

# JSP DAN JAVA SERVLET GENERATOR UNTUK APLIKASI DATABASE BERBASIS WEB DENGAN MENGIMPLEMENTASIKAN FRAMEWORK JAKARTA STRUTS

**Imam Kuswardayan, Nanik Suciati, Dody Rachmat Wicaksono**

Jurusan Teknik Informatika,  
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804  
Email: imam@its-sby.edu, nanik@its-sby.edu, dody@nicecoder.com

## ABSTRAK

Untuk meningkatkan produktifitas dalam pengembangan aplikasi, seringkali digunakan tool-tool untuk mempercepat proses pengkodean. Salah satu jenis tool yang mulai banyak digunakan adalah tool yang mampu mengkodekan sebuah aplikasi hanya dengan memberikan data-data logic, dan tanpa perlu melakukan pengkodean.

Dalam penelitian ini akan dibuat aplikasi code generator untuk bahasa java, khususnya JSP dan JavaServlet. Spesifikasi dari aplikasi yang akan dihasilkan oleh code generator ini adalah aplikasi yang memiliki arsitektur J2EE, yaitu dengan mengimplementasikan framework Jakarta Struts yang berdasarkan pada blue print Model View Controller. Framework Jakarta Struts akan bekerja sama dengan framework iBATIS yang akan menangani Persistent Database Layer. Kedua framework akan bekerja bersama-sama membangun pondasi yang kokoh dalam membangun aplikasi berarsitektur J2EE.

Uji coba dilakukan dengan sebuah skenario pembuatan aplikasi PIM (Personal Info Manager). Hasil uji coba menunjukkan bahwa aplikasi JSP dan Servlet Generator berjalan dan berfungsi sesuai dengan tujuan yang ditetapkan.

**Kata Kunci :** Struts, iBATIS, J2EE, MVC, Servlet, JSP, code generator.

## 1. PENDAHULUAN

Terdapat banyak pilihan dalam pengembangan aplikasi berbasis web. Sun Java mengeluarkan teknologi *JavaServer Pages* (JSP) dan *Java Servlet*. JSP dan Servlet yang tergabung didalam pake J2EE ditujukan untuk pengembangan aplikasi berbasis web dengan skala *enterprise*.

Pengembangan aplikasi berbasis web pada platform java bukan lah hal yang mudah dan dapat dengan cepat dapat diselesaikan, terutama pada saat melakukan pengkodean. Sering kali pengkodean dalam java merupakan sesuatu yang berulang-ulang dan memiliki persamaan karakteristik, misalnya pada saat pembuatan *java bean*, yang merukan sebuah *mapping* dari struktur data. Seharusnya pengkodean kode-kode seperti ini dapat dipersingkat dengan adanya sebuah tool *generator*.

Aplikasi yang akan dibuat dalam penelitian ini adalah aplikasi yang dapat membantu pengembangan aplikasi berbasis web dengan mengikuti metodologi MVC yang mampu membantu *developer* untuk mempersingkat waktu pengkodean, yaitu dengan cara mengkodekan kode-kode JSP dan Java Servlet,

beserta file-file xml penunjangnya. *Source code* yang dihasilkan akan menerapkan framework *Jakarta Struts* yang merupakan implementasi dari metodologi MVC.

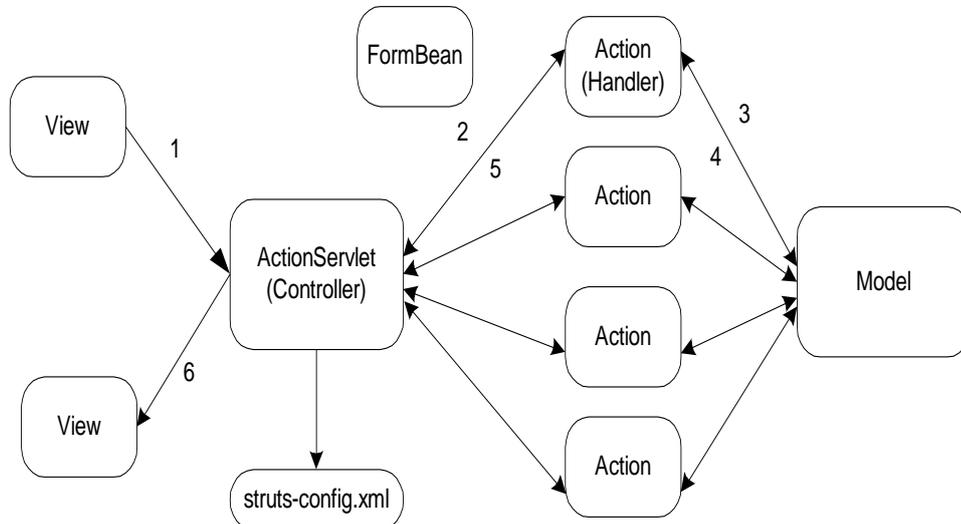
## 2. STRUTS APPLICATION FRAMEWORK

Struts Application Framework merupakan open source web application framework yang berdasarkan paradigma Model-View-Controller (MVC), yang biasanya dikenal dengan nama Model 2. Struts mengintegrasikan Java Servlet, JSP, Java Bean serta dilengkapi dengan flow control, error handling, resource-management serta internationalization support.

Dalam *Struts Application Framework*, flow dari aplikasi diatur oleh *controller* terpusat (*ActionServlet*). *Controller* akan mendelegasikan *request*, dalam hal ini *HTTP request*, ke *handler* (*Action*) yang sesuai. Setiap *handler* berperan sebagai *adapter* antara *request* dan model. Model sendiri merepresentasikan atau mengenkapsulasi *business logic* atau *state*. Setelah proses pada *business logic* selesai, flow dilanjutkan kembali ke *handler* dan

*controller* untuk selanjutnya ditampilkan ke *view* sesuai. Alur pendelegasian ini ditentukan melalui

suatu *mapping* yang diletakkan dari file konfigurasi dari *struts*.



**Gambar 1** Cara kerja Struts

### 2.1. CONTROLLER

Bagian *controller* dari aplikasi difokuskan pada penerimaan *request* dari klien (biasanya berupa *web browser*) untuk menentukan fungsi bisnis apa yang harus dilakukan untuk selanjutnya didelegasikan ke fase berikutnya, yakni komponen *view*. Pada *struts*, komponen *controller* yang paling utama adalah *ActionServlet*. Konfigurasi *Servlet* ini dilakukan dengan cara mendefinisikan *ActionMapping* yang sesuai. *ActionMapping* menentukan *path* yang bersesuaian dengan *request URI* dari *request* yang diterima, dan biasanya juga menentukan *Action class* yang nantinya memproses *request* tersebut. Semua *Action* merupakan subclass dari `org.apache.struts.action.Action`. *Action* akan mengenkapsulasi meneruskan *request* ke model, menampung hasilnya untuk selanjutnya meneruskan *control* ke komponen *view* yang bersesuaian.

### 2.2. MODEL

Model pada aplikasi biasanya berupa *class* yang berfungsi untuk merepresentasikan *object-object* dalam *web* aplikasi. *Object* ini nantinya dapat disimpan dalam *page*, *request*, *session* dan *application scope* untuk diproses lebih lanjut. Maksud dari masing-masing *scope* dapat dijelaskan sebagai berikut :

- a. *Page Scope*  
Artinya object tersebut hanya dapat diakses pada halaman JSP yang bersangkutan saja.
- b. *Request Scope*  
Artinya object tersebut hanya dapat diakses pada halaman JSP yang bersangkutan dan juga *Servlet*

atau halaman JSP lain di-include serta menerima forward pada suatu request.

- c. *Session Scope*  
Artinya object tersebut dapat diakses pada semua halaman JSP maupun *Servlet* yang untuk suatu user session tertentu. Tidak terbatas pada satu request saja, asalkan ada dalam satu session maka Bean tersebut masih dapat diakses.
- d. *Application Scope*  
Artinya object tersebut dapat diakses pada semua halaman JSP maupun *Servlet* pada suatu web application, tidak terbatas pada suatu user atau session tertentu.

Dalam implementasinya, salah satu dari model komponen ini biasanya berupa *ActionForm*. *ActionForm* merupakan *class* bawaan dari *Struts* yang merepresentasikan *form-form* pada *web application*. *ActionForm* ini dapat digunakan untuk populate data pada *form* serta melakukan validasi. Implementasi model yang lain dapat berupa *class-class ValueObject* atau jika diintegrasikan dengan EJB maka model biasanya dapat berupa *statefull* maupun *stateless session bean*. Dalam penelitian ini digunakan framework iBATIS untuk menangani akses database.

### 2.3. VIEW

Bagian *view* pada aplikasi biasanya dibuat dengan menggunakan teknologi *JavaServer Pages (JSP)*. Halaman JSP dapat berisi *static HTML text* maupun *content* dinamis. JSP juga dilengkapi dengan beberapa *taglibs* standar seperti yang tercantum dalam *JavaServer Pages Specification*, misalnya :

<jsp:useBean>, <jsp:include> dan sebagainya. Disamping *taglibs* standar tersebut, JSP juga mensupport *custom taglibs* yang dibuat oleh developer sesuai dengan kebutuhan dari aplikasi.

*Struts* menyediakan beberapa *taglibs* tambahan yang sangat berguna dalam membangun *user interface* yang mendukung *internationalization* juga memudahkan interaksi dengan *ActionForm Bean* yang termasuk dalam lingkup *Model Component*.

Beberapa *taglibs* tambahan yang telah disediakan oleh *Struts* dapat dijelaskan sebagai berikut :

- a. *Bean Tags*  
Taglib ini sangat berguna untuk mengakses *Bean* beserta peopertinya, dan juga memudahkan untuk membuat *Bean* baik dalam scope *request*, *session* maupun *application*.
- b. *HTML Tags*  
Taglib ini sangat berguna untuk membuat *Struts* input form dan juga tag-tag HTML lainnya yang biasa digunakan dalam membuat halaman web.
- c. *Logic Tags*  
Taglib ini biasa digunakan untuk dalam melakukan perulangan atau perbandingan untuk suatu *bean*.
- d. *Template Tags*  
Taglib ini biasa berguna untuk hal-hal yang berhubungan layout JSP, terutama jika menggunakan template.

#### 2.4. STRUTS GENERATOR

Aplikasi yang dibuat dalam penelitian ini diberi name *Struts Generator*. Sesuai dengan namanya, aplikasi ini memiliki kemampuan untuk mengkodekan source code JSP dan Servlet dengan mengikuti framework *struts* yang telah dijelaskan di atas. Context diagram dari *Struts Generator* bisa dilihat pada gambar 2.

*Struts Generator* dibuat dengan menggunakan *Visual Basic 6*. Dalam penelitian ini juga dibutuhkan perangkat lunak pendukung, yaitu:

- a. Ant  
<http://ant.apache.org>
- b. Tomcat  
<http://jakarta.apache.org/tomcat/index.html>
- c. MySQL  
<http://www.mysql.com>

*Struts Generator* menggunakan pendekatan lebih ke sisi developer web pada umumnya daripada mendekati ke developer java. Hampir semua jenis masukan merupakan istilah-istilah yang umum dijumpai oleh developer web, dan sebisa mungkin menghilangkan istilah-istilah dalam java. Dengan demikian diharapkan *Struts Generator* dapat digunakan tidak hanya oleh developer java, namun

dapat juga digunakan oleh developer web pada umumnya, dan bagi yang sedang mempelajari bagaimana *Struts* bekerja. Alur kerja *Struts Generator* bisa dilihat di gambar 3.

#### 3. ARSITEKTUR APLIKASI WEB YANG DIHASILKAN

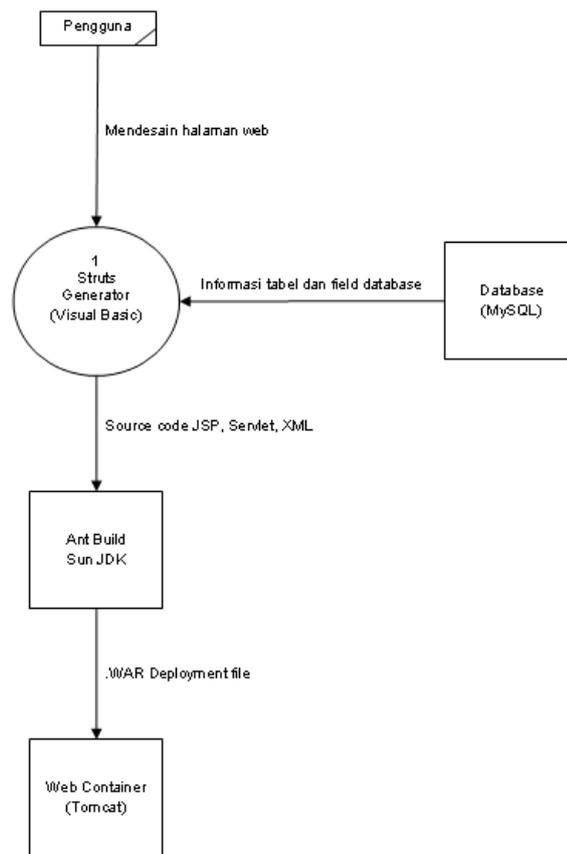
Aplikasi yang dihasilkan oleh *Struts Generator* akan memiliki arsitektur seperti pada gambar 4. Aplikasi akan terbagi menjadi 5 layar, yaitu:

- a. *Presentation Layer*  
Hal-hal yang berkenaan dengan layout dari aplikasi web masuk pada layer ini. Diantaranya adalah HTML, JSP, Javascript, CSS, gambar, dan juga taglib yang berfungsi untuk *formatting* data.
- b. *Application Layer*  
*Application layer* merupakan penghubung antara presentasi dan *services layer* ini bertugas untuk mengatur *client state (HTTPSession)*, melakukan validasi pada input client, dan melakukan delegasi ke *service layer*.
- c. *Service Layer*  
*Service layer* merupakan titik utama untuk akses ke framework ibatis. *Application layer* akan menggunakan layer ini untuk menjalankan suatu fungsi bisnis tertentu. *Service layer* selanjutnya akan meneruskannya pada satu atau beberapa *domain object* yang terkait dengan fungsi tersebut.
- d. *Domain Layer*  
*Domain layer* merupakan layer dimana semua obyek yang muncul pada masalah bisnis berada. Obyek-obyek ini direpresentasikan dalam bentuk *JavaBean*. *Service layer* sebagian besar mendelegasikan *request* yang diterimanya ke *domain layer* untuk diproses lebih lanjut.
- e. *Persistence Layer*  
*Persistence layer* adalah layer dimana *logic* yang berkaitan representasi domain pada database berada. Pada layer inilah operasi-operasi yang berkaitan dengan database dilakukan, seperti : *insert*, *delete*, *update* maupun *select*. Layer ini akan diimplementasikan dengan menggunakan framework iBATIS yang mencakupi pembuatan *DAO* dan *sqlmapping*.

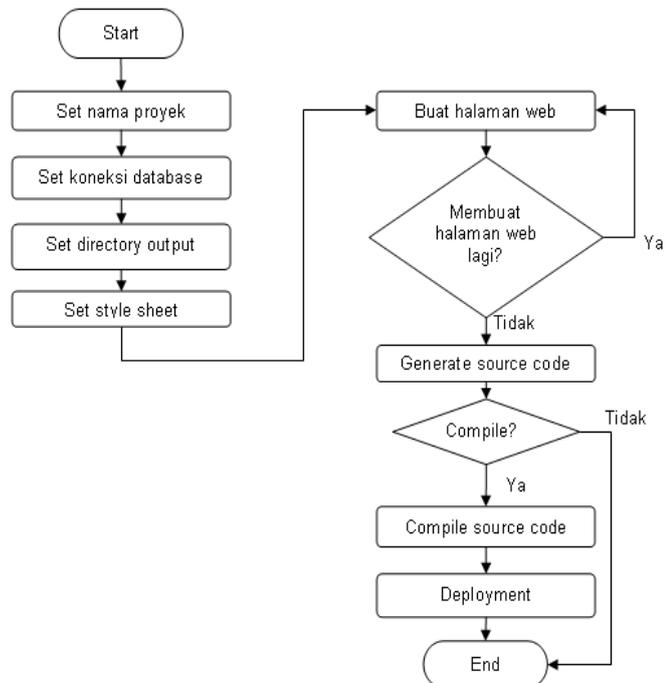
#### 4. PENGKODEAN

Sebelum melakukan pengkodean source code, terdapat langkah persiapan yang harus dijalankan sebelumnya, yaitu

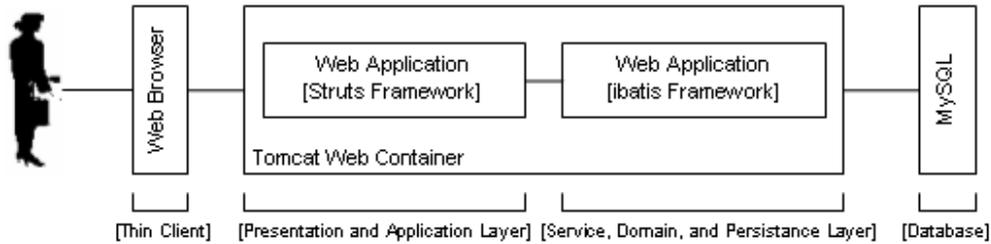
- a. Membuat struktur directory
- b. Menyalin file-file library dan file penunjang lainnya



**Gambar 2** Context Diagram Struts Generator



**Gambar 3** Alur Kerja Struts Generator



**Gambar 4 Arsitektur Aplikasi Web**

Struktur directory yang akan dibuat adalah seperti sebagai berikut:

- \\${project\_name}\META-INF
- \\${project\_name}\WEB-INF
- \\${project\_name}\WEB-INF\src
- \\${project\_name}\WEB-INF\lib
- \\${project\_name}\WEB-INF\src\com
- \\${project\_name}\WEB-INF\src\com\struts-gen
- \\${project\_name}\WEB-INF\src\com\struts-gen\\${project\_name}
- \\${project\_name}\WEB-INF\src\com\struts-gen\\${project\_name}\conf
- \\${project\_name}\WEB-INF\src\com\struts-gen\\${project\_name}\sql

Berikut adalah file-file library yang perlu disalin pada directory yang telah dibuat di atas.

- \\${project\_name}\error.jsp
- \\${project\_name}\META-INF\MANIFEST.MF
- \\${project\_name}\WEB-INF\struts-html-el.tld
- \\${project\_name}\WEB-INF\c.tld
- \\${project\_name}\WEB-INF\struts-bean-el.tld
- \\${project\_name}\WEB-INF\struts-logic-el.tld
- \\${project\_name}\WEB-INF\web.xml
- \\${project\_name}\WEB-INF\validator-rules.xml
- \\${project\_name}\WEB-INF\lib\commons-collections.jar
- \\${project\_name}\WEB-INF\lib\struts.jar
- \\${project\_name}\WEB-INF\lib\jstl.jar
- \\${project\_name}\WEB-INF\lib\standard.jar
- \\${project\_name}\WEB-INF\lib\commons-logging.jar
- \\${project\_name}\WEB-INF\lib\commons-digester.jar
- \\${project\_name}\WEB-INF\lib\commons-beanutils.jar
- \\${project\_name}\WEB-INF\lib\struts-el.jar
- \\${project\_name}\WEB-INF\lib\ibatis-common-2.jar
- \\${project\_name}\WEB-INF\lib\ibatis-sqlmap-2.jar
- \\${project\_name}\WEB-INF\lib\commons-validator.jar

Terdapat berbagai banyak class dan file-file xml penunjang yang harus dikodekan. Untuk komponen view, diperlukan pengkodean source code JSP. Untuk komponen controller, diperlukan pengkodean source code untuk class FormBean, Action, validator.xml, dan struts-config.xml. Sedangkan untuk komponen model, diperlukan pengkodean class Data Transfer Object (DTO), Service, Data Access Object (DAO), file database.properties, sql-map-config, dan file sql mapping dalam bentuk xml untuk setiap tabel database.

Dalam pengkodean source code file-file tersebut diatas, diperlukan sebuah file template untuk membantu proses pengkodean. Berdasarkan file template ini lah kode akan dihasilkan.

Berikut adalah file-file template yang dibutuhkan untuk menghasilkan source code untuk file-file di atas.

- DaoException.java
- BaseDAO.java
- DAO.java
- DTOBean.java
- Service.java
- FormBean.java
- InputAction.java
- SetUpEditAction.java
- EditAction.java
- DeleteAction.java
- ListAction.java
- main.css
- InputFormOk.jsp
- InputForm.jsp
- ListForm.jsp
- EditForm.jsp
- EditFormOk.jsp
- DeleteOk.jsp
- build.xml
- ApplicationResources.properties
- validation.xml
- struts-config.xml

Untuk mengilustrasikan bagaimana file template dapat bekerja untuk membantu proses pengkodean, berikut adalah contoh file template, yaitu untuk pembuatan class DTO.

```

package com.strutsgen.\${project_name};
import java.io.Serializable;
public class \${table} implements Serializable
{
    \${fields}
    \${getter_setter}
}
  
```

File template di atas memiliki 4 variabel yaitu \\${project\_name}, \\${table}, \\${fields}, dan \\${getter\_setter}. Seluruh variabel ini akan digantikan dengan nilai sesungguhnya, sesuai dengan masukan yang diberikan user pada saat melakukan desain. Berikut adalah class DTO yang dihasilkan dari template di atas. Teknik yang sama juga diterapkan dalam pengkodean file-file yang lain.

```

package com.strutsgen.pim;
import java.io.Serializable;
  
```

```

public class contact implements Serializable
{
    private String contactId;
    private String name;
    private String address;
    private String country;
    private String email;
    private String yahoo;
    private String msn;
    private String icq;
    private String website;

    public void setContactId(String s) {
        contactId = s;
    }
    public String getContactId() {
        return contactId;
    }
    public void setName(String s) {
        name = s;
    }
    public String getName() {
        return name;
    }
    public void setAddress(String s) {
        address = s;
    }
    public String getAddress() {
        return address;
    }
    public void setCountry(String s) {
        country = s;
    }
    public String getCountry() {
        return country;
    }
    public void setEmail(String s) {
        email = s;
    }
    public String getEmail() {
        return email;
    }
    public void setYahoo(String s) {
        yahoo = s;
    }
    public String getYahoo() {
        return yahoo;
    }
    public void setMsn(String s) {
        msn = s;
    }
    public String getMsn() {
        return msn;
    }
    public void setIcq(String s) {
        icq = s;
    }
    public String getIcq() {
        return icq;
    }
    public void setWebsite(String s) {
        website = s;
    }
    public String getWebsite() {
        return website;
    }
}

```

## 5. KOMPILASI DAN DEPLOYMENT

Untuk keperluan kompilasi dibutuhkan dua buah program external yaitu Sun Java SDK dan Apache Ant. Java SDK diperlukan untuk melakukan kompilasi file source code java menjadi file java binary code. Sedangkan Apache Ant digunakan untuk mengkonfigurasi keperluan kompilasi, seperti menentukan letak class library, dan file output tujuan.

Untuk dapat menggunakan Apache Ant, dibutuhkan file build.xml yang berisi konfigurasi prosedur kompilasi. File build.xml ini dibuat oleh oleh Struts Generator dan kemudian dijalankan sehingga didapat hasil berupa file .war untuk keperluan deployment.

Pada penelitian ini, digunakan Tomcat Server, yang mana mendukung proses auto-deploy dan remove invocation untuk mengatur life-cycle aplikasi web yang terinstall. Untuk melakukan deployment, Struts Generator akan menempatkan file .war yang dihasilkan pada folder /webapps. Dan apabila diperlukan, dilakukan proses undeployment aplikasi yang lama sehingga proses auto-deploy dapat terjadi.

Berikut adalah cara untuk memanggil fungsi undeploy dari build.xml

```

<target name="undeployweb"
depends="clean,prepare,compile">
<get
src="http://localhost:8080/manager/html/undep
loy?path=/pim"
dest = "tomcat.html"
verbose = "true"
UserName = "admin"
password = "admin"/>
</target>

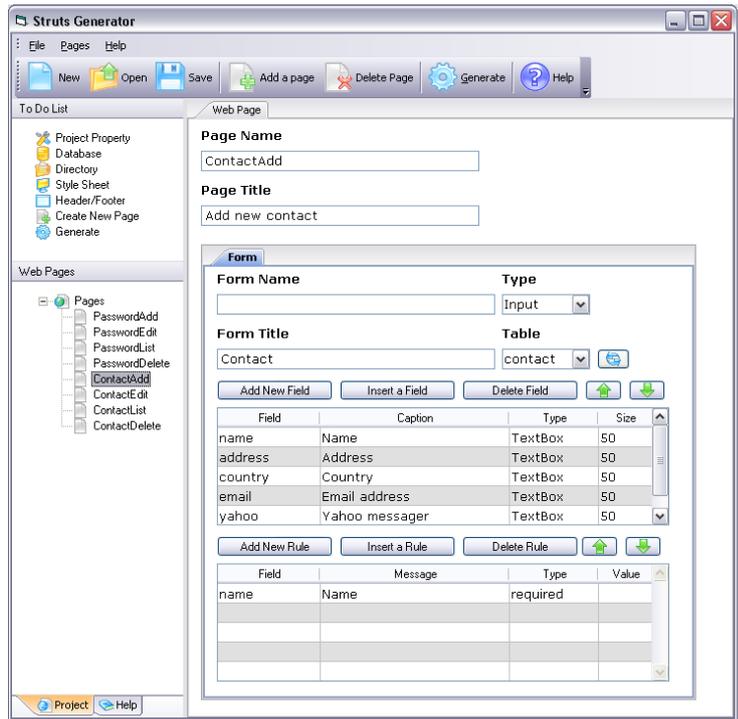
```

## 6. HASIL UJI COBA

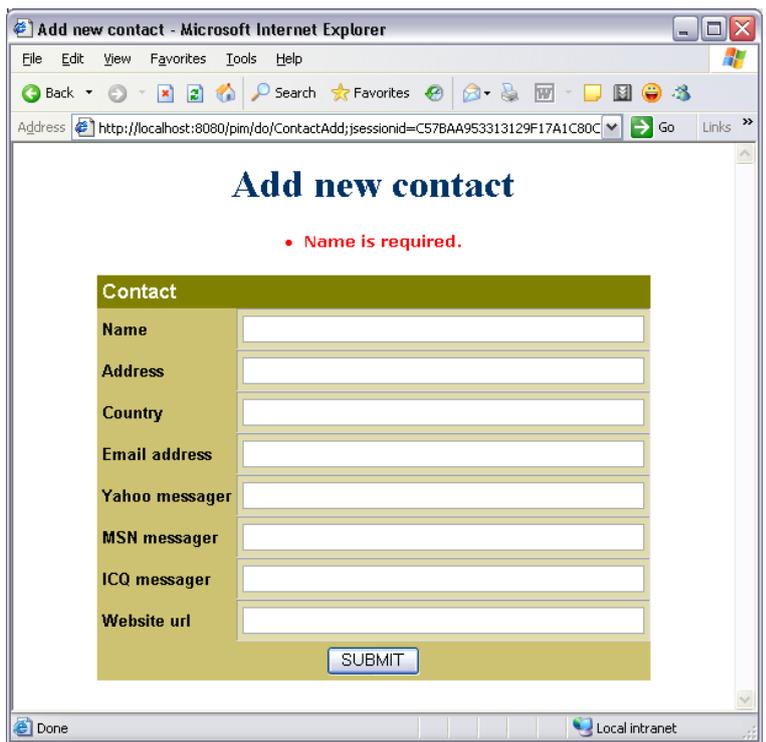
Dalam uji coba ini mengambil contoh skenario pembuatan aplikasi PIM (Personal Info Manager) yang di dalamnya berisi manajemen kontak dan manajemen password. Operasi yang dapat dilakukan oleh aplikasi PIM ini adalah menambah, mengubah, melihat daftar, dan menghapus kontak maupun password. Contoh tampilan hasil uji coba program bisa dilihat pada gambar 5 sampai dengan gambar 8.

Dari hasil uji coba dapat dievaluasi hal-hal sebagai berikut :

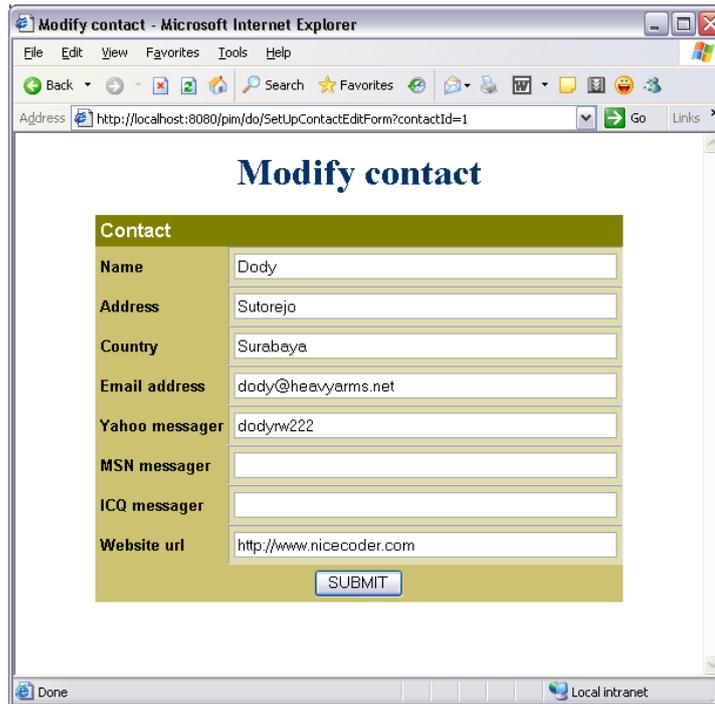
- Fitur-fitur *JSP dan Servlet Generator* secara umum berjalan dengan baik sesuai dengan yang diharapkan.
- JSP dan Servlet Generator* mampu melakukan kompilasi secara langsung dengan memanfaatkan Sun JDK dan Apache Ant yang telah terinstall sebelumnya.



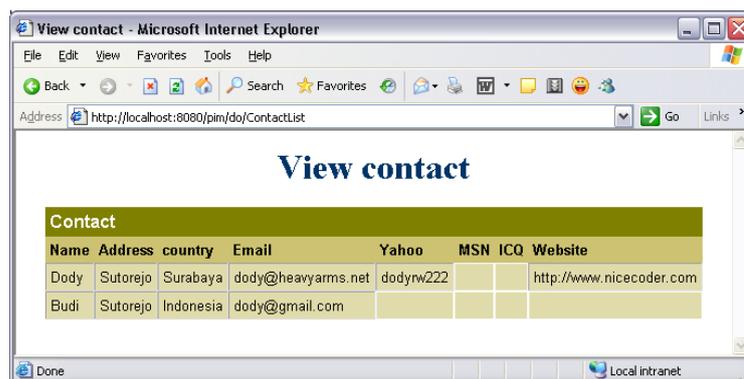
**Gambar 5 Mendesain halaman web**



**Gambar 6 Hasil pengkodean yang telah dijalankan, halaman Add New Contact**



Gambar 7. Hasil pengkodean yang telah dijalankan, halaman Modify Contact



Gambar 8. Hasil pengkodean yang telah dijalankan, halaman View Contact

- c. *JSP dan Servlet Generator* mampu menghasilkan kode JSP dan Servlet tanpa menimbulkan *error* pada saat kompilasi.
- d. Aplikasi yang dihasilkan oleh *JSP dan Servlet Generator* dapat dengan lancar dijalankan secara langsung, tanpa perlu mengubah kode program.
- e. *JSP dan Servlet Generator* masih memiliki banyak keterbatasan dalam membangun website yang lebih kompleks, baik keterbatasan dalam membangun antar muka maupun dalam menangani database yang kompleks.

## 7. KESIMPULAN DAN SARAN

Dari aplikasi yang telah dibuat beserta uji coba yang telah dilakukan terhadapnya, dapat diambil kesimpulan sebagai berikut:

Kesimpulan yang dapat ditarik dari tugas akhir ini adalah sebagai berikut :

- a. *JSP dan Servlet Generator* dapat menghasilkan source code JSP dan Servlet dengan menerapkan

framework Jakarta Struts.

- b. *JSP dan Servlet Generator* dapat melakukan pembuatan layout halaman secara otomatis.
- c. Aplikasi yang dihasilkan oleh *JSP dan Servlet Generator* dapat melakukan validasi data.
- d. Framework iBATIS yang bekerja pada layer *model* dapat diintegrasikan dengan Framework Jakarta Struts yang bekerja pada layer *view* dan *controller*.

## 8. DAFTAR PUSTAKA

1. James Turner dan Kevin Bedell, "Struts Kick Start", Sams Publishing, 2002.
2. Chuck Cavaness, "Programming Jakarta Struts 2<sup>nd</sup> Edition", O'Reilly, 2004.
3. Jeff Heaton, "JSTL: JSP Standard Tag Library Kick Start", O'Reilly, 2002.