

MODUL PEMBELAJARAN 1



ARSITEKTUR ENTERPRISE

Oleh : Vadlya Maarif, M.Kom.

STMIK NUSA MANDIRI

DAFTAR ISI

Lembar Cover	1
BAB 1. Konsep Dasar Arsitektur Enterprise	3
BAB 2. Arsitektur Enterprise Dan Instrumen Tata Kelola	9
BAB 3. Metode Dan Kerangka Kerja	17
BAB 4. Bahasa Arsitektur	25
BAB 5. Kompleksitas Arsitektur Dan Penggambaran Arsitektur Enterprise	35
BAB 6. Gambar, Model Dan Semantik	42
Daftar Pustaka	47

BAB 1

Konsep Dasar Arsitektur Enterprise

Arsitektur adalah konsep dasar sebuah sistem dalam lingkungannya terletak pada elemen- elemennya, hubungannya, serta prinsip dari rancangan dan evolusinya.

Arsitektur diperlukan untuk mengelola kompleksitas organisasi atau sistem yang sangat besar:

1. gambaran struktur organisasi
2. proses bisnis didalamnya,
3. dukungan aplikasi, dan
4. infrastruktur teknis

Apa itu Arsitektur Enterprise?

Seorang Arsitek perlu untuk mendiskusikan arsitektur dengan para stakeholder kemudian menjelaskan arsitektur serta dampaknya kepada semua stakeholder yang terlibat, yang biasanya memiliki latar belakang berbeda-beda.

Stakeholder

Seorang individu, kelompok, atau organisasi yang memiliki kepentingan/ketertarikan dalam, atau pertimbangan terhadap, sebuah sistem.

Enterprise Architecture (EA) adalah satu kesatuan yang koheren tentang prinsip-prinsip, metode dan model yang digunakan dalam desain dan realisasi dari struktur organisasi, proses bisnis, sistem informasi dan infrastruktur sebuah organisasi (Lankhorst, 2005).

Enterprise architecture mencakup inti-inti dari bisnis, TI dan evolusinya, tanpa arsitektur yang bagus, akan sulit untuk mencapai kesuksesan bisnis. Sebuah

enterprise architecture yang baik menyediakan kebutuhan mendatang untuk menyeimbangkan dan membantu translasi kebutuhan tersebut dari bentuk strategi perusahaan menjadi operasi sehari-hari.

Arsitektur berupa model, views, presentasi dan semua analisis yang mampu menjembatani “gap” antara arsitek dan stakholders.

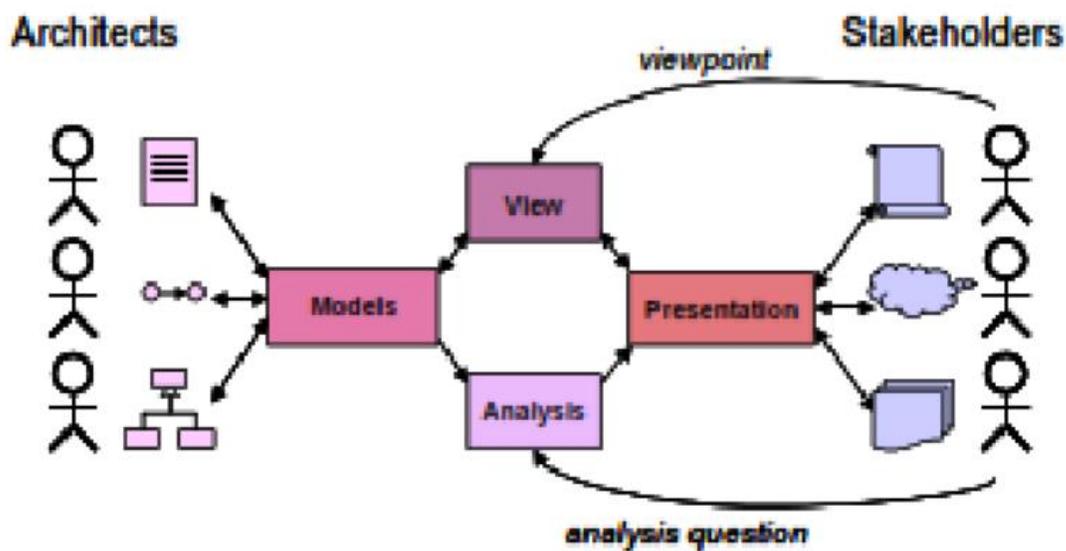


Fig. 1.1. Communicating about architecture.

Arsitektur Enterprise berupa cetak biru organisasi yang berisi proses bisnis, data, aplikasi dan infrastruktur IT, yang dirancang dan diterapkan secara terpadu untuk membantu berjalannya kegiatan organisasi dengan lebih efektif dan efisien.



Proses **BISNIS** dan aktifitas organisasi menggunakan ...



DATA, yang harus dikumpulkan, dikelola, diamankan dan didistribusikan menggunakan ...



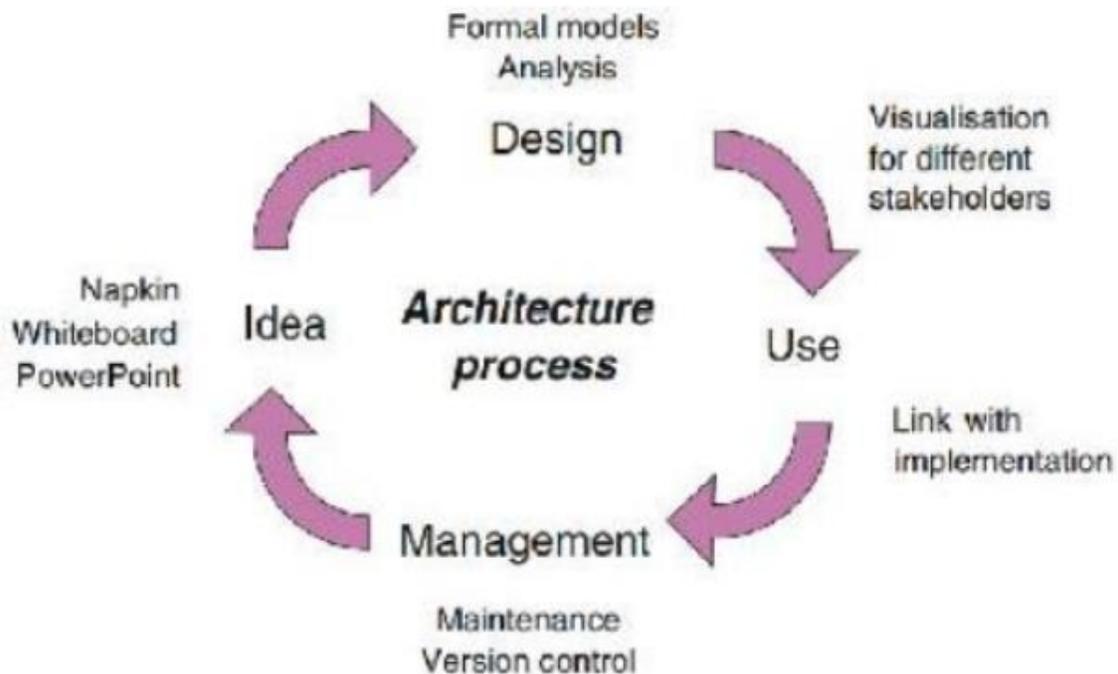
APLIKASI, baik itu aplikasi custom atau generic, yang berjalan di atas ...



TEKNOLOGI, seperti infrastruktur, sistem dan jaringan komputer

Proses Arsitektur

Proses arsitektur terdiri dari langkah-langkah yang membawa ide awal melalui tahapan perancangan dan implementasi menjadi sebuah sistem yang operasional



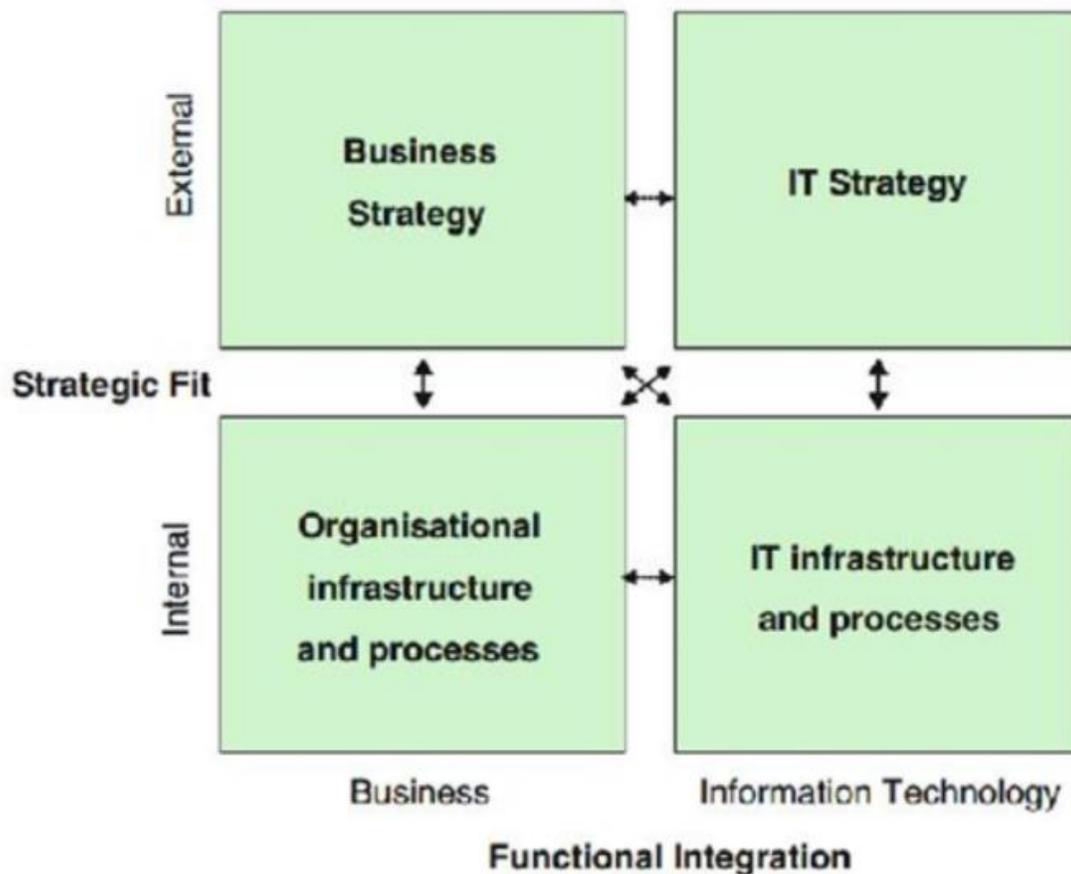
Dalam semua tahapan proses arsitektur, komunikasi yang jelas dengan dan antara stakeholder menjadi penting.

Faktor Pendukung Arsitektur Enterprise (Faktor Internal)

1. Penyesuaian Bisnis-TI dianggap sebagai instrumen penting untuk mewujudkan efektifitas organisasi

Model penyesuaian strategis oleh Henderson dan Venkatraman (1993) membedakan antara aspek strategi bisnis dan infrastruktur organisasi di satu sisi, dengan strategi dan infrastruktur TI di sisi lainnya.

Model ini menyediakan 4 sudut pandang dominan yang digunakan untuk mengatasi penyesuaian antara aspek2 tersebut.



2. Enterprise architecture diposisikan dalam konteks pengelolaan enterprise



Pada bagian atas piramida, kita harus melihat misi dari enterprise. Mengapa? Karena Visi dan misi menyatakan ‘gambaran masa depan’ dan nilai2 yang dipegang teguh.

Berikutnya adalah strategi, yang menyatakan jalur yang ditempuh enterprise untuk mencapai misi dan visinya ditranslasikan ke dalam tujuan konkrit yang memberi arah dan menjadi penanda dalam mengeksekusi strategi.

Tujuan ditranslasi menjadi perubahan konkrit hingga ke operasi sehari-hari perusahaan adalah tugas enterprise architecture dalam menyediakan sudut pandang holistik dari operasi saat ini dan saat nanti, serta aksi yang harus dilakukan untuk mencapai tujuan perusahaan.

Arsitektur dipandang sebagai bagian ‘hard’ dari perusahaan, sementara bagian ‘soft’ nya adalah culture yang dibentuk oleh orang-orang dan kepemimpinan (atau lebih penting) untuk mencapai tujuan perusahaan.

Sedangkan bagian bawah piramida adalah operasi harian perusahaan yang mencakup product, proses, people dan IT yang digunakan.

3. Arsitektur adalah instrumen strategis dalam memandu organisasi menjalani arah pengembangan yang terencana.

Enterprise yang sukses menjalankan ‘operating model’ dengan pemilihan yang jelas pada level integrasi dan standarisasi proses bisnis di seluruh enterprise.

Peran enterprise architecture sebagai pengorganisasian logis dari proses bisnis dan infrastruktur TI, harus mencerminkan kebutuhan integrasi dan standarisasi dari operating model.

Faktor Pendukung Arsitektur Enterprise (Faktor Eksternal)

Framework pengaturan menuntut agar perusahaan dan institusi pemerintah dapat membuktikan bahwa mereka memiliki pandangan yang jelas terhadap operasi mereka dan bahwa mereka telah mematuhi hukum yang berlaku.

Di USA, Clinger–Cohen Act (1996), juga dikenal sebagai Information Technology Management Reform Act, menuntut bahwa semua biro pemerintah harus memiliki sebuah IT architecture.

Framework lain Basel II (2004), didukung oleh gubernur bank sentral dan pimpinan otoritas pengawas perbankan negara2 Group of Ten (G10), menempatkan kebutuhan pada organisasi perbankan untuk memperhatikan financial risk management, untuk mendorong stabilitas di dunia keuangan.

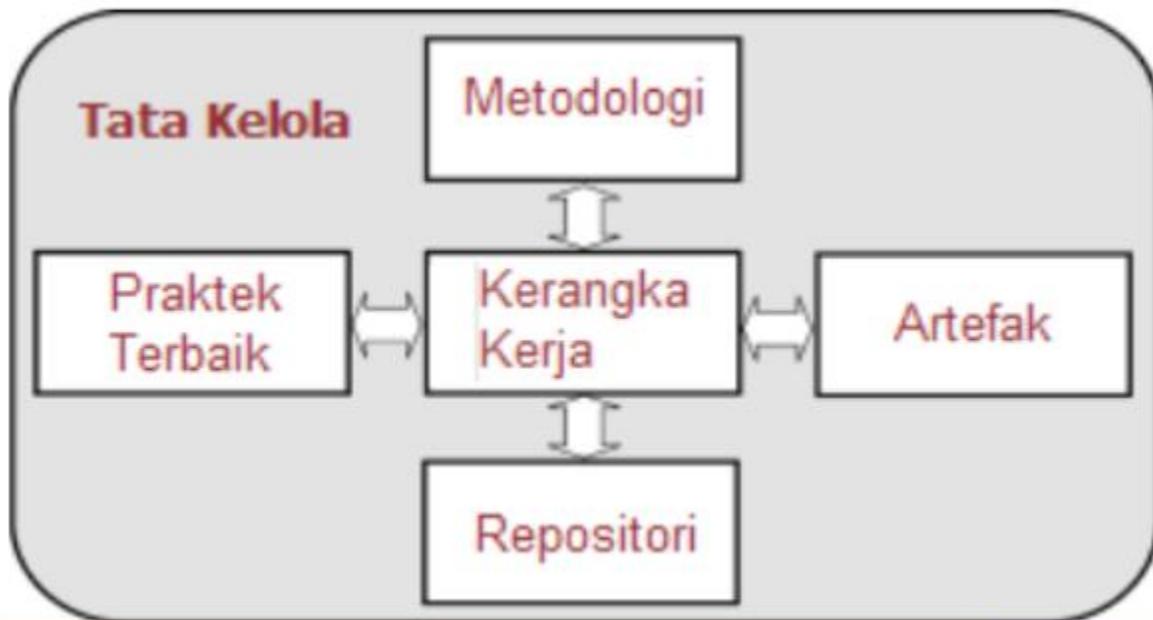
US act lainnya, the Sarbanes–Oxley Act (2002), dikenal sebagai Public Company Accounting Reform and Investor Protection Act, muncul setelah skandal Enron, untuk mendorong perusahaan mengadopsi praktek good corporate governance dan untuk membuat executive perusahaan bertanggung jawab secara personal.

BAB 2

Arsitektur Enterprise Dan Instrumen Tata Kelola

Pendekatan EA yang lengkap harus mencakup enam elemen inti, yang harus dirancang untuk bekerja bersama-sama:

1. Architecture Governance (Tata kelola arsitektur)
2. Architecture Framework (Kerangka kerja arsitektur)
3. Implementation Methodology (Metodologi Implementasi)
4. Documentation Artifacts (Dokumentasi Artefak)
5. Architecture Repository (Penyimpanan Arsitektur)
6. Associated Best Practices (Preaktek Terbaik yang terkait)



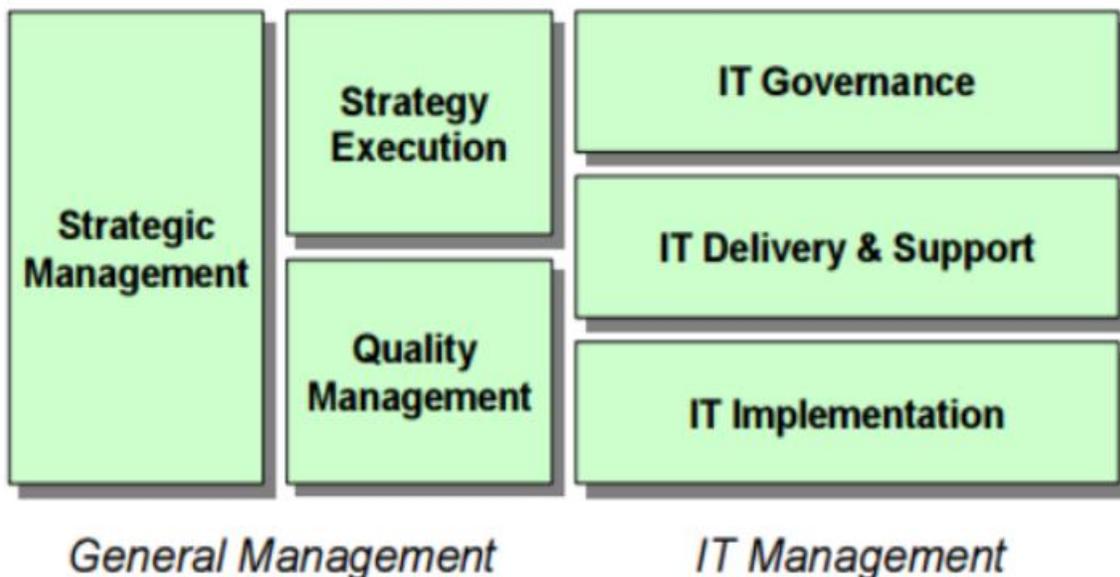
Elemen dasar pertama adalah “Governance” atau “tata kelola” yang mengidentifikasi perencanaan, pengambilan keputusan, dan proses pengawasan dan kelompok yang akan menentukan bagaimana Enterprise Architecture ini

dikembangkan dan dikelola – sebagai bagian dari tata kelola secara keseluruhan suatu perusahaan.

EA Governance atau tata kelola bertujuan untuk mendukung tata kelola yang terpadu, kelompok kebijakan manajemen yang terintegrasi dan proses yang membentuk struktur tata kelola secara keseluruhan.

Architecture Governance harus sesuai dengan tata kelola perusahaan secara keseluruhan maupun mekanisme dan struktur tata kelola TI yang ditetapkan.

Hubungan arsitektur enterprise dengan beberapa praktek manajemen



1. Strategic Management : Balanced Scorecard

Kaplan dan Norton (1992) memperkenalkan balanced scorecard (BSC) sebagai manajemen sistem yang membantu perusahaan untuk memperjelas dan melaksanakan visi dan strategi. Fokus manajemen berada di aspek keuangan dilengkapi dengan langkah-langkah mengenai kepuasan pelanggan, proses internal dan kemampuan untuk berinovasi.

Cara kerja pendekatan BSC :

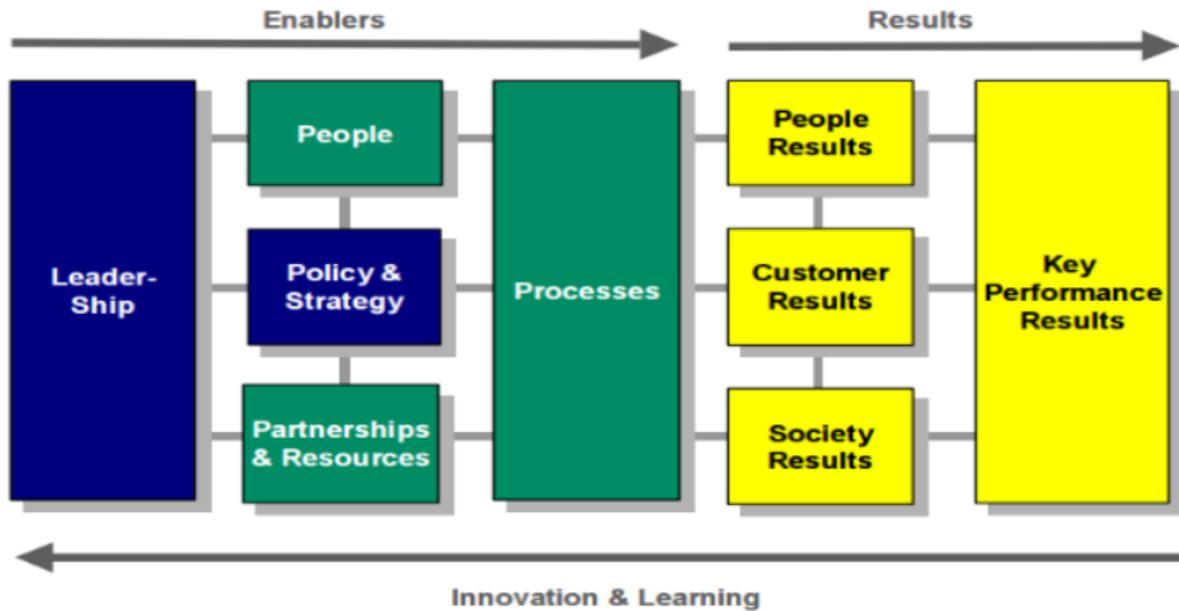
Sebuah perusahaan harus mendefinisikan misinya, tujuan dan langkah- langkah untuk setiap perspektif dan kemudian menerjemahkan ke sejumlah target dan inisiatif yang tepat untuk mencapai tujuan tersebut. Yang paling penting adalah gagasan dari dua loop umpan balik.

Pertama-tama harus mengukur output dari proses bisnis internal dan tidak hanya memperbaiki penyebabnya. Selain itu, pengukuran kinerja dan manajemen oleh fakta .

2. Strategy Execution : EFQM

Pendekatan manajemen penting lainnya adalah EFQM (Yayasan Eropa untuk Quality Management) Excellence Model (EFQM 2003). Model ini pertama diperkenalkan tahun 1992 sebagai kerangka kerja untuk menilai aplikasi The European Quality Award dan terinspirasi oleh Malcolm Baldrige Model di Amerika Serikat dan Hadiah eming di Jepang.

- Model EFQM memiliki cakupan yang lebih luas daripada ISO 9001 yaitu memberikan manajemen secara keseluruhan untuk kinerja seluruh organisasi.
- Model EFQM terdiri dari 9 kriteria unggulan, 5 diantaranya adalah “enabler”, meliputi apa yang dilakukan oleh sebuah organisasi dan 4 adalah “hasil” yang meliputi apa yang dicapai oleh organisasi itu.



- BSC membantu untuk membuat pilihan strategis dan model EFQM membantu dalam terus menerus memperbaiki yang diperlukan untuk melaksanakan strategi.

3. Quality Management : ISO 9001

ISO 9001 : 2000 standar (ISO 2000) dari Organisasi Internasional untuk menguraikan kriteria sistem manajemen mutu yang baik (QMS). Berdasarkan kebijakan mutu dan kualitas tujuan, sebuah perusahaan mendesain dan mendokumentasikan QMS untuk mengontrol bagaimana proses dilakukan.

Mulai dari umum, persyaratan keseluruhan, negara-negara standar tanggung jawab manajemen untuk QMS. Kemudian memberikan persyaratan untuk sumber daya, termasuk personil, pelatihan, fasilitas dan lingkungan kerja.

Tuntutan pada apa yang disebut 'realisasi produk' yaitu proses bisnis yang menyadari produk atau jasa perusahaan adalah inti dari standar.

Arsitektur enterprise dari perspektif manajemen mutu di umum dan ISO 9001b pada khususnya. Terlihat kontribusi utama dalam terpadu desain, manajemen dan dokumentasi proses bisnis dan mendukung sistem TI.

Sebuah arsitektur enterprise yang dirancang dengan baik dan didokumentasikan membantu organisasi agar sesuai dengan persyaratan ISO 9001 pada identifikasi proses dan dokumentasi.

Sebaliknya kebutuhan untuk SMM dapat mengarahkan perusahaan fokus ke inisiatif arsitektur, dengan menempatkan penekanan pada proses-proses dan sumber daya yang sangat penting untuk produk atau jasa yang berkualitas.

4. IT Governance : COBIT

COBIT (Control Objectives for Information and related Technology) standar untuk IT governance awalnya diterbitkan tahun 1996 oleh Audit Sistem Informasi dan Control Association.

Sebuah kerangka IT kontrol Internasional yang menyediakan organisasi dengan 'praktek yang baik' yang membantu dalam menerapkan struktur tata kelola TI di seluruh perusahaan.

Hal ini bertujuan untuk menjembatani kesenjangan antara risiko bisnis, kebutuhan kontrol dan masalah teknis.

Inti dari kerangka COBIT adalah tujuan pengendalian dan manajemen pedoman untuk 34 identifikasi proses TI yang dikelompokkan ke dalam empat domain : perencanaan dan organisasi, akuisisi dan implementasi, pengiriman dan dukungan, dan pemantauan.

COBIT menawarkan model kematangan untuk tata kelola TI yang terdiri dari lima tingkat kematangan :

- Ad Hoc : Tidak ada proses standar. Ad hoc pendekatan yang diterapkan pada kasus per kasus.
- Repeatable : Manajemen menyadari masalah. Indikator kinerja yang sedang dikembangkan, pengukuran dasar telah diidentifikasi karena memiliki penilaian metode dan teknik.

- Defined : Kebutuhan untuk bertindak dipahami dan diterima. Prosedur telah standar, didokumentasikan dan diimplementasikan.
- Managed : Pemahaman penuh akan masalah di semua tingkatan telah dicapai. Proses keunggulan dibangun pada pelatihan formal. IT sepenuhnya selaras dengan strategi bisnis.
- Optimised : Perbaikan terus menerus adalah ciri khas. Proses telah disempurnakan ke tingkat praktik terbaik eksternal berdasarkan hasil perbaikan terus menerus dengan organisasi lain.

5. IT Service Delivery and Support : ITIL

ITIL (IT Infrastructure Library) (Hanna et al. 2008) adalah himpunan yang paling banyak diterima praktek terbaik dalam domain pelayanan TI.

ITIL terdiri dari serangkaian dokumen yang memberikan bimbingan pada penyediaan baik IT layanan dan fasilitas yang dibutuhkan untuk mendukung TI. ITIL memiliki proses berorientasi pendekatan untuk manajemen layanan. Ini memberikan kode praktek yang membantu organisasi untuk membangun manajemen kualitas layanan TI dan infrastruktur di mana 'kualitas' didefinisikan sebagai 'sangat cocok untuk kebutuhan bisnis dan kebutuhan pengguna'.

Inti dari ITIL terdiri dari dua kelompok besar proses yaitu :

- Service Delivery : terdiri manajemen tingkat layanan, manajemen ketersediaan, manajemen keuangan untuk layanan TI, manajemen TI layanan darurat dan manajemen kapasitas.
- Service Support : meliputi manajemen masalah, manajemen insiden, layanan meja, manajemen perubahan, manajemen rilis dan manajemen konfigurasi.

ITIL adalah pelengkap dari COBIT. Tujuan pengendalian tingkat tinggi dari COBIT dapat diimplementasikan melalui penggunaan ITIL. COBIT merupakan tujuan pengendalian yang membantu apa yang harus dilakukan dan ITIL menjelaskan bagaimana melakukannya yaitu untuk mewujudkan tujuan tersebut.

6. IT Implementation : CMM and CMMI

The Capability Maturity Model untuk Software (Paulk et al. 1993) juga dikenal sebagai CMM dan SW-CMM adalah model untuk menilai kematangan organisasi proses rekayasa perangkat lunak dan menyediakan organisasi dengan praktik kunci yang diperlukan untuk membantu mereka meningkatkan kematangan proses.

Dalam model CMMI bentuk yang paling umum, ada lima tingkat kematangan, setiap lapisan di dasari untuk perbaikan proses yang berkelanjutan (CMMI Product Team 2002) :

- Initial : Proses biasanya ad hoc dan kacau. Organisasi tidak menyediakan lingkungan yang stabil. Keberhasilan dalam organisasi ini tergantung pada kompetensi dan heroik dari orang-orang dalam organisasi dan bukan pada penggunaan proses.
- Managed : Proyek-proyek organisasi telah memastikan bahwa persyaratan yang dikelola dan bahwa proses yang direncanakan, dilakukan, diukur dan dikendalikan. Namun proses mungkin sangat berbeda dalam setiap contoh spesifik, misalnya, pada proyek tertentu.
- Defined : Proses ditandai dengan baik dan dipahami, dan dijelaskan dalam standar, prosedur, alat dan metode. Standar ini digunakan untuk membangun konsistensi di seluruh organisasi. Proyek menetapkan proses mereka didefinisikan dengan menyesuaikan set organisasi proses standar sesuai dengan pedoman.

- Quantitatively Managed : Tujuan kuantitatif untuk kualitas dan proses kinerja ditetapkan dan digunakan sebagai kriteria dalam mengelola proses. Tujuan kuantitatif didasarkan pada kebutuhan pengguna pelanggan akhir, organisasi dan pelaksana proses.
- Optimising : Kinerja proses terus ditingkatkan melalui kedua perbaikan teknologi tambahan dan inovatif. Proses improvement kuantitatif bertujuan bagi organisasi yang telah ditetapkan, terus direvisi untuk mencerminkan perubahan tujuan bisnis dan digunakan sebagai kriteria dalam proses pengelolaan perbaikan.

BAB 3

Metode Dan Kerangka Kerja

Kebanyakan kerangka kerja arsitektur cukup tepat dalam membangun elemen apa yang harus menjadi bagian dari arsitektur enterprise. Namun untuk memastikan kualitas arsitektur enterprise selama siklus hidupnya adopsi kerangka tertentu tidak mencukupi. Hubungan antara berbagai jenis domain, pandangan atau lapisan arsitektur harus tetap jelas dan setiap perubahan harus dilakukan melalui metode. Untuk tujuan ini, beberapa metode yang tersedia yang dapat membantu arsitek melalui semua fase siklus hidup arsitektur.

Metode Arsitektur Enterprise

Sebuah metode arsitektur adalah kumpulan terstruktur teknik dan langkah-langkah proses untuk menciptakan dan mempertahankan arsitektur enterprise.

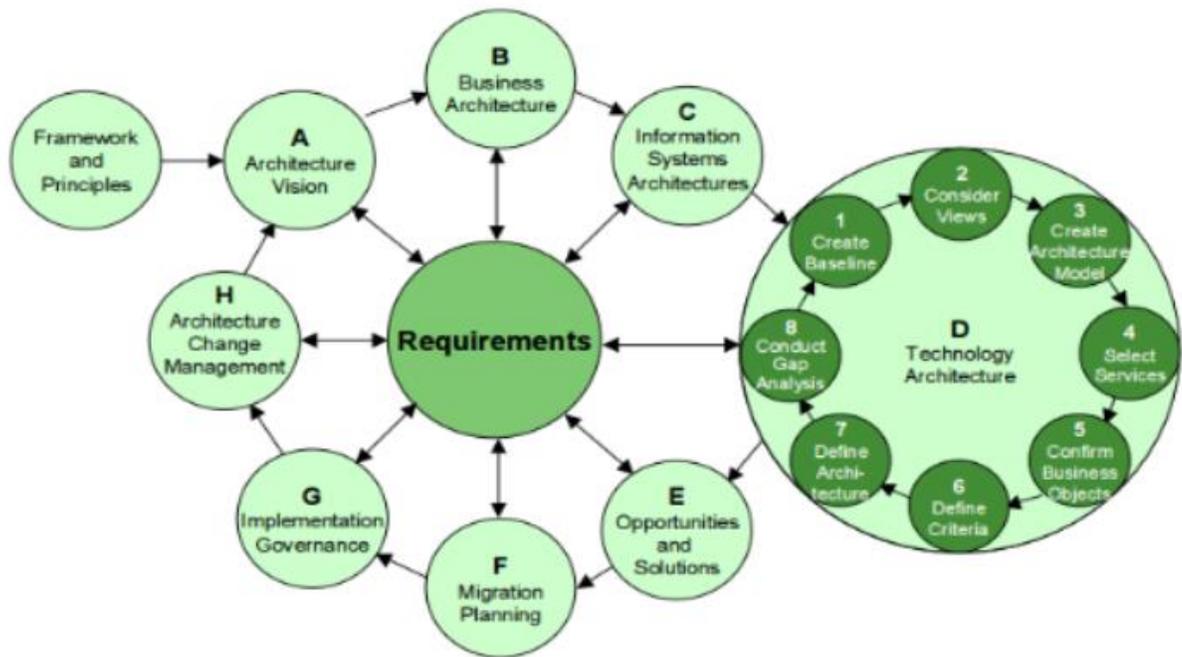
Metode biasanya menentukan berbagai tahapan siklus hidup arsitektur, apa yang harus diproduksi pada setiap tahap dan bagaimana mereka diverifikasi atau di tes.

Metode berikut yang layak untuk pengembangan arsitektur :

1. Rational Unified Process (RUP) (Jacobson et al. 1999) mendefinisikan proses berulang, seperti bertentangan dengan proses air terjun klasik, yang menyadari software dengan menambahkan fungsi untuk arsitektur pada setiap kenaikan. Perpanjangan terhadap perusahaan arsitektur TI diberikan oleh McGovern dkk (2004) dalam bentuk Enterprise Unified Process.
2. UN/Cefact Modelling Methodology (UMM) adalah bisnis tambahan proses dan model informasi metodologi konstruksi. Ruang lingkup adalah sengaja dibatasi untuk operasi bisnis, menghilangkan teknologi spesifik aspek. Business

Collaboration Framework (BCF) yang saat ini dalam pengembangan, akan menjadi spesialisasi dari UMM bertujuan mendefinisikan pertukaran informasi eksternal perusahaan dan bisnis yang mendasari kegiatan mereka. Lihat UN/CEFACT (2004).

3. TOGAF Architecture Development Method (ADM) dikembangkan oleh The Open Group menyediakan pentahapan rinci dan baik dijelaskan untuk mengembangkan arsitektur TI. Versi saat ini dari TOGAF (The Open Group 2011) menyediakan metode kerangka kerja dan pengembangan untuk mengembangkan arsitektur enterprise.



4. Chief Information Officers Council telah menciptakan The Federal Enterprise Architecture Framework (FEAF) disertai dengan panduan praktis dan berguna untuk mengembangkan arsitektur enterprise untuk organisasi pemerintah (CIO Council 2004). Inisiatif lain dari pemerintah AS termasuk Federal Enterprise Architecture (FEA) Federal Enterprise Architecture Program Management Office (FEAPMO 2004) dan Pengembangan Keuangan Arsitektur Proses oleh Departemen Keuangan (Treasury AS 2004).

The IEEE 1471-2000/ISO/IEC 42010 Standard

Pada tahun 2000, IEEE Computer Society disetujui IEEE Standard 1471-2000 (IEEE Computer Society 2000), yang membangun dasar teoritis yang kuat untuk definisi, analisis dan deskripsi dari arsitektur sistem. IEEE 1471, sejak itu telah terserap oleh 42.010 standar ISO/IEC (ISO/IEC/IEEE 2011), berfokus terutama pada sistem software intensif seperti sistem informasi, embedded system dan sistem komposit dalam konteks komputasi.

IEEE 1471 menggunakan sipil arsitektur metafora untuk menggambarkan arsitektur sistem perangkat lunak. Mirip dengan kerangka Zachman, meskipun tidak mencoba standarisasi arsitektur sistem dengan mendirikan sejumlah tetap atau sifat pandangan. IEEE 1471 juga tidak mencoba untuk standarisasi proses pengembangan arsitektur karena tidak merekomendasikan pemodelan bahasa, metodologi atau standar. Sebaliknya IEEE 1471 memberikan dalam hal sebuah ‘praktek yang disarankan’ nomor konsep berharga dan kerangka acuan, yang mencerminkan ‘yang berlaku umum tren dalam praktek untuk deskripsi arsitektur dan yang menyusun unsur- unsur yang ada konsensus.

	What?	How?	Where?	Who?	When?	Why?	
Planner							Scope (contextual)
Owner							Enterprise model (conceptual)
Designer							System model (logical)
Builder							Technology model (physical)
Subcontractor							Detailed representations (out of context)
	Data	Function	Network	People	Time	Motivation	

Pandangan (perspektif) pada Zachman Framework :

1. Planner (perencana) : Yang menetapkan objek dalam pembahasan, latar belakang, lingkup dan tujuan enterprise.
2. Owner (pemilik) : penerima atau pemakai produk/jasa akhir dari enterprise.
3. Designer (perancang) : perantara antara apa yang diinginkan (pemilik) dan apa yang dapat di capai.
4. Builder (pembangun) : pengawas/pengatur dalam menghasilkan produk/jasa akhir.
5. Subkontraktor : bertanggung jawab membangun dan merakit bagian-bagian dari produk/jasa akhir.
6. Functioning enterprise : wujud nyata dari produk/jasa akhir.

The Open Group Architecture Framework (TOGAF)

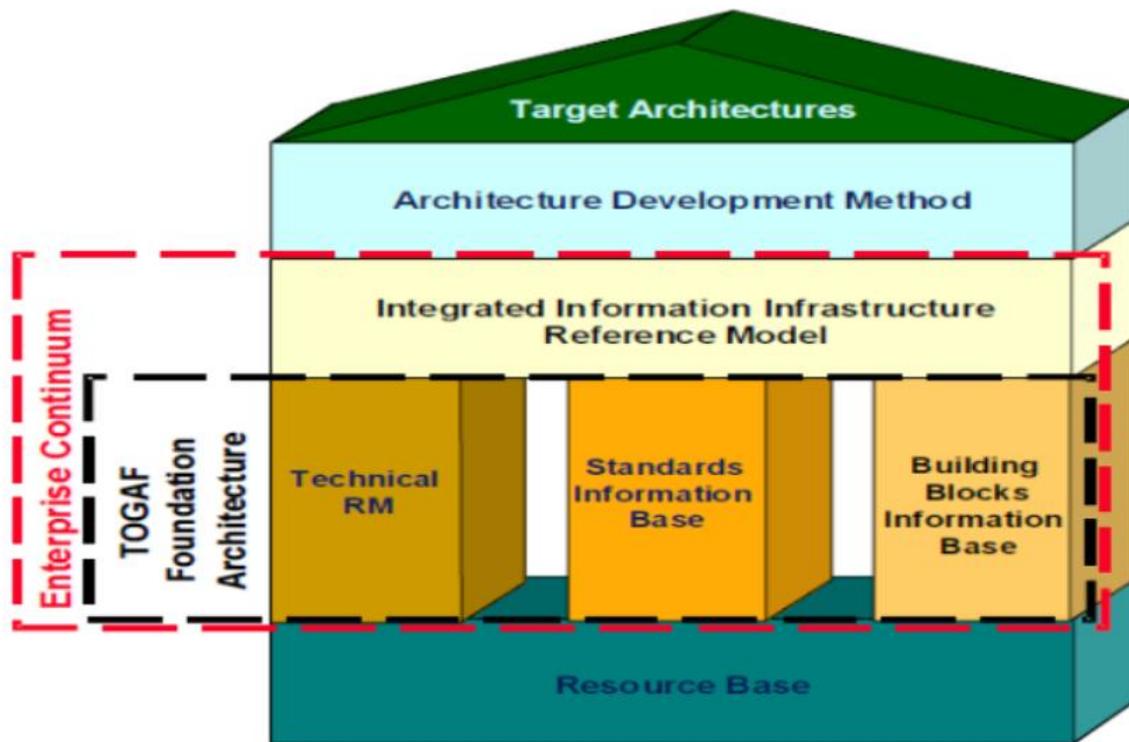
TOGAF berasal sebagai kerangka kerja umum dan metodologi untuk pengembangan arsitektur teknis, tetapi berkembang menjadi kerangka arsitektur enterprise dan metode. Dari versi 8 dan seterusnya, TOGAF (The Open Group 2011) didedikasikan untuk arsitektur enterprise.

TOGAF memiliki komponen utama berikut : Sebuah kerangka arsitektur kemampuan, yang membahas organisasi, proses, keterampilan, peran dan tanggung jawab yang diperlukan untuk membangun dan mengoperasikan fungsi arsitektur dalam suatu perusahaan.

Architecture Development Method (ADM) yang menyediakan jalan bekerja untuk arsitek. ADM dianggap inti dari TOGAF dan terdiri dari pendekatan siklik bertahap untuk pengembangan keseluruhan arsitektur enterprise.

Arsitektur Konten Framework yang menganggap suatu perusahaan secara keseluruhan arsitektur terdiri dari empat arsitektur yang saling terkait : Bisnis Arsitektur, data arsitektur, arsitektur aplikasi dan Teknologi (IT) Arsitektur.

The Enterprise Continuum terdiri dari berbagai model referensi seperti Teknis Referensi Model, Open Group Standards Information Base (SIB) dan The Building Blocks Information Base (BBIB). Ide di balik enterprise Continuum adalah untuk menggambarkan bagaimana arsitektur dikembangkan di seluruh kontinum mulai dari arsitektur dasar, melalui sistem umum arsitektur dan arsitektur industri spesifik untuk individu suatu enterprise arsitektur sendiri.

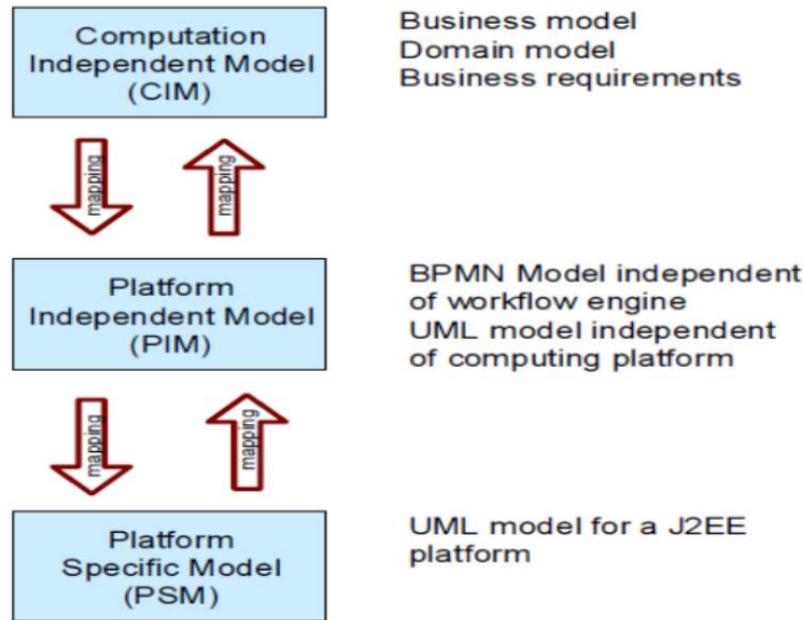


OMG's Model Driven Architecture (MDA)

Model Driven Architecture bertujuan untuk memberikan pemecahan masalah yang terbuka dalam integrasi sistem. Yang dapat meliputi sebuah program, komputer mandiri, kumpulan sistem atau sebuah enterprise.

MDA ingin menaikkan tingkat abstraksi di mana solusi perangkat lunak ditentukan dengan mendefinisikan kerangka kerja yang didukung oleh koleksi standar yang menetapkan standar untuk menghasilkan kode dari model dan sebaliknya.

Sekarang, MDA berbasis alat pengembangan perangkat lunak sudah mendukung spesifikasi perangkat lunak dalam UML bukannya dalam bahasa pemrograman seperti Java.



MDA terdiri dari tiga tingkat abstraksi dengan pemetaan antara lain :

1. Persyaratan untuk sistem dimodelkan dalam Computation Independent Model (CIM) menggambarkan situasi di mana sistem akan digunakan. Misalnya beberapa model kadang-kadang disebut model domain atau model bisnis. Menyembunyikan banyak atau sebuah informasi tentang penggunaan otomatis sistem pengolahan data.
2. The Platform Independent Model (PIM) menjelaskan pengoperasian sistem sementara menyembunyikan rincian yang diperlukan untuk platform tertentu. Sebuah PIM menunjukkan bahwa bagian dari spesifikasi lengkap yang tidak berubah dari satu platform yang lain.
3. Platform Specific Model (PSM) menggabungkan spesifikasi di PIM dengan rincian yang menentukan bagaimana sistem yang menggunakan jenis platform tertentu.

BAB 4

Bahasa Arsitektur

Saat ini, tidak ada bahasa pemodelan yang khusus ditujukan untuk menggambarkan arsitektur enterprise. Hanya di sub-domain seperti pemodelan proses bisnis dan desain perangkat lunak yang dapat ditemukan. Untuk pemodelan perangkat lunak, UML adalah bahasa dominan yang digunakan. Disini akan digambarkan sejumlah bahasa untuk pemodelan bisnis dan TI yang digunakan secara luas dalam menggambarkan bahasa arsitektur enterprise.

IDEF

IDEF adalah bahasa yang digunakan untuk melakukan pemodelan untuk perusahaan dan analisis. Awalnya dikembangkan oleh Program Angkatan Udara AS untuk Integrated Computer Aided Manufacturing (ICAM). Saat ini ada 16 metode IDEF. Terdiri dari IDEF0, IDEF3 dan IDEF1X('inti) yang paling umum digunakan.

Ruang lingkup IDEF meliputi:

1. Pemodelan Fungsional, IDEF0

Gagasan dibalik IDEF0 adalah untuk memodelkan elemen mengendalikan pelaksanaan suatu fungsi, para aktor menampilkan fungsinya, objek atau data dikonsumsi dan diproduksi oleh fungsi, dan hubungan antara fungsi bisnis (sumber daya bersama dan dependensi).

2. Pemodelan Proses, IDEF3

IDEF3 menangkap alur kerja bisnis proses melalui diagram alir proses. Ini menunjukkan urutan tugas untuk proses yang dilakukan oleh organisasi, logika keputusan, menggambarkan skenario berbeda untuk melakukan fungsi bisnis yang sama dan aktifkan analisis dan peningkatan alur kerja.

3. Pemodelan Data, IDEF1X

IDEF1X digunakan untuk membuat model data logis dan model data fisik dengan menggunakan diagram model logis. IDEF1X terdiri dari beberapa diagram bidang subjek logis dan beberapa diagram fisik.

IDEF banyak digunakan di industri. Ini menunjukkan bahwa itellah memenuhi kebutuhan pengguna dalam batas yang dapat diterima. IDEF taat pada proses pengembangan dan peningkatan berkelanjutan. Namun, IDEF0, IDEF1X, dan IDEF3 adalah bahasa yang agak stabil dan kaku, dan IDEF0 dan IDEF1X telah dipublikasikan sebagai standar dari National Institute of Standar dan Teknologi.

BPMN

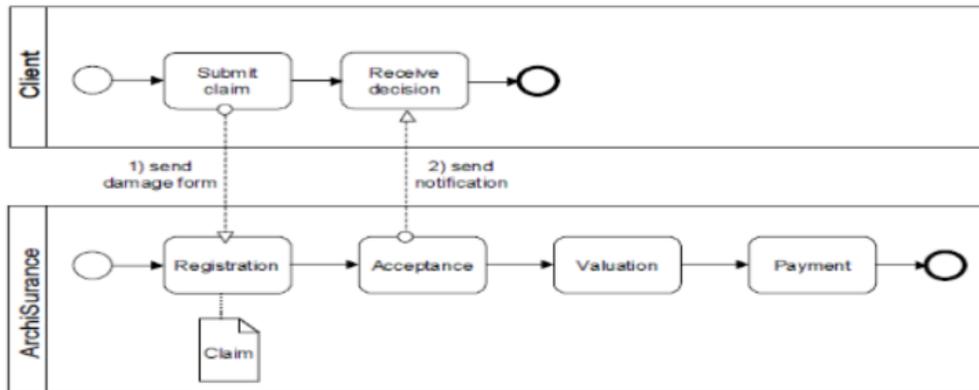
BPMN (Business process Modelling Notation) adalah salah satu standar yang sedang dikembangkan oleh Business Process Management Inittiative (BPMI).

BPMI adalah organisasi nirlaba yang mempunyai tujuan :

1. Spesifikasi standar terbuka untuk desain proses
2. Dukungan pemasok dan pengguna teknik dan alat manajemen proses bisnis.

BPMN standar (Object Management Group 2011) spesifik sebagai notasi grafis yang berfungsi sebagai dasar umum untuk berbagai proses bisnis peodelan dan eksekusi bahasa. Seperti namanya BPMN dibatasi untuk memproses pemodelan, aplikasi atau infrastruktur tidak tercakup oleh bahasa.

Contoh bisnis notasi proses yang telah ditinjau adalah Diagram Aktivitas UML, UML EDOC Proses Bisnis, IDEF, ebXML BPSS, Activity-Decision Flow (ADF), RosettaNet, LOVeM, dan Event Process Chains (EPC).



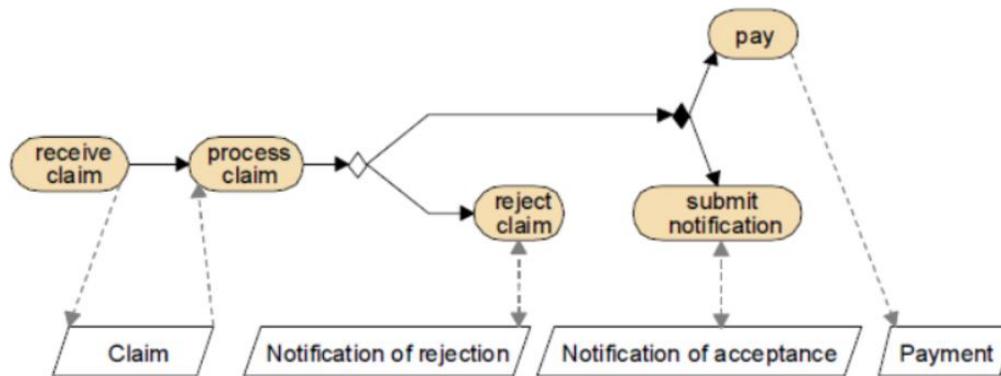
TESTBED

Tesbed adalah bahasa pemodelan bisnis dan metode awalnya dikembangkan oleh Telematika Institute bersama-sama dengan konsorsium perusahaan. Hal ini dimaksudkan untuk proses bisnis dan pemodelan organisasi serta pengguna target yang sebagian besar bisnis konsultan. Akibatnya bahasa tidak memiliki perspektif arsitektur sistem informasi dan konsep-konsep yang berkaitan dengan ini.

Testbed mengakui tiga domain aspek :

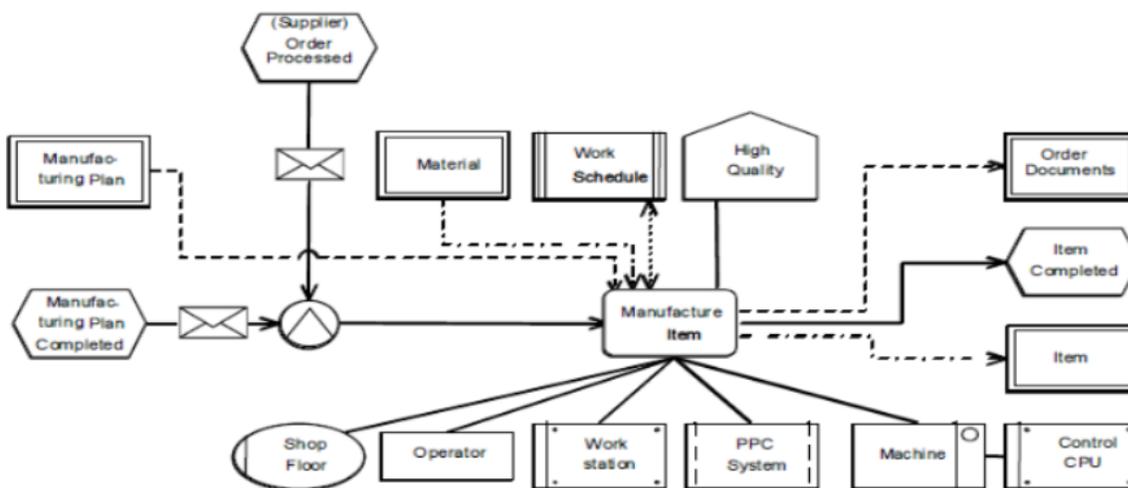
1. Domain aktor, yang menggambarkan sumber daya untuk melaksanakan kegiatan mereka.
2. Domain perilaku, yang menggambarkan proses bisnis yang dilakukan oleh sumber daya.
3. Domain item, yang menggambarkan objek data ditangani oleh proses bisnis.

Tiga domain di Testbed juga dapat dilihat sebagai spesifik jenis sudut pandang.



ARIS

ARIS (Architecture of Integrated Information Systems, Scheer 1994) adalah pendekatan terkenal untuk pemodelan perusahaan. Selain kerangka arsitektur tingkat tinggi, ARIS adalah metode pemodelan bisnis yang didukung oleh perangkat lunak. ARIS dimaksudkan untuk melayani berbagai keperluan : dokumentasi jenis proses bisnis yang ada, cetak biru untuk menganalisis dan merancang proses bisnis, dan dukungan untuk desain sistem informasi. Alat ini ditujukan untuk perancang sistem. Untuk memodelkan proses bisnis dalam model perusahaan, ARIS menyediakan bahasa pemodelan yang dikenal sebagai event- driven process chains (EPC). Sebuah EPC adalah grafik acara dan fungsi yang diurutkan. Ini menyediakan berbagai konektor yang memungkinkan eksekusi proses alternatif dan paralel.



UML (Unified Modeling Language)

UML (Unified Modeling Language) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek. Karena UML merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma object oriented. UML adalah salah satu tool/model untuk merancang pengembangan software yang berbasis object oriented. UML juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database dan komponen-komponen yang diperlukan dalam sistem software.

UML adalah sebuah bahasa standar untuk pengembangan sebuah software yang dapat menyampaikan bagaimana membuat dan membentuk model-model tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan software. UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman seperti Java, C++, Visual Basic atau bahkan dihubungkan secara langsung ke dalam sebuah object oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti : requirements, arsitektur, design, source code, project plan, tests dan prototypes.

ADL (Architecture Description Language)

Istilah ‘Architecture Description Language’ (ADL) merujuk kepada bahasa untuk menggambarkan arsitektur perangkat lunak dalam hal umum. Berbagai ADL ada dengan beberapa perbedaan dalam konsep yang tepat yang mereka tawarkan : beberapa berfokus pada aspek struktural arsitektur dan yang lain lebih memperhatikan untuk aspek dinamis. Secara umum, konsep-konsep mereka didefinisikan pada tingkat yang agak umum, meskipun mereka biasanya ditujukan

untuk pemodelan tingkat aplikasi. Keuntungannya adalah definisi yang tepat dan pondasi formal bahasa, yang memungkinkan mereka cocok sebagai bahasa yang mendasari untuk lebih spesifik.

Pada prinsipnya, konsep ADL yang cukup fleksibel untuk membuat model di beberapa domain. Namun, mereka terutama diterapkan dan yang paling cocok untuk aplikasi domain (yaitu untuk menggambarkan arsitektur perangkat lunak). Sebagai Acme (1998) ini diklaim cocok sebagai gambaran arsitektur umum dan bahasa interchange.

Konsep ADL adalah :

1. Komponen
2. Konektor
3. Sistem (Konfigurasi komponen dan konektor)
4. Port (titik interaksi dengan komponen)
5. Role (titik interaksi dengan konektor)
6. Representasi (digunakan untuk model komposisi hirarkis)
7. Rep-map (yang memetakan komponen atau konektor komposit arsitektur internal yang untuk elemen antarmuka eksternal)

Kesesuaian Untuk Enterprise Architecture

Pembahasan sebelumnya telah digambarkan bahasa saat pemodelan di bidang organisasi, proses bisnis, aplikasi dan teknologi. Hal ini jelas bahwa tidak satu pun telah berhasil menjadi 'bahasa' yang dapat mencakup semua domain. Secara umum ada sejumlah aspek yang hampir semua bahasa memiliki nilai rendah :

1. Hubungan antara domain (tampilan) tidak didefinisikan dengan baik, dan modelnyadibuat dalam berbagai tampilan tidak terintegrasi lebih lanjut.
2. Sebagian besar bahasa memiliki dasar formal yang lemah dan tidak memiliki semantik yang jelas.

3. Sebagian besar bahasa kehilangan visi arsitektur keseluruhan dan terbatas baik sub-domain bisnis atau aplikasi dan teknologi.

Berbeda dengan organisasi dan pemodelan proses bisnis yang tidak ada bahasa dominan, dalam pemodelan aplikasi dan teknologi UML telah menjadi standar dunia sejati. UML adalah pemodelan arus utamapendekatan dalam TIK, dan penggunaannya berkembang ke bidang lain. Inimenjadikan UML bahasa penting tidak hanya untuk pemodelan sistem perangkat lunak,tetapi juga untuk proses bisnis dan untuk arsitektur bisnis umum.Namun, UML tidak mudah diakses dan dimengerti oleh manajerdan konsultan bisnis; oleh karena itu, visualisasi dan pandangan khususmodel UML harus disediakan. Mengingat pentingnya UML, lainnyabahasa pemodelan kemungkinan akan menyediakan antarmuka atau pemetaan untuk itu.

Arsitektur Berorientasi Layanan

Munculnya paradigma komputasi berorientasi layanan (SOC) dan web teknologi jasa, khususnya telah menimbulkan minat besar dalam berorientasi layanan arsitektur (SOA). Mungkin karena hype seperti telah dibuat di sekitarnya, ada banyak kesalahpahaman tentang apa SOA sebenarnya. Berbagai layanan Web membuat kita percaya bahwa jika anda bisa membagi dunia ke dalam layanan pemohon, penyedia layanan dan registri layanan, anda akan memiliki SOA (misalnya, Ferris dan Farrell (2003). Lainnya menekankan bahwa SOA adalah cara untuk mencapai interoperabilitas antara komponen perangkat lunak yang di distribusikan dan heterogen, sebuah platform untuk komputasi terdistribusi (misalnya Stevens 2002).

Meskipun penemuan dinamis dan interoperabilitas manfaat penting dari layanan web, fokus murni teknologi akan terlalu terbatas dan akan gagal menghargai nilai konsep layanan. SOA merupakan set prinsip- prinsip desain yang memungkinkan unit fungsi yang akan diberikan dan dikonsumsi sebagai layanan. Hal yang menarik

adalah bahwa konsep layanan berlaku sama baiknya dengan bisnis seperti halnya untuk perangkat lunak aplikasi. Jasa menyediakan ‘unit bebas’ yang mewakili proposisi nilai dalam rantai nilai atau dalam proses bisnis. Konsep dasarnya sederhana ini dapat dan harus digunakan bukan hanya dalam rekayasa perangkat lunak, tetapi juga di semua tingkatan lain dari arsitektur enterprise untuk mencapai fleksibilitas dalam bisnis dan desain IT.

Konsep layanan adalah hasil dari pemisahan ‘eksternal’ dan ‘internal’ perilaku dari suatu sistem. Dengan demikian harus mandiri dan memiliki tujuan yang jelas dari perspektif lingkungannya. Perilaku internal di sisi lain, mewakili apa yang diperlukan untuk mewujudkan layanan. Untuk ‘konsumen’ dari layanan, perilaku internal sistem atau organisasi biasanya tidak relevan, mereka hanya tertarik pada fungsi dan kualitas yang akan diberikan.

Teknologi Berorientasi Layanan

Layanan web adalah teknologi yang relatif muda dalam pengembangan penuh, berkelanjutan oleh seperangkat standar industri yang berkembang pesat. Penerimaan luas mereka dijamin oleh status global organisasi seperti W3C, UN-CEFACT, OMG, The Open Kelompok, dan OASIS yang memimpin pekerjaan standarisasi di bidang ini. Masalah awal seperti keamanan, interoperabilitas dan keandalan sebagian besar telah diatasi dan ada pasar sangat kompetitif untuk teknologi layanan web.

Sebuah perkembangan paralel dalam orientasi pelayanan adalah kemampuan untuk mengakses sumber daya TIK, seperti daya komputasi, kapasitas penyimpanan, perangkat dan aplikasi sebagai layanan melalui Internet. Perkembangan ini berawal tidak hanya di lingkungan e-science tetapi juga memiliki potensi besar untuk berbagai area aplikasi seperti kesehatan, pendidikan, keuangan, ilmu kehidupan, industri dan hiburan. Idennya adalah bahwa pengguna atau perusahaan hanya dapat

plug ke dinding untuk mendapatkan akses ke komoditi layanan komputasi dan penyimpanan.

Penyediaan kemampuan ICT melalui Internet secara kolektif disebut Cloud Computing dengan Software as a Service (SaaS), Platform as a Service (PaaS) dan Infrastruktur as a Service (IaaS) sebagai kategori penting. Hal ini memberikan organisasi besar dan kecil akses ke sumber daya TIK dinyatakan keluar dari jangkauan dan memberikan keuntungan yang mengenai biaya dan skalabilitas. Integrasi tersebut harus menawarkan operasional keselarasan bisnis IT memberikan wawasan kinerja dan tingkat layanan real time. Perkembangan ini membuat kasus yang kuat untuk metode berorientasi layanan karena mereka menerapkan orientasi pelayanan secara real time manajemen pelayanan operasional yang memungkinkan layanan yang akan digunakan untuk online pengambilan keputusan dan pemecahan masalah.

Relevansi dan Manfaat untuk Arsitektur Enterprise

Orang mungkin bertanya mengapa kita harus fokus pada layanan untuk arsitektur enterprise dan dukungan TI? Apa yang membuat konsep layanan begitu menarik untuk perusahaan praktek arsitektur?

Pertama, ada fakta bahwa konsep layanan yang digunakan dan dipahami dalam domain yang berbeda yang membentuk suatu perusahaan.

Kedua, orientasi pelayanan memiliki efek positif pada sejumlah differentiators kunci dalam pasar yang kompetitif saat ini dan masa depan yaitu interoperabilitas, fleksibilitas, efektivitas biaya dan tenaga inovasi.

Tentu saja layanan web dan terbuka, standar berbasis XML terlampir digembar-gemborkan untuk memberikan interoperabilitas yang benar di tingkat teknologi informasi (Stevens 2002).

Oleh karena itu, layanan dapat digunakan oleh pihak-pihak yang berbeda dari yang awalnya dirasakan atau digunakan dengan menerapkan proses di berbagai tingkatan agregasi.

Perilaku internal dan eksternal memberikan dimensi baru dari fleksibilitas: Fleksibilitas untuk mengganti atau jasa pengganti dalam kasus kegagalan, fleksibilitas untuk mengupdate atau mengubah layanan tanpa mempengaruhi operasi perusahaan, fleksibilitas untuk mengubah pemasok layanan, fleksibilitas untuk menggunakan kembali layanan yang ada untuk penyediaan produk atau jasa baru. Ini akan menciptakan peluang baru untuk outsourcing, rendering lebih banyak kompetisi dan lebih efisien rantai nilai. Orientasi layanan memungkinkan juga mengadopsi strategi bottom-up, dimana proses bisnis hanya mekanisme untuk instantiate dan komersial mengeksplorasi layanan tingkat rendah ke dunia luar. Dalam pandangan ini, aset yang paling berharga adalah kemampuan untuk menjalankan layanan tingkat yang lebih rendah dan proses bisnis hanyalah sarana eksploitasi.

BAB 5

Kompleksitas Arsitektur Dan Penggambaran Arsitektur Enterprise

Pertemuan ini akan memaparkan ide-ide dan pilihan pendekatan dasar arsitektur enterprise dengan :

- Mengidentifikasi kebutuhan arsitek dalam desain, komunikasi, realisasi dan perubahan dalam enterprise arsitektur.
- Mendeskripsikan peranan utama model arsitektur dalam pendekatan, kegunaan model dalam komunikasi, hubungan antara model dan presentasinya.

Mengatasi Kompleksitas Arsitektur

Perusahaan telah lama menyadari perlunya pendekatan arsitektur yang terintegrasi. Namun, mereka masih mengalami kurangnya dukungan dalam desain, komunikasi, realisasi, dan manajemen arsitektur.

Beberapa hal yang harus dipenuhi dalam menghadapi siklus hidup arsitektur yaitu :

1. Design
2. Komunikasi
3. Realisasi
4. Change

Design

Saat mendesain arsitektur, arsitek harus menggunakan common, kosakata yang didefinisikan dengan baik untuk menghindari kesalahpahaman dan mempromosikan yang jelas. Kosakata seperti itu tidak boleh hanya fokus pada arsitektur tunggal domain, tetapi harus memungkinkan untuk integrasi berbagai jenis arsitektur terkait dengan berbagai domain. Di sebelah bahasa yang sama, arsitek harus didukung

dalam kegiatan desain mereka dengan menyediakan metodis dukungan, pedoman umum dan khusus organisasi, praktik terbaik, menggambar standar, dan cara lain yang mempromosikan kualitas dari arsitektur. Selanjutnya, untuk memudahkan proses desain, yang mana berulang dan membutuhkan perubahan dan pembaruan untuk arsitektur, dukungan untuk melacak keputusan arsitektur dan perubahan diinginkan.

Komunikasi

Arsitektur dibagikan dengan berbagai pemangku kepentingan di dalam dan di luar organisasi, mis., manajemen, perancang sistem, atau mitra outsourcing. Untuk memfasilitasi komunikasi tentang arsitektur, harus dimungkinkan untuk memvisualisasikan secara tepat aspek-aspek yang relevan untuk kelompok pemangku kepentingan tertentu. Sangat penting dalam hal ini penghormatan adalah untuk menghasilkan komunikasi yang sukses dalam hubungan antardomain berbeda yang dijelaskan oleh berbagai arsitektur (mis., proses vs. aplikasi), karena ini akan sering melibatkan banyak kelompok pemangku kepentingan. Komunikasi yang jelas juga sangat penting dalam hal outsourcing dari bagian implementasi arsitektur ke eksternal organisasi. Arsitek asli sering tidak tersedia untuk menjelaskan arti suatu desain, sehingga arsitektur harus berbicara untuk dirinya sendiri.

Realisasi

Untuk memfasilitasi realisasi arsitektur dan menyediakan umpan balik dari realisasi ini ke arsitektur asli, tautan harus ditetapkan dengan kegiatan desain pada tingkat yang lebih rinci, mis., bisnis desain proses, pemodelan informasi, atau pengembangan perangkat lunak. Perusahaan menggunakan konsep dan alat yang berbeda untuk kegiatan ini, dan hubungan dengan ini harus didefinisikan.

Selanjutnya, integrasi dengan yang ada alat desain dalam domain ini harus disediakan.

Change

Arsitektur sering kali mencakup sebagian besar organisasi dan mungkin terkait dengan beberapa arsitektur lainnya. Oleh karena itu, perubahan ke Arsitektur mungkin memiliki dampak mendalam. Menilai konsekuensinya perubahan tersebut sebelumnya dan dengan hati-hati merencanakan evolusi arsitektur karena itu sangat penting. Sampai sekarang, dukungan untuk ini sudah hampir tidak ada.

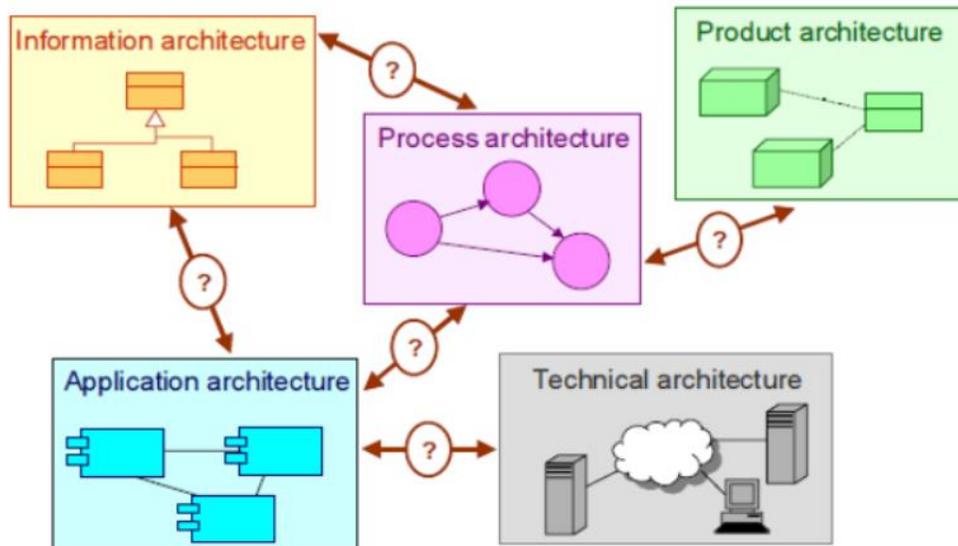
KOMPOSISI

Dalam praktiknya saat ini, enterprise arsitektur dalam perusahaan terdiri dari model yang berbeda-beda dan deskripsi yang heterogen, hal ini harus dapat dipahami oleh setiap stakeholder sesuai dengan porsinya masing-masing.

Pendekatan yang biasa digunakan untuk mengatasi kompleksitas sistem yaitu pendekatan komposisi yang membedakan antara bagian dari sistem dan hubungannya. Pendekatan komposisi berperan ketika ada beberapa sudut pandang dalam mendeskripsikan sistem lalu fungsi dalam arsitektur diuraikan dan fungsi bagian-bagiannya dan hubungannya.

Untuk memahami bagaimana fungsi mobil pertama-tama kita menggambarkan bagian-bagian mobil seperti mesin, roda, sistem pendingin udara, dan kemudian menggambarkan hubungan antara bagian-bagian ini. Selain itu juga memahami sistem informasi perusahaan sebagai seperangkat sistem dan hubungan mereka, dan memahami perusahaan mengatur proses bisnis dan hubungan mereka. Komposisionalitas juga memainkan peran sentral dalam pendekatan arsitektur. Sebagai contoh, standar IEEE 1471 mendefinisikan arsitektur sebagai dasar organisasi suatu sistem yang terkandung dalam komponen-komponennya, hubungan

mereka satu sama lain, dan ke lingkungan (bersama dengan prinsip-prinsip memandu desain dan evolusinya). Selain itu, komposisionalitas juga memainkan peran ketika berbagai sudut pandang pada suatu sistem didefinisikan. Tipe terakhir dari dekomposisi biasanya fungsional, dalam arti fungsionalitas suatu arsitektur didekomposisi dalam fungsi bagian- bagiannya dan hubungannya.



Integrasi Domain Arsitektur

Sebuah Gambaran :



Sebuah perusahaan telah memodelkan aplikasinya dalam UML dan proses bisnis di BPMN, dalam masalah ini tidak jelas bagaimana konsep dalam satu tampilan terkait dengan konsep tampilan lain.



Maka integrasi arsitektur akan sangat bermasalah sebab arsitektur dikembangkan oleh stakeholder dengan kepentingan masing-masing

Masalah pada contoh disebut juga **masalah keselarasan bisnis IT**. Untuk mengatasi masalah integrasi dalam domain arsitektur ini maka kita membutuhkan sebuah bahasa yang dapat mendeskripsikan domain-domain yang ada, juga sebagai prasyarat untuk menghubungkan berbagai alat yang digunakan dalam domain arsitektur.

Tujuan utama dari pendekatan ini adalah integrasi domain arsitektur, untuk menangani kompleksitas arsitektur sebagai suatu disiplin, dan untuk menyediakan wawasan untuk semua orang yang harus berurusan dengan arsitektur.

Penggambaran Arsitektur Enterprise

Untuk mengatasi kompleksitas dalam arsitektur enterprise, sebenarnya hanya berfokus pada relasi dan interaksi diantara domain arsitektur. Dengan definisi dan batasan yang tepat maka akan membantu dalam membangun wawasan kompleksitas dan menghindari konflik serta ketidak konsistensian antara domain, untuk itu digunakanlah model. Dalam konteks disini, model biasanya dipresentasikan dengan menggunakan bahasa grafis atau tekstual yang diformalkan. Terdapat bahasa terkait selanjutnya ialah mengenai :

1. Observing the universe
2. Concern
3. Memahami Domain
4. Tampilan dan Sudut Pandang
5. Cara kerja
6. Model Arsitektur Enterprise

Observing The Universe

Jika diibaratkan stakeholder adalah penonton dari sebuah organisasi, teknikal atau sistem lain yang membentuk perusahaan atas alam semesta yang diobservasi. Sedangkan konsepsi dari alam semesta tersebut adalah sebuah arsitektur enterprisenya, representasi dari arsitektur ini adalah deskripsi arsitektur yang dapat berisi model arsitektur, contoh, serta deskripsi tekstual.

Hubungan antara stakeholder, enterprise, arsitektur dan deskripsi arsitektur dapat digambarkan secara tetrahedron seperti pada gambar disamping yang berdasarkan pada FRISCO tetrahedron.

Concern

Dalam menyusun bagian enterprise, stakeholder biasanya akan terpengaruhi oleh beberapa fokus mereka masing-masing atau kepentingan masing-masing. Tetapi kepentingan juga bukan merupakan satu-satunya faktor yang mempengaruhi konsepsi stakeholder, lingkungan, kultur, pendidikan serta background masing-masing turut mengambil peran. Oleh karena itu ketika arsitek membuat pemodelan, mereka melakukannya dengan perspective tertentu.

Memahami Domain

Domain yaitu subset konsepsi dari alam semesta yang dipahami sebagai beberapa ‘bagian’ atau ‘aspek’ dari alam semesta. Dalam konteks pengembangan sistem informasi, kita akan mempelajari secara khusus model, yaitu abstrak dan kejelasan konsep dari domain. Sedangkan modeling adalah tindakan dalam mengabstrak bentuk model (apa saja yang dimiliki) dari apa saja yang dimiliki oleh alam semesta.

Tampilan dan Sudut Pandang

Gambaran mudahnya, tampilan merupakan apa yang kita lihat dan sudut pandang menunjukkan dari mana kita mencari tahu, misalnya dalam ‘sudut pandang keuangan’ menunjukkan bagaimana untuk menampilkan, apa saja, biaya itu membangun sebuah aplikasi tertentu.

Cara Kerja

Membuat dan menggunakan model arsitektur bisanya melibatkan beberapa ‘cara kerja’ terkait (Wijers dan Heijes 1990) :



Model Arsitektur Enterprise

Memiliki model tunggal pokok memungkinkan untuk menciptakan teknik yang kuat untuk menggambarkan dan menganalisa arsitektur enterprise, bahkan sekalipun model tersebut kurang lengkap dan tidak sepenuhnya konsisten.

Jika pemangku kepentingan membutuhkan informasi tentang aspek arsitektur mana saja yang memotong domain, maka akan ditutup bersama-sama secara manual dengan mengintegrasikan informasi dari berbagai sumber di domain tersebut.

Jadi misalkan ada model tunggal pokok dari sebuah arsitektur, maka pandangan arsitektur ini dapat dinyatakan sebagai proyeksi atau bagian dari model lain.

BAB 6

Gambar, Model Dan Semantik

Dalam penekanan akan pentingnya pemodelan bahwa ada suatu visualisasi arsitektur tersendiri terdapat pandangan berbeda yang berdasarkan sudut pandang para pemangku kepentingan dengan perhatian yang masing- masing berbeda, namun masih memiliki suatu tujuan bersama, hal ini disebut Semantik arsitektur.

Semantik tidak telah secara eksplisit diberikan, karena merupakan pemahaman yang tak terucapkan antara pengguna dari arsitektur. Dibeberapa bagian ilmu komputer, istilah ‘semantik’ dalam model ini sering digunakan untuk merujuk pada ‘efek’ dari sesuatu dimodel, dimana mengacu pada dinamika dalam model tersebut.

Model Simbolik Dan Semantik

1. Model Simbolik

Yaitu sifat arsitektur sistem yang mengandung simbol-simbol yang merujuk pada penjelasan nama jenis model. Peranan simbol-simbol sangat penting, karena akan sulit membicarakan sistem tanpa simbol. Jadi simbol digunakan sebagai media komunikasi dan penafsiran stakeholder.

Ketika para pemangku kepentingan merujuk pada arsitektur dan sistem, mereka dapat melakukannya hanya dengan menafsirkan simbol dalam model simbolik. Kami menyebutnya interpretasi model simbolik model semantik.

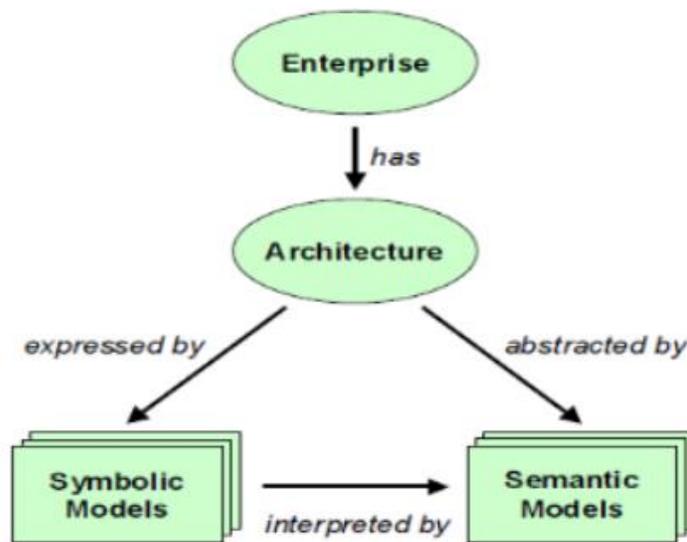
2. Model Semantik

Yaitu interpretasi dari sebuah model simbolik yang menerangkan makna di model simbolik. Model semantik adalah suatu abstraksi dari arsitektur.

Hubungannya dengan model simbolis biasanya lebih dipahami melalui operasi dan objek matematika. Sebagai contoh, signature formal semantik disediakan

oleh koleksi offsets (satu untuk setiap jenis signature) dan satu set hubungan dan fungsi antara mereka, satu untuk setiap hubungan simbol dan fungsi simbol di signature.

Masalah yang relevan dalam hubungan antara sistem dan semantik modelnya adalah bagaimana kita bisa menerjemahkan hasil sehingga kita bisa melakukan tes kasus untuk model simbolik. Ada berbagai cara di mana kita dapat memvisualisasikan hubungan di antara keduanya empat konsep utama perusahaan, arsitektur, model simbolik, dan model semantik.



Ada tiga pengamatan penting yang harus dilakukan di sini yaitu :

Pertama, empat konsep di atas dan hubungan mereka digunakan dalam rekayasa baik untuk informal serta model formal. Perbedaan yang relevan kami tekankan antara model simbolik dan semantik adalah perbedaan antara menggunakan simbol untuk merujuk pada realitas, dan abstraksi realitas yang hanya merujuk pada realitas dengan menafsirkan simbol- simbol model simbolik.

Kedua, arsitektur dapat diekspresikan oleh beberapa model simbolik, dan satu model simbolis pada gilirannya dapat ditafsirkan oleh beberapa semantik model.

Sebagai contoh, kita dapat mendefinisikan model semantik terpisah untuk kinerja dan untuk biaya sistem yang dinyatakan oleh satu simbol model, mis., di UML. Ketiga, dalam arsitektur seringkali dibuat pembedaan antara arsitektur semantik dan semantik formal bahasa pemodelan. Seperti yang dijelaskan, perusahaan yang dipertimbangkan dalam hal konsep arsitektur, yang ada di benak, misalnya, arsitek perusahaan. Konsep-konsep ini dapat direpresentasikan dalam model, yang diekspresikan dalam bahasa pemodelan.

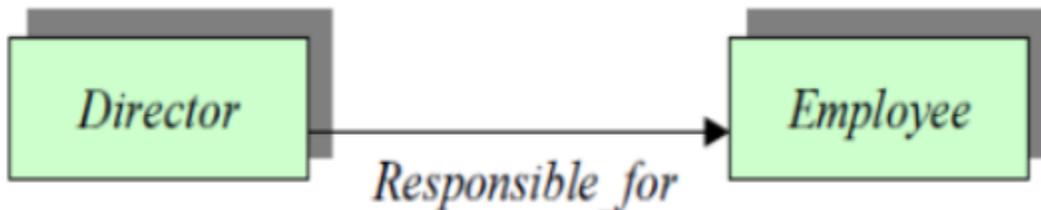
Semantik arsitektur adalah didefinisikan sebagai hubungan antara konsep arsitektur dan kemungkinannya representasi dalam bahasa pemodelan (Turner 1987). Untuk mengerti perbedaan ini, pertimbangkan diagram Venn. Mereka adalah struktur yang berguna untuk visualisasi bahasa logika Boolean, tetapi mereka bukan model diri. Model semantik mereka diberikan oleh penjelasan set-teoretis dari arti mereka. Semantik formal model atau bahasa, pada sisi lain, adalah representasi matematis dari sifat formal tertentu dari model atau bahasa itu. Semantik formal program komputer, misalnya, menyatakan kemungkinan perhitungan program itu. Berbeda cabang-cabang semantik formal ada, seperti denotasional, operasional, aksiomatik, dan semantik aksi. Harel dan Rumpe (2004) memberikan penjelasan yang jelas dari kebutuhan untuk secara ketat mendefinisikan semantik pemodelan bahasa.

Model Simbolik

Model simbolik adalah formalisasi satu atau lebih aspek arsitektur dari sistem konkret. Ini terdiri dari bagian-bagian arsitektur yang dapat dimodelkan secara matematis, sebagai lawan dari aspek yang lebih pragmatis dari arsitektur yang berkaitan dengan gagasan karakteristik seperti alasan, tujuan, dan rencana. Model

simbolik diekspresikan menggunakan bahasa deskripsi, representasi dari model yang sering bingung dengan interpretasinya.

Sebagai contoh, ungkapan $3 + 5$ dapat berarti alami angka, tetapi di sini hanya notasi untuk model sintaksis yang alami angka. Sebenarnya, bahasa deskripsi menggambarkan sintaksis keduanya struktur model dan notasinya, yaitu, kata-kata atau simbol digunakan untuk konsep-konsep dalam bahasa. Inti dari setiap model simbolik adalah ciri khasnya. Ini mengategorikan entitas dari model simbolik menurut beberapa nama yang terkait, linguistik atau dengan konvensi, untuk hal-hal yang mereka wakili. Hubungan antara entitas dari beberapa jenis dan operasi mereka juga dinyatakan sebagai simbol relasi dalam tanda tangan. Setelah hubungan telah ditentukan, mereka dapat digunakan dalam bahasa untuk membatasi lebih lanjut atau menganalisis sifat model simbolik.



Model Semantik

Makna formal dari model simbolik diberikan oleh semantik model, sebuah interpretasi dari model simbolik. Model semantik biasanya mengasumsikan adanya beberapa objek matematika (set, misalnya), digunakan untuk mewakili elemen dasar dari model simbolik. Operasi dan hubungan model simbol dipetakan ke operasi yang biasanya lebih dipahami dan hubungan antara objek matematika.

Sebagai contoh, semantik formal tanda tangan disediakan oleh koleksi set (satu untuk setiap jenis tanda tangan), dan satu set hubungan dan fungsi di antara mereka, satu untuk setiap simbol relasi dan simbol fungsi di tanda tangan.

Informasi hierarkis antara macam ditangkap oleh inklusi subset biasa, sedangkan informasi kontainmen dilambangkan dengan elemen-relasi yang biasa.

UML vs ArchiMate

Pendekatan ArchiMate dapat dikontraskan dengan pendekatan asli di UML. Dalam pendekatan ini, semantik secara eksplisit ditinggalkan keluar dari program. Orang yang menggunakan model bisa berkembang semantik untuk mereka, tetapi semantik umum tidak disediakan. Pendekatan ini juga berasal dari asal usul UML sebagai kombinasi dari tiga notasi yang ada yang tidak memiliki semantik formal. Karenanya, fokus dari UML dulu dan sekarang pada notasi, mis., Sintaks, dan bukan pada semantik. Meskipun beberapa diagram versi UML yang lebih baru memiliki formal semantik (lihat, mis., semantik kegiatan yang mirip dengan Petrinet diagram dalam UML 2.0), tidak ada semantik keseluruhan untuk seluruh bahasa. Kami tidak menempatkan notasi Pusat bahasa ArchiMate, tetapi lebih fokus pada makna bahasa konsep dan hubungan mereka. Tentu saja, bahasa pemodelan apa pun membutuhkan notasi dan kami menyediakan cara standar untuk menggambarkan Konsep Mate, tapi ini lebih rendah dari semantik arsitektur bahasa.

DAFTAR PUSTAKA

1. Desfray, Philippe & Gilbert Raymond (2014), *Modeling Enterprise Architecture with TOGAF*, Morgan Kaufmann
2. Steven Alter, (2002), *Information System: Foundation of e-Business*, Prentice Hall
3. Gustavo Alonso, Fabio Casati, et. al, (2004), *Web Services: Concepts, Architectures and Applications*, Springer
4. Glenn Hostetler, (2009), *Web Service and SOA Technologies* by Glenn Hostetler, Practicing Safe Tech
5. Thomas Erl, (2005), *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR