

Pengenalan JSP

Chocolove Mic

chocolove_mic@yahoo.co.uk

<http://www.mycgiserver.com/~chocolove2003>

Lisensi Dokumen:

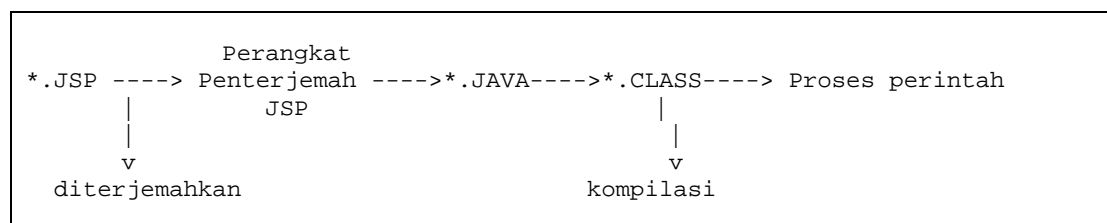
Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

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.

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



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.

Apache Tomcat Web Server

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.

Instalasi Apache Tomcat

Paket yang diperlukan untuk proses instalasi adalah :

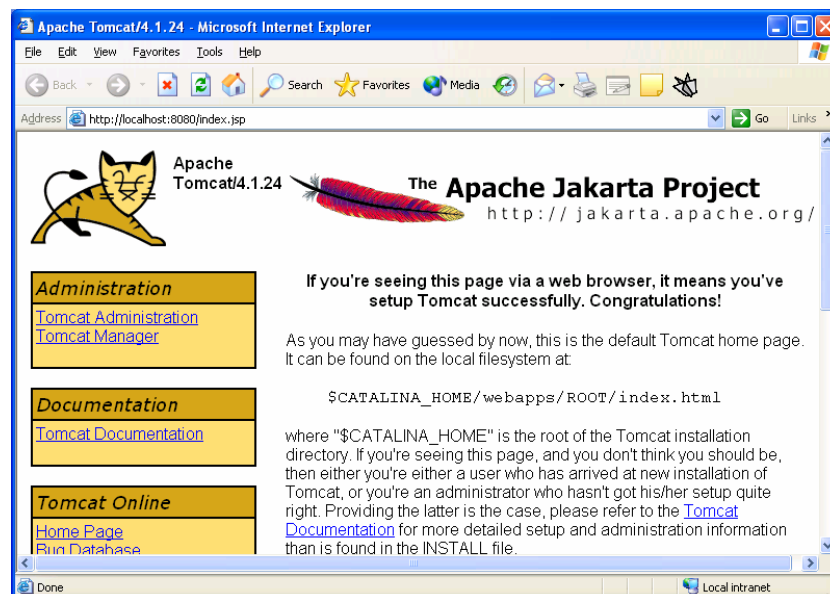
- Java 2 Software Development Kit (J2SDK) (<http://java.sun.com/j2se/>)
- Apache Tomcat (<http://jakarta.apache.org>)

Langkah atau urutan instalasi :

- install J2SDK
- install Apache Tomcat

Setelah melewati tahapan instalasi di atas maka untuk mencoba maka tahapan yang dilakukan adalah :

- Start Tomcat (pada sistem operasi MS Windows, pilih Start | Program | Apache Tomcat | Start Tomcat).
- Buka web browser, kemudian ketikkan URL berikut <http://localhost:8080>. Hasilnya bisa dilihat pada gambar berikut.



Konfigurasi

File konfigurasi Apache Tomcat ada 3 yaitu :

- `server.xml`, setting konfigurasi server seperti port atau direktori yang digunakan.
- `users.xml` atau `tomcat-users.xml`, menyimpan user dan password pada server. Selain itu juga menentukan hak akses dari user.
- `web.xml`, setting konfigurasi untuk setiap aplikasi web.

Secara default “direktori kerja utama” adalah <TOMCAT-HOME>/webapps/ROOT. Apabila user ingin mendeklarasikan direktori misalnya pada D:\Dokumen\Web\JSP sebagai “direktori kerja utama” maka yang perlu dilakukan adalah memodifikasi file server.xml pada bagian:

```
<!-- Tomcat Root Context -->  
<Context path="" docBase="ROOT" debug="0"/>
```

menjadi :

```
<!-- Tomcat Root Context -->  
<Context path="" docBase="D:\Dokumen\Web\JSP" debug="0"/>
```

apabila kita ingin mengakses suatu direktori kerja C:\app\JSP dengan URL sebagai berikut <http://localhost:8080/jsp/> maka pada server.xml bisa disetting seperti berikut :

```
<!-- Tomcat Examples Context -->  
<Context path="/jsp" docBase="C:\app\JSP\" debug="0" reloadable="true"/>
```

Selain dengan cara di atas, untuk melakukan konfigurasi “direktori kerja” dan URL yang diinginkan bisa digunakan fasilitas yang diberikan oleh Apache Tomcat 4.1 yaitu Manger. Untuk mengaksesnya dapat digunakan URL sebagai berikut:

<http://localhost:8080/manager/html>.

Maka akan dimunculkan tampilan sebagai berikut :



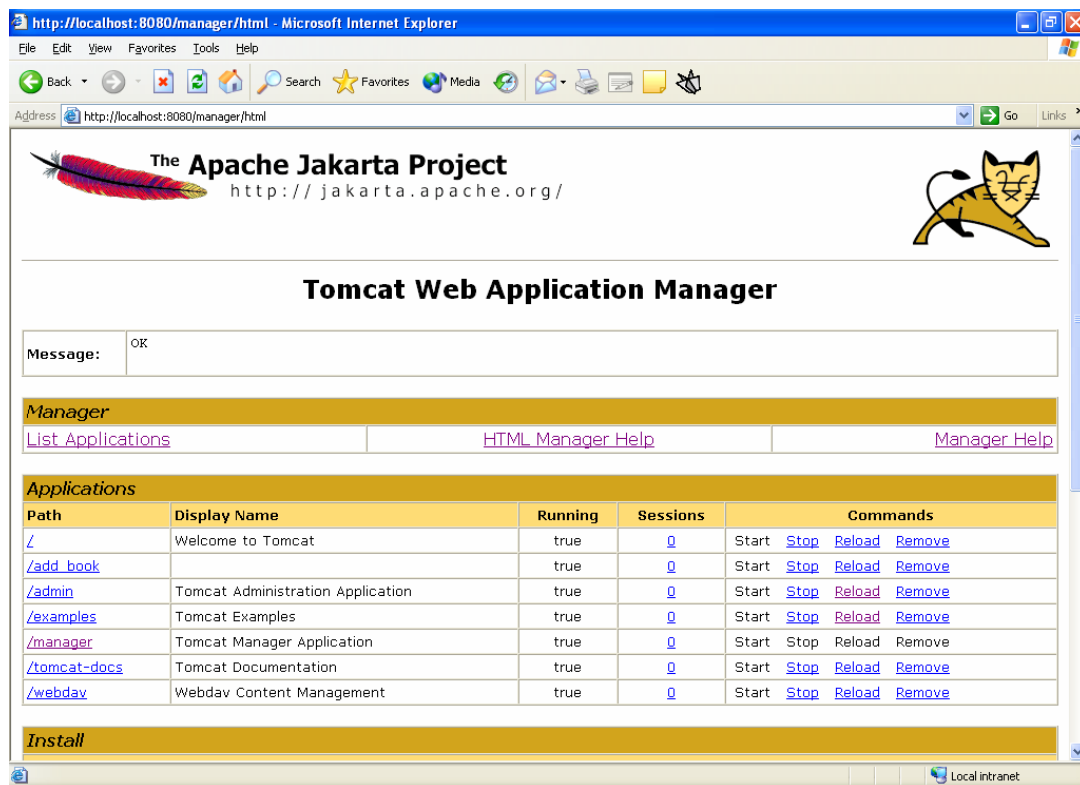
Untuk mengetahui user apa yang bisa mengakses direktori manager dan password dari user tersebut bisa dilihat daftar user pada file users.xml atau tomcat-users.xml. Sebagai contoh isi file tersebut adalah:

```
<?xml version='1.0' encoding='utf-8'?>  
<tomcat-users>  
  <role rolename="tomcat"/>  
  <role rolename="role1"/>  
  <role rolename="manager"/>  
  <role rolename="admin"/>  
  <user username="tomcat" password="tomcat" roles="tomcat"/>  
  <user username="role1" password="tomcat" roles="role1"/>  
  <user username="both" password="tomcat" roles="tomcat,role1"/>  
  <user username="admin" password="admin" roles="admin,manager"/>  
</tomcat-users>
```

Pada baris berikut dapat diketahui bahwa user admin dengan password admin dapat mengakses manager.

```
<user username="admin" password="admin" roles="admin,manager"/>
```

Dan berikut adalah tampilan dari URL <http://localhost:8080/manger/html>



Pada halaman di atas bisa dilihat “direktori kerja” yang telah dibuat, serta fasilitas untuk Stop, Reload dan Remove. Perintah Reload biasanya digunakan untuk me-refresh Servlet yang baru dimodifikasi, karena untuk melihat perubahan dari modifikasi sebuah Servlet tidak bisa dilakukan dengan cara me-refresh web browser.

Penutup

Seri tulisan Tips dan Trik JSP ini akan diberikan banyak contoh pembaca bisa langsung mencoba dan mengerti akan materi dan penjelasan yang diberikan. Materi yang akan diberikan diantaranya adalah :

- Tata cara penulisan pada JSP.
- Penggunaan taglib
- Koneksi ke database
- Implementasi java bean
- Komunikasi dengan servlet
- Dan lain-lain