

SUPPLY CHAIN MANAGEMENT BERBASIS LAYANAN: DESAIN DAN IMPLEMENTASI PROTOTIPE SISTEM

Jarot Budiasto, Fransiskus X. Manggau
jr.budiasto@gmail.com, fransiskus@unmus.ac.id

Jurusan Sistem Informasi Fakultas Teknik
Universitas Musamus Merauke

Abstrak

Supply Chain Management (SCM) adalah sebuah konsep pengaturan aliran proses perdagangan yang menghubungkan antara produsen, supplier, dan konsumen secara langsung. Dengan berevolusinya konsep perangkat lunak dan sistem informasi menuju era berbasis layanan, pengembangan SCM berbasis layanan menjadi sangat relevan. Studi kasus kebutuhan akan SCM berbasis layanan pada pengelola pusat perbelanjaan di Jakarta menjadi fokus pembahasan. Pada makalah ini dibahas sebuah model pengembangan sistem SCM berbasis layanan yang menghubungkan tiga stakeholder yang berperan dalam proses perdagangan pada pusat perbelanjaan modern.

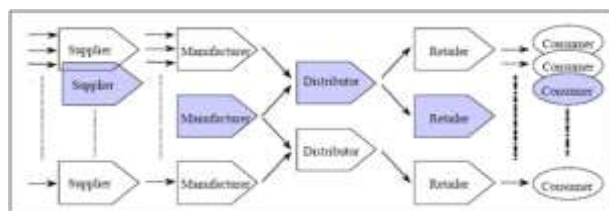
Kata kunci: SCM, prototyping, tenant, perdagangan.

A. PENDAHULUAN

Kompetisi yang semakin sengit dalam pasar global, inovasi produk yang memiliki siklus hidup yang semakin singkat, serta ekspektasi pelanggan yang semakin tinggi memaksa seluruh perusahaan untuk berinvestasi dan fokus pada *supply chain* mereka [1]. Hal ini disebabkan karena para eksekutif sudah semakin menyadari bahwa kesuksesan suatu perusahaan sangat bergantung pada kesuksesan koordinasi, integrasi, dan manajemen proses-proses bisnis penting dari berbagai anggota *supply chain* [2]. Dengan kata lain, keberhasilan suatu perusahaan sangat bergantung pada *Supply Chain Management (SCM)* yang mereka kelola. Bidang SCM kemudian berkembang pesat seiring dengan peran TIK yang mampu mempermudah koordinasi pertukaran data dalam jumlah yang sangat besar [2].

Supply chain tidak dapat dilepaskan dari konsep SCM. Secara definisi yang dimaksud dengan *supply chain* merupakan serangkaian proses pengambilan keputusan dan tindakan serta aliran/alur bahan baku, informasi, dan uang yang bertujuan untuk memenuhi kebutuhan pelanggan yang berlangsung pada suatu tahap yang sama atau berbeda [2]. Untuk mempermudah

pemahaman, berikut ini adalah gambaran umum mengenai *supply chain*



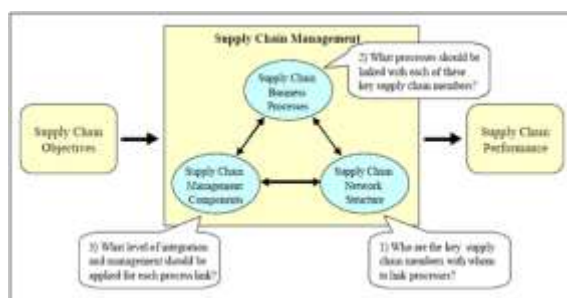
Gambar 1. Diagram *supply chain* dalam jaringan *supply chain total* [2].

Gambar 1 mengilustrasikan bagaimana *supply chain* suatu perusahaan yang biasanya memiliki lebih dari satu supplier dan konsumen. Ilustrasi tersebut menggambarkan proses bisnis dari hulu hingga hilir, mulai dari supplier hingga distribusi kepada pelanggan. Rantai yang berada di antara keduanya tidak terbatas hanya pada pabrikan, distributor, dan retailer saja, tetapi dapat ditambah dengan *transporter*, *warehouse*, *marketing*, *finance*, dan *customer service*, tergantung pada proses bisnis [3]. Jika berbagai rantai dan entitas yang tergabung dalam *supply chain* tidak dikelola dengan baik, hal itu dapat menyebabkan inefisiensi dan berpotensi merugikan. Pihak yang sering dirugikan adalah

para produsen berskala kecil yang memiliki posisi tawar yang rendah.

SCM merupakan rangkaian kegiatan perencanaan, koordinasi, dan pengendalian seluruh proses bisnis dan aktivitas dalam *supply chain* untuk menciptakan *consumer value* terbaik dengan biaya yang efisien namun tetap memenuhi seluruh kebutuhan *stakeholder* lain dalam *supply chain* [2]. *Value* atau yang lebih dikenal dengan *added value* [4] adalah sesuatu yang ingin diperoleh bagi para konsumen dan tercermin dari pendapatan yang dihasilkan oleh perusahaan. Untuk menghasilkan *value* yang

28 al, ada tiga keputusan yang harus pkan oleh para eksekutif: bagaimana struktur jaringan; proses bisnis; dan komponen manajemen dari *supply chain* [5] seperti digambarkan pada Gambar 2.



Gambar 2. Tiga keputusan penting dalam menerapkan SCM [5.]

SCM sangat bergantung pada tujuan awal dari *supply chain* dan pencapaiannya harus dapat diukur melalui *Key Performance Indicators* (KPI). Akan tetapi, tidak mudah dalam menentukan tujuan karena *supply chain* dikelola oleh suatu entitas yang dominan atau melalui kerjasama antar entitas dalam *supply chain* yang membutuhkan kooperasi dan koordinasi yang baik. Setelah tujuan dan pengukuran kinerjanya sudah terdefinisi, ada beberapa keputusan yang harus ditetapkan dalam menganalisa dan merancang *supply chain*. Keputusan yang pertama adalah menentukan siapa saja pihak-pihak yang terlibat dalam *supply chain* untuk menentukan entitas mana yang menjadi kunci penentu kesuksesan organisasi. Keputusan berikutnya yang perlu ditetapkan adalah proses-proses mana saja yang harus dihubungkan pada setiap entitas kunci dalam *supply chain*. Hal ini penting untuk meningkatkan efektifitas dan efisiensi proses bisnis yang ada. Keputusan yang terakhir adalah menentukan tingkat integrasi dan manajemen dalam setiap proses yang saling terhubung. Pada bagian ini harus didefinisikan

dengan jelas bagaimana peranan komponen manajemen sehingga dapat selaras dengan seluruh proses bisnis yang ada dalam *supply chain*.

SCM banyak diimplementasikan oleh perusahaan berskala besar yang memiliki volume aliran bahan baku, informasi, dan uang dalam jumlah besar. Faktor yang sangat relevan mengingat biaya yang harus dikeluarkan untuk membangun sistem dan melakukan *maintainance* yang sangat besar. Lantas, apakah para pengusaha bisnis dalam skala kecil tidak dapat menikmati akselerasi bisnis dengan SCM yang ternyata membutuhkan investasi IT yang tidak kecil? Tren perangkat lunak yang mulai bergeser ke arah perangkat lunak berbasis layanan tentu saja mempengaruhi pola pengembangan SCM.

Produk SCM berbasis layanan yang dibangun oleh sebuah penyedia jasa utama yang *men-support* aktivitas *tenant-tenant* merupakan sebuah produk layanan jasa yang akan membantu peningkatan aktivitas bisnis pelaku bisnis berskala kecil. Pada makalah ini akan dibahas bagaimana desain sebuah sistem SCM berbasis layanan yang dirancang untuk membantu aktivitas perdagangan antara distributor besar (*wholeseller*) dengan para penjual langsung (*retailer*) dalam sebuah komunitas pusat perdagangan modern. Prototipe pengembangan sistem ini akan mengambil lokasi pada sebuah pusat perdagangan besar di kota Jakarta.

1. Studi Kasus

Jakarta selain menjadi daerah pusat pemerintahan juga merupakan tempat yang strategis untuk melakukan aktivitas perdagangan dengan jumlah penduduknya yang besar dan memiliki pola hidup dinamis serta gaya hidup konsumtif. Oleh karena itu, Jakarta merupakan tempat yang subur untuk tumbuhnya pusat-pusat perdagangan. Pertumbuhan pusat perdagangan dihiasi dengan makin maraknya pusat-pusat perbelanjaan modern (*mall*) yang bertebaran di daerah yang memiliki penduduk kurang lebih sebesar 8,5 juta jiwa. Data Dinas Penataan dan Pengawasan Bangunan DKI Jakarta menunjukkan, jumlah *mall* yang berdiri di Jakarta mencapai 130 gedung. Keberadaannya terpusat di kawasan Mangga Dua, Bundaran HI, Blok M, Kelapa Gading, dan Senayan. Dengan jumlah *mall* yang sangat banyak, persaingan diantaranya pun semakin ketat. Sehingga pengelola sebuah *mall* perlu mengupayakan adanya strategi dan perencanaan bisnis yang baik

untuk menciptakan lingkungan aktifitas jual beli yang nyaman sehingga pengunjung semakin tertarik untuk berbelanja.

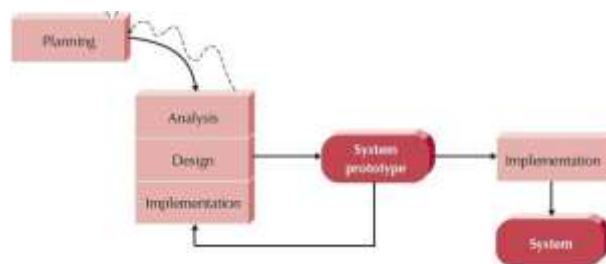
PT. XYZ merupakan perusahaan pengelola *mall* di kawasan pusat bisnis Jakarta yang mempunyai segmentasi pasar khusus penjualan telepon selular dan alat-alat elektronik. Klien-klien utama merupakan penyewa tempat yang terdiri dari para *wholeseller (distributor)* dan penjual langsung (*retailer*). Di dalam aktifitas perdagangan yang berlangsung di *mall* tersebut, interaksi antara distributor dan penjual mempunyai intensitas yang tinggi. Aktifitas tersebut antara lain seperti pengecekan barang, melihat status ketersediaan, membeli barang dan sebagainya. Namun, dalam operasional selama ini untuk memperoleh data dan informasi masih dilakukan dengan cara manual, seperti menggunakan sarana komunikasi dua arah secara langsung atau dengan cara melihat satu-persatu melalui katalog. Segala proses tersebut membutuhkan waktu lama dan membutuhkan konsumsi kertas yang tinggi. Proses tersebut menghabiskan biaya yang besar dan mengakibatkan ekonomi biaya tinggi dalam berwirausaha dan berpotensi mengurangi kenyamanan berbelanja bagi pengunjung.

Solusi yang dapat diambil dari permasalahan kecepatan dan ketepatan dalam memperoleh, mencari serta pertukaran data dan informasi antar distributor dan penjual adalah dengan cara mengimplementasikan sistem informasi SCM tunggal yang terintegrasi. Sistem informasi ini dapat meningkatkan kinerja bagi kedua pihak (*distributor* dan *retailer*) dalam menjalankan aktifitas bisnisnya. Sistem ini berupa layanan dimana pihak pengelola *mall* merupakan *intermediary* yang menyediakan jasa bagi *distributor* dan *retailer* yang pada akhirnya berujung pada peningkatan pendapatan baru dan memberikan pelayanan terbaik bagi para penyewa tempat dan para pengunjung *mall*.

2. Metodologi Pengembangan

Metodologi yang dipakai dalam pengembangan sistem dalam makalah ini adalah *prototyping*. Metodologi pengembangan berbasis *prototyping* memungkinkan pengembang untuk melakukan fase analisis, desain dan implementasi secara bersamaan. Ketiga fase dilakukan berulang kali dalam sebuah siklus pengembangan sistem sehingga seluruh kebutuhan dan fitur-fitur dalam sistem lengkap. Hasil dari *prototyping* merupakan representasi

versi lebih kecil dari sistem dengan jumlah fitur yang minimal (*workflow* yang menjelaskan metodologi *prototyping* dapat dilihat pada Gambar 3.). Alasan dalam memutuskan penggunaan metodologi *prototyping* dalam pengembangan sistem SCM berbasis layanan ini karena sistem yang dibangun memiliki *user requirement* yang belum begitu jelas serta waktu yang tersedia untuk membangun sangat terbatas.



Gambar 3. Tahapan-tahapan dalam metodologi *prototyping*

Keuntungan dengan menggunakan metodologi ini adalah memungkinkan pengguna berinteraksi dengan cepat dengan model sistem yang akan dibangun sehingga kebutuhan-kebutuhan yang belum teridentifikasi dengan jelas, dapat dipenuhi. Sebaliknya, metodologi ini juga memiliki kerugian yaitu seringkali prototipe dari sistem mengalami perubahan yang signifikan sehingga menyulitkan pengembang untuk menyelesaikan tugasnya.

3. Identifikasi Kebutuhan Sistem

Proses identifikasi kebutuhan sistem untuk membangun SCM berbasis layanan sebetulnya merupakan proses yang sulit. Dalam kondisi nyata, *users* dari sistem ini tidak terbatas kepada satu pedagang (*distributor* atau *retailer*) saja. Apabila pengembangan sistem ini merujuk kepada proses *waterfall*, maka tahapan identifikasi kebutuhan sistem akan memakan waktu yang sangat lama mengingat banyaknya *tenant* yang menjadi calon pengguna dari sistem. Proses *prototyping* memotong waktu yang signifikan pada tahapan ini. Karakteristik metodologi *prototyping* yang tidak meng-*capture* informasi awal secara detail akan memudahkan proses identifikasi awal dari kebutuhan para pengguna. Identifikasi kebutuhan secara umum dilakukan pada tahapan awal untuk kemudian direvisi sesuai dengan *feedback* dari *users* setelah sistem SCM berbasis layanan *go live*.

Identifikasi kebutuhan dari sistem SCM berbasis layanan dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non-

fungsional. Tidak ada prioritas diantara kedua aspek ini. Keduanya memiliki peranan yang sangat penting dalam keberhasilan pengembangan sistem. Secara umum kebutuhan fungsional dari sistem ini adalah sebagai berikut

- a. Fungsi mengelola data *customer (distributor dan retailer)* oleh pengelola *mall* sebagai penyedia layanan.
- b. Fungsi mengelola data barang yang dilakukan oleh *customer (distributor dan retailer)*.
- c. Fungsi untuk mengotomasi proses transaksi keuangan baik itu berupa pembayaran tagihan maupun aktivitas *cash flow* keuangan.
- d. Fungsi otomasi *capturing* dan *input* data menggunakan *third party device*.
- e. Fungsi monitoring proses transaksi yang dilakukan oleh pengelola.
- f. Fungsi *summarizing* dalam bentuk laporan penjualan
- g. Fungsi analisa statistik dalam bentuk diagram dan grafik.

Identifikasi kebutuhan fungsional dapat beubah sesuai dengan proses iterasi dalam metodologi *prototyping*. Kebutuhan utama yang berhasil diidentifikasi dapat bertambah dan juga berkurang sesuai dengan *feedback* dari *users* dalam proses evaluasi prototipe sistem. Sementara itu, daftar kebutuhan non fungsional dari sistem SCM berbasis layanan dapat dilihat dalam Tabel 1.

Tabel 1. Kebutuhan non fungsional sistem SCM berbasis layanan

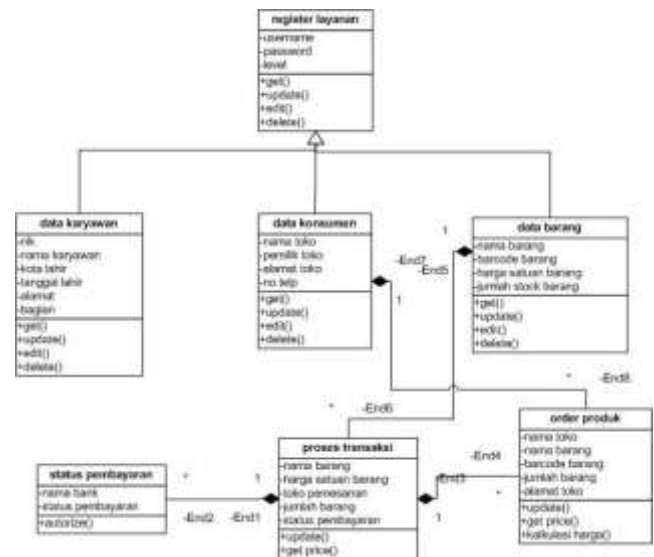
Kebutuhan	Penjelasan
1. <i>Performanc</i>	<ol style="list-style-type: none"> a) Mengefisienkan waktu proses pengolahan data sistem, mulai dari penginputan hingga pelaporan. b) Membantu peningkatan pemantauan perkembangan. c) Mengurangi tingkat kesalahan dan ketidaklengkapan data

2. <i>Data Managemen t</i>	<ol style="list-style-type: none"> a) Melakukan penyimpanan data berupa informasi data barang, data konsumen (<i>retail</i>), karyawan, fasilitas dan data transaksi. b) Mencegah terjadinya penyimpanan data yang redundant. c) Mencegah hilangnya data. d) Sistem pusat dan cabang terintegrasi sehingga memudahkan untuk mendapatkan data yang paling aktual. e) Format penyajian laporan dibuat sehingga lebih mudah dipahami. f) Meminimalisasi terjadinya kesalahan penginputan data. g) Data terdokumentasi dan terstruktur.
----------------------------	--

3. <i>Economic</i>	<ol style="list-style-type: none"> a) Mengurangi biaya operasional untuk transfer informasi atau dokumen ke pusat yang selama ini dilakukan secara manual. b) Memperlancar aliran informasi antara bagian administrasi ke managerial
--------------------	--

4. <i>Control</i>	<ol style="list-style-type: none"> a) Meningkatkan keamanan terhadap pelaksanaan proses penyimpanan data. b) Membatasi akses penggunaan terhadap sistem dengan cara menerapkan privilege. c) Adanya operator data entry yang bertanggungjawab terhadap pelaksanaan pemasukan data dan administrator yang bertanggung jawab atas semua jalannya aktifitas pada aplikasi d) Mencegah akses penuh dari pengguna-pengguna yang tidak berwenang.
-------------------	---

5. <i>Efficiency</i>	<p>a) Menggunakan sistem penyimpanan data yang terpusat untuk memudahkan proses pendistribusian barang.</p> <p>b) Mengefisienkan waktu untuk pelaksanaan proses validasi penginputan data</p> <p>c) Meminimalisasi biaya dan sumber daya yang dibutuhkan untuk pelaksanaan proses pelaporan.</p>
6. <i>Service</i>	<p>a) Menghasilkan informasi yang akurat untuk bahan pertimbangan dan evaluasi.</p> <p>b) Memberi kemudahan dalam penggunaan operasional sistem.</p>



Gambar 4. Pemodelan struktur berbasis *object oriented* dengan UML *class diagram*

4. Analisa dan Pemodelan Masalah

Pemodelan dalam proses analisa masalah dalam makalah ini menggunakan *Unified Modelling Language* (UML). Ada dua jenis tipe diagram dalam UML, yang pertama adalah diagram yang memodelkan struktur dari sistem dan yang kedua adalah diagram yang memodelkan perilaku dari sistem. Penggunaan metodologi *prototyping* membuat proses pemodelan tidak perlu dilakukan secara mendetail. Untuk memodelkan sistem SCM berbasis layanan yang dibahas pada makalah ini, digunakan dua jenis diagram untuk memodelkan masing-masing karakteristik. Pemodelan *structure* dari sistem digunakan *class diagram* yang merupakan representasi dari pemodelan berbasis *object oriented* dan untuk memodelkan *behavior* digunakan *use case diagram*. Tidak ada pemodelan data dalam bentuk *entity relationship diagram* karena struktur penyimpanan data tidak menjadi masalah yang cukup *urgent* untuk dimodelkan.

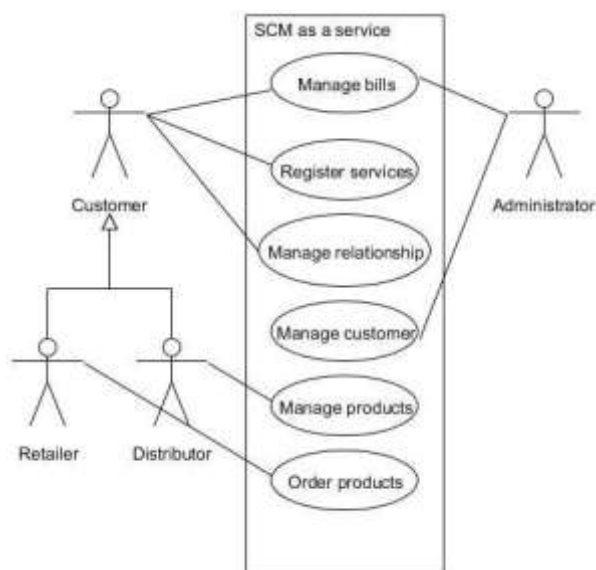
Struktur dari sistem dimodelkan dengan *class diagram* yang merepresentasikan masing-masing modul yang dibangun untuk merepresentasikan sistem secara utuh. Berikut ini adalah model dari struktur sistem yang digambarkan dalam *class diagram*

Struktur dari sistem SCM berbasis layanan dibagi menjadi tiga bagian utama yaitu modul yang merepresentasikan pengelola *mall* yang menjadi administrator dari sistem SCM berbasis layanan, pada *class diagram* direpresentasikan dengan *class* ‘register layanan’. Modul fungsi dari kedua jenis *customer* yang menjadi *users* atau *tenant* dari *mall* ini (*distributor* dan *retailer*) direpresentasikan menjadi dua jenis melalui *class diagram*. Struktur yang pertama menggambarkan aktivitas internal *customer* yang melakukan *management* internal terhadap kebutuhan dari setiap *customer*, proses ini direpresentasikan dengan *class* ‘data karyawan’, ‘data konsumen’, dan ‘data barang’. Ketiga *class* yang menggambarkan fungsi internal dari setiap *customer* ini terhubung dengan *class* yang merepresentasikan pengelola *mall* dan tiga *class* lain yang merupakan representasi dari aktivitas antar *customer* yaitu proses jual-beli dalam *mall* yang terjadi antara *distributor* dan *retailer*.

Tiga *class* terakhir dibuat untuk menunjukkan struktur sistem yang merepresentasikan proses aktivitas utama dari organisasi. *Class* ‘order produk’, ‘proses transaksi’, dan ‘status pembayaran’ secara eksplisit menggambarkan bagaimana keterhubungan antara *distributor* dan *retailer* dalam melakukan aktivitas jual-beli dalam sebuah pusat perdagangan. Ketiga *class* yang memodelkan struktur aktivitas perdagangan ini tidak berdiri sendiri, mereka memiliki keterkaitan erat dengan empat *class* lain yang juga memiliki fungsi untuk merepresentasikan

modul lain dalam sistem SCM berbasis layanan ini.

Pemodelan lain yang digunakan dalam makalah ini untuk merepresentasikan *behavior* dari sistem adalah *use case diagram*. Dari *use case diagram*, dapat dilihat apa saja aktivitas dan perilaku *users* dengan sistem dan dapat dilihat pula seberapa jauh interaksi itu membutuhkan fungsi yang perlu diimplementasikan dalam sistem. Berikut ini adalah model dari *behavior* sistem yang direpresentasikan menggunakan *use case diagram*

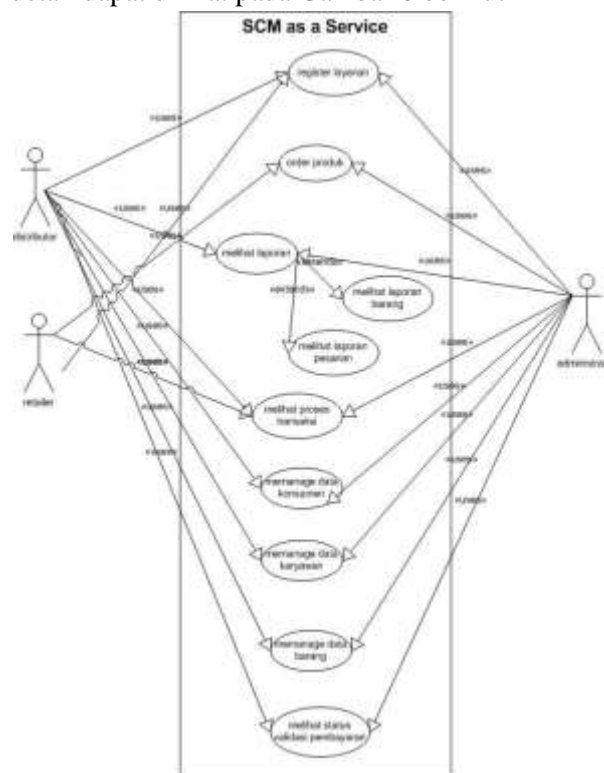


Gambar 5. *Use case diagram* dari sistem SCM berbasis layanan versi 1.0

Gambar 5 memperlihatkan model yang paling sederhana dari *behavior* sistem SCM berbasis layanan. Ada dua jenis aktor yaitu penyedia layanan yang direpresentasikan dengan ‘*administrator*’ dan pihak pengguna jasa layanan yang digambarkan dengan ‘*customer*’. Aktor ‘*customer*’ merupakan generalisasi dari *distributor* dan *retailer* dimana keduanya memiliki irisan dalam melaksanakan fungsi tertentu dalam sistem. Selain itu, terdapat enam aktivitas utama yang menggambarkan fitur sistem yang menjadi tulang punggung layanan SCM ini.

Proses pemodelan *behavior* dari sistem tidak berhenti pada *use case diagram* versi 1.0 yang dimuat dalam Gambar 5 saja. Ada beberapa *behavior* yang perlu dijabarkan dengan lebih detail sehingga dapat merepresentasikan *behavior* sistem secara lebih lengkap. Pemodelan

versi 2.0 dengan *use case diagram* yang lebih detail dapat dilihat pada Gambar 6 berikut ini



Gambar 6. *Use case diagram* dari sistem SCM berbasis layanan versi 2.0

Tidak ada perubahan dari aktor yang terlibat, ketiga aktor yaitu ‘*administrator*’ yang merepresentasikan penyedia layanan dan dua aktor lainnya yang merepresentasikan pengguna jasa layanan yaitu ‘*distributor*’ dan ‘*retailer*’. Hal yang cukup signifikan dari *use case diagram* SCM berbasis layanan versi 2.0 adalah proses detail dari setiap *use case* yang disusun digambarkan dengan lebih jelas.

Pemodelan yang digunakan dalam makalah ini menggunakan *object oriented* dengan harapan proses pemahaman terhadap fungsi-fungsi utama sistem dapat dilakukan dengan lebih mudah. Dua *diagram* yang digunakan untuk memodelkan *structure* dan *behavior* dari sistem dirasa cukup untuk menggambarkan proses bisnis yang di-*capture* dari identifikasi kebutuhan fungsional dan kebutuhan non fungsional dari sistem SCM berbasis layanan. Hanya digunakannya *class diagram* dan *use case diagram* tanpa melibatkan *diagram* lain dalam pemodelan sistem dilakukan dengan argumentasi, simplifikasi proses *prototyping* tidak memerlukan pemodelan detail yang merepresentasikan seluruh proses bisnis dari sistem.

5. Desain Sistem

Perbedaan utama dari sebuah sistem yang dibangun untuk keperluan internal dengan sistem yang dibangun untuk memberikan layanan kepada *users* terletak pada bagian desain. Pengembangan sistem yang berorientasi ‘*goods*’ atau produk menonjolkan prinsip *capturing requirements* yang tersentralisasi pada sebuah entitas organisasi yang merupakan *user* utama dari sistem. Sementara itu, sistem yang dibangun yang dimaksudkan untuk memberikan layanan kepada banyak *users* dalam waktu yang bersamaan harus dapat mengakomodir kebutuhan umum dari setiap *users*. Ilustrasi yang memberikan gambaran lengkap mengenai perbedaan sistem yang berorientasi ‘*goods*’ dengan sistem yang berorientasi ‘*services*’ dapat dilihat pada Gambar 7

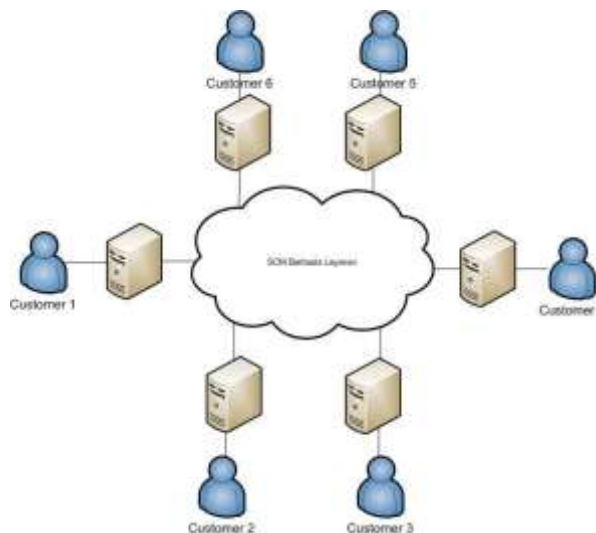


Gambar 7. Pengembangan perangkat lunak berorientasi ‘*goods*’ dan ‘*services*’

Pada makalah ini, sistem SCM berbasis layanan yang akan dibangun direpresentasikan dalam dua jenis diagram yang memperlihatkan desain dari sistem secara keseluruhan. Diagram yang pertama menggambarkan *network design* dari sistem SCM berbasis layanan dan diagram yang kedua menggambarkan *database design* dari sistem. Penggunaan kedua diagram ini sudah cukup memperlihatkan bagaimana desain dari arsitektur sistem SCM berbasis layanan yang diusulkan.

Desain dari sistem yang diusulkan memiliki kekuatan utama pada desain arsitektur yang berbasis layanan. Ilustrasi yang digunakan pada Gambar 7 yang memperlihatkan perbedaan paradigma pengembangan sistem yang berorientasi ‘*goods*’ dan ‘*services*’ mewarnai proses desain arsitektur yang dirancang untuk

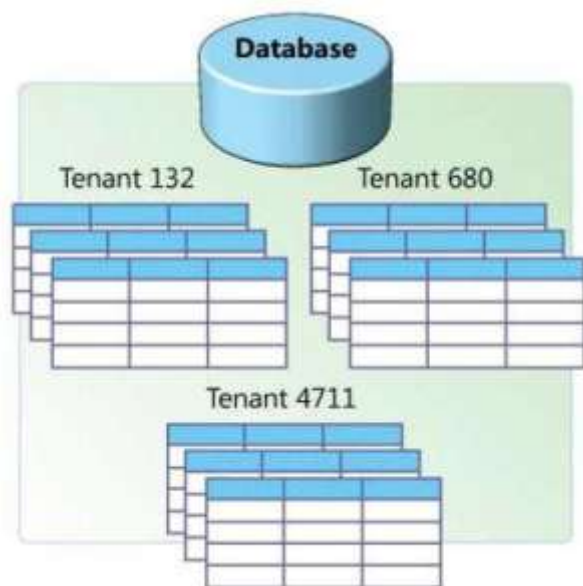
membangun sistem SCM berbasis layanan. Diagram pertama adalah diagram yang merepresentasikan desain dari topologi jaringan dari sistem. Desain dari topologi jaringan dapat dilihat pada gambar 8 berikut



Gambar 8. Desain topologi jaringan dari sistem SCM berbasis layanan.

Sistem SCM berbasis layanan digambarkan dalam sebuah *cloud* yang merepresentasikan sebuah sistem pada layer yang diakses secara *transparent* oleh *users*. *Users* yang terdiri dari para *distributor* dan *retailer* tidak perlu tahu bagaimana struktur dari sistem tersebut secara detail. Salah satu keuntungan dari sistem semacam ini adalah penambahan jumlah *users* tidak akan mempengaruhi perubahan struktur program dalam sistem karena setiap kali dilakukan penambahan *users*, yang terjadi adalah hanya penambahan *server* aplikasi pada level *client* yang mengakses *server* utama yang berisi layanan sistem SCM pada *cloud* yang merupakan representasi layanan yang diberikan. Gambar 8 memperlihatkan bagaimana setiap *users* berinteraksi dengan *cloud* dalam sebuah topologi jaringan. Satu diagram lagi yang perlu untuk dimuat untuk lebih memperjelas bagaimana perbedaan antara sistem SCM yang dibangun secara *dedicated* untuk satu entitas organisasi dengan sistem SCM berbasis layanan yang dibangun untuk melayani banyak *users*. Diagram terakhir yang digunakan dalam proses desain untuk menggambarkan keunikan dari sistem SCM berbasis layanan ini adalah *database design*. Konsep utama dari *database design* yang digunakan adalah konsep *shared database* dengan *multi-tenant* [6]. Desain dari konsep

shared database dengan *multi-tenant* dapat dilihat pada Gambar 9 berikut ini



Gambar 9 . *Database design* dengan konsep *shared database* dan *multi-tenant* yang digunakan dalam sistem SCM berbasis layanan [6].

Dengan model *shared database* dengan *multi-tenant*, setiap *users* dimungkinkan untuk memiliki skema *database* masing-masing yang unik dan bersifat *customized*. Dengan desain *database* seperti ini akan mengurangi *cost* untuk membangun *dedicated database* untuk setiap *users* (ilustrasi dapat dilihat pada Gambar 10) yang menginginkan *customized data model* yang pasti terjadi. Proses *customized* yang efisien dimungkinkan untuk dibangun dengan mendasarkan pengembangan sistem yang berorientasi '*services*'.

Ilustrasi pada Gambar 10 memperlihatkan bagaimana konsep *dedicated database* dengan *multi-tenant* yang memiliki perbedaan yang cukup signifikan dari konsep yang dibahas sebelumnya. Model desain pada Gambar 10 pun merupakan *database design* yang biasa digunakan dalam membangun sistem berbasis layanan. Namun demikian, sistem SCM berbasis layanan tidak menggunakan desain yang relatif *costly* untuk sebuah sistem sederhana yang memberikan layanan SCM kepada para *distributor* dan *retailer* yang tidak memiliki skala *enterprise*. Kebutuhan untuk *customized database schema* yang murah dapat dipenuhi dengan konsep *shared database* dengan *multi-tenant* dibanding

mengimplementasikan konsep *database design* yang ditunjukkan pada Gambar 10.



Gambar 9. Ilustrasi *database design* dengan konsep *dedicated database* dengan *multi-tenant* [6]

6. Kajian dan Diskusi

Pemaparan mengenai bagian kedua dari tahapan *prototyping* yaitu *analysis* dan *design* (minus *implementation*) telah dibahas pada makalah ini. Ada beberapa isu penting yang harus dijelaskan terkait studi kasus yang digunakan, metodologi pengembangan, dan proses analisis dan desain yang dilakukan. Isu-isu ini dapat dikatakan sebagai hambatan dalam pengembangan tetapi di pihak lain menjadi *challenge* dan *knowledge* baru dalam melakukan pengembangan sistem berbasis layanan.

Isu yang pertama adalah pada analisis dari studi kasus yang ada sebelum menentukan jenis pengembangan sistem yang berbasis layanan atau berorientasi '*services*' ketimbang berorientasi '*goods*'. Ada satu hal penting yang perlu diperhatikan yang menjadi isu yaitu adanya banyak *users* yang merupakan entitas bisnis yang berbeda namun memiliki kesamaan fungsi umum yaitu sebagai *tenant* pada sebuah *mall* yang dikelola oleh perusahaan tertentu. Salah satu ciri dari sistem berbasis layanan adalah memiliki banyak *users* yang merupakan entitas organisasi yang saling berbeda dengan kebutuhan yang berbeda namun memiliki satu atau beberapa kebutuhan layanan yang sama yang disediakan oleh penyedia jasa layanan. Karena itu pula, ada pertimbangan *personalization* atau *customization* yang harus dipenuhi oleh penyedia jasa layanan tanpa harus mengorbankan sistem utama.

Isu yang kedua ketika memilih metodologi yang tepat untuk dijadikan acuan pengembangan sistem. Pemilihan metodologi *prototyping* tidak semata-mata karena studi kasus dalam makalah ini memiliki karakteristik *requirements* yang belum begitu detail dan durasi waktu pengerjaan yang pendek. Sistem SCM berbasis layanan memiliki banyak *users* yang berperan sebagai *tenant*, tidak mungkin pendekatan *Joint*

Application Development (JAD) digunakan untuk mengakomodir permasalahan tersebut. Selain jumlah *tenant* yang banyak, potensi penambahan jumlah *tenant* akan menjadi masalah jika pengembangan sistem yang dilakukan berorientasi akomodatif. Proses dalam *prototyping* yang memungkinkan siklus evaluasi dan perubahan terhadap sistem yang dibangun memungkinkan pengembangan sistem secara langsung dan cepat dengan memenuhi kebutuhan dasar dari para *users* dan melakukan proses revisi berkelanjutan seiring dengan berjalannya waktu dan penambahan jumlah *tenant* yang menggunakan layanan ini.

Isu yang ketiga terkait dengan tahapan *system analysis*. Sebetulnya, tidak banyak isu yang dapat diangkat pada fase ini yang terkait erat dengan konsep sistem berbasis layanan. Namun demikian, penggunaan diagram-diagram yang merepresentasikan prinsip *object oriented* dengan *Unified Modelling Language* (UML) akan sangat membantu dan memudahkan proses pemodelan pada tahap analisis.

Isu yang terakhir dan merupakan isu yang cukup penting dalam pengembangan sistem berbasis layanan adalah isu yang terkait dengan fase desain. Perbedaan utama antara sistem yang berorientasi ‘*goods*’ dengan sistem yang berorientasi ‘*services*’ terletak pada desain arsitektur dari sistem baik itu desain topologi jaringan dan desain *database* yang digunakan. Isu dalam desain arsitektur perlu mendapat perhatian khusus dalam pengembangan sistem berbasis layanan. perlu diingat bahwa konsep sistem berbasis layanan erat kaitannya dengan teknologi *cloud*, karenanya ketika mendesain topologi jaringan, prinsip *user transparency* perlu dimasukkan sebagai variabel utama. *User transparency* memastikan *users* yang merupakan *tenant* pengguna jasa layanan SCM tidak mengetahui secara detail bagaimana fasilitas infrastruktur dari sistem ini. *Users* cukup mengetahui bagaimana cara penggunaan layanan dan bagaimana proses *customization* dapat dilakukan. Begitupun pada proses perancangan *database*, perlu diperhatikan bahwa sistem *database* harus mengakomodir konsep *multi-tenant* yang memungkinkan setiap *users* memiliki skema *database* sendiri yang unik. Namun demikian, faktor *cost* dalam pemilihan konsep *database* pun harus mendapat perhatian sehingga pengembang tidak perlu menginvestasikan terlalu banyak dana untuk fitur yang sebetulnya tidak diperlukan. Dua konsep

yang sudah dijelaskan dalam makalah ini yaitu *shared database* dengan *multi-tenant* dan *dedicated database* dengan *multi-tenant* dapat menjadi pertimbangan dalam melakukan perancangan.

7. Kesimpulan

Proses pengembangan sistem informasi yang didasarkan kepada metodologi pengembangan sistem informasi merupakan sebuah *best practice* yang walaupun tidak menjamin kesuksesan tetapi memberikan panduan dan arahan yang cukup lengkap mengenai tahapan-tahapan dalam pengembangan sistem. Proses pengembangan sistem informasi berbasis layanan memiliki kesamaan dengan pengembangan konvensional. Namun demikian, ada beberapa faktor yang perlu diperhatikan dengan lebih baik untuk mendapatkan hasil yang lebih maksimal dalam pengembangan sistem. pada makalah ini telah dipaparkan bagaimana proses pengembangan sistem informasi berbasis layanan (dibatasi sampai tahapan desain) dengan membahas isu-isu yang kemungkinan akan muncul dan perlu mendapat perhatian khusus.

REFERENCE

- [1] Simchi-Levi, E., dan Kaminsky, P. (2003). *Designing and Managing The Supply Chain: Concepts, Strategies, and Case Studies*. Irwin/McGraw-Hill.
- [2] Van der Vorst, J. (2004). *Supply Chain Management: Theory and Practices*. IN: CAMPS, T. et al. *The Emerging World of Chains and Networks: Bridging Theory and Practice*. Den Haag: Red Business Information, pp. 105-128.
- [3] Meindl, P., dan Chopra, S. (2005). *Supply Chain Management: Strategy, Planning, and Operation*. Pearson Education International/Prentice Hall.
- [4] Porter, M. (1998). *Competitive Advantage: Creating and Sustaining Superior Performance: With a New Introduction*. Free Pr.
- [5] Lambert, D., dan Cooper, M. (2000). “Issues in Supply Chain Management”. *Journal of Industrial Marketing Management*, vol. 29, pp. 65-83.
- [6] Chong, F., Carraro, G., dan Wolter, R. (2011). “Multi-Tenant Data Architecture”. [Online] tersedia di <http://msdn.microsoft.com/en-us/library/aa479086.aspx>