

**PENGUNAAN PARADIGMA SOA (SERVICE ORIENTED ARCHITECTURE)
UNTUK MEREALISASIKAN INTEROPERABILITAS
DAN INTEGRITAS SISTEM INFORMASI**

Rini Astuti

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI

Jl. Ir. H. Juanda 96 Bandung 40132

e-mail: riniastuti@likmi.ac.id

ABSTRAK

Permasalahan umum teknologi informasi atau sistem informasi yang dihadapi dalam suatu organisasi adalah bagaimana merealisasikan interoperabilitas dan integrasi di satu sisi, dengan fleksibilitas penyelenggaraan layanan di sisi lain. Kebutuhan tersebut dapat diakomodir oleh paradigma SOA (*Service Oriented Architecture*), arsitektur berorientasi layanan).

SOA membagi fungsi-fungsi menjadi unit-unit yang berbeda (layanan), yang dapat didistribusikan melalui suatu jaringan dan dikombinasikan serta digunakan ulang untuk membentuk aplikasi bisnis. Layanan-layanan ini saling berkomunikasi dengan mempertukarkan data antar mereka atau dengan mengkoordinasikan aktivitas antara dua atau lebih layanan.

Implementasi SOA secara umum didasarkan pada potensi untuk tetap memakai sistem lama (*legacy*) (jika menurut analisa strategis dan risiko masih dirasakan relevansinya) dengan cara mengembangkan layanan (*service*) berdasarkan sistem lama tersebut dan kemudian mengembangkan sistem baru dengan menyusun dan mengkombinasikan layanan-layanan (*service-service*) yang ada. Kemampuan untuk menyusun dan mengelola layanan (*service*) yang ada secara *enterprise wide* merupakan kunci keberhasilan implementasi SOA.

Kata-kata kunci: interoperabilitas, SOA

1. PENDAHULUAN

Arsitektur berorientasi layanan (SOA) adalah sebuah paradigma untuk mengorganisir dan menggunakan kemampuan terdistribusi yang mungkin di bawah kendali kepemilikan *domain* yang berbeda dan diimplementasikan dengan menggunakan berbagai teknologi. Secara umum, entitas (orang dan organisasi) menciptakan kemampuan untuk memecahkan atau mendukung solusi bagi permasalahan yang dihadapi dalam perjalanan bisnis. Hal ini wajar untuk memikirkan kebutuhan satu orang dipenuhi oleh kemampuan yang ditawarkan oleh orang lain, atau dalam dunia komputasi terdistribusi, persyaratan satu komputer agen terpenuhi oleh agen komputer milik pemilik yang berbeda. Istilah pemilik di sini dapat digunakan untuk menunjukkan divisi yang berbeda dari satu bisnis atau mungkin entitas yang tidak terkait.

Tidak perlu selalu terjadi korelasi satu-ke-satu antara kebutuhan dan kemampuan, kebutuhan dan kemampuan bervariasi dari dasar sampai yang kompleks, dan kebutuhan apa pun yang diberikan mungkin membutuhkan suatu kombinasi dari kemampuan lain, bahkan mungkin terdapat kemampuan tunggal yang dapat mengatasi lebih dari satu kebutuhan. Satu nilai yang dirasakan dari SOA adalah bahwa SOA dapat menyediakan kerangka kerja yang kuat untuk penyesuaian kebutuhan dan kemampuan serta dapat menggabungkan kemampuan untuk mengatasi kebutuhan tersebut dengan memanfaatkan kemampuan lainnya. Salah satu kemampuannya adalah dapat menerima berbagai format *platform* di banyak kebutuhan.

SOA adalah sebuah "pandangan" arsitektur yang berfokus pada layanan dalam batas-batas tertentu sebagai tindakan antara kebutuhan dan kemampuan dengan cara yang kondusif untuk menemukan suatu layanan yang tepat dari berbagai platform.

SOA memiliki kemampuan interoperabilitas yaitu kemampuan dari suatu produk perangkat lunak atau sistem untuk berinteraksi dan berfungsi dengan sistem lain di saat ini atau di masa mendatang tanpa batasan akses atau implementasi.

2. KEBUTUHAN UNTUK SISTEM DENGAN SOA

Sebuah SOA adalah pendekatan teknologi informasi atau strategi yang menggunakan atau mengandalkan aplikasi dengan layanan yang tersedia dalam jaringan seperti *World Wide Web*. Penerapan arsitektur berorientasi layanan dapat

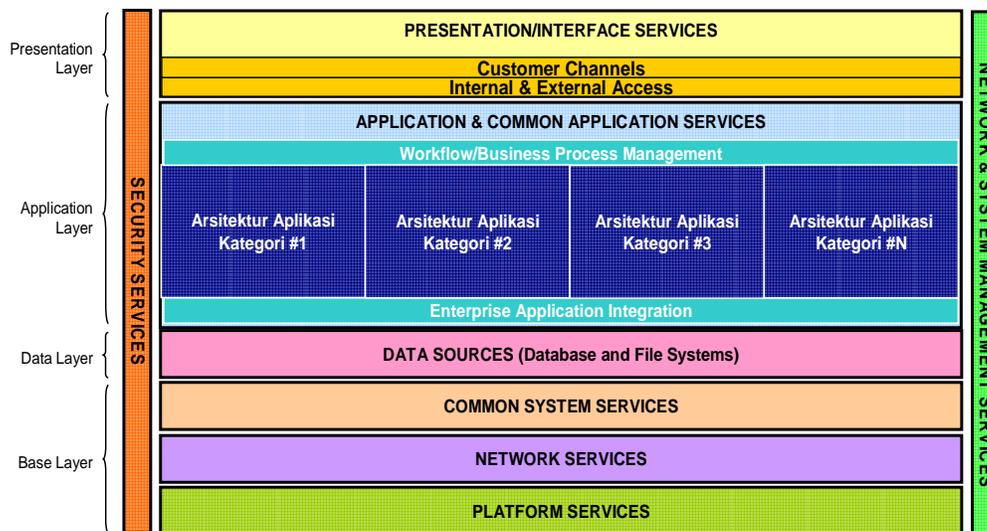
melibatkan pengembangan aplikasi yang menggunakan layanan, membangun aplikasi yang tersedia seolah sebagai layanan sehingga aplikasi lain dapat menggunakan layanan-layanan tersebut.

Dengan demikian arsitektur sistem yang sudah ada mungkin dibutuhkan untuk mengimplementasikan sistem tersebut atau migrasi ke sebuah SOA.

Arsitektur sistem memberikan deskripsi tentang lapisan-lapisan logikal yang membentuk sistem teknologi informasi secara keseluruhan, yaitu lapisan presentasi, lapisan aplikasi, lapisan data dan lapisan dasar. Cara pandang ini akan membantu dalam mengalokasikan sumber dayanya, sekaligus sebagai panduan dalam menetapkan standarisasi teknologi melalui penyediaan *framework-framework* sebagai model untuk setiap lapisan secara spesifik.

Sebagai panduan dalam menetapkan standarisasi teknologi, arsitektur sistem ini harus selalu dirujuk dalam pengadaan teknologi atau sistem informasi baru, yaitu ketika menyusun rencana detail untuk setiap blok dalam arsitektur sistem atau keterhubungannya dalam model secara keseluruhan.

Model arsitektur sistem generik yang akan digunakan dalam suatu organisasi dapat diperlihatkan oleh gambar di bawah ini.



Gambar 1. Model arsitektur sistem generik di suatu organisasi

Pada gambar tersebut, kategori aplikasi dalam *Application Layer* merupakan aplikasi-aplikasi yang nantinya akan didefinisikan dalam penentuan Arsitektur Aplikasi.

Berikut ini adalah deskripsi umum untuk setiap lapisan yang menyusun

Arsitektur Sistem. Detail penjelasan untuk setiap layer akan disampaikan di sub-bab berikutnya:

No	Lapisan	Deskripsi Umum
1.	Presentation Layer	Layanan ini diperuntukkan bagi pengelolaan akses atas layanan TI. Akses harus dapat dibedakan berdasarkan orang yang melakukan akses, haknya masing-masing, saluran akses yang digunakan serta tipe aksesnya. Tipe-tipe akses bisa saja internal maupun eksternal. Berdasarkan parameter-parameter ini, layanan presentasi atau antarmuka akan menentukan tipe akses yang bisa dan akan disediakan.
2.	Application Layer	Lapisan ini mengelola aplikasi apa saja yang akan membangun sistem TI, dibagi dalam klasifikasi: <ol style="list-style-type: none"> a. Aplikasi-aplikasi yang langsung terkait dengan transaksi operasional, monitoring dan pelaporan proses-proses bisnis. b. <i>Platform</i> integrasi yang memungkinkan komunikasi antar aplikasi.
3.	Data Layer	Lapisan ini mengelola data-data perusahaan yang secara umum dapat dibagi dalam klasifikasi: <ol style="list-style-type: none"> a. Data operasional, digunakan sehari-hari oleh operasi bisnis. Data ini merupakan data biasa yang diekstraksi dari basis data relasional. b. Data terkonsolidasi, data ini berasal dari data operasional yang dibersihkan, diringkas serta dikelompokkan untuk kepentingan proses pengambilan keputusan dan analisa. c. Data tak terstruktur, yaitu semua data yang terkandung dalam pesan <i>e-mail</i>, dokumen, lembar presentasi bahkan yang berada dalam format audio dan video. <i>Document management, file</i> dan <i>messaging server</i> merupakan solusi manajemen yang umum dalam area ini. d. Data referensi, data ini digunakan oleh aplikasi untuk otentikasi, konfigurasi serta keperluan perbandingan.

Beberapa keuntungan yang umumnya dicapai dengan implementasi SOA:

1. Efisiensi (*efficiency*)

Melakukan transformasi proses bisnis yang bersifat terkelompok atau pulau-pulau yang independen, replikasi menjadi *service* yang dipakai bersama dan lebih mudah dan murah untuk dikelola.

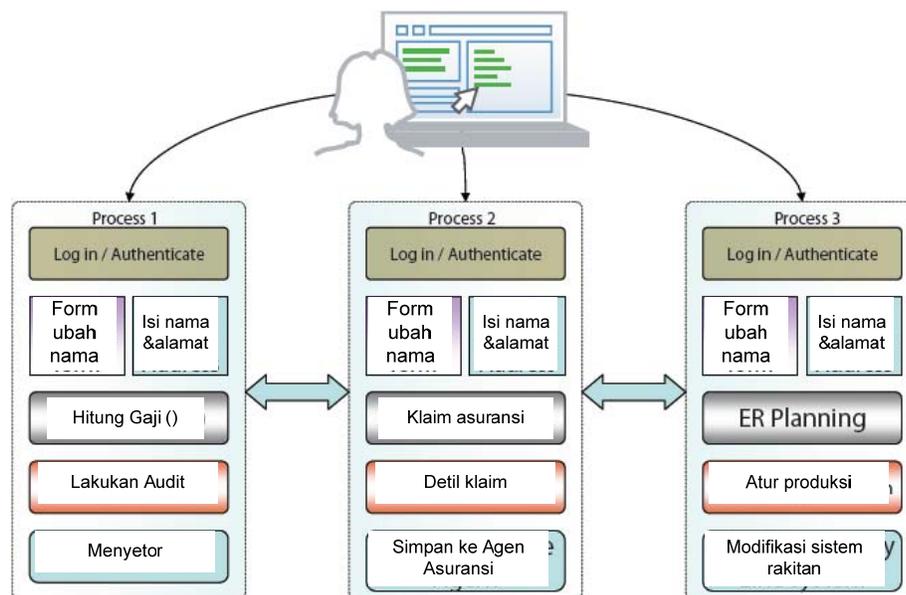
2. Responsif (*responsiveness*)

Adaptasi yang cepat dan kemampuan menghasilkan *service* baru yang relatif lebih cepat merupakan kunci untuk industri telekomunikasi untuk memenuhi kebutuhan pasar akan produk dan program yang inovatif.

- a. *Adaptability*: Rollout perubahan yang relative lebih cepat dan efektif dengan kompleksitas yang minimal dan menghemat waktu, biaya dan sumber daya.
- b. *Reduced Complexity*: Kompleksitas berkurang dengan adanya standar pembuatan *service*.
- c. *Increased Reuse*: Layanan yang ada dapat digunakan kembali untuk aplikasi baru.
- d. *Legacy Integration*: Aplikasi lama dapat dimanfaatkan kembali dengan mengekspose sebagai layanan baru.

3. PERANCANGAN DAN IMPLEMENTASI

Secara umum implementasi SOA didasarkan pada potensi sistem yang akan tetap dipertahankan. Gambar di bawah memperlihatkan sebuah contoh sistem informasi yang memungkinkan untuk migrasi ke SOA.

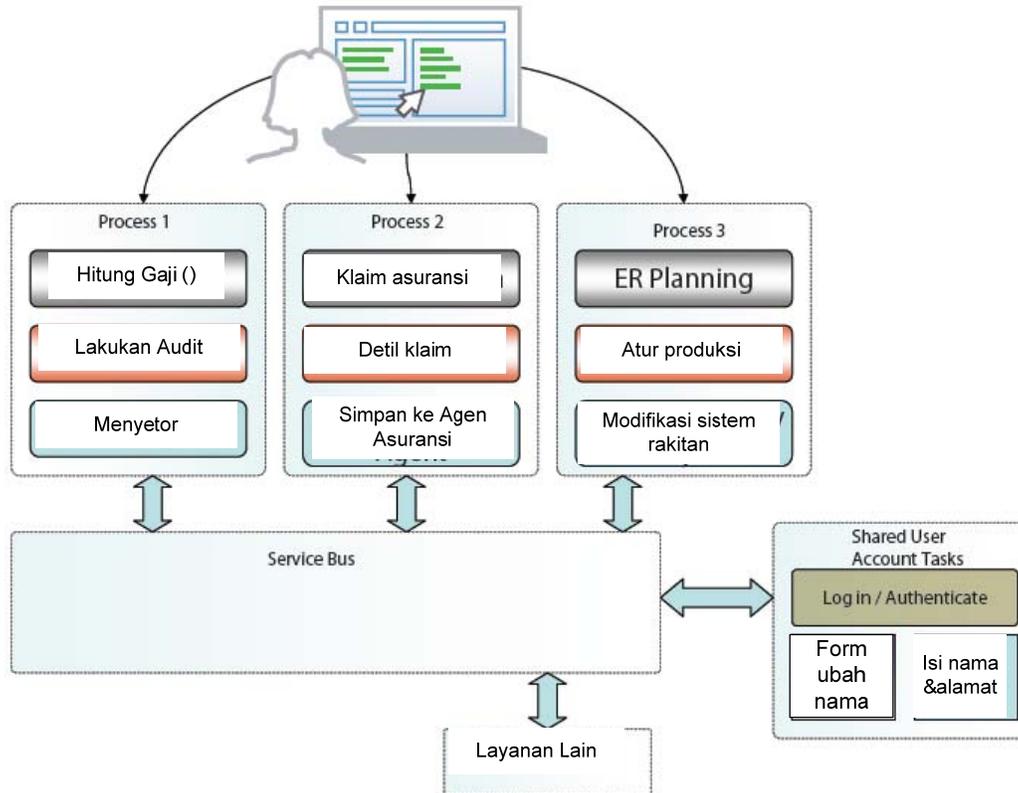


Gambar 2. Model proses bisnis dalam suatu sistem informasi organisasi

Dalam satu organisasi, mungkin terdapat tiga proses bisnis yang terpisah dengan menggunakan fungsi yang sama, masing-masing terenkapsulasi dalam aplikasi. Dalam skenario ini adalah fungsi login, kemampuan untuk mengubah nama pengguna, dan kemampuan untuk tetap dengan data yang ada adalah tugas yang dilaksanakan dalam ketiga proses tersebut. Hal ini adalah situasi yang memang diterapkan karena merupakan kebutuhan sistem organisasi ketika identifikasi kebutuhan dilakukan.

Skenario seperti ini sangat tidak efisien dan akan menambah kompleksitas dalam pemeliharaan sistem. Sebagai contoh, bila sebuah implementasi yang tidak mensinkronkan status pengguna di ketiga proses. Dalam lingkungan ini pengguna mungkin perlu mengingat *multiple login username & password* dan mengelola perubahan profinya di tiga wilayah terpisah. Selain itu, jika seorang manajer ingin menolak akses pengguna ke tiga proses, kemungkinan bahwa tiga prosedur yang berbeda akan diperlukan (satu untuk masing-masing aplikasi). Artinya penanggung jawab sistem teknologi informasi organisasi akan mengelola sistem tersebut secara efektif tiga kali lipat baik untuk perangkat keras dan perangkat lunaknya.

Dalam skenario yang lebih efisien, tugas-tugas umum akan dibagi di semua tiga proses. Hal ini dapat dilaksanakan oleh fungsi yang dikolaborasikan dari setiap proses atau aplikasi yang ada dan membangun otentikasi mandiri serta aplikasi manajemen pemakai (*user*) yang dapat diakses sebagai layanan. Dalam skenario seperti itu, layanan tersebut dapat dikonversikan di beberapa proses dan aplikasinya serta organisasi harus mempertahankan fungsi tersebut di satu pusat tempat. Alternatif solusi infrastruktur yang dihasilkan dapat terlihat seperti pada berikut:



Gambar 3. Alternatif model solusi beberapa proses bisnis dalam suatu layanan

Dalam alternatif model solusi tersebut, tugas-tugas akun pengguna yang terbagi telah dipisahkan dari setiap proses dan dilaksanakan dengan cara yang memungkinkan proses-proses lainnya disebut sebagai sebuah layanan. Hal ini memungkinkan fungsi bersama yang akan dikonversi dengan kolaborasi di ketiga proses. Layanan *bus* umum (berupa suatu fungsi) adalah benar-benar sebuah lingkungan virtual dimana layanan yang dibuat tersedia untuk semua pengguna potensial pada struktur tertentu. Hal ini biasanya disebut sebagai *Enterprise Service Bus* (ESB) yaitu sebuah model arsitektur perangkat lunak yang digunakan untuk merancang dan melaksanakan layanan interaksi dan komunikasi antara aplikasi perangkat lunak yang saling berinteraksi. ESB memiliki koleksi subkomponen khusus termasuk penamaan direktori dan pencarian, registri-repositori, dan *interface* serta kumpulan standar-standar dan protokol untuk membuat komunikasi tanpa batas di semua perangkat yang terhubung.

Dalam implementasi SOA, mungkin terjadi komplikasi namun relatif kecil dari seluruh infrastruktur yang ada di dalam satu domain. Pada kenyataannya, SOA suatu perusahaan akan jauh lebih sulit karena layanan dapat digunakan di beberapa domain kepemilikan. Untuk membuat interaksi yang mungkin, suatu mekanisme harus dibuat untuk menyampaikan semantik data, menyatakan dan menegakkan kebijakan dan kontrak, kemampuan untuk menggunakan batasan data yang diberikan di dalam dan keluar dari layanan serta ketentuan untuk model perilaku layanan. Kemampuan untuk memahami kedua struktur dan semantik data yang dilewati antara titik layanan sangat penting bagi semua pihak yang terlibat.

Sebagian besar contoh implementasi SOA biasanya ditampilkan sebagai pola interaksi permintaan-respon, sehingga diperlukan pertukaran yang lebih kuat. Selain itu, *platform* layanan modern juga membutuhkan fleksibilitas untuk mendukung pola-pola pertukaran pesan lanjutan.

SOA juga dapat dinyatakan dengan cara yang dipisahkan dari implementasi misalnya dengan menggunakan servis infrastruktur. Layanan yang ada harus dipandang sebagai aset teknologi informasi yang memiliki daur hidup (*life cycle*) seperti layaknya aplikasi. Interaksi antara suatu *service* dengan *service* lainnya digambarkan dalam suatu *reference architecture* seperti gambar di bawah ini.



Gambar 4. Reference architecture

5. KESIMPULAN

Dalam arsitektur teknologi informasi tradisional, aktifitas-aktifitas proses bisnis, aplikasi dan data terkunci umumnya masih berbentuk pulau-pulau (silos) independen dan seringkali tidak saling kompatibel. Pengguna harus melakukan navigasi ke jaringan, aplikasi dan basisdata yang berbeda untuk menjalankan rangkaian aktivitas yang merepresentasikan proses bisnis yang menjadi tanggung jawabnya. Silos yang independen tersebut membutuhkan anggaran dan staff yang relatif banyak untuk memeliharanya. Kondisi demikian memungkinkan untuk migrasi ke SOA.

SOA akan *men-deliver* data yang dibutuhkan untuk aktifitas-aktifitas proses bisnis sebagai layanan terintegrasi (*integrated services*). Pengguna tidak perlu melakukan *log* ke beberapa sistem, mencari data yang relevan dan mengintegrasikan data secara manual. Informasi nampak sebagai aplikasi tunggal, *di-deliver* dalam *screen* tunggal, dan keseluruhannya menggunakan login tunggal.

Implementasi SOA dapat memberikan manfaat diantaranya:

1. Memungkinkan independensi atas *platform* teknologi, karena standar-standar yang mendukung SOA telah diterima luas oleh seluruh *vendor* yang terkait dengan pengembangan solusi.
2. Memberikan lingkungan pengembangan, *deployment*, manajemen dan *monitoring* yang umum, lintas seluruh komponen integrasi, sehingga akan menghemat anggaran teknologi informasi dan waktu pemeliharaan oleh staff dibandingkan dengan poin-poin solusi yang terpisah.
3. Untuk memenuhi interoperabilitas, pengembang dapat mengintegrasikan, berbagi, dan menggunakan kembali sistem yang ada, menggunakan Java EE, dan .NET dalam satu *environment*.

5. DAFTAR PUSTAKA

- [1]. Duane Nickul et.al *Service Oriented Architecture (SOA) and Specialized Messaging Patterns*, 2007. Thecnical White Paper, USA
- [2]. Thomas Erl , *Introducing SOA Design Patterns* ,*SOA World Magazine*, June 2008, Vol. 8.
- [3]. Ed Ort (2005), *Service-Oriented Architecture and Web Services: Concepts, Sun Microsystems*

- [4]. D. Clark and D. L. Tennenhouse *Architectural Considerations for a New Generation of Protocols*
- [5]. MIT Tech Review, Dec 05/Jan 06 issue, article, p. 62–69
- [6]. <http://msdn.microsoft.com/en-us/library/>
(diakses tanggal 20 Februari 2012 jam 9.30)