**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

Nama : **Ahmad Dzulfikar Adi Putra**

NRP : **5107 100 088**

Dosen Wali :  **Sarwosri, S.Kom,MT**

1. **JUDUL TUGAS AKHIR**

***Rancang Bangun Koreografi dan Orkestrasi Web Service***

***pada Multisite 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. Eksekusi proses bisnisnya dilakukan melalui web service dengan menggunakan BPEL. Sedangkan web service yang digunakan untuk memproses proses bisnis dapat dilakukan dengan dua cara yaitu orkestrasi (orchestration) dan koreografi (choreography).

Implementasi ERP dapat dilakukan dengan cukup mudah apabila struktur perusahaan simpel dan beroperasi di satu lokasi. Namun ketika perusahaan mempunyai struktur yang kompleks dan tersebar di beberapa lokasi, 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. Sehingga untuk pengimplementasian ERP multi lokasi akan berjalan dengan lebih mudah.

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. Pada akhirnya SOA akan membantu organisasi dalam melakukan proses bisnis secara lebih efisien dan beradaptasi dengan perubahan lingkungan.

ESB (Enterprise Service Bus) merupakan infrastruktur yang menyediakan penerapan konsep SOA (Service Oriented Architecture). Pada ESB komunikasi antara provider dan client tidak dilakukan secara n-to-n melainkan melalui service bus. Hal ini memberikan keuntungan pada klien karena tidak akan terpengaruh oleh perubahan letak dari provider servis, karena yang diketahui adalah servis yang ada pada service bus. Dengan menggunakan ESB servis-servis yang ada pada proses bisnis perusahaan dapat diintegrasikan sehingga dengan mudah dikoreografi dan diorkestrasikan menjadi servis baru yang sesuai dengan proses bisnis.

BPEL merupakan bahasa yang mendeskripsikan proses bisnis melalui web service. BPEL menggunakan XML untuk mendefinisikan komposisi fungsi-fungsi web service yang akan dieksekusi. Pemrosesan proses bisnis menggunakan web service dilakukan melalui orkestrasi dan koreografi.

Pada orkestrasi, terdapat sebuah proses terpusat yag mengatur semua web service yang terlibat. Web service tidak perlu tahu bahwa sebenarnya ia terlibat dalam bagian dari proses bisnis yang lebih tinggi tingkatannya karena hanya proses terpusat yang mengetahui hal ini. Sedangkan pada koreografi, proses tidak dijalankan secara terpusat. Setiap web service dalam koreografi harus mengetahui kapan mengeksekusi operasi dan dengan siapa web service tersebut akan berinteraksi. Susunan koreografi mempunyai fokus pada pertukaran pesan. Semua web service pada susunan ini harus tahu proses bisnis apa yang dijalankan, operasi apa yang dieksekusi, pesan apa yang akan dipertukarkan, dan kapan waktunya web service harus berjalan.

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 pada multi lokasi dengan menggunakan Enterprise Service Bus.”

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. Framework yang digunakan untuk mengintegrasikan servis yaitu Open ESB versi 2.2.
      3. Framework yang digunakan untuk User Interface menggunakan JSF dan server yang digunakan adalah Glassfish versi 2.1.

1. **TUJUAN DAN MANFAAT TUGAS AKHIR**

Tugas akhir ini bertujuan untuk mengintegrasikan servis-servis dari berbagai aaplikasi yang terpisah ke dalam Enterprise Service Bus untuk kemudian diorkestrasikan menjadi aplikasi ERP yang terintegrasi.

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. **RINGKASAN TUGAS AKHIR**

**7.1 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 7.1.

**Gambar 7.1 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 setelah sebelumnya alur proses bisnisnya diintegrasikan melalui service bus.

Secara umum ada tiga proses bisnis dalam perusahaan yaitu procure to pay, order to cash dan treasury/perbendaharaan yang membutuhkan integrasi antar modul. Komunikasi antar modul menggunakan web service. Tabel 7.1 menunjukkan daftar web service dalam beberapa proses bisnis tersebut.

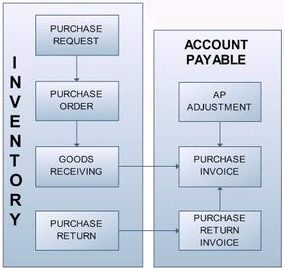
|  |  |  |  |
| --- | --- | --- | --- |
| **Process** | **Functional Domain** | **Business Process** | **Web Services** |
| Procure to pay | Inventory | Purchase Request | ProvidingPurchaseRequestIntegration |
| Purchase Order | ProvidingPurchaseOrderIntegration |
| Good Receiving | ProvidingReceivingIntegration |
| Purchase Return | ProvidingPurchaseReturnIntegration |
| Account Payable | AP Adjusment | APAdjustmentService |
| AP Down Payment | APDownPaymentService |
| Ap Invoice | APInvoiceService |
| AP Return Invoice | APReturnInvoiceService |
| Cash Bank | Cash Transaction | CashTransactionManagement |
| Bank Transaction | BankTransactionManagement |
| Cheque Transaction | ChequeTransactionManagement |
| Giro Transaction | GiroTransactionManagement |
| Transaction Code Management | TransactionCodeManagement |
| Currency Management | CurrencyManagement |
| Cash Management | CashManagement |
| Bank Management | BankManagement |
| Cheque Management | ChequeManagement |
| Giro Management | GiroManagement |
| Order to Cash | Inventory | Providing Sales Order | ProvidingSalesOrder |
| Providing Revenue | ProvidingRevenue |
| Providing Item | ProvidingItem |
| Account Receivable | AR Adjusment | ARAdjustmentService |
| AR Down Payment | ARDownPaymentService |
| AR Invoice | ARInvoiceService |
| AR Return Invoice | ARReturnInvoiceService |
| Cash Bank | --Business process sama dengan pada procure to pay-- | --Web service sama dengan pada procure to pay-- |
| Treasury | General Ledger | Generate Financial Report | FinancialReport |
| Account Management | Account Management |
| Period Management | AccountingPeriodManagement |

**Tabel 7.1 Web service yang digunakan dalam proses bisnis perusahaan manufaktur**

**7.2 Aplikasi dalam Perusahaan Enterprise**

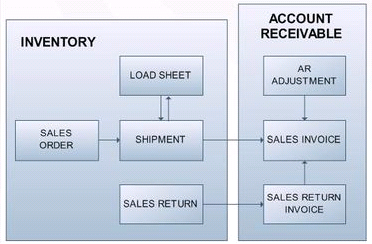
Berikut ini merupakan modul-modul aplikasi yang terdapat dalam perusahaan skala enterprise yang di gunakan oleh depatemen yang berbeda-beda.

**7.2.1 Inventory & Distribution**

Modul yang digunakan untuk pengelolaan aktifitas penjualan, pembelian, pengontrolan stok barang dan distribusi barang.

**Gambar 7.2 Integrasi Inventory dengan Account Payable**

Inventory Planning dilakukan bertujuan untuk melakukan perencanaan mengenai persediaan dan pembelian barang guna memenuhi kebutuhan perusahaan. Kemudian berdasarkan hal tersebut, dibuat permintaan pembelian (Purchase Request) terhadap barang-barang yang dibutuhkan untuk dibeli. Sesuai dengan permintaan yang ada, dibuatkan order/pesanan pembelian (Purchase Order) ke supplier tertentu. Kemudian barang-barang yang dipesan tersebut diterima (Receiving), dan dari penerimaan barang ini dibuatkan faktur pembelian (Purchase Invoice) terhadap supplier tersebut.

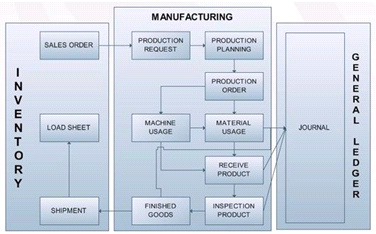
Jika terdapat retur dari pembelian barang, maka dibuatkan Purchase Return. Retur ini kemudian difakturkan (Purchase Return Invoice) dan mengurangi hutang.

**Gambar 7.3 Integrasi Inventory dengan Account Receivable**

Alur transaksi AR dimulai dari order/pesanan oleh customer (Sales Order). Berdasarkan order dari customer dilakukan pengiriman ke alamat customer (Shipment). Dari pengiriman ini, dibuatkan faktur penjualan (Sales Invoice) terhadap customer tersebut.

Jika terdapat retur dari penjualan barang, maka dibuatkan Sales Return. Retur ini kemudian difakturkan dan mengurangi piutang.

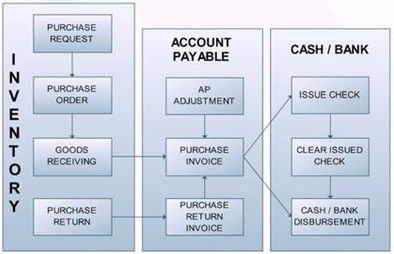
**7.2.2 Manufacturing**

Modul yang digunakan untuk pengelolaan proses produksi mulai dari penjadwalan produksi oleh bagian PPIC dan realisasi oleh bagian Produksi.

**Gambar 7.4 Integrasi Manufacture dengan Inventory dan General Ledger**

Gambar 7.4 menunjukkan alur proses manufacturing *make-to-order* dimulai dari sales order customer yang dibuat oleh bagian marketing, kemudian dilanjutkan oleh bagian PPIC untuk membuatkan perencanaan produksinya dan di kerjakan oleh bagian produksi. Hasil dari produk jadi yang sudah lolos dari pemeriksaan standar mutu akan di pindahkan kelokasi barang jadi yang kemudian akan di kirimkan ke customer oleh bagian distribusi yang menggunakan transaksi shipment dan load sheet pada modul inventory/distribution.

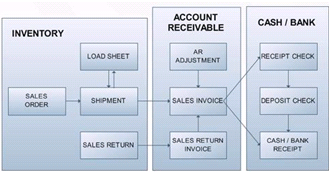
**7.2.3 Accounting & Finance**

Modul yang digunakan untuk mengelola aktifitas yang berkaitan dengan masalah akuntansi dan keuangan.

**Gambar 7.5 Integrasi Account Payable dengan Inventory dan Cash Bank**

Inventory Planning dilakukan bertujuan untuk melakukan perencanaan mengenai persediaan dan pembelian barang guna memenuhi kebutuhan perusahaan. Kemudian berdasarkan hal tersebut, dibuat permintaan pembelian (Purchase Request) terhadap barang-barang yang dibutuhkan untuk dibeli. Sesuai dengan permintaan yang ada, dibuatkan order/pesanan pembelian (Purchase Order) ke supplier tertentu. Kemudian barang-barang yang dipesan tersebut diterima (Receiving), dan dari penerimaan barang ini dibuatkan faktur pembelian (Purchase Invoice) terhadap supplier tersebut.

Dari pembelian barang ini akan menambah hutang terhadap supplier tersebut. Pembayaran hutang ke supplier bisa dilakukan melalui modul Cash/Bank Management yaitu dengan cash disbursement, bank disbursement maupun issue check. Jika terdapat retur dari pembelian barang, maka dibuatkan Purchase Return. Retur ini kemudian difakturkan (Purchase Return Invoice) dan mengurangi hutang.

  
**Gambar 7.6 Integrasi Account Receivable dengan Inventory dan Cash Bank**

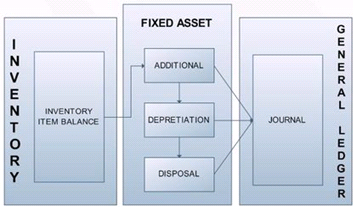
Pada gambar 7.6 , alur transaksi Account Receivable dimulai dari order/pesanan oleh customer (Sales Order). Berdasarkan order dari customer dilakukan pengiriman ke alamat customer (Shipment). Dari pengiriman ini, dibuatkan faktur penjualan (Sales Invoice) terhadap customer tersebut.

Penjualan ini akan menambah piutang customer tersebut. Pembayaran piutang oleh customer bisa dilakukan melalui modul Cash/Bank Management yaitu dengan cash receipt, bank receipt maupun check receipt (penerimaan cek).

Jika terdapat retur dari penjualan barang, maka dibuatkan Sales Return. Retur ini kemudian difakturkan dan mengurangi piutang

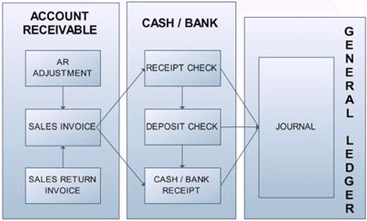
**7.2.4 Fixed Asset Management**

Sub modul yang digunakan untuk mengelola aktiva tetap.

  
**Gambar 7.7 Integrasi Fixes Assets dengan Inventory dan General Ledger**

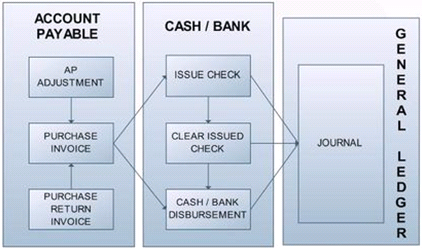
**7.2.5 Cash & Bank Management**

Sub modul yang digunakan untuk mengelola transaksi uang tunai, bank, uang muka, cek, bank rekonsiliasi dan melakukan pemantauan arus kas.



**Gambar 7.8 Integrasi Cach Bank dengan Account Receivable dan General Ledger**

Piutang perusahaan dari Sales Invoice, Sales Return, atau Account Receivable Adjustment dapat dibayar melalui Cash & Bank, yaitu dengan Cash Receipt, Bank Receipt maupun Receive Check. Kemudian pembayaran tersebut akan diposting menjadi jurnal-jurnal di modul General Ledger.

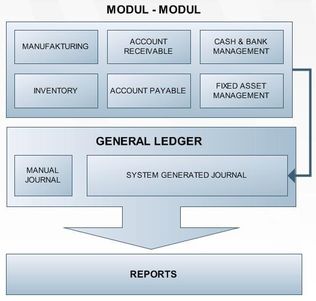


**Gambar 7.9 Integrasi Cash Bank dengan Account Payable dan General Ledger**

Hutang perusahaan dari Purchase Invoice, Purchase Return, atau Account Payable Adjustment dapat dibayar melalui modul Cash & Bank Management, yaitu dengan Cash Disbursement, Bank Disbursement maupun Issue Check. Kemudian pembayaran tersebut akan diposting menjadi jurnal-jurnal di modul General Ledger.

**7.2.6 General Ledger**

Merupakan ujung dari semua transaksi dalam proses bisnis perusahaan. General Ledger ini digunakan untuk melacak semua transaksi-transaksi keuangan, memonitor saldo rekening dan menghasilkan laporan keuangan.

****

**Gambar 7.10 Integrasi General Ledger dengan modul-modul lainnya**

Transaksi-transaksi pada General Ledger dapat dibedakan menjadi 2 berdasarkan asal datanya, yaitu: dari hasil posting modul lainnya (System Generated Journal) dan dari pengisian secara manual oleh user (General Journal).

Pada dasarnya modul General Ledger dapat berdiri sendiri, tanpa modul-modul lainnya. Jika berdiri sendiri, maka transaksi-transaksinya murni berasal dari hasil isian user. Transaksi-transaksi tersebut, khususnya jurnal-jurnal umum, akan di-post untuk membentuk buku besar. Begitu pula jika modul General Ledger digunakan bersama dengan modul-modul lainnya. Semua transaksi dari modul lain: Inventory, Account Payable, Account Receivable, Cash & Bank Management, dan Fixed Asset, pada akhirnya harus di-posting untuk membentuk jurnal di GL. Jurnal-jurnal tersebut akan dikelompokkan dalam System Generated Journal dan akan di-post ke buku besar.

Buku besar akan diolah untuk menghasilkan laporan-laporan keuangan, seperti: Balance Sheet atau Neraca, Income Statement atau Laporan Laba/Rugi

**7.3 Skenario Implementasi**

Implementasi ERP dapat dilakukan dengan cukup mudah apabila struktur perusahaan simpel dan beroperasi di satu lokasi. Namun ketika perusahaan mempunyai struktur yang kompleks dan tersebar di beberapa lokasi, implementasi ERP akan lebih kompleks, sulit ,unik, dan akan menghadapi tantangan di tingkat teknikal dan manajerial. Perusahaan biasanya melakukan perencanaan pada level strategic/manajerial sebelum mengimplementasikannya dalam level teknikal (software dan hardware).

Yang perlu diperhatikan dalam pengimplementasian ERP multi lokasi adalah scope-nya (Markus, Lynne. 2000. Multisite ERP Implmentation), karena scope ini akan berpengaruh pada :

1. Luasnya jangkauan modul dari ERP yang diimplementasikan.
2. Dampak yang ditimbulkan ERP dalam perubahan struktur organisasional perusahaan, koordinasi kerja, dan integrasi proses dari proses bisnis dalam perusahaan.
3. Tingkat kesulitan dalam level manajerial

Dalam implementasi ERP multilokasi setidaknya ada beberapa tipe hubungan unit bisnis dalam perusahaan yaitu :

1. Otonomi total pada tiap unit bisnis.
2. Peran minimal Headquarter (HQ) dalam mengkontrol proses lokal.
3. HQ berperan dalam kordinasi transaksi antar unit bisnis.
4. Kordinasi antar unit bisnis langsung melalui network
5. Sentralisasi total, HQ mengontrol tiap keputusan pada tiap unit bisnisnya.

Dari kelima tipe tersebut, pada tugas akhir ini akan menggunakan tiga skenario dalam pengimplementasiannya nanti. Yaitu single location, HQ coordination dan Network Coordination.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Single Location | | HQ Coordination | | Network Coordination | |
|  | HQ | Branch | HQ | Branch | HQ | Branch |
| Business Intelligence | V |  |  | V |  | V |
| CRM | V |  | V |  |  | V |
| General Ledger | V |  |  | V |  | V |
| Career Planning | V |  | V |  |  | V |
| Transportation | V |  |  | V |  | V |
| Manufacturing | V |  |  | V |  | V |
| HRM | V |  | V |  |  | V |
| Inventory | V |  |  | V |  | V |
| Sales | V |  | V |  |  | V |
| Account Payable | V |  |  | V |  | V |
| Account Receivable | V |  |  | V |  | V |
| Fixed Assets | V |  |  | V |  | V |

**Tabel 7.1 Rencana Skenario Impelementasi ERP multi lokasi**

Beberapa task yang diprediksi muncul dalam pengimplementasian HQ coordination :

* 1. Approval untuk pembelian dengan jumlah nominal tertentu. Hal ini tentunya berkaitan dengan kebijakan masing-masing perusahaan.
  2. Kordinasi pembelian material produksi. Karena dengan pembelian lebih banyak akkan didapat harga yang lebih murah.
  3. Perpindahan karyawan ke branch lain.
  4. Order yang dilakukan oleh customer secara terpusat.
  5. Recruitment karyawan secara terpusat.

Beberapa task yang diprediksi muncul dalam pengimplementasian network coordination :

1. Kordinasi pembelian material produksi. Karena dengan pembelian lebih banyak akkan didapat harga yang lebih murah.
2. Perpindahan karyawan ke branch lain.
3. Koordinasi antar branch ketika ada order dari customer namun branch tidak mampu memenuhi order tersebut dan membeli beberapa stok dari branch lain.
4. Order yang dilakukan oleh customer dapat dilakukan secara langsung pada branch.
5. Branch memilik kewnwngan sendiri untuk melakukan recruitmen karyawan.

**7.4 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. Dalam pengerjaan tugas akhir ini menggunakan framework Java Server Face (JSF).
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 Java Server Face (JSF).

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

**Gambar 7.11 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.

**Gambar 7.12 Arsitektur Integrasi Enterprise Resource Planning**

1. **DASAR TEORI**

**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

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)

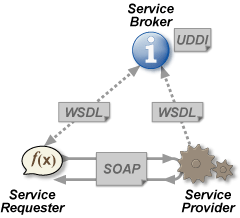
**8.2 Service Oriented Analysis and Design**

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 Web Services**

****

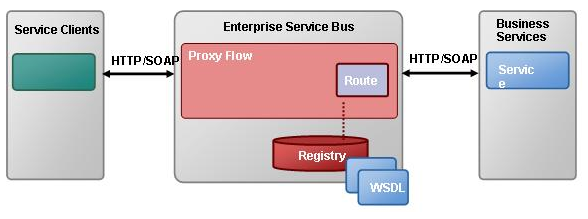
**Gambar 8.2 Komunikasi menggunakan web service**

Ada berbagai versi definisi mengenai Web Service, yang pada intinya menggambarkan

karakteristik dari Web Service, yaitu antara lain sebagai berikut :

* Merupakan application logic yang dapat diakses dan dipublikasikan menggunakan standard Internet (TCP/IP, HTTP, SMTP, FTP, JMS, Web).
* Dideskripsikan dalam format XML.
* Didentifikasikan dengan Universal Resources Identifier (URI).
* Bersifat Loosely coupled, self-contained, modular dan terbuka (nonproprietary).
* Digunakan untuk mendukung interoperabilitas interaksi machine-to-machine melalui jaringan Internet/Intranet.

**8.4 Enterprise Service Bus**

****

**Gambar 8.3 Komunikasi data pada Enterprise Service Bus**

Enterprise Service Bus (ESB) merupakan infrastruktur software yang memfasilitasi integrasi antar aplikasi. ESB berguna dalam implementasi service-oriented-architecture (SOA) karena pertukaran message yang banyak terjadi dalam SOA, mengeksekusi transaksi, orkestrasi service, dan melakukan publikasi dan persetujuan fungsi antar aplikasi yang terpisah dan terdistribusi.

ESB cocok untuk SOA, hal ini karena ESB memfasilitasi loose-coupling dari sistem dan menggunakan open-standard yang merupakan dua fitur dalam keberhasilan implementasi SOA. ESB dapat diimplementasikan dengan cepat dan berbiaya rendah, sehingga hal ini lebih disukai oleh developer yang bekerja dibawah deadline yang ketat dan berbiaya rendah. Developer yang menggunakan ESB mempunyai visi jangka panjang dan menyukai fungsionalitas ESB yang komprehensif untuk mensupport strategi SOA secara menyeluruh.

**8.5 Business Process Execution Language (BPEL)**

BPEL adalah bahasa yang berbasis pada workflow yang menggabungkan service yang berperan dalam suatu interaksi yang dapat berupa orkestrasi dan koreografi service. BPEL memanfaatkan deskripsi WSDL untuk mendefinisikan fungsi yang disediakan oleh service yang ada dan model interaksi yang dibentuk. XPath merupakan standar yang dapat digunakan untuk mengakses dan memanipulasi data yang digunakan dalam koreografi BPEL. Untuk manajemen endpoint dari service yang disusun dapat digunakan standar WS-Addressing.

Service yang berpartisipasi dalam proses disebut dengan nama *partner service*. Ketika suatu proses BPEL dibangkitkan, *instance*-nya akan tetap ada selama eksekusi belum selesai. Ada dua cara pembentukan proses yaitu *partner service* membangkitkan *instance* yang baru dari proses atau *partner service* berinteraksi dengan *instance* yang telah ada.

1. **PERENCANAAN 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 java.

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 java.

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 | 2010 | | | | | 2011 | |
| Sep | Okt | Nov | Des | Jan | | Feb |
| 1. | Penyusunan Proposal Tugas Akhir |  |  |  |  |  | |  |
| 2. | Studi Literatur |  |  |  |  |  | |  |
| 3. | Implementasi |  |  |  |  |  | |  |
| 4. | Pengujian dan Evaluasi |  |  |  |  |  | |  |
| 5. | Penyusunan Buku Tugas Akhir |  |  |  |  |  | |  |

1. **DAFTAR PUSTAKA**
2. Enterprise Resource Planning, <url: <http://en.wikipedia.org/wiki/Enterprise_resource_planning>>, diakses tanggal 20 September 2010.
3. Lynne Markus, Cornelis Tanis, and Van Venema, Paul C. (April 2000), “Multisite ERP Implementation” ; Communication of the Association of Computing Machinery 0002-0782/00/0400
4. Sinulingga, Sukaria. 2009. “Perencanaan dan Pengendalian Produksi”. Yogyakarta. Penerbit Graha Ilmu
5. Salter, David and Jennings, Fark. 2008. “Building SOA-Based Composite Applications Using Netbeans IDE 6”. Birmingham. Packt Publishing
6. Arinta P, Riska. 2010. Service Oriented Analysis and Design (SOAD) untuk Perangkat Lunak Cash Bank dan General Ledger dan Pembuatan Prototypenya. Surabaya. Tugas Akhir Teknik Informatika ITS

**LEMBAR PENGESAHAN**

###### **Surabaya, 1 November 2010**

Menyetujui,

Pembimbing I

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

NIP : 195908031986011001

Pembimbing II

Dwi Sunaryono S.Kom

NIP : 197205281997021001