

# PERANCANGAN PROTOTIPE SISTEM INFORMASI AKADEMIK TEKNIK ELEKTRO UNIVERSITAS DIPONEGORO MENGGUNAKAN KERANGKA KERJA SPRING WEB MVC

Ibnu Fauzan Al Aula<sup>\*)</sup>, Maman Somantri, dan Yuli Christyono

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

<sup>\*)</sup>E-mail: [ibnu\\_fauzanalaula@yahoo.co.id](mailto:ibnu_fauzanalaula@yahoo.co.id)

## Abstrak

Teknik Elektro Universitas Diponegoro merupakan sebuah instansi akademik di mana terdapat berbagai macam kegiatan akademik yang melibatkan banyak pihak baik mahasiswa, dosen, maupun staf-staf. Kegiatan akademik tersebut tidak terlepas dari kebutuhan penyebaran informasi antarpihak untuk menunjangnya. Oleh karena itu, dibutuhkan suatu cara atau teknologi yang dapat menunjang kegiatan akademik di Teknik Elektro Universitas Diponegoro agar penyebaran informasi dapat berlangsung dengan baik. Salah satu solusinya adalah dengan membuat Sistem Informasi Akademik (SIA). Pada penelitian ini, dirancang suatu aplikasi web bernama Prototipe Sistem Informasi Akademik Teknik Elektro UNDIP. Aplikasi tersebut digunakan oleh empat macam pengguna, yaitu mahasiswa, dosen, petugas akademik, dan admin. Aplikasi ini dirancang dengan menggunakan teknologi Java EE dan kerangka kerja Spring Web MVC. Selain itu, basisdata yang digunakan adalah MySQL. Pengujian yang dilakukan menggunakan metode *whitebox* dengan jenis pengujian unit. Hasil pengujian menunjukkan bahwa semua kelas yang berhubungan dengan fitur inti berhasil diuji dengan benar.

Kata kunci: Sistem Informasi Akademik, Java EE, Spring Web MVC, MySQL, pengujian whitebox

## Abstract

Department of Electrical Engineering Diponegoro University is an academic institution where there are many academic activities that involve many people including students, lecturers, and staff. The academic activities do need the spreading of information to support them. Therefore, a method or technology which can support academic activities in Department of Electrical Engineering Diponegoro University is needed, in order to spread information properly. Building Academic Information System is one solution that can resolve the problem. In this research, a web application named *Prototipe Sistem Informasi Akademik Teknik Elektro UNDIP* is built. The application is used by four types of user, those are student, lecturer, academic staff, and administrator. It is built by using Java EE technology and Spring Web MVC framework. In addition, MySQL is used as database. The testing was done by using whitebox method, specifically unit testing. The result of the testing showed that all classes that are related to main features, were tested successfully.

*Keywords: Academic Information System, Java EE, Spring Web MVC, MySQL, whitebox testing*

## 1. Pendahuluan

Teknik Elektro Universitas Diponegoro merupakan sebuah instansi akademik di mana terdapat berbagai macam kegiatan akademik yang melibatkan banyak pihak baik mahasiswa, dosen, maupun staf-staf. Pesatnya perkembangan teknologi mempengaruhi cepatnya penyebaran informasi. Informasi merupakan komponen penting bagi sebuah instansi, terlebih lagi instansi akademik seperti Teknik Elektro Universitas Diponegoro. Maka dari itu, dibutuhkan suatu cara agar

penyebaran informasi di Teknik Elektro Universitas Diponegoro berjalan dengan lancar. Salah satunya adalah dengan cara merancang sebuah Sistem Informasi Akademik (SIA).

Adapun tujuan penelitian ini adalah untuk merancang Prototipe Sistem Informasi Akademik Teknik Elektro Universitas Diponegoro berbasis Java EE (*Enterprise Edition*) dengan menggunakan kerangka kerja Spring Web MVC.

Penelitian ini dibatasi pada beberapa masalah sebagai berikut :

1. Hanya membahas perancangan SIA dari segi aplikasi.
2. Pengguna SIA terdiri dari mahasiswa, dosen, staf akademik, dan admin.
3. Beberapa mekanisme kegiatan akademik disederhanakan dari mekanisme yang sebenarnya.
4. Fitur-fitur SIA lebih terfokus pada kegiatan akademik mahasiswa.
5. Tidak ada pembagian konsentrasi jurusan pada fitur SIA.
6. Menggunakan teknologi Java EE 6 Web.
7. Aplikasi web servernya adalah Apache Tomcat.
8. DBMS yang digunakan adalah MySQL.
9. Pengujian menggunakan metode *whitebox* dengan jenis pengujian unit, dan hanya dilakukan pada fitur inti kegiatan akademik.

## 2. Metode

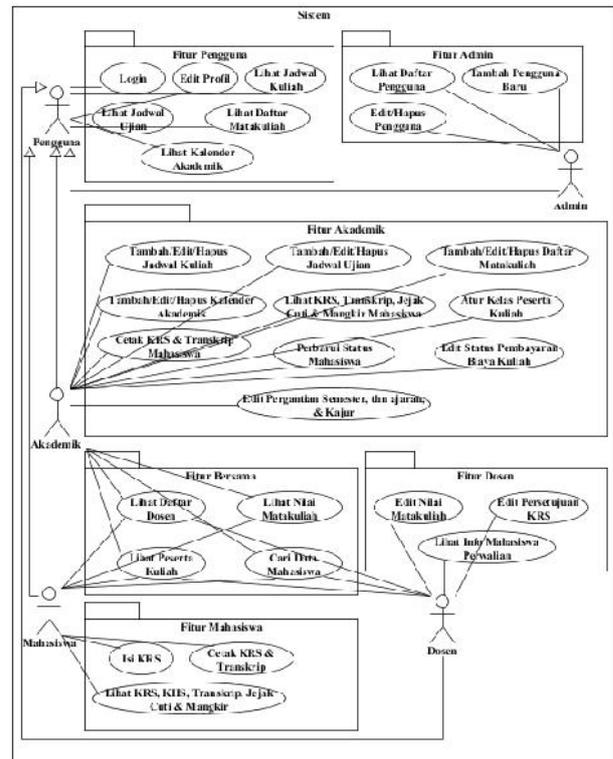
Pada penelitian ini, sistem dibangun pada teknologi Java EE dengan menggunakan kerangka kerja Spring Web MVC. Aplikasi dibuat dengan pola MVC (*Model View Controller*) sehingga komponen-komponen aplikasi terbagi menjadi *model* (model data), *view* (tampilan), dan *controller* (pengatur jalannya program). Untuk menyimpan data secara permanen, digunakan My SQL sebagai basis data.

Aplikasi SIA Elektro digunakan oleh empat macam pengguna yaitu mahasiswa, dosen, petugas akademik, dan admin. Fitur-fitur pada aplikasi SIA Elektro dibagi menjadi fitur inti dan fitur tambahan. Fitur inti merupakan fitur yang berhubungan dengan kegiatan inti akademik mahasiswa. Sedangkan fitur tambahan adalah fitur yang menunjang berlangsungnya kegiatan inti.

### 2.1 Perancangan Diagram Use Case

Diagram *use case* menggambarkan fungsi-fungsi yang ada pada sistem. Diagram ini lebih berfokus pada fitur-fitur sistem dari sudut pandang pihak luar, yang dalam hal ini adalah pengguna aplikasi. Gambar 1 menunjukkan diagram *use case* SIA Elektro.

Pada Gambar 1 terlihat bahwa fitur aplikasi dilihat dari sudut pandang pengguna dapat dibagi menjadi fitur pengguna, admin, akademik, dosen, mahasiswa, dan bersama. Selain itu, terdapat lima macam aktor yaitu pengguna, admin, akademik, dosen, dan mahasiswa. Aktor pengguna merupakan aktor induk dari aktor-aktor lainnya. Semua hal yang dapat dilakukan aktor induk, dapat pula dilakukan oleh aktor anak.



Gambar 1 Diagram use casesistem

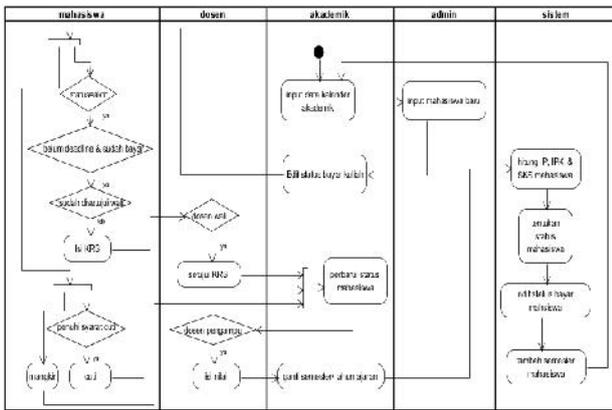
### 2.2 Perancangan Diagram Kelas

Diagram kelas menggambarkan komponen-komponen pembentuk sistem yang berupa kelas atau *interface*, dan hubungan antar komponen tersebut. Di aplikasi SIA Elektro ini, diagram kelas dibagi menjadi dua, yaitu diagram kelas fitur inti dan fitur tambahan. Diagram kelas menggambarkan komponen-komponen pembentuk sistem yang berupa kelas atau *interface*, dan hubungan antar komponen tersebut. Di aplikasi SIA Elektro ini, diagram kelas dibagi menjadi dua, yaitu diagram kelas fitur inti dan fitur tambahan. Diagram kelas fitur inti bisa dilihat pada Gambar 2, sedangkan diagram kelas fitur tambahan bisa dilihat pada Gambar 3.

Jika dilihat dari sisi proses kerjanya, kelas-kelas pada aplikasi SIA Elektro dapat dibagi menjadi kelas *controller*, DAO (*Data Access Object*), dan *service*. Kelas *controller* berfungsi sebagai pengatur jalannya aplikasi, kelas DAO berfungsi sebagai pengakses basis data, dan kelas *service* berfungsi sebagai *logic business*.

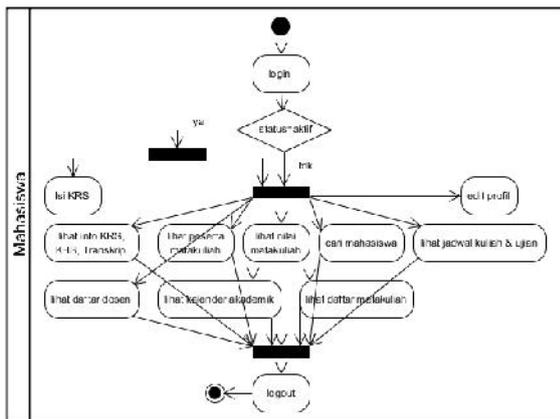
Pada Gambar 2, terlihat diagram kelas fitur inti aplikasi SIA Elektro. Kelas *UploadMhsBaruController* digunakan untuk menambahkan data mahasiswa baru, kelas *EditKalenderController* digunakan untuk menambah atau mengedit kalender akademik, kelas *BayarController* digunakan untuk mengedit status pembayaran biaya kuliah mahasiswa sebagai penyederhanaan dari proses pembayaran yang sesungguhnya, kelas *KrsController*





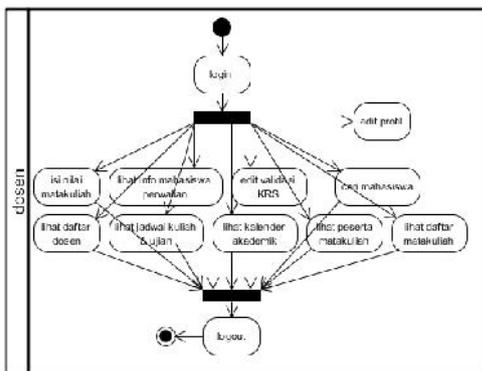
Gambar 4 Diagram aktivitas kegiatan inti akademik

Pada Gambar 5, dapat dilihat diagram aktivitas mahasiswa. Diagram ini menggambarkan aktivitas pengguna berjenis mahasiswa dari saat login ke aplikasi, mengakses fitur-fitur mahasiswa, sampai logout dari aplikasi.



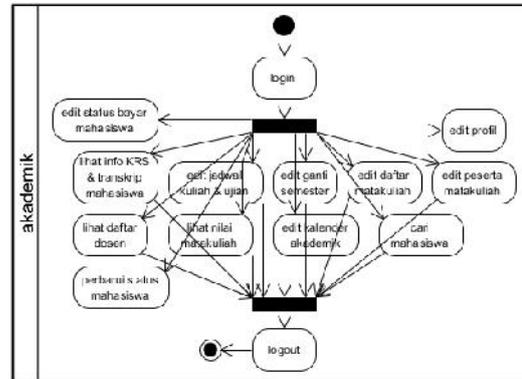
Gambar 5 Diagram aktivitas mahasiswa

Pada Gambar 6, dapat dilihat diagram aktivitas dosen. Diagram ini menggambarkan aktivitas pengguna berjenis dosen dari saat login ke aplikasi, mengakses fitur-fitur dosen, sampai logout dari aplikasi.



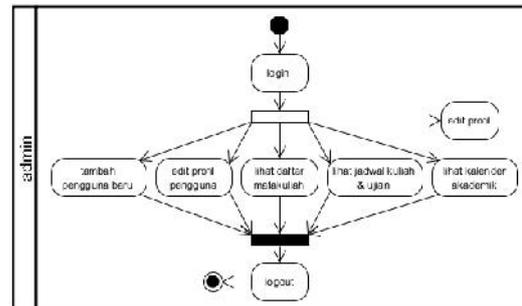
Gambar 6 Diagram aktivitas dosen

Pada Gambar 7, dapat dilihat diagram aktivitas akademik. Diagram ini menggambarkan aktivitas pengguna berjenis petugas akademik dari saat login ke aplikasi, mengakses fitur-fitur akademik, sampai logout dari aplikasi.



Gambar 7 Diagram aktivitas akademik

Gambar 8 menunjukkan diagram aktivitas admin. Diagram ini menggambarkan aktivitas pengguna berjenis admin dari saat login ke aplikasi, mengakses fitur-fitur admin, sampai logout dari aplikasi.



Gambar 8 Diagram aktivitas admin

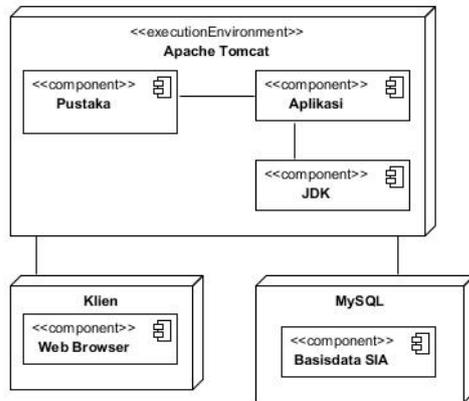
## 2.4 Perancangan Diagram Deployment

Diagram *deployment* menggambarkan arsitektur sistem yang dapat berupa konfigurasi komponen-komponen perangkat keras, atau konfigurasi komponen-komponen perangkat lunak dan perangkatnya. Diagram *deployment* aplikasi SIA Elektro dapat dilihat pada Gambar 9.

Secara garis besar, sistem terdiri dari klien, server web, dan server basisdata. Pada sisi klien terdapat aplikasi web browser untuk mengakses aplikasi SIA Elektro.

Pada sisi server web terdapat tiga komponen, yaitu pustaka, aplikasi, dan JDK. Komponen pustaka berisi pustaka-pustaka Java yang dibutuhkan dalam pembuatan aplikasi. Komponen aplikasi berisi komponen-komponen pembangun aplikasi. Sedangkan JDK merupakan komponen yang harus ada dalam pengembangan aplikasi Java. Aplikasi server web yang digunakan di sistem ini adalah Apache Tomcat.

Pada sisi server basisdata, dibuatlah basisdata SIA yang digunakan untuk menyimpan data-data aplikasi SIA Elektro. Server basisdata yang digunakan adalah MySQL.



Gambar 9 Diagram deployment sistem

### 3. Hasil dan Analisa

#### 3.1 Implementasi Konfigurasi Aplikasi

Implementasi berkas konfigurasi berisi tentang pembuatan dan pengaturan berkas-berkas untuk konfigurasi aplikasi dalam bentuk berkas XML. Berkas web.xml merupakan berkas yang berisi konfigurasi aplikasi web secara umum. Kode di bawah ini merupakan berkas web.xml.

```

<context-param>
<param-name>
contextConfigLocation
</param-name>
<param-value>
/WEB-INF/applicationContext.xml
</param-value>
</context-param>
<listener>
<listener-class>
org.springframework.web.context.ContextLoad
erListener
</listener-class>
</listener>
<servlet>
<servlet-name>
dispatcher</servlet-name>
<servlet-class>
org.springframework.web.servlet.DispatcherS
ervlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>dispatcher</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
...
<welcome-file-list>
<welcome-file>redirect.jsp
</welcome-file>
</welcome-file-list>
</web-app>
  
```

Padaberkas web.xml, diatur servlet dispatcher sebagai controller utama. pemetaan URL aplikasinya adalah “/”, dan halaman yang pertama kali diakses aplikasi adalah redirect.jsp.

#### 3.2 Implementasi Kelas Aplikasi

Implementasi kelas aplikasi berisi kelas-kelas Java yang digunakan untuk membangun aplikasi. Kode di bawah ini merupakan contoh implementasi kelas Java.

```

public class MasukController extends
SimpleFormController {
...
@Override
protected ModelAndView onSubmit(
...
) throws Exception {
ModelAndView mv= new
ModelAndView(getSuccessView());
HttpSession session =
request.getSession();
session.setMaxInactiveInterval(-1);
User user = (User) command;
List<User> users = new ArrayList<User>();
users = userDao.cekUser("admin",
user.getUsername());
if (!users.isEmpty()) {
users.get(0).setJenis("admin");
} else {
users = userDao.cekUser("aka",
user.getUsername());
if (!users.isEmpty()) {
users.get(0).setJenis("akademik");
} else {
users =
userDao.cekUser("dosen",
user.getUsername());
if (!users.isEmpty()) {
users.get(0).setJenis("dosen");
} else {
users =
userDao.cekUser("mhs", user.getUsername());
if (!users.isEmpty()) {
users.get(0).setJenis("mahasiswa");
String status =
userDao.dptStatus(users.get(0).getUsername(
));
session.setAttribute("status", " "+status);
}
}
}
session.setAttribute("menu",
"beranda");
session.setAttribute("user",
users.get(0));
return mv;
}
}
  
```

Method on Submit () berfungsi untuk menangani input data dari form. Sebuah sesi dibuat terlebih dahulu di method ini. Objek dari kelas User diambil dari form. Kemudian ditentukan jenis pengguna dengan bantuan methoduser Dao. Cek User() untuk mengakses data di basis data. Setelah itu, beberapa variabel dimasukkan ke sesi.

### 3.3 Implementasi Kelas Pengujian

Implementasi kelas pengujian berisi kelas-kelas yang digunakan untuk menguji kelas aplikasi secara unit. Kode di bawah merupakan contoh kelas pengujian untuk menguji kelas aplikasi.

```
public class TestMasukController {
    ...
    @Before
    public void setUp(){
    ...
    }
    @Test
    public void onSubmitTest() throws
    Exception{
        request.setParameter("username",
        "danang");
        request.setParameter("password",
        "danang1992");
        binder.bind(request);
        User u = new User();
        u.setUsername("danang");
        u.setPassword("danang1992");
        u.setKode("21060110141001");
        u.setJenis("mahasiswa");
        List<User> users = new ArrayList<User>();
        users.add(u);

        ModelAndView mv =
        masukController.onSubmit(request, response,
        user, new BindException(new Object(),
        "obj"));

        assertEquals("menuView",
        mv.getViewName());
        assertEquals(" aktif",
        session.getAttribute("status"));
        assertEquals(users.get(0),
        session.getAttribute("user"));
    }
}
```

Pada method on Submit Test(), ditambahkan beberapa parameter pada request sebagai pengganti input dari form. Pengujian dilakukan dengan membandingkan variabel hasil method on Submit() dengan nilai yang diharapkan. Perbandingan ini menggunakan method assertEquals().

### 3.4 Hasil Pengujian Sistem

Berdasarkan serangkaian pengujian yang sudah dilakukan pada tiap kelas fitur inti, didapatkan hasil seperti pada Tabell.

Tabel 1 Hasil pengujian sistem

Kelas	Method	sukses
Masuk Controller	On Submit	ya
	On Submit	ya
Edit Kalender Controller	Show Form	ya
	FormBacking Object	ya
Upload Mhs Baru Controller	On Submit	ya
	Tampil bayar	ya
Bayar Controller	Tampil dft mhs	ya
	simpanbayar	ya
	Isi krs	ya
Krs Controller	Tambah krs	ya
	Hapus krs	ya
	Simpan krs	ya
Edit mhs Controller	On Submit	ya
	Show Form	ya
	FormBacking Object	ya
Wali Controller	Dft wali	ya
	Simpan wali	ya
Status Mhs Controller	Tampil hal	ya
Upload Nilai Controller	Perbarui status	ya
Kurikulum Controller	On Submit	ya
	kurikulum	ya
Isi nilai Controller	Simpan kurikulum	ya
	Isi nilai	ya
Aksi Controller	Pilih makul	ya
	Nilai excel	ya
	keluar	ya
Something	Kalender aka	ya
	Hapus kalender	ya
	Enkripsi	ya
Something	Get Total SKS	ya
	Get Total Bobot	ya
Something	Get IP	ya

Berdasarkan Tabel 1, seluruh pengujian kelas-kelas yang berhubungan dengan fitur inti sistem berhasil dilakukan. Oleh karena itu, dapat disimpulkan bahwa fitur-fitur inti sistem telah terimplementasi dengan benar.

## 4. Kesimpulan

Kesimpulan yang didapat dari perancangan dan pengujian sistem adalah sebagai berikut:

1. Aplikasi SIA Elektro merupakan aplikasi web yang dibuat dengan menggunakan teknologi Java EE, dan memakai kerangka kerja Spring Web MVC.
2. SIA Elektro dapat digunakan oleh empat macam pengguna, yaitu mahasiswa, dosen, petugas akademik, dan admin.
3. Fitur-fitur pada SIA Elektro dibedakan menjadi fitur inti dan fitur tambahan.
4. Proses inti kegiatan akademik pada SIA Elektro meliputi penginputan kalender akademik, penginputan data mahasiswa baru, pengeditan status pembayaran biaya kuliah, pengisian KRS/cuti/mangkir, persetujuan KRS, pembaruan status mahasiswa, kegiatan belajar mengajar, pengisiannilai, dan pergantian semester.
5. Pada aplikasi SIA Elektro, digunakan Apache Tomcat sebagai server webnyadan MySQL sebagai server basisdatanya.

6. Konfigurasi aplikasi SIA Elektro dilakukan melalui berkas-berkas konfigurasi yang berupa web.xml, applicationContext.xml, dan dispatcher-servlet.xml.
7. Pengujian aplikasi dilakukan dengan menggunakan metode *whitebox* dengan jenis pengujian unit.
8. Semua kelas yang berhubungan dengan fitur inti sistem berhasil diuji dengan pengujian unit.

Saran untuk pengembangan selanjutnya adalah sebagai berikut:

1. Aplikasi dikembangkan dengan menggunakan kombinasi kerangka kerja lain, misalnya kerangka kerja pada sisi *view* dan *data access*.
2. Pengujian *whitebox* yang dilakukan tidak hanya pengujian unit saja, melainkan juga pengujian integrasi.

## Referensi

- [1]. Bawiskar, A, P. Sawant, V. Kankate, B.B. Meshram, "Spring Framework: A Companion to Java EE", *IJCEM International Journal of Computational Engineering & Management*, 15, 47, 2012.
- [2]. Bima, Ifnu., *Java Desktop Aplikasi POS Berarsitektur Three Tier Menggunakan Swing, Hibernate dan Spring*, Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License, California, 2011.
- [3]. Deinum Marten and Serneels Koen, *Pro Spring MVC: With Web Flow*, Apress, New York, 2012.
- [4]. J. L. Whitten and L. D. Bentley, *Systems Analysis and Design Methods 7th ed*, McGraw-Hill, New York, 2007.
- [5]. Ladd Seth, Davison Darren, Devijver Steven, and Yates Colin, *Expert Spring MVC and Web Flow*, Apress, USA, 2006.
- [6]. Murach Joel and Steelman Andrea, *Murach's Java Servlets and JSP 2nd ed*, Mike Murach & Associates, Inc, USA, 2008.
- [7]. Pitone Dan and Pitman Neil, *UML 2.0 in a Nutshell*, O'Reilly, USA, 2005.
- [8]. Schildt, Herbert, *C# 3.0: A Beginner's Guide*, The McGraw-Hill Companies, USA, 2009.
- [9]. ---, Peraturan Rektor Universitas Diponegoro Nomor: 209/PER/UN7/2012 tentang Peraturan Akademik Bidang Pendidikan Universitas Diponegoro.
- [10]. ---, "Spring Framework 3.1 Tutorial", Tutorialspoint.com.
- [11]. ---, "SQL Tutorial", Tutorialspoint.com.
- [12]. ---, *TIOBE Index For November 2014*, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>, November 2014.