

# PERANCANGAN PROTOTIPE SISTEM INFORMASI BERBASIS JAVA EE 7 UNTUK SISTEM KP DAN TA TEKNIK ELEKTRO UNIVERSITAS DIPONEGORO

Bagus Satrio U. PH.<sup>\*)</sup>, Maman Somantri, and Sudjadi

Jurusan Teknik Elektro, Universitas Diponegoro Semarang  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*)</sup>E-mail : bagussat66@gmail.com

## Abstrak

Sistem pemrosesan informasi manual yang dulunya merupakan satu – satunya cara dalam memproses informasi semakin ditinggalkan dan tergantikan oleh sistem informasi berbasis komputer. Namun, tidak semua sistem informasi yang berjalan berfungsi dengan baik dan optimal. Bertujuan untuk memberi rancangan prototipe pengembangan dari sistem KP dan TA Elektro Universitas Diponegoro yang tergolong dalam sistem yang disebutkan sebelumnya, diangkat judul penelitian ini. Menggunakan JSR 345: Enterprise JavaBeans sebagai jantung dari teknologi Java EE, sistem akan lebih baik dalam menanggapi beban client dan juga lebih baik pada sisi user experience, yang gagal dicapai sistem sebelumnya. Sistem informasi yang diimplementasikan menggunakan rancangan arsitektur 3-tier berbasis web dan merupakan aplikasi yang web berdiri sendiri pada server tunggal. Dengan penggunaan framework Java Enterprise Edition dari platform Java Web yang sudah teruji tangguh pada kelas Enterprise. Sistem Informasi yang dibangun tentu akan terjamin berjalan dengan baik dan teruji bebas dari masalah. Pada segi user experience, sisi ini tetap dijaga dengan membuat sistem yang mudah digunakan, eye-catching dan memiliki dukungan HTML5, CSS3, serta AJAX yang modern, yang diliputi pada JSR 127: Java Server Faces pada Presentation Layer. Singkatnya, sistem informasi ini merupakan sistem yang memiliki fitur massive, dengan tetap menjaga kemudahan penggunaan untuk user dari sistem.

*Kata Kunci: Sistem Informasi, Pengembangan Web Enterprise, Java EE, 3 Tier.*

## Abstract

Manual information processing system which on the past is the one and only solution, turns to be impractical and getting replaced by the computer-based ones. However, not all of this information systems that runs are functioning and well optimized. Therefore, intended for providing prototype designs of KP and TA systems on Electrical Engineering Department of Diponegoro University which is considered belongs to the systems mentioned before, this title is taken. With JSR 345: Enterprise JavaBeans as the core of Java EE technology, the system will be better at managing client loads and also scores better on the user experience, which failed to achieve on the past system. The implemented information system uses web-based 3-tier architecture design and poses as a standalone web application hosted on a singular server. With the use of Java Enterprise Edition framework of the Java Web platform, which is tested as robust and stable framework. On the user experience side of the system, is kept by making the system easy to use, eye-catching, and having support of modern HTML5, CSS3, and AJAX, under JSR 127: Java Server Faces on the presentation layer. Shortly described, this information system is a system with massive features, but also keeping track for the ease of use of the system.

*Keywords: Information System, Enterprise Web Development, Java EE, 3 Tier.*

## 1. Pendahuluan

Dengan berkembangnya teknologi *internet*, informasi semakin mudah didapatkan. Segala hal yang berhubungan dengan informasi semakin terotomatisasi dan pemrosesan informasi manual semakin tergantikan dengan pemrosesan berbasis web atau online yang lebih singkat dalam proses dan dapat diakses dari mana saja.

Dikhususkan pada Universitas Diponegoro, tempat penulis melakukan kegiatan akademis. Pemrosesan informasi berbasis web juga terbukti praktis. Sistem – sistem informasi yang sudah berjalan saat ini jelas memudahkan mahasiswa, seperti Sistem Informasi Akademik yang menjadi bagian penting dalam proses akademik mahasiswa.

Dan pada pada kasus khusus yang perlu disorot, Sistem Informasi KP dan TA Teknik Elektro Universitas

Diponegoro menggunakan CMS (*Content Management System*) berupa *Wordpress* yang tujuan utamanya adalah untuk menulis dan display entri *Weblog (blog)*.

Dasar dari penulisan penelitian ini sendiri adalah untuk menunjukkan suatu percontohan *upgrade* yang tidak hanya bisa dilakukan pada infrastruktur jaringan atau perangkat keras *server*. Hal tersebut juga bisa dilakukan pada sisi aplikasi, yang merupakan tujuan paling utama dari *enterprise web development*.

Tujuan dari penulisan penelitian ini adalah perancangan protipe aplikasi sistem informasi untuk Sistem KP dan TA Teknik Elektro Universitas Diponegoro, dengan mengimplementasi bermacam komponen dan *tools* dari *framework Java EE 7* yang diperlukan untuk membangun sistem.

Adapun pembatasan masalah pada penelitian ini adalah sebagai berikut :

1. Hanya membahas perancangan dan implementasi *software / perangkat lunak* tentang Sistem Informasi KP dan TA Teknik Elektro Universitas Diponegoro.
2. Terbatas pada platform aplikasi *Java Web* menggunakan *Java EE 7*, yang merupakan pemrograman *object-oriented* dan *web-based*.
3. Hanya membahas pada lingkup teknologi *Java EE* yang digunakan, yaitu *Java Persistence API (JPA)*, *Enterprise JavaBeans (EJB)* hanya pada *Session Beans*, *Java Server Faces (JSF)* dan *Oracle Glassfish Application Server* pada *Developer Profile*.
4. Untuk teknologi diluar *Java EE*, hanya membahas *JavaMail API*, *PrimeFaces* dan *JasperReports*.

## 2. Metode

### 2.1 Metodologi Perancangan

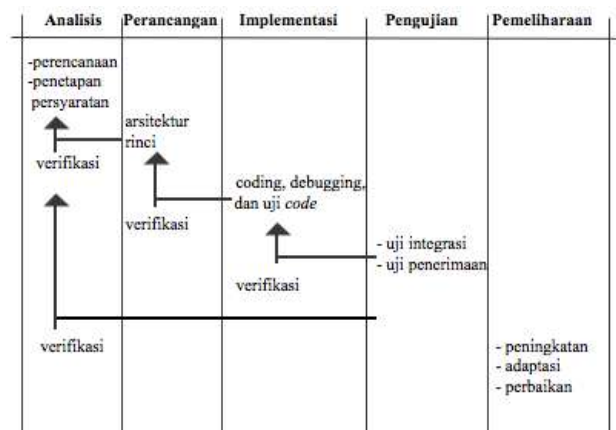
Dalam perencanaan sistem dengan alasan sistem dibangun dari awal, penulis menggunakan model fase<sup>[2]</sup> untuk merancang aplikasi. Model ini membagi *life cycle* ke dalam sederetan kegiatan (fase). Setiap fase membutuhkan informasi masukan, proses, dan produk yang terdefinisi dengan baik. Deretan fase tersebut adalah : analisa, perancangan, implementasi, pengujian, dan pemeliharaan. Model fase dasar yang dinyatakan sebagai *waterfall chart* ditunjukkan pada **gambar 1**.

### 2.2 Perancangan Arsitektur Sistem

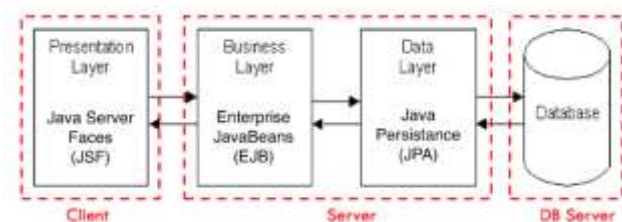
Secara garis besar, aplikasi sistem informasi yang dibuat merupakan aplikasi *web* berbasis *Java EE* yang berdiri sendiri. Oleh karena itu sistem dirancang menggunakan *server* tunggal, yaitu *server* aplikasi yang ditangani oleh *Oracle Glassfish Application Server*, diikuti dengan *server* basis data yang terpisah dari aplikasi untuk menangani penyimpanan data dari aktifitas *business logic*. *Server* aplikasi di sini memiliki fungsi dasar untuk *hosting* halaman *web* dan fungsi lain yang lebih penting yakni

menyediakan *container* untuk memungkinkan fitur yang spesifik pada *container* untuk bisa digunakan pada aplikasi sistem informasi. Fitur yang paling penting dari *container* dari *server* aplikasi yang menjadi fokus di sini adalah fungsi *business logic* dari *Enterprise JavaBeans* yang hanya bisa disediakan pada *container EJB* dan *container Web*.

Aplikasi yang dirancang menggunakan arsitektur 3 tier, yang terdiri dari *Data Layer*, *Business Layer*, serta *Presentation Layer*. Secara singkat arsitektur dari sistem dapat dilihat pada **Gambar 2**.



Gambar 1. Life-cycle mode fase perangkat lunak



Gambar 2. Arsitektur 3-tier aplikasi Java EE

### 2.3 Perancangan Basis Data

Untuk mencapai basis data yang memiliki standar *software engineering* yang baik dengan tujuan untuk mempermudah proses *development*, setelah melalui proses analisa sistem basis data akan memiliki rancangan seperti yang ditunjukkan pada *Entity Relationship Diagram* di **Gambar 3**.

### 2.4 Perancangan Model Fungsional Use Case Diagram

*Use case*<sup>[2]</sup> digunakan untuk merepresentasikan fungsionalitas dari sistem. *Use case* berfokus pada sifat sistem dari sudut pandang luar sistem. Sebuah *use case* mendeskripsikan fungsi yang disediakan sistem dan memiliki hasil yang diterima oleh actor. Dan actor pada

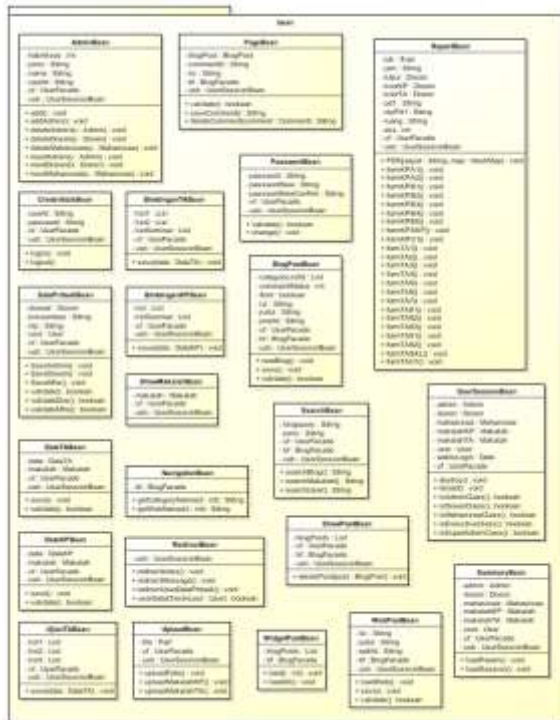


## 2.7 Perancangan Model Objek pada Presentation Layer

Package bean pada sistem menyimpan keseluruhan Managed bean dari aplikasi, yang memiliki fungsi Model dari JSF pada Presentation Layer. Setiap class pada package bean memiliki fungsi tersendiri, dari yang sekedar menampilkan data, hingga merubah data, yang menjadikan Managed bean sebagai perantara business transaction yang dilakukan oleh Facade.

Setiap class ini rata – rata berhubungan dengan satu view, atau bisa dibilang satu halaman web, karena class ini mengatur konten dinamis dari halaman web tersebut. Konten yang dinamis tersebut bisa berubah, tergantung dari parameter / atribut yang tersimpan pada request dan juga session.

Hal yang paling sering dijumpai pada class Managed bean adalah UserFacade dan BlogFacade yang didapatkan dari package Facade atau Business Layer. Dua atribut Facade ini didapatkan melalui injeksi, sehingga atribut ini bisa melakukan business transaction pada Managed bean tersebut karena sudah ter-injek properti EJB.

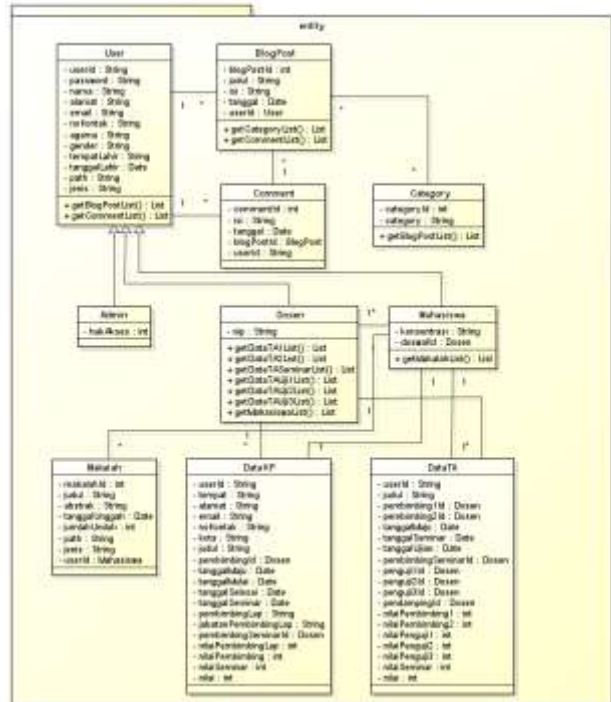


Gambar 7. Class Diagram pada presentation layer aplikasi Sistem Informasi

## 2.8 Perancangan Model Objek pada Data Layer

Pada aplikasi rancangan, sistem menggunakan API ORM yang memetakan tabel basis data menjadi objek. ORM

yang digunakan, tepatnya JPA langsung memetakan tabel basis ke objek, serta meng-generate atribut maupun objek dengan nama yang sesuai standar objek pada pemrograman berorientasi objek. Class diagram dari class entity yang di-generate oleh JPA dari tabel basis data dapat dilihat lebih jelas pada Gambar 8.



Gambar 8. Class Diagram pada data layer aplikasi Sistem Informasi

## 2.9 Perancangan Antarmuka

Presentation Layer ini memiliki fungsi utama untuk menerjemahkan tugas dan hasil menjadi sesuatu yang dapat dimengerti user. Oleh karena itu, Presentation Layer aplikasi ini menyediakan rancangan antarmuka atau user interface seperti yang ditunjukkan pada gambar 9.

## 3. Hasil dan Analisa

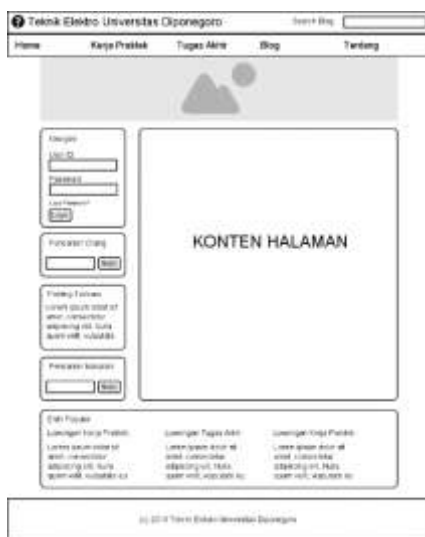
### 3.1 Implementasi Data Layer

Sesuai dengan perancangan pada Data Layer, sistem diimplementasikan menggunakan API Object-Relational Mapping (ORM) yang memetakan tabel basis data menjadi objek. ORM yang digunakan, tepatnya Java Persistence API (JPA) langsung memetakan tabel basis ke objek, serta meng-generate atribut maupun objek dengan nama yang sesuai standar objek pada pemrograman berorientasi objek. Contoh kode ditunjukkan pada class User dibawah ini:

```
@Entity
public class User implements Serializable {
    private static final long
    serialVersionUID = 1L;
```

```
@Id @Basic(optional = false)
@NotNull
@Size(min = 1, max = 20)
@Column(name = "user_id")
protected String userId;
@Size(max = 20)
private String password;
...
}
```

Entity Class yang diimplementasikan secara garis besar merupakan class yang *Serializable* dan memiliki atribut yang diwakili oleh *private variable*, dengan ditemani *public setter* dan *getter* untuk atribut tersebut.



Gambar 9. Rancangan Antarmuka Sistem Informasi

### 3.2 Implementasi Business Layer

Pada aplikasi, *business transaction* pada *business layer* dilaksanakan oleh *EJB Session Bean*. Dan dikarenakan *EJB* pada aplikasi hanya digunakan oleh aplikasi tunggal yang berada pada satu *server* yang sama, *EJB* tersebut merupakan *Local EJB* yang hanya bisa diakses *client EJB* pada *server* yang sama. Contoh kode ditunjukkan pada class *UserFacade* dibawah ini:

```
@Stateless
@LocalBean
public class UserFacade {
    @PersistenceContext(unitName =
    "KPTA_TEPU")
    EntityManager em;
    public User findUser(String userId) {
        return em.find(User.class, userId);
    }
    ...
}
```

Sebuah *EJB session bean* memerlukan *EntityManager* sebagai pelaku dari *business transaction*. Untuk bisa memfungsikan *EntityManager* tersebut, diperlukan injeksi

*Persistence Unit* pada *EntityManager*, dan *Persistence Unit* yang digunakan pada aplikasi adalah "KPTA\_TEPU" seperti yang sudah dipersiapkan pada *persitence.xml*.

### 3.3 Implementasi Presentation Layer

Halaman *Web* yang merupakan bagian dari *presentation layer* yang diatur oleh *JSF* memiliki dua komponen inti, yaitu halaman *web XHTML* dan *Managed Bean*. Contoh kode *managed bean* ditunjukkan pada class *UserSessionBean* dibawah ini:

```
@ManagedBean(name = "userSession")
@SessionScoped
public class UserSessionBean implements
Serializable{
    private User user;
    private Mahasiswa mahasiswa;
    private Admin admin;
    private Dosen dosen;
    @EJB
    private UserFacade uf;
    public void reload();
    public void setUser(User user);
    public void destroy();
    ...
}
```

*Managed Bean* berfungsi sebagai *Model* dari *JSF*. Sebuah *Managed Bean* memiliki akses ke sesama *Managed Bean* serta *EJB* pada *Business Layer*. *Managed Bean* memiliki lingkup yang menentukan sejauh apa *managed bean* tersebut masih berlaku. Selain *UserSessionBean*, rata – rata *managed bean* pada sistem hanya mencapai lingkup *request*.

Selanjutnya akan ditampilkan halaman *master* yang selalu digunakan sebagai *template* pada setiap halaman di sistem informasi, yang ditunjukkan pada gambar 10.

Halaman *master* yang berfungsi sebagai kerangka dasar dari setiap halaman web pada aplikasi menggunakan fitur *Facelets Templating* dari *JSF*. *Template* pada aplikasi terbagi menjadi dua tingkat / *level*. Dimana pada *level* dasar, *template* berisikan *tag HTML5* yang berfungsi sebagai struktur dasar dari *web* dan dihubungkan dengan *script CSS3* untuk memberikan *styling* pada *web*. Sedangkan pada *level 2*, seperti yang ditunjukkan pada gambar 10, *template* hanya merakit halaman – halaman *web* dari *directory includes* yang berlaku sebagai komponen *modular* dari aplikasi ke *template* dasar.

### 3.4 Pengujian Sistem

Metode pengujian yang digunakan untuk menguji sistem adalah suatu metode pengujian yang bertujuan memastikan sistem yang diuji menjalankan seluruh fungsinya dengan baik. Metode pengujian tersebut adalah metode pengujian *use case*, atau *use case testing*. *Use*

case testing<sup>[1]</sup> menjelaskan fungsi apa yang *end-user* dapatkan tanpa mengetahui cara kerja *internal* sistem, dan merupakan salah satu metode dari *black box testing*.

Selain menjalankan *use case testing* yang merupakan pengujian fungsional, penulis juga menjalankan serangkaian pengujian non-fungsional untuk menguji apakah sistem yang diimplementasikan juga sudah memenuhi kebutuhan yang berkaitan dengan *internal* sistem.



Gambar 10. Struktur Halaman Master Template

Berdasarkan serangkaian pengujian yang sudah dilakukan, didapatkan hasil sebagai berikut:

Tabel 1 Hasil Pengujian Use Case Actor Mahasiswa

No.	Use Case yang diuji	Hasil Pengujian
1.	Registrasi Data Mahasiswa	Tidak terdapat <i>error</i> .
2.	Registrasi Data KP	Tidak terdapat <i>error</i> .
3.	Registrasi Makalah KP	Tidak terdapat <i>error</i> .
4.	Registrasi Data TA	Tidak terdapat <i>error</i> .
5.	Registrasi Makalah TA	Tidak terdapat <i>error</i> .

Tabel 2 Hasil Pengujian Use Case Actor Dosen

No.	Use Case yang diuji	Hasil Pengujian
1.	Registrasi Data Dosen	Tidak terdapat <i>error</i> .
2.	Edit Nilai KP Mahasiswa	Tidak terdapat <i>error</i> .
3.	Edit Nilai TA Mahasiswa	Tidak terdapat <i>error</i> .
4.	Edit Nilai Ujian TA Mahasiswa	Tidak terdapat <i>error</i> .

Tabel 3 Hasil Pengujian Use Case Actor Admin

No.	Use Case yang diuji	Hasil Pengujian
1.	Registrasi Data Admin	Tidak terdapat <i>error</i> .
2.	Manajemen Mahasiswa / Dosen	Tidak terdapat <i>error</i> .
3.	Manajemen Admin	Tidak terdapat <i>error</i> .

4.	Edit Halaman Web	Tidak terdapat <i>error</i> .
----	------------------	-------------------------------

Tabel 4 Hasil Pengujian Use Case Actor User Umum

No.	Use Case yang diuji	Hasil Pengujian
1.	Search User / Makalah	Tidak terdapat <i>error</i> .
2.	Display Data User	Tidak terdapat <i>error</i> .
3.	Display Makalah Mahasiswa	Tidak terdapat <i>error</i> .

Tabel 5 Hasil Pengujian Use Case fungsi Blog

No.	Use Case yang diuji	Hasil Pengujian
1.	Search Posting Blog	Tidak terdapat <i>error</i> .
2.	Display Posting Blog	Tidak terdapat <i>error</i> .
3.	Edit / Posting Blog Baru	Tidak terdapat <i>error</i> .
4.	Komentar Blog	Tidak terdapat <i>error</i> .

Tabel 6 Hasil Pengujian Non-Fungsional

No.	Fitur yang diuji	Hasil Pengujian
1.	Keberadaan Session	Session terbukti ada.
2.	Proteksi Direktori	Direktori terproteksi programatis.
3.	Kompresi File Upload	Melalui <i>resize</i> untuk file citra.
4.	Simplified Searching	Implementasi <i>search</i> relatif sederhana.
5.	Form Mahasiswa PDF	Form / Surat memiliki format PDF.
6.	Fasilitas Lupa Password	Ada dan dilengkapi konfirmasi email.

Tabel 7 Hasil Perbandingan dengan Sistem yang Existing

No.	Fitur dibandingkan	Hasil Perbandingan
1.	Fungsional Mahasiswa	Tidak ada fungsi hilang, dan terdapat fitur baru.
2.	Fungsional Dosen	Merupakan Fitur baru.
3.	Fungsional Admin	Terdapat fungsi hilang, namun juga terdapat fitur baru.
4.	Fungsional Blog	Memiliki fungsi lebih.
5.	Fungsional Search dan Display Makalah / User	Merupakan Fitur Baru.
6.	Non-Fungsional	Merupakan perbaikan Sistem existing.

Dari rekapitulasi seluruh pengujian yang dilakukan pada sistem, tidak terdapat satu pun hambatan yang ditemui saat pengujian. Oleh karena itu, dapat disimpulkan bahwa aplikasi yang diimplementasikan lulus pengujian. Namun untuk perbandingan, biarpun sistem relatif lebih baik dibandingkan sistem *existing*, terdapat fungsi hilang pada sisi fungsional admin.

#### 4. Kesimpulan

Berdasarkan keseluruhan penelitian dari perancangan, implementasi dan pengujian dapat disimpulkan bahwa aplikasi sistem informasi ini dikembangkan menggunakan Arsitektur 3-Tier berbasis *web* dengan menggunakan framework Java EE 7, yang terdiri dari tiga komponen inti, yaitu; EJB, JPA, dan JSF.

Lebih lanjut lagi, setiap *layer* dari aplikasi *web* dengan arsitektur 3-tier diatur oleh komponen – komponen

*framework* dari *Java EE* yang terpisah. Namun, setiap komponen tersebut memiliki kapabilitas untuk saling berkomunikasi pada *layer* yang saling berdekatan.

Pada sisi fungsional sistem, terdapat tiga *actor* pada aplikasi sistem informasi ini, yaitu Admin, Mahasiswa, dan Dosen. Ketiga *actor* ini memiliki sifat dan hak khusus yang hanya dimiliki oleh *actor* tersebut. Namun, ketiga *actor* ini memiliki kesamaan sifat, seperti yang dimiliki oleh *parent actor* dari ketiga *actor* ini, yaitu *User*.

Adapun saran yang dapat diberikan untuk menjadi masukan pada penelitian lebih lanjut yaitu pengembangan lebih lanjut menggunakan teknologi *Web Service* pada *middleware* atau *business layer*, sehingga *resource* tertentu dari aplikasi bisa ikut diakses oleh aplikasi lain yang terdapat pada lingkungan yang terpisah.

Disarankan juga untuk mahasiswa lain supaya bisa mengambil judul penelitian yang bisa menerapkan fitur – fitur dari *Java EE 7* yang masih belum terliput dalam penelitian ini, tanpa melupakan *business process* dari aplikasi yang dibuat.

## Referensi

- [1]. Ammann, P., dan Jeff O. 2008. *Introduction to Software Testing*. Fairfax: Cambridge University Press.
- [2]. Bruegge, B., dan Dutoit, A. H. 2009. *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Edisi ke 3. New Jersey: Prentice Hall.
- [3]. Goncalves, A. 2013. *Beginning Java EE 7*. New York City: Apress.
- [4]. Jacobi, J. dan Fallows, J. R. 2006. *Pro JSF and AJAX*, New York City: Apress.
- [5]. Keith, M. dan Merrick S. 2013. *Pro JPA 2*, Edisi ke 2. New York City: Apress.
- [6]. Kou, X. 2010. *GlassFish Administration*, Birmingham: Packt Publishing.
- [7]. Papagelis, M. 2012. *Web App Architectures*. *Computer Science University of Toronto*. (Online). (<http://queens.db.toronto.edu/~papaggel/courses/csc309/docs/lectures/web-architectures.pdf> , diakses 31 Desember 2013)
- [8]. Winterbee, J., dkk. 2013. *Beginning EJB 3*, Edisi *Java EE 7*. New York City: Apress.