

Model Driven Architecture (MDA) Untuk Customization dan Integrasi Layanan Fungsionalitas SIMRS

Slamet Sudaryanto N¹, Fikri Budiman²

^{1,2}Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang 50131

E-mail : slametalica301@gmail.com, fikri@dosen.dinus.ac.id

ABSTRAK

Model Driven Architecture (MDA) adalah arsitektur yang diperkenalkan oleh Object Management Group (OMG). MDA ini merupakan pendekatan untuk spesifikasi sistem dan mendukung interoperabilitas berdasarkan penggunaan model-model formal. Pada model MDA ini memiliki beberapa tahapan. Platform aplikasinya dimulai dari tahapan Computation Independent Model (CIM) yang berisi aturan formal dari proses bisnis modul aplikasi. Selanjutnya Platform Independent Model (PIM), PIM dinyatakan dalam bahasa pemodelan platform independen seperti UML. Selanjutnya PIM diterjemahkan kedalam Platform Specific Model yang bisa diimplementasikan kedalam beberapa platform bahasa pemrograman dengan aturan-aturan formal. Inti dari konsep MDA adalah menggunakan standar penting dari OMG seperti Unified Modeling Language (UML), Meta Object Facility (MOF), XML Metadata Interchange (XMI) dan Common Warehouse Metamodel (CWM). Pada penerapan MDASIMRS (Sistem anajemen Informasi Rumahsakit) ini CIM SIMRS, membuat model target yang berisi semua aturan bisnis yang ditetapkan dalam bisnis inti SIMRS. Pada kasus CIM SIMRS ini target yang ingin dicapai adalah aturan yang diterapkan pada poli rawat jalan. Dimana poli rawat jalan dan tindakan pada masing-masing poli bisa di create secara dinamis dan independen. Tindakan medis bisa di distribusikan pada setiap poliklinik secara mandiri. Selanjutnya PIM SIMRS, membuat model target yang hanya berisi unsur data dalam menentukan model konseptual SIMRS. PIM SIMRS ini merupakan mapping independen antara PIM dan PSM sesuai dengan aturan bisnis dari CIM SIMRS maka didapatkan platform service yang independen yaitu : service create frame modul rawat jalan, create pelayanan medis/non medis, create integras dan distribusi pelayanan medis/non medis serta service alokasi kelompok aplikasi pada user tertentu.

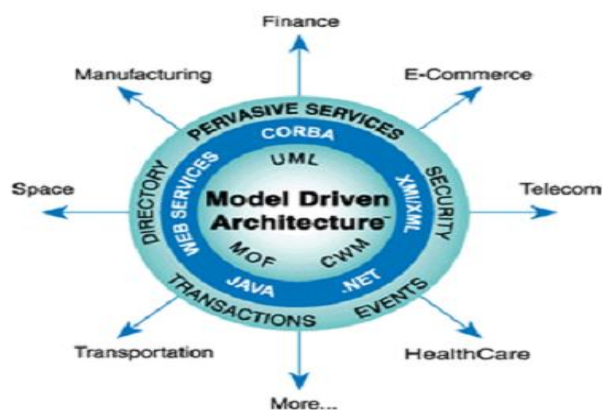
Kata kunci : MDA, CIM, PIM, PSM, SIMRS

1. Pendahuluan

Model Driven Architecture (MDA) tidak sekedar merupakan arsitektur, MDA lebih merupakan strategi untuk mengimplementasikan services. Ide tentang MDA ini adalah bagaimana menyediakan suatu pendekatan yang terbuka dalam menghadapi tantangan perubahan teknologi dan bisnis [2]. MDA merupakan inisiatif pendekatan untuk spesifikasi dan interoperabilitas berdasarkan penggunaan model formal. Sesuai dengan namanya strategi ini memberikan perhatian utama pada model. Model dalam konteks MDA didefinisikan sebagai Platform Independent Model (PIM) yang digunakan untuk menciptakan fungsionalitas system. PIM dibangun menggunakan Unified Modeling Language (UML). Atau standar pemodelan Object Management Group (OMG). Pada model MDA diawali dari PIM yang dinyatakan dalam bahasa pemodelan platform independent seperti UML. Model PIM selanjutnya diterjemahkan ke Platform Spesifik Model (PSM). Inti dari konsep MDA adalah penggunaan standar OMG : Unified Modeling Language (UML), Meta Object facility (MOF), XML Meta Data Interchange (XMI), dan Common Warehouse meramodel (CWM). Standar ini mendefinisikan infrastruktur inti dari MDA dan telah menyumbang state of the art pada pemodelan system. Sebagai proses OMG, MDA merupakan langkah evolusi besar dalam cara OMG mendefinisikan standar operabilitas. MDA memiliki implikasi signifikan pada metamodeling dan Adaptive Obyek Model (AOMs). Metamodeling adalah kegiatan utama dalam spesifikasi atau pemodelan metadata. Interoperabilitas dalam lingkungan heterogen pada akhirnya dapat dicapai dengan melalui metadata bersama dan strategi keseluruhan untuk berbagi dan memahami metadata yang terdiri dari pembangunan otomatis, penerbitan, penagaturan dan interpretasi model. Teknologi AOM menyediakan perilaku system dinamis berdasarkan run-time interpretasi model tersebut. Arsitektur berdasarkan AOMs sangat interoperable, mudah diperluas pada saat run-time dan benar-benar dinamis dalam hal menspesifikasikan perilaku secara keseluruhan (perilaku tidak terikat dalam code-program). Standar inti MDA (UML, MOF, XMI, CWM) membentuk dasar bangunan skema koheren untuk authoring, publishing, managing model dalam MDA.

2. Model Driven Architecture (MDA)

Salah satu aspek fundamental dari MDA adalah kemampuannya untuk mengatasi secara lengkap siklus pengembangan, yang meliputi analisis dan desain, pemrograman, pengujian, perakitan komponen, deployment dan maintenance [4]. MDA sendiri bukanlah spesifikasi baru dari OMG melainkan sebuah pendekatan untuk pembangunan perangkat lunak yang dapat menspesifikasikan keberadaan OMG seperti *Unified Modeling Language* (UML), *Meta Object facility* (MOF), dan *Common Warehouse Metamodel* (CWM). Teknologi lainnya yang diadopsi adalah profil UML *Enterprise Distributed Object Computing* (EDOC) termasuk pemetaan untuk EJB dan CORBA Component Model (CMM). Dengan platform dan teknologi yang bermunculan MDA dapat mempercepat pembangunan spesifikasi baru dengan memanfaatkan proses integrasi MDA. Dengan cara ini maka MDA menyediakan secara komprehensif solusi terstruktur untuk interoperabilitas aplikasi dan portabilitas. Pemodelan yang tepat dari solusi di UML menambah manfaat dari menangkap sifat yang melekat dalam MDA seperti diilustrasikan dalam diagram berikut yang menyajikan berbagai macam layanan yang ‘meresap’ pada MDA tersebut.



Gambar 1 : Model Driven Architecture

3. Sudut Pandang MDA

Dalam sudut pandang MDA menetapkan tiga sudut pandang secara default pada system yaitu : *Computation Independent Model* (CIM), *Platform Independent Model* (PIM) dan *Platform Specification Model* (PSM). CIM berfokus pada pada konteks dan persyaratan system tanpa mempertimbangkan struktur atau pengolahan. PIM berfokus pada kemampuan operasional sebuah system diluar konteks platform tertentu (*set platform*) untuk menampilkan bagian – bagian dari spesifikasi yang lengkap yang dapat diabstraksikan keluar dari platform tersebut. PSM menambah platform independen dengan rincian yang berkaitan dengan penggunaan platform tertentu.



Gambar 2: Tiga Sudut Pandang MDA

3.1. Computation Independent Model (CIM)

CIM juga sering disebut sebagai model bisnis atau domain, untuk itu CIM sering menggunakan kosakata yang familier dengan subyek permasalahan bisnis. Hal ini menyajikan harapan bagaimana system bekerja dengan menyembunyikan

spesifikasi teknologi informasi yang terkait untuk tetap independen terhadap bagaimana system akan dilaksanakan. CIM memainkan peran penting dalam menjembatani kesenjangan yang biasanya ada antara tanggung jawab ahli bisnis dengan dan teknologi informasi dalam penerapan sistem. Dalam spesifikasi MDA persyaratan CIM harus dilacak ke PIM dan konstruksi PSM dalam implementasinya, begitu juga sebaliknya.

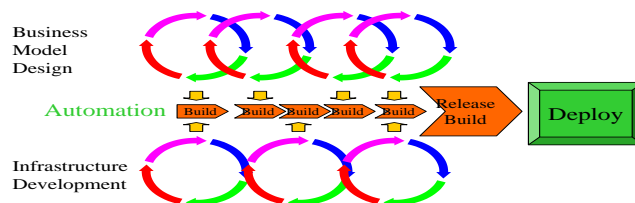
3.2. Platform Independen Model (PIM)

Suatu PIM menunjukkan tingkat independensi sehingga memungkinkan pemetaan untuk satu atau beberapa platform. Hal ini biasanya dicapai dengan mendefinisikan suatu set services dengan cara mengabstraksikan rincian teknis. Model lain kemudian ditentukan spesifikasi pada service ini secara platform tertentu.

3.3. Platform Specification Model (PSM)

PSM menggabungkan spesifikasi di PIM dengan rincian yang diperlukan untuk menetapkan bagaimana system menggunakan jenis platform tertentu. Jika PSM tidak mencakup semua rincian yang diperlukan untuk menghasilkan implementasi dimana platform dianggap abstrak, yang berarti bahwa hal ini tergantung pada model eksplisit atau implicit yang mengandung rincian detail yang diperlukan.

Dengan demikian model MDA merupakan inti dalam mengatasi kebutuhan untuk tingkat fleksibilitas sekaligus mengurangi biaya dan resiko. Kombinasi manfaat visibilitas awal dan implementasi proses yang bertahap serta kombinasi ketersediaan otomatisasi dan pengujian yang berulang (iterative) akan memberi manfaat efektifitas system secara keseluruhan dari waktu ke waktu..



Gambar 3 : Iterative Spiral Process

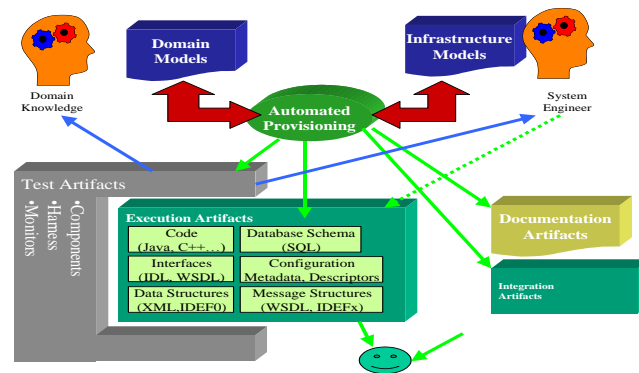
4. Cara Kerja MDA

MDA digunakan untuk membuat dan memelihara system berbasis pada model tingkat tinggi dari domain subjek. MDA memisahkan keawutiran dari arsitektur domain operasional dari infrastruktur teknis. Arsitektur domain operasional mendefinisikan konsep dan struktur yang diperlukan untuk menentukan dan memahami domain. Arsitektur ini mencakup istilah dan konsep proses, aturan, kebijakan, sumberdaya, unit tindakan dan kemampuan domain sehingga model dapat didefinisikan dan disempurnakan. Model domain inti akan disediakan dengan system bersama dengan alat untuk memperluas model ini sebagai system matures, termasuk spesifikasi dasar dan penentuan kebijakan dan proses. Model ini merupakan metadata yang mengatur system pada saat desain dan runtime.

“*Provisioning Model*” juga akan diberikan suatu system, model ini akan menentukan proses dan transformasi yang diperlukan untuk mengimplementasikan domain menggunakan infrastruktur system. Model bawaan akan digunakan untuk mengkonfigurasi alat yang mampu menghasilkan komponen system eksekusi berdasar model domain bawaan. Proses MDA otomatis akan menghasilkan sebagian besar implementasi manual dan menyediakan template keman program menempatkan kosde program untuk algoritma kompleks yang tidak mudah dihasilkan secara otomatisasi.

Model domain, infrastruktur dan penyediaan model yang mampu mampu berkembang secara mandiri, menyediakan pemisahan keprihatinan domain dari keprihatinan teknologi dan pengurangan risiko yang merupakan landasan MDA. Selain memproduksi komponen pelaksanaan, teknologi pengadaan yang sama akan digunakan untuk mengotomatisasi produksi kasus uji dan dokumentasi. Persyaratan, model, dokumen dan sistem selalu disinkronisasi, dapat dilacak dan konsisten. MDA Otomatis penyediaan didasarkan pada model yang disediakan menjadi keduanya pakar domain

(dengan mudah menggunakan antarmuka pengguna) dan insinyur sistem menyediakan teknologi dan spesifikasi infrastruktur. Otomatis MDA ketentuan sebagian besar eksekusi artefak, artefak tes, dokumentasi, spesifikasi integrasi dan kebijakan konfigurasi runtime dan proses yang menghasilkan dan drive sistem.



Gambar 4 : Cara Kerja MDA

5. Desain MDA Pada SIMRS

Beberapa target utama dari platform yang dapat dicapai melalui beberapa tahap dalam membangun aplikasi SIMRS berbasis MDA adalah untuk menciptakan sebuah model platform-independent yang dapat dideskripsikan melalui UML. Model ini kemudian dapat dirubah menjadi satu atau beberapa platform yang spesifik seperti CCM, EJB, .NET, SOAP dan lain-lain. Sebuah sistem yang kompleks dapat terdiri dari beberapa model yang saling terkait dan diselenggarakan secara bersama dengan layer yang didefinisikan sebagai abstraksi, dengan pemetaan yang didefinisikan dari satu set model menjadi model yang lain. Didalam set model global horizontal dapat ditransformasi menjadi suatu layer abstraksi disamping transformasi vertikal sebagai transformasi antar layer.

Dalam menerapkan gaya arsitektur yang konsisten dalam seluruh sudut pandang SIMRS adalah salah satu ilustrasi transformasi horizontal tersebut.

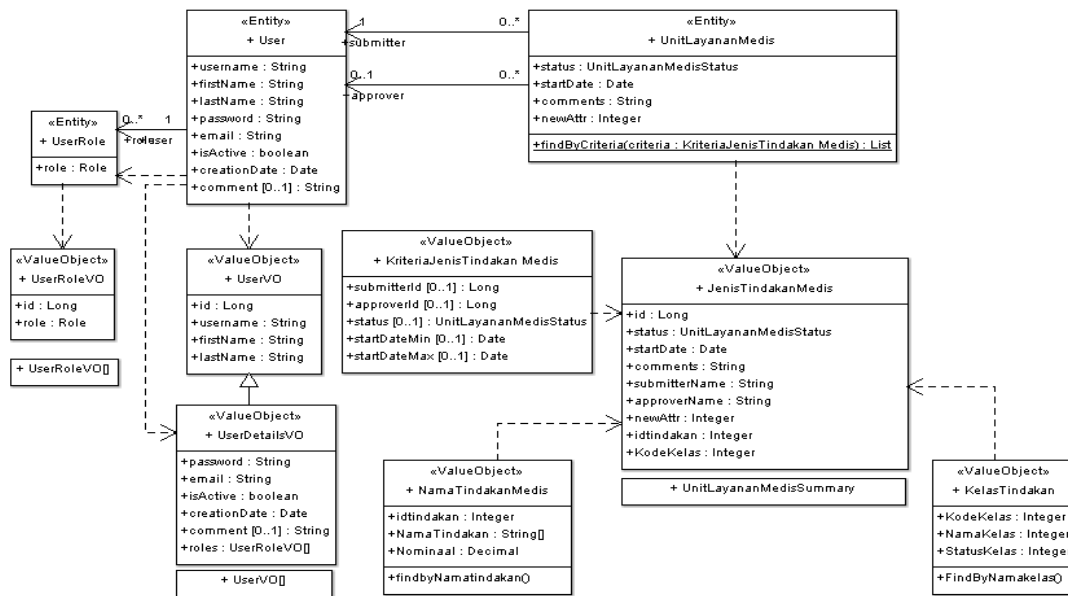
- 1) Dalam CIM SIMRS, membuat model target yang berisi semua aturan bisnis yang ditetapkan dalam bisnis inti SIMRS. Pada kasus CIM SIMRS ini target yang ingin dicapai adalah aturan yang diterapkan pada poli rawat jalan. Dimana poli rawat jalan dan tindakan pada masing-masing poli bisa di create secara dinamis dan independen. Tindakan medis bisa di distribusikan pada setiap poliklinik secara mandiri.
- 2) Dalam PIM SIMRS, membuat model target yang hanya berisi unsur data dalam menentukan model konseptual SIMRS. PIM SIMRS ini merupakan mapping independen antara PIM dan PSM sesuai dengan aturan bisnis dari CIM SIMRS maka didapatkan platform service yang independen yaitu :
 - a. Service create poli rawat Jalan
 - b. Service create tindakan medis rawat jalan
 - c. Service create integrasi dan distribusi tindakan medis/non medis pada poliklinik.
 - d. Service alokasi kelompok aplikasi pada user tertentu.
- 3) Dalam PSM SIMRS , membuat target model sebagai unit fungsional tertentu yang independen. Dimana Target model dalam SIMRS ini adalah *framework* modul poli rawat jalan yang bisa diberi buat secara dinamis untuk diberi fungsi pelayanan tindakan medis dan nonmedis sesuai dengan kebutuhan poli tersebut.

5. Hasil Pembahasan

5.1. Penentuan Kebutuhan CIM SIMRS

Tahap ini bertujuan untuk membangun CIM yang menggambarkan proses bisnis organisasi (bagian rawat jalan) yang

telah dilakukan analisis. Proses bisnis tersebut dapat menggambarkan lingkungan dan kebutuhan dari sistem perangkat lunak yang akan dibangun. Tahap awal yang dilakukan yaitu menentukan deskripsi umum dari perangkat lunak hasil solusi permasalahan, termasuk fungsi secara umum, pengguna perangkat lunak, batasan, dan asumsi yang digunakan dalam pengembangan. Dari deskripsi umum tersebut dapat didefinisikan kebutuhan sistem perangkat lunak yang akan dibangun. Terdapat dua jenis kebutuhan yang dijabarkan di sini, yaitu kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional menyatakan fitur-fitur utama perangkat lunak, sedangkan kebutuhan nonfungsional yang menyatakan fitur-fitur tambahan untuk mendukung kinerja fitur utama tersebut.



Gambar 5 : Komponen View Unit Layanan Medis

5.2. Analisis PIM SIMRS

Dalam tahap analisis PIM ini bertujuan untuk membangun PIM sesuai dengan kebutuhan. PIM ini fokus pada operasi sistem rawat jalan tanpa tergantung pada platform tertentu. PIM dibuat dari transformasi CIM yang sesuai dengan prinsip MDA. PIM SIMRS ini merupakan mapping independen antara PIM dan PSM sesuai dengan aturan bisnis dari CIM SIMRS maka didapatkan platform service yang independen yaitu :

- Service create poli rawat Jalan
- Service create pelayanan medis rawat jalan
- Service create integrasi dan distribusi tindakan medis/non medis pada poliklinik.
- Services alokasi kelompok aplikasi pada user tertentu.

5.3. Desain PSM

Pada tahap ini tujuan utamanya adalah membangun PSM yang sesuai dengan platform dan service yang didapatkan saat analisa PIM. Dengan demikian pada tahap ini semua services harus terbangun secara independen dan dapat didistribusikan dan diintegrasikan sehingga bisa menjadi bangunan yang utuh sesuai

dengan target model. Dengan kata lain target model adalah integrasi kebutuhan service yang sesuai dengan aturan bisnis organisasi.

KODE UNIT	NAMA UNIT	STAT. AKTIF	IJIN AKSES	STAT. LAYAN
P13	POLI ULMUM	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
p33	poli jiwa	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
p34	poli rambut	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
p35	poli kecantikan	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
P66	POLI CACAR	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
U10	P	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

BARU HAPUS
 KODE : U10
 NAMA UNIT : POLI UMUM
 NO. IP TERMINAL : 192.168.0.10
 -STATUS AKTIF: TIDAK YA
 -IJIN AKSES PROGRAM: TIDAK YA
 -UNIT PELAYANAN: TIDAK YA
 -BANGSAL: TIDAK YA
 REKAM
 KEMBALI

Gambar 6 : Service Create Poli/Unit Poli Umum (rawat jalan)

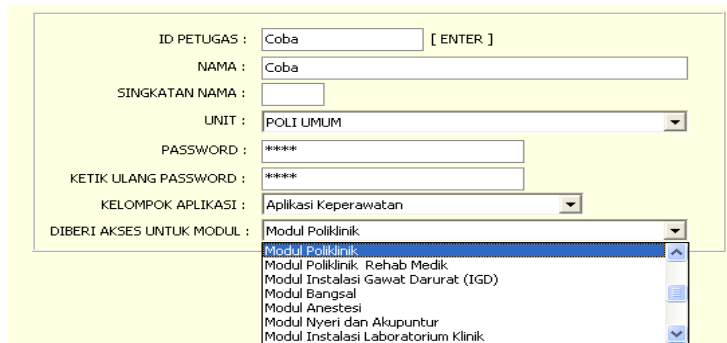
UNTK JENIS PASIEN : PASIEN UMUM
 KELOMPOK : TINDAKAN MEDIK NON OPERATIF DI POLI UMUM
 KODE : 1.63.01
 DESKRIPSI : Tindakan Asuhan Keperawatan
 TIPE : HEADER DETAIL
 TARIP DIBAGI DALAM KELAS : TIDAK YA
 REKAM
 KEMBALI

Gambar 7 : Service Create Pelayanan Medis

KODE UNIT (SMF)	NAMA UNIT (SMF)
P08	POLI KULIT & KELAMIN
P09	POLI MATA
P10	POLI REHAB MEDIK
P11	POLI SYARAF
P12	POLI THT
P13	POLI UMUM

KODE	DESKRIPSI	PILIHAN
1.63.99	PEMERIKSAAN DI POLI UMUM	<input type="checkbox"/>
1.63.Coba	Coba Masukan Tindakan	<input checked="" type="checkbox"/>
2.63.99	PEMERIKSAAN DI POLI UMUM	<input type="checkbox"/>
3.63.99	PEMERIKSAAN DI POLI UMUM	<input type="checkbox"/>
4.63.99	PEMERIKSAAN DI POLI UMUM	<input type="checkbox"/>

Gambar 8 : Service Mapping Pelayanan Medis Pada Poli Umum



ID PETUGAS : [ENTER]
NAMA :
SINGKATAN NAMA :
UNIT :
PASSWORD :
KETIK ULANG PASSWORD :
KELOMPOK APLIKASI :
DIBERI AKSES UNTUK MODUL :
Modul Poliklinik
Modul Poliklinik Rehab Medik
Modul Instalasi Gawat Darurat (IGD)
Modul Bangsal
Modul Anestesi
Modul Nyeri dan Akupuntur
Modul Instalasi Laboratorium Klinik

Gambar 9 : Service Create User dan Mapping menjadi Target Aplikasi

5. Penutup

Berdasarkan kajian dan evaluasi pada penerapan MDA prototype SIMRS tersebut, dapat dihasilkan sebuah gambaran dan pedoman yang lengkap mengenai perancangan perangkat lunak dengan pendekatan MDA. Penerapan pendekatan MDA pada aplikasi SIMRS tersebut dapat diambil kesimpulan sebagai berikut:

- SIMRS merupakan kumpulan aturan bisnis pelayanan medis dan non medis yang dapat diuraikan menjadi frame platform aplikasi yang tersusun menjadi beberapa spesifikasi fungsi, dimana spesifikasi fungsi dapat diuraikan menjadi beberapa service dasar pembentuk fungsi.
- Service dasar didesain dan dikelola secara independen sehingga bisa didistribusikan dan diintegrasikan pada frame aplikasi secara mandiri dalam membentuk target aplikasi. Dalam kasus ini target aplikasi dibangun dari service create aplikasi, create pelayan medis, mapping pelayanan medis pada aplikasi. Selanjutnya create user dan mapping aplikasi yang sudah memiliki layanan medis.
- Dalam pendekatan MDA pada SIMRS ini maka SIMRS bisa memiliki performansi yang baik seperti portability, interoperability, reusability.

Namun demikian untuk pengembangan selanjutnya dapat menggunakan *framework* PIM4SOA, web service dan transformasi antar model, terutama pengembangan perangkat pendukungnya dengan teknologi terkini. Dengan demikian *framework* tersebut dapat diterapkan dengan baik sesuai dengan perubahan dan perkembangan teknologi yang terbaru.

DAFTAR PUSTAKA

- Oswalds, G (2001). *Definition of Enterprise Architecture – Centric Models for The System Engineers*, TASC Inc.
- Almeida, J. P. dkk.. (2009). Model-Driven Service-Oriented Architectures. *International Journal Business Process Integration and Management*, Vol.4, No.1, 2-4.
- Sarno, R. and Herdiyanti A (2010). “ A Service Portofolio for an Enterprise Resource Planning” ; *International Journal of Computer Science and Network Security*, ISSN -1738 – 7906, Vol. 10, No. 3, app. 144-156.
- Larrucea, Xabier dkk.. (2007). MDSOA for Achieving Interoperability. *Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'07)*, 247.
- ATHENA. (2005). Deliverable DA1.3.1 – Report on Methodology description and guidelines definition. *Integrating and Strengthening the European Research in Advance Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application*. http://interop-vlab.eu/ei_public_deliverables/athena-deliverables/A1/d-a1-3.1, diakses 20 Maret 2012.
- Benguria, Gorka dkk.. (2007). A Platform Independent Model for Service Oriented Architecture. *International Conference on Interoperability of Enterprise Software and Applications 2006*, 23-32. Bordeaux: Springer.
- IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. NJ: IEEE.
- Mahfudhi, Muhammad Ghuftron. (2012). *Kajian dan Evaluasi Pengembangan Pedoman Model Driven Service Oriented Architecture, Studi Kasus: Rekam Medis Rawat Jalan*.