

PERANCANGAN MODEL PENGEMBANGAN PERANGKAT LUNAK BERBASIS SERVICE ORIENTED ARCHITECTURE MENGGUNAKAN METODOLOGI UNIFIED PROCESS

THE MODEL DEVELOPMENT DESIGN OF SOFTWARE BASED SERVICE ORIENTED ARCHITECTURE USING UNIFIED PROCESS METHODOLOGY

¹Ade Hodijah

¹Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Jl. Geger Kalong Hilir, Desa Ciwaruga, Parongpong, Kabupaten Bandung Barat, Jawa Barat 40012

e-mail: ¹adehodijah@jtk.polban.ac.id

Abstract. *This study examines the development model of software service-oriented (Service Oriented Architecture / SOA) using object oriented methodology (the Unified Process / UP). The details of the model in this study consisted of: Process, Notation, and Tools. Where the process uses the Rational Unified Process (RUP); Notation uses Business Process Modeling (BPM); Tool uses Business Process Management System (BPMS). To validate the results of the design, used the validation on basic theory supporting research and expert judgment in order to proof hipotesis. The research result shows that activities in the development of software service-oriented can use the object-oriented methodology by adding activities, namely the analysis of service and design service of the concept of BPM. Through the analysis of current business processes (as-is) will identify the desgin of service candidates according to service model (to-be)that will be developed. These additional activities conducted on the process design work and will continue to be refined according iteration of each phase of the RUP.*

Keywords: *SOA, RUP, BPM and BPMS*

Abstrak. *Penelitian ini mengkaji model pengembangan perangkat lunak berorientasi service (Service Oriented Architecture / SOA) menggunakan metodologi berorientasi objek (Unified Process / UP). Rincian dari model pada penelitian ini terdiri dari: Proses, Notasi, dan Alat Bantu. Dimana Proses menggunakan Rational Unified Process (RUP); Notasi menggunakan Business Process Modeling (BPM); Alat Bantu menggunakan Business Process Management System (BPMS). Untuk melakukan validasi terhadap hasil rancangan, digunakan validasi pada teori dasar mendukung penelitian dan expert judgement guna pembuktian hipotesis. Hasil penelitian menunjukkan bahwa aktifitas-aktifitas dalam pengembangan perangkat lunak berorientasi service dapat menggunakan metodologi berorientasi objek dengan melakukan penambahan aktifitas, yakni analisis service dan perancangan service dari konsep BPM. Melalui analisis dari proses bisnis saat ini (as-is) maka akan teridentifikasi rancangan kandidat-kandidat service sesuai model service (to-be) yang akan dikembangkan. Aktifitas tambahan ini dilakukan pada kerja proses Design dan akan terus disempurnakan sesuai iterasi dari setiap fase RUP.*

Kata kunci: *SOA, RUP, BPM, dan BPMS*

1. Pendahuluan

SOA adalah sebuah bentuk teknologi arsitektur yang mengikuti prinsip-prinsip berorientasi *service*, yakni sebuah pendekatan dalam penyelesaian masalah besar dengan membaginya menjadi sekumpulan layanan (*service*) kecil yang menyelesaikan permasalahan spesifik [1]. BPM adalah sekumpulan teknologi dan standard untuk perancangan, otomasi, eksekusi, pengelolaan, dan monitoring proses bisnis [2]. Dimana

proses bisnis adalah aliran aktifitas dimana setiap aktifitas tersebut menyatakan pekerjaan dari pelaksana bisnis, sebuah sistem internal atau proses dari perusahaan partner yang diarahkan kepada pencapaian tujuan bisnis [2]. BPM memberikan nilai tambah kemudahan dan kecepatan untuk melakukan adaptasi terhadap perubahan proses bisnis, sedangkan SOA sebagai pendekatan arsitektural [3].

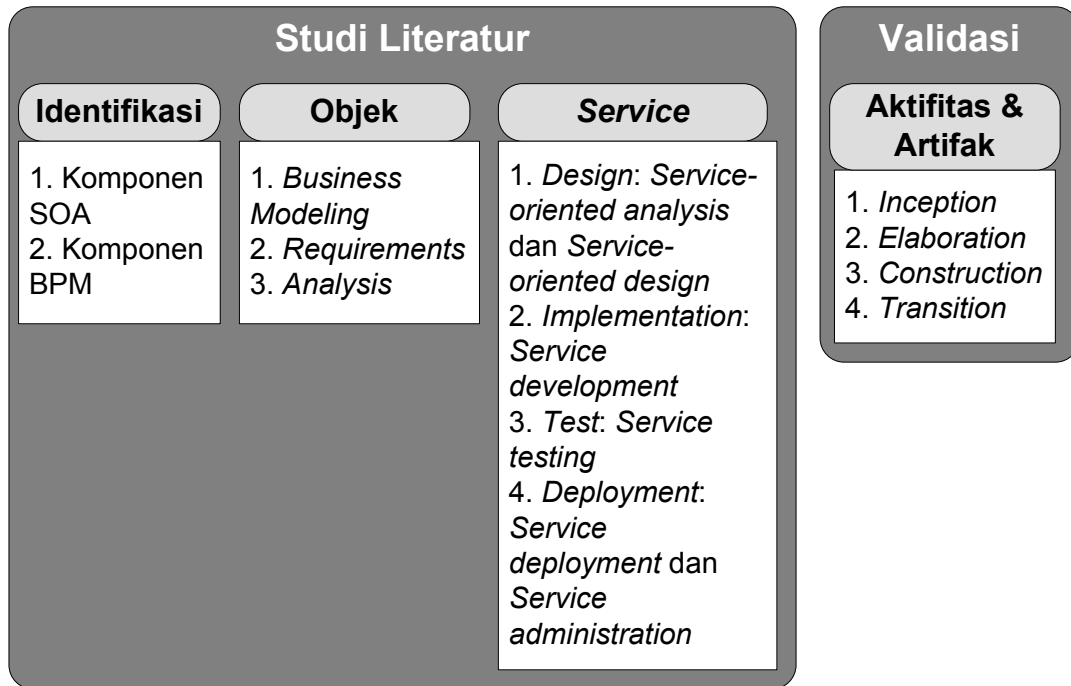
Dalam pengembangan aplikasi SOA diperlukan sebuah metodologi, seperti *Service Oriented Unified Process* (SOUP) mencakup siklus rekayasa perangkat lunak yang lengkap dari sisi pengembangan (*SOUP for initial SOA deployment*) sampai sisi pemeliharaan (*SOUP for ongoing SOA management*) [4]. Bagaimana fase pengembangan perangkat lunak berbasis SOA dari SOUP menjadi ide dari penelitian ini, yakni didasarkan pada RUP. Namun SOUP tidak memiliki dokumentasi rinci dan menyisakan ruang untuk adaptasi. Sehingga diperlukan pendekatan praktis, dalam penelitian ini menggunakan alat bantu BPMS untuk menerapkan bagaimana aplikasi berorientasi *service* dikembangkan.

Berdasarkan pendahuluan di atas, maka masalah-masalah penelitian yang dikaji adalah sebagai berikut: (1) Aktifitas apakah yang diperlukan dalam melakukan pengembangan perangkat lunak berorientasi *service* dengan menggunakan metodologi pengembangan perangkat lunak berorientasi objek? (2) Apa sajakah artifak yang dihasilkan dari dalam melakukan pengembangan perangkat lunak berorientasi *service* dengan menggunakan metodologi pengembangan perangkat lunak berorientasi objek?

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut: (1) Menganalisis aktifitas yang diperlukan dalam melakukan pengembangan perangkat lunak berorientasi *service* dengan menggunakan metodologi pengembangan perangkat lunak berorientasi objek; (2) Menganalisis artifak yang diperlukan dalam melakukan pengembangan perangkat lunak berorientasi *service* dengan menggunakan metodologi pengembangan perangkat lunak berorientasi objek. Dengan urgensi penelitian sebagai berikut: (1) Penelitian ini dianggap penting karena hasilnya dapat memberikan pendekatan praktis khususnya aktifitas dan artifak dari pengembangan perangkat lunak berorientasi *service* menggunakan metodologi pengembangan perangkat lunak berorientasi objek; (2) Bagi peneliti, dari hasil analisis terhadap langkah-langkah bagaimana mengembangkan aplikasi berbasis SOA dengan alat bantu BPMS, sehingga dapat menyusun dan mengembangkan bahan ajar dan buku ajar sebagai acuan bagi pengajar.

Faktor-faktor yang dianalisis dalam penelitian ini adalah rekayasa perangkat lunak meliputi tiga elemen kunci dalam pembuatannya, yaitu: Proses, Notasi, dan Alat Bantu [5]. Pada Proses, fokus penelitian ini meliputi bagaimana Pemodelan Bisnis, Spesifikasi Kebutuhan, Analisis & Perancangan, Implementasi, Pengujian. Sementara Notasi menggunakan standard untuk memodelkan *web service* dan proses *web service* yang kemudian dapat diimplementasikan dengan Alat Bantu BPMS.

Berdasarkan aktifitas dari RUP [6], BPM [7], dan siklus hidup SOA [1] berikut langkah-langkah yang akan diadaptasi sebagai metode penelitian dalam pengembangan perangkat lunak berbasis SOA pada RUP.



Gambar 1. Gabungan komponen BPM dan SOA beserta fase RUP

2. Proses

Proses pada penelitian ini menggunakan proses RUP sebagai acuan pelaksanaan dan urutan aktifitas-aktifitas dalam rekayasa perangkat lunak beserta artifak dari setiap fasenya. RUP adalah salah satu metodologi pengembangan perangkat lunak, terutama perangkat lunak berorientasi objek dengan aktifitas yang pada pengembangan model dengan menggunakan *Unified Modeling Language* (UML) [8]. Dimana pengembangan perangkat lunak sendiri adalah sekumpulan aktifitas yang dibutuhkan untuk mentransformasi kebutuhan pengguna menjadi sebuah sistem perangkat lunak.

Pada penelitian ini, terdapat lima alur kerja proses dan tiga fase RUP yang akan dirincikan aktifitas beserta artifak yang diperlukan dalam pengembangan aplikasi berbasis *service* menggunakan metodologi berorientasi objek. Lima alur kerja proses tersebut sebagai berikut: (1) Pemodelan Bisnis; (2) Spesifikasi Kebutuhan; (3) Analisis & Perancangan; (4) Implementasi; (5) Pengujian dan tiga fase RUP tersebut sebagai berikut: (1) *Inception*; (2) *Elaboration*; (3) *Construction*. Pada ketiga fase yang terdapat pada RUP dilakukan beberapa iterasi sesuai kebutuhan. Berikut keterhubungan fase, aktivitas, artifak dari RUP.

Tabel 1. Keterhubungan fase, aktifitas, dan artifak RUP [6]

Fase	Aktifitas	Artifak
<i>Inception</i>	Analisis <i>domain</i> masalah	- Hasil analisis <i>domain model</i> , yakni diagram <i>class</i> tanpa <i>method</i> .
	Analisis perangkat lunak	- Diagram <i>use case</i>
<i>Elaboration</i>	Analisis <i>domain</i> masalah	- Revisi diagram <i>class</i> tanpa <i>method</i>
	Analisis perangkat lunak	- Revisi diagram <i>use case</i> - Skenario <i>use case</i> - Diagram <i>sequence</i> - Identifikasi paket dan kelas

		analisis
	Perancangan perangkat lunak	- Perancangan antarmuka - Perancangan subsistem - Diagram <i>deployment</i> - <i>Throw-away prototype</i> - Identifikasi kelas perancangan
	Implementasi perangkat lunak	- Kode program
<i>Construction</i>	Perancangan perangkat lunak	- Revisi identifikasi kelas perancangan - Revisi perancangan antarmuka - Revisi perancangan subsistem - Revisi diagram <i>deployment</i>
	Impelementasi perangkat lunak	- Revisi kode program
	Pengujian Terintegrasi	- Skenario pengujian - Evaluasi pengujian

Menurut Kunal Mittal (2007), terdapat enam tahapan pengembangan, yaitu: (1) *Incept*; (2) *Define*; (3) *Design*; (4) *Construct*; (5) *Deploy*; (6) *Support*. Dimana arsitektur SOA mulai digunakan pada tahap *Design* dari kategori *SOUP for Initial SOA deployment*. Dimana siklus hidup SOA terdiri dari enam tahapan, yaitu: (1) *service-oriented analysis*; (2) *service-oriented design*; (3) *service development*; (4) *service testing*; (5) *service deployment*; (6) *service administration*, dengan rincian kegiatan sebagai berikut [1].

Tabel 2. Siklus hidup SOA [1]

Siklus SOA	Kegiatan
<i>Service-oriented analysis</i>	Menentukan lingkup (<i>domain</i>) aplikasi SOA yang akan dibangun, yakni <i>requirement</i> otomasi bisnis, sistem <i>legacy</i> , memodelkan kandidat <i>service</i> [1].
<i>Service-oriented design</i>	Memodelkan <i>service</i> , yakni menentukan teknologi implementasi SOA, merancang <i>service</i> (<i>entity-centric</i> atau <i>task-centric</i>), menentukan <i>service</i> untuk setiap <i>layer</i> SOA (<i>application, business, orchestration</i>) [1].
- <i>Service development</i> - <i>Service testing</i> - <i>Service deployment</i> - <i>Service administration</i>	Implementasi <i>service</i> untuk setiap <i>layer</i> SOA, menguji <i>service</i> , hingga instalasi dan konfigurasi komponen terdistribusi [1].

Berdasarkan siklus hidup SOA diatas, berikut tambahan aktifitas dan artefak untuk tabel 1 tersebut.

Tabel 3. Keterhubungan fase RUP dan siklus hidup SOA

Fase UP	Siklus SOA	Aktivitas	Artefak
<i>Inception</i>	<i>Service-oriented analysis</i>	Analisis <i>domain</i> masalah	- Hasil analisis <i>domain model</i> , yakni diagram <i>class</i> tanpa <i>method</i> .
		Analisis sistem <i>as-is</i>	- Identifikasi sistem <i>as-is</i> dan <i>glossary</i> terkait - Identifikasi proses bisnis sistem <i>as-is</i> - Spesifikasi bisnis <i>use case</i> (lingkup masalah), <i>hardware, software</i>
<i>Elaboration</i>	<i>Service-oriented design</i>	Analisis <i>domain</i> masalah	- Revisi diagram <i>class</i> tanpa <i>method</i> . - Revisi <i>glossary</i> terkait - Revisi spesifikasi bisnis <i>use case</i> (lingkup masalah) - Identifikasi proses bisnis <i>to-be</i>

		Analisis model <i>service to-be</i>	- Identifikasi kandidat <i>service (task, entity, utility)</i> - Identifikasi kandidat otomatisasi operasi <i>service</i>
		Analisis perangkat lunak <i>client</i>	- Diagram <i>use case client</i> - Skenario/deskripsi <i>use case</i>
		Perancangan <i>service</i>	- Identifikasi <i>service</i> dan teknologinya - Identifikasi operasi <i>service</i> - Diagram <i>class</i> - Diagram <i>deployment</i>
		Perancangan perangkat lunak <i>client</i>	- Realisasi <i>use case</i> (diagram <i>sequence</i>) - Perancangan antar muka - Diagram <i>class</i> - Diagram <i>deployment</i>
		Perancangan arsitektur <i>service</i>	- Perancangan infrastruktur aplikasi berbasis <i>desktop, web, mobile</i>
<i>Service development</i>	Implementasi	- Revisi Spesifikasi <i>hardware/software</i> - Kode program (model, <i>controller</i> , UI)	
<i>Service testing</i>	Pengujian unit	- Skenario pengujian (fungsional, nonfungsional): <i>unit, integration, system, stress</i>	
Construction	<i>Service development</i>	Perancangan <i>service</i>	- Revisi identifikasi <i>service</i> dan teknologinya - Revisi identifikasi operasi <i>service</i> - Revisi diagram <i>class</i> - Revisi diagram <i>deployment</i>
		Perancangan perangkat lunak <i>client</i>	- Revisirealisasi <i>use case</i> (diagram <i>sequence</i>) - Revisi perancangan antar muka - Revisi diagram <i>class</i> - Revisidiagram <i>deployment</i>
		Implementasi	- Revisi kode program (model, <i>controller</i> , UI)
	<i>Service testing</i>	Pengujian terintegrasi	- Revisi skenario pengujian (fungsional, nonfungsional): <i>unit, integration, system, stress</i>
	<i>Service testing</i>	Pengujian terintegrasi	- Evaluasi pengujian (fungsional, nonfungsional)

3. Notasi

Notasi pada penelitian ini menggunakan notasi BPM untuk merepresentasikan artifak-artifak perangkat lunak. BPM adalah disiplin ilmu untuk memodelkan, otomatisasi, mengelola, dan mengoptimasi proses bisnis untuk meningkatkan *profitability* [9]. Tujuan dari BPM seperti yang dituliskan pada [10] sebagai berikut.

1. Mengurangi ketidakcocokan antara kebutuhan perusahaan dengan sistem teknologi informasi yang terimplementasi dengan memungkinkan pihak analis bisnis memodelkan proses bisnis dan membiarkan bagian teknologi informasi menyediakan infrastruktur untuk mengeksekusi dan mengontrol proses bisnis tersebut.
2. Meningkatkan produktifitas pegawai dan mengurangi biaya operasional dengan mengotomatisasi dan melancarkan proses bisnis.
3. Meningkatkan *agility* dan fleksibilitas pada perusahaan dengan secara eksplisit memisahkan antara logika proses dari aturan bisnis dan merepresentasikan proses bisnis dalam bentuk yang mudah diubah jika terjadi perubahan kebutuhan.
4. Mengurangi biaya pengembangan dan usaha dengan menggunakan bahasa pemrograman *highlevel* yang memungkinkan analis bisnis dan pengembang

secara cepat membangun dan memperbaharui sistem teknologi informasi dalam *domain* permasalahan tertentu.

SOA merepresentasikan sebuah pendekatan komputasi terdistribusi yang menganggap *resource-resource* dari aplikasi sebagai *service* yang dapat ditemukan pada jaringan. Oleh karena itu, *resource* yang ada dapat digunakan oleh banyak proses bisnis (*reusable*) atau dalam skala yang lebih besar dan membedakan antara fungsi dan proses untuk memungkinkan pendefinisian ulang proses bisnis dengan tetap menggunakan fungsi-fungsi yang ada (*loosely coupled*) [11]. *Reusable* dalam konteks ini mengandung makna secara teknis maupun secara proses bisnis. Secara teknis, suatu layanan dapat memanfaatkan komponen yang telah ada sebelumnya, atau bahkan menggabungkan beberapa layanan yang telah ada sebelumnya ke dalam suatu antarmuka tersendiri. Secara proses bisnis, hendaknya suatu layanan dapat dimanfaatkan oleh lebih dari satu jenis *client* [1].

Berdasarkan tujuan dari BPM diatas, berikut tambahan aktifitas dan artifak untuk tabel 3 tersebut.

Tabel 4. Keterhubungan fase RUP dan siklus hidup SOA dan BPM

Fase UP	Siklus SOA	Aktivitas	Artifak
<i>Inception</i>	<i>Service-oriented analysis</i>	Analisis <i>domain</i> masalah	- Hasil analisis <i>domain model</i> , yakni diagram <i>class</i> tanpa <i>method</i> .
		Analisis sistem <i>as-is</i>	- Identifikasi sistem <i>as-is</i> dan <i>glossary</i> terkait - Identifikasi proses bisnis sistem <i>as-is</i> - Spesifikasi bisnis <i>use case</i> (lingkup masalah), <i>hardware</i> , <i>software</i>
<i>Elaboration</i>	<i>Service-oriented design</i>	Analisis <i>domain</i> masalah	- Revisi <i>class</i> tanpa <i>method</i> . - Revisi <i>glossary</i> terkait - Revisi spesifikasi bisnis <i>use case</i> (lingkup masalah) - Identifikasi proses bisnis <i>to-be</i>
		Analisis model <i>service to-be</i>	- Identifikasi kandidat <i>service</i> (<i>task</i> , <i>entity</i> , <i>utility</i>) - Identifikasi kandidat otomatisasi operasi <i>service</i>
		Analisis perangkat lunak <i>client</i>	- Diagram <i>use case client</i> - Skenario/deskripsi <i>use case</i>
		Perancangan <i>service</i>	- Identifikasi <i>service</i> dan teknologinya - Identifikasi operasi <i>service</i> - Diagram <i>class</i> - Diagram <i>deployment</i> - Rancangan proses bisnis dalam <i>Business Process Modeling Notation(BPMN)</i>
		Perancangan perangkat lunak <i>client</i>	- Realisasi <i>use case</i> (diagram <i>sequence</i>) - Perancangan antar muka - Diagram <i>class</i> - Diagram <i>deployment</i>
	Perancangan arsitektur <i>service</i>	- Perancangan infrastruktur aplikasi berbasis <i>desktop</i> , <i>web</i> , <i>mobile</i>	
	<i>Service development</i>	Implementasi	- Revisi Spesifikasi <i>hardware/software</i> - Kode program (model, <i>controller</i> , UI)
	<i>Service</i>	Pengujian	- Skenario pengujian (fungsional, nonfungsional):

	<i>testing</i>	unit	<i>unit, integration, system, stress</i>
	<i>Service-oriented deployment</i>	Implementasi	- Implementasi proses bisnis menggunakan BP <i>Execution Language</i> (BPEL)
Construction	<i>Service-oriented design</i>	Perancangan <i>service</i>	- Revisi rancangan proses bisnis dalam BPMN
	<i>Service development</i>	Perancangan <i>service</i>	- Revisi identifikasi <i>service</i> dan teknologinya - Revisi identifikasi operasi <i>service</i> - Revisi diagram <i>class</i> - Revisi diagram <i>deployment</i>
		Perancangan perangkat lunak <i>client</i>	- Revisirealisasi <i>use case</i> (diagram <i>sequence</i>) - Revisi perancangan antar muka - Revisi diagram <i>class</i> - Revisidiagram <i>deployment</i>
		Implementasi	- Revisi kode program (model, <i>controller</i> , UI)
Construction	<i>Service testing</i>	Pengujian terintegrasi	- Revisi skenario pengujian (fungsional, nonfungsional): <i>unit, integration, system, stress</i>
	<i>Service testing</i>	Pengujian terintegrasi	- Evaluasi pengujian (fungsional, nonfungsional)
	<i>Service-oriented deployment</i>	Implementasi	- Revisi implementasi proses bisnis menggunakan BPEL

4. Alat Bantu

Untuk menunjang pengerjaan BPM pada perangkat lunak, terdapat sebuah teknologi yang disebut BPMS. Menurut [10], umumnya komponen yang ada di dalam BPMS adalah sebagai berikut.

1. *Process Modeling*, kaskas ini berfungsi untuk memodelkan proses bisnis dalam bentuk visual. Kaskas ini memungkinkan pihak yang tidak mengerti dalam hal implementasi, tapi terlibat penting dengan proses tersebut dapat memodelkan proses bisnis untuk dijalankan pada sistem.
2. *Process Execution*, kaskas ini menyediakan mesin untuk mengimpor model proses yang telah dimodelkan pada kaskas *process modeling* untuk dapat dijalankan dalam sistem. Beberapa contoh tanggung jawab dari kaskas ini seperti:
 - Menghubungkan beberapa *task* dan mengeksekusinya di dalam keterurutan
 - Memberikan dan mengalami *task* kepada pengguna yang berhak
 - Mengakses sistem IT lokal atau *remote* untuk mengambil informasi yang dibutuhkan proses, memperbaharui informasi yang dibuat oleh proses, dan mengeksekusi transaksi yang didefinisikan oleh proses
3. *Process Monitoring*, kaskas ini menyediakan pengawasan proses bisnis yang berjalan. Beberapa kemampuan pada kaskas ini seperti:
 - Melihat laporan dari seluruh eksekusi proses
 - Melihat laporan dari seluruh proses yang komplit
 - Melihat status dari proses, apakah telah selesai atau masih berjalan
 - Menghentikan sementara proses dan menyambungannya lagi
 - Mengatur prioritas dari proses

4. *Business Activity Monitoring* (BAM). Komponen ini berfungsi untuk menganalisis kejadian dan informasi yang tersimpan mengenai proses bisnis untuk menyediakan *feedback* secara *real-time* untuk perbaikan proses bisnis. Saat ini terdapat banyak sekali BPMS, diantaranya yaitu: Intalio BPMS, jBoss jBPM, OpenESB.

5. Validasi Hipotesis

Validasi expert judgement dilakukan dengan melakukan diskusi dengan ahli. Pada penelitian ini dilakukan dengan Bambang Wisnuadhi, S.Si.,M.T. Kegiatan dilakukan pada hari Jumat, 16 September 2016 bertempat di Laboratorium RPL, Jurusan Teknik Komputer dan Informatika, POLBAN. Berikut adalah masukan yang diberikan terhadap penelitian yaitu: (1) SOA adalah suatu cara mengorganisir perangkat lunak, sehingga organisasi dapat dengan cepat merespon perubahan kebutuhan; (2) SOA dimulai pada alur kerja proses *Design* yang mendukung implementasi aplikasi berbasis *service*; (3) Aktifitas dan artifak dari SOA dijelaskan pada setiap iterasi dari setiap fase RUP; (4) Untuk memperjelas tahapan pengembangan, dilengkapi dengan contoh studi kasus. Berdasarkan pembahasan di atas, maka validasi hipotesis “penerapan BPM pada RUP dapat digunakan sebagai alternatif metode rekayasa perangkat lunak berorientasi *service*”.

6. Kesimpulan dan Saran

Berdasarkan rumusan masalah, hasil penelitian, dan pembahasan yang disajikan sebelumnya dapat ditarik kesimpulan sebagai berikut: (1) Terdapat 2 aktifitas beserta artifaknya sebagai tambahan dalam mengembangkan perangkat lunak berbasis SOA dengan menggunakan metodologi pengembangan perangkat lunak berorientasi objek; (2) Penambahan aktifitas dan artifak ini sesuai konsep BPM, dimana pada model tersebut tercakup alat bantu BPMS yang berfungsi untuk memfasilitasi perancangan, eksekusi, dan pengawasan proses bisnis.

Berdasarkan kesimpulan di atas, peneliti mengusulkan rekomendasi bagi pengembang pendidikan, peneliti lebih lanjut, dan pihak terkait yaitu: (1) Meneliti aktifitas beserta artifak yang harus dilakukan pada tahap *Configuration & Change Management*, *Project Management*, *Environment* untuk setiap fase UP dari pengembangan aplikasi berbasis SOA.

7. Ucapan Terima Kasih

Ucapan terima kasih disampaikan kepada seluruh dosen koordinator mata kuliah Proyek 3 D4 2016 (penulis) dan kepada dosen mata kuliah lain pendukung Proyek 3 yang telah bersedia meluangkan waktu untuk memberikan data dan informasi terkait penelitian berlangsung. Selanjutnya kepada segenap staf UPPM Politeknik Negeri Bandung yang telah memberikan kesempatan untuk melaksanakan penelitian Pemula tahun 2016.

Daftar pustaka

- Erl, Thomas. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR. [Online]. <http://hexawebhosting.com/getfile-326.zip>. [Dikunjungi Mei 2016].
- Mike Havey. (2005). *Essential Business Process Modeling*, O'Reilly.
- Faouzi Kamoun. (2007). *A Roadmap toward the Convergence of Business Process Management and Service Oriented Architecture*, ACM Ubiquity.
- Kunal Mittal. *Service Oriented Unified Process (SOUP)*. [Online]. <http://www.kunalmittal.com/html/soup.html>. [Dikunjungi Agustus 2016].
- Pressman, R.S. (1997). *Software Engineering: A Practitioners's Approach*, McGraw-Hill.
- Desenta, Salma. (2006). *Pembangunan Kakas Konfigurasi dan Pengawasan Proyek Perangkat Lunak Bermetodologi Unified Process*. *Laporan Tugas Akhir*. STEI-ITB.
- Panggabean, G.F. (2010). *Studi Framework Pembangunan Aplikasi Berbasis SOA dan BPM*. Tesis. STEI-ITB.
- Larman, Craig. (2004). *Applying UML And Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Edisi 3.
- Khan, Rashid. *Evaluating BPM Software*. [Online]. <http://bpm.com/FeatureRO.asp?FeatureId=148>. [Dikunjungi Juli 2016].
- Newcomer, Eric dan Greg Lomow. (2004). *Understanding SOA With Web Services*. Addison Wesley Professional.
- MATEI, Gheorghe. (2011). "SOA and BPM, a Partnership for Successful Organizations". *Informatica Economica*, vol. 15, no. 4/2011.