambar 3.13



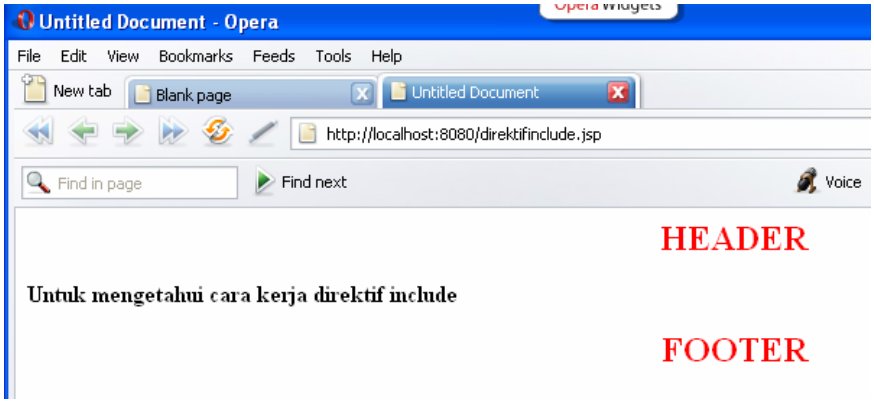
Gambar 3.14

7. Penggunaan direktif include
Ketikkan program di bawah ini. Output ditunjukkan pada gambar 3.15

```
<html >
<body>
<%@include file="header.html"%>
<p>
<strong>Untuk mengetahui cara kerja direktif
include</strong>
</p>
<%@include file="footer.html"%>
</body>
</html>
```

```
<html >
<body>
<div align="center" class="style1">HEADER</div>
</body>
</html>
```

```
<html>
<body>
<div align="center" class="style1">FOOTER</div>
</body>
</html>
```



Gambar 3.15

8. Penggunaan atribut isErrorPage.

errorPage.jsp

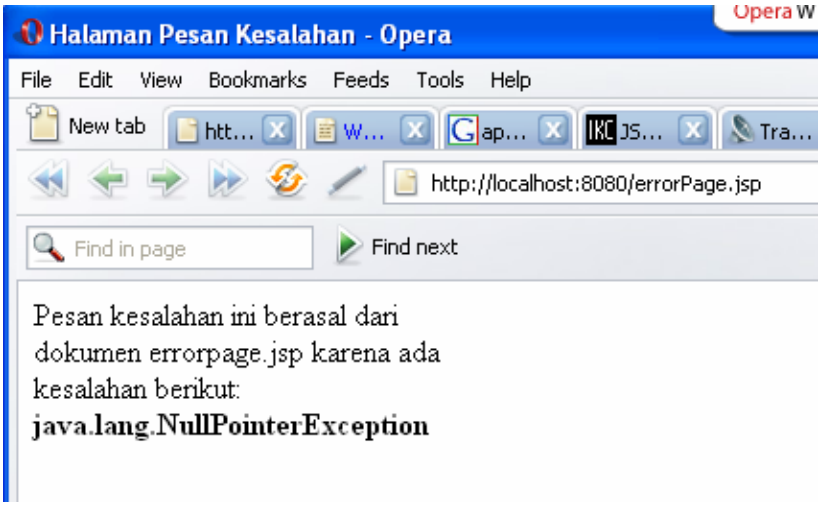
```
<HTML>
<HEAD>
<TITLE>Tes Atribut errorPage</TITLE>
</HEAD>
<BODY>

<%@ page errorPage="pesan.jsp" %>

<%
   String strBilA = request.getParameter("bila");
   String strBilB = request.getParameter("bilb");

   float bilA = Float.valueOf(strBilA).floatValue();
   float bilB = Float.valueOf(strBilB).floatValue();

   out.println(bilA + "/" + bilB + " = " + bilA / bilB);
%>

</BODY>
</HTML>
```

pesan.jsp

```
<HTML>
<HEAD>
<TITLE>Halaman Pesan Kesalahan</TITLE>
</HEAD>
<BODY>

<%@ page isErrorPage="true" %>

Pesan kesalahan ini berasal dari<BR>
dokumen errorpage.jsp karena ada<BR>
kesalahan berikut:<BR>
<B><%= exception %></B><BR>

</BODY>
</HTML>
```

Jalankan program dengan cara sebagai berikut: Apa yang terjadi?
http://localhost:8080/errorPage.jsp
http://localhost:8080/errorPage.jsp?bila=23&bilb=2
http://localhost:8080/errorPage.jsp?bila=23&bilb=b

Output program :



Gambar 3.16

3.5 Latihan Soal

1. Apa yang dimaksud dengan directive?
2. Sebutkan jenis directive ?
3. Sebutkan cara pembuatan directive JSP?
4. Apa yang dimaksud dengan deklaratif?
5. Sebutkan cara pembuatan deklaratif ?
6. Buat sebuah program masukan user dengan memanfaatkan tag deklaratif dan deklarasi. Data yang dimasukkan adalah nama user dan waktu saat ini.