MODUL BAHAN AJAR

ARSITEKTUR ENTERPRISE

Di Susun oleh :

Elly Firasari, M.kom

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS NUSA MANDIRI

2023

KATA PENGANTAR

Dengan memanjatkan puji syukur ke hadirat Allah SWT akhirnya penulis menyelesaikan penulisan modul bahan ajar mata kuliah Arsitektur Enterprise dalam waktu yang telah di tentukan bagi Program Studi Teknologi Informasi Fakultas Teknik dan Informatika Universitas Nusa Mandiri

Penulis sangat berterima kasih yang sebesar-besarnya kepada :

1. Prof. Dr. Ir. Dwiza Riana, S.Si MM, M.Kom, IPU, ASEAN.Eng Selaku Rektor Universitas Nusa Mandiri
2. Anton, M. Kom Selaku Dekan Fakultas Teknologi Informasi Universitas Nusa Mandiri.
3. Rekan-rekan dosen Universitas Nusa Mandiri yang telah membantu dan memberikan saran dalam pembuatan modul ini.
4. Para Mahasiswa program studi Teknologi Informasi yang telah memberikan masukan guna kesempurnaan pembuatan modul ini.

Penulis berharap, modul ini dapat membantu mempermudah dalam proses belajar mengajar dan dapat dijadikan panduan oleh mahasiswa yang mengambil mata kuliah Arsitektur Enterprise.

Akhirnya, semoga penulisan modul ini dapat bermanfaat bagi kita semua.

Jakarta, Maret 2023

Penulis,

Pertemuan 1
Konsep Dasar Arsitektur Enterprise

Setiap organisasi memerlukan berbagai macam sistem informasi untuk mendukung kegiatan operasionalnya dengan memanfaatkan kemajuan teknologi informasi saat ini. Sebaiknya organisasi perlu melakukan perencanaan sistem informasi yang dibutuhkannya secara menyeluruh dan terintegrasi. Hal tersebut dimaksudkan untuk mencegah sistem informasi yang terpisah-pisah dalam suatu *enterprise* dan berpotensi untuk menyulitkan pimpinan dalam memperoleh informasi yang diperlukan.

Setiap organisasi perlu menyusun arsitektur *enterprise* berupa rencana induk sistem informasi, agar dapat digunakan untuk membangun sistem informasinya secara terencana, terarah, efisien dan terjadwal, sehingga dapat mendukung strategi bisnis organisasi.

Arsitektur memiliki definisi-definisi sebagai berikut:

1. Menurut *Software Engineering Institute* (www.sei.org), arsitektur merupakan struktur (gambaran terstruktur) dari tiap aktivitas.
2. Berdasarkan IEEE 1417-2000, arsitektur merupakan pengorganisasian yang fundamental dari suatu sistem, terdiri dari beberapa komponen. Relasi yang terjadi antar komponen dengan lingkungannya, serta prinsip-prinsip yang digunakan sebagai petunjuk dalam desain dan evolusinya.
3. Berdasarkan EIA (*Electronic Industry Assocation*), arsitektur merupakan komponen-komponen suatu sistem yang terdiri dari jaringan, perangkat keras dan lunak yang distrukturkan.
4. Berdasarkan ICH *Architecture Resource Center*, arsitektur merupakan rancangan keseluruhan jenis konstruksi baik fisik maupun konteks, nyata atau maya.

Aristektur juga dapat berarti perencanaan berupa cetak biru (*blueprint)* dari suatu struktur baik fisik maupun maya. Arsitektur diwujudkan dalam bentuk model dan gambar suatu komponen yang ditinjau dari berbagai sudut pandang.

Enterprise memiliki pengertian sebagai berikut:

1. Menurut Spewak, *enterprise* merupakan suatu organisasi atau badan lintas organisasi yang mendukung lingkup bisnis dan misi yang telah ditetapkan.
2. Menurut *Software Engineering Institute* (www.sei.org), *enterprise* merupakan

aktivitas yang memiliki suatu tujuan tertentu. Sekelompok orang dengan tujuan tertentu, yang memiliki sumber daya untuk mencapai tujuan tersebut.

3. Berdasarkan EIA (*Electronic Industry Association*), *enterprise* merupakan kumpulan organisasi yang memiliki beberapa tujuan/prinsip umum, dan/atau suatu garis dasar. *Enterprise* dapat berupa keseluruhan korporasi, divisi, dari suatu korporasi, organisasi pemerintah, departemen tunggal, atau suatu jaringan organisasi dengan geografis berbeda yang dikaitkan dengan tujuan tertentu.

Enterprise dapat berupa organisasi profit dan non-profit seperti pemerintahan dan institusi pendidikan. *Enterprise* juga dapat terdiri dalam berbagai ukuran kecil dan besar. Untuk ukuran kecil seperti kafe, toko. Sedangkan ukuran besar contohnya perusahaan tekstil atau perusahaan penerbangan.

Arsitektur *Enterprise* merupakan salah satu bagian dari sistem informasi, yang memiliki pengertian sebagai berikut:

1. Menurut Lankhorst, arsitektur *enterprise* didefinisikan sebagai prinsip-prinsip yang saling berkaitan, metode dan model yang digunakan untuk mendesain dan merealisasikan struktur organisasi, bisnis proses, sistem informasi dan infrastruktur perusahaan.
2. Minoli mendefinisikan arsitektur *enterprise* sebagai kumpulan dari bisnis proses, aplikasi, teknologi dan data yang mendukung strategi bisnis suatu perusahaan.
3. Menurut Osvald, arsitektur *enterprise* merupakan deskripsi dari misi *stakeholder* yang di dalamnya termasuk informasi, fungsionalitas/kegunaan, lokasi organisasi dan parameter kinerja. Arsitektur *enterprise* mendeskripsikan rencana pengembangan suatu sistem atau sekumpulan sistem.
4. Menurut *Federal Chief Information Officer Council*, arsitektur *enterprise* merupakan basis aset informasi strategis, yang menentukan misi, informasi dan teknologi yang diperlukan untuk melaksanakan misi, dan proses transisi untukmengimplementasikan teknologi baru sebagai tanggapan terhadap perubahan kebutuhan misi.
5. Menurut Roger Session, arsitektur *enterprise* merupakan deskripsi dari tujuan organisasi, bagaimana tujuan-tujuan ini direalisasikan oleh proses bisnis, dan bagaimana proses bisnis ini dapat lebih baik dilayani melalui teknologi.
6. Menurut John Ward dan Joe Peppard, arsitektur *enterprise* merupakan cetak biru pemetaan hubungan antar komponen dan semua orang yang bekerja di dalam

perusahaan secara konsisten untuk meningkatkan kerja sama/kolaborasi, serta koordinasi diantaranya.

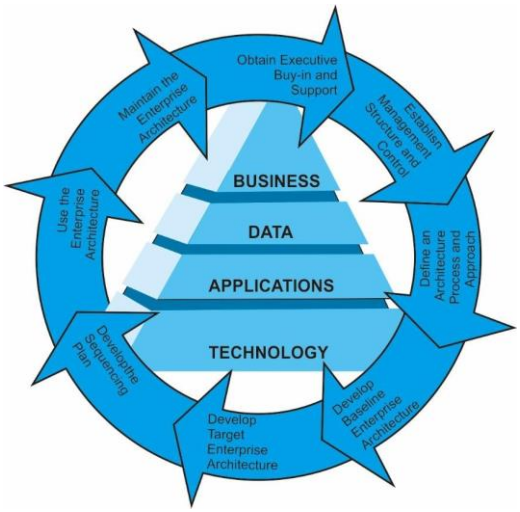
7. Menurut Bernard, arsitektur *enterprise* merupakan analisis dan dokumentasi keadaan saat ini dan keadaan masa depan suatu perusahaan dari perspektif strategi, bisnis, dan teknologi yang terintegrasi.

Arsitektur *Enterprise* = Strategy + Business + Teknologi

Berdasarkan definisi-definisi yang disampaikan sebelumnya, arsitektur *enterprise* merupakan deskripsi tentang struktur organisasi, yang terdiri dari proses bisnis, prinsip pengorganisasian dalam merespon perubahan dan mekanisme menyederhanakan proses bisnis sebagai suatu standarisasi model operasi suatu perusahaan. Arsitektur *enterprise* menjelaskan terminologi komposisi komponen organisasi, beserta korelasinya dengan lingkungan eksternal.

Arsitektur *enterprise* menyediakan prinsip-prinsip untuk memenuhi kebutuhan analisis, desain dan evolusi dari suatu organisasi. Sehingga dapat disimpulkan bahwa arsitektur *enterprise* merupakan deskripsi yang bersifat komprehensif, meliputi tujuan organisasi, proses bisnis, peran, struktur organisasi, perilaku organisasi, informasi bisnis dan teknologi informasi. Arsitektur *enterprise* mengidentifikasi komponen utama dari suatu organisasi dan mekanisme komponen di dalam sistem berfungsi secara bersama-sama untuk mencapai tujuan bisnis yang didefinisikan.

Komponen-komponen ini terdiri dari sumber daya manusia, proses bisnis, teknologi dan sumber daya lainnya.



Sumber: Federal Chief Information Officer Council (2007)

Dalam sudut pandang *enterprise*, organisasi sebagai suatu sistem dan masing-masing departemen adalah subsitem. Informasi tentang seluruh aspek organisasi disimpan dan dikelola secara terpusat dan dapat diakses oleh departemen lain yang membutuhkannya. Informasi yang bersifat transparan dalam suatu *enterprise* dapat mengakibatkan setiap departemen dapat mengetahui apa yang dikerjakan oleh departemen lain, dan bagaimana mendukung pekerjaan tersebut sehingga tujuan organisasi secara keseluruhan dapat dicapai.

A. PERAN ARSITEKTUR *ENTERPRISE*

Penerapan arsitektur *enterprise* menjadi penting terutama jika perusahaan menjadi semakin besar dan semakin kompleks. Arsitektur *enterprise* merupakan upaya untuk mengoptimalkan:

1. kontribusi sumber daya;
2. investasi teknologi informasi;
3. aktivitas pengembangan sistem.

Untuk mencapai tujuan kinerja perusahaan, misi organisasi dapat dicapai melalui kinerja optimal dari proses bisnis yang didukung teknologi informasi secara efektif dan efisien. Arsitektur *enterprise* dapat mengorganisasi dan memperjelas hubungan di antara tujuan strategis perusahaan, investasi, solusi bisnis dan peningkatan kinerja yang terukur. Untuk mencapai peningkatan kinerja sasaran, arsitektur *enterprise* harus terintegrasi dengan perencanaan strategis, perencanaan modal dan investasi.

Arsitektur *enterprise* dapat menyebabkan aliran informasi dan proses bisnis yang berlangsung dan sistem informasi yang diterapkan pada suatu perusahaan akan teridentifikasi. Dampak yang ditimbulkan jika suatu perusahaan tidak memiliki arsitektur *enterprise* antara lain:

1. memungkinkan terjadinya duplikasi sistem informasi;

2. antar sistem tidak saling terintegrasi;
3. pembiayaan kegiatan pemeliharaan menjadi kurang efisien.

Bernard mengemukakan bahwa arsitektur *enterprise* merupakan suatu praktik manajemen yang digunakan untuk meningkatkan kinerja suatu perusahaan dengan cara membuat perusahaan tersebut mampu secara keseluruhan mengintegrasikan strategi praktik-praktik bisnis, alur informasi dan sumber daya teknologi.

Sedangkan menurut Pearlson dan Saunders, arsitektur *enterprise* digunakan untuk mengatur logika seluruh organisasi, menentukan mekanisme teknologi informasi dapat mendukung proses bisnis.

Penerapan arsitektur *enterprise* memiliki manfaat sebagai berikut.

1. Informasi tentang misi, fungsi dan landasan bisnis dapat diketahui dalam bentuk yang relatif mudah dipahami sehingga dapat mendukung proses perencanaan dan pengambilan keputusan.
2. Mempercepat proses integrasi data dan sistem yang sudah ada maupun yang baru dibangun.
3. Dapat menjadi penghubung antara teknologi informasi dan bisnis suatu perusahaan.
4. Fokus pada penggunaan strategi teknologi informasi terhadap pengelolaan informasi perusahaan dan meningkatkan konsistensi, akurasi, integritas, kualitas, ketersediaan dan berbagi informasi pengelolaan sistem informasi perusahaan.

Arsitektur *enterprise* mendeskripsikan konektivitas antara tujuan organisasi dengan sistem informasi. Arsitektur *enterprise* memanfaatkan teknologi informasi untuk mendukung aktivitas operasi organisasi, dengan cara memanfaatkan informasi dan komponen perangkat lunak, serta melakukan pemilihan solusi dan teknologi baru secara efektif. Arsitektur *enterprise* merupakan *enabler* yang menyediakan model bisnis baru dengan memperhatikan pengaruh teknologi informasi terhadap bisnis. Arsitektur *enterprise* harus memiliki kemampuan untuk beradaptasi terhadap perubahan organisasi dan pertumbuhan pasar, bisnis, dan teknologi yang dinamis.

Arsitektur *enterprise* merupakan suatu proses sekaligus sebagai suatu produk yang menjamin bahwa sumber daya informasi yang dimiliki *enterprise* digunakan untuk mendukung strategi *enterprise*. Produk Arsitektur *enterprise* akan menjadi panduan bagi

pimpinan dalam mendesain proses bisnis dan panduan bagi pengembang sistem dalam membuat aplikasi yang selaras dengan tujuan dan kebijakan bisnis.

Arsitektur *enterprise* dipengaruhi oleh beberapa faktor antara lain:

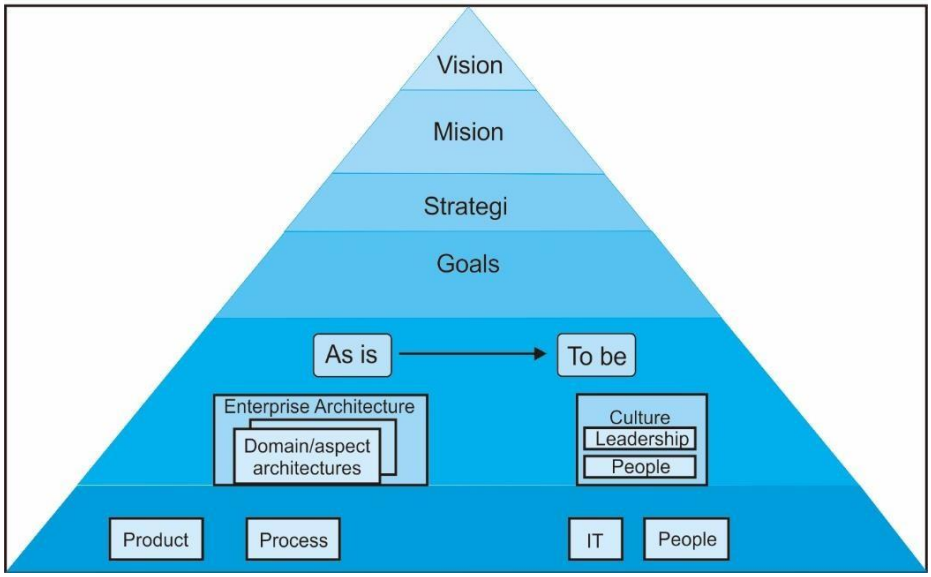
1. **Faktor internal,** efektifitas faktor internal digerakkan oleh adanya hubungan antar komponen *enterprise*. Tugas arsitektur *enterprise* menyediakan sudut pandang holistik dari kegiatan operasional saat ini dan masa depan, serta aksi yang harus dilakukan untuk mencapai tujuan perusahaan.

Aktivitas yang dilakukan pada suatu perusahaan akan mengacu pada visi yang kemudian diturunkan dalam bentuk misi perusahaan, kemudian diikuti dengan strategi perusahaan. Strategi perusahaan menyatakan langkah-langkah yang diambil untuk mencapai visi dan misi. Strategi diterjemahkan menjadi sasaran konkrit sebagai arahan dalam mengeksekusi strategi yang ada. Arsitektur *enterprise* berperan dalam menerjemahkan sasaran-sasaran tersebut menjadi kegiatan operasional. Arsitektur *enterprise* memberikan gambaran operasional yang ada saat ini dan rencana, beserta langkah-langkah yang harus diambil untuk mencapai tujuan perusahaan.

Beberapa pengertian visi menurut ahli antara lain sebagai berikut :

a. Menurut Kotler visi merupakan tujuan organisasi yang diekspresikan dalam produk dan pelayanan yang ditawarkan, kelompok masyarakat yang dilayani, nilai-nilai yang diperoleh serta aspirasi dan cita-cita masadepan
b. Menurut Robbin dan Judge visi merupakan strategi jangka panjang untukmencapai satu atau beberapa tujuan.

Visi menjelaskan secara singkat strategi bersaing perusahaan. Visi merupakansesuatu yang diharapkan untuk dimiliki dimasa depan (*what do they want to have*). Visi yang jelas memberikan fondasi untuk mengembangkan suatu pernyataan visi yang komprehensif. Visi menggambarkan aspirasi masa depan tanpa menspesifikasi cara-cara untuk mencapainya. Banyak organisasi memberikan pernyaataan visi dan misi, pernyataan visi harus dikemukakan terlebih dahulu. Pernyataan visi harus jelas dan sebaiknya singkat, visi yang efektif adalah visi yang mampu membangkitkan inspirasi. Jadi berdasarkan deskripsi di atas, visi merupakan suatu gambaran cita-cita yang ingin dicapai di masa depan.



Gambar 1.2

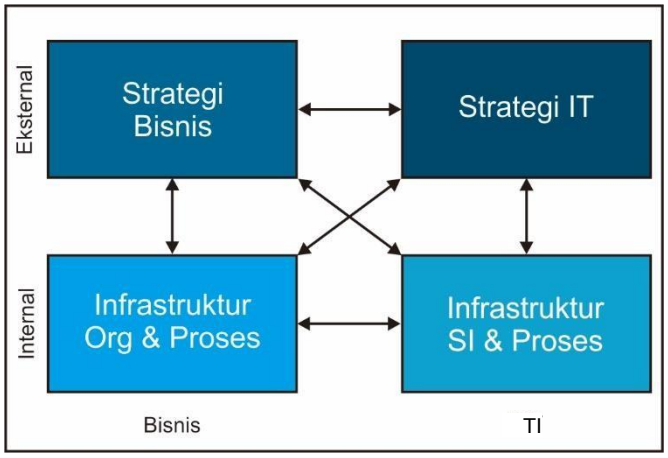
Arsitektur Enterprise sebagai Instrument Manajemen

Misi merupakan pernyataan yang harus dikerjakan oleh organisasi dalam usahanya mewujudkan Visi. Misi adalah bentuk yang diharapkan dimasa depan (*what do they want to be*). Misi merupakan pernyataan yang menegaskan visi melalui bentuk atau garis besar mekanisme yang akan diambil untuk sampai pada visi yang telah lebih dulu dirumuskan. Misi berisi tahapan-tahapan yang harus dilalui untuk mencapai visi. Misi bisa dikatakan sebagai penjabaran suatu visi.

Untuk mewujudkan visi dan misi, organisasi memerlukan strategi untuk mencapai tujuan dalam jangka panjang berupa tindakan potensial yang memerlukan keputusan manajemen tingkat atas dan sumber daya perusahaan dalam jumlah besar yang pada akhirnya juga mempengaruhi keberhasilan organisasi dalam jangka panjang. Menurut Ward dan Peppard strategi didefinisikan sebagai suatu tindakan-tindakan yang tergabung untuk meningkatkan keberhasilan dan kekuatan jangka panjang dari suatu perusahaan yang terkait dengan para pesaingnya. Sedangkan menurut Grant & Jordan strategi merupakan sarana untuk mencapai tujuan individu atau organisasi.

Faktor internal lainnya adalah keselarasan antara bisnis dan teknologi informasi dianggap sebagai instrumen penting dalam mewujudkan efektifitas organisasi. Model

Penyelarasan Strategis (*Strategic Alignment Model*) Henderson dan Venkatraman membedakan antara aspek strategi bisnis dan infrastruktur organisasi dengan strategi teknologi informasi dan infrastruktur teknologiinformasi.



Gambar 1.3

Strategic Alignment Model

Model Penyelarasan Strategis (*Strategic Alignment Model*) terdiri daribeberapa unsur, antara lain:

a. Infrastruktur sistem informasi dan proses
    1) Arsitektur teknologi informasi
    2) Pemrosesan teknologi informasi
    3) Kemampuan/keterampilan teknologi informasi
b. Infrastruktur organisasi dan proses
    1) Infrastruktur administratif organisasi
    2) Pemrosesan organisasi
    3) Kemampuan organisasi
c. Strategi teknologi informasi
    1) Ruang lingkup teknologi informasi
    2) Kompetisi sistemik
    3) Tata kelola teknologi informasi

d. Strategi bisnis

1) Ruang lingkup bisnis
2) Kompetisi yang berbeda
3) Tata kelola bisnis

2. Faktor eksternal, selain dari faktor internal yang lebih fokus pada suatu pendekatan agar strategi perusahaan dapat dieksekusi dengan efektif dan efisiensi dari sisi operasional, terdapat juga faktor eksternal yang mendorong perusahaan untuk menerapkan arsitektur *enterprise*. Faktor eksternal pada umumnya lebih dipengaruhi oleh regulasi yang ada. Contohnya penerapan sistem informasi di lembaga pemerintahan Indonesia dipengaruhi oleh Peraturan Presiden tentang Sistem Pemerintahan Berbasis Elektronik (SPBE). Hal tersebut menuntut agar setiap lembaga pemerintahan di Indonesia harus mempunyai arsitektur teknologi informasi. Dalam hal ini arsitektur teknologi informasi didefinisikan sebagai kerangka kerja yang terintegrasi untuk mengembangkan atau memelihara teknologi informasi yang ada dan memperoleh teknologi informasi yang baru untuk mencapai tujuan strategis organisasi.

Penerapan arsitektur *enterprise* akan menghasilkan blueprint, yang isinya merupakan rincian dinamis untuk arsitektur-arsitektur yang memanfaatkan proses dan kerangka yang terstruktur. Suatu blueprint mengandung rincian proses bisnis, informasi dan teknologi yang ada saat ini dan yang diusulkan organisasi untuk masa depan.

B. KOMPONEN ARSITEKTUR *ENTERPRISE*

Arsitektur *enterprise* memiliki empat komponen arsitektur utama.

1. **Arsitektur bisnis** mendefinisikan strategi bisnis, tata kelola, organisasi dan proses bisnis. Arsitektur bisnis merupakan arsitektur yang menggambarkan strategi, maksud, fungsi, proses, informasi dan aset bisnis untuk memberikan layanan bagi masyarakat, bisnis dan pemerintah. Arsitektur bisnis memberikan pemahaman umum tentang organisasi dan digunakan untuk menyelaraskan tujuan strategis dan panduan taktis. Arsitektur bisnis sebagai landasan bagi pengembangan dan implementasi rencana bisnis, teknologi dan penggunaan aplikasi. Rincian arsitektur bisnis mendukung pengambilan keputusan bisnis dengan menyediakan dokumentasi

tentang kondisi organisasi saat ini dan kondisi yang diharapkan pada masa depan. Dengan memiliki arsitektur bisnis yang baik maka diharapkan dapat mempermudah dalam mengambil keputusanyang tepat dan sesuai sehingga organisasi memiliki daya saing secara lebih efektif.

2. **Arsitektur data/informasi** mendeskripsikan struktur dari satu organisasi data logis dan fisik manajemen asset dan data sumber daya. Arsitektur informasi mendefinisikan struktur informasi bagi organisasi berupa entitas data dan hubungan antar entitas yang diperlukan dalam mendukung proses bisnis. Arsitektur informasi merupakan desain terstruktur dari lingkungan informasi bersama berupa data/informasi sebagai aset pendukung bisnis dan merupakan kebutuhan sistem aplikasi. Arsitektur informasi terdiri dari sekumpulan kebutuhan bisnis, proses, informasi.
3. **Arsitektur aplikasi** mendefinisikan jenis aplikasi utama yang diperlukan dalam mengelola data dan informasi untuk mendukung fungsi bisnis *enterprise*. Arsitektur aplikasi fokus pada pengembangan dan penerapan program aplikasi sebagai solusi atau layanan yang dibuat dan digunakan oleh suatu organisasi.
4. **Arsitektur teknologi** mendefinisikan platform teknologi yang diperlukan untuk menyediakan lingkungan yang mendukung aplikasi dalam mengelola data dan mendukung fungsi bisnis. Arsitektur teknologi mendeskripsikan perangkat lunak dan kemampuan perangkat keras yang diperlukan untuk mendukung penyebaran dari bisnis, data, dan jasa aplikasi. Ini meliputi infrastruktur teknologi informasi, *middleware*, jaringan, komunikasi, proses, standar dan sebagainya. Suatu pendekatan dalam menjelaskan struktur dan hubungan teknologi yang digunakan oleh suatu organisasi pada saat ini dan kondisi yang diharapkan untuk masa depan untuk meningkatkan daya saing perusahaan.

Gambar 1.4 Komponen Arsitektur *Enterprise*

C. METODE DAN KERANGKA KERJA ARSITEKTUR *ENTERPRISE*

Metode merupakan mekanisme untuk menjelaskan berbagai tahapan dari suatusiklus hidup arsitektur, menentukan sesuatu yang dihasilkan pada setiap tahapan dan cara melakukan verifikasi atau pengujian.

Metode arsitektur merupakan kumpulan terstruktur dari teknik dan langkah- langkah proses untuk membuat dan memelihara suatu arsitektur *enterprise*. Umumnya metodologi terdiri dari prosedur, teknik dan disiplin tertentu. Untuk menentukan ruang lingkup, batasan dan konten suatu arsitektur *enterprise* dapat menggunakan suatu kerangka kerja.

Kerangka kerja merupakan suatu struktur logis yang dapat diperluas untuk menggolongkan dan mengorganisasi serangkaian konsep, metode, teknologi dan perubahan pada suatu perancangan atau proses pengolahan. Menurut Zachman, kerangka kerja dapat dianggap sebagai dasar berpikir untuk mengelompokkan dan mengorganisasikan representasi suatu perusahaan yang penting bagi manajemen perusahaan dan pengembangan sistem selanjutnya. Kerangka kerja arsitektur *enterprise* adalah suatu model komunikasi untuk mengembangkan arsitektur *enterprise*. Kerangka kerja ini menampilkan kumpulan model, prinsip, pendekatan standar, konsep perancangan, komponen, visualisasi, dan konfigurasi yang memandupengembangan aspek spesifik arsitektur.

Kerangka kerja dapat memandu pemikiran arsitektur yang lebih luas, tidak

hanya yang ditampilkan pada suatu diagram. Kerangka kerja biasanya mengadopsi definisi arsitektur yang serupa tetapi berbeda dalam fokus, lingkup, dan tujuannya. Beberapa kerangka kerja yang umum digunakan antara lain adalah:

1. **Kerangka kerja Zachman,** dikeluarkan oleh *Zachman Institut for Framework Advancement* (ZIFA). Kerangka kerja ini menggambarkan arsitektur organisasisecara umum.
2. ***Architecture Development Method*** **(ADM)** dari TOGAF (*The Open Group Architecture Framework*) dikembangkan oleh The Open Group, menyediakan tahapan rinci dan jelas untuk mengembangkan suatu arsitektur teknologi informasi dan arsitektur *enterprise*.
3. ***Enterprise Architecture Planning*** **(EAP),** merupakan metode yang dikembangkan untuk membangun arsitektur *enterprise*. Tahapan pembangunan *Enterprise Architecture Planning* adalah tahap untuk memulai (permulaan), tahap memahami kondisi saat ini, tahap pendefinisian visi masa depan, dan tahap menyusun rencana dalam mencapai visi masa depan. *Enterprise Architecture Planning* diperkenalkan oleh Steven H. Spewak. Komponen dari metodologi *Enterprise Architecture Planning* menurut Spewak menggunakan dasar 2 baris atas dari kerangka kerja Zachman dalam hal ini tentang ruang lingkup dan model bisnis.

Hambatan pengembangan AE

- Tingkat usaha perencanaan jauh lebih komprehensif dan kaku daripada kebiasaan manajemen
- Kekurangan metodologi atau teknik untuk membuat perencanaan
- Keberhasilan tim AE tergantung pada budaya organisasi yang bersedia menerima konsep bahwa pengembangan perangkat lunak seharusnya bersifat *architecture driven*.

Karakteristik AE

- Mendukung organisasi/perusahaan dalam penyampaian informasi
- Memberikan pelayanan yang efektif dan tepat waktu
- Mendukung peningkatan fungsi dan bisnis organisasi/perusahaan
- Menghasilkan jalur bimbingan investasi masa depan
- Membantu menyelesaikan kesenjangan antara fungsi bisnis dan teknologi informasi dalam organisasi/perusahaan

Pertemuan 2

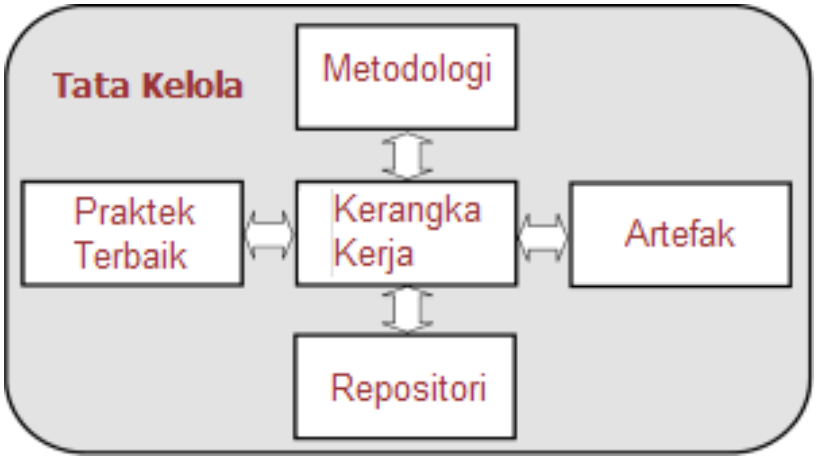
Arsitektur Enterprise Dan Instrumen Tata Kelola

Arsitektur enterprise biasanya digunakan sebagai instrumen dalam mengelola operasi harian perusahaan dan pengembangan di masa depan. Tapi bagaimana cocok

dengan praktik dan instrumen manajemen lainnya?

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

Hubungan arsitektur enterprise dengan beberapa praktek manajemen

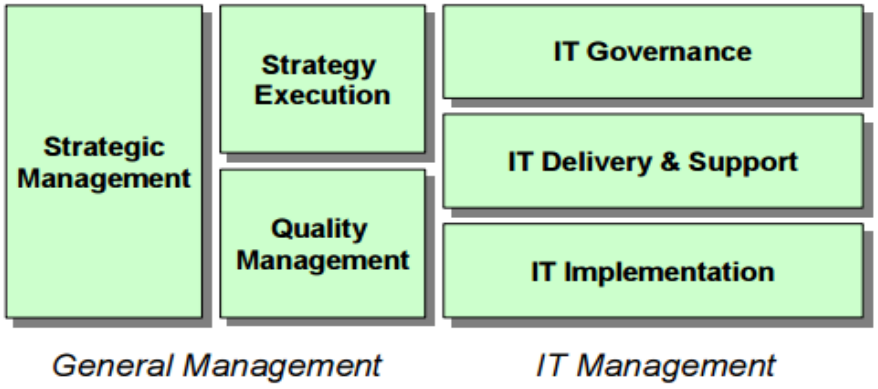


Fig. 2.1. Management areas relevant to enterprise architecture.

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, Kaplan dan Norton berpendapat bahwa langkah-langkah keuangan saja tidak memadai untuk memandu perkembangan masa depan suatu organisasi maka harus dilengkapi dengan langkah-langkah mengenai kepuasan pelanggan, proses internal dan kemampuan untuk berinovasi.

Oleh karena itu BSC menyarankan untuk melihat perusahaan dari empat perspektif. Perspektif Pelanggan bertanya bagaimana perusahaan terhadap pelanggannya, dengan ukuran seperti kepuasan pelanggan. Perspektif Keuangan difokuskan pada nilai bisnis yang dibuat oleh perusahaan, yang mencakup langkah-langkah seperti nilai pemegang saham. Proses Bisnis Internal perspektif melihat efektivitas dan efisiensi operasi internal perusahaan, memberikan perhatian khusus pada yang utama, berorientasi misi proses. Akhirnya, perspektif Pembelajaran dan Pertumbuhan membahas kemampuan perusahaan dan individu untuk berubah dan meningkat, yang sangat penting bagi organisasi apa pun.

Untuk masing-masing dari empat perspektif BSC mengusulkan struktur tiga lapis :

1. misi (contoh : Untuk menjadi pemasok pilihan pelanggan);
2. tujuan (contoh : Untuk menyediakan produk baru kepada pelanggan);
3. tindakan (contoh : Persentase omset yang dihasilkan oleh produk baru).

Agar BSC berfungsi, sebuah perusahaan harus terlebih dahulu mendefinisikan misinya, tujuan, dan langkah-langkah untuk setiap perspektif, dan kemudian menerjemahkannya ke dalam sejumlah target dan inisiatif yang tepat untuk mencapai tujuan ini. Yang penting dalam BSC adalah gagasan tentang umpan balik dua putaran.

Pertama-tama, seseorang harus mengukur output dari proses bisnis internal dan tidak hanya memperbaiki cacat pada keluaran ini tetapi juga mengidentifikasi dan memperbaiki penyebab cacat ini. Selain itu, umpan balik semacam itu juga harus dilemahkan untuk hasil dari strategi bisnis. Pengukuran kinerja dan manajemen sebenarnya adalah inti dari pendekatan BSC. Jika kita melihat peran arsitektur perusahaan sebagai instrumen manajemen, itu akan sangat berguna dalam Proses Bisnis Internal perspektif BSC. Banyak metrik operasional dapat dikaitkan dengan arsitektur perusahaan yang terdefinisi dengan baik dan berbagai analisis kinerja yang mungkin dilakukan. Namun, arsitektur enterprise memiliki penggunaan yang lebih luas. Dalam Perspektif Pembelajaran dan Pertumbuhan, kemampuan perusahaan untuk berkembang, mengantisipasi, dan merespons lingkungan yang berubah sangat penting. Untuk menentukan kelincahan organisasi, penting untuk menilai apa dampak dan kelayakan perubahan di masa depan. Analisis dampak suatu arsitektur perusahaan dapat membantu dalam penilaian semacam itu.

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 Baldridge 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.

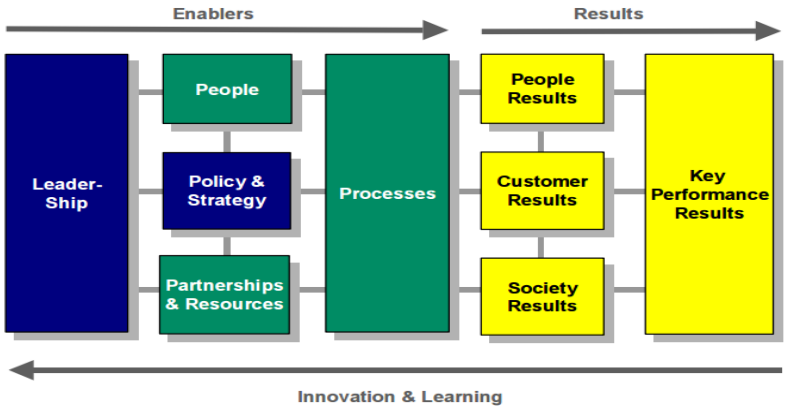


Fig. 2.2. The EFQM Excellence Model (EFQM 2003).

EFQM model menyediakan prinsip, tindakan, dan indikator untuk menilai kinerja-kinerja suatu perusahaan dalam semua aspek ini, dan pengukuran ini adalah dasar untuk pembelajaran berkelanjutan, inovasi, dan peningkatan. Semua ini juga menunjukkan perbedaan utama antara model EFQM dan BSC sedangkan yang terakhir difokuskan pada pengembangan strategi yang efektif manajemen, berkonsentrasi pada pengukuran dan pembandingan kinerja suatu organisasi sehubungan dengan sejumlah praktik terbaik.

Keduanya saling melengkapi: BSC membantu untuk membuat pilihan strategis, dan Model EFQM membantu dalam perbaikan berkelanjutan yang diperlukan untuk melaksanakan strategi.

Berdasarkan misi dan visinya, sebuah organisasi akan menentukan kebijakan dan strategi yang diperlukan untuk memenuhi kondisi saat ini dan kebutuhan masa depan serta harapan para pemangku kepentingan. Arsitektur enterprise adalah sebuah instrumen berharga dalam mengoperasionalkan dan mengimplementasikan kebijakan dan strategi. Pertama-tama, ia menawarkan wawasan tentang struktur dan operasi perusahaan secara keseluruhan dengan menciptakan pandangan luas tentang struktur organisasinya, proses bisnis, sistem informasi, dan infrastruktur. Tinjauan seperti itu sangat diperlukan ketika

merumuskan strategi yang koheren. Selain itu, arsitektur enterprise membantu dalam pengembangan, mengelola, dan mengomunikasikan standar operasi di seluruh perusahaan, diperlukan untuk memastikan bahwa kebijakan perusahaan benar-benar diterapkan. Akhirnya, dengan memberikan pemahaman yang lebih baik tentang dampak perubahan, bantuan yang sangat berharga dalam menciptakan jalan untuk masa depan, diperlukan untuk menilai dan melaksanakan strategi perusahaan jangka panjang.

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. Kemuadian 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.

Proses utama, yaitu, proses yang memengaruhi kualitas produk atau layanan, harus diidentifikasi dan didokumentasikan. Ini termasuk perencanaan, proses, desain, pembelian, dan kontrol proses yang berhubungan dengan pelanggan. Biasanya, persyaratan diberikan pada pengukuran, analisis, dan peningkatan proses bisnis ini. Setelah sistem kualitas diinstal, sebuah perusahaan dapat meminta audit oleh Panitera. Jika memenuhi semua kriteria, maka perusahaan akan terdaftar sebagai ISO 9001. Meskipun standar telah mendapatkan reputasi sebagai 'dokumen-berat', terutama berkaitan dengan versi sebelumnya yaitu 1987 dan 1994. Terlepas dari kritik-kritik ini, nilai bisnis dari SMM yang baik adalah diakui secara universal. Di Eropa, perusahaan industri semakin meningkat membutuhkan pendaftaran ISO 9001 dari pemasok mereka, dan penerimaan universal sebagai standar internasional.

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.
- Premis dasar COBIT adalah untuk memberikan informasi yang dibutuhkan organisasi untuk mencapai tujuannya, sumber daya TI perlu dikelola oleh serangkaian proses yang dikelompokkan secara alami.
- Inti dari kerangka COBIT adalah tujuan pengendalian dan manajemen pedoman untuk 34 identifikasi proses TI yang dikelompokan ke dalam empat domain : perencanaan dan organisasi, akuisisi dan implementasi, pengiriman dan dukungan, dan pemantauan.

Di sini, 'kontrol' didefinisikan oleh COBIT karena kebijakan, prosedur, praktik, dan struktur organisasi tidak ditandatangani untuk memberikan jaminan yang masuk akal bahwa tujuan bisnis akan tercapai dan bahwa peristiwa yang tidak diinginkan akan dicegah atau dideteksi dan dikoreksi. Tujuan kontrol dapat membantu mendukung tata kelola TI dalam suatu perusahaan. Misalnya, tujuan kontrol dari ‘Membantu dan memberi saran kepada IT proses pelanggan terdiri

dari mendirikan help desk, pendaftaran permintaan pelanggan, peningkatan permintaan pelanggan, pemantauan izin, dan analisis dan pelaporan tren.

Di samping kerangka kerja tujuan pengendalian, COBIT menyediakan hal yang kritis faktor keberhasilan untuk mencapai kontrol optimal atas proses TI, tujuan utama indikator, yang mengukur apakah suatu proses TI telah memenuhi persyaratan bisnisnya, dan indikator kinerja utama, yang menentukan ukuran seberapa baik kinerja proses TI dalam mencapai tujuannya.

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

1. Ad Hoc : Tidak ada proses standar. Ad hoc pendekatan yang diterapkan pada kasus per kasus.
2. Repeatable : Manajemen menyadari masalah. Indikator kinerja yang sedang dikembangkan, pengukuran dasar telah diidentifikasi karena memiliki penilaian metode dan teknik.
3. Defined : Kebutuhan untuk bertindak dipahami dan diterima. Prosedur telah standar, didokumentasikan dan diimplementasikan.
4. Managed : Pemahaman penuh akan masalah di semua tingkatan telah dicapai. Proses keunggulan dibangun pada pelatihan formal. IT sepenuhnya selaras dengan strategi bisnis.
5. Optimised : Perbaikan terus menerus adalah ciri khas. Proses telah disempurnakan ke tingkat praktik terbaik eksternal berdasarkan hasil perbaikan terus menerus dengan organisasi lain.

Menurut COBIT, arsitektur yang terdefinisi dengan baik adalah dasar dari suatu kebaikan lingkungan kontrol internal. Di banyak perusahaan, organisasi TI akan melakukannya bertanggung jawab untuk membangun dan memelihara arsitektur perusahaan. Sedangkan COBIT berfokus pada bagaimana seseorang harus mengatur TI (sekunder) fungsi suatu organisasi, arsitektur perusahaan berkonsentrasi pada (primer) bisnis dan struktur TI, proses, informasi dan teknologi perusahaan. Dengan demikian, arsitektur perusahaan membentuk komplemen alami untuk COBIT. Relatif terhadap tingkat kematangan COBIT, perusahaan arsitektur tentu saja akan paling relevan di tingkat atas. Pada tingkat

berikutnya, kesadaran pertama tentang nilai arsitektur mungkin muncul, tetapi biasanya tidak ada praktik arsitektur yang mapan di perusahaan. Hanya dari level yang ditentukan ke atas yang dikenali dan digunakan sebagai instrumen penting dalam merencanakan dan mengelola perkembangan TI dalam koordinasi dengan kebutuhan bisnis.

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 membangu 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 :

1. Service Delivery : terdiri manajemen tingkat layanan, manajemen ketersediaan, manajemen keuangan untuk layanan TI, manajemen TI layanan darurat dan manajemen kapasitas.
2. 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 sebagail 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) :

1. 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.
2. 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.
3. 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.
4. 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.
5. Optimising : Kinerja proses terus ditingkatkan melalui kedua perbaikan teknologi tambahan dan inivatif. Proses improvement kuantitatif bertujuan bagi organisasi yang telah ditetapkan, terus direvisi untuk mencerminkan perubahan tujuan bisnis dan digunakan sebagai kriteria dalam proses pengelolaan perbaikan.

Pertemuan 3

Metode Dan Kerangka Kerja

A. Pendahuluan

- 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.

B. 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 :

- 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.

- UN/Cefact Modelling Methodology (UMM)

  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).

- 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.

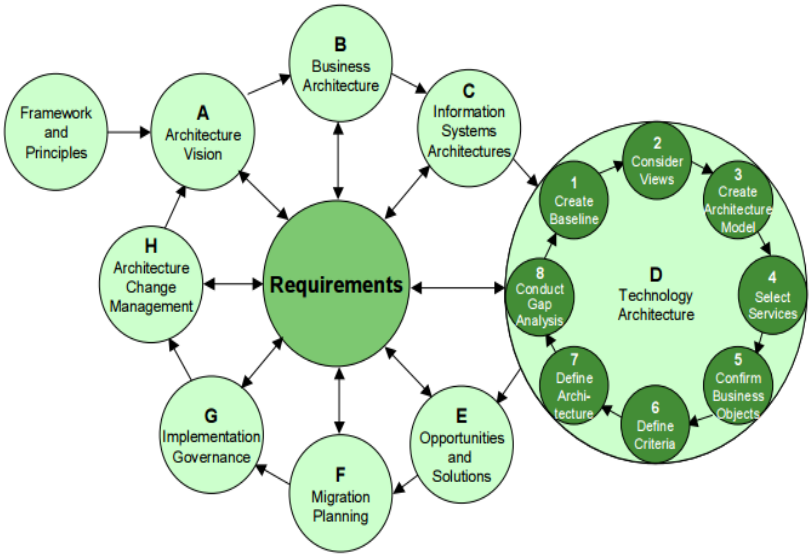


Fig. 2.3. TOGAF architecture development cycle (based on The Open Group 2002).

- Chief Information Officers Council telah menciptakan The Federal Enterprise Architekture 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).

C. 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.

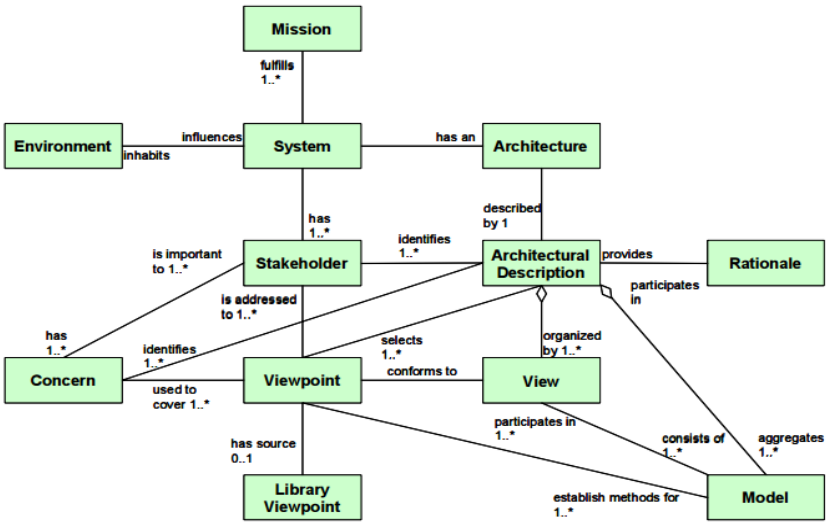


Fig. 2.4. Conceptual model of architecture description (based on IEEE Computer Society 2000).

IEEE 1471 juga menyediakan sejumlah sudut pandang arsitektur yang relevan bersama-sama dengan spesifikasi mereka dalam hal keprihatinan, bahasa dan pemodelan dan metode analisis. Hal ini penting untuk dicatat bahwa deskripsi arsitektur yang compliant dengan IEEE 1471 dapat digunakan untuk memenuhi persyaratan standar lainnya, seperti Reference Model of Open Distributed Processing.

D. The Zachman Framework

Pada tahun 1987, John Zachman memperkenalkan arsitektur enterprise pertama dan paling terkenal, meskipun saat itu disebut 'Kerangka Sistem Informasi Arsitektur'. Kerangka yang berlaku untuk perusahaan adalah hanya struktur logis untuk mengklasifikasikan dan mengorganisir deskriptif representasi dari suatu perusahaan yang signifikan terhadap pengelolaan perusahaan serta pengembangan sistem perusahaan. Kerangka Zachman didasarkan pada prinsip-prinsip arsitektur klasik yang membentuk persamaan umum. kosa kata dan seperangkat perspektif untuk menggambarkan sistem perusahaan yang kompleks. Kerangka kerja ini tidak memberikan panduan mengenai urutan, proses, atau implementasi, namun berfokus pada memastikan bahwa semua pandangan telah mapan, memastikan sistem yang lengkap terlepas dari urutan kemunculannya.

	Data ***(What)***	**Fungsi** ***(How)***	**Jaringan** ***(Where)***	**Orang** ***(Who)***	**Waktu** ***(When)***	**Motivasi** ***(Why)***
Tujuan/ Cakupan (Perspektif Perencana)	Daftar hal-hal yang penting bagi *enterprise*	Daftar proses bisnis yang dilakukan *enterprise*	Daftar lokasi operasional *enterprise*	Daftar unit organisasi	Daftar kejadian /siklus bisnis	Daftar tujuan/ strategi bisnis
Model Bisnis (Perspektif Pemilik)	Diagram Relasiantar Entitas	Model proses bisnis (diagramalur data fisik)	Konfigurasi jaringan (nodedan link)	Struktur organisasi (peran, kumpulan keahlian, kebutuhan keamanan)	Jadwal bisnis induk	Perencanaanbisnis
Model Sistem Informasi (Perspektif Arsitek)	Model data (entitas konvergen, Full Normalisasi)	Sistem DFD, arsitektur aplikasi	Arsitektur sistem bisnis	Arsitektur antarmuka manusia (peran, hak akses)	Diagram ketergantungan, history entitas (struktur proses)	Model aturan bisnis

Model Teknologi (Perspektif Builder)	Arsitektur data (dipetakan dalam hubungan antar sistem)	Rancangan sistem; strukturchart, psedo- code	Arsitektur sistem teknologi	Antarmuka pengguna (sesuatu yg dilihat pengguna), rancangan keamanan	Diagram alur kendali (struktur kendali)	Rancangan aturan bisnis
Definisi Sistem detail (*Perspektif Analyst System/ Programmer*)	Rancangan data (denormalisasi) , rancangan penyimpanan fisik	Rancangan program detail	Arsitektur sistem dan jaringan	Arsitektur tampilan dan keamanan (siapa dapat melihat apa)	Definisi waktu	Spesifikasi aturan dlm program logis
Pelaksanaan dan Sistem operasi (Perspektif Pengguna)	Data yang dikonversi, diinisialisasi dan diproduksi	Program aplikasi yang dikonversi dan diproduksi	Infrastruktur sistem dan jaringan	Personil pendukung, operator dan pengguna yangsudah dilatih	Kejadian/ Aktivitas bisnis	Prosedur dan aturan yang memaksa

Secara umum tiap kolom dalam kerangka kerja Zachman diuraikan lebih lanjutsebagai berikut:

1. **Data (*what*),** mendefinisikan material yang digunakan untuk membangun sistem (*inventory set*). Kolom ini menggambarkan hubungan antar entitas dengan menguraikan relasi antar data.
2. **Proses atau fungsi (*how*),** mendefinisikan aktivitas atau proses transformasi. Kolom ini memberikan uraian fungsional atas komponen sistem informasi fokus pada penyataan fungsi atau input dan output dengan mendeskripsikan keseluruhan proses yang terjadi dalam organisasi, proses kegiatan pemenuhan kebutuhan stakeholder dan proses input output yang terjadi pada organisasi.
3. **Jaringan (*where*),** mendefinisikan lokasi, topografi dan teknologi penghubung jaringan. Kolom ini mengilustrasikan node (perangkat penghubung) beserta

konektivitasnya yang menjelaskan lokasi operasional dari organisasi, struktur bangunan dan lokasi hingga peta instalasi jaringan yang dimiliki oleh organisasi tersebut.

4. **Sumber daya manusia (*who*),** mendefinisikan aturan dan organisasi. Kolom ini mendeskripsikan peran dan tanggung jawab, alokasi sumber daya manusia menurut struktur dan tanggung jawab dalam organisasi.

5. **Waktu (*when*),** mendefinisikan kejadian, siklus, jadwal (*time period*). Kolom ini fokus pada siklus waktu yang mendeskripsikan waktu terjadinya suatu proses dalam organisasi yang memiliki relasi untuk membangun kriteria kinerja dan tingkat kualitatif sumber daya organisasi
6. **Motivasi (*why*),** mendefinisikan tujuan, motivasi dan inisiatif. Kolom ini memaparkan visi, misi, dan tujuan organisasi yang mendeskripsikan motivasi dan tujuan akhir organisasi berserta strategi dan metode pencapaian yang digunakan oleh organisasi.

Setiap baris pada kerangka kerja Zachman mewakili perspektif yang berbeda dan unik.

1. **Perencana *(Scope Context),*** yaitu menetapkan gambaran umum sistem informasi, latar belakang dan tujuan *enterprise* berupa daftar lingkup penjelasan unsur bisnis.
2. **Pemilik *(Business Concept),*** yaitu menetapkan model semantik keterhubungan bisnis antara komponen-komponen bisnis yang didefinisikan oleh pimpinan eksekutif sebagai pemilik
3. **Perancang *(System Logic),*** yaitu menetapkan model-model sistem informasi sekaligus menjembatani hal-hal yang diinginkan pemilik dan hal-hal yang dapat direalisasikan secara teknis dan fisik.
4. **Pengembang *(Technology Physics),*** yaitu menetapkan rancangan teknis dan fisik yang digunakan dalam mengawasi implementasi teknis dan fisik.
5. **Implementator *(Component Assemblies),*** menetapkan peran dan rujukan bagi pihak yang bertanggung jawab untuk melakukan pembangunan secara teknis dan fisik serta mengadakan komponen-komponen yang diperlukan.
6. **Pengguna *(Operation Classes),*** yaitu merepresentasikan antarmuka dan fungsionalitas dari produk akhir yang merupakan produk dari semua aktivitas perencanaan, perancangan dan pengembangan yang berjalan sebelumnya.

Karakteristik kerangka kerja Zachman:

1. Kerangka kerja Zachman dapat memandu teknik pengumpulan data.
2. Kerangka kerja Zachman memerlukan dan melibatkan pemangku kepentingandalam desain dan pengembangan sistem.
3. Kerangka mengharuskan desainer untuk mempertimbangkan semua aspekdesain.
4. Kerangka ini bersifat generik dan dapat diterapkan ke domain apa pun.

Kelebihan dari Zachman Framework adalah sebagai berikut:

1. Zachman Framework merupakan standar secara de-facto untuk mengklasifikasikan artefak arsitektur Enterprise.
2. Struktur logikal untuk analisis dan presentasi artefak dari suatu perspektif manajemen.
3. Zachman Framework menggambarkan secara parallel baik dari sisi enjinering yang sudah sangat dimengerti maupun paradigma konstruksi.
4. Zachman Framework dikenal secara luas sebagai tool manajemen untuk memeriksa kelengkapan arsitektur dan maturity level.

Sedangkan kekurangan dari Zachman Framework antara lain:

1. Tidak ada proses untuk tahap implementasi.
2. Sulit untuk diimplementasikan secara keseluruhan.
3. Tidak ada contoh maupun ceklis yang siap secara utuh.
4. Perluasan coverage sel-sel tidak jelas.

E. The Open Group Architecture Framework (TOGAF)

The Open Group Architecture Technique (TOGAF) adalah sebuah framework yang dikembangkan oleh The Open Group's Architecture Framework pada tahun 1995. Awalnya TOGAF digunakan oleh Departemen Pertahanan Amerika Serikat namun pada perkembangannya TOGAF banyak digunakan pada berbagai bidang seperti perbankan, industri manufaktur dan juga pendidikan. TOGAF ini digunakan untuk mengembangkan enterprise architecture, dimana terdapat metode dan tools yang detil untuk mengimplementasikannya, hal inilah yang membedakan dengan framework EA lain misalnya framework Zachman. Salah satu kelebihan menggunakan framework TOGAF ini adalah karena sifatnya yang fleksibel dan bersifat open source.

Kerangka Arsitektur Open Group Architectural (TOGAF): Kerangka Arsitektur Open Group (TOGAF) pertama kali dikembangkan pada tahun 1995 dan didasarkan pada Kerangka Arsitektur Teknis Departemen Pertahanan untuk

Manajemen Informasi. TOGAF berfokus pada aplikasi bisnis missioncritis yang menggunakan blok bangunan sistem terbuka. “Elemen kunci TOGAF adalah Architecture Development Method (ADM) yang menentukan proses pengembangan arsitektur enterprise”. TOGAF menjelaskan peraturan untuk mengembangkan prinsip-prinsip yang baik, daripada menyediakan seperangkat prinsip arsitektur. Tiga tingkat prinsip mendukung pengambilan keputusan di seluruh perusahaan; memberikan panduan sumber daya TI; dan mendukung prinsip-prinsip arsitektur untuk pengembangan dan implementasi.

TOGAF berasal sebagai kerangka kerja umum dan metodologi umtuk 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.

Sebagai kerangka kerja perancangan arsitektur, TOGAF memiliki beberapa karakteristik, antara lain:

1. Termasuk dalam 3 kerangka kerja perancangan arsitektur yang paling sering digunakan (Schekkerman, 2003).
2. Merupakan kerangka kerja yang bersifat open-standard.
3. Bersifat netral –> *fits all*
4. Diterima oleh masyarakat internasional secara luas –> *fits all*
5. Pendekatannya bersifat menyeluruh (holistic).
6. Dibutuhkan metode yang *fleksibel*untuk mengintegrasikan unit-unit informasi dan juga sistem informasi dengan platform dan standar yang berbeda-beda.
7. TOGAF mampu untuk melakukan integrasi untuk berbagai sistem yang berbeda-beda
8. TOGAF adalah kerangka kerja umum dan dimaksudkan untuk digunakan dalam berbagai macam lingkungan, ia menyediakan konten kerangka kerja yang *fleksibel*dan extensible yang mendasari seperangkat pengiriman arsitektur generik.
9. TOGAF cenderung bersifat *generik*dan *fleksibel* karena dapat mengantisipasi segala macam artefak yang mungkin muncul dalam proses perancangan (Resource base TOGAF menyediakan banyak material

referensi), standarnya diterima secara luas, dan mampu mengatasi perubahan.

10. Fokus pada siklus implementasi (ADM) dan proses –> *process driven*
11. Kunci TOGAF adalah metode – TOGAF Architecture Development Method (ADM – Metode Pengembangan Arsitektur) – untuk mengembangkan suatu arsitektur enterprise yang membahas kebutuhan bisnis.
12. TOGAF relatif mudah diimplementasikan –> *fits all*
13. TOGAF bersifat *open source*, sehingga bersifat netral terhadap teknologi dari *vendor*tertentu –> *fits all*

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 pengembanga 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 begaimana arsitektur dikembangkan di seluruh kontinum mulai dari arsitektur dasar, melalui sistem umum arsitektur dan arsitektur industri spesifik untuk individu suatu enterprise arsitektur sendiri.

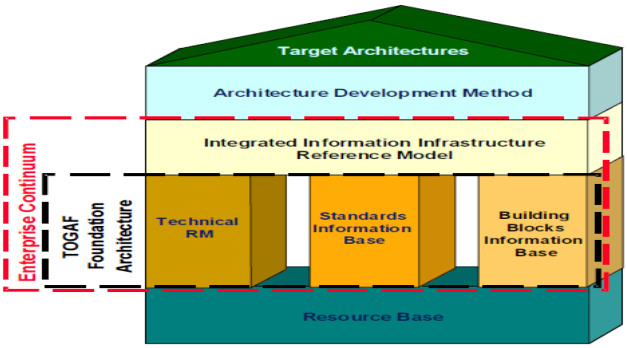


Fig. 2.6. TOGAF (based on The Open Group 2002).

F.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.

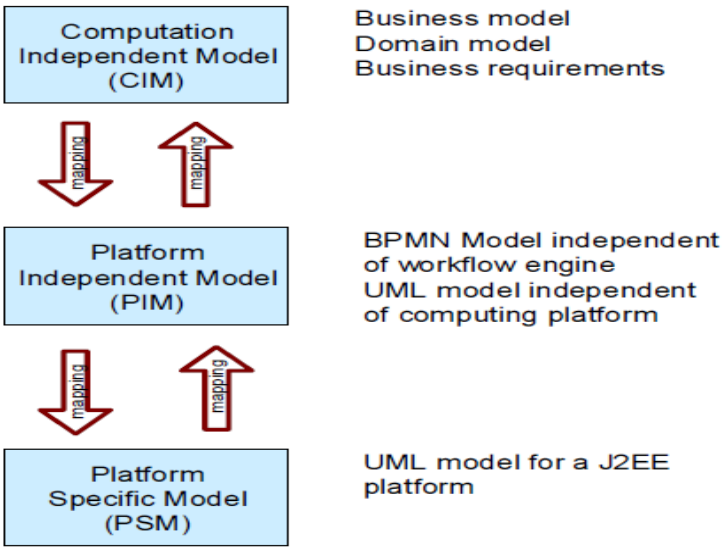


Fig. 2.7. MDA framework.

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.

Pertemuan 4

Bahasa Arsitektur

A. Pendahuluan

- 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.

B. 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 menunjukan 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.

C. BPMN

- BPMN (Business process Modelling Notation) adalah salah satu standar yang sedang dikembangkan oleh Business Process Management Inititative (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.
- Tujuan utama dari BPMN adalah untuk memberikan notasi seragam untuk proses pemodalan bisnis dalam hal kegiatan dan hubungan mereka.
- 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).

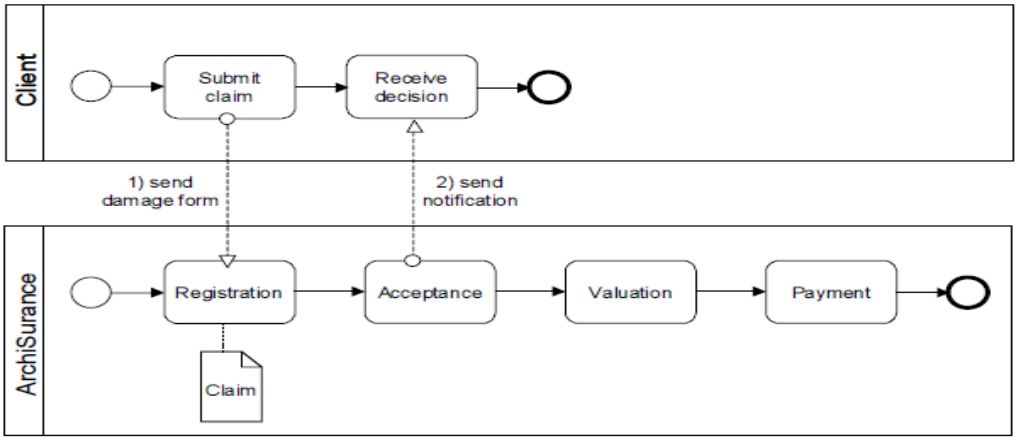


Fig. 2.9. Example model in BPMN.

D. 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 :

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

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

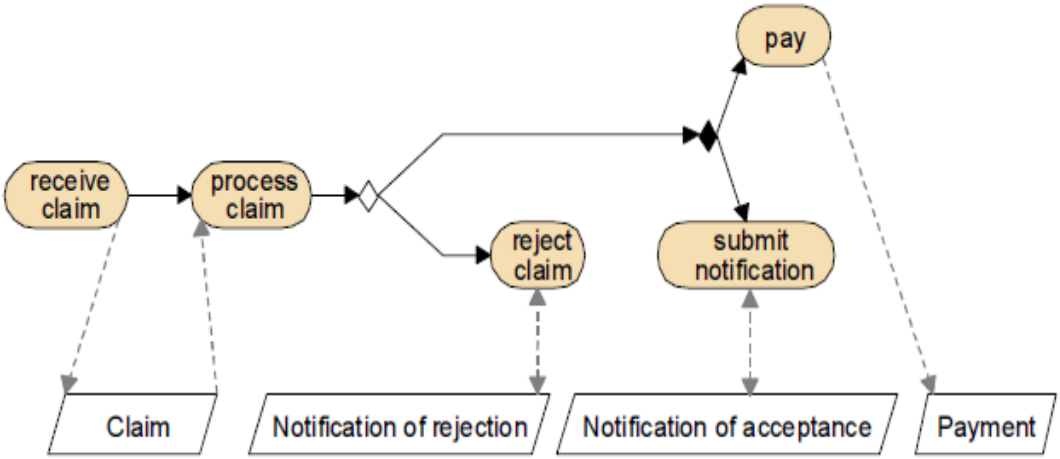


Fig. 2.10. Example of a business process model in Testbed.

E. 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.

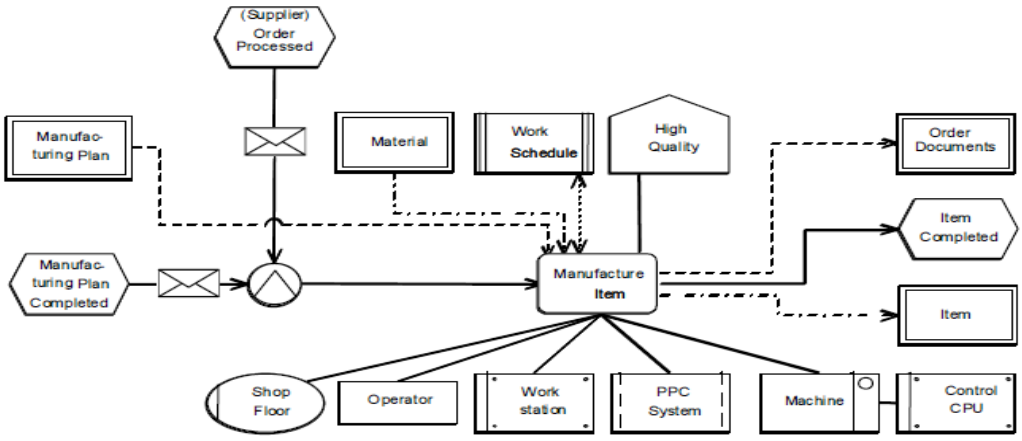


Fig. 2.11. Events, functions and control flows in ARIS.

F. 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.

G. 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 :

a. Komponen
b. Konektor
c. Sistem (Konfigurasi komponen dan konektor)
d. Port (titik interaksi dengan komponen)
e. Role (titik interaksi dengan konektor)

f. Representasi (digunakan untuk model komposisi hirarkis)
g. Rep-map (yang memetakan komponen atau konektor komposit arsitektur internal yang untuk elemen antarmuka eksternal)

H. 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.

I. 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 hnaya 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.

J. 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.
- Idenya 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.

K. 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, fleksibiltas untuk mengubah pemasok layanan, fleksibilitas untuk menggunakan kembali layanan yang ada untuk penyediaan produk atau jasa baru. Ini akan menciptakan peluang baru untuk outsourching, 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 mengeksploitasi 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 ekspoitasi.

Pertemuan 5

Kompleksitas Arsitektur Dan Penggambaran Arsitektur Enterprise

A. Pendahuluan

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

- ❖ Mengidentifikasikan 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.

B. 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 :

- ✓ Design
- ✓ Komunikasi
- ✓ Realisasi
- ✓ Change

1. Design

Saat mendesain arsitektur, arsitek harus menggunakan common,kosakata yang didefinisikan dengan baik untuk menghindari kesalahpahaman dan mempromosikan yang jelas. Kosa kata 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.

2. 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.

3. 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.

4. 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.

C. 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.

D. Integrasi Domain Arsitektur

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

Ada banyak contoh masalah integrasi, di mana akan dibahas dua contoh di bawah. Secara umum, beberapa masalah integrasi dapat dengan mudah dipecahkan: untuk contoh, dengan menggunakan standar yang ada; yang lain bersifat intrinsik dengan pendekatan arsitektur dan tidak dapat 'diselesaikan' dalam arti biasa. Kasus-kasus sulit intrinsik dengan kompleksitas arsitektur, dan menghilangkan masalah

juga akan menghapus gagasan arsitektur itu sendiri. Kita tidak bisa menyingkirkannya masalah integrasi; kita hanya bisa mengembangkan konsep dan alat untuk membuat lebih mudah untuk menangani masalah ini. Ini diilustrasikan oleh Contoh 1 di bawah ini. Contoh 1. Sebagai contoh pertama dari masalah integrasi, pertimbangkan Gambar 3.1, yang berisi beberapa arsitektur. Lima arsitektur dapat menjadi model yang diungkapkan dalam UML, atau model dari sel kerangka kerja arsitektur Zachman, atau jenis apa pun. Misalnya, mungkin ada perusahaan yang memodelkan aplikasinya di UML dan proses bisnisnya di BPMN. Dalam semua kasus ini, tidak jelas bagaimana konsep dalam satu pandangan terkait dengan konsep dalam pandangan lain. Apalagi itu tidak jelas apakah tampilan kompatibel satu sama lain. Integrasi arsitektur pada Gambar 3.1 bermasalah karena ini lima arsitektur dikembangkan oleh pemangku kepentingan yang berbeda dengan keprihatinan mereka sendiri. Mengaitkan arsitektur berarti mengaitkan gagasan para pemangku kepentingan ini, sebagian besar tetap implisit. Konsekuensinya adalah kita sering tidak dapat berasumsi memiliki pemetaan satu-

ke-satu yang lengkap, dan yang terbaik yang kami bisa minta adalah bahwa pandangan dalam beberapa hal konsisten satu sama lain. Ini sering disebut masalah penyelarasan, dan contoh UML-BPMN disebut masalah penyelarasan bisnis – TI.

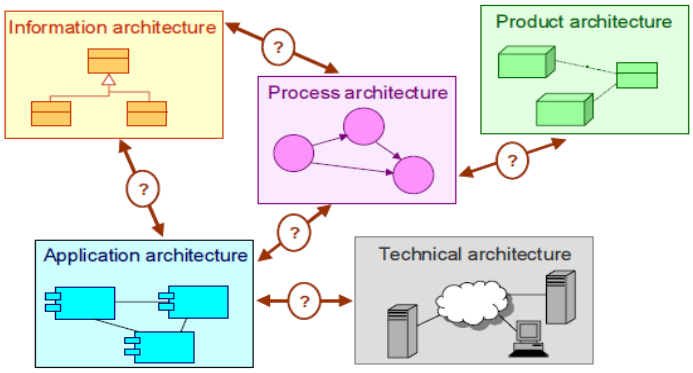


Fig. 3.1. Heterogeneous architectural domains.

E. 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 kontek disini, model biasanya dipresentasikan dengan menggunakan bahasa grafis atau tekstual yang diformalkan. Karena struktur formal mereka, model sendiri untuk berbagai macam pemrosesan otomatis, visualisasi, analisis, tes, dan simulasi. Lebih jauh lagi, kekakuan dari pendekatan berbasis model juga memaksa arsitek untuk bekerja dengan cara yang lebih teliti dan membantu menghilangkan reputasi arsitektur yang tidak menguntungkan hanya sebagai beberapa 'gambar cantik'. Namun, para pemangku kepentingan yang berbeda memiliki pandangan yang berbeda tentang dunia. Kebutuhan setiap orang tidak dapat dengan mudah diakomodasi oleh satu model.

Terdapat bahasa terkait selanjutnya ialah mengenai :

- ✓ Observing the universe
- ✓ Concern
- ✓ Memahami Domain

- ✓ Tampilan dan Sudut Pandang
- ✓ Cara kerja
- ✓ Model Arsitektur Enterprise

1. 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 enterprise nya, 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 di bawah yang berdasarkan pada FRISCO tetrahedron.

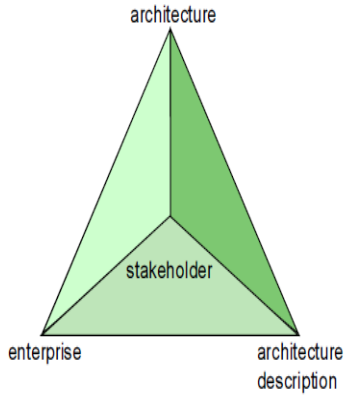


Fig. 3.2. Relationship between enterprise, stakeholder, architecture, and architecture description.

2. Concern

Dalam menyusun bagian enterprise, stakeholder biasanya akan terpengaruhi oleh beberapa fokus mereka masing-masing atau kepentingan masing-masing. Perhatikan bahwa para pemangku kepentingan, serta kepentingan mereka, dapat dianggap pada diagregasi serta di tingkat individu. Misalnya, satu bisnis manajer yang memahami sistem informasi adalah pemangku kepentingan. Kolektif manajemen bisnis, bagaimanapun juga dapat dilihat sebagai pemangku kepentingan sistem Informasi.

Tetapi kepentingan juga bukan merupakan satu-satunya faktor yang mempengaruhi konsepsi stakeholder, lingkungan, kultur, pendidikan serta background masing-masing turut mengambil peran.

Lebih khusus lagi, di konteks pengembangan sistem, arsitek akan mendekati domain dengan tujuan mengekspresikan domain dalam beberapa konsep, seperti kelas, kegiatan, kendala, dll. Konsep yang digunakan arsitek untuk digunakan saat memodelkan beberapa domain, akan sangat mempengaruhi konsepsi arsitek itu. Seperti Abraham Maslow mengatakan: ‘Jika satu-satunya alat yang Anda miliki adalah palu, Anda cenderung melihat setiap masalah seperti paku. "

Oleh karena itu ketika arsitek membuat pemodelan, mereka melakukannya dengan perspective tertentu.

Secara umum, orang cenderung memikirkan alam semesta ('dunia di sekitar kita') yang terdiri dari elemen terkait. Namun, untuk menganggap bahwa alam semesta terdiri dari seperangkat elemen pilihan subyektif, dibuat (secara sadar atau tidak). Pilihan yang dibuat adalah 'elemen' (atau 'benda') dan 'Hubungan' adalah konsep paling dasar untuk memodelkan alam semesta.

3. 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.

Untuk alasan praktis, kita akan memahami tindakan pemodelan juga untuk dimasukkan kegiatan yang terlibat dalam representasi model dengan cara beberapa bahasa dan media. Kami menganggap arsitek tidak hanya mampu mewakili (bagian dari) konsepsi perusahaan mereka, tetapi juga untuk dapat mewakili (bagian dari) perspektif yang mereka gunakan dalam menghasilkan konsepsi ini.

Ini mengharuskan arsitek untuk dapat merefleksikan pekerjaan mereka sendiri. Saat memodelkan suatu domain dalam hal, katakanlah, diagram kelas UML, kami berasumsi bahwa mereka dapat mengungkapkan fakta yang mereka gunakan kelas, agregasi, asosiasi, dll., untuk menjelaskan domain tersebut dimodelkan.

4. 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.

Oleh karena itu, para pemangku kepentingan memerlukan pandangan spesifik tentang arsitektur yang fokus tentang kepentingan mereka dan tinggalkan informasi yang tidak perlu. Sejak kami menempatkan model sentral dalam deskripsi kami tentang arsitektur, ini menyiratkan yang kami miliki untuk memberikan pandangan berbeda dari model-model ini untuk mengakomodasi para pemangku kepentingan. Tampilan ditentukan melalui sudut pandang, yang menentukan caranya

pandangan yang membahas kepentingan khusus dari para pemangku kepentingan dibangun, mengingat arsitektur yang dipertimbangkan. Apa yang seharusnya dan tidak seharusnya terlihat dari sudut pandang tertentu dengan demikian sepenuhnya tergantung pada pemangku kepentingan.

Standar IEEE 1471 (IEEE Computer Society 2000) mendefinisikan pandangan dan sudut pandang sebagai berikut:

Pandangan : representasi sistem dari perspektif terkait mengatur kepentingan.

Sudut Pandang: spesifikasi konvensi untuk konstruksi dan menggunakan tampilan; sebuah pola atau tempat untuk mengembangkan individu tampilan dengan menetapkan tujuan dan audiens untuk tampilan dan

teknik untuk pembuatan dan analisisnya.

5. Cara Kerja

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

6. Model Arsitektur Enterprise

Dalam situasi yang ideal, kita akan memiliki model tunggal untuk arsitektur enterprise, untuk memastikan koherensi dan konsistensi antara semua bagian yang berbeda. Pada kenyataannya, model seperti itu mungkin tidak akan pernah ada, terutama ketika kita berbicara tentang beberapa domain arsitektur. Dalam prakteknya suatu arsitektur (dan terutama arsitektur perusahaan) akan muncul pada mode dari bawah ke atas. Model parsial dari domain yang berbeda akan dibangun sesuai dengan kebutuhan di domain tersebut. Di mana ini menyentuh satu sama lain, inkonsistensi dapat muncul, yang perlu diselesaikan pada akhirnya karena sistem dunia nyata yang dirancang tentunya harus konsisten. Dengan cara ini, kita perlahan-lahan bergerak menuju model yang mendasari Platonis ini, dan model parsial dari mana ia dibangun dapat dilihat sebagai pandangan arsitektur total.

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.

Pertemuan 6

Gambar, Model Dan Semantik

A. Pendahuluan

Dalam banyak disiplin ilmu teknik, pemodelan suatu sistem terdiri dari konstruksi model matematika. Di bidang perusahaan dan arsitektur perangkat lunak, ada kecenderungan yang luar biasa untuk melihat gambar dan diagram sebagai bentuk model daripada sebagai bentuk bahasa, atau, lebih tepatnya, sebagai bentuk struktur yang membantu dalam memvisualisasikan dan mengkomunikasikan deskripsi sistem. Dengan kata lain, dalam arsitektur ada kecenderungan untuk mengganti pemodelan matematika dengan visualisasi hoc.

Oleh karena itu bedakan antara konten dan visualisasi model atau pandangan, di mana yang pertama mengacu pada konsep yang terlibat, dan yang kedua merujuk ke bentuk di mana ini disajikan. Misalnya, dalam satu visualisasi arsitektur proses mungkin divisualisasikan sebagai sebuah lingkaran, dan yang lainnya berbentuk bujur sangkar. Apalagi isinya dapat menyatakan bahwa satu konsep lebih penting daripada yang lain, yang divisualisasikan dengan menggambar konsep pertama di atas yang kedua. Itu hubungan yang sama pentingnya juga dapat divisualisasikan dengan intensitas warna yang digunakan

untuk memvisualisasikan konsep. Arsitek termotivasi untuk membuat eksplisit apakah informasi visual seperti 'di atas' atau 'merah' memiliki makna dalam model, atau bersifat insidental. Ketika sesuatu itu kebetulan arsitek termotivasi untuk menghapusnya dari gambar, karena hanya mengalihkan perhatian dari pesan gambar. Ketika itu bermakna, artinya harus dibuat eksplisit. Ketika sudut pandang baru ditentukan, konten dan visualisasi dapat didefinisikan dalam dua fase terpisah. 'Konten' dan 'visualisasi' harus ditafsirkan secara longgar di sini cara. Misalnya, visualisasi juga dapat mencakup perangkat input seperti menu atau tombol, dan konten juga dapat mencakup tindakan yang mengubah contohnya dengan menambahkan atau menghapus konsep. Tindakan dalam model adalah digunakan di sini untuk menangani interaksi dengan pengguna.

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 merajuk pada ‘efek’ dari sesuatu dimodel, dimana mengacu pada dinamika dalam model tersebut.

Dalam linguistik ada perbedaan yang jauh lebih tua antara sintaksis, semantik, dan pragmatik. Contoh lain adalah dalam arti informasi di Internet Web: Halaman web secara tradisional telah digunakan untuk menggambarkan semua jenis masalah, tetapi mereka sering merujuk ke objek yang sama menggunakan terminologi yang berbeda. Ini membawa Tim Berners-Lee ke penemuan Web semantik, tempat ontologi memainkan peran penting.

B. Model Simbolik Dan Semantik

Untuk membuat gagasan semantik eksplisit, dibedakan antara simbolik model dan model semantik.

❖ 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.

❖ 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, danmodel semantik.

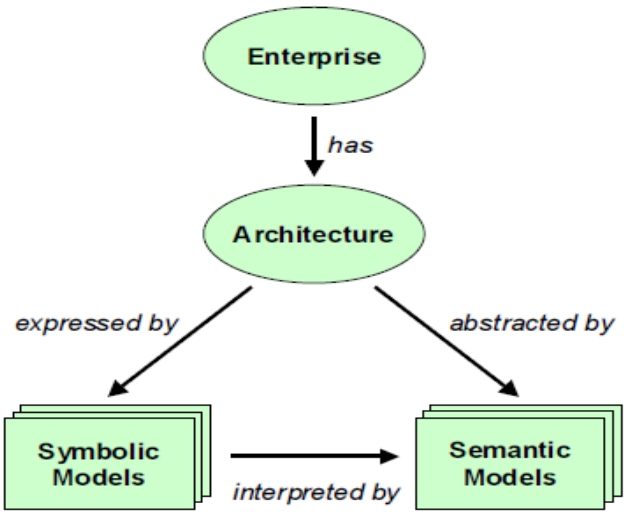


Fig. 3.3. The enterprise, its architecture, symbolic and semantic models.

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. Perhatikan bahwa ini adalah bukan perbedaan yang sama dengan model formal dan informal: dalam kelas model informal, dinyatakan misalnya dalam bahasa alami, kedua jenis ada, serta dalam kelas model formal, diekspresikan misalnya dalam logika tingkat pertama.

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.

Ada dua jenis abstraksi yang kita gunakan dalam menciptakan model realitas. Yang pertama adalah abstrak dari (properti) entitas yang tepat dalam kenyataan ke yang merujuk konsep. Ini terjadi misalnya ketika kita membuat model struktur statis suatu aplikasi dalam hal komponennya. Jenis kedua adalah abstraksi dari perbedaan antara entitas dalam kenyataan dengan mengelompokkannya menjadi satu konsep. Ini kadang-kadang disebut sebagai generalisasi, dan terjadi misalnya ketika kita menggunakan konsep 'karyawan', yang mengelompokkan individu di sebuah perusahaan. Ini terkait dengan gagasan 'macam' yang dibahas di bawah ini.

1. 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 membatasil lebih lanjut atau menganalisis sifat model simbolik.

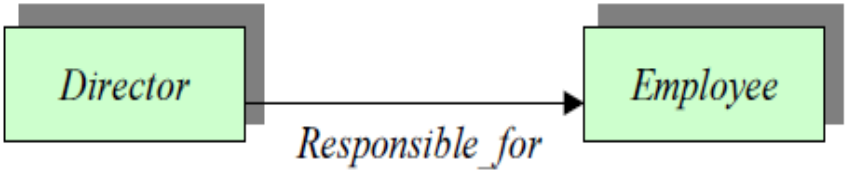


Fig. 3.4. Symbolic model of the director–employee relationship.

Kita perlu mengingat kembali bahwa di atas adalah struktur sintaksis; itu adalah deskripsi dari model simbolik dengan tanda tangan yang jenisnya adalah Karyawan dan Direktur, dan dengan masing-masing entitas terkait dengan relasi bernama Responsible_untuk. Sampai saat ini kami belum memberikan makna apa pun, kita hanya mengkategorikan entitas model simbolik menjadi dua kategori dan bernama relasi antara entitas milik dua macam. Sintaksisnya nama yang digunakan untuk jenis dan hubungan mendorong intuisi beberapa langkah ke depan: kita tahu apa itu karyawan, apa itu direktur, dan bagaimana cara bertanggung jawab. Namun, sementara nama sintaksis ini membantu kita dalam pemahaman, mereka juga merupakan sumber utama kebingungan dalam komunikasi dan analisis suatu arsitektur. Kita bisa menamai macam di atas X dan Y lebih baik mempertahankan kualitas sintaksis yang tidak berarti, dan menghindari kebingungan dengan semantik. Tanda tangan dengan demikian memberikan glosarium konseptual yang istilahnya semuanya lain dalam model simbolis harus dijelaskan, mirip dengan bahasa Inggris kamus untuk bahasa Inggris. Selain itu, tanda tangan terdiri informasi untuk menangkap aspek-aspek tertentu dari ontologi arsitektur. Misalnya, mungkin menyertakan informasi hierarkis di antara macam-macam syarat hubungan ‘is-a’, atau informasi penahanan dalam hal suatu ‘Termasuk’ hubungan, atau informasi ketergantungan dalam hal ‘mengharuskan’ hubungan. Tanda tangan yang berisi informasi tambahan ini lebih banyak umum daripada glosarium. Mereka menyediakan skema konseptual, mirip dengan skema disediakan untuk ahli biologi dengan klasifikasi spesies.

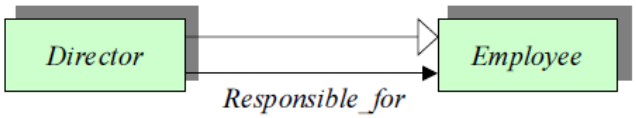


Fig. 3.5. Extended symbolic model.

Sebagai contoh, Gambar 3.5 memperluas tanda tangan sebelumnya dengan hubungan relationship is-a ’ antara jenis Direktur dan Karyawan (dilambangkan dengan warisan UML hubungan), secara intuitif menyarankan bahwa setiap direktur juga seorang karyawan. Selain itu, model simbolik juga dapat berisi serangkaian tindakan, dan tanda tangan seperangkat simbol tindakan, artinya kita bahas di bawah ini.

2. 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.

Jelas bahwa, secara umum, ada sejumlah besar interpretasi yang berbeda untuk model simbolis yang sama. Ini mencerminkan intuisi itu mungkin ada banyak arsitektur yang sesuai dengan deskripsi arsitektur tertentu. Bahkan, tanda tangan model simbolik dari arsitektur hanya menentukan beberapa blok bangunan dasar yang menggambarkan arsitekturnya. Dengan kata lain, kita melihat semantik formal model simbolik sebagai koleksi konkret objek matematika menafsirkan suatu sistem sesuai ke deskripsi arsitektur tertentu. Karena itu, ini melibatkan komponen dan hubungan konkret mereka yang dapat berubah dalam waktu karena perilaku dinamis dari suatu sistem. Situasi konkrit dari suatu sistem dijelaskan dengan cara variabel yang diketik sesuai dengan jenis individu yang mereka

maksud. Lebih konkret, untuk model simbolik, kita akan menyatakan dengan x: T suatu variabel x yang berkisar pada individu-individu dari jenis T. Misalnya, kita bisa menggunakan kalimat logis untuk membatasi penafsiran Direktur menjadi set yang tidak kosong. Perhatikan bahwa karena Direktur adalah Karyawan, juga interpretasi yang terakhir sort akan non-kosong. Tindakan yang terjadi dalam model simbol ditafsirkan sebagai perubahan model berdasarkan interaksi dengan pengguna. Untuk mendefinisikan tindakan, kita harus mendefinisikan variabel input tindakan, dan bagaimana kita dapat mengambil variabel dari pengguna. Formal semantik cukup kaya untuk menangkap dinamika suatu sistem dengan menafsirkannya informasi simbolis (dan sering bergambar) tersedia untuk menggambarkan proses bisnis dan perangkat lunak dalam bahasa ArchiMate

C. 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.

D. Ringkasan

Pendekatan arsitektur terintegrasi sangat diperlukan untuk mengendalikan organisasi dan sistem informasi yang kompleks. Secara luas diakui bahwa

perusahaan perlu 'melakukan arsitektur'; warisan masa lalu telah menunjukkan kepada kita bahwa pengembangan bisnis dan TIK tanpa arsitektur Visi mengarah pada sistem yang tidak terkendali yang hanya bisa disesuaikan dengan kesulitan yang hebat. Namun, arsitektur jarang didefinisikan pada tingkat tunggal. Dalam suatu perusahaan, banyak masalah yang berbeda tetapi terkait perlu ditangani. Proses bisnis harus berkontribusi pada produk organisasi dan layanan, aplikasi harus mendukung proses, sistem dan jaringan harus dirancang untuk menangani aplikasi, dan semua ini harus sejalan dengan tujuan keseluruhan organisasi. Banyak dari domain memiliki praktik arsitektur mereka sendiri, dan karenanya aspek yang berbeda perusahaan akan dijelaskan dalam berbagai arsitektur. Arsitektur ini tidak dapat dilihat secara terpisah. Misalnya, domain arsitektur terkait, dan struktural dan perilaku sudut pandang terkait. Integrasi harus berurusan dengan fakta bahwa berbagai sudut pandang didefinisikan oleh para pemangku kepentingan dengan sudut pandang mereka sendiri. Inti dari pendekatan kami terhadap arsitektur enterprise adalah domain harus dilihat dengan cara yang koheren dan terintegrasi. Kami menyediakan dukungan untuk arsitek dan pemangku kepentingan lainnya dalam desain dan penggunaannya arsitektur terintegrasi. Untuk tujuan ini, kami harus memberikan konsep yang memadai untuk menentukan arsitektur di satu sisi, dan di sisi lain mendukung arsitek dengan teknik visualisasi dan analisis yang menciptakan wawasan ke dalam struktur dan hubungan mereka. Dalam pendekatan ini, hubungan dengan yang ada standar dan alat harus ditekankan; kami bertujuan untuk mengintegrasikan apa yang sudah tersedia dan bermanfaat. Pendekatan yang kami ikuti sangat umum dan secara sistematis mencakup konsep arsitektur yang diperlukan dan teknik pendukung untuk visualisasi, analisis, dan penggunaan arsitektur. Kami mengadopsi kerangka kerja di sekitar pemangku kepentingan, perusahaan, arsitektur, dan deskripsi arsitektur sebagai penonton dengan alam semesta, konsepsi, dan perwakilan. Pandangan dan sudut pandang pemangku kepentingan adalah hasil dari pemodelan, tindakan sengaja mengabstraksi model dari kenyataan, yaitu, dari domain yang dipahami sebagai bagian dari alam semesta. Pandangan-pandangan ini terdiri dari satu set model arsitektur perusahaan. Dalam kerangka kerja ini, perbedaan dibuat antara konten pandangan

dan visualisasi, dan perbedaan juga dibuat antara simbolis model, yang mengacu pada arsitektur perusahaan, dan semantik model sebagai abstraksi dari arsitektur dan yang mengartikan model simbolik. Inti dari setiap model simbolik adalah ciri khasnya, yaitu mengelompokkan entitas model simbolik.

PERTEMUAN 9
PENGANTAR PADA FAMEWORK TOGAF

1. PENGANTAR PADA FRAMEWORK TOGAF

Sejarah TOGAF

TOGAF (The Open Group Architecture Framework) muncul dengan cepat dan merupakan kerangka kerja serta metode yang dapat diterima secara luas dalam pengembangan arsitektur perusahaan.

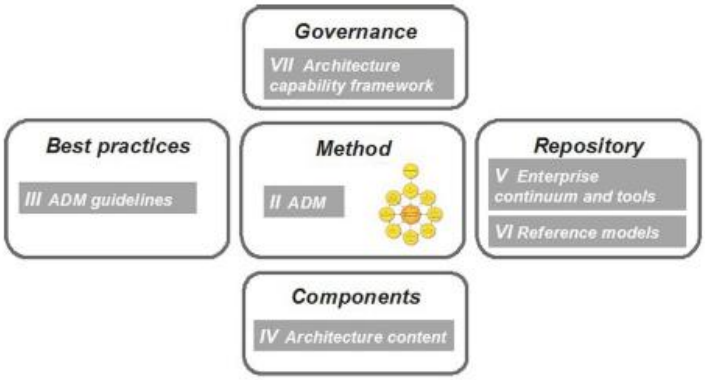
TOGAF dimulai awal 1990-an sebagai metodologi untuk pengembangan arsitektur teknis, dan telah dikembangkan oleh The Open Group ke dalam kerangka arsitektur enterprise yang luas. Pada tahun 1995 , versi pertama dari TOGAF (TOGAF 1.0) disajikan. Versi ini terutama didasarkan pada Architecture Framework Teknis Pengelolaan Informasi (TAFIM), dikembangkan sejak tahun 1980 oleh an Departemen Pertahanan AS.

Pada bulan Desember 2001 TOGAF 7, "Edisi Teknis ", diterbitkan TOGAF 8 ("Enterprise Edition") pertama kali diterbitkan pada bulan Desember 2002 dan diterbitkan dalam bentuk diperbarui TOGAF 8.1 pada bulan Desember 2003. Sekitar tahun 2005 menjadi TOGAFTM merek dagang terdaftar dari The Open Group. Pada bulan November 2006 Open Group dirilis TOGAF 8.1.1. Menurut The Open Group , pada Februari 2011, lebih dari 15.000 individu TOGAF Bersertifikat. Pada September 2012 register resmi memiliki lebih dari 20.000 individu bersertifikat.

The TOGAF Document?

Dokumen Togaf dibagi menjadi 7 bagian yaitu:

1. Introduction
2. ADM(Architecture Development Method)
3. ADM Guidelines
4. Architecture Content
5. Enterprise Continuum and Tools
6. Reference Models7.
7. Architecture Capability Framework



Gambar di atas menyajikan pengelompokan berbagai macam bagian pada dokumen TOGAF yaitu : Method, Best practice, Component, Repository dan Governance ADM (“Architecture Development Method”) pada bagian II adalah point utama dari TOGAF reference document, dengan menggunakan crop circle

diagram (or TOGAF wheel), yang mendeskripsikan perbedaan fase-fase dan metodenya.

Part III membahas mengenai guidelines and best practices linked ke dalam ADM, yang mencakup keamanan and gap analysis to stakeholder management. It should be noted that in general TOGAF does not provide “standard solutions” but rather a series of practices “that work,” accompanied by more or less detailed examples.

Part IV (architecture content) is dedicated to the tangible elements used in development work: deliverables, catalogs, matrices, diagrams, or the “building blocks” that constitute the architecture.

Parts V and VI berfokus pada repository enterprise architecture, dan partitioning, typology, dan tools.

Part VII (“Architecture Capability Framework”) deals with architecture governance, including repository management.

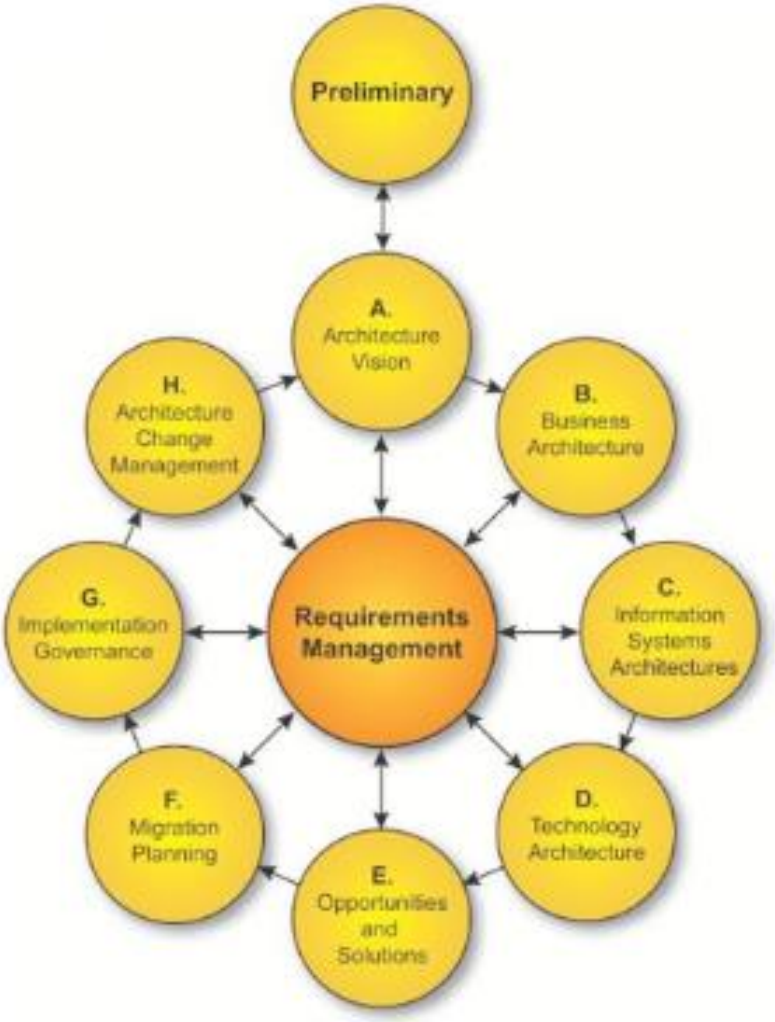
TOGAF : KEY POINT

The ADM crop circle diagram menyajikan struktur dari method with its phases and transitions and is the first striking image encountered when broaching TOGAF. Pada fase ini mendefiniskan high-level work stages, which consume and provide products (deliverables). Masing-masing dari delapan fase berkontribusi untuk mencapai tujuan strategis yang ditentukan, dari visi keseluruhan arsitektur (fase A) hingga pemeliharaan arsitektur yang digunakan (fase H). Urutan ini, yang disebut siklus ADM, terjadi dalam konteks proyek arsitektur yang dikelola oleh manajemen eksekutif perusahaan.

Pekerjaan yang dilakukan diawasi oleh dewan arsitektur, dalam kemitraan dengan semua bisnis dan pemangku kepentingan SI. Seperti yang Anda lihat, path yang diajukan adalah sebuah siklus, yang selesai dengan mengulang kembali dengan sendirinya. Diakui, ini hanyalah representasi skematis yang hanya sebagian mewakili kenyataan. Namun, itu berhasil mengekspresikan sifat berkelanjutan dari pekerjaan arsitektur perusahaan, yang menanggapi tuntutan konstan dari bisnis.

Posisi sentral ditempati oleh manajemen persyaratan dalam diagram adalah bukti peran penting yang dimainkannya dalam siklus ADM. Sebenarnya, manajemen persyaratan lebih merupakan kegiatan permanen daripada fase. Namun, istilah "fase" digunakan untuk menunjuknya untuk menyelaraskan kosa kata. Hal yang sama berlaku untuk fase awal, yang mengelompokkan kegiatan lintas organisasi seperti definisi konteks, metode dan alat untuk arsitektur perusahaan, dan awal siklus ADM.

Pada dasarnya, tujuan siklus ADM adalah untuk berhasil menyelesaikan proyek transformasi, yang tujuannya adalah untuk memungkinkan perusahaan untuk menanggapi serangkaian tujuan bisnis.



The "TOGAF crop circle diagram" with the ADM phases

Sebagai komponen inti, TOGAF ADM menyediakan serangkaian proses iteratif mulai dari menyusun arsitektur, transisi, hingga mengelola proses realisasi arsitektur. TOGAF ADM terdiri atas sepuluh fase sebagai berikut:

1. Preliminary Phase – fase ini mencakup aktivitas persiapan untuk menyusun kapabilitas arsitektur termasuk kustomisasi TOGAF dan mendefinisikan prinsip-prinsip arsitektur. Tujuan fase ini adalah untuk menyakinkan setiap orang yang terlibat di dalamnya bahwa pendekatan ini untuk mensukseskan proses arsitektur. Pada fase ini harus menspesifikasikan who, what, why, when, dan where dari arsitektur itu sendiri.

- *f What* adalah ruang lingkup dari usaha.
- *Who* adalah siapa yang akan memodelkannya, siapa orang yang akan bertanggung jawab untuk mengerjakan arsitektur tersebut, dimana mereka akan dialokasikan dan bagaimana peranan mereka.
- *How* adalah bagaimana mengembangkan arsitekture interprise, menentukan framework dan metode apa yang akan digunakan untuk menangkap informasi.
- *When* adalah kapan tanggal penyelesaian arsitektur
- *Why* adalah mengapa arsitektur ini dibangun. Hal ini berhubungan dengan tujuan organisasi yaitu bagaimana arsitektur dapat memenuhi tujuan organisasi.

2. Phase A: Architecture Vision – fase ini merupakan fase inisiasi dari siklus pengembangan arsitektur yang mencakup pendefinisian ruang lingkup, identifikasi stakeholders, penyusunan visi arsitektur, dan pengajuan persetujuan untuk memulai pengembangan arsitektur.

Beberapa tujuan dari fase ini adalah :

- Menjamin evolusi dari siklus pengembangan arsitektur mendapat pengakuan dan dukungan dari manajemen enterprise.
- Mensyahkan prinsip bisnis, tujuan bisnis dan pergerakan strategis bisnis organisasi.
- Mendefinisikan ruang lingkup dan melakukan identifikasi dan memprioritaskan komponen dari arsitektur saat ini.
- Mendefiniskan kebutuhan bisnis yang akan dicapai dalam usaha arsitektur ini dan batasannya.
- Menghasilkan visi arsitektur yang menunjukan respon terhadap kebutuhan dan batasannya.

Beberapa langkah yang dilakukan pada fase ini adalah :

- Menentukan / menetapkan proyek

- Mengindentifikasi tujuan dan pergerakan bisnis. Jika hal ini sudah didefinisikan, pastikan definisi ini masih sesuai dan lakukan klarifikasi terhadap bagian yang belum jelas.
- Meninjau prinsip arsitektur termasuk prinsip bisnis. Meninjau ini berdasarkan arsitektur saat ini yang akan dikembangkan. Jika hal ini sudah didefinisikan, pastikan definisi ini masih sesuai dan lakukan klarifikasi terhadap bagian yang belum jelas.
- Mendefinisikan apa yang ada di dalam dan di luar rungan lingkup usaha saat ini.
- Mendefinisikan batasan-batasan seperti waktu, jadwal, sumber daya dan sebagainya.
- Mengindentifikasikan stakeholder, kebutuhan bisnis dan visi arsitektur.
- Mengembangkan Statement of Architecture Work.

3. Phase B: Business Architecture – fase ini mencakup pengembangan arsitektur bisnis untuk mendukung visi arsitektur yang telah disepakati. Pada tahap ini tools dan method umum untuk pemodelan seperti: Integration DEFinition (IDEF) dan Unified Modeling Language (UML) bisa digunakan untuk membangun model yang diperlukan.

Beberapa tujuan dari fase ini adalah :

- Menguraikan deskripsi arsitektur bisnis dasar.
- Mengembangkan arsitektur bisnis tujuan, menguraikan strategi produk dan/atau service dan aspek geografis, informasi, fungsional dan organisasi dari lingkungan bisnis yang berdasarkan pada prinsip bisnis, tujuan bisnis dan penggerak strategi.
- Menganalisi gap antara arsitektur saat ini dan tujuan.
- Memilih titik pandang yang relevan yang memungkinkan arsitek mendemokan bagaimana maksud stakeholder dapat dicapai dalam arsitektur bisnis.
- Memilih tools dan teknik relevan yang akan digunakan dalam sudut pandang yang dipilih.

Beberapa langkah yang dilakukan di fase ini adalah :

- Mengembangkan deskripsi asitektur bisnis saat ini untuk mendukung arsitektur bisnis target.
- Mengindentifikasi reference model, sudut pandang dan tools
- Melengkapi arsitektur bisnis
- Melakukan gap analisis dan membuat laporan

4. Phase C: Information Systems Architectures – Pada tahapan ini lebih menekankan pada aktivitas bagaimana arsitektur sistem informasi dikembangkan. Pendefinisian arsitektur sistem informasi dalam tahapan ini meliputi arsitektur data dan arsitektur aplikasi yang akan digunakan oleh organisasi. Arsitektur data lebih memfokuskan pada bagaimana data digunakan untuk kebutuhan fungsi bisnis, proses dan layanan. Teknik yang bisa digunakan dengan yaitu: ER-Diagram, Class Diagram, dan Object Diagram.

Tujuan dari fase ini adalah mengembangkan arsitektur tujuan dalam domain data dan aplikasi. Ruang lingkup dari proses bisnis yang didukung dalam fase C dibatasi pada proses-proses yang didukung oleh TI dan interface dari proses-proses yang berkaitan dengan non-TI. Implementasi dari arsitektur ini mungkin tidak perlu dalam urutan yang sama, diutamakan terlebih dahulu yang begitu sangat dibutuhkan.

Tujuan dari arsitektur data adalah untuk mendefinisikan tipe dan sumber utama data yang diperlukan untuk mendukung bisnis dengan cara yaitu dapat dipahami oleh stakeholder, lengkap, kosisten, dan stabil. Penting untuk diketahui bahwa arsitektur ini tidaklah memperhatikan perancangan database. Tujuannya adalah untuk mendefinisikan entitas data yang relevan dengan enterprise, bukanlah untuk merancang sistem penyimpanan fisik dan logik.

Beberapa langakah yang diperlukan untuk membuat arsitektur data adalah:

- Mengembangkan deskripsi arsitektur data dasar
- Review dan validasi prinsip, reference model, sudut pandang dan tools.

- Membuat model arsitektur
- Memilih arsitektur data building block
- Melengkapi arsitektur data
- Melakukan gap analysis arsitektur data saat ini dengan arsitektur data target dan membuat laporan.

Tujuan dari arsitektur aplikasi adalah untuk mendefinisikan jenis-jenis utama dari sistem aplikasi yang penting untuk memproses data dan mendukung bisnis. Penting untuk diketahui bahwa arsitektur aplikasi ini tidaklah memperhatikan perancangan sistem aplikasi. Tujuannya adalah untuk mendefinisikan jenis-jenis sistem aplikasi yang relevan dengan enterprise dan aplikasi apa saja yang diperlukan untuk mengatur data dan menghadirkan informasi kepada aktor manusia dan komputer di enterprise. Aplikasi tidak diuraikan sebagai sistem komputer tetapi sebagai grup logik dari kemampuan untuk mengatur objek data dalam arsitektur data dan mendukung fungsi-fungsi bisnis dalam arsitektur bisnis. Aplikasi dan kemampuan didefinisikan tanpa mereferensikan ke teknologi khusus. Suatu aplikasi bersifat stabil dan relatif tidak berubah sepanjang waktu sedangkan teknologi yang digunakan untuk mengimplementasikannya akan barubah sepanjang waktu, berdasarkan pada teknologi yang sekarang tersedia dan perubahan kebutuhan bisnis.

Beberapa langkah yang diperlukan untuk membuat arsitektur aplikasi adalah :

- Mengembangkan deskripsi arsitektur aplikasi dasar
- Review dan validasi prinsip, reference model, sudut pandang dan tools.
- Membuat model arsitektur
- Indentifikasi sistem aplikasi kandidat
- Melengkapi arsitektur aplikasi
- Mealakukan gap analysis dan membuat laporan

5. Phase D: Technology Architecture –Membangun arsitektur teknologi yang diinginkan, dimulai dari penentuan jenis kandidat teknologi yang diperlukan dengan menggunakan Technology Portfolio Catalog yang meliputi perangkat

lunak dan perangkat keras. Dalam tahapan ini juga mempertimbangkan alternatif-alternatif yang diperlukan dalam pemilihan teknologi.

Beberapa langkah yang diperlukan untuk membuat arsitektur teknologi yaitu:

- Membuat deskripsi dasar dalam format TOGAF
- Mempertimbangkan reference model arsitektur yang berbeda, sudut pandang dan tools.
- Membuat model arsitektur dari building block
- Memilih services portfolio yang diperlukan untuk setiap building block
- Mengkonfirmasi bahwa tujuan bisnis tercapai
- Menentukan kriteria pemilihan spesifikasi
- Melengkapi definisi arsitektur
- Melakukan gap analysis antara arsitektur teknologi saat ini dengan arsitektur teknologi target.

6. Phase E: Opportunities and Solutions – Pada tahap ini akan dievaluasi model yang telah dibangun untuk arsitektur saat ini dan tujuan, indentifikasi proyek utama yang akan dilaksanakan untuk mengimplementasikan arsitektur tujuan dan klasifikasikan sebagai pengembangan baru atau penggunaan kembali sistem yang sudah ada. Pada fase ini juga akan direview gap analysis yang sudah dilaksanakan pada fase D.

Tujuan dari fase ini adalah :

- Mengevaluasi dan memilih pilihan implementasi yang diidentifikasikan dalam pengembangan arsitektur target yang bervariasi
- Identifikasi parameter strategik untuk perubahan dan proyek yang akan dilaksanakan dalam pergerakan dari lingkungan saat ini ke tujuan.
- Menafsirkan ketergantungan, biaya dan manfaat dari proyek-proyek yang bervariasi.
- Menghasilkan sebuah implementasi keseluruhan dan strategi migrasi dan sebuah rencana implementasi detail.

7. Phase F: Migration and Planning – Pada fase ini akan dilakukan analisis resiko dan biaya. Tujuan dari fase ini adalah untuk memilih proyek implementasi yang bervariasi menjadi urutan prioritas. Aktivitas mencakup penafsiran ketergantungan, biaya, manfaat dari proyek migrasi yang bervariasi. Daftar prioritas proyek akan berjalan untuk membentuk dasar dari perencanaan implementasi detail dan rencana migrasi.

8. Phase G: Implementation Governance – fase ini mencakup pengawasan terhadap implementasi arsitektur.

Tujuan dari fase ini adalah :

- Untuk merumuskan rekomendasi dari tiap-tiap proyek implementasi
- Membangun kontrak arsitektur untuk memerintah proses deployment dan implementasi secara keseluruhan
- Melaksanakan fungsi pengawasan secara tepat selagi sistem sedang diimplementasikan dan dideploy
- Menjamin kecocokan dengan arsitektur yang didefinisikan oleh proyek implementasi dan proyek lainnya.

9. Phase H: Architecture Change Management – fase ini mencakup penyusunan prosedur-prosedur untuk mengelola perubahan ke arsitektur yang baru. Pada fase ini akan diuraikan penggerak perubahan dan bagaimana memanajemen perubahan tersebut, dari pemeliharaan sederhana sampai perancangan kembali arsitektur. ADM menguraikan strategi dan rekomendasi pada tahapan ini. Tujuan dari fase ini adalah untuk menentukan/menetapkan proses manajemen perubahan arsitektur untuk arsitektur enterprice yang baru dicapai dengan kelengkapan dari fase G. Proses ini akan secara khusus menyediakan monitoring berkelanjutan dari hal-hal seperti pengembangan teknologi baru dan perubahan dalam lingkungan bisnis dan menentukan apakah untuk menginisialisasi secara formal siklus evolusi arsitektur yang baru. Fase H juga menyediakan perubahan kepada framework dan pendirian disiplin pada fase Preliminary.

10. Requirements Management – menguji proses pengelolaan architecture requirements sepanjang siklus ADM berlangsung

PERTEMUAN 10
ARSITEKTUR ENTERPRISE MENGGUNAKAN TOGAF

Teknologi Informasi (TI) telah menjadi bagian penting dalam suatu organisasi, terutama bagi organisasi yang bisnisnya berorientasi pada keuntungan. Saat ini, infrastruktur bisnis tidak dapat dipisahkan dari Teknologi Informasi. Tetapi yang perlu diperhatikan adalah pengembangan komponen-komponen Teknologi Informasi yang digunakan dalam suatu organisasi. Dalam banyak organisasi setiap unit fungsional organisasi membutuhkan aplikasi teknologi informasi yang berbeda yang kemudian dikembangkan berdasarkan gaya dan platform masing-masing.

Hal ini dapat menyebabkan organisasi tersebut memiliki beberapa macam aplikasi Teknologi Informasi yang berbeda dan kurang terintegrasi antara yang satu dengan yang lainnnya. Selain itu, dalam banyak organisasi sering terjadi kurang selarasnya pengembangan komponen-komponen Teknologi Informasi yang digunakan dengan kebutuhan bisnis dalam organisasi tersebut. Oleh sebab itu, diperlukan adanya suatu perancangan yang terintegrasi dan menyeluruh dengan menggunakan perancangan Enterprise Architecture.

Salah satu kerangka kerja yang dapat digunakan untuk pengembangan Enterprise Architecture adalah kerangka kerja The Open Group Architecture Framework (TOGAF).

Kerangka kerja TOGAF merupakan kerangkakerja dengan metode yang diterima secara luas dalam pengembangan Enterprise Architecture. Kerangka kerja TOGAF memberikan metode yang rinci mengenai bagaimana membangun, mengelola, dan menerapkan Enterprise Architecture yang dikenal dengan Architecture Development Method (ADM).

Kita telah membahas mengenai architecture dan transformation, pada pembahasan ini akan dibahas kembali mengenai istilah “architecture” dan isinya. Dalam pengantarnya, TOGAF memberikan definisi mengenai “architecture”:

1. “A formal description of a system, or a detailed plan of the system at component level, to guide its implementation.”
2. “The structure of components, their inter-relationships, and the principles and guidelines
governing their design and evolution over time.”

Definisi pertama menganggap istilah "arsitektur" sebagai sinonim dari "deskripsi sistem." Untuk yang kedua, "arsitektur" menunjuk struktur dan prinsip-prinsip sistem, terlepas dari deskripsinya.

Definisi ganda ini mungkin tampak mengejutkan, tetapi mencerminkan situasi yang sangat nyata. Sistem perangkat lunak, pada dasarnya, buram, sedemikian rupa sehingga strukturnya hanya terlihat melalui representasi.

Berdasarkan definisi-definisi tersebut dapat disimpulkan bahwa Enterprise Architecture adalah kumpulan prinsip, metode, dan model yang dapat digunakan untuk mendisain dan merealisasikan sebuah struktur organisasi enterprise, proses bisnis, teknologi informasi dan infrastrukturnya. Enterprise Architecture memiliki empat komponen atau domain utama yaitu arsitektur bisnis, arsitektur informasi, arsitektur aplikasi, dan arsitektur teknologi. Enterprise Architecture mempunyai arti penting bagi organisasi karena salah satu hasilnya adalah terwujudnya keselarasan antara teknologi informasi dan kebutuhan bisnis.

Pabrik, kapal, atau mesin semuanya memiliki struktur fisik, yang lebih atau kurang terlihat. Namun, tidak mungkin untuk "lift the hood" dari sistem TI, yang arsitekturnya hanya ada melalui perwakilannya. Ini juga merupakan kasus untuk elemen bisnis seperti proses, organisasi, atau strategi, yang hanya dapat dikomunikasikan melalui deskripsi atau model. Proliferasi skema, diagram, dan tabel dalam perusahaan adalah bukti dari kenyataan ini.

Dalam konteks ini, komunikasi pada arsitektur memainkan peran menentukan. Sama seperti cetak biru sebuah bangunan, komunikasi adalah alat vital bagi mereka yang mengerjakan berbagai tugas yang terlibat: pengembangan, evaluasi, pertukaran, dan konstruksi.

- **Domain and Phases**

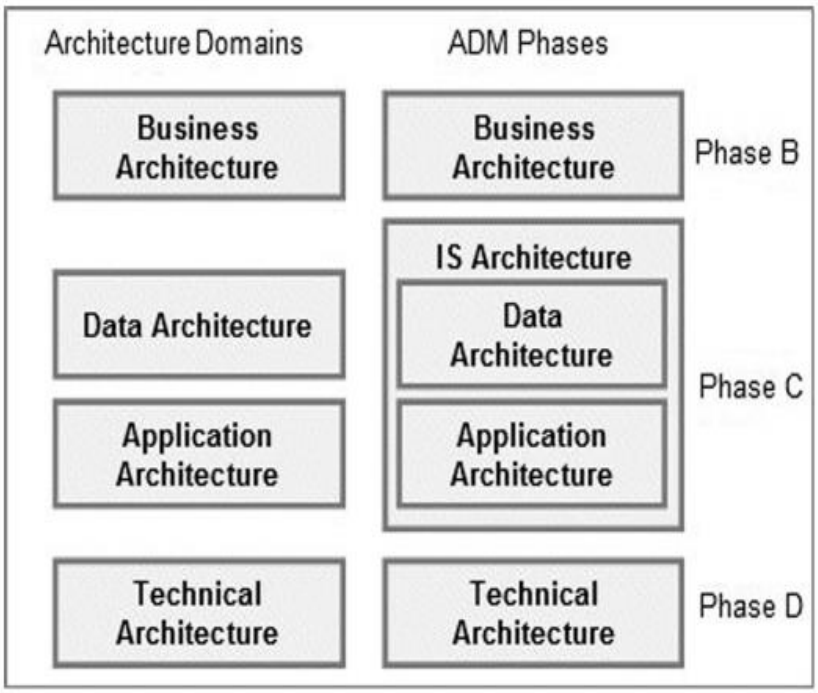
Apa saja subjek utama pada area di dalam enterprise architecture? TOGAF mengusulkan high-level breakdown ke dalam empat bagian besar:

1. **Business architecture**, yang mencakup strategi, tujuan, proses bisnis, fungsi, dan organisasi.
2. **Data architecture**, didedikasikan untuk organisasi dan manajemen informasi.
3. **Application architecture**, menyajikan applications, software components, dan interaksinya.
4. **Technology architecture**, yang menjelaskan teknik dan komponen yang digunakan, serta jaringan dan infrastruktur fisik tempat aplikasi dan sumber data berjalan..

- **Architecture repository**

Secara alami, enterprises perlu mengkonservasi, menyebar, dan menggunakan kembali informasi EA yang merupakan salah satu aset utama mereka. Ini adalah peran repositori arsitektur, yang mencakup deskripsi dari masing-masing dari empat domain, serta seluruh host pengetahuan, prinsip panduan, dan teknik yang terkait dengan arsitektur enterprise.

Sebelum menjadi sumber informasi yang statis, repository terus berkembang sepanjang transformasi arsitektur, sehingga berpartisipasi dalam kapitalisasi pengetahuan. Ini juga memberikan gambaran arsitektur, yang memfasilitasi pengambilan keputusan pada level strategis.



Untuk pembaca TOGAF, satu item kosa kata masih harus dijelaskan. TOGAF sering merujuk pada arsitektur solusi. Di sini, istilah "arsitektur" menunjuk deskripsi, dan lebih tepatnya pandangan logis, sebagai lawan dari "solusi," yang mewakili realitas teknis.

Perbedaan ini dapat dilihat dengan jelas dalam istilah “Architecture Building Block” dan “Solution Building Block” (respectively ABB and SBB). Spesifikasi logis suatu elemen adalah ABB, sedangkan padanan fisiknya adalah SBB. Kedua jenis elemen ini ada dalam repositori arsitektur, yang memungkinkan dokumentasi atau komponen fisik untuk digunakan kembali, sesuai dengan konteksnya.

Goals, constraints, and requirements

Agar berhasil melakukan operasi transformasi, kita harus benar-benar jelas tentang hasil yang ingin kita dapatkan. Pernyataan ini mungkin tampak sepele,

tetapi perlu diingat. Dalam domain ini, TOGAF membedakan serangkaian elemen yang berpartisipasi dalam formalisasi yang lebih terstruktur:

• **Strategic objectives**, or goals, yang menggambarkan orientasi umum.
• **Operational objectives**, or objectives, yang memformalkan tujuan-tujuan ini dalam hasil yang terukur.
• **Drivers**, yang sering memotivasi keputusan mengenai perubahan arsitektur, seperti perubahan dugaan atau kebutuhan untuk beradaptasi dengan evolusi teknis. Inilah "mengapa", yang membenarkan dan mengarahkan tujuan.
• **Requirements**, yang menentukan dengan tepat apa yang akan diterapkan secara konkret untuk mencapai tujuan-tujuan ini.
• **Constraints**, yang merupakan elemen eksternal yang mempengaruhi sistem, terkadang menahan kapasitasnya

Stakeholders and the human factor

Kita tahu bahwa aspek organisasi adalah salah satu poin paling rumit dalam jenis operasi ini. Seperti halnya proses perusahaan, transformasi arsitektur melibatkan kombinasi kegiatan yang melibatkan peserta yang berbeda, masing-masing dari mereka adalah “pemangku kepentingan” dalam operasi yang dia lakukan. TOGAF menangani pertanyaan ini melalui hal-hal berikut:

1. Stakeholder management
2. Transformation Readiness Assessment
3. Efficiency of communication through the concept of viewpoints

1) **Managing Stakeholder**

   Pertama, penting untuk mendefinisikan dengan jelas setiap pemangku kepentingan sedini mungkin selama siklus ADM. Identifikasi ini terutama menggunakan pendekatan pragmatis untuk menghindari hanya menggunakan kembali struktur organisasi yang ada, yang hanya sebagian mewakili realitas kegiatan dan tanggung jawab yang akan digerakkan.

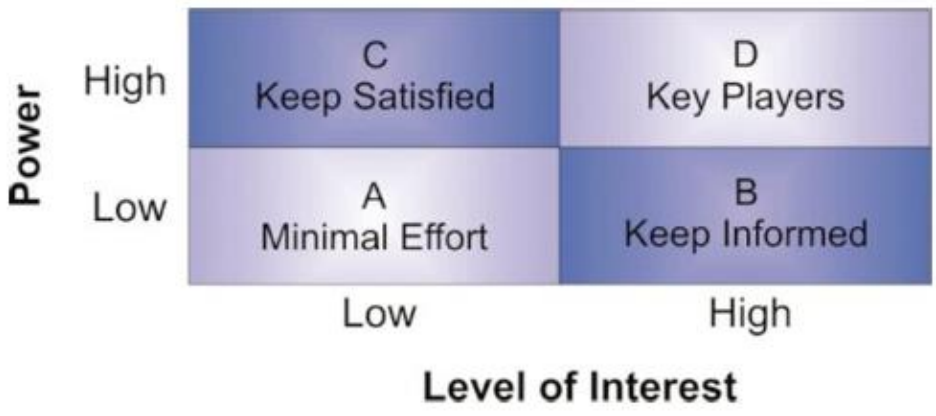
Meninggalkan peserta kunci di pinggir jalan akan secara signifikan mempengaruhi kualitas hasil.

Akibatnya, untuk menentukan dengan siapa dan dalam bentuk apa pekerjaan akan dilakukan, serangkaian pertanyaan kunci harus dijawab pada subjek:

- Who defines goals?
- Who gains and who loses from this change?
- Who controls the transformation process?
- Who designs new systems?
- Who will make the decisions?
- Who procures IT systems and who decides what to buy?
- Who controls resources?
- Who has or controls the necessary specialist skills?
- Who influences the project?

Dari pertanyaan-pertanyaan ini, TOGAF merekomendasikan agar posisi masing-masing pemangku kepentingan diklarifikasi, terutama tingkat keterlibatannya. Gambar ini menyajikan berbagai tingkatan ini. Setiap pemangku kepentingan akan diposisikan menggunakan tingkat keterlibatan ini, yang menentukan hubungan untuk dikembangkan dan tingkat keterlibatan dalam komite pengarah proyek arsitektur.

Secara alami, pemain kunci memainkan peran menentukan dan harus berada di garis depan dalam semua bidang pengambilan keputusan.



Kualifikasi ini cross-referenced dengan peran yang dimainkan dalam konteks proyek saat ini:

• Manajemen eksekutif, yang menetapkan tujuan strategis

• Klien, yang bertanggung jawab atas anggaran yang dialokasikan, sehubungan dengan tujuan yang diharapkan

• Pengguna, yang secara langsung berinteraksi dengan sistem selama kegiatannya

• Penyedia, yang memberikan elemen komponen arsitektur, terutama perangkat lunaknya

komponen

• Sponsor, yang mengendarai dan memandu pekerjaan

• Arsitek perusahaan, yang mengubah tujuan bisnis menjadi kenyataan dalam struktur sistem

2) Transformation Readiness Assessment

• Apakah organisasi siap untuk perubahan yang dibayangkan? Pertanyaan ini mungkin kelihatannya tidak sesuai, tetapi berapa banyak proyek yang gagal diselesaikan karena dimensi ini tidak diperhitungkan? Hanya dalam beberapa tahun, "manajemen perubahan" telah menjadi disiplin tersendiri, menghasilkan sejumlah besar artikel dan seminar.

- TOGAF sepenuhnya didedikasikan untuk pertanyaan ini, yang dibahas secara luas dalam deskripsi fase ADM. Identifikasi risiko resistensi perubahan dan definisi tindakan yang harus diambil untuk membatasi risiko ini adalah tugas penting yang harus dilakukan sebelum memulai proyek transformasi. Ini sangat penting untuk operasi yang mencakup ruang lingkup luas dan menghasilkan restrukturisasi yang signifikan.
- Meskipun tidak mungkin untuk memberikan *turnkey solution* pada subjek seperti itu, dimungkinkan untuk menggunakan teknik tertentu yang akan membantu mengurangi jenis risiko ini:
    1. Presentasi yang jelas tentang dampak perubahan yang dilakukan, terutama pada tingkat organisasi

2. Pandangan konkret tentang manfaat bisnis yang diharapkan, dalam bentuk "kasus bisnis"
3. Penilaian obyektif atas TI perusahaan, bisnis, dan bakat keuangan, tanpa estimasi kapasitas sebenarnya yang berlebihan
4. Tim manajemen eksekutif diakui mampu mempertahankan proyek dalam jangka panjang
5. Komunikasi berkualitas tinggi, yang bertujuan untuk mempromosikan pemahaman bersama tentang pasak dan solusi untuk diterapkan

3) Views and viewpoints

- Jika pesan ingin dipahami dengan sukses, aspek yang paling penting untuk dipertimbangkan adalah bahwa konten dan bentuknya harus disesuaikan dengan penerima yang dimaksud.
- Untuk ini, TOGAF menggunakan konsep viewpoints. Viewpoints menunjuk perspektif yang paling tepat untuk peserta tertentu, dan diwujudkan oleh sejumlah pandangan arsitektur, dalam bentuk diagram, dokumen, atau elemen lainnya. Sebagai contoh, manajemen eksekutif akan lebih tertarik pada deskripsi tingkat tinggi, sementara komunikasi dengan staf operasional akan membutuhkan representasi yang jauh lebih rinci.
- Ini adalah titik kritis, yang akan mengkondisikan kualitas komunikasi dan yang akan ditemui selama setiap fase siklus ADM. Akibatnya, sangat penting bahwa views dan viewpoints harus ditentukan untuk masing-masing pemangku kepentingan sebelum mulai bekerja pada empat domain arsitektur (bisnis, data, aplikasi, dan teknologi).

Pertemuan 11

Metode ADM, Fase pada TOGAF ADM

Metode ADM

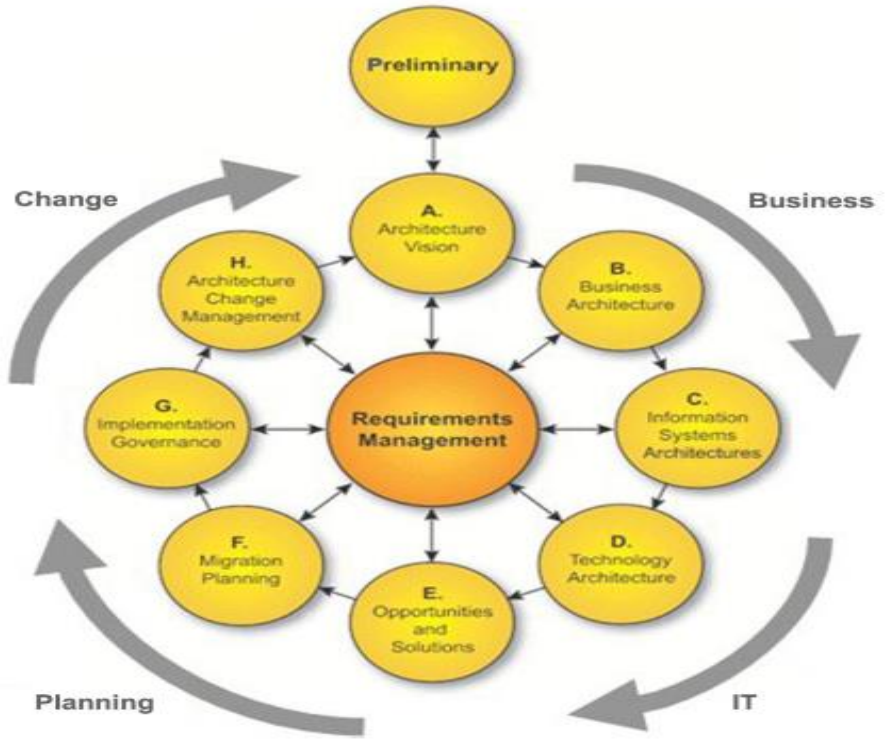
Metode pada ADM dibagi menjadi delapan sequential phases (A to H) dan dua phase khusus lainnya: preliminary phase dan requirements management phase. Gambar 1 menampilkan diagram TOGAF yang paling sering dirujuk, yang merangkum pendekatan ini melalui pemecahan empat bagian tingkat tinggi: bisnis, teknologi informasi (TI), perencanaan, dan perubahan. Urutan fase A ke H dipecah sebagai berikut:

- Phase A: Vision
- Phase B: Business architecture
- Phase C: Information system architecture
- Phase D: Technology architecture
- Phase E: Opportunities and solutions
- Phase F: Migration planning
- Phase G: Implementation governance
- Phase H: Architecture change management

Semua phases dideskripsikan seperti ini:

- The objectives, which define the expected results
- The approach, which provides a guide and recommended strategy
- The input and output, which specify what each phase consumes or modifies
- The different steps, in the form of a breakdown of the work to be carried out

Meskipun perkembangan fase dijelaskan secara berurutan (dari A ke H), urutan ini dapat ditinjau dan disesuaikan sesuai dengan konteksnya, terutama dalam bentuk iterasi dalam siklus ADM.

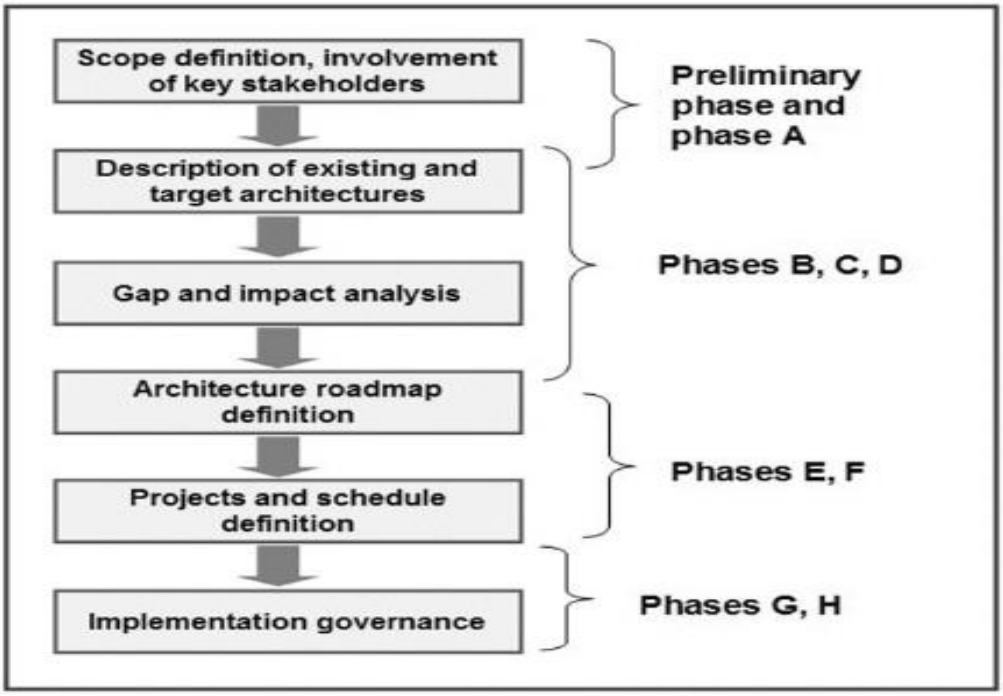


Gambar 1. Architecture development method (ADM)—TOGAF9

- Meskipun perkembangan fase dijelaskan secara berurutan (dari A ke H), urutan ini dapat ditinjau dan disesuaikan sesuai dengan konteksnya, terutama dalam bentuk iterasi dalam siklus ADM. Secara lebih umum, diagram “crop cycle" harus dianggap sebagai struktur referensi alih-alih suatu perkembangan yang tidak berubah, terutama karena tetap mungkin, dan bahkan lebih disukai, untuk mempertanyakan atau menyesuaikan kapan saja bagian dari hasil yang diperoleh sebelumnya.

- Identifikasi dari kendala baru atau formulasi ulang atau penambahan rincian persyaratan dapat menyebabkan aspek baru tertentu muncul, aspek yang tidak cukup dieksploitasi selama fase sebelumnya. Sebagai contoh, dokumen keluaran utama fase A, “architecture vision,” hanya divalidasi secara definitif selama fase F. Namun, high-quality elaboration

menyiratkan perkembangan konvergen, yang tidak mempersoalkan prinsip-prinsip dan foundations yang didefinisikan pada permulaan.



Gambar 2. Typical path of an ADM cycle.

Gambar menyajikan tinjauan umum tentang perkembangan siklus ADM, dari fase awal hingga fase H. Jalur tipikal ini dipandu oleh satu tujuan utama: kebutuhan untuk mendapatkan hasil yang diharapkan dengan menguasai setiap langkah proses. Tujuan ini membutuhkan persiapan yang ketat, deskripsi target berkenaan dengan apa yang sudah ada untuk semua aspek (bisnis, sistem informasi, dan teknologi), evaluasi yang tepat dari kesenjangan dan risiko menentukan pilihan lintasan, dan akhirnya evaluasi hasil dan manajemen yang cermat dari setiap penyesuaian yang dilakukan.

- TOGAF menjelaskan setiap langkah dari setiap fase secara terperinci. Ini tidak berarti bahwa realisasi dari setiap langkah ini sistematis. Beberapa hasil sudah tersedia, baik karena dihasilkan oleh entitas lain atau karena terkait dengan kegiatan yang lebih umum. Misalnya, jika prinsip-prinsip arsitektur dicatat sebagai hasil dari fase A, prinsip-prinsip tersebut dapat dengan mudah diperiksa di mana mereka sudah ada. Hal utama di sini

adalah bahwa keberadaan dan kesesuaian setiap hasil harus diperiksa, seperti daftar periksa yang terkait dengan setiap fase.

Fase-Fase Pada TOGAF ADM

1. The preliminary phase

Tujuan pada phase ini adalah untuk mempersiapkan enterprise untuk merealisasikan pekerjaan architecture :

• The organization and governance of the architecture
• General principles
• Methods
• Tools
• The architecture repository
• The start of the ADM cycle

Dengan cara ini, preliminary phase bukan bagian dari siklus ADM tetapi dapat dipertimbangkan kapan saja selama siklus ADM sebagai bagian dari evolusi praktik EA, terutama dalam konteks menggunakan maturity model sebagai cara mengidentifikasi peluang untuk inisiatif transisi. Kegiatan pada preliminary phase pada dasarnya cross-organizational, terkait dengan tata kelola umum arsitektur perusahaan, dan tujuannya adalah untuk memungkinkan perusahaan menguasai manajemen dan transformasi arsitekturnya. Namun, selama fase pendahuluan itulah dimulainya siklus ADM tertentu dan disiapkan. Ini dirinci dalam dokumen "Request for Architecture Work", yang berisi semua elemen yang membentuk dasar dari proyek perubahan arsitektur perusahaan (sponsor, tujuan strategis, kendala, kerangka anggaran, dan rencana strategis). Dokumen ini merupakan panduan referensi kontrak untuk kemajuan seluruh siklus TOGAF itu sendiri, dari fase A dan seterusnya.

Akhirnya, TOGAF memberikan ringkasan yang baik dari fase ini mencakup where, what, why, who, and how.

2. Phase A (vision)

Phase A adalah fase pertama dari ADM cycle, yang akan memvalidasi “Request for Architecture Work” document. Phase A memiliki dua tujuan utama:

- Pertama, fase A selanjutnya mengembangkan dan memperkaya elemen-elemen yang dihasilkan dari fase awal, seperti prinsip arsitektur, indikator utama, dan organisasi atau perencanaan kerja elaborasi.
- Kedua, fase A menyiapkan fase selanjutnya dengan menyediakan representasi umum dari dasar dan target arsitektur. Pada tahap ini, adalah high-level representations, dimana tujuannya untuk highlight structuring points and typical solutions.

Komunikasi memainkan peran penting selama fase ini. Semua pemangku kepentingan harus memiliki pemahaman yang sama, untuk mendapatkan konsensus tentang orientasi dan hasil yang diharapkan. Poin-poin lain juga dibahas, seperti fundamental requirements, kaitannya dengan tujuan strategis, atau manajemen risiko. Dokumen berupa "Architecture vision" merupakan bentuk output dari fase ini.

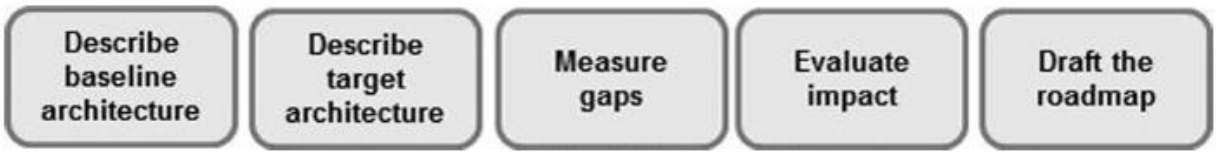
Dapat disimpulkan, pada akhir fase A visi mencakup:

- Organization: the stakeholders, their roles, their respective involvement
- Orientation: a consensus on the principles, goals, major requirements, and constraints
- The scope covered, the most impacted parts
- The roadmap: the ADM cycle development plan, the resources, and the budget allocated
- A macroscopic vision of baseline architecture and target architecture
- Major risks and associated risk reduction actions

In other words, we know where we're going, how we're getting there, and with whom. Note that at this stage the perspective is horizontal, and covers all architecture domains (business, information system, and technology), unlike the following three phases, which operate vertically, focusing on one particular domain

2. Phases B, C, and D (Elaboration of Business, Information System, and Technology Architectures)

- Most of the content of the following three phases—B (Business), C (Information System), and D (Technology)—consists in detailing the target and baseline architecture, measuring the gap between the two, and evaluating the impact of change on all facets of the enterprise. The combination of these elements is used to draft the roadmap for transition. This first draft of the roadmap is elaborated progressively throughout phases B, C, and D, and serves as the foundation for phases E and F, which are in charge of defining the transformation plan (Figure 3).
- Each phase begins with the definition of the views that will be used to materialize the baseline and target architectures. Remember that the goal of these views is to adapt the representations of the architecture to each stakeholder's viewpoint.
- Architecture descriptions are consigned to the architecture definition document (central document). This document is enriched during each phase, before being finalized and validated prior to the start of migration work. Concretely, each phase will complete the chapter(s) that concerns it, so that the document spans all architecture domains.
- Of course, as well as depending on each situation, the choice of target architecture also integrates recurrent questions. Consequently, TOGAF recommends that the repository be reviewed before each decision in order to reuse the experience accumulated during earlier work wherever possible. This repository review is noted as a "checklist" action at the start of each phase, so as to conform to the norms in place within the enterprise and to promote general harmonization.



Gambar 3. Main activities common to phases B, C, and D.

Impact assessment should be considered in a cross-organizational way, for two reasons. First, because each phase evaluates its impact beyond its own scope. During phase B, for example, the impact of evolutions on technical elements is also assessed. If, for example, the executive management team decides to remove a product range, it is easy to work out the consequences of this decision on the corresponding database. Second, because the sheer number of relationships within an enterprise can lead to all sorts of unexpected side effects on entities outside the initial scope.

Phase B (business architecture)

The structural similarity of phases B, C, and D should not detract from the determining role of phase B, since it is the business that drives the architecture in all its forms. The formalization of business elements (requirements, processes, entities) is the prelude to all valid logical or technical constructions. This is all the more true when we consider that the goal of phase B is also to demonstrate the pertinence of the work being carried out. Goals are established during the earlier phases, but it is only when business architecture elements are precisely developed that the target solution can be installed and its consequences observed. For example, the description of modifications carried out on a business process shows the real-life result of these modifications on tasks run by operators, new services to provide,
or modifications applied to exchanged information.

In terms of architecture descriptions, phase B mainly concentrates on the following elements:

- Business motivation elements (drivers, goals, objectives)
- Organizational units
- Business functions and services
- Business processes
- Business roles and actors
- Business entities

Business entities describe key business concepts and provide the essential entry point to phase C (in the Data Architecture subphase). Business processes are often the key to understanding an enterprise's real activity, and by extension its architecture.

Phase C (information systems architecture)

Information system architecture is a kind of bridge between the business view and its physical translation. It defines software components (applications and data) that support the automation or realization of business capabilities and functions, without integrating technological realities (this point is dis-cussed in the Phase D part).

Remember that phase C (information system architecture) is itself composed of two subphases: data architecture and application architecture.

These two facets (data and application) are reunited in a single phase because of their proximity in the construction of information system architecture. One of the expected results consists in allocating each data group to one application component, which will handle its management, becoming, as it were, the owner of the data group in question.

Phase D (technology architecture)

- Peran fase D adalah untuk membangun teknologi dan fisik dari unsur-unsur yang dikembangkan selama fase sebelumnya. Secara khusus, arsitektur teknologi mendefinisikan platform dan lingkungan eksekusi tempat aplikasi dijalankan dan sumber data di-host untuk digunakan.

- So what are the links between application architecture and technology architecture? A first approach consists in considering them as two separate elements, so as to avoid any technical "intrusion" into the work of the

application architect. The opposite approach would lead us to consider application architecture as a simple reformulation of the technical reality.

- A position that is too dogmatic will lead to a dead end: What is the point of developing a "virtual" application architecture with no link to the reality of the deployed applications? Common sense (and purse strings) calls for more realism. Even though it must remain logical, application architecture (including its service-oriented architecture (SOA) formulation) is not completely separate from its physical translation. The most important thing here is the identification of the role of each application or component, independent of its technical implementation: the fundamental structure is similar and the viewpoint is different, just like a logical service interface, which is not fundamentally modified by its implementation in Java or via a web service.

- Bearing in mind these two perspectives, a question comes to mind: Should we start by describing the technical architecture or the application architecture? This point is linked to the iterations of the ADM cycle, which will be more generally dealt with in Section 2.3. Remember that the ADM cycle is a generic framework, which does not forbid intrusions into earlier or later phases (the TOGAF document is strewn with suggestions of this type). In practice, no preestablished choices exist: this is the famous choice between "top down" and "bottom up," which always finishes with a compromise. The deployment of external tools imposes a type of architecture that can sometimes have a significant impact on application architecture solutions. In other contexts, architecture will be more oriented by architectural principles, for example to obtain a more progressive structure.

- However, let's get back to the result of phase D: the technological architecture, in other words, a coherent set of software components, infrastructures, and technical platforms. These elements can come from

external providers or be produced directly by teams within the enterprise. Moreover, the choice between deploying tools that are available in the marketplace or tools resulting from specific developments is a recurrent theme for an enterprise architect. Here too, the repository will assist in this type of choice by making available a set of common norms, patterns, tools, and practices, which will help harmonize solutions within the enterprise.

3. Phases E and F (opportunities and solutions, migration planning)

- Pada titik siklus ADM ini, realisasi operasional transformasi arsitektur harus benar-benar dimulai: proyek disiapkan, jadwal ditetapkan, sumber daya diidentifikasi, dan pemantauan operasional dilakukan. Fase sebelumnya telah memberikan target, roadmap keseluruhan, dan sekarang implementasi konkretnya harus ditentukan.
- Fase E dan F melihat penjadwalan dan pengorganisasian implementasi arsitektur baru. Penekanan ditempatkan pada membangun jalur migrasi, yang harus membawa manfaat bisnis sejati untuk setiap langkah.
- Selama fase E, hasil fase elaborasi (B, C, dan D) dikonsolidasikan: arsitektur, persyaratan, dan kesenjangan. Konsolidasi ini merupakan bahan baku yang digunakan untuk menentukan arsitektur transisi, sambil mengingat kemampuan perusahaan untuk perubahan (misalnya, aplikasi baru untuk mengembangkan dan evolusi aplikasi yang ada, sesuai dengan cakupan fungsi bisnis). Kelayakan teknis dan organisasi, kompromi antara persyaratan dan biaya, dan kendala integrasi juga dipelajari.
- Fase F secara tepat menetapkan penjadwalan migrasi, serta konstitusi implementasi proyek dan organisasi, tujuan, dan biaya.

4. Phases G and H (implementation governance, architecture change management)

- Fase G menetapkan versi definitif kontrak arsitektur dengan implementasi proyek, termasuk rekomendasi dari dewan arsitektur. Kontrak yang

ditandatangani ini merupakan dasar untuk tinjauan kesesuaian implementasi proyek.

- Fase H menangani manajemen arsitektur yang digunakan: manajemen perubahan, termasuk evaluasi permintaan perubahan yang berdampak pada arsitektur. Perlu dicatat bahwa permintaan evolusi tertentu dapat menyebabkan siklus ADM baru.