

RANCANG BANGUN SISTEM PERSEDIAAN (INVENTORY) DENGAN MODEL SOFTWARE AS A SERVICE MENGUNAKAN SERVICE ORIENTED ARCHITECTURE

Ahmad Rifai¹, Riyanarto Sarno², Dwi Sunaryono³

^{1,2,3} Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo Surabaya, 60111, Indonesia

¹ riff.blend@gmail.com, ²riyanarto@its-sby.edu, ³dwi@its-sby.edu

Abstrak

Persediaan (Inventory) adalah sistem manajemen dalam menentukan keseimbangan antara investasi penyimpanan persediaan dengan pelayanan pelanggan. Sistem persediaan adalah salah satu bagian dari ERP (enterprise resource planning). Manajemen persediaan yang baik sangatlah penting untuk menekan biaya pengeluaran suatu perusahaan. Di satu sisi, sebuah perusahaan dapat mengurangi biaya dengan mengurangi persediaan. Di sisi lain, produksi dapat berhenti dan pelanggan menjadi tidak puas, ketika sebuah barang tidak tersedia. Solusi informatika perangkat lunak inventory adalah salah satu cara efisiensi perusahaan dalam aktifitas proses bisnis transaksi gudang. Namun pada pelaksanaan implementasi perangkat lunak, beberapa perusahaan dihadapkan masalah dalam ketersediaan infrastruktur perangkat keras yang diperlukan untuk mendukung perangkat lunak.

Permasalahan ini mendorong munculnya sebuah pendekatan baru dalam pengembangan perangkat lunak, yang disebut dengan Software as a Service (Perangkat Lunak sebagai Layanan). Suatu konsep dimana terdapat satu aplikasi terpusat yang dikelola oleh suatu penyedia jasa perangkat lunak, yang dapat menangani beberapa data perusahaan yang berbeda. Dengan konsep ini, penyimpanan data dan transaksi pada aplikasi inventory dipisah pada schema yang berbeda. Agar dalam penggunaan aplikasi oleh lebih dari satu perusahaan ini tidak saling mempengaruhi.

Aplikasi ini dibangun dengan menggunakan analisis dan desain SOAD. Dan berhasil diimplementasikan menggunakan teknologi Java serta web service sebagai komunikasi antar domain fungsi yang dilakukan dalam enterprise resource planning. Aplikasi dalam tugas akhir ini juga mengimplementasikan konsep Software as a Service (SaaS) dengan membagi hak otorisasi untuk schema yang berbeda.

Aplikasi yang dibuat pada tugas akhir ini mampu mengakomodasi fitur Inventory yaitu pembelian material (purchasing) dan pengiriman barang (delivering). Dari data transaksi purchasing dan delivering, aplikasi ini dapat memperkirakan

jumlah yang optimal suatu barang yang tersimpan di gudang (controlling stock). Sehingga tidak terjadi kelebihan stok maupun kekosongan stok.

Kata kunci : sistem persediaan, inventory, SOA, SOAD, ERP, service

1. Pendahuluan

Dalam pengembangan teknologi informasi, seharusnya perusahaan mempertimbangkan beberapa aspek salah satunya yaitu biaya agar penggunaan biaya ini dapat tepat sasaran sehingga tidak mengeluarkan biaya yang sangat besar untuk maintenance ataupun memodifikasi aplikasi.

Software as a service (SaaS), memungkinkan salah satu cara penghematan biaya dalam penggunaan software. Dengan pengimplementasian desain perangkat lunak yang baik, SaaS menyediakan keuntungan penggunaan perangkat lunak tanpa harus berurusan dengan kerumitan dan potensi biaya tinggi dari penyediaan perlengkapan infrastruktur perangkat keras dan aplikasi yang dibutuhkannya.

SOA (Service Oriented Architecture) adalah suatu model arsitektural untuk membangun solusi enterprise berdasarkan service. Secara lebih spesifik, SOA berhubungan dengan pembangunan independen dari layanan bisnis yang dapat dikombinasikan menjadi proses bisnis pada level tinggi dan solusi dalam konteks enterprise.

Tujuan utama dari Service Oriented Architecture (SOA) adalah untuk menunjang dunia usaha dengan dunia teknologi informasi (TI) dengan cara membuat keduanya lebih efektif. SOA adalah sebuah jembatan yang menciptakan suatu simbiosis dan hubungan sinergis antara keduanya yang lebih kuat. Selain itu, SOA adalah tentang hasil usaha yang dapat dicapai dengan keselarasan yang lebih baik antara bisnis dan TI.

2. Kajian Pustaka

Bagian ini menjelaskan dasar teori yang digunakan dalam pembuatan makalah.

2.1 Inventory (Persediaan)

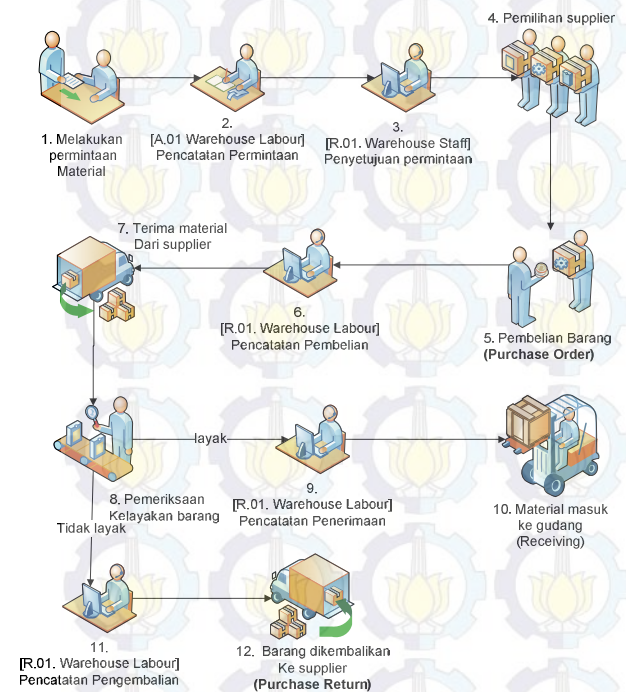
Sistem persediaan (inventory) adalah salah satu aset termahal dari banyak perusahaan, mewakili sebanyak 50% dari keseluruhan modal yang diinvestasikan^[1].

Perencanaan sumber daya perusahaan, atau sering disingkat ERP dari istilah bahasa Inggrisnya, *enterprise resource planning*, adalah sistem informasi yang diperuntukkan bagi perusahaan manufaktur maupun jasa yang berperan mengintegrasikan dan mengotomasi proses bisnis yang berhubungan dengan aspek operasi, produksi maupun distribusi di perusahaan bersangkutan.

Aplikasi Inventory yang akan dikembangkan adalah salah satu bagian dari perangkat lunak ERP (Enterprise Resource Planning).

Purchasing

Purchasing atau pembelian barang merupakan unit bisnis yang melayani transaksi pembelian barang kebutuhan perusahaan pada supplier penyedia barang. Pada proses purchasing terbagi menjadi beberapa sub bisnis proses yang dijelaskan sebagai berikut :



Gambar 1 Alur Kerja Purchasing

1. Purchase Request

Purchase Request merupakan transaksi yang digunakan untuk mencatat permintaan pembelian atas barang yang dibutuhkan.

2. Purchase Order

Purchase Order merupakan transaksi yang digunakan untuk mencatat pembelian barang kepada supplier.

3. Receiving

Receiving merupakan transaksi yang digunakan untuk mencatat penerimaan barang dari supplier.

4. Purchase Return

Purchase Return merupakan transaksi yang digunakan untuk mencatat pengembalian barang ke supplier.

Delivering

Delivering merupakan unit bisnis yang melayani transaksi barang dibeli oleh pelanggan. Terdapat 2 transaksi pada delivering yaitu :

1. Delivery Order

Delivery Order merupakan transaksi yang digunakan untuk mencatat barang yang telah dibeli oleh konsumen.

2. Sales Return

Sales Return merupakan transaksi yang digunakan untuk mencatat pengembalian barang dari pelanggan.

Controlling Stock

Controlling Stock atau pengecekan stock barang di gudang merupakan unit bisnis yang melayani proses pengecekan kondisi stock barang di gudang. Pada proses controlling stock terbagi menjadi beberapa sub bisnis proses yang dijelaskan sebagai berikut :

1. Stock Management

Stock Management bertujuan untuk menentukan keseimbangan barang yang ada di gudang.

2. Stock Adjustment

Stock Adjustment merupakan transaksi penyesuaian untuk menambah atau mengurangi jumlah saldo persediaan barang.

3. Item Balance

Item Balance merupakan transaksi yang mencatat perubahan stock barang pada gudang dalam suatu periode tertentu.

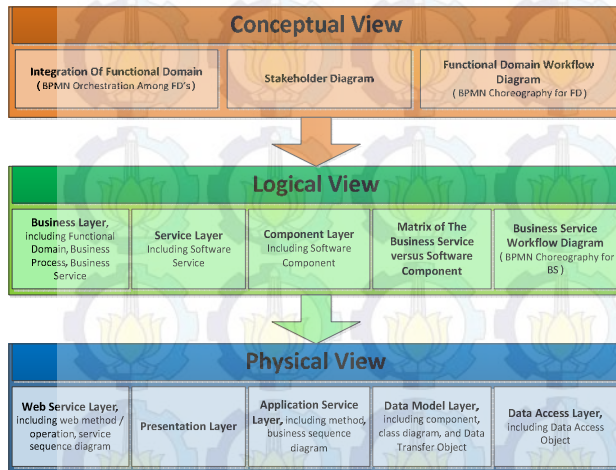
4. Location Transfer

Location Transfer merupakan transaksi untuk mencatat proses pemindahan barang dari satu lokasi gudang ke lokasi gudang lainnya.

2.2 Service Oriented Analysis and Design (SOAD)

SOAD (Service Oriented Analysis and Design) membantu dalam proses perancangan sistem dengan menggunakan metode SOA. SOAD mencakup tahapan proses analisis kebutuhan dan perancangan desain sistem yang menjadi dasar arsitektur suatu sistem.

Untuk mengidentifikasi masalah dengan menggunakan pendekatan top-down dalam SOAD, service portfolio menyediakan tiga tingkatan view seperti yang ditunjukkan pada Gambar 2^[2].



Gambar 2 Lapisan SOAD

Perancangan Service Oriented Architecture yang dilakukan pada Tugas akhir kali ini mengacu pada SOAD beserta lapisannya.

Conceptual View

Conceptual View adalah suatu level design tingkat atas yang hanya meliputi bagian komponen dan entitas terpenting. Tujuan utama suatu desain konseptual adalah untuk memberikan gambaran keseluruhan dari tujuan sistem. *Conceptual View* terdiri dari Business Service, Software Service, dan Software Component.

Logical View

Logical view adalah suatu desain yang lebih detail yang memasukkan seluruh komponen dan entitas utama ditambah dengan relasi antar komponen atau entitas tersebut. Aliran data dan koneksi didetailkan dalam tahapan ini. Dalam analisis tugas akhir ini, logical view terdiri dari Activity diagram Business Service, beserta Sequence diagramnya yang menggambarkan alur proses.

Physical View

Physical view terdiri dari seluruh komponen dan entitas yang berada dalam server, software service, object, atau solution. Pada analisis tugas akhir ini, Physical view merupakan seluruh layering dan bagian yang terdapat dalam arsitektur aplikasi untuk mendukung Service Oriented Architecture. Physical view meliputi Data Access Layer, Business Layer, Service Layer, Web Services, dan Presentation Layer.

3. Analisis Perangkat Lunak

Pada bagian ini membahas tahap analisis dan desain perangkat lunak dan kebutuhan bisnis dari proses bisnis yang ada pada sistem persediaan.

3.1 Desain Conceptual View

Conceptual View merupakan desain analisa tingkat atas yang mudah dipahami oleh pelaku bisnis

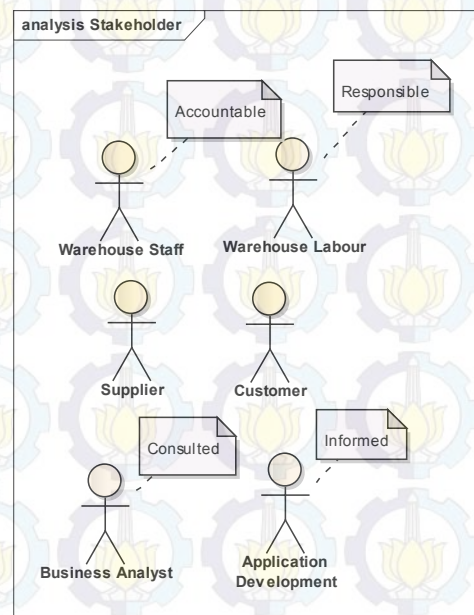
sebagai salah satu *stakeholder*. Gambar 3 adalah diagram stakeholder. Diagram stakeholder adalah diagram yang menggambarkan *actor* yang terlibat dalam penggunaan aplikasi ini.

Functional domain workflow diagram adalah desain diagram yang menggambarkan alur kerja dari setiap proses bisnis dalam aplikasi sistem persediaan ini. Gambar 4 menggambarkan alur kerja pada bisnis proses *delivery order*.

3.2 Desain Logical View

Desain *logical view* adalah suatu desain yang menjembatani antara *conceptual view* dan *physical view*. Dalam analisis ini, logical view terdiri dari pemetaan domain fungsi, matriks *business service*, dan *Business Services Workflow Diagram*.

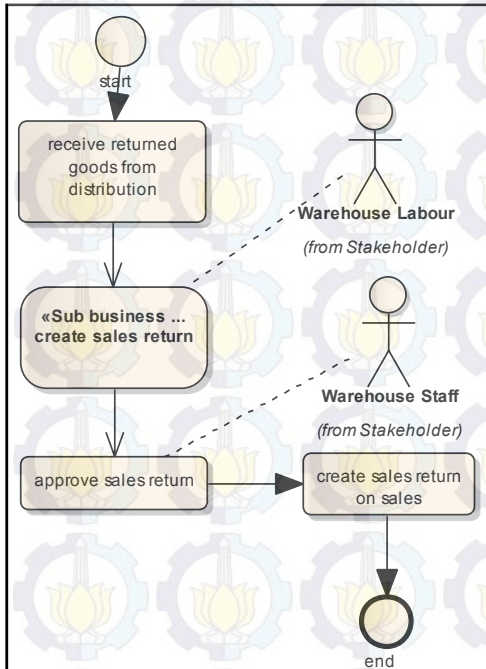
Pemetaan domain fungsi pada inventory terdiri dari 3 lapisan yaitu lapisan bisnis, lapisan layanan, dan lapisan komponen.



Gambar 3 Diagram Stakeholder

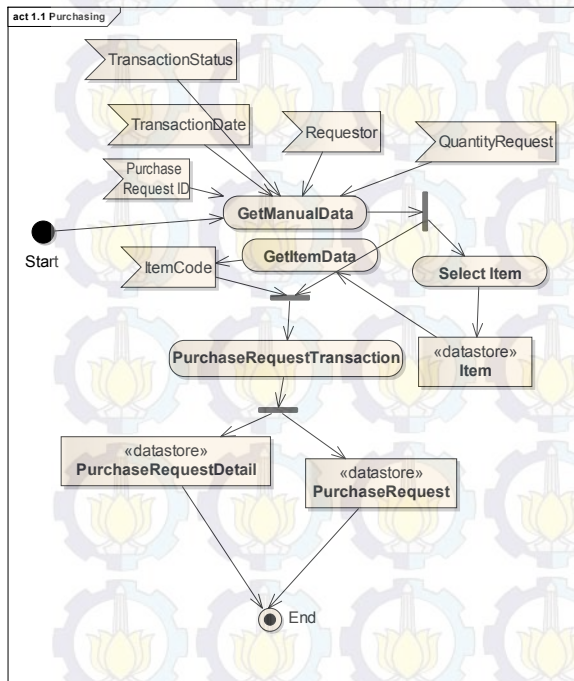
Pada lapisan bisnis terdiri dari 3 lapisan, yaitu domain fungsi, proses bisnis, dan bisnis proses. Domain fungsi dari analisa ini adalah *inventory*. Inventory memiliki 3 proses bisnis yaitu *Purchasing*, *Delivering*, dan *Controlling Stock*.

Pada setiap bisnis proses akan menghasilkan *business service*. *Business service* terdiri dari *software service* dan *internal service*.



Gambar 4 Alur Kerja Delivery Order

Business service workflow diagram adalah diagram yang menggambarkan alur kerja dari setiap business service aplikasi sistem persediaan. Interaksi ini digambarkan dengan sebuah diagram aktivitas (*activity diagram*) seperti yang digambarkan pada Gambar 5.



Gambar 5 Alur Kerja Business Service Purchase Request

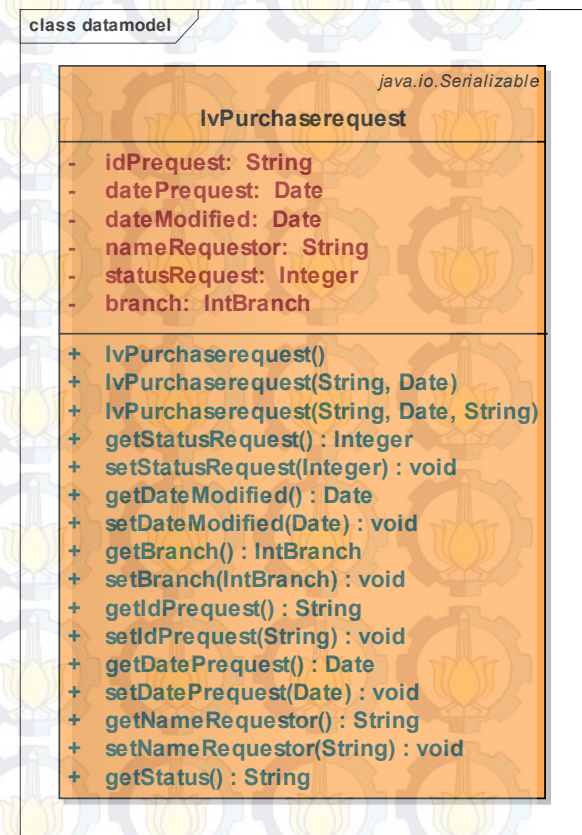
3.3 Desain Physical View

Physical view merupakan bagian yang merealisasikan dari desain pada conceptual view dan logical view. Pada physical view menggambarkan perancangan sistem yang terdiri dari *presentation layer*, *application service layer*, *web service layer*, *data model*, dan *data access layer*.

Data Model adalah permodelan konsep model bisnis yang diimplementasikan ke dalam *class*.^[3] Domain model berisi class mapping hibernate yang mendefinisikan objek yang terdapat pada database relasional.

Misalnya pada transaksi *purchase request*, jika diimplementasikan dalam bentuk class, maka akan seperti pada Gambar 6. Objek transaksi ini memiliki atribut id, tanggal transaksi, nama peminta, dan status transaksi.

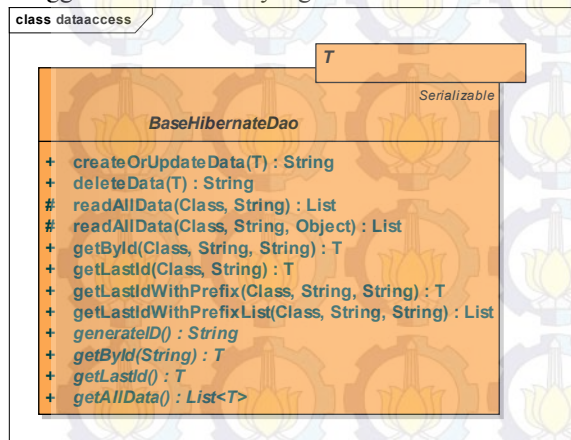
Lapisan *Data Access* adalah lapisan di dalam aplikasi sistem persediaan yang menangani mekanisme yang berhubungan dengan database yaitu *create*, *read*, *update*, dan *delete*. Semua class pada lapisan ini mengimplementasikan abstract class *basehibernatedao* yang ada pada Gambar 7.



Gambar 6 Data Model Class IvPurchaserequest

Aplikasi ini menggunakan konsep ORM, yang artinya pada lapisan *data access* tetap menggunakan

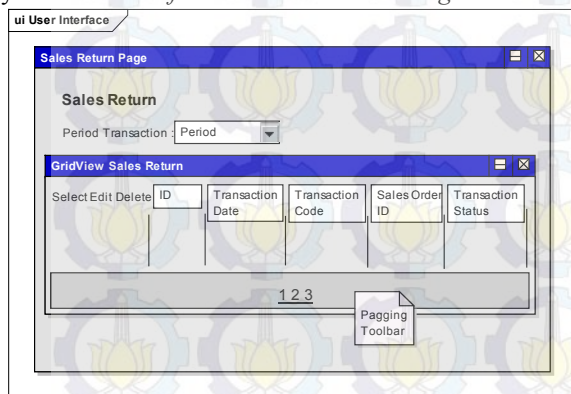
konsep pemrograman berorientasi objek, walaupun menggunakan *database* yang bersifat relasional.



Gambar 7 Class BaseHibernateDao

Lapisan *Application Service* adalah lapisan yang menangani permasalahan *business logic*. Business Logic terdiri dari sejumlah operasi yang menggunakan data. Data dimodelkan dengan entitas bisnis dalam domain permasalahan.

Lapisan presentasi bertanggungjawab terhadap tampilan yang ditampilkan ke pengguna. Presentation Layer hanya bertugas sebagai media aplikasi berinteraksi dengan pengguna. *Presentation layer* tidak bertanggung jawab terhadap pengolahan data. Presentation layer terbagi menjadi 2 komponen yaitu *User Interface* dan *Presentation Logic*.



Gambar 8 Desain Antar Muka Sales Return

Web service layer adalah lapisan yang bertanggung jawab terhadap implementasi desain web service.

4. Uji Coba dan Evaluasi

Uji coba dibagi menjadi 2 yaitu uji coba aplikasi dan uji coba *web service* yang dihasilkan oleh *Inventory*. Pada uji coba yang akan dilakukan diberikan beberapa skenario untuk mengetahui fungsionalitas-fungsionalitas dari program. Uji coba dilakukan mulai dari proses input data dan diproses sampai akhirnya menghasilkan output.

4.1 Uji Coba Aplikasi

Skenario ini merupakan uji coba untuk mengetahui fungsionalitas internal aplikasi untuk proses transaksi permintaan barang (*Purchase Request*). Adapun data-data masukan yang akan digunakan akan dijelaskan pada **Error! Reference source not found.**

Tabel 1 Data Masukan Transaksi Purchase Request

Transaction Date	Requestor
01-JUL-2011	Warehouse Labour

Sedangkan detail data masukan transaksi purchase request dijelaskan pada Tabel 2.

Tabel 2 Data Masukan Detail Transaksi Purchase Request

Item	Quantity
Teak 10	40
Medium-density fibreboard 9	35

Sedangkan skenario jalannya aplikasi dijelaskan pada Tabel 3.

Tabel 3 Skenario Transaksi Purchase Request

Tujuan	Menguji fungsionalitas transaksi purchase request	Cek
Aksi Aktor	Reaksi Aplikasi	Cek
User memilih tautan 'new' pada halaman purchase request	Aplikasi menampilkan halaman detail purchase request	OK
User memasukkan data uji ke dalam form detail transaksi purchase request dan menekan tombol add detail.	Aplikasi menampilkan data purchase request dan purchase request detail	OK
User menekan tombol 'save'	Aplikasi menyimpan data purchase request.	OK

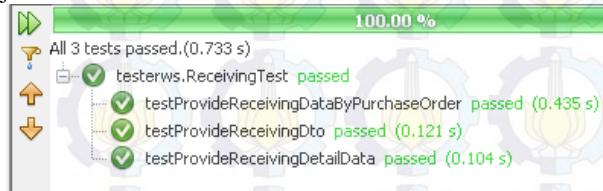
4.2 Uji Coba Web Service

Skenario ini merupakan uji coba bertujuan untuk mengetahui fungsionalitas jalannya web service yang disediakan untuk kebutuhan informasi proses bisnis pada aplikasi ini.

Tabel 4 Uji Coba Web Service ProvidingReceiving

No	Web Method	Parameter	Exp	Res	Stat
1	Provide Receiving DataBy Purchase Order	PO.000013	Not null	Not null	ok
2	Provide Receiving Dto	RC.000003	Not null	Not null	ok
3	Provide Receiving DetailData	RC.000003	Not null	Not null	ok

Gambar 9 menampilkan hasil uji coba yang dilakukan dengan menggunakan unit testing pada java.



Gambar 9 Hasil Uji Coba Web Service ProvidingReceiving

5. Kesimpulan

Dari hasil pengamatan mulai tahap analisis, perancangan, implementasi dan uji coba, penulis mengambil kesimpulan sebagai berikut:

1. Untuk menghasilkan arsitektur SOA yang baik, harus dilakukan suatu analisis dan desain berorientasi Service (SOAD) dengan mengikuti suatu acuan. Dari hasil analisis SOAD, dilakukan Mapping ke dalam arsitektur aplikasi enterprise beserta teknologi yang digunakan. Ini dilakukan agar aplikasi yang dibuat tetap konsisten dengan hasil analisis dan desain.
2. Untuk menghasilkan suatu aplikasi Enterprise yang baik, diperlukan suatu arsitektur yang baik yang dapat memenuhi permintaan dari suatu arsitektur SOA yaitu reusability, loose coupling dan integration.
3. Analisis proses bisnis sangat penting dilakukan untuk dapat menghasilkan suatu aplikasi inventory yang baik.

Daftar Pustaka

- [1] Heizer, J., & Render, B. (2010). *Manajemen Operasi*. Jakarta: Salemba Empat.

- [2] Sarno, R., & Herdiyanti, A. (2010, Maret). A Service Portfolio for an Enterprise Resource Planning. *IJCSNS International Journal of Computer Science and Network Security*, 10.

- [3] Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley.