

PENGEMBANGAN WEB PENJUALAN & LAYANAN PRODUK MITRA BERBASIS SOA SEBAGAI PENYEDIA SERVICE MITRA (STUDI KASUS SOBATH COLLECTION & MITRA RAMAYANA - ROBINSON)

Kartini

Sistem Informasi Fakultas Ilmu Komputer Universitas Esa Unggul
Jalan Arjuna Utara no.9, Tol Tomang, Kebon Jeruk, Jakarta Barat 11510
kartinimusa2@gmail.com

Abstrak

Untuk mengembangkan aplikasi sesuai dengan teknologi informasi yang ada, sangat sulit dan mahal serta membutuhkan waktu yang lama dikarenakan adanya perbedaan platform antar aplikasi serta keterbatasan akses dalam server demi keamanan server itu sendiri. Sobath Collection memiliki banyak mitra yang menjual produknya. Sobath Collection ingin mengembangkan Web Penjualannya menjadi aplikasi yang mampu menyediakan produk bagi aplikasi web penjualan lain (yaitu web penjualan mitra mereka) dengan menggunakan SOA (*Service Oriented Architecture*). Dalam perancangan SOA pada aplikasi penjualan berbasis web menggunakan metode *extreme programming* yang terdiri dari tahap *Planning, Designing, Writing and Editing Code* serta *Testing*. Apabila hasil telah memenuhi sesuai yang diinginkan pengguna maka implementasi dan *maintenance*. Dengan SOA, aplikasi Sobath Collection akan mampu mengintegrasikan data antar aplikasi yang berbeda platform dan server dengan menyediakan service-service yang mengenkapsulasi proses-proses yang sudah tersedia.

Kata kunci: SOA, service-service, web,

Pendahuluan

Dunia usaha saat ini menjadikan informasi dan teknologi sebagai sarana penting dalam berjalannya kegiatan suatu usaha demi tercapainya tujuan. Perkembangan teknologi yang pesat memicu setiap perusahaan untuk ikut mengembangkan atau memperluas usahanya dengan menyediakan kemudahan-kemudahan bagi para konsumen dalam mengakses usaha yang dijalankan. Tidak memandang besar atau kecilnya usaha tersebut,

Aplikasi teknologi yang kembangkan tersebut diatas tentunya mampu memenuhi aspek *interoperability, multi platform*, dan dapat disatukan melalui proses bisnis. Namun, untuk membuat aplikasi yang memenuhi aspek-aspek tersebut membutuhkan biaya yang tidak murah. Selain itu, juga membutuhkan waktu yang tidak sedikit dan sulit, dikarenakan adanya perbedaan *platform* antar aplikasi, dan juga ada keterbatasan dari server yang membatasi akses dari luar demi keamanan.

SOA (*Service Oriented Architecture*) merupakan suatu pendekatan aplikasi yang dibangun secara modular atau berupa *service*. *Service* yang dibangun dapat digunakan

kembali komponen-komponen yang sudah ada sebelumnya sesuai dengan integrasi usaha yang ada. Dalam hal ini, komponen-komponen tersebut memberikan suatu jenis layanan *service* tertentu.

Sobath Collection merupakan sebuah toko eceran dan grosir yang melayani penjualan produk sepatu, sandal dan tas berbasis web. Untuk perkembangan usaha penjualan ke depan, Sobath Collection berencana untuk mengembangkan Web penjualannya, dimana Sobath Collection ingin menjadi penyedia produk bagi mitra-mitranya. Oleh sebab itu, Sobath Collection mengharapkan aplikasi yang digunakan mampu mengintegrasikan data dengan mitra melalui enkapsulasi proses-proses yang sudah ada.

Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka dapat dirumuskan permasalahan yang menjadi dasar, yaitu:

1. Bagaimana mengembangkan web penjualan dan layanan produk untuk mitra Sobath Collection?

2. Seberapa mahal dan sulitkah mengembangkan aplikasi penjualan & layanan produk pada Sobath Collection?
3. Seberapa Sulitkah mengintegrasikan data bisnis Sobath Collection dengan mitra.

Tujuan

1. Membangun *Service Oriented Architecture* (SOA) pada aplikasi penjualan berbasis web sebagai penyedia layanan Mitra.
2. Membangun aplikasi penjualan yang bisa dikembangkan dengan mudah dan murah.
3. Mengintegrasikan data antara Sobath Collection dengan mitra.

Metode Penelitian

Dalam tahap pengembangan sistem menggunakan dua Metode yaitu Studi Pustaka dan metode *extreme programming*

1. Metode Studi Pustaka dan *browsing* internet
Dalam metode ini penulis mempelajari dan mengumpulkan berbagai referensi buku-buku, jurnal-jurnal, *e-book* dan artikel lainnya serta studi banding website terkait di internet yang berhubungan dengan masalah penulisan.

2. Metode *extreme programming* ada beberapa *rules* yang harus diikuti:

a. *Tahap Planning*; Melibatkan beberapa hal:

- 1) Membuat daftar kebutuhan mitra atau fitur yang dibutuhkan oleh pengguna sistem.
- 2) Memperkirakan waktu yang dibutuhkan dalam proses *writing and editing code*.
- 3) Membagi *project system* kedalam beberapa bagian kecil.
- 4) Membuat rencana *release* aplikasi yang menjelaskan jadwal dari *project system* yang dikerjakan.

b. *Tahap Designing*; diantaranya:

- 1) *Design* harus sederhana dan interaktif sesuai yang diminta *user*, hanya melayani kebutuhan *user* saat ini.
- 2) Penamaan yang konsisten mengikuti dokumen yang sebelumnya digunakan untuk memberikan pemahaman yang lebih baik.

c. *Tahap Writing and Editing Code*

- 1) Sebelum memulai pengembangan web ini, membuat daftar test yang sesuai dengan requirement dan meminta pengguna memberikan *feedback* segera.
- 2) *Writing and editing code* harus menggunakan standard sehingga lebih mudah dipahami dan diadaptasi.
- 3) Semua dokumen harus ditinjau secara berurutan dan berkala.
- 4) Proses *development* harus dilakukan sesuai timeline tidak ada *overtime*.

d. *Tahap Testing*; Tahap ini harus mengikuti aturan :

1. Setiap dokumen yang dibuat harus terkait secara fungsional dengan dokumen sebelumnya.
2. Setiap dokumen yang akan diterbitkan hanya setelah semua test yang dibutuhkan selesai.
3. Tes tambahan untuk menguji kekurangan yang ada.

Penelitian terdahulu yang sudah dilakukan oleh:

1. Kurniadi Priguna, tahun 2010 yang berjudul "*Analisis Penggunaan Service Oriented Architecture Dalam Mengintegrasikan dan Mempercepat Proses-Proses Bisnis Perusahaan*". Hasil Penelitian ini dapat memperjelas pihak-pihak yang tertarik dengan SOA terhadap pengertian dan gambaran SOA, cara kerjanya, manfaat, dan peran SOA yang diimplementasikan dalam perusahaan Universitas Binus, Jakarta.
2. Haq Chairul Azizi, tahun 2012 yang berjudul "*Rancangan Bangun sistem Informasi apotek Terintegrasi menggunakan Service Oriented Architecture*", dan memberikan kontribusi pada pengguna mampu mendekatkan pihak apotek dengan pelanggan (CRM). UIN Sunan Kali Jaga, Jogjakarta.
3. Sarno Riyanarto, Sunaryono Dwi, Ventyana Gita, tahun 2012 yang berjudul "*Rancang Bangun Aplikasi Cash Bank dan Sales dengan Service Oriented Architecture pada Platform Java*," aplikasi ini dapat diintegrasikan dengan aplikasi-aplikasi lain yang mewakili berbagai

departemen dalam sebuah perusahaan, sehingga terbentuk sebuah sistem ERP, ITS, Surabaya.

SOA (Service Oriented Architecture)

Pengertian SOA

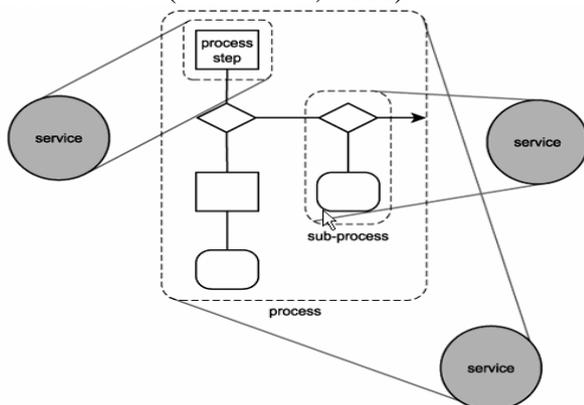
SOA adalah sebuah bentuk teknologi arsitektur yang mengikuti prinsip-prinsip *service-orientation* atau berorientasi *service* (Brahmasta, 2008).

SOA merupakan sekumpulan layanan (*service*) yang berkomunikasi satu sama lain untuk memenuhi proses bisnis tertentu. SOA juga merupakan sebuah strategi yang populer untuk memberikan layanan *web*. SOA juga merupakan sebuah teknologi yang menyediakan *service-service* yang dapat digunakan kembali dan menggunakan basis XML sehingga aplikasi ini mudah untuk dikembangkan meskipun aplikasi yang berbeda *platform* (Alaedin, 2012)

Konsep Service Oriented

Service-oriented merupakan sebuah pendekatan dalam penyelesaian masalah besar dengan membaginya menjadi sekumpulan *service* (layanan) kecil yang menyelesaikan permasalahan spesifik (Brahmasta, 2008; Alaedin, 2012).

Lingkup dari *service* tidak terbatas, *service* dapat mengenkapsulasi sebuah proses besar atau hanya sebuah proses kecil. Hal ini sesuai dengan kebutuhan. Setelah seluruh permasalahan dibagi dalam beberapa *service*. Ada beberapa permasalahan yang harus dimiliki oleh *service*, yaitu cara *service* berhubungan, cara *service* berkomunikasi, cara *service* didesain, dan cara pesan antar *service* didefinisikan (Brahmasta, 2008).



Gambar 1

Encapsulasi business process dengan service

Sifat-Sifat Service Oriented

Ada beberapa-beberapa sifat yang dimiliki oleh *service oriented*, yaitu: (Alaedin, 2012).

1. *Loosely coupled*, yaitu setiap *service* berdiri sendiri secara independen dan tidak bergantung pada *service* lain untuk berjalan.
2. *Service contract*, yaitu setiap *service* memiliki kesepakatan mengenai cara untuk komunikasi.
3. *Autonomy*, yaitu *service* memiliki hak penuh terhadap semua logik yang dienkapsulasi.
4. *Abstraction*, yaitu *service* tidak memperlihatkan bagaimana logik diimplementasi di dalamnya.
5. *Reusability*, yaitu logik dibagi menjadi sekumpulan *service* yang dapat memudahkan penggunaan kembali.
6. *Statelessness*, yaitu *service* tidak memiliki status tertentu terkait dengan aktivitas yang dilakukannya.
7. *Discoverability*, yaitu *service* didesain untuk deskriptif sehingga bisa ditemukan dan diakses melalui mekanisme pencarian tertentu.

Prinsip-Prinsip SOA

Ada beberapa prinsip yang mendefinisikan aturan-aturan dasar dalam pengembangan, perawatan dan penggunaan dari SOA, yaitu (Stenly, 2008; Joko, 2010).

1. Konsep utama dari SOA adalah layanan.
2. Setiap layanan didefinisikan dengan sebuah kontrak yang formal.
3. Layanan-layanan hanya berinteraksi dengan layanan yang lain melalui antarmuka kontrak yang telah didefinisikan terlebih dahulu.
4. Layanan-layanan harus dapat diakses melalui standard teknologi yang tersedia pada lingkungan secara umum. Mekanisme-mekanisme yang digunakan harus dapat diterima oleh standard-standard industri.
5. Layanan-layanan harus dapat didefinisikan kedalam level abstraksi yang tinggi yang berhubungan aktifitas-aktifitas pada dunia nyata dan fungsi-fungsi bisnis dapat di selaraskan dengan tepat.

6. Layanan-layanan yang tersedia harus memiliki arti yang penuh atau mudah dipahami.
7. Layanan-layanan harus bersifat *loosely coupled*.
8. Kumpulan layanan harus memiliki tipe dokumen yang sama, yaitu dokumen XML. Hal ini untuk memfasilitasi pertukaran informasi diantara layanan-layanan.
9. Layanan-layanan harus memberikan tugas-tugas yang spesifik dan menyediakan antarmuka yang sederhana untuk mengakses atau menggunakan fungsionalitas yang disediakan.
10. Layanan-layanan harus menyediakan informasi yang menjelaskan kemampuan dan keterbatasan dari layanan yang disediakan. Informasi tersebut harus tersedia pada *repository*.

Komponen-Komponen SOA

Kunci dari pembangunan arsitektur berbasis layanan (*service*) adalah komponen-komponen yang digunakan untuk menyusun SOA. Ada beberapa komponen yang terlibat dalam arsitektur berbasis layanan(*service*) adalah: (Stenly, 2008; Joko, 2010).

1. *Services*

Layanan dapat didefinisikan sebagai interface untuk menerima request dari pengguna layanan dan kemudian memberikan respon terhadap request tersebut. SOA memiliki tiga karakteristik, yaitu:

- a. *Contract Interface*. Kontrak antarmuka layanan adalah bersifat *platform independent*.
- b. *Dinamically found and assessment*. Layanan secara dinamis dapat ditemukan dan digunakan.
- c. *Self Contained*. Setiap layanan dibangun secara independen. Layanan memiliki *business logic* dan definisi yang tidak bergantung terhadap layanan yang lain.

2. *Message*

Penyedia layanan dan pemakai layanan berkomunikasi melalui pesan. Layanan-layanan menggunakan kontrak antarmuka, yang mendefinisikan sifat layanan dan pesan yang diterima dan dikembalikan. Untuk mendukung kontrak antarmuka yang bersifat

independen, teknologi yang digunakan harus mampu mendefinisikan pesan yang dikirim oleh *platform* atau bahasa yang lain. Oleh karena itu, pesan secara tipikal dibangun menggunakan dokumen-dokumen XML.

3. *Dynamic Discovery*

Dynamic discovery merupakan sebuah bagian penting dari SOA. Pada level tinggi, SOA dibentuk oleh tiga bagian penting, yaitu penyedia layanan, pengguna layanan, dan direktori layanan. Direktori layanan adalah sebuah penghubung antara penyedia layanan dan pengguna layanan. penyedia layanan mendaftarkan layanan pada direktori layanan dan pengguna layanan meminta atau mencari layanan dari direktori layanan. Sebagian besar direktori layanan secara umum diatur berdasarkan kriteria dan kategori.

UDDI (*Universal Deskripsi Discovery and Integration*)

UDDI adalah *registry* berbasis XML yang bersifat *plafform-independent*. UDDI memungkinkan sebuah bisnis untuk mempublikasikan *service* mereka sehingga aplikasi mereka dapat digunakan oleh aplikasi lainnya." Dalam aplikasi yang menggunakan SOA, UDDI merupakan *registry* yang berisi *service-service* yang bisa dipanggil oleh aplikasi-aplikasi lain dan *service-service* dalam UDDI ini akan dipanggil melalui SOAP yang tersedia pada aplikasi saat aplikasi menggunakan SOA (Youssef, 2012),

Analisa Produk dan proses bisnis Sobath Collection saat ini.

Merek-merek produk yang dijual pada Sobath *Collection*, dapat dilihat pada tabel 1 dibawah ini.

Tabel 1
Merek Produk

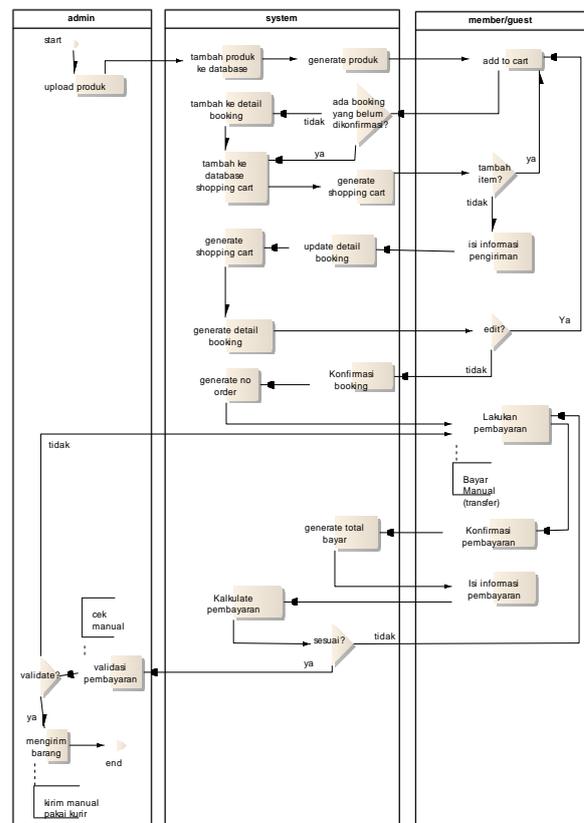
Gambar	Merek	Keterangan
	Grutty	Menyediakan berbagai jenis sepatu, sandal, tas
	Carwil	Menyediakan Berbagai jenis sepatu, sandal, tas.
	Homyped	Menyediakan Berbagai jenis sepatu dan sandal.
	Ardiles	Menyediakan Berbagai jenis sepatu dan sandal.
	Ando	Menyediakan Berbagai jenis sepatu dan sandal.
	Pakalolo	Menyediakan Berbagai jenis sepatu dan sandal.
	Pro ATT	Menyediakan Berbagai jenis sepatu dan sandal.
	Palazzo	Menyediakan Berbagai jenis tas.

Dan gambar dibawah ini adalah Activity Diagram Proses bisnis yang sedang berjalan pada Sobath Collection saat ini, dimana :

1. *Member/Guest* mengakses halaman produk, dan sistem akan menampilkan produk kepada *Member/Guest*.
2. *Member/guest* memilih serta menambah item di daftar belanja dan jika belum ada *booking-an* sebelumnya, sistem akan membuat *booking* yang baru setelah itu sistem akan menambah *shopping cart* baru.
3. *Member/guest* mengakses ke proses selanjutnya dengan mengakses dan mengisi data pengiriman di halaman pengiriman dan sistem akan menampilkan data *member/guest* yang sudah ada dan sistem akan mengupdate informasi *booking* setelah *member/guest* mengakses proses selanjutnya.
4. *Member/guest* menuju ke proses selanjutnya dengan mengakses halaman konfirmasi *booking* untuk mengecek kembali informasi belanja dan informasi pengiriman serta *booking-an* dikonfirmasi setelah *member/guest* mengakses proses selanjutnya dan *member/guest* akan mendapatkan nomor *order* yang bisa digunakan untuk mengkonfirmasi pembayaran.
5. *Member/guest* melakukan pembayaran secara manual setelah itu melakukan konfirmasi pembayaran melalui website, dengan menginput nomor *order* maka

sistem akan mengenerate total bayar sesuai dengan nomor *order*.

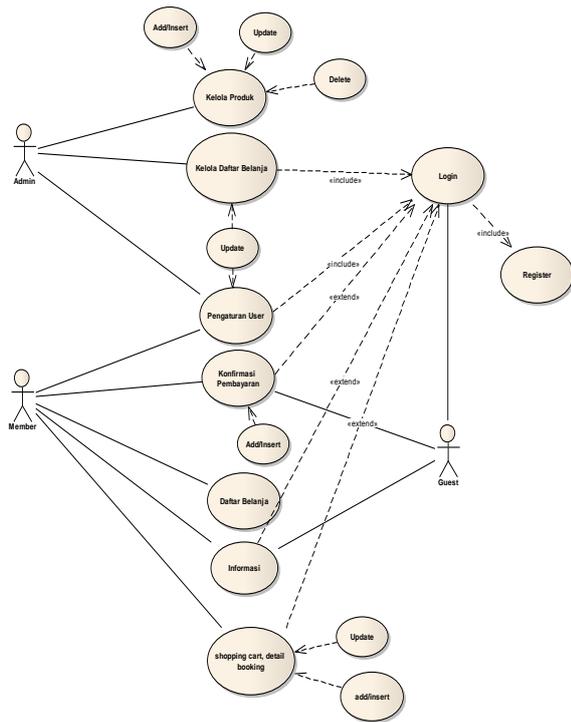
6. *Member/guest* mengisi informasi pembayaran dan sistem akan mengkalkulasi pembayaran dan total bayar jika tidak sesuai *member/guest* akan melakukan pembayaran kembali.
7. Admin melakukan validasi pembayaran yang dilakukan oleh *member/guest* jika tidak sesuai *member/guest* akan melakukan pembayaran kembali dan jika sesuai maka admin akan melakukan pengiriman barang melalui kurir.



Gambar 2
Proses bisnis yang berjalan

Rancangan usulan Proses bisnis Sobath Collection dan Mitra

Use case diagram Gambar 3 dibawah ini, menjelaskan fungsi dari user:



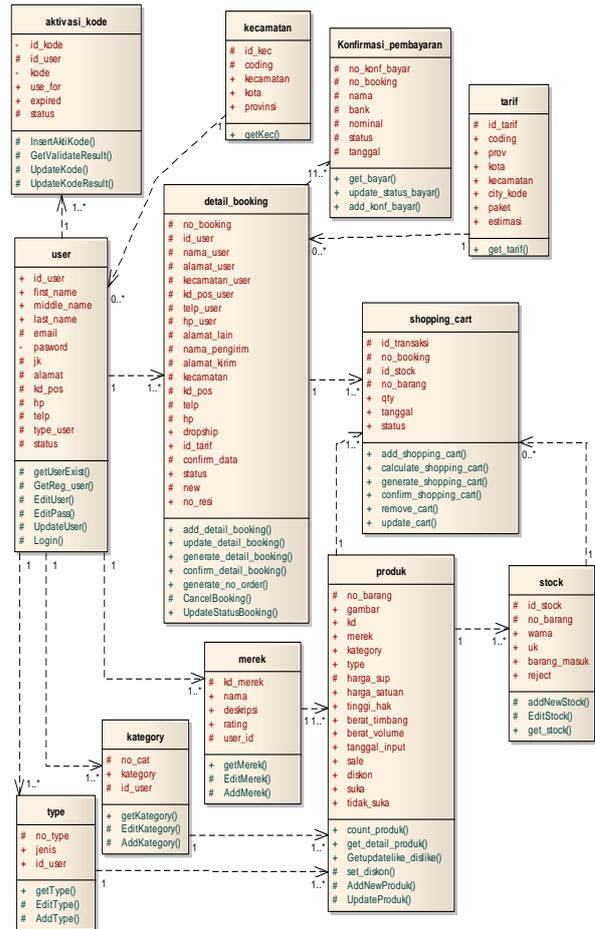
Gambar 3
Proses User (pelanggan & mitra) Sobath Collection

Ada 3 jenis pengguna (*user*) yang berinteraksi dengan aplikasi Sobath Collection adalah :

1. Admin merupakan administrator dari aplikasi. Bertugas untuk mengelola aplikasi.
2. *Member* merupakan pengunjung (konsumen) yang sudah terdaftar pada aplikasi. bisa melakukan pembelian dan dapat melihat semua transaksi yang pernah dilakukan tanpa harus mengingat *no order*.
3. *Guest* merupakan pengunjung (konsumen) yang belum terdaftar pada aplikasi. *Guest* dapat melakukan pembelian melalui aplikasi baik aplikasi penjualan Sobath Collection maupun *Reseller guest* hanya dapat melihat daftar transaksi dan konfirmasi pembayaran sesuai dengan *no order* yang diingatnya dengan sistem *search* yang disediakan pada aplikasi

Rancangan usulan Proses Pengolahan Data Sobath Collection.

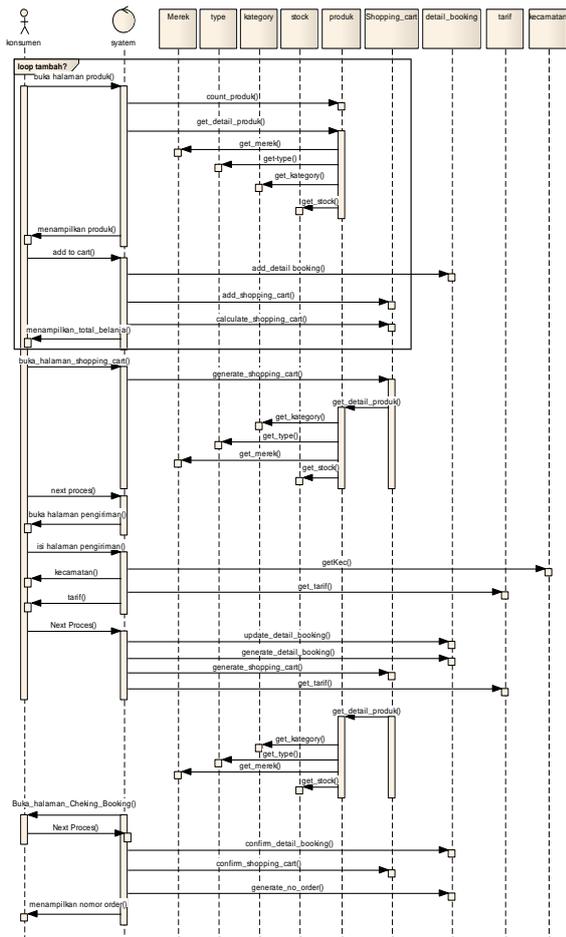
Menjelaskan bagaimana komunikasi antara data pada Sobath Collection sehingga menghasilkan informasi-informasi yang digunakan pada aplikasi Sobath Collection.



Gambar 4
Rancangan Proses Pengolahan Data Sobath Collection

Rancangan Proses Transaksi Penjualan Sobath Collection.

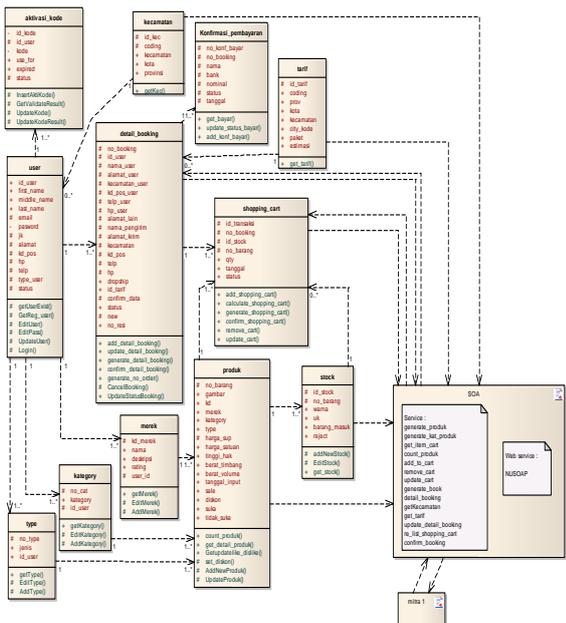
Untuk proses transaksi penjualan pada Sobath Collection digambarkan dengan menggunakan *Sequence Diagram*.



Gambar 5

Rancangan Proses Transaksi Penjualan Sobath Collection

Rancangan Proses Integrasi database Sobath Collection dengan mitra

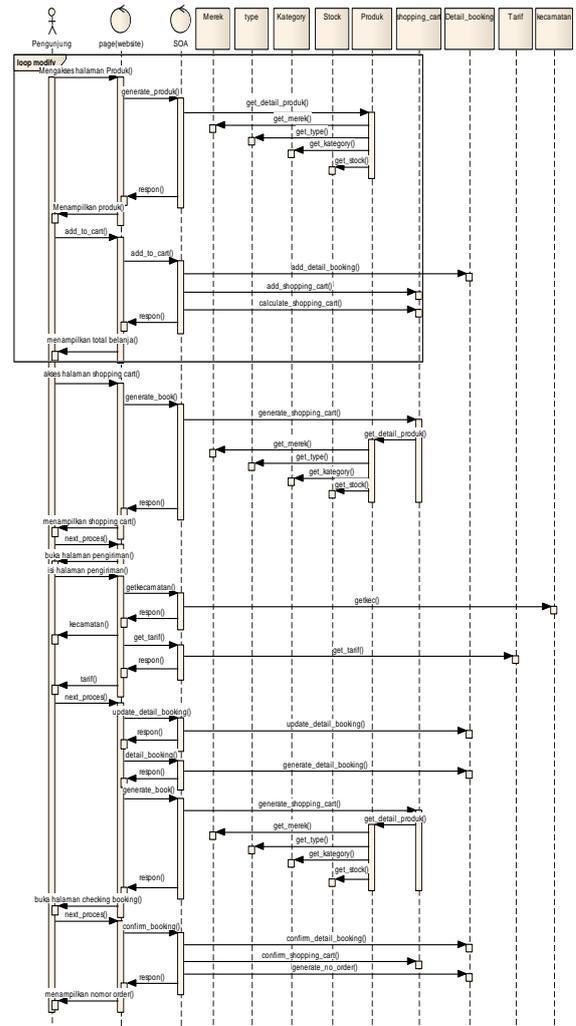


Gambar 6

Rancangan Proses Integrasi Database Dengan Mitra Melalui SOA pada Sobath Collection

Proses ini bertujuan untuk menggambarkan komunikasi data antara Sobath Collection dengan mitra melalui *service-service* yang disediakan dengan menggunakan SOA dari database yang sudah ada pada Sobath Collection

Rancangan Proses transaksi penjualan pada aplikasi mitra melalui SOA



Gambar 7

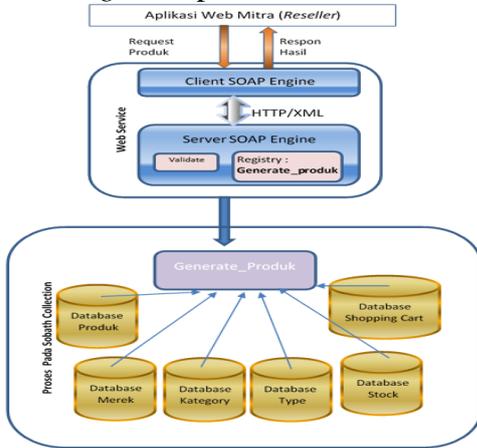
Rancangan Proses Transaksi Penjualan Pada Aplikasi Mitra Melalui SOA

Proses ini digambarkan dengan *Sequence Diagram* yang bertujuan untuk menggambarkan proses transaksi penjualan melalui SOA sehingga dapat berkomunikasi dengan aplikasi Sobath Collection saat pengunjung melakukan transaksi melalui aplikasi mitra.

Service generate_produk

Pada desain SOA ini menggambarkan bagaimana membentuk *service-service* yang akan digunakan untuk berinteraksi dengan

mitra (*reseller*) melalui *webservice* dengan menggunakan proses-proses yang sudah ada pada aplikasi Sobath Collection. Desain SOA ini juga bertujuan untuk menjelaskan bagaimana aplikasi mitra mendapatkan data produk dan stok terbaru, *shopping cart*, serta data *detail booking* dari aplikasi Sobath Collection

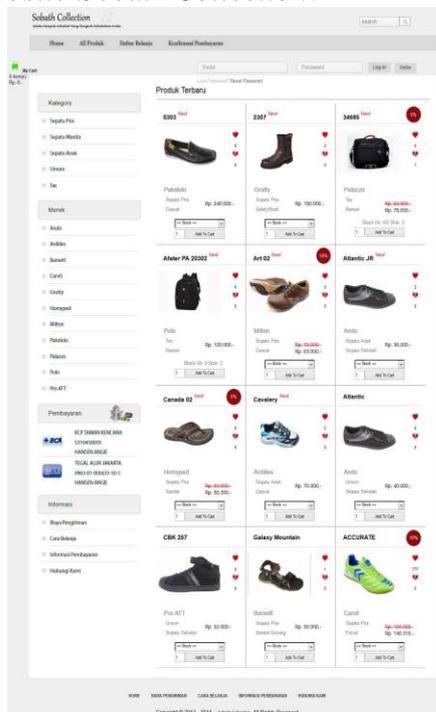


Gambar 8

Desain *service get_item_cart* pada SOA

Website Sobath Collection.

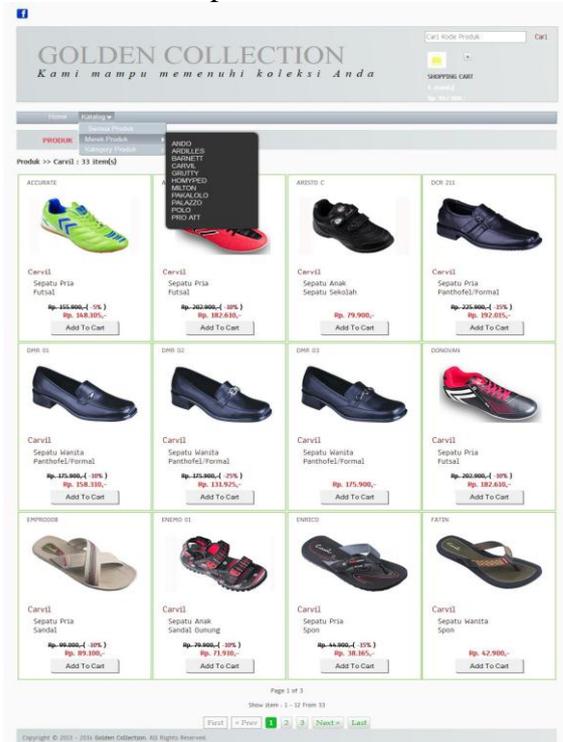
Toko penjualan berbasis web pada Sobath Collection dapat diakses di <http://sobath-market.com> dan berikut adalah tampilan depan dari *website* Sobath Collection.



Gambar 9

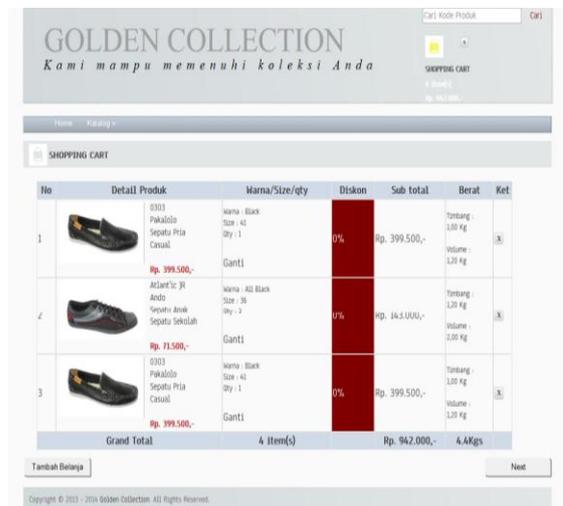
Tampilan depan *website* (Home) Sobath Collection

Tampilan produk yang masih tersedia sesuai category, merek terbaru yang dipilih oleh pengunjung serta pencarian produk berdasarkan kode produk



Gambar 10

Tampilan Halaman Produk dengan SOA

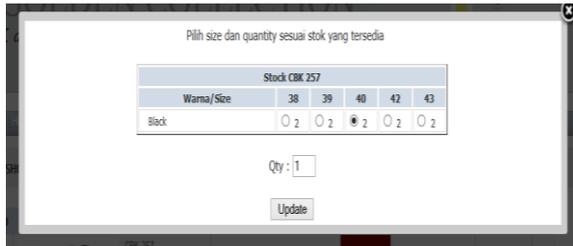


Gambar 11

Tampilan Halaman *Shopping Cart* dengan SOA



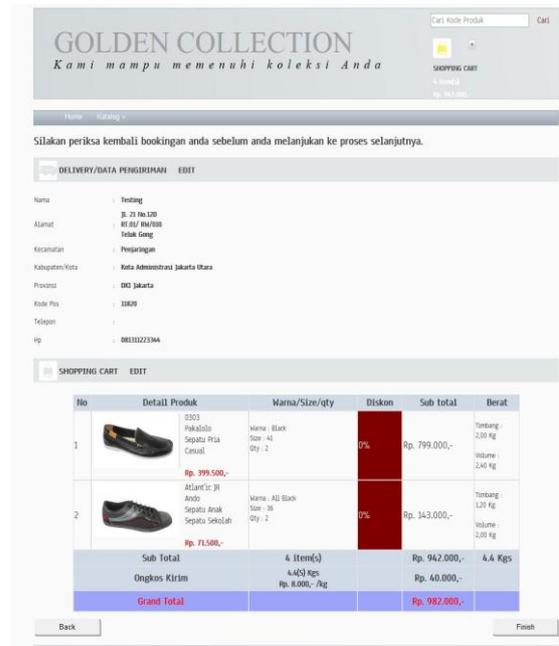
Gambar 12
Tampilan Stok (Halaman Depan dan Produk) dengan SOA



Gambar 13
Tampilan Stok (Halaman Shopping Cart) dengan SOA



Gambar 14
Tampilan Halaman Delivery Dengan SOA



Gambar 15
Tampilan Halaman Konfirmasi Booking dengan SOA



Gambar 16
Tampilan Halaman Finish Dengan SOA



Gambar 17
Tampilan Daftar Belanja Aplikasi Sobath Collection

Gambar 18

Tampilan Konfirmasi Pembayaran Aplikasi Sobath Collection

Kesimpulan

Model *Waterfall* memiliki tahapan-tahapan yang jelas, sehingga mendapatkan daur hidup sistem dengan baik mulai dari analisis hingga pemeliharaan system. Dengan adanya SOA pada aplikasi penjualan Sobath Collection yang mampu mengintegrasikan data dengan mitra (*reseller*) sehingga memudahkan pembaharuan (*update*) informasi baik informasi produk ataupun informasi lainnya yang sudah terhubung dengan SOA karena tidak memerlukan aktifitas yang berulang-ulang untuk meng-*update* informasi pada aplikasi Sobath Collection dan aplikasi mitra (*Reseller*). Dengan adanya SOA pada aplikasi penjualan Sobath Collection akan memberikan kemudahan dalam pengembangan aplikasi dalam *platform* yang berbeda serta biaya yang lebih murah di kemudian hari.

Daftar Pustaka

- Adipradana, Brahmasta. (2008). *Penerapan Business Process Management Dalam Service-Oriented Architecture*. Bandung: Institut Teknologi Bandung.
- Bassil, Youssef. (Februari 2012). *Service Oriented Architecture For Weaponry and Battle Command and Control Systems in warfighting*. *International Journal of Information and Communication Technology Research*. 2(2), 185-196
- Christian, Joko. (September 2010). Model Data Warehouse Dengan *Service Oriented Architecture* Untuk Menunjang Sistem Informasi Eksekutif. *Jurnal TELEMATIKA MKOM*. 2(2), 113-105.

Karitari, Alaedin, dkk. (Oktober 2012). A Service Oriented Security Reference Architecture. *International Journal of Advanced Computer Science and Information Technology (IJACSIT)*. 1(1), 24-31.

Madcoms. (2009). *Aplikasi Web Database dengan Dreamweaver dan PHP-MySQL*, Yogyakarta: Andi.

Pungus, Stenly R. (2008). *Penerapan Service Oriented Architecture Untuk Pengintegrasian Sisten Informasi Perguruan Tinggi*. Bandung: Institut Teknologi Bandung.