

PERANCANGAN SISTEM BASISDATA ERP (*ENTREPRISE RESOURCE PLANNING*) PENDIDIKAN TINGGI DENGAN METODE SOA (*SERVICES ORIENTED ARCHITECTURE*)

Anief Rufiyanto

Universitas Pandanaran

Jl. Banjarsari Barat No. 1, Pedalangan, Banyumanik, Semarang

email: aniefrufiyanto@gmail.com

Abstract

ERP system for higher education is a software in order to accommodate the resource needs in an educational institution. ERP (enterprise resource planning) is an information system companies that designed to coordinate all resource, information and activities that need for complete business process. ERP system based in database, at general and software plan modular. Primal condition from ERP system is integration. Integration, it's mean to unite various need in one software in one logical database. Database as base application, built consideringly base SOA (service oriented architecture). SOA be software planning that designed in such a manner, up to there flexible moment development phase system and or integration process, where prepare loosely-integrated suit of services that at use in various business domain. The system assesses the development of the utilization of existing resources in an educational institution. Because after all educational institutions should have policies not only make a profit as an economic enterprise. Implemented starting from the data of teachers, students, subject matter to, evaluation of students' learning, including the annual report. Design and implementation of system 3 components, namely: People (human resources and student), Process (interaction of the learning process and the system) and Product (learning content, output).

Keywords: higher education, ERP, database, SOA

PENDAHULUAN

Dalam era informasi saat ini, globalisasi dunia pendidikan mengharuskan perguruan tinggi meningkatkan pelayanan pengelolaan informasi, sehingga kebutuhan informasi terhadap stake holder yang berkaitan dengan pendidikan bisa mendapatkan respon pelayanan yang cepat dan selalu mengikuti kondisi (*real time*) saat sebuah informasi tersebut dibutuhkan. Untuk mencapai hal itu diperlukan suatu pengelolaan teknologi informasi yang ada secara terencana dan terstruktur. Tata kelola teknologi informasi merupakan struktur hubungan dan proses untuk mengarahkan dan mengendalikan organisasi untuk mencapai tujuannya, dengan menampilkan nilai fungsional prosesnya guna meningkatkan kinerja lembaga layanan pendidikan yang lebih cepat, tepat dan akurat.

Pengembangan database dalam sebuah pendidikan tinggi niscaya amat diperlukan bagi organisasi yang semakin kompleks dengan berbagai kebutuhan untuk menuju pengembangan perangkat lunak ERP (*Enterprise Resource Planning*). ERP adalah solusi enterprise seluruh aset dalam organisasi

dalam sebuah sistem dengan proses terintegrasi. (Robby, Owen Kwanentent, Frans Mei Wardana). ERP merupakan perangkat lunak bersifat terpadu untuk mewadahi kebutuhan sumber daya dalam sebuah institusi pendidikan baik sumber daya manusia (*human resource*) maupun secara sumber daya fisik (*materiil resource*) dan non-fisik (*immaterial resource*), dimana dari segala informasi sumber daya yang ada tersebut dapat ditampilkan proyeksi lembaga atau institusi tersebut dalam kurun waktu tertentu di masa mendatang. Dengan demikian sebuah pengambilan keputusan manajerial dalam institusi ini, salah satu pertimbangan adalah berdasar fakta dan data akurat tentang organisasi tersebut berlangsung. Institusi pendidikan tinggi sebagai salah satu bentuk organisasi nirlaba, berbeda dengan perusahaan yang berorientasi profit.

Konsep pengembangan basisdata untuk ERP berbasis SOA (*Services Oriented Architecture*) ini bertujuan untuk mendukung pelayanan pendidikan terpadu yang berbasiskan paradigma baru pendidikan di saat ini dan masa depan, yang salah satu factor

pendukung utamanya adalah pelayanan (*services*) terhadap segala pihak terkait. Implementasi dan perencanaan basis data harus mengacu pada 3 komponen utama, yaitu: orang (mahasiswa, dosen, karyawan), proses (proses pembelajaran, penilaian, segala interaksi kegiatan pendukung) dan hasil (kelulusan, kurikulum, dan lainnya).

Konsep ke depan ERP (*Enterprise Resource Planning*) untuk pendidikan tinggi ini dikembangkan terutama untuk membantu *stake holder* utama sebuah institusi pendidikan, agar organisasi yang dikelolanya bisa berkembang dengan baik sesuai acuan garis besar sistem pendidikan nasional yang ada. Dan juga bisa sesegera mungkin mengantisipasi kejadian atas proyeksi sumber daya institusi tersebut yang terjadi di masa mendatang. Dengan demikian institusi pendidikan bisa meningkatkan mutu dengan acuan standar nasional.

Sistem basis data ERP yang berbasis SOA (*Services Oriented Architecture*) dikembangkan berdasar suatu aplikasi dimodelkan sebagai komposisi dari sekumpulan service yang disediakan suatu komponen. SOA adalah sebuah perancangan perangkat lunak yang dirancang sedemikian rupa sehingga flexible saat digunakan dalam fase pembangunan sebuah system ataupun proses integrasi, dimana dengan perancangan perangkat lunak yang berorientasi service ini menyediakan servis tertentu yang bisa di gunakan dalam berbagai bisnis domain. (Prasetyo, Hendro Joko).

Basis data yang ada, harus dalam kelompok-kelompok kesatuan yang mudah diakses. Ciri utama SOA adalah mempromosikan melalui penggabungan komponen-komponen perangkat lunak. Diantara komponen perangkat lunak tersebut saling berinteraksi dalam satu kesatuan. Sedangkan keuntungan dari penggunaan Arsitektur SOA adalah *flexible, scalable, replacability dan fault tolerance*.

LANDASAN TEORI

Sistem Basisdata

Sistem basis data mempunyai lingkup yang lebih luas dibandingkan dengan basis data. Sistem basis data bisa diartikan kumpulan basis data dalam sebuah sistem, yang secara keseluruhan mempunyai hubungan dengan

didukung komponen lainnya. (Edhy Sutanta, 2011). Sistem basis data berupa sekumpulan sub sistem yang terdiri atas basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, dengan aplikasi terpisah dalam sistem komputer pendukungnya. Sistem basis data mempunyai beberapa elemen penting, yaitu : (Edhy Sutanta, 2011)

1. Basis data sebagai inti dari sistem basis data.
2. Perangkat lunak untuk perancangan dan pengelolaan basis data.
3. Perangkat keras sebagai pendukung operasional pengolahan data.
4. Manusia/brainware, sebagai pemakai, perancang dan pengelola basis data.

Basis data adalah suatu koleksi data yang saling berhubungan secara logis dan menggambarkan integrasi antara suatu tabel dengan tabel lain, dibuat untuk memenuhi kebutuhan dari sebuah organisasi. (Robby, Owen Kwanentent, Frans Mei Wardana).

Pengembangan basis data, secara prinsip mempunyai tujuan utama keuntungan seperti berikut ini: (Edhy Sutanta, 2011)

1. Data dapat digunakan oleh banyak pemakai.
Basis data harus mampu mewujudkan relasi diantara item-item data dari banyak file. Sehingga pemakai yang berbeda atau program aplikasi yang berbeda dapat menggunakan basis data yang sama dengan cara yang berbeda.
2. Investasi intelektual.
Program aplikasi dan struktur logik yang telah ada pada saat ini, tidak perlu dikerjakan ulang ketika ada perubahan pada basis data. Data baru dapat dengan mudah diintegrasikan dengan mudah terhadap data yang tersedia.
3. Penekanan biaya.
Penekanan biaya penyimpanan dimungkinkan karena kerangkapan data dapat dihindarkan. Biaya perubahan kebutuhan basis data yang semakin kompleks, tidak perlu dilakukan dari awal, sehingga pengelolaan tidak butuh waktu lama.
4. Menghilangkan sistem ganda.
Pengembangan sub sistem baru dilakukan dengan tetap mengacu pada basis data yang sama sehingga menghindari terjadinya pengembangan sistem ganda. Konsep ini

memungkinkan pengembangan dapat terjadi pada struktur, volume basis data, dan sub sistem pengolahan data dengan tetap mempertahankan integrasi antar sub sistem.

5. Kinerja meningkat.
Peningkatan kinerja secara personal dapat dicapai dengan kebutuhan informasi yang dapat terpenuhi dengan cepat, tepat, mudah dan akurat bersumber dari basis data.
6. Kejelasan data.
Setiap pemakai dapat mengetahui dengan jelas data apa saja yang tersedia dan kewenangan data yang dapat diakses.
7. Kemudahan pemakaian.
Dengan program aplikasi dan sistem pengolahan terstruktur, pemakai tidak perlu berpikir kerumitan teknis yang dihadapi saat penyimpanan maupun pengaksesan yang berlangsung otomatis.
8. Fleksibilitas penggunaan.
Pengaksesan basis data dapat dilakukan dengan berbagai cara berbeda, karena basis data tak tergantung aplikasi yang dipakai.
9. Kemudahan perubahan.
Basisdata dapat berubah tanpa mempengaruhi cara-cara untuk menggunakan data.
10. Akurasi dan konsistensi.
Akurasi data dalam basis data dilakukan sejak proses penangkapan data hingga menampilkan informasi; sedang konsistensi data dapat terjaga karena sejak awal penyimpanan terbebas dari kerangkapan data lewat pengendalian data.

Perancangan basisdata dapat dipandang dari dua sisi yaitu sisi pemakai dan sisi perancang. Pemakai adalah orang atau program aplikasi yang mengakses basis data. Sedang perancang ialah perancang sistem atau program, maupun pengelola basisdata. Dari sisi perancangan dapat dibedakan dua pandangan berbeda yakni secara konseptual dan secara fisis. Pandangan konseptual berupa data-data yang butuh disimpan serta relasi antara data satu dan lainnya. Pandangan fisis merupakan bentuk terapan berkaitan secara teknik, misal: teknik penyimpanan data, besaran data dan kapasitas penyimpanan, waktu penyimpanan basisdata dan backup data.

Kriteria basisdata yang utama adalah bersifat *integrated* dan *shared*. *Integrated* maksudnya adalah basisdata berupa penggabungan beberapa file data yang berbeda, dengan menghindari pengulangan

sebagian atau seluruh data. Pengertian *shared* artinya masing-masing file data secara individu dapat dipakai secara bersama oleh banyak pemakai, dalam berbagai aplikasi program tanpa saling terganggu.

ERP – Enterprise Resource Planning

ERP biasanya digunakan untuk perancangan dalam perusahaan manufaktur dan berorientasi laba, sedangkan suatu perguruan tinggi bukanlah suatu perusahaan manufaktur dan seharusnya tidak berorientasi laba. Dalam penelitian ini konsep perancangan sistem basis data perguruan tinggi akan mengadopsi terminologi ERP dengan arsitektur sistem berbasis pada servis.

ERP terdiri dari banyak terminologi yang mencakup perencanaan sumber daya manusia maupun aset perusahaan/organisasi secara keseluruhan dan berbagai proses didalamnya. (Muh. Ibnu Choldun R). Secara modular, *software* ERP biasanya terbagi atas modul utama yakni :

- Modul Operasi. Logistik umum, Distribusi dan penjualan, Manajemen material, Manajemen Mutu, Perencanaan Perawatan, Layanan Pelanggan, Manajemen Lingkungan, *Production Planning and Control, Project System*.
- Modul Finansial dan Akuntansi. *General Accounting, Financial Accounting, Controlling*, Manajemen Investasi, Pengendali Aset & Kekayaan Perusahaan.
- Modul Sumber Daya Manusia. Manajemen Personalia, Manajemen Waktu, *Payroll, Management Training* dan Kegiatan, Manajemen Organisasi, *Travel Management*.

ERP diproyeksikan sebagai langkah peningkatan pengelolaan perusahaan/ organisasi. Secara garis besar, ERP dapat digambarkan sebagai sekumpulan *tools* manajemen (*forecasting, payroll, logistik, inventori*, dan lainnya), penggunaan proses bisnis dan pengambilan keputusan yang cepat dan teruji, pelayanan konsumen dalam perdagangan secara elektronik (*customer, supplier*), perencanaan dan penjadwalan sumber daya (manusia, material, mesin, dsb.).

Sedang aplikasi ERP adalah perangkat lunak yang dikembangkan untuk menjawab kebutuhan bisnis dalam hal integrasi dan sinkronisasi proses bisnis. Aplikasi ini akan menumbuhkan fungsi otomatisasi dalam penyajian

data dan informasi perusahaan/organisasi secara *real time*.

Sistem ERP dapat mendatangkan manfaat jika kita tahu bagaimana cara mendapatkannya serta fasilitas yang ada pada ERP. Kaitan antara manfaat dengan cara mendapatkannya, tergambar pada Tabel 1:

Tabel 1. Kemanfaatan ERP

Manfaat	Cara Mendapatkannya
Akses informasi yang andal	DBMS yang fleksibel, data yang konsisten, dan akurat, sistem pelaporan lebih baik
Menghindari duplikasi data dan operasi	Modul-modul yang mengakses data dari basisdata terpusat, sehingga menghindari proses pemasukan dan modifikasi data dari berbagai titik yang berbeda, duplikasi tak terjadi.
Mempercepat waktu pemrosesan data	Meminimasi waktu pengambilan data dan pembuatan laporan.
Mengurangi biaya	Menghemat waktu, meningkatkan kontrol dengan melakukan analisis menyeluruh terhadap keputusan organisasional.
Kemudahan adaptasi	Perubahan pada proses bisnis dapat diadaptasi dengan mudah
Meningkatkan skalabilitas	Struktur sistem bersifat modular dan mudah dikustomisasi.
Kemudahan pemeliharaan	Dukungan purnajual sistem yang berjangka panjang.
Pengembangan global	Ekstensi modul hingga meliputi supply chain management, customer relationship management
e-commerce	Bisnis internet, kultur kolaboratif.

Selanjutnya dari aktifitas penggunaan ERP akan diperoleh keuntungan antara lain:

1. Integrasi data keuangan. Puncak manajemen bisa melihat dan mengontrol kinerja keuangan perusahaan/organisasi dengan lebih baik, berkat data keuangan yang terintegrasi.
2. Standarisasi Proses Operasi. Proses operasi dibakukan melalui terapan dan uji optimal di lapangan sehingga terjadi peningkatan produktivitas, penurunan inefisiensi dan peningkatan kualitas produk.
3. Standarisasi Data dan Informasi. Data dan informasi dibuat dalam keseragaman pelaporan, terutama untuk perusahaan besar yang biasanya terdiri dari banyak business unit dengan jumlah dan jenis bisnis yg berbeda-beda.

SOA – Service Oriented Architecture

SOA-Arsitektur berbasis layanan mengambil fungsi aplikasi warisan dan mengekspos ke Internet dengan cara yang handal, sangat tersedia, *scalable*, fleksibel, mudah diatur dan aman, metode berbasis internet mudah dan dapat diandalkan untuk membuat dan belajar akses. layanan teknologi web telah muncul sebagai paradigma baru komputasi terdistribusi. Lapisan layanan terpadu menyediakan sarana yang kuat dimana solusi berorientasi layanan kontemporer dapat mengerti beberapa manfaat utama. Kontribusi paling signifikan sub-lapisan ini membawa ke SOA adalah sebuah abstraksi logika dan tanggung jawab yang menghilangkan jasa berdasarkan sejumlah kendala desain.

Sebagai contoh, dengan abstrak logika proses bisnis:

- Aplikasi dan layanan bisnis dapat dirancang dengan bebas untuk menjadi proses-agnostik dan dapat digunakan kembali.
- Layanan Proses mengasumsikan tingkat yang lebih besar dari statefulness, sehingga lebih membebaskan jasa lainnya dari keharusan untuk mengelola negara.
- Proses logika bisnis terpusat di satu lokasi, sebagai lawan yang didistribusikan di seluruh dan tertanam dalam beberapa layanan.

Layanan dapat eksis di berbagai tingkatan abstraksi, tetapi umumnya mereka dapat dibagi menjadi empat kategori:

- Infrastruktur Layanan

- Point-to-Point Layanan
- Layanan Bisnis
- Composite Layanan Bisnis
- Arsitektur berorientasi layanan menyajikan sebuah pendekatan untuk membangun sistem terdistribusi yang memberikan fungsionalitas aplikasi sebagai layanan baik pengguna akhir aplikasi atau layanan lain. Hal ini terdiri dari unsur-unsur yang bisa dikategorikan menjadi kualitas fungsional dan pelayanan.

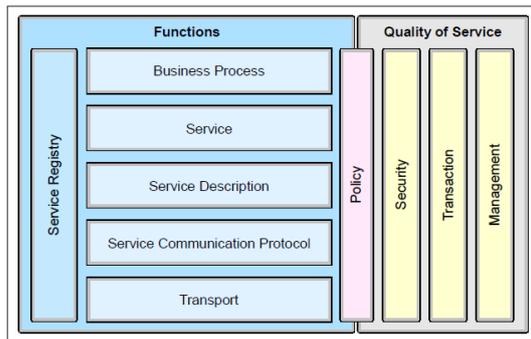


Figure 2-6 Elements of a service-oriented architecture

Tumpukan arsitektur dibagi menjadi dua bagian, dengan setengah kiri menangani aspek fungsional arsitektur dan bagian kanan menangani kualitas aspek pelayanan. Elemen-elemen SOA digambarkan dalam rincian berikut dibawah ini:

1. Aspek fungsional meliputi:

- Transport adalah mekanisme untuk memindahkan permintaan layanan dari layanan konsumen ke layanan penyedia, dan layanan tanggapan dari layanan penyedia untuk layanan konsumen.
- Layanan Komunikasi Protokol adalah aturan mekanisme dalam layanan penyedia dan layanan penggunaan konsumen untuk mengkomunikasikan apa yang diminta dan apa dikembalikan.
- Layanan Keterangan adalah skema yang disepakati untuk menggambarkan apa layanan ini, bagaimana harus dipanggil, dan data apa yang diperlukan untuk menjalankan layanan tersebut berhasil.
- Layanan menggambarkan layanan aktual yang dibuat tersedia untuk digunakan.
- Proses Bisnis adalah kumpulan layanan, dipanggil dalam urutan tertentu dengan set tertentu dari aturan, untuk memenuhi kebutuhan bisnis. Perhatikan bahwa proses bisnis dapat dianggap sebagai

layanan yang berdiri sendiri, yang mengarah ke gagasan bahwa proses bisnis dapat terdiri dari jasa tersendiri yang berbeda.

- Layanan Registry adalah gudang layanan dan data deskripsi yang dapat digunakan oleh penyedia layanan untuk mempublikasikan layanan mereka, dan konsumen layanan untuk menemukan atau menemukan layanan yang tersedia. Registri layanan dapat memberikan fungsi lain untuk layanan yang membutuhkan repositori terpusat.

2. Kualitas aspek pelayanan meliputi:

- Kebijakan adalah serangkaian kondisi atau aturan di mana penyedia layanan membuat layanan tersedia untuk konsumen. Ada aspek kebijakan yang fungsional, dan aspek yang berhubungan dengan kualitas pelayanan;
- Keamanan adalah seperangkat aturan yang bisa diterapkan untuk identifikasi, otorisasi, dan kontrol akses konsumen layanan jasa memohon.
- Transaksi adalah set atribut yang mungkin diterapkan untuk kelompok jasa untuk memberikan hasil yang konsisten. Misalnya, jika kelompok tiga layanan yang akan digunakan untuk menyelesaikan fungsi bisnis, semua harus menyelesaikan atau tidak harus menyelesaikan.
- Manajemen adalah set atribut yang mungkin diterapkan untuk mengelola layanan yang disediakan atau dikonsumsi.

Service Oriented Architecture (SOA) adalah prinsip desain yang diinginkan untuk membangun system terdistribusi yang reliable dan mengirimkan *service* fungsionalitasnya, dengan menambahkan penekanan pada loose coupling antar pengguna *service*. Dalam hal ini service mempunyai beberapa karakteristik :

- Logical view*, service yang dilihat dari level operasi bisnis yang diidentifikasi sebagai interface yang independen.
- Message orientation*, Sebuah service yang berhubungan dengan client yang bertukar *message*.
- Description Orientation*, Service yang didekripsikan sebagai mesin pengolah metadata.
- Network Orientation*, Service dituntut untuk dipakai di dalam jaringan. Hal ini

menekankan pada kebutuhan service yang secara otomatis serta mudah ditemukan.

- e. *Platform neutrality*, Pesan disampaikan melalui interface yang menggunakan platform netral (*multi platform*) dan format data yang standart seperti XML.

METODOLOGI

Dari berbagai review mengenai basis data dan ERP diatas yang bisa diterapkan dalam lingkup pendidikan tinggi dengan kebutuhan pelayanan saat ini bagi perguruan tinggi, dengan berpedoman pada berbagai keunggulan sistem SOA dalam mengikuti perubahan sistem maka konsep sistem basis datanya harus bersifat mudah dan terotomatisasi.

Pendidikan tinggi punya karakteristik berbeda dengan organisasi lain karena perbedaan lingkungan dan lingkaran budayanya serta penggunaan ERP untuk kegiatan akademik. Interaksi umum staf akademik dengan kegiatan utama institusi lewat ERP dan mahasiswa membutuhkan informasi yang lebih baik dan lebih banyak guna mendukung e-learning. Organisasi akan melaksanakan tata kelola institusi sesuai visi dan misi, melakukan kegiatan menguntungkan maupun even berorientasi nir-laba. Namun secara pasti sistem ERP jelas bertujuan utama meningkatkan persentase tinggi atas implementasi kesalahan hingga semakin cepat diketahui bila terjadi penyimpangan saat tak sesuai tujuan yang ditetapkan.

Konsep basis data bagi keperluan ini yang mampu mendukung kriteria ERP dan sesuai dengan arsitektur sistem berbasis pelayanan, sekaligus multiplatform tidak tergantung pada sistem operasi ataupun aplikasi, basis data tetap tidak terpengaruh. Namun demikian sistem basis datanya tetap mampu untuk dikembangkan di masa mendatang, tanpa perubahan yang signifikan pada data itu sendiri. Hanya kebutuhan sistem dan aplikasi saja yang berkembang sesuai kebutuhan kompleksitas organisasi.

PEMBAHASAN

Lingkup pekerjaan basisdata bagi perguruan tinggi diharapkan memenuhi kesesuaian dengan Pangkalan Data Pendidikan Tinggi Direktorat Jenderal Pendidikan Tinggi Departemen Pendidikan Nasional yang dibangun memiliki fasilitas lengkap untuk

menunjang proses pengelolaan dan informasi Ditjen DIKTI. Disamping itu fasilitas yang dimiliki dapat mampu menunjang proses pada tingkat operator sampai dengan tingkat eksekutif yang membutuhkan alat bantu dalam menganalisa informasi untuk pengambilan keputusan yang tepat.

Ruang lingkup atau batasan pekerjaan sesuai dengan maksud, tujuan dan sasaran diatas dengan pertimbangan persyaratan, antara lain Sistem yang dirancang memiliki spesifikasi sebagai berikut :

- a. Menggunakan Konsep *Open System*.
- b. Berbasis *Relational Database Management System* (RDBMS).
- c. Software pembangunan sistem aplikasi dan tools yang digunakan.
- d. Untuk pembangunan dan pengembangan aplikasi.
- e. Memiliki prosedur backup dan recovery secara otomatis.
- f. Database hasil *input user* dapat di *query* dan *sort* untuk kepentingan pelaporan tertentu dan tersedia interface untuk melakukan query dimaksud.

Sistem yang dibangun harus dapat menerapkan *security system* yang bekerja pada tiga level yang dapat diterapkan sampai dengan fungsi modul, atau submodul dan juga dapat diterapkan ke user/group user, yaitu :

- a. *Level Operating System*
- b. *Level Database*
- c. Level Aplikasi

Aplikasi yang dibangun berbasis web yang dirancang dengan platform open source dengan berbagai tingkatan akses user sesuai dengan kewenangannya, antara lain : administrator, operator, pimpinan, pemohon dan user atau pengguna lainnya.

Aplikasi yang dibangun dapat dijalankan pada platform OS Windows server. XP, Vista atau Linux, Database RDBMS dan script pemrograman yang berorientasi object, sehingga mampu menangani dan mengelola data.

Membangun perangkat lunak sistem yang telah dirancang secara terpusat dan ditempatkan di Data Center.

Aplikasi yang dibangun dapat diakses secara terbatas melalui jaringan internet dan jaringan lokal (LAN) dengan menggunakan browser internet. Aplikasi yang dibangun harus *user fiendly* dengan *interface* berbasis *Graphical User Interface* (GUI). Aplikasi

dibangun dengan pendekatan metodologi "Software Engineering" yang baik.

Dari berbagai review mengenai basis data dan ERP diatas yang bisa diterapkan dalam lingkup pendidikan tinggi dengan kebutuhan pelayanan saat ini bagi perguruan tinggi, dengan berpedoman pada berbagai keunggulan sistem SOA dalam mengikuti perubahan sistem maka konsep sistem basis datanya harus bersifat mudah dan terotomatisasi.

Perguruan tinggi berbeda dari organisasi lainnya sebab mempunyai lingkungan dan situasi yang berbeda, dan penggunaan teknologi ERP dilakukan untuk menunjang tujuan akademik. Saat ini di Indonesia, ujung tombak pelaksanaan pendidikan di perguruan tinggi adalah program studi. Maka pengembangan ERP sudah seharusnya berawal dari seluruh kegiatan di tiap program studi. Mahasiswa memerlukan informasi lebih lanjut dan lebih baik saat ini dalam lingkungan e-learning.

Empat faktor keberhasilan implementasi ERP sebagai berikut:

1. Top-manajemen kontribusi dukungan, perencanaan, pelatihan, dan tim;
2. Usaha pilihan perangkat lunak;
3. Informasi partisipasi sistem dan sub-sistem; dan kemampuannya;
4. Konsultasi dan dukungan. Mereka juga menyediakan lima pertanyaan hasil, yang terbukti berkorelasi secara signifikan dan harus, karena itu, dikombinasikan untuk membentuk faktor hasil tunggal, efektivitas. Analisis regresi membuktikan bahwa semua faktor keberhasilan dapat mempengaruhi hasil secara signifikan; Oleh karena itu, faktor-faktor ini dapat dianggap sebagai faktor keberhasilan perwakilan dalam implementasi ERP.

Sistem *Enterprise Resource Planning* adalah kunci untuk mengoptimalkan kinerja organisasi; Namun, memilih sistem yang menerapkan, ketika melakukan implementasi, dan bagaimana untuk meminimalkan biaya sekaligus memaksimalkan sistem penerimaan semua isu-isu penting untuk bisnis mempertimbangkan implementasi ERP.

Perancangan *Enterprise Resource Planning* dengan terapan di perguruan tinggi, sesuai dengan paradigma sekarang harus berbasis pelayanan. Oleh karena saat ini dan selanjutnya di masa mendatang, butuh kapasitas dan kompleksitas yang semakin

tinggi. Demikian, sistem basisdata yang dibutuhkan lebih tepat menyesuaikan ke penataan simple namun mudah tumbuh, tanpa merubah ulang secara keseluruhan sistem dan prosesnya.

Service Oriented Architecture (SOA) adalah prinsip desain yang diinginkan untuk membangun system terdistribusi yang reliable dan mengirimkan service fungsionalitasnya, dengan menambahkan penekanan pada *loose coupling* antar pengguna service. Dalam hal ini service mempunyai beberapa karakteristik :

1. *Logical view*, service yang dilihat dari level operasi bisnis yang diidentifikasi sebagai interface yang independen
2. *Message orientation*, Sebuah service yang berhubungan dengan client yang bertukar message.
3. *Discription Orientation*, Service yang didekripsikan sebagai mesin pengolah metadata.
4. *Network Orientation*, Service dituntut untuk dipakai di dalam jaringan. Hal ini menekankan pada kebutuhan service yang secara otomatis serta mudah ditemukan.

Arsitektur berorientasi layanan software memiliki prinsip-prinsip ini, seperti:

1. Layanan yang ditemukan dan dinamis terikat.

SOA mendukung konsep penemuan layanan. Seorang konsumen layanan yang membutuhkan layanan, menemukan layanan apa yang harus digunakan berdasarkan seperangkat kriteria saat runtime. Layanan konsumen meminta registry untuk layanan yang memenuhi kebutuhannya. Cara terbaik untuk menjelaskan dinamis mengikat dan menemukan adalah dengan menggunakan contoh. Sebagai contoh, aplikasi perbankan (konsumen) meminta registry untuk semua layanan yang melakukan validasi kartu kredit. registri mengembalikan semua entri yang mendukung ini. Entri juga berisi informasi tentang layanan, termasuk biaya transaksi. konsumen memilih layanan (provider) dari daftar berdasarkan biaya transaksi terendah. Deskripsi layanan terdiri dari semua argumen yang diperlukan untuk menjalankan layanan. konsumen format pesan permintaan dengan data, berdasarkan deskripsi yang disediakan oleh pointer direktori Satu-satunya ketergantungan antara produsen dan konsumen adalah

- kontrak, yang registri pihak ketiga menyediakan. ketergantungan adalah ketergantungan runtime, bukan waktu kompilasi ketergantungan. Semua informasi kebutuhan konsumen tentang layanan diperoleh dan digunakan pada saat runtime. Contoh ini menunjukkan bagaimana konsumen mengeksekusi layanan dinamis.
2. Layanan mandiri dan modular.

Salah satu aspek yang paling penting dari SOA adalah konsep modularitas. Sebuah layanan mendukung satu set antarmuka. Antarmuka ini harus kohesif, yang berarti bahwa mereka semua harus berhubungan satu sama lain dalam konteks modul. Prinsip-prinsip modularitas harus ditaati dalam merancang layanan yang mendukung aplikasi agar layanan dapat dengan mudah digabungkan ke dalam aplikasi dengan beberapa dependensi terkenal. Kriteria ini berlaku juga ketika menentukan apakah layanan cukup modular. *Modular decomposability*, dari layanan mengacu pada pemecahan aplikasi menjadi banyak modul yang lebih kecil modular composability, dari layanan mengacu pada produksi jasa perangkat lunak yang dapat secara bebas dikombinasikan secara keseluruhan dengan layanan lain untuk menghasilkan baru sistem. *Modular Understandability*, berarti modular dari layanan adalah kemampuan seseorang untuk memahami fungsi dari layanan tanpa memiliki pengetahuan tentang layanan lainnya. *Modular Continuity*, kelangsungan modular dari layanan mengacu pada dampak perubahan dalam satu layanan yang membutuhkan perubahan dalam layanan lain atau konsumen dari layanan. Antarmuka yang tidak cukup menyembunyikan rincian implementasi layanan menciptakan efek domino ketika perubahan yang diperlukan. *Modular Pengamanan*, Perlindungan modular dari layanan ini cukup jika kondisi abnormal dalam pelayanan tidak kaskade ke layanan lain atau konsumen. Misalnya, jika kesalahan dalam pelayanan rekening menyebabkan data yang tidak valid untuk disimpan pada database, ini dapat mempengaruhi pengoperasian layanan lain menggunakan tabel yang sama untuk data mereka.
 3. Penekanan Layanan interoperabilitas.

Arsitektur berorientasi pada layanan interoperabilitas, kemampuan sistem yang menggunakan platform dan bahasa yang berbeda untuk berkomunikasi satu sama lain. Setiap layanan menyediakan antarmuka yang dapat dipanggil melalui tipe konektor. Konektor interoperable terdiri dari protokol dan format data yang masing-masing klien potensial dari layanan mengerti. Interoperabilitas dicapai dengan mendukung protokol dan format data dari layanan ini klien saat ini dan potensi. Teknik untuk mendukung protokol standar dan format data terdiri dari pemetaan karakteristik dan bahasa setiap platform untuk spesifikasi mediasi. Mediasi peta spesifikasi antara format dari format data interoperable dengan format data spesifik platform. Kadang-kadang ini memerlukan set karakter pemetaan seperti ASCII ke EBCDIC serta jenis data pemetaan. Misalnya, layanan Web adalah spesifikasi mediasi untuk berkomunikasi antara sistem. JAX-RPC dan peta JAXB tipe data Java untuk SOAP. platform lain bahwa layanan supportWeb memediasi spesifikasi layanan betweenWeb dan spesifikasi internal mereka sendiri untuk set karakter dan tipe data.
 4. Kemudahan Layanan ditingkatkan.

Coupling mengacu pada jumlah ketergantungan antara modul. Ada dua jenis kopling: longgar dan ketat. Longgar modul digabungkan memiliki beberapa dependensi terkenal. Modul ketat digabungkan memiliki banyak dependensi yang tidak diketahui. Setiap arsitektur perangkat lunak berusaha untuk mencapai kopling longgar antara modul. arsitektur berorientasi layanan mempromosikan kopling longgar antara konsumen layanan dan penyedia layanan dan ide dari beberapa dependensi terkenal antara konsumen dan penyedia. Sebuah gelar sistem kopling langsung mempengaruhi modifiability nya. Semakin erat digabungkan sistem, semakin perubahan layanan akan memerlukan perubahan dalam layanan konsumen. Coupling meningkat ketika konsumen layanan memerlukan sejumlah besar informasi tentang penyedia layanan untuk menggunakan layanan ini. Dengan kata lain, jika konsumen layanan tahu lokasi dan format data rinci untuk penyedia layanan,

konsumen dan penyedia yang lebih erat digabungkan. Konsumen tidak tergantung secara langsung pada pelaksanaan layanan, tetapi hanya pada kontrak layanan pendukung.

5. Layanan mempunyai antarmuka yang saling mengerti.

Konsep granularity berlaku untuk layanan dalam dua cara. Pertama, itu diterapkan untuk lingkup domain seluruh layanan mengimplementasikan. Kedua, itu diterapkan untuk lingkup domain yang masing-masing metode dalam antarmuka mengimplementasikan. Tingkat granularity yang relatif satu sama lain. Misalnya, jika layanan mengimplementasikan semua fungsi dari sistem perbankan, maka kita menganggap itu berbutir kasar. Jika mendukung hanya validasi kartu kredit, kami menganggap itu halus. Selain itu, jika sebuah metode untuk bertanya tentang pelanggan mengembalikan semua informasi pelanggan, termasuk alamat, metode ini akan menjadi kasar-grained dari metode yang tidak mengembalikan alamat pelanggan.

Tingkat yang tepat dari granularity untuk layanan dan metode yang relatif kasar. Sebuah layanan umumnya mendukung konsep bisnis yang berbeda tunggal atau proses. Ini berisi perangkat lunak yang mengimplementasikan konsep bisnis sehingga dapat digunakan kembali dalam beberapa besar, sistem terdistribusi.

6. *Services are location-transparent.*

Lokasi transparansi merupakan karakteristik kunci dari arsitektur berorientasi layanan. Konsumen dari layanan tidak tahu lokasi layanan sampai mereka menemukannya di registri. Lookup dan dinamis mengikat ke layanan pada saat runtime memungkinkan implementasi layanan untuk pindah dari lokasi ke lokasi tanpa sepengetahuan klien. Kemampuan untuk memindahkan layanan meningkatkan ketersediaan layanan dan kinerja. Dengan menggunakan sebuah penyeimbang beban yang meneruskan permintaan untuk beberapa contoh layanan tanpa sepengetahuan layanan klien, kita dapat mencapai ketersediaan yang lebih besar dan kinerja. Seperti disebutkan sebelumnya, prinsip desain sentral dalam sistem

berorientasi objek adalah pemisahan pelaksanaan dari antarmuka. Ini berarti bahwa antarmuka objek dan pelaksanaannya dapat bervariasi secara independen.

7. Layanan yang mudah dipadukan.

Sebuah composability layanan terkait dengan struktur modular. Struktur modular memungkinkan layanan yang akan dirakit menjadi aplikasi pengembang tidak punya gagasan ketika merancang layanan. Menggunakan yang sudah ada sebelumnya, layanan diuji sangat meningkatkan kualitas sistem dan meningkatkan pengembalian atas investasi karena kemudahan penggunaan kembali.

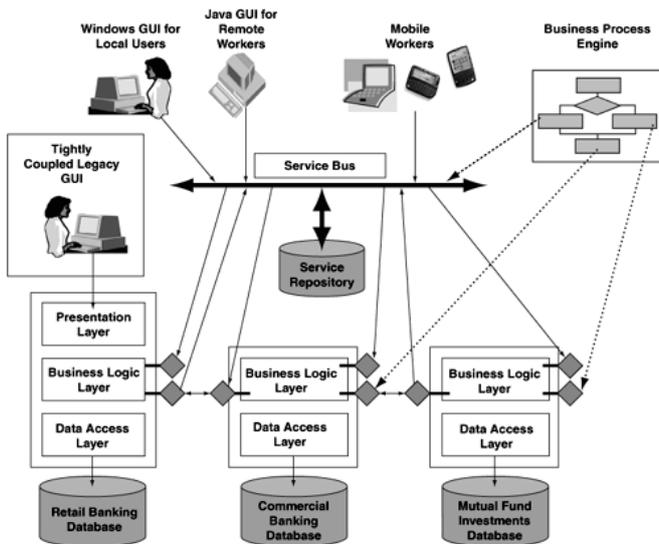
Services Oriented Architecture

Kebutuhan untuk merealisasikan konsep kerja sama yang dinamis antar perusahaan seperti contoh tersebut, maka perusahaan memfokuskan pada *system enterprise* dan proses integrasi antar enterprise. Disinilah profesional IT secara langsung menghadapi tantangan untuk mampu membuat

dan mengontrol *legacy* teknologi dalam batasan perusahaan/enterprise dan heterogenitasnya, yaitu suatu system yang berbasis *service-oriented*.

Service Oriented Architecture (SOA) adalah prinsip desain yang diinginkan untuk membangun system terdistribusi yang reliable dan mengirimkan service fungsionalitasnya, dengan menambahkan penekanan pada loose coupling antar pengguna service. Dalam hal ini service mempunyai beberapa karakteristik :

- *Logical view*, service yang dilihat dari level operasi bisnis yang diidentifikasi sebagai interface yang independen.
- *Message orientation*, Sebuah service yang berhubungan dengan client yang bertukar message.
- *Description Orientation*, Service yang didekripsikan sebagai mesin pengolah metadata.
- *Network Orientation*, Service dituntut untuk dipakai di dalam jaringan. Hal ini menekankan pada kebutuhan service yang secara otomatis serta mudah



Gambar 1. System dengan Arsitektur berbasis SOA

1. Service Oriented Architecture.

Arsitektur berorientasi servis, demikian bila kata *'service oriented architecture'* di terjemahkan ke dalam bahasa Indonesia. Jika mendengar kata-kata ini, mungkin kita akan terbayang dengan yang namanya rancangan rancangan perangkat lunak yang menggunakan web service atau ada juga yang menyebutnya web method..

Menurut wikipedia, Service Oriented Architecture (SOA) adalah " a flexible set of design principles used during the phases of system development and integration". Jika kita artikan secara harfiah, maka SOA adalah sebuah perancangan perangkat lunak yang dirancang sedemikian rupa sehingga flexible yang digunakan dalam fase pembangunan sebuah system ataupun proses integrasi, dimana dengan perancangan perangkat lunak yang berorientasi service ini menyediakan loosely-integrated suit of services yang bisa di gunakan dalam berbagai bisnis domain. Maksudnya dari loosely-integrated suit of services adalah layanan-layanan yang dapat dengan mudah di integrasikan.

SOA adalah sebuah pendekatan dalam merancang suatu aplikasi dengan me-reuse komponen-komponen yang sudah ada sebelumnya dalam hal ini komponen-komponen tersebut memberikan layanan bisnis tertentu.

2. Karakteristik

Karakteristiknya SOA dapat dilihat sebagai berikut:

a. Komponen-komponen yang dibangun dalam SOA berinteraksi bebas dan lepas satu sama lain, dengan bersifat *loose coupled* sebuah servis dapat digunakan oleh program atau servis lainnya tanpa perlu memperhatikan platform atau teknologi yang digunakan oleh servis tersebut, sebagai contoh, misalnya sebuah aplikasi HR yang bertugas untuk menyimpan data data elearning dari divisi learning center, maka aplikasi tersebut tinggal memanggil servis dari aplikasi e-learning tersebut tanpa harus memperhatikan platform dari aplikasi e-learning tersebut.

b. Karakteristik ke dua adalah, SOA tersusun dari dua bagian, yaitu service interface dan *service implementation*. Sebagai contoh misalnya ada servis cetak sertifikat pelatihan, dalam mencetak sebuah sertifikat pelatihan maka ada data yang diinputkan kedalam servis tersebut misalnya NIK, Kode Pelatihan, dan ada data yang di outputkan yaitu sertifikat pelatihan. Input/Output inilah yang dikatakan service interface atau dengan kata lain parameter parameter input/output dari sebuah service.

Service implementation adalah bagaimana sebuah data input di proses dalam servis tersebut hingga menghasilkan output, atau bisa dikatakan service implementation berisi logic proses dari sebuah servis. SOA tidak memperdulikan *service implementation*-nya di bangun dengan bahasa pemrograman apa, yang penting implementation dapat menerima input dan mengeluarkan output sesuai dengan Interfacenya. Karena penerapan SOA pada umumnya akan menggunakan bahasa sejuta platform yaitu *eXtensible Markup Language (XML)*. Servis harus business oriented, maksudnya disini adalah setiap service yang disediakan harus melakukan suatu aktifitas bisnis tertentu.

3. Kelebihan Service Oriented Architecture

Ada beberapa kelebihan SOA, diantaranya:

- a. Dapat menyatukan berbagai platform yang berbeda, pada pengimplementasian SOA menggunakan bahasa berbagai macam platform yaitu XML sehingga bisa menyatukan berbagai platform.
- b. Tahan terhadap perubahan. Perusahaan besar atau organisasi seringkali berubah untuk meningkatkan efisiensi serta menghadapi tantangan bisnis yang semakin menantang. Akibatnya perubahan dari sisi perangkat lunak yang menjadi bisnis support harus mampu dan menyelaraskan diri dengan perubahan bisnis tersebut, bahkan apabila memungkinkan perangkat lunak menjadi enabler untuk proses bisnis tersebut, sehingga IT tidak hanya menjadi bisnis support dalam perusahaan ini, namun menjadi bisnis enabler yang bisa menyelesaikan, menjawab tantangan bisnis yang dihadapi perusahaan.

4. *Service Oriented Architecture dan Web Services*

Terapan SOA diperusahaan ini, salah satunya dengan menggunakan Web Service, walaupun ada cara cara yang lain yang bisa digunakan untuk implementasi SOA ini. *Web Service* merupakan sekumpulan fungsionalitas yang dapat diakses melalui *Internet Protocol* (IP) standar. *Web Service* banyak digunakan untuk aplikasi sistem yang terdistribusi secara heterogen baik dari sistem operasi sampai model objek. Kekuatan dari *Web Service* adalah menggunakan model *messaging* berbasis teks untuk komunikasi yang memungkinkan dapat beroperasi secara efektif pada platform yang berbeda. Oleh karena itulah *Web Service* banyak digunakan dan diakui secara luas. *Web service* mempunyai prinsip kerja mengembalikan hasil service dalam bentuk XML yang berorientasi pada data yang tidak dapat langsung di konsumsi oleh user. Oleh sebab itu *web service* sering juga dikatakan bersifat *application-centric* bukan *human-centric*.

5. *Enterprise Service Oriented Architecture*

Enterprise Service Oriented Architecture, bisa diartikan sebagai rancangan aplikasi enterprise yang berbasis services. Jika kita punya sebuah aplikasi yang menggunakan data karyawan dimana masing masing

aplikasi punya database sendiri sendiri yang berisi data karyawan, dengan penerapan ESOA ini maka, tidak perlu ada beberapa table karyawan untuk aplikasi yang berbeda cukup satu dan mengambilnya dari service yang kita buat di SAP. Keuntungan dari hal ini, adalah menyediakan berbagai servis yang di perlukan dalam skala Enterprise, mengurangi redundancy pembuatan fungsional dalam sebuah perangkat lunak, dan sebagainya.

“*Database is a self-describing collection of integrated tables*”, yang berarti *database* adalah sebuah koleksi data yang menggambarkan integrasi antara tabel yang satu dengan tabel yang lainnya. “*Database is a self-describing*”, disini dijelaskan bahwa struktur data saling terintegrasi dalam suatu tempat yang dikenal sebagai kamus data atau metadata. Berikut ini merupakan alasan dari penggunaan *database*: (Robby, Owen Kwanentent, Frans Mei Wardana)

1. Padat. Tidak perlu lagi membuat arsip kertas dalam ukuran besar.
2. Kecepatan. Mesin dapat mendapatkan kembali dan mengubah data jauh lebih cepat daripada yang manusia yang dapat lakukan.
3. Mengurangi pekerjaan yang membosankan. Rasa bosan dari proses memelihara arsip – arsip berupa kertas dapat dikurangi.
4. Aktual. Informasi yang terbaru dan akurat selalu tersedia disetiap waktu ketika dibutuhkan.

Database dengan data warehouse berbasis object oriented project (perintah sql) yang dihasilkan akan lebih ramping dan mengakses data secara nyata akan lebih cepat, yaitu jumlah record yang diproses makin sedikit dan proses join berkurang. Dari keseluruhan efisiensi kenaikan prosentase jika digabungkan akan menghasilkan rata-rata efisiensi kenaikan prosentase 461.801,84%. Yang menunjukkan bahwa penggunaan data warehouse lebih handal dan efisien dibandingkan penggunaan database model lama. (Spits Warnars)

Model data relational sering juga disebut basisdata relational. Model ini menunjukkan suatu mekanisme dengan mengelola /mengorganisasi data secara fisik dalam memori sekunder yang akan berdampak pula pada pengelompokan keseluruhan data dalam sistem. (Edhy Sutanta, 2011). DBMS (*Database Management System*) adalah sistem

perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara dan mengontrol akses ke *database*. Database Management System adalah perangkat lunak khusus yang digunakan untuk membuat, mengakses, mengontrol, dan mengatur sebuah *database*. Database Management System merupakan kumpulan data yang saling berhubungan dan juga mengandung kumpulan program untuk mengakses data tersebut.

Jadi, Database Management System adalah perangkat lunak yang berinteraksi dengan program aplikasi pengguna dan *database*. Database Management System menyediakan beberapa fasilitas sebagai berikut : (Robby, Owen Kwanentent, Frans Mei Wardana)

1. DDL (*Data Definition Language*) memungkinkan pengguna untuk mendefinisikan tipe data (*data type*), struktur (*structure*), dan batasan – batasan (*constraints*) pada data yang disimpan ke dalam *database*.
 2. DML (*Data Manipulation Language*) memungkinkan pengguna untuk memasukkan (*insert*), mengubah (*update*), menghapus (*delete*), dan menampilkan (*retrieve*) data dari *database*
 3. *Access control* menyediakan akses yang terkontrol ke *database*, seperti *security system*, *integrity system*, *concurrency control system*, *recovery control system*, dan *user-accessible catalog*.
- DBMS memiliki beberapa fungsi, yaitu:
- a. Sebuah. penyimpanan data, pengambilan, dan update. Sebuah dbms harus membekali / mempersiapkan pengguna cakap penyimpanan, kembali penelusuran, dan mengubah data dalam *database*.
 - b. Katalog diakses pengguna. Sebuah dbms harus mempersiapkan katalog yang menggambarkan lokasi penyimpanan data dan dapat diakses oleh pengguna dukungan transaksi. Dbms harus mempersiapkan mekanisme yang akan menjamin setiap dari semua kegiatan akan mengubah itu juga tidak transaksi.
 - c. layanan kontrol konkurensi. Dbms harus mempersiapkan mekanisme untuk menjamin *database* yang dapat diubah-benar ketika beberapa pengguna mengubah kapan *database* yang bersamaan.
 - d. layanan pemulihan. Dbms harus mempersiapkan mekanisme untuk

memperbaiki basis data yang rusak karena insident tertentu.

- e. layanan otorisasi. Dbms harus mempersiapkan mekanisme untuk menjamin bahwa hanya pengguna yang diberi wewenang yang dapat mengakses basis data.
Dukungan untuk komunikasi data. Dbms harus dapat ber-Integrasi dengan software komunikasi.
- f. layanan integritas. Dbms harus mempersiapkan cara untuk menjamin bahwa data dalam basis data dan perubahan data, keduanya mengikuti aturan yang benar. Layanan untuk mempromosikan independensi data. Dbms harus menutupi *facilitiess* yang mendukung program kemandirian dari struktur basis data yang TERBARU
- g. layanan utilitas. Dbms harus mempersiapkan fasilitas sarang layanan.
Saat ini, model objek semantik merupakan salah satu [yang] model data teknik pembuat teknologi layanan web sering dipakai, karena kesopanan dalam bentuk kata-kata. Model data ini lapisan mentransformasikan akan desain *database*. data dari bentuk, laporan dan permintaan diwakili dalam bentuk objek semantik. sama seperti entitas, objek semantik terdiri dari atribut - atribut. atribut objek semantik terdiri dari 3 jenis:
 - a. Sebuah. atribut sederhana. elemen tunggal.
 - b. atribut kelompok. Koleksi dari elemen tunggal yang memiliki karakter yang sama, misalnya: alamat terdiri dari jalan, kota, provinsi, kode pos.
 - c. atribut objek semantik. Atribut yang menghubungkan antara objek semantik satu dengan yang lain.
Setiap memiliki kardinalitas dengan format yang n. m yang n kardinalitas minimal dan m maksimal kardinalitas. model objek semantik tidak memiliki satu pelanggan cara, jika objek yang mengandung objek lainnya, sehingga objek kedua juga mengandung objek pertama, ini disebut atribut dipasangkan. ditemukan objek identifier menjadi salah satu atau objek atribut lebih yang digunakan untuk mengidentifikasi objek dan kelompok identifier menjadi identifier yang memiliki lebih dari 1 atribut.
Ada 7 jenis objek dari diagram objek semantik:

1. objek sederhana. objek hanya memiliki atribut sederhana atau atribut kelompok, dan nilai tunggal-nilai.
2. Benda *Composite*. objek hanya memiliki atribut sederhana atau atribut kelompok, dan nilai ada multi-nilai.

<i>Object 2 can Object 1 contain can contain</i>	<i>One</i>	<i>Many</i>
<i>One</i>	1:1	1:N
<i>Many</i>	M:1	M:N

3. *Compound* benda. objek yang paling mengandung sedikit satu atribut objek (mungkin menambah satu / dua jenis atribut lainnya) tetapi nilai tunggal atribut dan nilai kelompok attribute-single-nilai. Ada 4 jenis senyawa objek, gambar di bawah ini menggambarkan tipe keempat.
4. Benda Hybrid. objek yang menjadi kombinasi antara objek komposit dan benda majemuk, telah satu objek sub-atribut dalam kelompok attribute- nilai multi-nilai.
5. Asosiasi benda. benda yang dua atau lebih atribut objek lain dan menyimpan data yang unik bagi pelanggan.
6. Induk benda/subtipe. sama seperti kelas konsep, prinsip kelas induk dan subtipe keturunan dari objek lain dan mengandung seluruh atribut dari objek.
7. Pola dasar benda. objek semantik yang membuat objek semantik lainnya yang mewakili versi, Rilis, atau edisi dari pola dasar.

KESIMPULAN

Berdasarkan hasil analisis dan perencanaan konsep yang dilakukan, sehingga dapat diambil kesimpulan sebagai berikut ini:

1. Erp (*enterprise resource planning*) dapat diterapkan dalam perencanaan sistem informasi di perguruan tinggi, sistem informasi yang dikembangkan menggunakan terminologi ERP seharusnya dapat menghasilkan perbaikan terus (terus menerus perbaikan) dari aktivitas dan sumber daya penyediaan.
2. Arsitektur berorientasi Layanan (SOA) dilaksanakan oleh teknologi selain layanan web. Layanan akses web memfokuskan kembali organisasi pada konsep arsitektur berorientasi layanan. meskipun sangat dapat

digunakan kembali, arsitektur longgar ditambah telah menjadi tujuan lapangan bagi banyak organisasi. Layanan akses web mendorong minat dan menyediakan teknologi untuk menerapkan arsitektur berorientasi layanan dalam satu visi.

3. Database yang mendukung ERP dalam lingkup pendidikan tinggi dengan layanan kebutuhan saat ini, dengan sistem panduan SOA dalam perubahan mengikuti teknologi yang cepat, sistem basis data harus telah terintegrasi, bersama, mudah dan otomatisasi.

REFERENSI

- Robby, Owen Kwanentent, Frans Mei Wardana; Analisis Dan Perancangan Basis Data Untuk Mendukung Aplikasi Erp Education Pada Universitas Bina Nusantara (Studi Kasus : Academic Management and Content Preparation), *Bina Nusantara University*
- Prasetyo, Hendro Joko; Implementasi Service Oriented Architecture (Soa) Menggunakan Teknologi Web Service; Hendrojogja@yahoo.com, *Fakultas MIPA Universitas Widya Dharma Klate*
- Sutanta, Edhy, *Basis Data Dalam Tinjauan Konseptual*, CV. Andi Offset, Yogyakarta, 2011;
- Choldun R, Muh. Ibnu; Perancangan Sistem Informasi Akademik Dengan Mengimplementasikan Erp; ibnucholdun@yahoo.co.uk; Laboratorium Sistem Informasi Politeknik Pos Indonesia; Jalan Sariasih 54 Bandung, 40151
- Jabr, Mohammed A. Al-Omari, Hussein K.; *e-Learning Management System Using Service Oriented Architecture*; 1Department of Computer Science, Amman Arab University for Graduate Studies, Amman, Jordan 2King Abdulaziz University for Science and Technology, Riyadh, Saudi Arabia
- Erl, Thomas; *Service-Oriented Architecture: Concepts, Technology, and Design*; Prentice Hall Professional Technical Reference, ISBN 0-13-185858-0, July 2005.
- Brown Paul C. Ph.D.; *Service-Oriented Architecture : A View From the Field Principal Software Architect*; TIBCO Software, 2006.

- Endrei, Mark, Ang,Jenni, etc; *Patterns: Service-Oriented Architecture and Web Services*; IBM Red Books, April 2004.
- Ahed Abugabah, Louis Sanzogni; Enterprise Resource Planning (Erp) System In Higher Education: A Literature Review And Implications; *International Journal of Human and Social Sciences* 5:6, 2010
- Chung,BooYoung; Skibniewski, Mirosław J.; Kwak, Young Hoon; Developing Erp Systems Success Model For The Construction Industry; *Journal Of Construction Engineering And Management*. © ASCE / March 2009
- Warnars, Spits; Tata Kelola Database Perguruan Tinggi Yang Optimal Dengan Data Warehouse; ISSN: 1693-6930_25, Fakultas Teknologi Informasi, Universitas Budi Luhur Jl. Petukangan Selatan, Kebayoran Lama, Jakarta 12260, 021-5853753,e-mail: spits@bl.ac.id
- Block, Jim; Service-Oriented Architecture: The Importance And Relevance Of Soa In The Financial Enterprise; Diebold Global Advanced Technology, January 2007
- Service Oriented Architecture; Implementation Frameworks; Fiorano Software, Inc. Copyright 2005.