

# Implementasi Modul-Modul Enterprise Resource Planning Multi Tenant pada Cloud Computing

Rizky W. Herlambang, Riyanarto Sarno, dan Dwi Sunaryono

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

*e-mail:* riyanarto@if.its.ac.id

**Abstrak**—Tantangan ketika sebuah perusahaan akan berpindah menggunakan sistem ERP adalah adanya ketakutan untuk menyediakan infrastruktur beserta kebutuhan lainnya. Dengan menggunakan *cloud computing*, perusahaan tidak perlu untuk menyediakan infrastruktur, konfigurasi perangkat lunak dan pemeliharannya, akan tetapi hanya tinggal menggunakannya langsung sebagai suatu layanan yang dapat diakses melalui berbagai media seperti komputer *desktop*, *laptop*, *mobile device* selama terkoneksi dalam jaringan internet. Sistem ERP yang dibangun menerapkan prinsip *multi tenant* pada *cloud computing*. Implementasi prinsip *multi tenant* dilakukan dengan pendekatan basis data terpisah (*separated database*). Dengan prinsip ini, satu perangkat lunak, akan mengakses banyak sekali basis data, sehingga data antara satu perusahaan, dengan perusahaan lain, tidak akan saling tercampur dan lebih memiliki tingkat integritas yang tinggi. Selain itu, perangkat lunak yang dibangun menyediakan beberapa konfigurasi yang dapat digunakan apabila suatu perusahaan ternyata memiliki proses bisnis yang berbeda. Dengan penerapan sistem ini, kekhawatiran perusahaan mengenai infrastruktur dan pemeliharaan sistem dapat diatasi sehingga tujuan penerapan ERP sebagai langkah mempermudah proses bisnis perusahaan dapat tercapai.

**Kata Kunci**—*cloud computing*, ERP, *multi tenant*, *separated database*.

## I. PENDAHULUAN

**E**NTERPRISE Resource Planning (ERP), adalah sistem informasi yang digunakan untuk mengintegrasikan dan mengotomasikan proses bisnis perusahaan, dari proses pengadaan barang, produksi, hingga proses penjualan. Dewasa ini, banyak perusahaan-perusahaan yang tertarik dan berpindah menggunakan sistem ERP ini.

Pengelolaan sistem jaringan, ketersediaan layanan dan konfigurasi sistem menjadi hal-hal utama yang ditakutkan ketika akan berpindah menggunakan sistem ERP. Selain itu kebanyakan perusahaan menengah tidak memiliki divisi atau bagian tertentu untuk mengelola dan memelihara infrastruktur IT sehingga mengharuskan untuk mencari pegawai lain atau dengan melakukan penyewaan pegawai yang tentunya akan memakan biaya dan mengubah proses bisnis perusahaan.

Saat ini, konsep *cloud computing* sedang marak diperbincangkan. Dengan menggunakan *cloud computing*, perusahaan tidak perlu untuk menyediakan infrastruktur, konfigurasi perangkat lunak dan pemeliharannya, akan tetapi

hanya tinggal menggunakannya langsung sebagai suatu layanan.

*Multi tenant* menjadi konsep wajib dalam *cloud computing*. *Multi tenant* yang digunakan dalam ERP memiliki konsep pemisahan basis data (*separated database*). Tiap-tiap perusahaan yang menggunakan ERP akan memiliki *instance* basis data tersendiri sehingga keamanan dan integrasi data tetap terjamin.

Dalam ERP, terdapat banyak sekali modul-modul yang dapat digunakan, tetapi tidak semua perusahaan butuh modul-modul tersebut. Dengan prinsip SaaS (*Software as a Service*), ERP dapat dikonfigurasi sesuai dengan keinginan dari perusahaan tersebut. Semisal, jika perusahaan hanya membutuhkan modul HRM (*Human Resource Management*) saja, maka perusahaan hanya perlu menggunakan modul tersebut saja. Hal ini tentu akan menghemat pengeluaran perusahaan daripada harus menggunakan semua paket yang tersedia.

## II. KAJIAN PUSTAKA

### A. Enterprise Resource Planning 2011 Teknik Informatika

Sistem ERP 2011 merupakan sistem terintegrasi yang dikembangkan di dalam lingkungan Teknik Informatika ITS. Sistem ini terdiri dari beberapa bagian yang disebut sebagai domain fungsi, diantaranya:

- *Inventory*  
*Inventory* atau sistem persediaan merupakan sistem yang mampu menangani proses pencatatan stok barang, baik itu pengeluaran barang maupun pemasukan barang dan juga pendistribusian barang hingga ke konsumen [1].
- *Manufacturing*  
*Manufacturing* adalah suatu sistem yang menangani proses produksi untuk menghasilkan produk-produk fisik (*tangible product*) [2].
- *Fixed Assets*  
Aktiva tetap (*Fixed Asset*) merupakan istilah yang digunakan untuk menyatakan sebuah aset atau properti perusahaan yang tidak dapat dengan mudah dikonversikan menjadi uang tunai. *Fixed assets* merupakan domain fungsi yang menangani hal tersebut [3].

- **Human Resources Management**  
*Human Resource Management* adalah suatu sistem manajemen manusia dalam sebuah perusahaan. Sistem ini bertujuan untuk mengatur pengelolaan sumber daya manusia secara efektif dan efisien [4].
- **Transportation**  
 Domain fungsi *transportation* memiliki fokus pada proses bisnis pemaketan barang kepada pembeli, sampai dengan pelacakan pengiriman barang [5].
- **Sales**  
 Domain fungsi *sales* menangani pencatatan *order* pelanggan (*sales order*), pengembalian barang dari pelanggan (*sales return*), pemberian harga jual dan diskon, serta hal-hal lain yang berkaitan dengan penjualan [6].
- **General Ledger**  
 Buku besar (*ledger*) adalah buku akun keuangan, yang mencerminkan pengaruh keuangan dari transaksi setelah dibukukan dari berbagai jurnal [7]. Domain fungsi ini menangani hal tersebut.
- **Career Planning**  
 Perencanaan karir atau *career planning* merupakan sebuah domain fungsi yang bertugas mengatur proses promosi dan evaluasi karyawan pada sebuah perusahaan [8].
- **Account Payable**  
*Account payable* (hutang dagang) adalah sebuah berkas atau akun pada *sub-ledger* yang mencatat jumlah yang harus dibayarkan oleh seseorang atau perusahaan kepada pemasok tetapi belum terbayarkan (masih merupakan suatu bentuk hutang) [3].
- **Account Receivable**  
*Account receivable* (piutang dagang) adalah jumlah uang yang harus dibayarkan oleh pelanggan kepada sebuah perusahaan karena membeli barang atau jasa secara kredit yang nantinya akan ditampilkan sebagai aktiva [3].
- **Cash Bank**  
 Domain fungsi *cash bank* secara umum menangani pencatatan transaksi penerimaan dan pengeluaran uang [6].

Keseluruhan domain fungsi terintegrasi melalui beberapa proses bisnis, yaitu *procure to pay* dan *order to cash*. *Procure to pay* merupakan proses pengadaan bahan baku dari *supplier* hingga proses pembayaran untuk memenuhi kebutuhan produksi. Alur proses ini ditunjukkan pada Gambar 1.

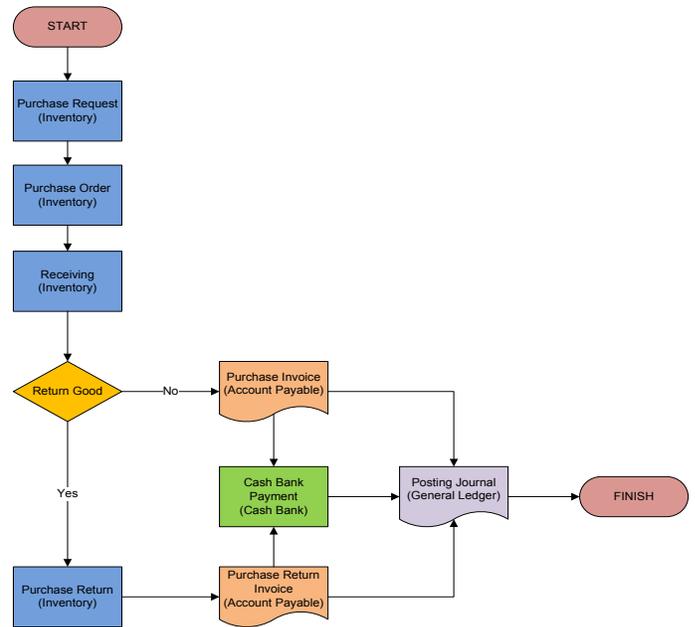
*Order to cash* adalah proses pemesanan barang oleh *customer* kepada perusahaan. Proses diawali dengan diterbitkannya *sales order* oleh domain fungsi *sales* (*sales invoice*). Alur proses ini ditunjukkan pada Gambar 2.

**B. Cloud Computing**

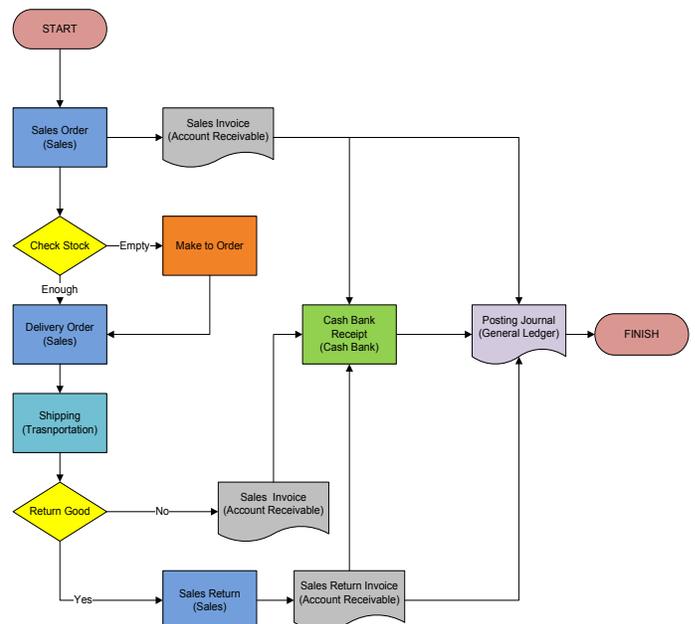
*Cloud computing* adalah suatu paradigma baru, dimana suatu informasi tersimpan secara permanen di suatu tempat di internet, dan dapat diakses melalui *desktop*, komputer *tablet*, *notebook*, *handheld*, sensor-sensor, *monitor* dan lain-lain [9].

Ada 3 tingkatan teknologi pada *cloud computing* [10] yaitu:

- **Infrastructure as service**: pemberian layanan terfokus pada infrastruktur. Contohnya seperti Amazon Elastic Compute Cloud dan *simple storage service*.
- **Platform as a service**: hal ini memungkinkan pengembang untuk tidak memikirkan *hardware* dan tetap fokus pada *application development* nya tanpa harus mengkhawatirkan *operating system*, *infrastructure scaling*, *load balancing* dan lainnya.
- **Software as a service**: hal ini memfokuskan layanan pada perangkat lunak dengan *web based interface* yang diakses melalui *web service* dan *web 2.0*.



Gambar 1. Alur proses bisnis *procure to pay*.



Gambar 2. Alur proses bisnis *order to cash*.

C. Multi Tenant

Perangkat lunak *multi tenant* merupakan suatu prinsip dari arsitektur perangkat lunak, dimana sebuah perangkat lunak yang berjalan di atas *server* melayani banyak pengguna/*tenant*. Dengan prinsip *multi tenant* ini, sebuah perangkat lunak dirancang untuk memiliki partisi data yang berbeda dan dapat dikonfigurasi. *Multi tenant* memiliki kaitan yang sangat erat dengan *cloud computing* [11].

Ada tiga pendekatan dalam manajemen data pada arsitektur *multi tenant*. Tiga pendekatan ini dibedakan berdasarkan rentang *shared data* (berbagi data) dan *isolated data* (data tertutup) [12].

1. *Separated database*. Dalam pelaksanaannya, perangkat lunak yang dibangun akan digunakan secara bersama-sama, tetapi tiap *tenant*/pengguna memiliki tempat penyimpanan data yang terisolasi satu sama lain.
2. *Shared database, separated schema*. Disebut *shared database*, dikarenakan semua pengguna berada pada satu lingkungan basis data yang sama, dan disebut *separated schema* dikarenakan pemisahan data tiap pengguna dilakukan dengan cara pengelompokan tabel-tabel dalam basis data pengguna ke dalam suatu kumpulan yang disebut sebagai *schema*. *Shared database, shared schema* Data tiap pengguna akan disimpan dalam satu basis data dan satu *schema*. Pada tiap tabel di basis data, akan ada tambahan kolom yang akan menyimpan identitas tiap *tenant*.

D. Service Oriented Architecture

*SOA (Service Oriented Architecture)* adalah suatu arsitektur sistem dalam bentuk layanan yang dapat digunakan melalui internet [14]. Secara mendasar, arsitektur ERP telah menerapkan *SOA* dalam pengembangannya. Oleh karena itu, penerapan ERP sebagai suatu perangkat lunak berbasis layanan menjadi lebih mudah untuk dilakukan.

III. METODOLOGI

A. Analisis Permasalahan

Permasalahan yang timbul dalam pembuatan sistem ini, dapat dibagi menjadi tiga bagian yaitu implementasi ERP secara *multi tenant*, pemaketan dan konfigurasi proses bisnis ERP, serta pembuatan perangkat lunak pengatur konfigurasi ERP.

1. Implementasi ERP secara *multi tenant* pada *cloud computing*. Pada pengembangan perangkat lunak ERP 2011 sebelumnya, sebagian besar domain fungsi telah menerapkan prinsip *multi tenant* pada fungsionalitas perangkat lunaknya. Namun, pada kenyataannya, ada beberapa fungsionalitas yang belum menerapkan sistem secara *multi tenant*.
2. Pemaketan dan konfigurasi proses bisnis ERP. Secara garis besar, beberapa variasi skenario yang direncanakan dapat didukung oleh sistem ERP *multi tenant*, dapat dijabarkan sebagai berikut:

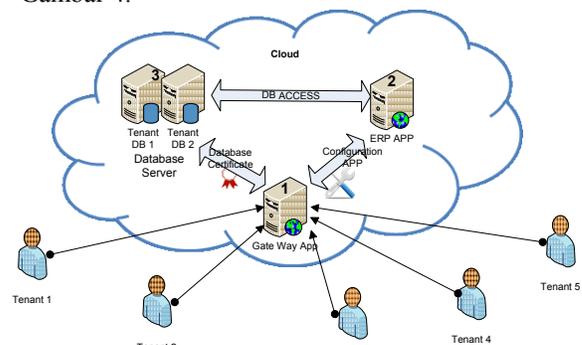
- Sistem ERP dapat digunakan secara penuh dalam satu paket, berarti semua modul digunakan oleh perusahaan. Seluruh alur bisnis *procure to pay* dan *order to cash* berjalan.
  - Sistem ERP dengan hanya menggunakan domain HRM dan *career planning*.
  - Sistem ERP dengan menggunakan domain HRM dan *career planning*. Domain *career planning* ditambah dengan penggajian, sehingga melibatkan domain *account payable, general ledger* dan *cash bank*.
  - Sistem ERP perusahaan hanya menangani masalah distribusi barang saja, artinya, tidak ada proses *manufacturing* di dalam perusahaan.
  - Alur bisnis perusahaan tidak mengenal adanya pengembalian barang ketika barang telah didistribusikan. Alur bisnis perusahaan tidak mengenal adanya pengembalian barang, ketika dilakukan pengadaan barang (*purchase request*).
3. Perangkat lunak pengatur konfigurasi ERP. Selain konfigurasi proses bisnis ERP, diperlukan suatu perangkat lunak yang mampu mendaftarkan, mengatur dan memelihara pemaketan sistem ERP secara keseluruhan. Dari Gambar 3, dapat dilihat bahwa ada suatu perangkat lunak terpusat yang akan mengatur tiap pengguna/*tenant*, mulai dari pendaftaran, hingga akses ke domain-domain ERP. Perangkat lunak ini akan menjadi pintu masuk utama untuk mengakses modul-modul ERP. Pemaketan dan konfigurasi ERP akan ditangani oleh perangkat lunak ini.

B. Perancangan Sistem

Untuk mengatasi semua permasalahan yang telah diidentifikasi, dilakukan proses perancangan dan desain sistem sebagai berikut:

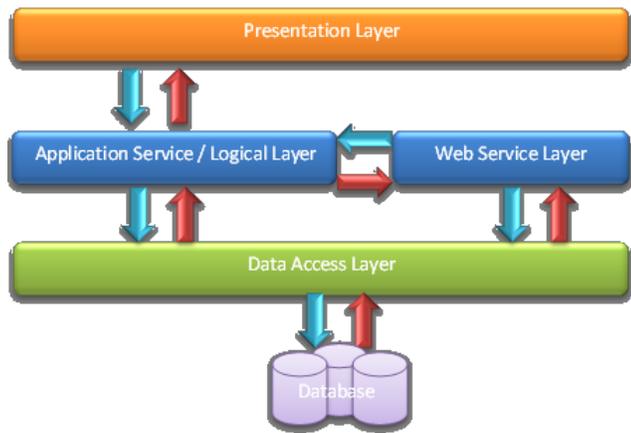
1. Arsitektur Perangkat Lunak

Arsitektur yang dipergunakan dalam pengembangan sistem ini adalah arsitektur berdasarkan pembagian *layer*. Tiap *layer* akan memiliki fungsi tersendiri dan pemisahan *layer* ini dilakukan agar perangkat lunak lebih modular. Pembagian *layer* perangkat lunak dapat dijabarkan dalam Gambar 4.



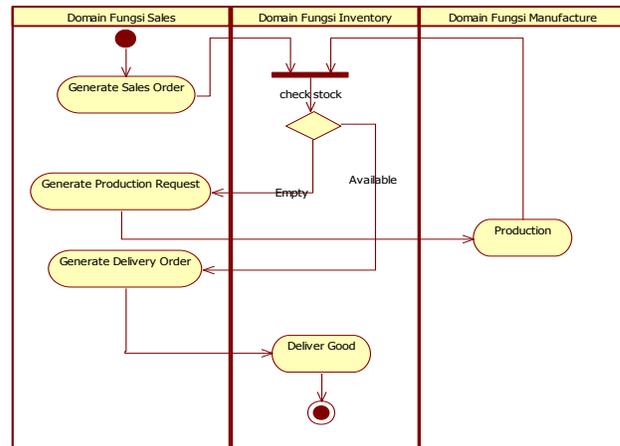
- Keterangan :
1. Gateway APP / Sistem yang akan dibangun, merupakan Sistem yang akan menangani konfigurasi dan manajemen terhadap aplikasi dan database ERP
  2. Merupakan kumpulan Modul – Modul ERP yang telah dikembangkan dalam TA mahasiswa Informatika 2007
  3. Merupakan kumpulan Database yang terpisah untuk menyimpan data tiap tenant

Gambar 3. Gambaran arsitektur sistem secara umum.

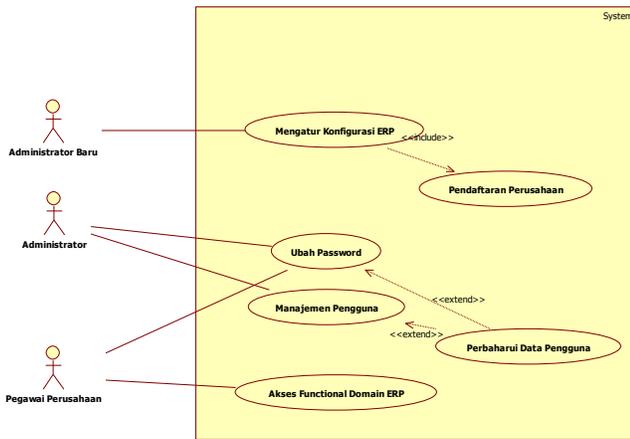


Gambar 4. Pembagian layer perangkat lunak.

memberikan hak akses beberapa domain fungsi yang terlibat yaitu



Gambar 6. Alur *order to cash* pada perusahaan *manufacturing*.

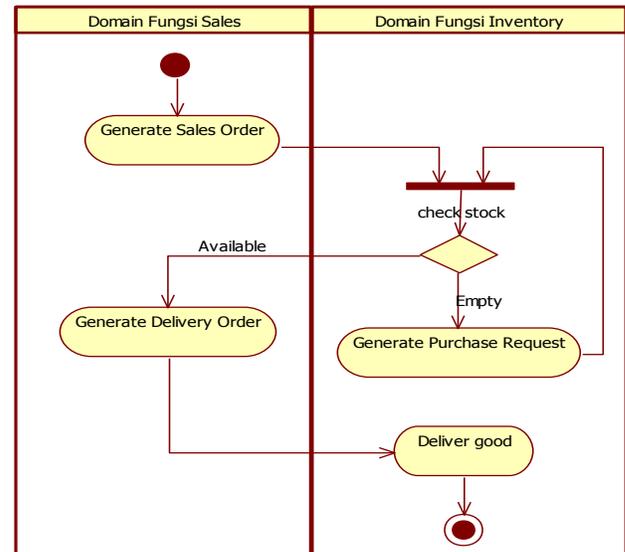


Gambar 5. Kasus penggunaan sistem.

2. **Desain Perangkat Lunak Pengatur Konfigurasi ERP**  
 Dalam pemenuhan kebutuhan fungsional perangkat lunak pengatur konfigurasi ERP, perangkat lunak ini memiliki kasus penggunaan seperti pada Gambar 5.
3. **Desain Pemaketan dan Konfigurasi Proses Bisnis ERP**  
 Proses konfigurasi dilakukan dengan mengubah alur bisnis di tiap modul-modul ERP yang terlibat. Perubahan ini, dilakukan dengan menambahkan beberapa fitur ataupun percabangan pada alur logika perangkat lunak di tiap-tiap modul.

Secara garis besar, beberapa variasi skenario yang direncanakan dapat didukung oleh sistem, diantaranya:

1. Sistem ERP dapat digunakan secara penuh dalam satu paket. Untuk konfigurasi ini, tidak diperlukan implementasi lanjutan, karena hal ini sudah didukung oleh ERP 2011.
2. Sistem ERP dengan hanya menggunakan domain HRM dan *career planning*. Pemaketan untuk konfigurasi ini, dilakukan dengan hanya memberikan hak akses kepada perusahaan untuk mengakses 2 domain fungsi saja yaitu domain fungsi HRM dan domain fungsi *career planning*.
3. Sistem ERP dengan menggunakan domain HRM dan *Career planning* ditambah dengan penggajian. Pemaketan untuk konfigurasi ini dilakukan dengan hanya



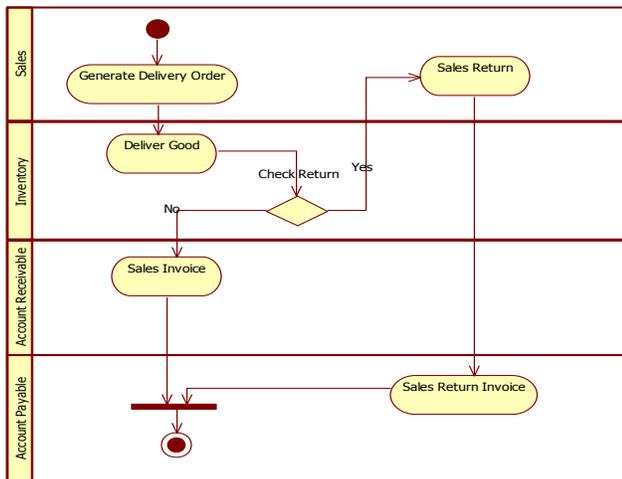
Gambar 7. Alur *order to cash* pada perusahaan *retail*.

HRM, *career planning*, *account payable*, *general ledger* dan *cash bank*.

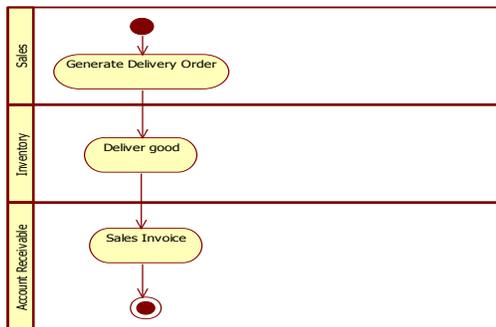
4. Sistem ERP perusahaan hanya menangani masalah distribusi barang saja, artinya, tidak ada proses *manufacturing* di dalam perusahaan. Alur proses bisnis awal ERP yang menangani proses penjualan dengan produksi barang terlebih dahulu, dapat dilihat pada Gambar 6. Proses dimulai dari domain fungsi *sales* yang akan mengeluarkan *sales order*. Kemudian, dilakukan pengecekan barang di *inventory* apakah sudah tersedia. Jika tersedia, maka dilakukan proses *delivery order* dan jika belum, maka dilakukan proses produksi. Untuk variasi skenario yang diajukan, terjadi perubahan alur proses seperti yang ditunjukkan Gambar 7. Perubahan yang terjadi adalah, untuk proses *order to cash*, hanya melibatkan 2 domain fungsi saja, dengan alur sebagai berikut. Jika ada *sales order* dari domain *sales*, dilakukan pengecekan barang di *inventory*. Jika tidak ada, maka

diterbitkan *purchase request* atau beli barang, namun jika barang ada, maka langsung diterbitkan *delivery order* dan dilakukan pengiriman barang.

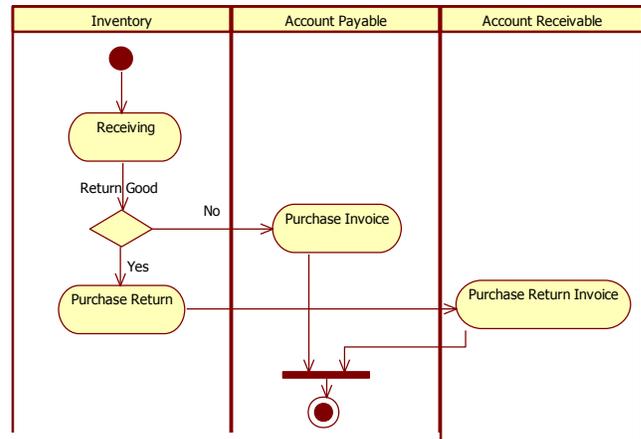
5. Alur bisnis perusahaan tidak mengenal adanya pengembalian barang ketika barang telah didistribusikan. Alur proses bisnis ERP secara umum untuk proses pengembalian barang adalah sebagai berikut: pada Gambar 8, ketika dilakukan perintah pengiriman barang, maka akan di periksa apakah ada proses *return*. Jika ada, maka diterbitkan *sales return invoice*. Jika tidak, maka hanya akan diterbitkan *sales invoice* saja. Pada Gambar 9, alur bisnis yang berubah yaitu, pada saat pengiriman, tidak ada kemungkinan barang yang akan di-*return* dan langsung diterbitkan *sales invoice* untuk barang tersebut (proses *return good* dihilangkan).
6. Alur bisnis perusahaan tidak mengenal adanya pengembalian barang, ketika dilakukan pengadaan barang (*purchase request*). Alur bisnis secara umum untuk proses penerimaan barang adalah, ketika barang diterima, dilakukan pengecekan apakah ada barang yang harus dikembalikan ke distributor barang, jika dikembalikan, maka dilakukan proses *purchase return* dan diterbitkan *purchase return invoice*. Jika tidak, maka langsung diterbitkan *purchase invoice*. Alur bisnis yang mengalami perubahan dapat dilihat pada Gambar 11. Ketika dilakukan pengadaan barang dan barang diterima, maka langsung diterbitkan *purchase invoice*, tanpa ada proses *return* barang.



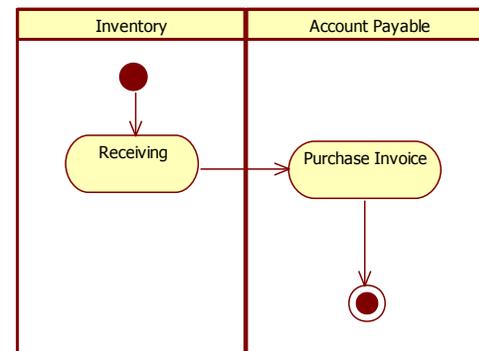
Gambar 8. Alur *delivery order* dengan *sales return*.



Gambar 9. Alur *delivery order* tanpa *sales return*.



Gambar 10. Alur *purchasing* dengan *purchase return*.



Gambar 11. Alur *purchasing* tanpa *purchase return*.

### C. Implementasi

Pada pengembangan perangkat lunak ini, dilakukan proses pengintegrasian modul-modul ERP berupa penambahan sebuah *gateway application* yang akan menangani masalah pendaftaran, konfigurasi dan pemeliharaan sistem ERP. Selain itu, sebelum dilakukan penambahan *gateway application*, dilakukan pembenahan pada perangkat lunak ERP yang sudah ada, dikarenakan ada beberapa kesalahan/*error* sistem pada perangkat lunak ketika digunakan secara *multi tenant*.

1. **Pembenahan Sistem ERP**  
Pembenahan sistem ERP dilakukan terhadap modul-modul yang tidak mendukung *multi tenant*. Pembenahan sistem ini meliputi pembenahan basis data dan pembenahan logika perangkat lunak.
2. **Pemaketan dan Konfigurasi Proses Bisnis ERP**  
Pada proses pemaketan dan konfigurasi ERP, dilakukan perubahan-perubahan alur proses bisnis ERP 2011 sehingga mampu memenuhi konfigurasi yang diinginkan sesuai dengan desain yang telah dibuat.
3. **Perangkat Lunak Pengatur Konfigurasi ERP**  
Pada implementasi perangkat lunak pengatur konfigurasi ERP, dilakukan pembagian implementasi sesuai dengan 4 lapisan desain perangkat lunak yang digunakan yaitu *data access layer*, *data model layer*, *application service layer*, dan *presentation service layer*.

#### 4 Implementasi ERP pada *Cloud Infrastructure*

Sesuai dengan prinsip perangkat lunak *cloud computing, software as a service* harus berjalan di atas infrastruktur *cloud* [13]. Untuk membangun infrastruktur *cloud*, digunakan perangkat lunak VmWare Vsphere sebagai perangkat lunak pengatur virtualisasi *server*. Dengan menggunakan VmWare Vsphere, sumber daya tiap komputer virtualisasi dapat diatur sesuai dengan kebutuhan perangkat lunak.

#### IV. UJI COBA

Uji coba pada perangkat lunak ini, dibagi menjadi 2 bagian yaitu uji coba fungsional dan uji coba skenario perangkat lunak.

##### a. Uji coba fungsional

Pengujian pada perangkat lunak ini dibagi berdasarkan fitur utama serta perubahan alur yang terjadi pada proses bisnis ERP. Skenario yang dijalankan adalah skenario yang menghasilkan hasil positif, maupun negatif (*error*). Hasil uji coba akan dibandingkan dengan reaksi perangkat lunak seharusnya, dan akan dicatat pada hasil pengujian. Pengujian dilakukan dengan metode *blackbox* yang sebatas menguji fungsionalitas dari perangkat lunak ini. Hasil uji coba menggambarkan bahwa semua fitur pada perangkat lunak ERP berfungsi dengan benar.

##### b. Uji coba skenario

Uji coba ini dilakukan dengan menyiapkan skenario proses bisnis perusahaan dan digunakan untuk melihat apakah perangkat lunak bisa berjalan sesuai dengan skenario yang telah diberikan. Dari hasil uji coba yang dilakukan, perangkat lunak ERP dapat berjalan sesuai dengan skenario yang diberikan dan mampu memenuhi kebutuhan proses bisnis perusahaan.

#### V. KESIMPULAN

Kesimpulan yang dapat diambil dari pengerjaan artikel ini adalah:

1. ERP 2011 dapat berjalan secara *multi tenant*.
2. Konfigurasi ERP dengan beberapa alur proses bisnis yang berbeda, dapat dilakukan. Konfigurasi yang dimaksud adalah ERP *retail*, ERP *manufacturing*, ERP dengan *purchase return*, ERP tanpa *purchase return*, ERP dengan *sales return*, ERP tanpa *sales return*, dan ERP dengan HRM.
3. ERP 2011 telah menerapkan prinsip *cloud computing* beraskan *software as a service*.

#### UCAPAN TERIMA KASIH

Penulis R.W.H. mengucapkan terima kasih kepada Allah SWT, kedua orang tua, dosen pembimbing dan semua rekan-rekan penulis yang telah membantu secara langsung maupun tidak langsung terhadap pengerjaan artikel ini.

#### DAFTAR PUSTAKA

- [1] Rifai, Ahmad. "Rancang Bangun Sistem Persediaan (Inventory) dengan Model Software as a Service Menggunakan Service Oriented Architecture". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [2] Airlangga, Notario. "Rancang Bangun Perangkat lunak Pelaksanaan Produksi Berorientasi Servis pada Perusahaan Furnitur Menggunakan Platform Java". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [3] Adhyasa, Tommy. "Rancang Bangun Perangkat lunak Account Payable, Account Receivable dan Fixed Assets dengan Service Oriented Architecture". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [4] Rahmawati, Rina. "Rancang Bangun Perangkat lunak Pengelola Sumber Daya Manusia Berorientasi Servis pada Platform Java". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [5] Dimas, Kemas. "Rancang Bangun Perangkat lunak Pelacakan Aset Bergerak Menggunakan Teknologi GPS dengan Model Perangkat Lunak sebagai Layanan Berbasis SOA pada Platform Java". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [6] Ventyana, Gita. "Rancang Bangun Perangkat lunak Cash Bank dan Sales dengan Service Oriented Architecture pada Platform Java". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [7] Gumilar, Izzudin. "Rancang Bangun Perangkat lunak General Ledger Berorientasi Service pada Platform Java". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [8] Suryawan, Dedy. "Rancang Bangun Perangkat lunak Career Planning Berorientasi Service pada Platform Java". Tugas Akhir Teknik Informatika ITS. Surabaya, 2011.
- [9] Carl Hewitt, "ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing". IEEE Internet Computing, vol. 12, no. 5, pp. 96-99, Sept.-Oct. 2008, doi:10.1109/MIC.2008.107
- [10] Teknologi Cloud Computing. <http://teknoinfo.web.id>. [Online] [Dikutip: 10 Mei 2012.] <http://teknoinfo.web.id/teknologi-cloud-computing>.
- [11] Multi Tenancy. <http://techtarget.com> [Online] [ Dikutip: 12 April 2012.] <http://whatis.techtarget.com/definition/multi-tenancy>.
- [12] Multi Tenant. [msdn.microsoft.com](http://msdn.microsoft.com). [Online] [Dikutip: 10 April 2012.] <http://msdn.microsoft.com/en-us/library/aa479086.aspx>.
- [13] Mel, Peter, dan Timothy Grace. 2011, "The NIST Definitions of Cloud Computing-Recommendations of the National Institute of Standards and Technology". NIST Special Publication 800-145,