

SISTEM INFORMASI KRS PADA UNIVERSITAS ISLAM OKI KAYUAGUNG MENGGUNAKAN METODE *SERVICE ORIENTED ARCHITECTURE (SOA)*

Susan Dian Purnamasari¹⁾, Muhammad Nasir²⁾, Irwan Syahputra³⁾

^{1,2,3)}Jurusan Sistem Informasi Universitas Bina Darma
Jl. Ahmad Yani no. 12 Palembang Telp (0711)-515679
e-mail : susandian@mail.binadarma.ac.id

Abstrak

Kegiatan yang paling menonjol pada suatu perguruan tinggi adalah pengolahan data, baik pengelolaan data administrasi mahasiswa sampai pengelolaan data akademisnya. Salah satu kegiatan untuk mendukung kegiatan pengolahan data akademis adalah pembuatan Kartu Rencana Studi (KRS). Pada Sistem Kartu Rencana Studi yang sedang berjalan saat ini pada Universitas Islam OKI (UNISKI) masih merupakan kegiatan yang manual. Dengan adanya rancangan Sistem Informasi Kartu Rencana Studi dengan menggunakan metode Service Oriented Architecture (SOA) akan mengintegrasikan proses bisnis dan mendukung infrastruktur IT yang aman. Service didefinisikan sesuai dengan kebutuhan spesifik bisnis yang dikelompokkan ke dalam proses-proses bisnis yang ada. Dengan membangun aplikasi berbasis SOA, maka aplikasi tersebut akan lebih mudah untuk diintegrasikan dan dimodifikasi apabila terjadi perubahan karena setiap service yang ada sudah memiliki standar yang dapat direuse dan digunakan antar domain fungsi yang berbeda.

Kata Kunci : Kartu rencana studi, SOA, Akademik, Service.

1. PENDAHULUAN

Pada setiap Universitas, pengisian Kartu Rencana Studi (KRS) merupakan kegiatan rutin yang selalu dilakukan pada awal semester ganjil maupun genap. Kartu Rencana Studi atau lebih dikenal dengan KRS merupakan kartu yang berisi daftar mata kuliah yang akan diikuti oleh setiap mahasiswa dalam setiap semester. Dalam KRS tercantum format data mahasiswa seperti : Nama, Nomor Induk Mahasiswa (NIM), Kelas, Fakultas, Jurusan, Jumlah Semester, Tahun Akademik, Kode Mata Kuliah, Nama Mata Kuliah, Bobot Sistem Kredit Semester (Bobot SKS), Ruang dan Waktu kuliah yang ditetapkan sebagai beban belajar semester bagi mahasiswa.

Untuk bisa mengetahui beban belajar semester berikutnya, mahasiswa harus berpatokan dengan Indeks Prestasi (IP) sebagai indikator yang digunakan untuk menentukan jumlah beban belajar yang dapat diambil mahasiswa dalam satu semester. Banyaknya beban belajar semester bergantung pada besarnya IP semester sebelumnya, kecuali mahasiswa baru yang jumlah beban belajar ditentukan. tergantung struktur kurikulum dan kebijakan Fakultas/Jurusan / Program Studi yang bersangkutan. Pada umumnya digunakan beban belajar antara 18-24 SKS sebagai paket semester atau patokan awal.

Dalam pengisian KRS secara manual, terkadang banyak ditemui kendala yaitu kurang telitian dalam pengisian *Form* KRS oleh mahasiswa dan ketidak telitian Penasehat Akademik dalam membimbing dan memeriksa isian *Form* KRS mahasiswa dan pada bagian BAAK semua berkas KRS mahasiswa akan diproses dengan sistem tumpukan (*batch processing*).

Dengan adanya perkembangan teknologi, terjadi pula perubahan-perubahan yang memanfaatkan perkembangan teknologi tersebut. Termasuk juga dalam pengisian KRS dapat di permudah dengan menggunakan metode *Service Oriented Architecture* (SOA) sebuah konsep arsitektur perangkat lunak yang mendefinisikan penggunaan layanan untuk memenuhi kebutuhan suatu perangkat lunak.

Penerapan SOA pada sistem memungkinkan sistem menyediakan fungsi yang dapat dipakai oleh sistem lain. Fungsi sistem harus mengakses basis data dan mengakomodasi proses bisnis pengentiran KRS di UNISKI sehingga dapat berinteraksi satu sama lain secara bebas tanpa perlu memperhatikan lokasi dan *platform* apa yang digunakan oleh *service* lainnya pada saat *run-time*. Dengan demikian sistem melibatkan banyak pengguna dan melibatkan jaringan komputer. Antarmuka sistem melibatkan antarmuka berbasis web untuk diakses melalui internet dan operator client untuk lingkungan intranet. Sistem KRS akan difokuskan untuk mengidentifikasi, membangun, mengubah dan memelihara proses bisnis KRS sebagai sekumpulan *service* yang akan dapat mempermudah mahasiswa dan bagian BAAK dalam proses pengentiran KRS.

2. TINJAUAN PUSTAKA

2.1. *Service oriented Architecture (SOA)*

Menurut Erl dalam Sarno (2012 : 25) Mendefenisikan "*Service Oriented Architecture (SOA)* adalah bentuk teknologi arsitektur yang mengikuti prinsip-prinsip *service-orientation*. Konsep *service-orientation* ini melakukan dengan membagi fungsionalitas yang besar menjadi sekumpulan *service* kecil dengan tujuan

tertentu". SOA merupakan salah satu pendekatan untuk memenuhi standar kebutuhan dan kualitas dari pembangunan sebuah perangkat lunak (*Software*).

Prinsipnya, SOA membungkus fungsionalitas sebuah sistem kedalam bentuk layanan layanan dengan demikian beragam aplikasi dan platform yang berbeda beda dapat saling terhubung dan melakukan pertukaran data tanpa harus terikat kuat satu sama lain (*loosely coupled*).

Komponen dari SOA (Erl, 2005) terdiri dari empat hal, yaitu *messages*, *operations*, *service*, dan *process*, yang akan dijabarkan sebagai berikut :

1. *Message*, yaitu data yang dibutuhkan untuk menyelesaikan sebagian atau sebuah unit kerja, yang dipertukarkan antara satu *service* dengan yang lainnya.
2. *Operation*, yaitu fungsi-fungsi yang dimiliki oleh sebuah *service* untuk memproses *message* hingga menghasilkan sesuatu. Fungsi-fungsi inilah yang nantinya akan saling berinteraksi untuk menyelesaikan sebuah unit kerja.
3. *Service*, merepresentasikan sekumpulan *operation* yang berhubungan untuk menyelesaikan sekumpulan unit kerja yang berhubungan.
4. *Process*, merupakan *business rule* yang menentukan operasi mana yang digunakan untuk mencapai tujuan tertentu

Keempat komponen tersebut terhubung satu sama lain dengan deskripsi sebagai berikut :

- a. Sebuah *operation* mengirim dan menerima *message* untuk mengerjakan sesuatu
- b. Sebuah *operation* didefinisikan oleh *message* yang memprosesnya
- c. Sebuah *service* mengelompokkan sekumpulan *operation* yang saling berhubungan.
- d. Sebuah *service* di definisikan oleh *operation* yang membentuknya
- e. Sebuah *instance* dari *service* dapat mengkomposisi *service* lain
- f. Sebuah *instance* dan proses tidak harus didefinisikan oleh *service* karena mungkin hanya membutuhkan sebagian dari fungsionalitas yang diberikan oleh *service*
- g. Sebuah *instance* dari proses memicu sekumpulan *operation* berjalan untuk menyelesaikan proses otomatisasi.

2.2. Web Service

Menurut Sarno (2012 : 32) "*Web Service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web Service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web Service* mampu menyimpan data informasi XML sehingga data dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*. *Web Service* bertujuan untuk meningkatkan *kolaborasi* antar pemrograman dan instansi yang memungkinkan sebuah fungsi didalam *web service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat didalamnya. *Web Service* tersusun dari beberapa komponen yang semuanya berbasis XML yaitu SOAP dan WSDL.

2.3. Extensible Markup Language (XML)

Menurut Lucky (2008 : 8) "XML merupakan dasar yang penting atas terbentuknya *web service*. *Web service* dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilnya dengan menggunakan XML. Kenapa menggunakan XML ? Karena XML berbentuk teks sehingga mudah untuk ditransportasikan menggunakan protokol HTTP. Selain itu, XML juga bersifat *platform* independen sehingga informasi didalamnya bisa dibaca oleh aplikasi apapun pada platform apapun selama aplikasi tersebut mampu menerjemahkan *tag-tag* XML.

2.4. Simple Object Access Protocol (SOAP)

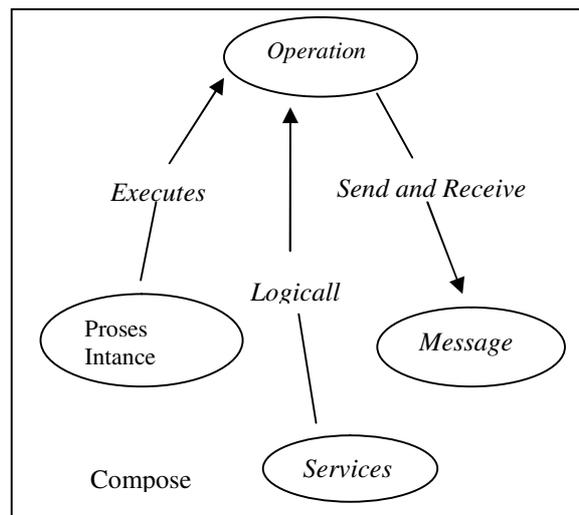
Menurut Sarno (2012 : 36) "*Simple Object Access Protocol* (SOAP) merupakan suatu protokol berbasis xml yang digunakan untuk kebutuhan pertukaran informasi dalam suatu sistem terdistribusi dan terdesentralisasi, seperti halnya IIOP (pada COBRA), ORCP (pada DCOM), dan JRMP (pada RMI)".

3. METODE PENELITIAN

3.1. Metode SOA.

Prinsipnya, SOA membungkus fungsionalitas sebuah sistem kedalam bentuk layanan layanan dengan demikian beragam aplikasi dan platform yang berbeda beda dapat saling terhubung dan melakukan pertukaran data tanpa harus terikat kuat satu sama lain (*loosely coupled*).

Komponen dari SOA (Erl, 2005) terdiri dari empat hal, yaitu *messages*, *operations*, *service*, dan *process*, yang akan dijabarkan sebagai berikut :



Gambar 1. Relasi Komponen SOA

1. *Message*, yaitu data yang dibutuhkan untuk menyelesaikan sebagian atau sebuah unit kerja, yang dipertukarkan antara satu service dengan yang lainnya.
2. *Operation*, yaitu fungsi-fungsi yang dimiliki oleh sebuah *service* untuk memproses message hingga menghasilkan sesuatu. Fungsi-fungsi inilah yang nantinya akan saling berinteraksi untuk menyelesaikan sebuah unit kerja.
3. *Service*, merepresentasikan sekumpulan operation yang berhubungan untuk menyelesaikan sekumpulan unit kerja yang berhubungan.
4. *Process*, merupakan business rule yang menentukan operasi mana yang digunakan untuk mencapai tujuan tertentu.

3.2. Metode Pengembangan sistem

Tahapan pengembangan sistem dalam perancangan Sistem Informasi Kartu Rencana Studi dengan menggunakan metodologi *Object Oriented Analysis and Design (OOAD)*², yaitu:

1. Tahap *Requirements* (Persyaratan)

Tahap *Requirements* merupakan dasar dari semua yang harus dilakukan sistem. Pada prinsipnya persyaratan hanyalah sebuah pernyataan dari apa yang harus dilakukan atau tidak harus dilakukan oleh sebuah sistem. Tugas utama dalam tahap ini yaitu :

- a. Menentukan *Actor* Dan Kasus Pengguna
- b. Menentukan Persyaratan Fungsional
- c. Menentukan Persyaratan Non Fungsional
- d. Memprioritaskan Persyaratan

2. Tahap *Analysis* (Analisis)

Tahap ini berfokus pada apa yang sistem harus lakukan tanpa mengetahui bagaimana sistem melakukannya pada bagian desain. Tahapan ini juga menentukan spesifikasi dari sistem yang diinginkan. Yaitu sebagai berikut:

- a. Menganalisa Permasalahan
- b. Menganalisa Aktivitas
- c. Menganalisis Spesifikasi Kebutuhan

3. Tahap *Design* (Desain)

Berdasarkan analisis sistem yang ada, maka diusulkan suatu program yang dapat mempermudah dalam melakukan pengertian Kartu Rencana Studi Mahasiswa di universitas Islam OKI kayuagung. Pada proses perancangan sistem yang akan dibangun dengan menggunakan beberapa bentuk rancangan dari *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram*, desain *database* dan rancangan struktur menu

a. *Usecase Diagram*

Diagram *use case* digunakan untuk memperlihatkan hubungan-hubungan yang terjadi antara *actor* dengan *use case* yang ada dalam sistem Kartu Rencana Studi, sehingga calon pengguna sistem/perangkat lunak dapat mendapatkan pemahaman tentang sistem yang akan dikembangkan. Untuk lebih jelas mengenai *use case diagram* sebagai berikut:

- b. *Activity Diagram*
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut ini akan ditampilkan *activity diagram* mahasiswa
- c. *Clas Diagram*
Arsitektur sistem di sini dapat digambarkan dengan class diagram, di mana class adalah sebuah spesifikasi yang jika di instasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan keadaan (atribut/property) sesuatu sistem, sekaligus menawarkan layanan untuk manipulasi keadaan tersebut (metode/fungsi).
- d. *Sequence Diagram*
Sequence diagram menggambarkan kelakuan objek pada use case sistem informasi Kartu Rencana Studi ini dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek, digunakan untuk menggambarkan arus pekerjaan, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil
- e. *Desain Database*
Basis Data merupakan sistem yang terdiri atas kumpulan *file (table)* yang saling berhubungan dalam sebuah basis data di sebuah sistem komputer yang memungkinkan untuk mengakses dan memanipulasi *file file (table-table)* tersebut

4. Tahap Implementasi

Pada tahap ini merupakan tahap *programming*, dimana pembuatan desain diterjemahkan kedalam bahasa pemrograman, basis data, dan implementasi perangkat keras. Kelas kelas yang dibentuk pada tahap desain dikonversi menjadi *code* sesungguhnya dalam bahasa pemrograman *objek-oriented* melalui proses *generate*. Hasil *generate* berupa skeleton dari program. Selanjutnya menjadi tugas programmer untuk menyelesaikan program hasil *generate*.

5. *Testing*

Testing terhadap sistem *software* biasanya berupa tes unit, tes integrasi, tes sistem dan tes *acceptance*. Tes unit adalah tes terhadap kelas individual atau terhadap sekelompok kelas, biasanya dilakukan oleh *programmer*. Tes integrasi mengintegrasikan komponen dan kelas-kelas dalam rangka verifikasi. Tes sistem memandang sistem sebagai kotak hitam (*black box*) dalam rangka validasi bahwa sistem berfungsi sesuai dengan harapan *end user*. Tes *acceptance* dilakukan oleh customer untuk verifikasi bahwa sistem sesuai dengan kebutuhan, sama seperti tes sistem.

4 HASIL DAN PEMBAHASAN

Skenario yang digunakan untuk mempersentasikan sebuah *interaksi* antara *actor* dengan sistem, Perilaku sistem adalah bagaimana sistem beraksi dan bereaksi. Perilaku ini merupakan aktifitas sistem yang bisa dilihat dari luar dan bisa diuji. Skenario pengguna yang menggambarkan alur penggunaan sistem, Setiap skenario digambarkan dari sudut pandang "aktor", seseorang atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara. Skenario didalam sistem ini dapat dilihat pada gambar berikut ini :

a. Skenario Admin KRS

Tabel 1. Skenario Admin KRS

IDENTIFIKASI	
Nomor	1
Aktor	Admin KRS
Tujuan	Memberi hak akses login mahasiswa dan mengolah data krs
Deskripsi	Melakukan pengecekan data mahasiswa serta mengkonfirmasi pembayaran mahasiswa yang sudah lunas untuk memberi hak akses login Mahasiswa

b. Skenario Login Admin KRS

Tabel 2. Skenario Login Admin KRS

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukan id dan password	
	2. Mengecek valid tidaknya data masukan
	3. Masuk ke aplikasi pengolahan Admin KRS
Skenario Alternatif	
1. Memasukan id dan password	
	2. Mengecek valid tidaknya data masukan
	3. Menampilkan pesan login tidak valid
4. Memasukan id dan password yang valid	
	5. Mengecek valid tidaknya data masukan
	6. Masuk ke aplikasi pengolahan Admin KRS

c. Skenario Mahasiswa

Tabel 3. Skenario Mahasiswa

IDENTIFIKASI	
Nomor	4
Aktor	Mahasiswa
Tujuan	Melakukan pengentrian KRS
Deskripsi	Mahasiswa telah membayar dan mendapat hak akses login dilanjutkan memilih mata kuliah, kelas, ruang dan waktu perkuliahan

d. Skenario Login Mahasiswa

Tabel 4. Skenario Login Mahasiswa

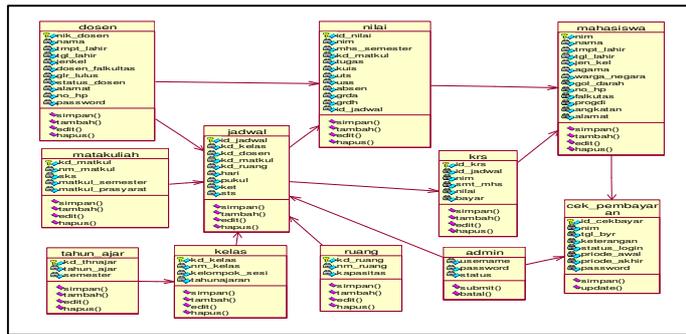
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mahasiswa telah membayar uang semesteran dan telah diaktivasi login oleh admin krs	
2. Memasukan nim dan password	
	3. Mengecek valid tidaknya data masukan
	4. Masuk ke sistem entry krs
Skenario Alternatif	
5. Memasukan nim dan password	
	6. Mengecek valid tidaknya data masukan
	7. Menampilkan pesan login tidak valid
8. Memasukan nim dan password yang valid	
	9. Mengecek valid tidaknya data masukan
	10. Masuk ke sistem entry krs

e. Use Case Diagram

Diagram use case digunakan untuk memperlihatkan hubungan-hubungan yang terjadi antara actor dengan use case yang ada dalam sistem Kartu Rencana Studi, sehingga calon pengguna sistem/perangkat lunak dapat mendapatkan pemahaman tentang sistem yang akan dikembangkan. Untuk lebih jelas mengenai use case diagram sebagai berikut:

g. *Class Diagram*

Arsitektur sistem di sini dapat digambarkan dengan *class diagram*, di mana *class* adalah sebuah spesifikasi yang jika di instansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan keadaan (atribut/property) sesuatu sistem, sekaligus menawarkan layanan untuk manipulasi keadaan tersebut (metode/fungsi). Kelas- kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi yang sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya ada pada stuktur kelas tetap. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai dengan kebutuhan. *Class diagram* ini berisikan objek-objek yang terdapat didalam sistem Kartu Rencana Studi Universitas Islam OKI yang akan di gambarkan sebagai berikut:



Gambar 4. *Class Diagram*

h. *Halaman Service KRS*

Pada halaman ini merupakan halaman yang menyediakan fungsi pemanggilan yang bermanfaat untuk mengirimkan data jadwal, data mahasiswa, data semester mahasiswa yang akan dipakai oleh sistem entry krs mahasiswa, begitupun juga halaman *service* ini menyediakan fungsi simpan untuk menyimpan kedalam *database*. *service* yang disediakan disini antara lain *getjadwal*, untuk mengirim kan data jadwal, *getmahasiswa* untuk mengirimkan data mahasiswa, *getsmt* untuk mengirimkan data semester, *simpanKrs* dan *simpanNilai* berfungsi untuk menyimpan *service* yang dipakai Tampilan hasil *Services*. Tampilan ini merupakan tampilan utama *services* yang disediakan untuk membantu proses pengentrian kartu rencana studi di universitas islam OKI Kayuagung yang terdiri dari *getjadwal*, *getpilihjadwal*, *getmahasiswa*, *getsmt*, *simpanKrs* dan *simpanNilai* seperti pada gambar berikut :



Gambar 5. Tampilan Hasil *Services*

i. *Tampilan getjadwal*

Tampilan ini merupakan tampilan *getjadwal* yang akan digunakan untuk mengirim jadwal dari bagian BAAK ke menu entri KRS



Gambar 6. Tampilan *getjadwal*

j. Tampilan getmahasiswa

Tampilan ini merupakan tampilan untuk mendapatkan *services* dari data mahasiswa

```
-<xsd:complexType name="mahasiswa">
- <xsd:all>
  <xsd:element name="nim" type="xsd:string"/>
  <xsd:element name="nm_mhs" type="xsd:string"/>
  <xsd:element name="progdi" type="xsd:string"/>
  <xsd:element name="semester" type="xsd:string"/>
</xsd:all>
</xsd:complexType>
-<xsd:complexType name="outmahasiswa">
- <xsd:complexContent>
  - <xsd:restriction base="SOAP-ENC:Array">
    <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:mahasiswa[]"/>
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
```

Gambar 7. Tampilan getmahasiswa

k. Tampilan simpanKrs dan simpanNilai

Tampilan ini merupakan tampilan disediakan agar menu entry KS dapat menyimpan data perkuliahan yang telah dipilih

```
<message name="simpanKRSRequest"/>
<message name="simpanKRSResponse"/>
<message name="simpanNilaiRequest"/>
<message name="simpanNilaiResponse"/>
```

Gambar 8. Tampilan getsmt

l. Tampilan Login Admin KRS dan Login BAAK

Tampilan ini merupakan tampilan menu login Admin KRS dan Juga BAAK jadi sebelum melakukan aktivitas pengolahan sistem admin KRS dan BAAK harus melakukan login terlebih dahulu dihalaman ini.



Gambar 9. Tampilan Login Admin KRS dan BAAK

5. KESIMPULAN

Setelah melihat perancangan sistem dan uraian dari implementasi dapat diambil beberapa kesimpulan sebagai berikut :

1. Dalam penelitian ini dihasilkan sebuah sistem informasi kartu rencana studi yang berbasis jaringan yang diharapkan dapat mempercepat dalam proses pengentrian KRS agar pendawalan perkuliahan yang selama ini sering mengalami keterlambatan sehingga dapat tepat waktu.
2. Fungsionalitas sistem Kartu Rencana Studi ini dengan menggunakan metode SOA dapat menyajikan *service* yang dapat disajikan di menu BAAK dalam mengatur penjadwalan serta dapat konsumsi oleh aplikasi entry KRS untuk mendapatkan jadwal tersebut

DAFTAR PUSTAKA

- Erl. 2005. Di kutip dalam buku, Analisis dan Desain Berorientasi Services Untuk Aplikasi Manajemen Proyek. Yogyakarta: Penerbit Andi.
- Lucky. 2008. *XML Web Service, Aplikasi Desktop, Internet & Handphone*, Penerbit Jasakom, Jakarta Barat.
- Sarno, Riyanto. 2012. *Analisis dan Desain Berorientasi Services Untuk Aplikasi Manajemen Proyek*, Penerbit Andi, Yogyakarta..