# untitledJURUSAN TEKNIK INFORMATIKA

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

Nama : **Husein Alhamid**

NRP : **5109 100 162**

Dosen Wali :  **Arya Yudha W, S.Kom,MT**

1. **JUDUL TUGAS AKHIR**

**Rancang Bangun Perangkat Lunak Berorientasi Arsitektur *Service* (SOA) dengan Pendekatan *Workflow* pada Portal untuk Sistem *Enterprise Resource Planning***

1. **LATAR BELAKANG**

Enterprise resource planning (ERP) merupakan salah satu solusi sistem informasi terintegrasi dan terpadu yang digunakan oleh sebuah perusahaan dalam menjalankan bisnisnya. ERP mencakup beberapa domain fungsi yang saling terintegrasi sehingga menjadi sebuah kesatuan sistem yang tepadu. Dalam ERP yang arsitekturnya berorientasi servis, sebuah domain fungsi berdiri sendiri.

Implementasi ERP dapat dilakukan dengan cukup mudah apabila struktur perusahaan simpel. Namun ketika perusahaan mempunyai struktur yang kompleks, implementasi ERP akan lebih kompleks, sulit ,unik, dan akan menghadapi tantangan di tingkat teknikal dan manajerial. Perusahaan biasanya melakukan perencanaan pada level manajerial sebelum mengimplementasikannya dalam level teknikal (software dan hardware). Ada satu solusi yang sedang booming akhir-akhir ini yaitu metode Software as a Service, yaitu memberikan kemudahan pada pengguna untuk pengimplementasian suatu sistem dengan meminimalkan penggunaan infrastuktur pada perusahaan tersebut. Dengan metode ini perusahaan hanya membayar jasa kepada vendor atas sistem yang digunakannya.

SOA (Service Oriented Architecture) merupakan model arsitektur untuk membuat aplikasi yang bersifat loose coupling, sehingga dapat di reuse. Aplikasi pada SOA dibangun berdasarkan pada servis, yang merupakan implementasi dari fungsi bisnis yang sudah terdefinisi dengan jelas dan dapat dikonsumsi oleh aplikasi lain maupun proses bisnis yang lain. Seringkali sebuah perusahaan dalam membangun aplikasi bersifat tambal sulam dan kurang terintegrasi dengan sistem aplikasi lainnya sehingga menimbulkan dampak semakin lama perusahaan akan semakin banyak memiliki aplikasi yang berbeda-beda, baik dari segi teknologi yang dipakai, standardisasi yang digunakan, cara pemakaian, maupun vendor pembuatnya. Masalah integrasi antar aplikasi saat ini memang telah menjadi isu yang cukup serius, khususnya bagi perusahaan-perusahaan berskala *enterprise.* Disinilah professional IT secara langsung menghadapi tantangan untuk mampu membangun dan mengontrol teknologi dalam batasan perusahaan/enterprise dan heterogenitasnya, yaitu suatu sistem yang berbasis *service-oriented*. Arsistektur sistem yang berbasis *service* dikenal dengan istilah *Service Oriented Architecture* (SOA).

*Service Oriented Architecture* (SOA) 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*. Dengan menerapkan metode SOA, permasalahan yang ada terkait dengan perkembangan dunia bisnis dapat terselesaikan. SOA merupakan suatu metode pengembangan berbasis arsitektur yang memodularisasi sistem informasi menjadi *services* dengan cara mengelompokkan proses bisnis perusahaan. SOA menawarkan sebuah infrastruktur yang baik dengan kemampuan integrasi yang sudah terkelola, yang tentunya akan berdampak pada meningkatnya reliability, kemudahan pertukaran informasi antar aplikasi, mengurangi pengaruh jika terjadi perubahan, dan yang tidak kalah pentingnya adalah dapat menekan biaya-biaya untuk keperluan integrasi dan modifikasi aplikasi. Pada akhirnya SOA akan membantu organisasi dalam melakukan proses bisnis secara lebih efisien dan beradaptasi dengan perubahan lingkungan.[1]

1. **RUMUSAN MASALAH**

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut “Bagaimana mengimplementasikan SOA untuk mengintegrasikan berbagai aplikasi menjadi Enterprise Resource Planning yang diimplementasikan menggunakan framework .NET”.

1. **BATASAN MASALAH**

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

* + - 1. Tugas akhir ini merancang dan membuat integrasi beberapa sistem aplikasi yang dibuat dengan berorientasikan servis untuk menghasilkan aplikasi Enterprise Resource Planning yang terintegrasi.
      2. Session handling, faulthandling,registring,
      3. ERP yang dibangun berorientasi pada aspek multi-tenant.
      4. Framework yang digunakan untuk menggunakan framework .NET dengan menggunakan teknologi, WCF, ASP .NET, Nhibernate,WWF.
      5. Database yang digunakan adalah SQLServer.

1. **TUJUAN TUGAS AKHIR**

Tugas akhir ini bertujuan untuk mengintegrasikan servis-servis dari berbagai aaplikasi yang terpisah untuk diorkestrasikan menjadi aplikasi ERP yang terintegrasi dengan menggunakan framework .NET.

1. **MANFAAT TUGAS AKHIR**

Manfaat pembuatan tugas akhir ini adalah mengurangi ketergantungan antar aplikasi penyusun ERP dalam menggunakan servis yang disediakan oleh provider dan menghasilkan aplikasi ERP yang terintegrasi dengan single sign-on.

1. **TINJAUAN PUSTAKA**

**8.1 Service Oriented Architecture**

SOA (service oriented architecture, arsitektur berorientasi layanan) merupakan sebuah arsitektur yang menghubungkan berbagai service (pada prinsipnya aplikasi-aplikasi dan data) sesuai dengan kebutuhan dan membentuk sebuah struktur aplikasi besar sebagai koleksi *ad hoc* dari service-service tersebut, untuk mencapai hasil yang diinginkan oleh service consumer (baik end user atau service lain). Fungsi-fungsi ini tidak terikat dengan sistem operasi dan bahasa pemrograman yang mendasari aplikasi-aplikasi tersebut[2]

Terdapat 10 prinsip arsitektural dalam merancang dan mendefinisikan service, antara lain :

1. Service encapsulation, sebagian besar service (web service) dikonsolidasikan untuk digunakan dengan berdasar pada SOA
2. Service loose coupling, service-service memelihara sebuah hubungan yang meminimalisasikan ketergantungan antar service
3. Service contract, service terpaku pada sebuah communication agreement, yang didefinisikan secara umum pada satu atau lebih service description documents
4. Service abstraction, di luar apa yang dideskripsikan dalam service contract, setiap service menyembunyikan logikanya dari dunia luar
5. Service reusability, logika dibagi ke dalam service-service yang berorientasi pada prinsip reusability (penggunaan kembali)
6. Service composability, kumpulan service dapat dikoordinasi dan dibentuk menjadi sebuah service gabungan
7. Service autonomy, service memegang control terhadap logika yang dia gunakan sendiri
8. Service optimization, high-quality service lebih diutamakan dibanding dengan low-quality service
9. Service discoverability, service dirancang sedemikian rupa sehingga service tersebut dapat ditemukan dan dinilai dengan mekanisme yang sudah ada
10. Service relevance, service dibuat sesuai dengan kebutuhan sehingga dikenali sebagai service yang memiliki nilai (meaningful service)

**8.2 Service Oriented Analysis and Design**

Dalam mendesain sebuah service yang berbasis SOA diperlukan tahapan yang mencakup proses analisis kebutuhan dan perancangan desain sistem yang menjadi dasar arsitektur suatu sistem, hal itu dapat diatasi dengan menggunakan SOAD (Service Oriented Analysis nad Design).

Perancangan desain secara top-down digunakan dalam SOAD, yaitu dengan menganalisa dan mengemlompokkan kebutuhan bisnis kemudian menerjemahkannya menjadi service, kemudian mendefinisikan komponen pendukung service untuk kemudian diimplementasikan pada sistem. Pola top down ini bisa digambarkan menjadi tiga bagian yaitu Conceptual View, Logical View dan Physical View.

**Gambar 8.1 Penggambaran SOAD secara top-down**

**8.3 Windows Workflow Foundation (WWF)**

WF merupaka sebuah teknologi dari Microsoft yang menyediakan sebuah API, *in-process workflow engine*, dan designer untu mengimplementasikan proses yang berkepanjangan sebagai sebuah workflow dalam aplikasi yang berbasis .NET. WF versi terbaru adalah versi 4 dan biasa disebut dengan WF4.

Sebuah *workflow* merupakan serangkaian langkah-langkah atau fase dalam pemrograman. Setiap fase dimodelkan dalam WF sebagai sebuah *Activity* dan framework .NET menyediakan sebuah *library* Activity yang mencakup keseluruhan Activity tersebut. Activity-activity tersebut dapat disusun secara visual dalam beberapa *workflow* menggunakan Workflow Designer, yaitu sistem desain yang terdapat dalam Microsoft Visual Studio, atau juga terdapat di aplikasi lainnya.

S*equential workflow* merupakan *workflow* yang bersifat “dapat diprediksi”, dimana alur eksekusi sebuah workflow dapat bercabang, berulang, ataupun menunggu sebuah *event* eksternal terjadi, namun *sequential workflow* akan menggunakan himpunan dari Activity maupun aturan lainnya yang telah tersedia untuk terus berjalan, sehingga *workflow* ini menjadi kontrol dari proses-proses tersebut.

Sebuah *state machine workflow* merupakan *workflow* yang digerakkan oleh *event-event* yang terjadi, yang berarti *workflow* tersebut bergantung kepada *event* eksternal hingga selesai. Caranya, pengguna mendefinisikan kondisi-kondisi (*state*) legal dari *workflow* tersebut, beserta transisi yang legal antar-*state* tersebut, dan pergerakan atau transisi tersebut digerakkan oleh *event* eksternal workflow tersebut. Kesimpulannya, *workflow* ini mendefinisikan sebuah struktur pemrograman yang dapat diikuti, namun pengendalian berada di luar kendalinya.

Pengguna menggunakan *sequential workflow* pada saat pengguna tersebut dapat melakukan pengkodean keputusan-keputusan aplikasi di dalam *workflow* tersebut, dan menggunakan *state machine workflow* apabila terjadi sebaliknya. Untuk selanjutnya, dalam penggunaan *workflow* tersebut dapat diatur *activity-activity* yang diinginkan pengguna dengan menggunakan *activity designer*, yaitu sebuah *base class* untuk mengatur *activity* dalam sebuah *workflow*.[3]

Selain itu dikenal juga suatu bentuk workflow yang mengkoordinasikan berbagai proses yang melibatkan manusia sebagai pengguna, yang disebut dengan human workflow. Workflow bentuk ini dapat mengotomasikan interaksi antarmanusia, bahkan antara manusia dengan sistem, sehingga memungkinkan peran besar manusia dalam sistem itu sendiri. *Workflow* tipe ini bergantung kepada *Activity* oleh pengguna (manusia) yang dihitung dalam suatu rentang waktu (harian, mingguan, bahkan bulanan), baik aktifitas terhadap suatu sistem atau aktifitas fisik diluar sistem, sehingga eksekusi *workflow* ini harus berhenti di suatu titik dan menunggu sebuah event eksternal yang biasanya berupa interaksi pengguna maupun event dalam sistem sendiri untuk bergerak te fase selanjutnya.[4]

**8.4 Windows Communication Foundation (WCF)**

WCF atau Windows Communication Foundation adalah salah satu teknologi baru dari Microsoft yang memungkinkan aplikasi dalam lingkungan terdistribusi berkomuniksi satu sama lain. WCF adalah model pemrograman lengkap untuk membangun aplikasi berorientasi layanan. Teknologi ini memungkinkan pengembang untuk membangun solusi aman, handal, dan mendukung transaksi, yang dapat terintegrasi lintas platform serta mampu beroperasi dengan investasi yang ada.

WCF merupakan cara termudah dalam membuat dan mengkonsumsi *service* pada platform microsoft maupun .NET. Dengan menggunakan WCF, pengembang dapat fokus pada aplikasi mereka daripada fokus ke protokol komunikasi, karena WCF mengenkapsulasi terhadap teknologi dan pekerjaan teknis, sehingga pengembang bisa lebih produktif. WCF sendiri mendukung berbagai teknologi seperti Web service, Binary .NET to .NET communication, Distributed Transaction, Queued Messaging, RESTful Communication, serta mendukung WS-\* Spesification.

Karena WCF mendukung komunikasi melalui Web Service maka WCF akan dapat saling beroperasi dengan aplikasi platform lain yang mendukung SOAP. Selain itu WCF juga mendukung spesifikasi WS-\* sehingga menyediakan kehandalan, keamanan, dan transaksi ketika berkomunikasi dengan platform lain yang juga mendukung spesifikasi WS-\*. WCF tidak mengharuskan untuk menggunakan protokol HTTP, tetapi juga bisa menggunakan TCP. Pertukaran message pun bisa menggunakan selain SOAP, tetapi bisa juga menggunakan XML sederhana ataupun JSON.

**8.5 Nhibernate**

NHibernate adalah salah satu dari mapper database object-relational yang *open source*, Dengan NHibernate kita dapat dengan mudah mendapat sebuah object standar .NET dari hasil eksekusi database. NHibernate tidak hanya memfokuskan kepada pemetaan dari kelas .NET ke tabel database (dan dari tipedata .NET ke tipe data SQL), tetapi juga menyediakan data query dan mendapatkan fasilitas-fasilitas yang dapat mengurangi waktu pengembangan perangkat lunak secarasignifikan. Tujuan dari NHibernate adalah adalah meringankan beban developer terhadap persoalan-persoalan programming yang berkaitan dengan data.

**8.6 Kontribusi Tugas Akhir**

Tugas akhir ini merupakan penyempurnaan riset dari tugas akhir sebelumnya, dimana mengimplementasikan pemodelan arsitektur Service Oriented Analysis and Design (SOAD) untuk merancang portal Enterprise Resource Planning. Kelebihan yang didapat dari penyempurnaan riset adalah adanya pemilihan paket functional domain untuk berbagai macam jenis usaha yang menyesuaikan kondisi dari perusahaan. Kompleksitas runtime dan proses pengembangan dari sistem lebih cepat dari sebelumnya. Sistem memiliki flexibilitas lebih tinggi karena menggunakan teknologi terbaru workflow.

1. **RINGKASAN TUGAS AKHIR**

**9.1 Gambaran Integrasi**

Perusahaan manufaktur yang berskala enterprise tentunya memiliki departemen-departemen (functional domain) yang berdiri sendiri karena memiliki kebijakan masing-masing dalam menjalankan proses bisnisnya, seperti yang ditunjukkan dalam gambar 9.1.

**Supplier Relationship Management**

**Enterprise Resource Planning**

Inventory

Manufacturing

General Ledger

Account Receivable

Account Payable

Fixed Assets

**Business Intelligence**

**Customer Relationship Management**

**Gambar 9.1 Aplikasi dalam perusahaan manufakturing**

Penyediaan service pada layanan bisnis oleh functional domain menggunakan web service sehingga memenuhi prinsip SOA yang loose coupling. Web service inilah yang nantinya akan dikonsumsi oleh functional domain lain.

Secara umum ada tiga proses bisnis dalam perusahaan yaitu procure to pay, order to cash dan treasury/perbendaharaan yang membutuhkan integrasi antar modul.

**9.2 Arsitektur Aplikasi**

Dalam segi arsitektur, perangkat lunak yang akan diintegrasikan memiliki lima buah layer yaitu :

1. Web Service Layer : Layer ini bertanggung jawab dalam hal implementasi web service yang disediakan.
2. Presentation Layer : Layer ini bertanggung jawab dalam hal user interface dengan pengguna.
3. Application Service Layer : Seluruh control logic yang ada di dalam perangkat lunak ini dibangun di dalam layer ini.
4. Domain Model Layer : Domain model layer berisi implementasi class-class yang digunakan dan data transfer object yang nantinya digunakan oleh web service dalam proses tranfer data.
5. Data Access Layer : Layer ini bertanggung jawab dalam hal koneksi dengan database yang digunakan. Dalam pengerjaan tugas akhir ini menggunakan framework Hibernate.

Secara garis besar, gambaran hubungan antara layer-layer yang telah disebutkan di atas adalah sebagai berikut :

Presentation Layer

Web Service

Application Service Layer

Data Acess Layer

HIbernate

Domain

Model Layer

Database

**Gambar 9.2 Arsiektur Aplikasi yang diintegrasikan**

Sedangkan untuk integrasi perangkat lunak ERP memiliki arsitektur sebagai berikut :

* + - 1. **Presentation :** Bagian ini merupakan antar muka bagi user untuk berinteraksi dengan sistem ERP yang telah terintegrasi.
      2. **Orchestrtion dan Coreography :** Bagian ini berperan untuk integrasi service dengan melakukan orkestrasi dan koreografi service-service dari functional domain.
      3. **Registered Web Service :** Bagian ini merupakan web service dari functional domain yang telah didaftarkan pada enterprise service bus.
      4. **Functional Domain :** Bagian ini merupaka aplikasi-aplikasi berorientasi servis yang akan diintegrasikan.

Presentation

Orchestration & Coreography

Web Service

Functional Domain

**Gambar 9.3 Arsitektur Integrasi Enterprise Resource Planning**

1. **METODOLOGI TUGAS AKHIR**

Pengerjaan tugas akhir ini dibagi dalam beberapa tahap yaitu :

1. Penyusunan Proposal Tugas Akhir

Tahap awal untuk memulai pengerjaan tugas akhir adalah penyusunan proposal tugas akhir. Pada proposal ini, penulis mengajukan gagasan perancangan dan pengembangan aplikasi perangkat lunak *tugas akhir* dengan menggunakan SOAD pada platform .NET.

1. Studi Literatur

Pada tahapan ini akan dilakukan studi literatur mengenai objek kasus dan metode yang digunakan, diantaranya :

* + - * 1. Pemahaman mengenai proses bisnis *tugas akhir*.
        2. Perancangan menggunakan SOAD
        3. Pemahaman mengenai framework aplikasi web dan web service pada platform .NET.

1. Implementasi

Implementasi merupakan tahap untuk merancang dan membangun aplikasi perangkat lunak *tugas akhir*.

1. Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap aplikasi yang telah dibuat.

Langkah-langkah uji coba yang akan dilakukan antara lain :

1. Menguji apakah aplikasi yang dibuat sesuai dengan service pada level bisnis yang telah ditentukan saat proses desain.
2. Menguji apakah aplikasi yang dibuat setelah diintegrasikan dapat menghasilkan data yang valid.
3. Penyusunan Buku Tugas Akhir

Tahap terakhir merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perancangan dan prototyping yang telah dibuat. Secara garis besar, buku laporan tugas akhir ini terdiri atas beberapa bagian yaitu :

1. Pendahuluan
   1. Latar Belakang
   2. Permasalahan
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Uji Coba dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka
7. **JADWAL KEGIATAN TUGAS AKHIR**

Tugas akhir ini diharapkan bisa dikerjakan menurut jadwal sebagai berikut:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Kegiatan | 2012 | 2013 | | | | | |
| Des | Jan | Feb | Mar | Apr | Mei | Jun |
| 1. | Penyusunan Proposal Tugas Akhir |  |  |  |  |  |  |  |
| 2. | Studi Literatur |  |  |  |  |  |  |  |
| 3. | Perancangan Sistem |  |  |  |  |  |  |  |
| 4. | Implementasi |  |  |  |  |  |  |  |
| 5. | Pengujian dan Evaluasi |  |  |  |  |  |  |  |
| 6. | Penyusunan Buku Tugas Akhir |  |  |  |  |  |  |  |

1. **DAFTAR PUSTAKA**
2. Salter, David and Jennings, Fark. 2008. “Building SOA-Based Composite Applications Using Netbeans IDE 6”. Birmingham. Packt Publishing

[2] Jørstad, Ivar et. al. 2005. A Service Oriented Architecture Framework for Collaborative Services. Institute of Electrical and Electronics Engineers.

[3] Chappel, David. 2009. The Workflow Way: Understanding Windows Workflow Foundation. Microsoft.

[4] Shukla, Dharma and Bob Schmidt. 2006. Essential Windows Workflow Foundation. Addison Wesley Professional.

**LEMBAR PENGESAHAN**

###### **Surabaya, Desember 2012**

Menyetujui,

Pembimbing I

Prof. Drs.Ec.Ir. Riyanarto Sarno,M.Sc,Ph.D

NIP : 195908031986011001

Pembimbing II

Dwi Sunaryono, S.Kom, M.Kom

NIP. 197205281997021001