

SLIDE MATA KULIAH ARSITEKTUR ENTERPRISE PROGRAM STUDI TEKNOLOGI INFORMASI



VISI PROGRAM STUDI TEKNOLOGI INFORMASI

Menjadi Program Studi yang unggul dalam pengembangan keilmuan teknologi informasi untuk mendukung ekonomi kreatif tahun 2033.

MISI PROGRAM STUDI TEKNOLOGI INFORMASI

1. Menyelenggarakan pendidikan pada bidang teknologi informasi yang berkualitas.
2. Menyelenggarakan Penelitian di bidang teknologi informasi yang berkualitas.
3. Menyelenggarakan pengabdian masyarakat di bidang teknologi informasi dalam rangka meningkatkan kualitas sumber daya manusia.
4. Mengelola Program Studi secara mandiri dengan tata kelola yang baik.

PROFIL LULUSAN PROGRAM STUDI TEKNOLOGI INFORMASI

1. System Administrator

Mampu dalam melakukan analisa terhadap kebutuhan pengguna sistem jaringan komputer, mengidentifikasi sistem jaringan dengan teknologi yang sesuai, mampu merancang arsitektur, sistem keamanan dan pengujian server, mampu menginstall dan mengkonfigurasi sistem operasi server, file sharing pada server, virtual server serta common network and application services server, membuat kode program server, mengimplementasikan dan memantau kinerja dan keamanan sistem, menginvestigasi dan memperbaiki kerusakan sistem serta mampu mengevaluasi dan melakukan restore system.

PROFIL LULUSAN PROGRAM STUDI TEKNOLOGI INFORMASI

2. Cyber Security Analyst

Mampu menerapkan prinsip perlindungan informasi, prinsip keamanan informasi untuk penggunaan jaringan internet, prinsip keamanan informasi pada transaksi elektronik, mampu menyusun dan melaksanakan dokumen kebijakan keamanan informasi, mampu mengaplikasikan ketentuan/persyaratan keamanan informasi, mengelola log dan Melaksanakan pencatatan asset, Mampu Menerapkan kontrol akses berdasarkan konsep/metodologi yang telah ditetapkan mampu Mengidentifikasi serangan-serangan terhadap kontrol akses dan mampu melakukan instalasi software aplikasi

PROFIL LULUSAN PROGRAM STUDI TEKNOLOGI INFORMASI

3. Object Programmer

Mampu melakukan identifikasi library, komponen atau framework yang diperlukan, dan menggunakan struktur data, Mampu mengimplementasikan user interface dan rancangan entitas serta keterkaitan antar entitas, Mampu menerapkan pemecahan permasalahan menjadi subrutin, menulis kode dengan prinsip sesuai guidelines dan best practices, dan membuat dokumen kode program, Mampu melakukan migrasi ke teknologi baru, debugging, dan menerapkan pemrograman paralel, Mampu melaksanakan pengujian kode program secara statis dan pengujian oleh pengguna (UAT), Mampu memberikan petunjuk teknis kepada pelanggan dan menganalisis dampak perubahan terhadap aplikasi serta menerapkan alert notification jika aplikasi bermasalah.

CAPAIAN PEMBELAJARAN LULUSAN

CPL Program Studi yang dibebankan pada Mata Kuliah

CPL1 (S8)	Menunjukkan sikap bertanggungjawab atas pekerjaan di bidang keahliannya secara mandiri
CPL2 (P3)	Mampu memahami dan menguasai metode pemodelan arsitektur bisnis, business building block, arsitektur teknologi dan technology building block yang diperlukan
CPL3 (KK9)	Mampu menyusun roadmap arsitektur bisnis, serta roadmap dan komponen arsitektur teknologi
CPL4 (KU5)	Mampu mengambil keputusan secara tepat dalam konteks penyelesaian masalah di bidang keahliannya, berdasarkan hasil analisis informasi dan data

Pertemuan 1

Konsep Dasar Arsitektur Enterprise

Arsitektur?

- Arsitektur adalah konsep dasar sebuah sistem dalam lingkungannya terletak pada elemen- elemennya, hubungannya, serta prinsip dari rancangan dan evolusinya
- Arsitektur diperlukan untuk mengelola kompleksitas organisasi atau sistem yang sangat besar:
 1. gambaran struktur organisasi
 2. proses bisnis didalamnya,
 3. dukungan aplikasi, dan
 4. infrastruktur teknis



Arsitektur Enterprise?

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



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

Arsitektur Enterprise?

- Enterprise Architecture (EA) adalah satu kesatuan yang koheren tentang prinsip-prinsip, metode dan model yang digunakan dalam desain dan realisasi dari struktur organisasi, proses bisnis, sistem informasi dan infrastruktur sebuah organisasi (Lankhorst, 2005)
- Enterprise architecture mencakup inti-inti dari bisnis, TI dan evolusinya, tanpa arsitektur yang bagus, akan sulit untuk mencapai kesuksesan bisnis
- Sebuah enterprise architecture yang baik menyediakan kebutuhan mendatang untuk menyeimbangkan dan membantu translasi kebutuhan tersebut dari bentuk strategi perusahaan menjadi operasi sehari-hari

Arsitektur Enterprise?

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

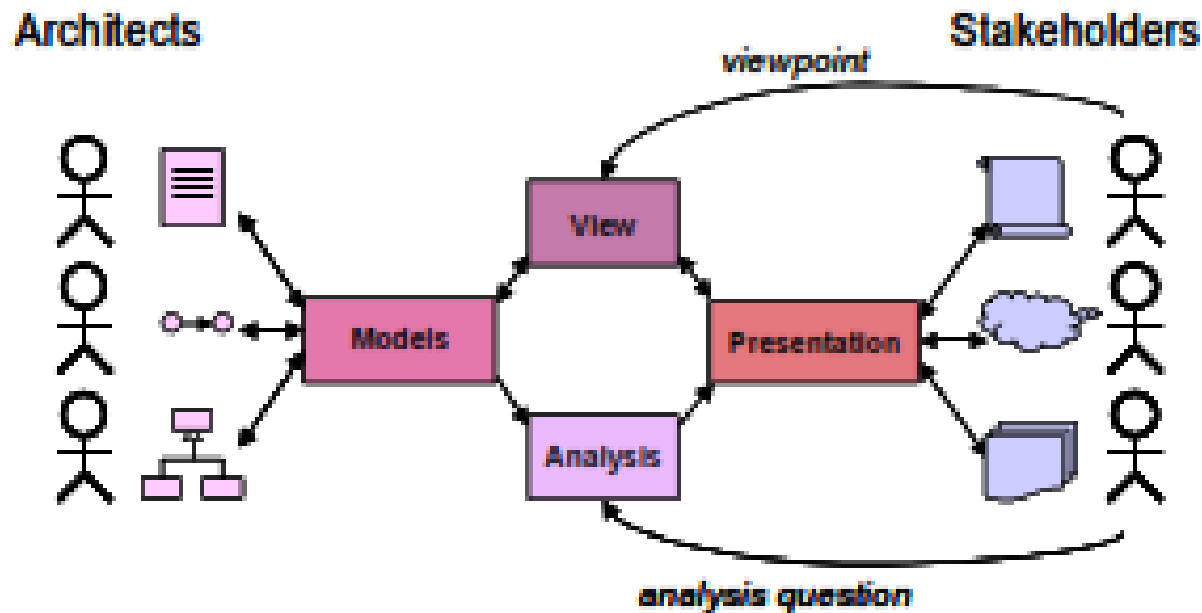


Fig. 1.1. Communicating about architecture.

Arsitektur Enterprise?

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



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



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



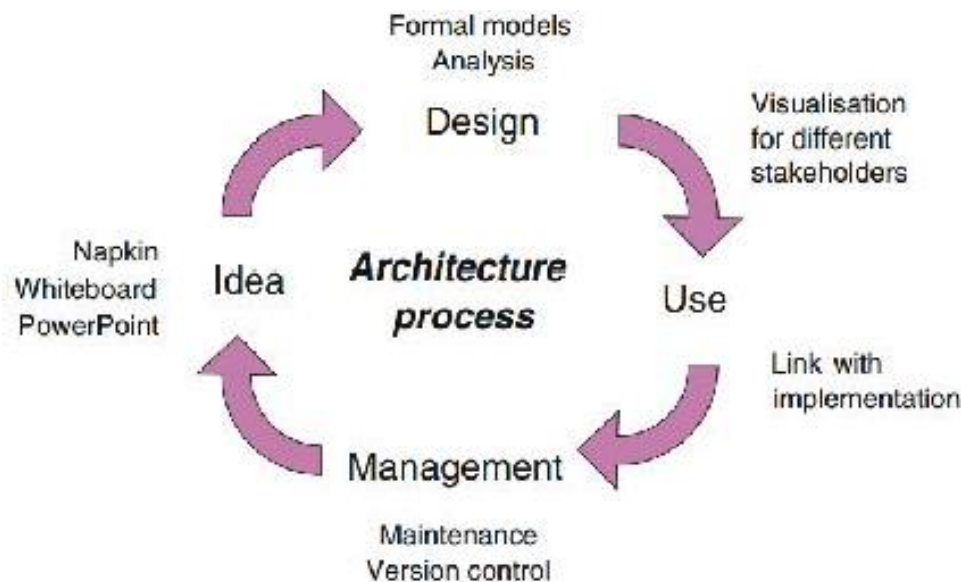
APLIKASI, baik itu aplikasi custom atau generic, yang berjalan di atas ...



TEKNOLOGI, seperti infrastruktur, sistem dan jaringan komputer

Proses Arsitektur

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



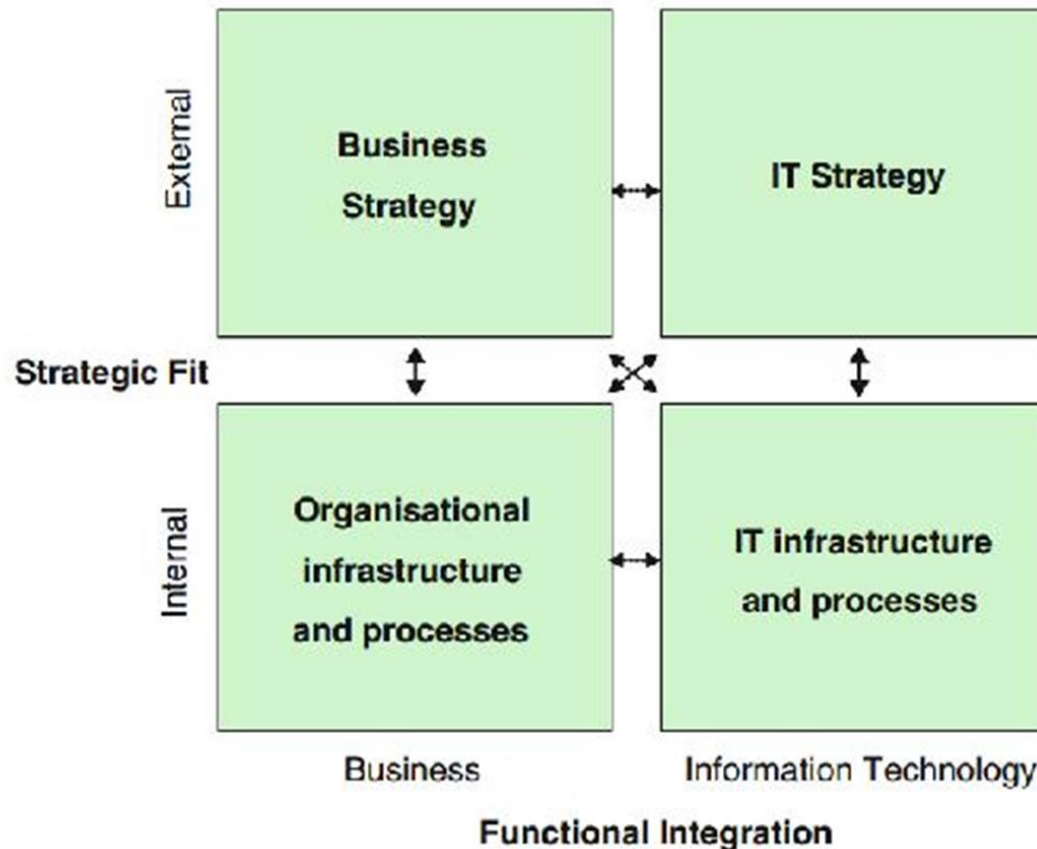
- Dalam semua tahapan proses arsitektur, komunikasi yang jelas dengan dan antara stakeholder menjadi penting

Faktor Pendukung Arsitektur Enterprise (Faktor Internal)

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

- Model penyesuaian strategis oleh Henderson dan Venkatraman (1993) membedakan antara aspek strategi bisnis dan infrastruktur organisasi di satu sisi, dengan strategi dan infrastruktur TI di sisi lainnya
- Model ini menyediakan 4 sudut pandang dominan yang digunakan untuk mengatasi penyesuaian antara aspek2 tersebut

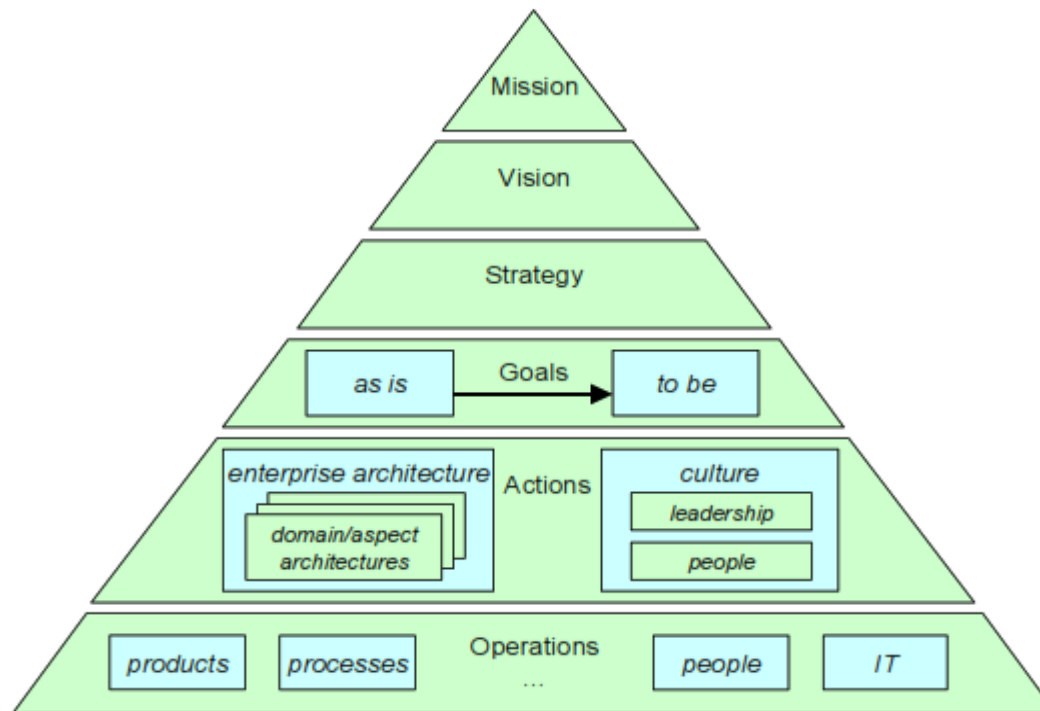
Faktor Pendukung Arsitektur Enterprise (Faktor Internal)



Strategic alignment model (Henderson and Venkatraman 1993)

Faktor Pendukung Arsitektur Enterprise (Faktor Internal)

2. Enterprise architecture diposisikan dalam konteks pengelolaan enterprise



Faktor Pendukung Arsitektur Enterprise (Faktor Internal)

- Pada bagian atas piramida, kita harus melihat misi dari enterprise. **Mengapa?** Karena Visi dan misi menyatakan 'gambaran masa depan' dan nilai2 yang dipegang teguh
- Berikutnya adalah strategi, yang menyatakan jalur yang ditempuh enterprise untuk mencapai misi dan visinya ditranslasikan ke dalam tujuan konkrit yang memberi arah dan menjadi penanda dalam mengeksekusi strategi
- Tujuan ditranslasi menjadi perubahan konkrit hingga ke operasi sehari-hari perusahaan adalah tugas enterprise architecture dalam menyediakan sudut pandang holistik dari operasi saat ini dan saat nanti, serta aksi yang harus dilakukan untuk mencapai tujuan perusahaan

Faktor Pendukung Arsitektur Enterprise (Faktor Internal)

- Arsitektur dipandang sebagai bagian 'hard' dari perusahaan, sementara bagian 'soft' nya adalah culture yang dibentuk oleh orang-orang dan kepemimpinan (atau lebih penting) untuk mencapai tujuan perusahaan
- Sedangkan bagian bawah piramida adalah operasi harian perusahaan yang mencakup **product, proses, people** dan **IT** yang digunakan

Faktor Pendukung Arsitektur Enterprise (Faktor Internal)

3. Arsitektur adalah instrumen strategis dalam memandu organisasi menjalani arah pengembangan yang terencana
- Enterprise yang sukses menjalankan 'operating model' dengan pemilihan yang jelas pada level integrasi dan standarisasi proses bisnis di seluruh enterprise
 - Peran enterprise architecture sebagai pengorganisasian logis dari proses bisnis dan infrastruktur TI, harus mencerminkan kebutuhan integrasi dan standarisasi dari operating model

Faktor Pendukung Arsitektur Enterprise (Faktor Eksternal)

- Framework pengaturan menuntut agar perusahaan dan institusi pemerintah dapat membuktikan bahwa mereka memiliki pandangan yang jelas terhadap operasi mereka dan bahwa mereka telah mematuhi hukum yang berlaku
- Di USA, Clinger–Cohen Act (1996), juga dikenal sebagai Information Technology Management Reform Act, menuntut bahwa semua biro pemerintah harus memiliki sebuah IT architecture

Faktor Pendukung Arsitektur Enterprise (Faktor Eksternal)

- Framework lain Basel II (2004), didukung oleh gubernur bank sentral dan pimpinan otoritas pengawas perbankan negara2 Group of Ten (G10), menempatkan kebutuhan pada organisasi perbankan untuk memperhatikan financial risk management, untuk mendorong stabilitas di dunia keuangan
- US act lainnya, the Sarbanes–Oxley Act (2002), dikenal sebagai Public Company Accounting Reform and Investor Protection Act, muncul setelah skandal Enron, untuk mendorong perusahaan mengadopsi praktek good corporate governance dan untuk membuat executive perusahaan bertanggung jawab secara personal

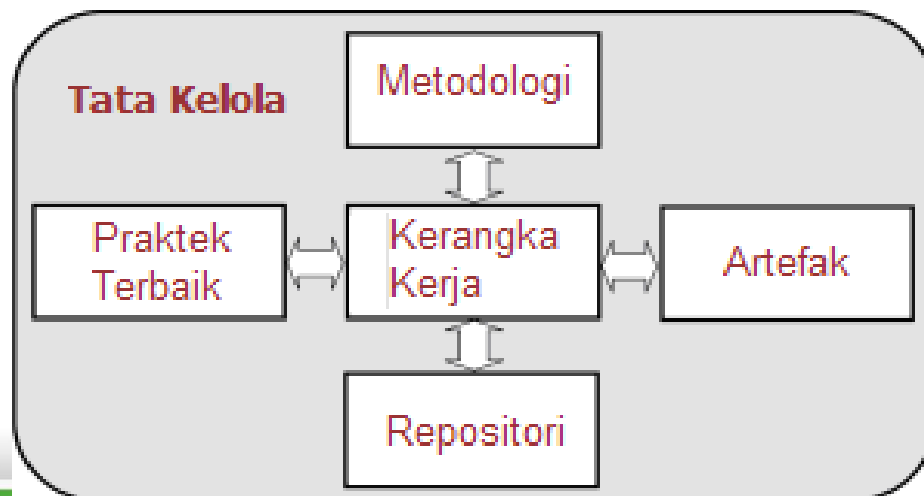
Pertemuan 2

Arsitektur Enterprise Dan Instrumen Tata Kelola

Pendahuluan

- 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)



Pendahuluan

- Elemen dasar pertama adalah “**Governance**” atau “**tata kelola**” yang mengidentifikasi perencanaan, pengambilan keputusan, dan proses pengawasan dan kelompok yang akan menentukan bagaimana *Enterprise Architecture* ini dikembangkan dan dikelola – sebagai bagian dari tata kelola secara keseluruhan suatu perusahaan.
- **EA Governance** atau **tata kelola** bertujuan untuk mendukung tata kelola yang terpadu, kelompok kebijakan manajemen yang terintegrasi dan proses yang membentuk struktur tata kelola secara keseluruhan
- *Architecture Governance* harus sesuai dengan tata kelola perusahaan secara keseluruhan maupun mekanisme dan struktur tata kelola TI yang ditetapkan.

Hubungan arsitektur enterprise dengan beberapa praktek manajemen

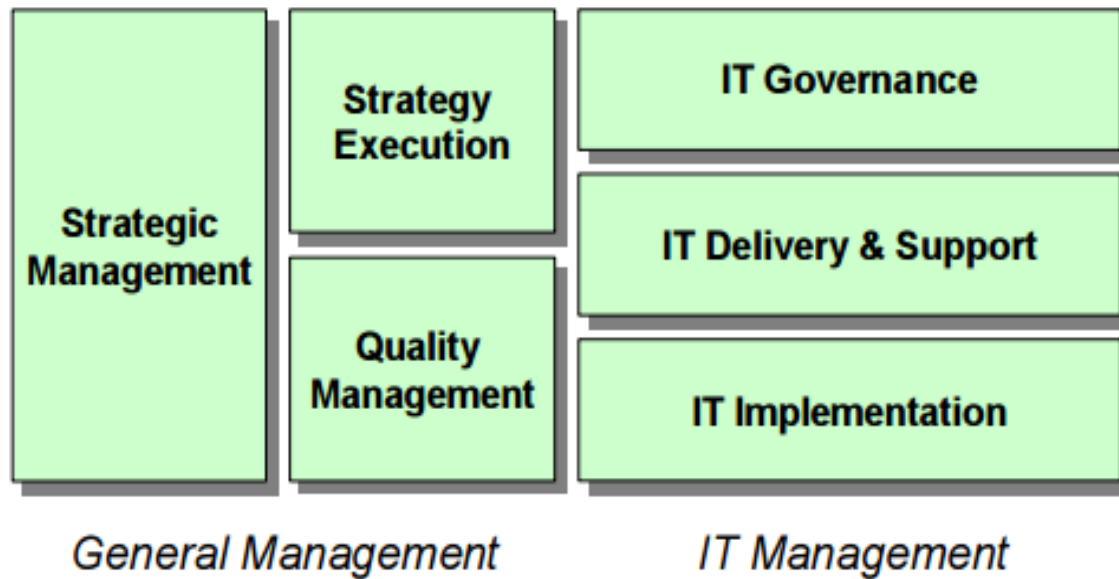


Fig. 2.1. Management areas relevant to enterprise architecture.

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 dilengkapi dengan langkah-langkah mengenai kepuasan pelanggan, proses internal dan kemampuan untuk berinovasi.

Cara kerja pendekatan BSC :

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

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

2. Strategy Execution : EFQM

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

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

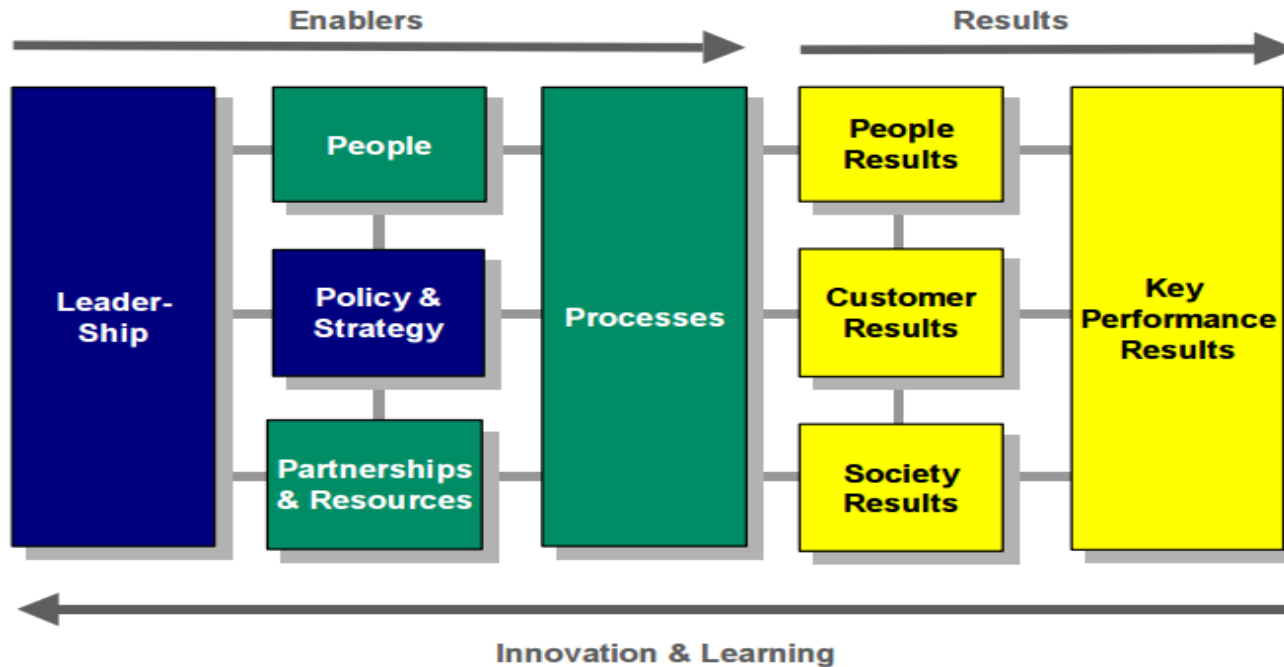


Fig. 2.2. The EFQM Excellence Model (EFQM 2003).

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

3. Quality Management : ISO 9001

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

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

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

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

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

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

4. IT Governance : COBIT

- COBIT (Control Objectives for Information and related Technology) standar untuk IT governance awalnya diterbitkan tahun 1996 oleh Audit Sistem Informasi dan Control Association.
- Sebuah kerangka IT kontrol Internasional yang menyediakan organisasi dengan 'praktek yang baik' yang membantu dalam menerapkan struktur tata kelola TI di seluruh perusahaan.
- Hal ini bertujuan untuk menjembatani kesenjangan antara risiko bisnis, kebutuhan kontrol dan masalah teknis.
- Inti dari kerangka COBIT adalah tujuan pengendalian dan manajemen pedoman untuk 34 identifikasi proses TI yang dikelompokkan ke dalam empat domain : perencanaan dan organisasi, akuisisi dan implementasi, pengiriman dan dukungan, dan pemantauan.

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

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.

5. IT Service Delivery and Support : ITIL

- ITIL (IT Infrastructure Library) (Hanna et al. 2008) adalah himpunan yang paling banyak diterima praktek terbaik dalam domain pelayanan TI.
- ITIL terdiri dari serangkaian dokumen yang memberikan bimbingan pada penyediaan baik IT layanan dan fasilitas yang dibutuhkan untuk mendukung TI
- ITIL memiliki proses berorientasi pendekatan untuk manajemen layanan. Ini memberikan kode praktek yang membantu organisasi untuk membangun manajemen kualitas layanan TI dan infrastruktur di mana 'kualitas' didefinisikan sebagai 'sangat cocok untuk kebutuhan bisnis dan kebutuhan pengguna'.

Inti dari ITIL terdiri dari dua kelompok besar proses yaitu :

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 sebagai CMM dan SW-CMM adalah model untuk menilai kematangan organisasi proses rekayasa perangkat lunak dan menyediakan organisasi dengan praktik kunci yang diperlukan untuk membantu mereka meningkatkan kematangan proses.

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

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 inovatif. Proses improvement kuantitatif bertujuan bagi organisasi yang telah ditetapkan, terus direvisi untuk mencerminkan perubahan tujuan bisnis dan digunakan sebagai kriteria dalam proses pengelolaan perbaikan.

Pertemuan 3

Metode Dan Kerangka Kerja

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.

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) adalah bisnis tambahan proses dan model informasi metodologi konstruksi. Ruang lingkup adalah sengaja dibatasi untuk operasi bisnis, menghilangkan teknologi spesifik aspek. Business Collaboration Framework (BCF) yang saat ini dalam pengembangan, akan menjadi spesialisasi dari UMM bertujuan mendefinisikan pertukaran informasi eksternal perusahaan dan bisnis yang mendasari kegiatan mereka. Lihat UN/CEFACT (2004).

- 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.

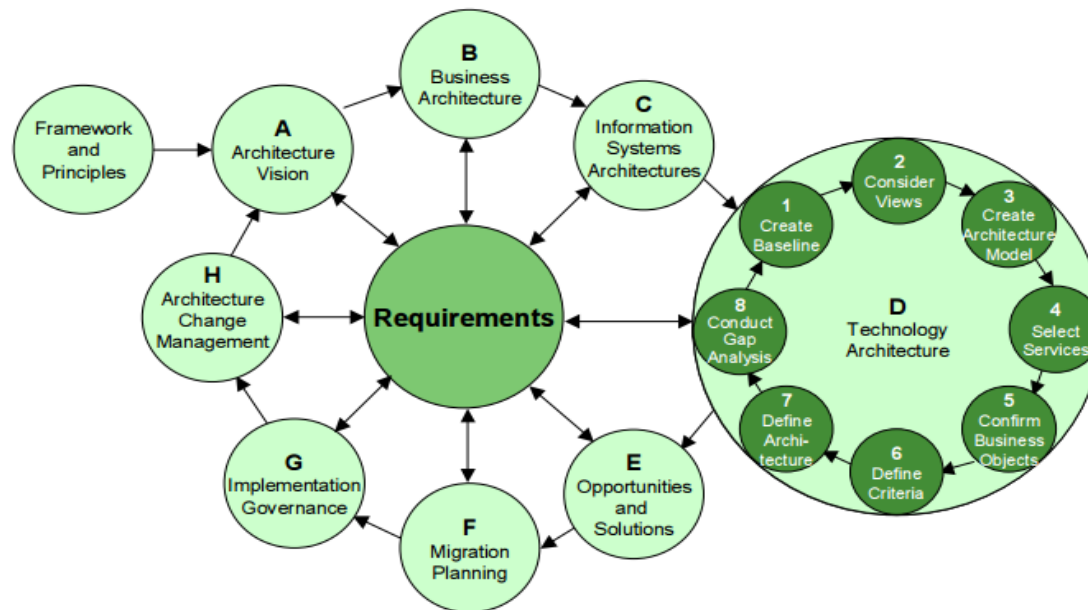


Fig. 2.3. TOGAF architecture development cycle (based on The Open Group 2002).

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

The IEEE 1471-2000/ISO/IEC 42010 Standard

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

Lanjutan...

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.

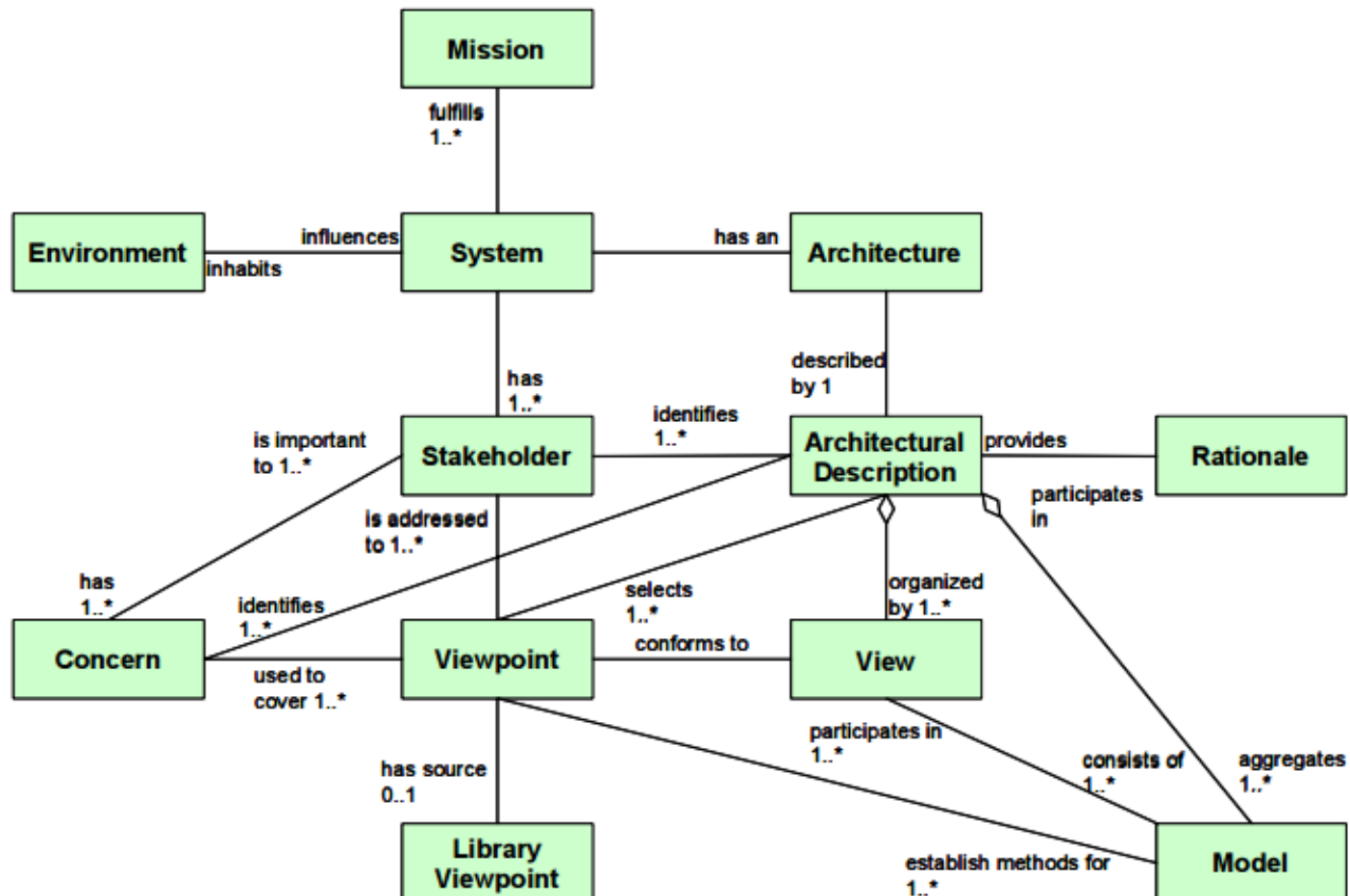


Fig. 2.4. Conceptual model of architecture description (based on IEEE Computer Society 2000).

Lanjutan...

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.

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.

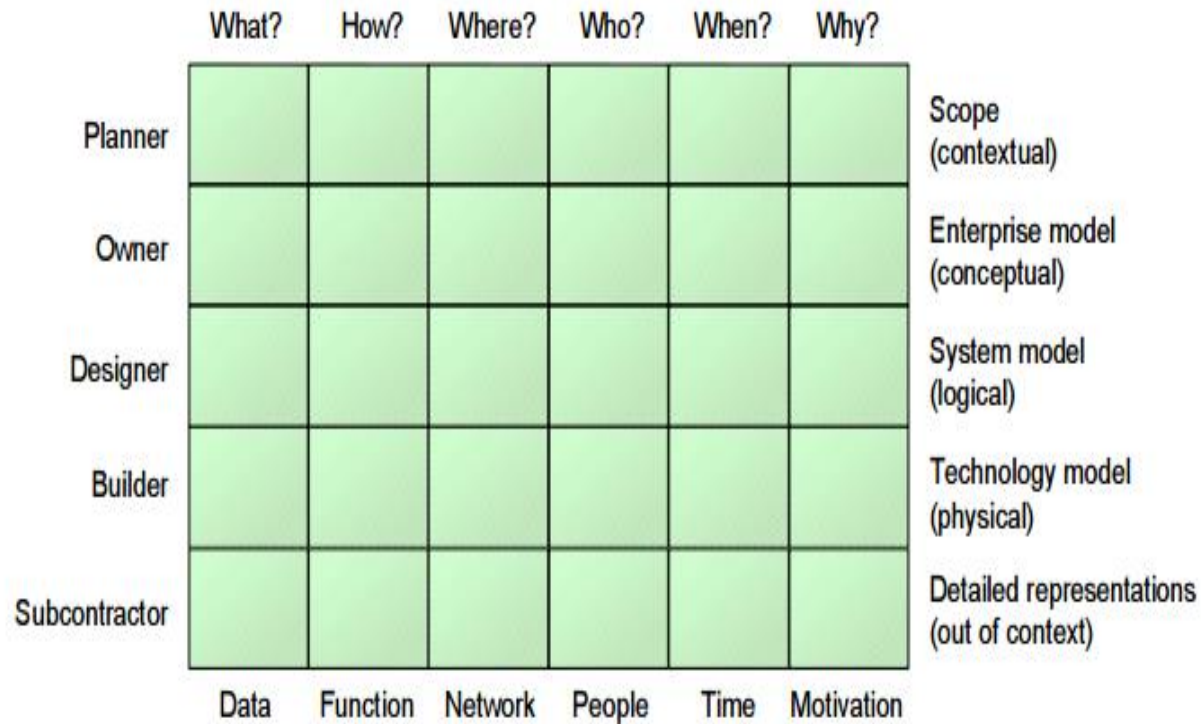


Fig. 2.5. The Zachman Framework (Zachman 1987).

Pandangan (perspektif) pada Zachman Framework :

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

The Open Group Architecture Framework (TOGAF)

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

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

- Architecture Development Method (ADM) yang menyediakan jalan bekerja untuk arsitek. ADM dianggap inti dari TOGAF dan terdiri dari pendekatan siklik bertahap untuk pengembangan keseluruhan arsitektur enterprise.
- Arsitektur Konten Framework yang menganggap suatu perusahaan secara keseluruhan arsitektur terdiri dari empat arsitektur yang saling terkait : Bisnis Arsitektur, data arsitektur, arsitektur aplikasi dan Teknologi (IT) Arsitektur.
- The Enterprise Continuum terdiri dari berbagai model referensi seperti Teknis Referensi Model, Open Group Standards Information Base (SIB) dan The Building Blocks Information Base (BBIB). Ide di balik enterprise Continuum adalah untuk menggambarkan bagaimana arsitektur dikembangkan di seluruh kontinum mulai dari arsitektur dasar, melalui sistem umum arsitektur dan arsitektur industri spesifik untuk individu suatu enterprise arsitektur sendiri.

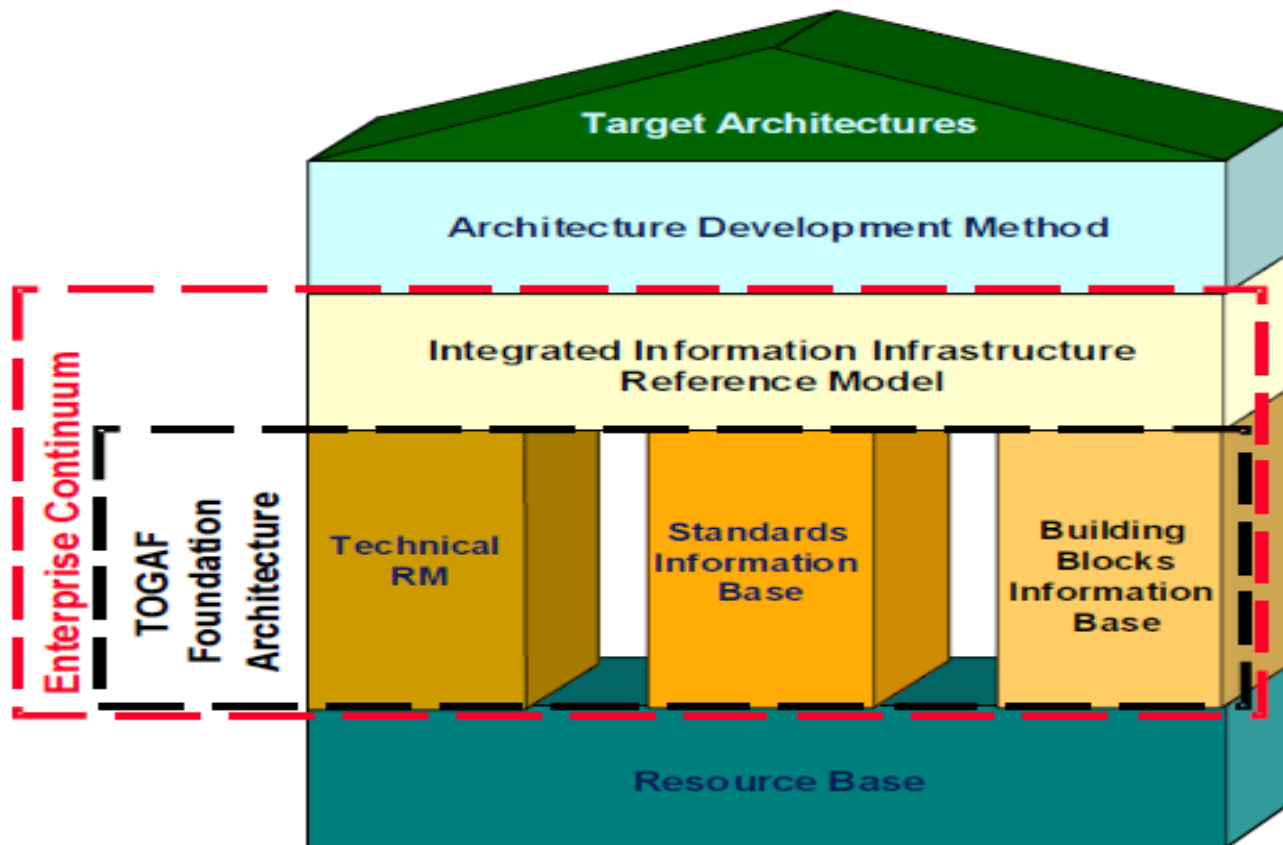


Fig. 2.6. TOGAF (based on The Open Group 2002).

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.

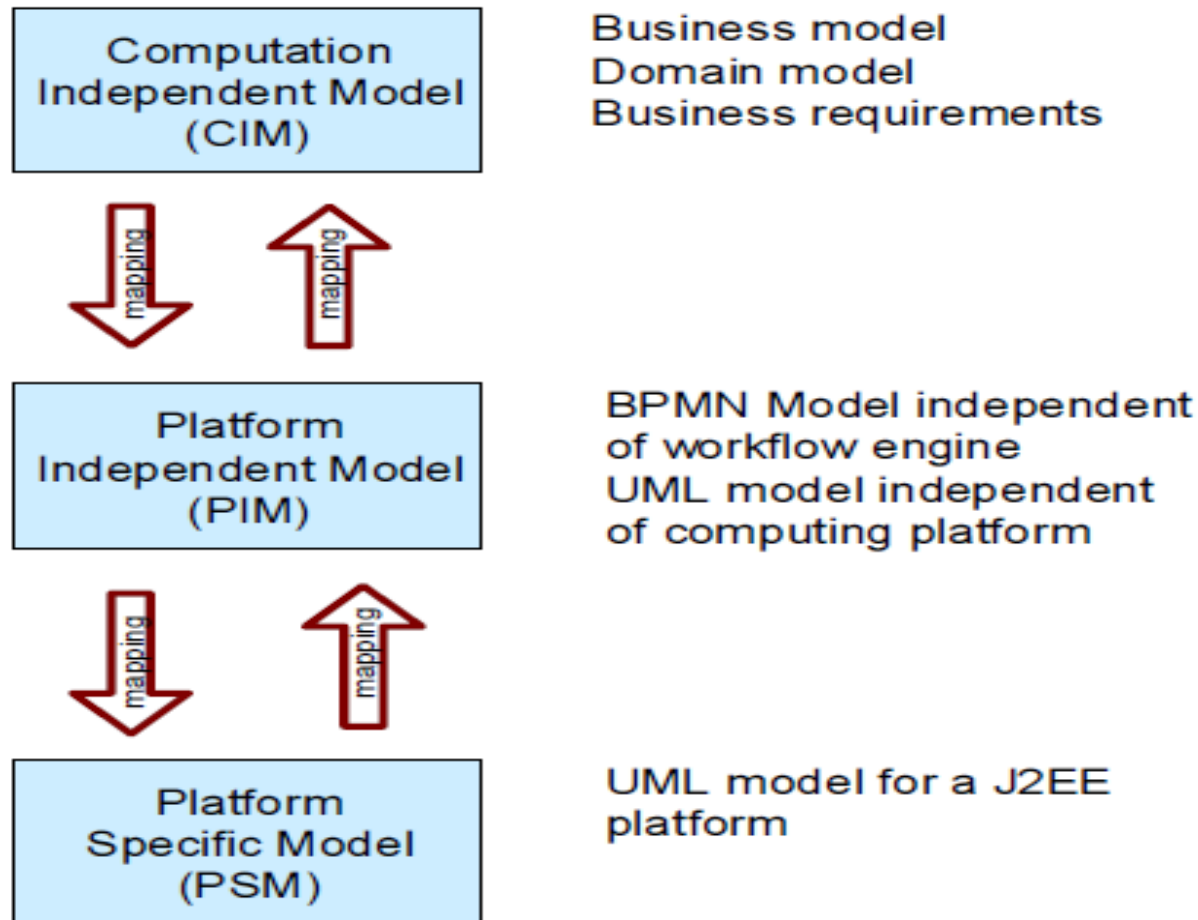


Fig. 2.7. MDA framework.

Lanjutan...

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

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.

IDEF

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

Ruang lingkup IDEF meliputi:

1. Pemodelan Fungsional, IDEF0

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

2. Pemodelan Proses, IDEF3

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

3. Pemodelan Data, IDEF1X

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

Lanjutan..

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

BPMN

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

Lanjutan...

- 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 pemodelan 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).

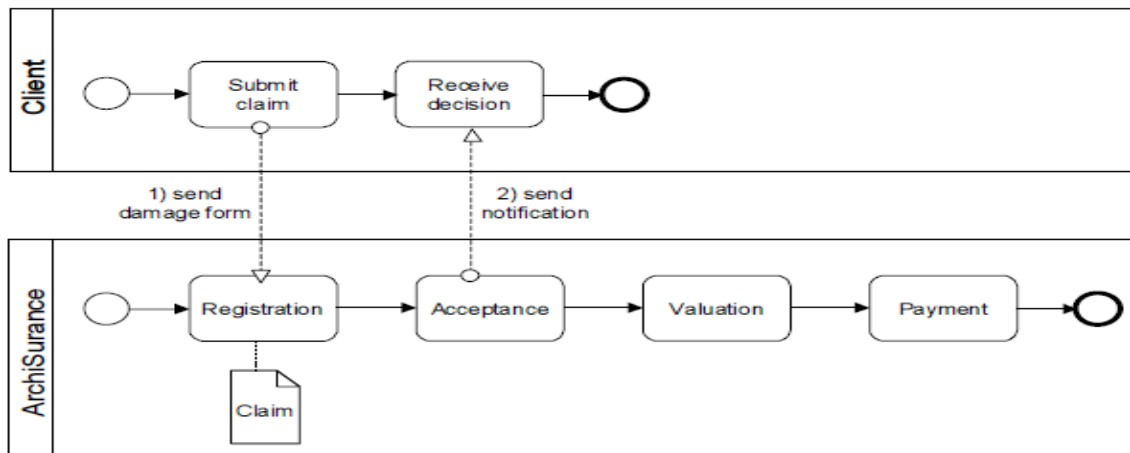


Fig. 2.9. Example model in BPMN.

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.

Lanjutan...

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.

Lanjutan...

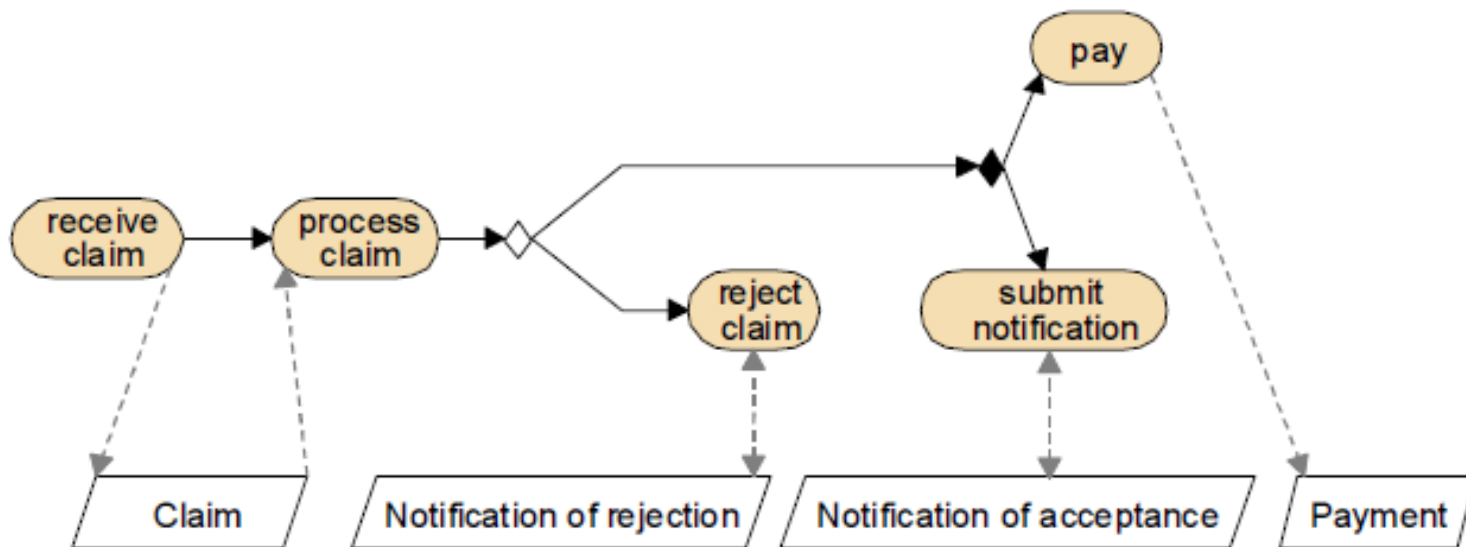


Fig. 2.10. Example of a business process model in Testbed.

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.

Lanjutan...

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.

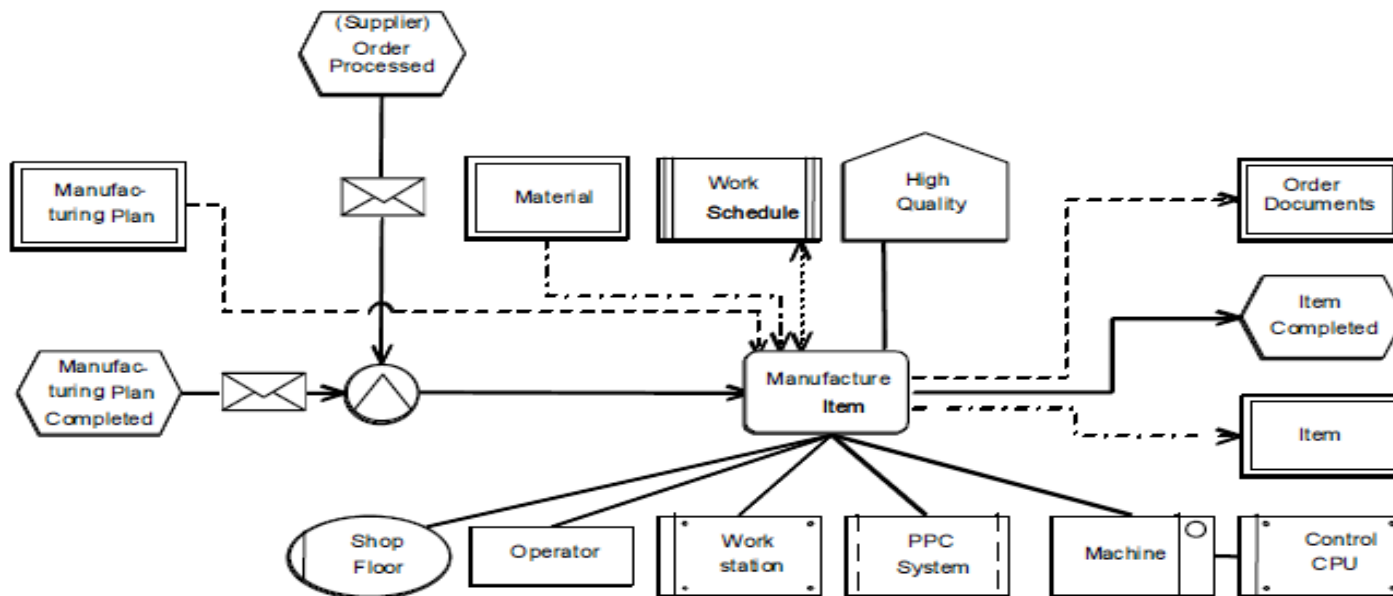


Fig. 2.11. Events, functions and control flows in ARIS.

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.

Lanjutan...

- 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 pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman seperti Java, C++, Visual Basic atau bahkan dihubungkan secara langsung ke dalam sebuah object oriented database.
- Begitu juga mengenai pendokumentasian dapat dilakukan seperti : requirements, arsitektur, design, source code, project plan, tests dan prototypes.

ADL (Architecture Description Language)

- Istilah 'Architecture Description Language' (ADL) merujuk kepada bahasa untuk menggambarkan arsitektur perangkat lunak dalam hal umum.
- Berbagai ADL ada dengan beberapa perbedaan dalam konsep yang tepat yang mereka tawarkan : beberapa berfokus pada aspek struktural arsitektur dan yang lain lebih memperhatikan untuk aspek dinamis.
- Secara umum, konsep-konsep mereka didefinisikan pada tingkat yang agak umum, meskipun mereka biasanya ditujukan untuk pemodelan tingkat aplikasi.
- Keuntungannya adalah definisi yang tepat dan pondasi formal bahasa, yang memungkinkan mereka cocok sebagai bahasa yang mendasari untuk lebih spesifik.

Lanjutan...

- 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.

Lanjutan...

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)

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 modelnya dibuat 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.

Lanjutan...

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 manajer dan konsultan bisnis; oleh karena itu, visualisasi dan pandangan khususmodel UML harus disediakan. Mengingat pentingnya UML, lainnyabahasa pemodelan kemungkinan akan menyediakan antarmuka atau pemetaan untuk itu.

Arsitektur Berorientasi Layanan

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

Lanjutan...

Meskipun penemuan dinamis dan interoperabilitas manfaat penting dari layanan web, fokus murni teknologi akan terlalu terbatas dan akan gagal menghargai nilai konsep layanan. SOA merupakan set prinsip-prinsip desain yang memungkinkan unit fungsi yang akan diberikan dan dikonsumsi sebagai layanan. Hal yang menarik adalah bahwa konsep layanan berlaku sama baiknya dengan bisnis seperti halnya untuk perangkat lunak aplikasi. Jasa menyediakan 'unit bebas' yang mewakili proposisi nilai dalam rantai nilai atau dalam proses bisnis. Konsep dasarnya sederhana ini dapat dan harus digunakan bukan hanya dalam rekayasa perangkat lunak, tetapi juga di semua tingkatan lain dari arsitektur enterprise untuk mencapai fleksibilitas dalam bisnis dan desain IT.

Lanjutan...

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

Teknologi Berorientasi Layanan

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

Lanjutan...

- 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.
- Idanya adalah bahwa pengguna atau perusahaan hanya dapat plug ke dinding untuk mendapatkan akses ke komoditi layanan komputasi dan penyimpanan.

Lanjutan...

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

Relevansi dan Manfaat untuk Arsitektur Enterprise

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

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

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

Lanjutan...

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

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

Perilaku internal dan eksternal memberikan dimensi baru dari fleksibilitas :

Fleksibilitas untuk mengganti atau jasa pengganti dalam kasus kegagalan, fleksibilitas untuk mengupdate atau mengubah layanan tanpa mempengaruhi operasi perusahaan, fleksibilitas untuk mengubah pemasok layanan, fleksibilitas untuk menggunakan kembali layanan yang ada untuk penyediaan produk atau jasa baru. Ini akan menciptakan peluang baru untuk outsourcing, rendering lebih banyak kompetisi dan lebih efisien rantai nilai.

Lanjutan...

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 eksploitasi.

Pertemuan 5

Kompleksitas Arsitektur Dan Penggambaran Arsitektur Enterprise

Pendahuluan

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

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

Mengatasi Kompleksitas Arsitektur

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

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

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

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.

Komunikasi

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

Realisasi

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

Change

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

KOMPOSISI

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

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

Lanjutan...

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.

Lanjutan..

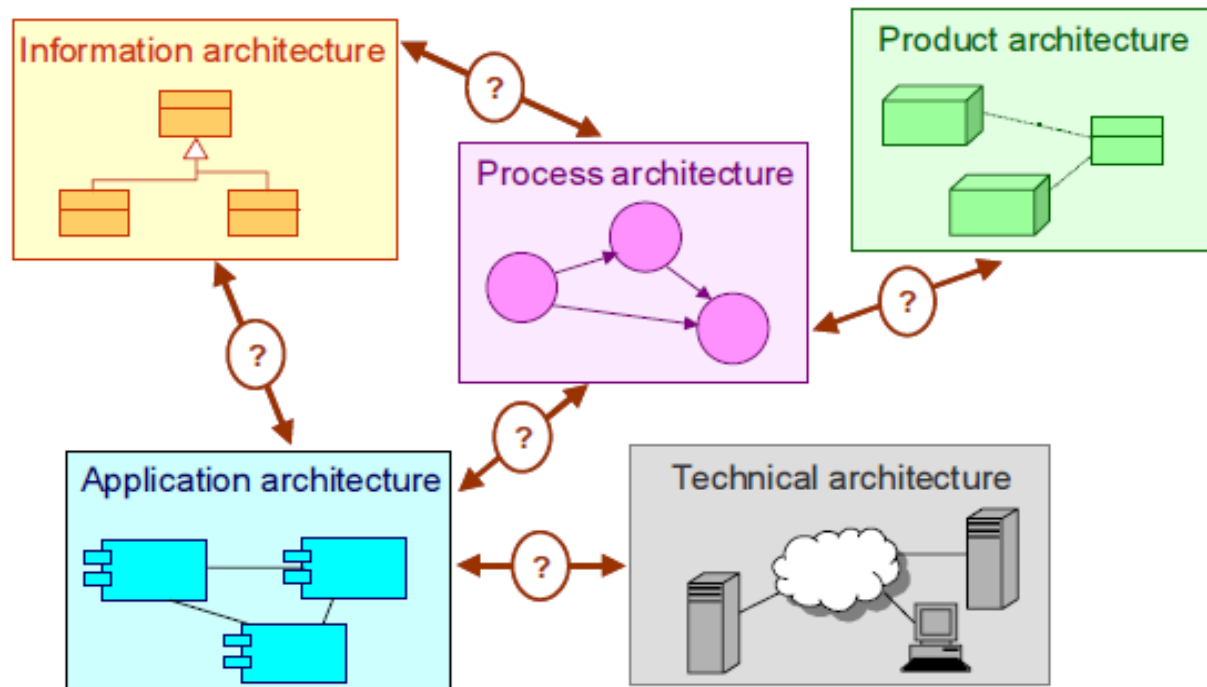


Fig. 3.1. Heterogeneous architectural domains.

Integrasi Domain Arsitektur

Sebuah Gambaran :



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



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

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

Lanjutan..

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

Penggambaran Arsitektur Enterprise

Untuk mengatasi kompleksitas dalam arsitektur enterprise, sebenarnya hanya berfokus pada relasi dan interaksi diantara domain arsitektur. Dengan definisi dan batasan yang tepat maka akan membantu dalam membangun wawasan kompleksitas dan menghindari konflik serta ketidak konsistensian antara domain, untuk itu digunakanlah model.

Dalam kontek disini, model biasanya dipresentasikan dengan menggunakan bahasa grafis atau tekstual yang diformalkan. Terdapat bahasa terkait selanjutnya ialah mengenai :

- ✓ Observing the universe
- ✓ Concern
- ✓ Memahami Domain
- ✓ Tampilan dan Sudut Pandang
- ✓ Cara kerja
- ✓ Model Arsitektur Enterprise

Observing The Universe

Jika diibaratkan stakeholder adalah penonton dari sebuah organisasi, teknikal atau sistem lain yang membentuk perusahaan atas alam semesta yang diobservasi. Sedangkan konsepsi dari alam semesta tersebut adalah sebuah arsitektur 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 disamping yang berdasarkan pada FRISCO tetrahedron.

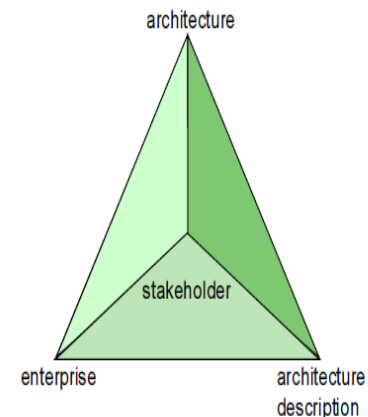


Fig. 3.2. Relationship between enterprise, stakeholder, architecture, and architecture description.

Concern

Dalam menyusun bagian enterprise, stakeholder biasanya akan terpengaruhi oleh beberapa fokus mereka masing-masing atau kepentingan masing-masing. Tetapi kepentingan juga bukan merupakan satu-satunya faktor yang mempengaruhi konsepsi stakeholder, lingkungan, kultur, pendidikan serta background masing-masing turut mengambil peran.

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

Memahami Domain

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

Tampilan dan Sudut Pandang

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

Cara Kerja

Membuat dan menggunakan model arsitektur biasanya melibatkan beberapa 'cara kerja' terkait (Wijers dan Heijes 1990) :



Model Arsitektur Enterprise

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

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

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

Pertemuan 6

Gambar, Model Dan Semantik

Pendahuluan



Dalam banyak disiplin ilmu teknik, pemodelan sistem yaitu membangun model matematis untuk menggambarkan dan menjelaskan. Namun, dalam arsitektur lebih ada kecenderungan untuk menggantikan model matematika dengan visualisasi ad hoc.

Disini kita perlu mengacu pada konsep yang terlibat, tetapi juga mengacu pada bentuk untuk disajikan.



'Konten' dan 'visualisasi' harus ditafsirkan di sini dengan cara yang longgar. Misalnya, perangkat untuk input juga bisa termasuk pemvisualan seperti menu atau tombol, lalu kontennya mencakup tindakan yang mengubah model seperti konsep menambahkan atau menghapus.

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

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

Model Simbolik Dan Semantik

❖ 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.

Lanjutan....

❖ 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.

Lanjutan...

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

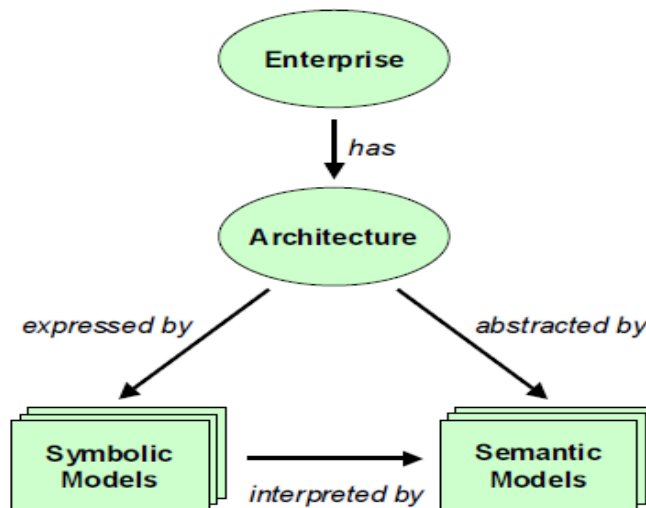


Fig. 3.3. The enterprise, its architecture, symbolic and semantic models.

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.

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.

Lanjutan..

Ketiga, dalam arsitektur seringkali dibuat perbedaan antara arsitektur semantik dan semantik formal bahasa pemodelan. Seperti yang dijelaskan, perusahaan yang dipertimbangkan dalam hal konsep arsitektur, yang ada di benak, misalnya, arsitek perusahaan. Konsep-konsep ini dapat direpresentasikan dalam model, yang diekspresikan dalam bahasa pemodelan.

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

Model Simbolik

Model simbolik adalah formalisasi satu atau lebih aspek arsitektur dari sistem konkret. Ini terdiri dari bagian-bagian arsitektur yang dapat dimodelkan secara matematis, sebagai lawan dari aspek yang lebih pragmatis dari arsitektur yang berkaitan dengan gagasan karakteristik seperti alasan, tujuan, dan rencana. Model simbolik diekspresikan menggunakan bahasa deskripsi, representasi dari model yang sering bingung dengan interpretasinya.

Lanjutan...

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

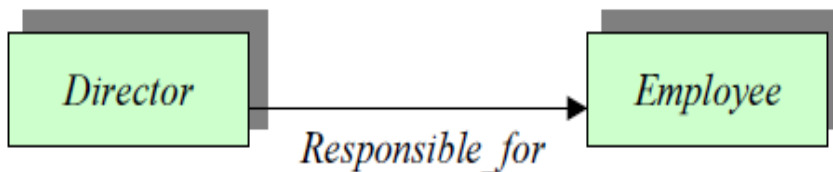


Fig. 3.4. Symbolic model of the director–employee relationship.

Model Semantik

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

Sebagai contoh, semantik formal tanda tangan disediakan oleh koleksi set (satu untuk setiap jenis tanda tangan), dan satu set hubungan dan fungsi di antara mereka, satu untuk setiap simbol relasi dan simbol fungsi di tanda tangan. Informasi hierarkis antara macam ditangkap oleh inklusi subset biasa, sedangkan informasi kontainmen dilambangkan dengan elemen-relasi yang biasa.

UML vs ArchiMate

Pendekatan ArchiMate dapat dikontraskan dengan pendekatan asli di UML. Dalam pendekatan ini, semantik secara eksplisit ditinggalkan keluar dari program. Orang yang menggunakan model bisa berkembang semantik untuk mereka, tetapi semantik umum tidak disediakan. Pendekatan ini juga berasal dari asal usul UML sebagai kombinasi dari tiga notasi yang ada yang tidak memiliki semantik formal. Karenanya, fokus dari UML dulu dan sekarang pada notasi, mis., Sintaks, dan bukan pada semantik. Meskipun beberapa diagram versi UML yang lebih baru memiliki formal semantik (lihat, mis., semantik kegiatan yang mirip dengan Petrinet diagram dalam UML 2.0), tidak ada semantik keseluruhan untuk seluruh bahasa.

Lanjutan..

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.

Pertemuan 7

REVIEW MATERI (QUIZ)

Pengantar pada Framework TOGAF

Sejarah TOGAF

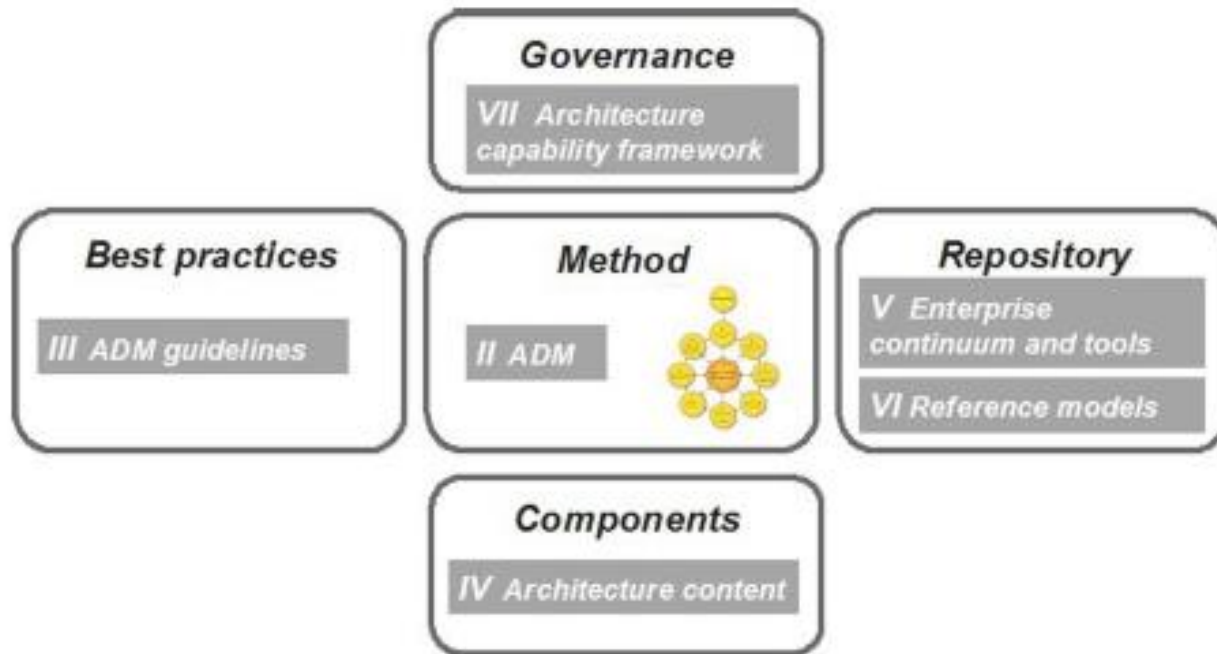
- **TOGAF** (The Open Group Architecture Framework) muncul dengan cepat dan merupakan kerangka kerja serta metode yang dapat diterima secara luas dalam pengembangan arsitektur perusahaan
- TOGAF dimulai awal 1990-an sebagai metodologi untuk pengembangan arsitektur teknis, dan telah dikembangkan oleh 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 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 merilis 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 Models
7. Architecture Capability Framework

The TOGAF Document?



Gambar di atas menyajikan pengelompokan berbagai macam bagian pada dokumen TOGAF yaitu : Method, Best practice, Component, Repository dan Governance

The TOGAF Document?

- ADM (“Architecture Development Method”) pada bagian II adalah is the main entry point to the TOGAF reference document, with its crop circle diagram (or TOGAF wheel), which describes the different phases of the method.
- Part III membahas mengenai guidelines and best practices linked to the ADM, from security 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.

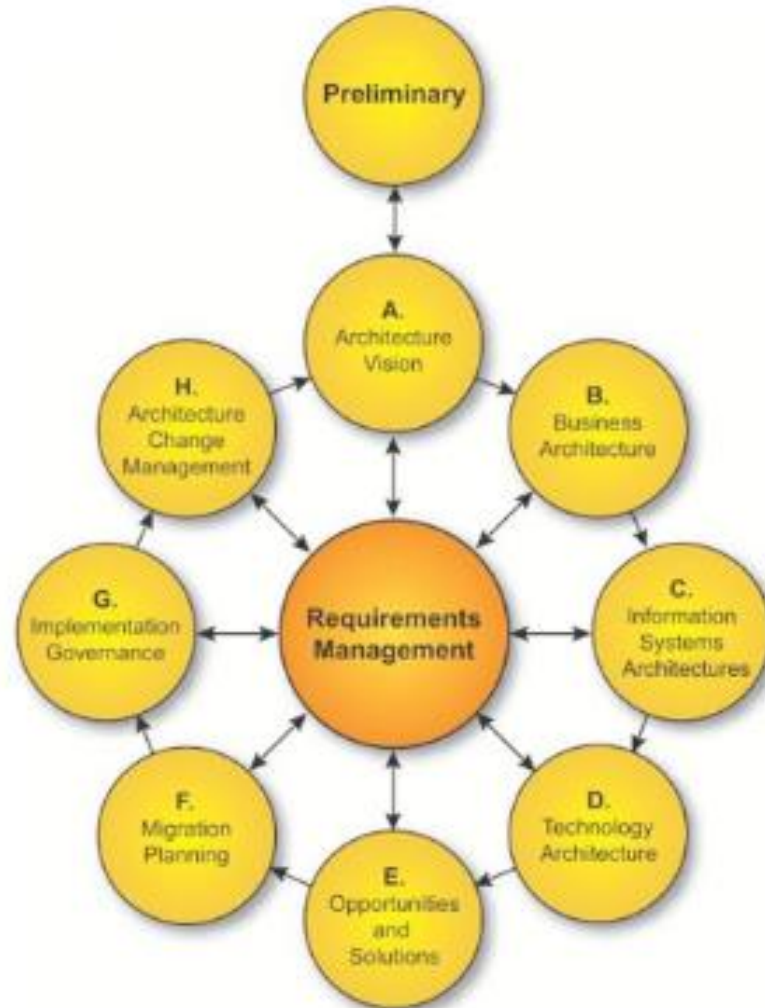
TOGAF : KEY POINT

- 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 mendefinisikan 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.
- The work carried out is supervised by the architecture board, in partnership with all the business and IS stakeholders. As you can see, the proposed path is a cycle, which finishes by looping back on itself. Admittedly, this is merely a schematic representation that only partially represents reality. However, it does successfully express the continuous nature of enterprise architecture work, which responds to the constant demands of businesses.

TOGAF : KEY POINT

- The central position occupied by requirements management in the diagram is testament to the pivotal role it plays within the ADM cycle. Strictly speaking, requirements management is more a permanent activity than a phase. However, the term “phase” is used to designate it in order to harmonize vocabulary. The same is true for the preliminary phase, which groups cross-organizational activities such as the definition of context, methods and tools for enterprise architecture, and the start of an ADM cycle.
- Fundamentally, the aim of an ADM cycle is to successfully complete a transformation project, whose aim is to enable the enterprise to respond to a set of business goals.

TOGAF : KEY POINT



The "TOGAF crop circle diagram" with the ADM phases

Arsitektur Enterprise Menggunakan TOGAF

Arsitektur Enterprise Menggunakan TOGAF

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.”

The first definition considers the term “architecture” as a synonym of “system description.” For the second, “architecture” designates the structure and principles of the system, independent of its description.

Arsitektur Enterprise Menggunakan TOGAF

- This double definition may seem surprising, but it does reflect a very real situation. Software systems are, by nature, opaque, so much so that their structure is only visible through representations.
- Factories, ships, or engines all have a physical structure, which is more or less visible. However, it is impossible to “lift the hood” of an IT system, whose architecture only exists through its representation.
- This is also the case for business elements such as processes, organization, or strategy, which can only be communicated through descriptions or models. The proliferation of schemas, diagrams, and tables within enterprises is testament to this reality.
- In this context, communication on architecture plays a determining role. Just like the blueprints of a building, communication is a vital tool for those working on the different tasks involved: development, evaluation, exchange, and construction.

Arsitektur Enterprise Menggunakan TOGAF

- 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..

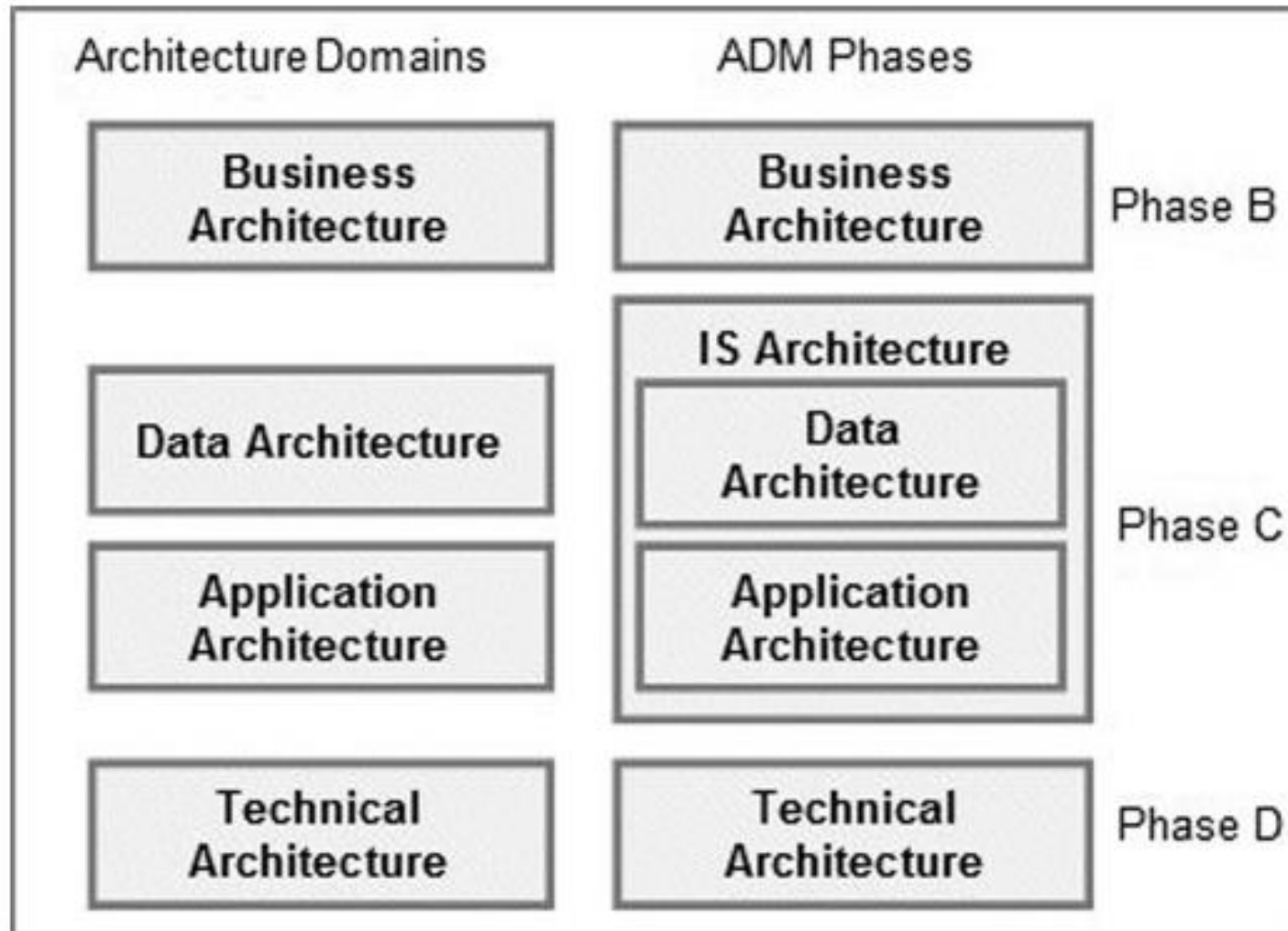
Arsitektur Enterprise Menggunakan TOGAF

- 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.

Arsitektur Enterprise Menggunakan TOGAF



Arsitektur Enterprise Menggunakan TOGAF

- 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.

Arsitektur Enterprise Menggunakan TOGAF

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

Arsitektur Enterprise Menggunakan TOGAF

Stakeholders and the human factor

- We know that the organizational aspect is one of the most delicate points in this type of operation. Like any enterprise process, architecture transformation involves a combination of activities involving different participants, each one of them a “stakeholder” in the operation he or she undertakes. TOGAF deals with this question through the following themes:
 1. Stakeholder management
 2. Transformation Readiness Assessment
 3. Efficiency of communication through the concept of viewpoints

Arsitektur Enterprise Menggunakan TOGAF

1. Managing Stakeholder

First, it is essential to clearly define each stakeholder as early as possible during the ADM cycle. This identification mainly uses a pragmatic approach in order to avoid simply reusing existing organizational structures, which only partially represent the reality of the activities and responsibilities that will be put in motion. Leaving a key participant by the wayside would significantly affect the quality of the results.

Arsitektur Enterprise Menggunakan TOGAF

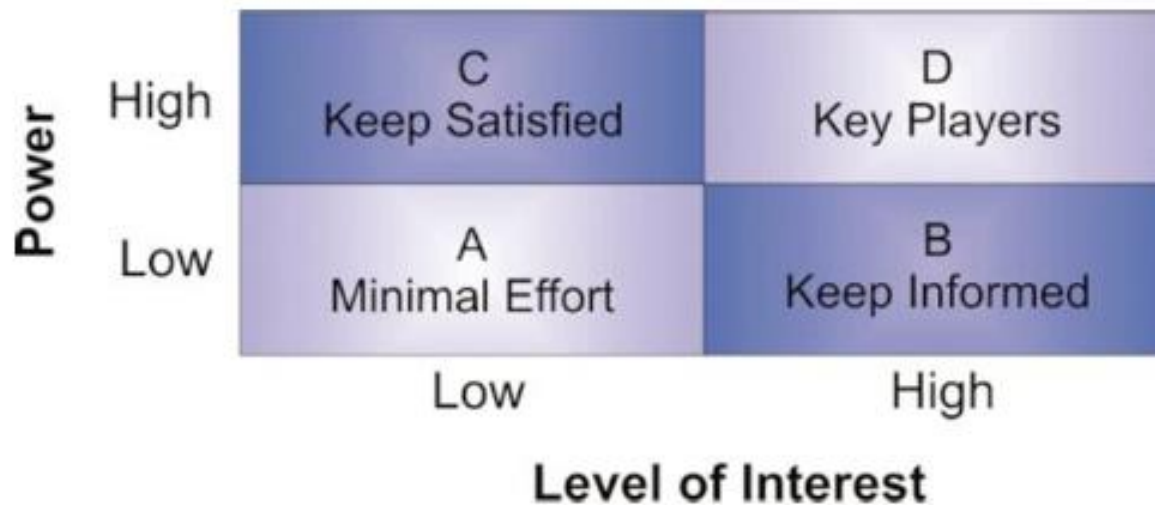
Consequently, in order to determine with whom and in what form work will be carried out, a series of key questions must be answered on the subject:

- 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?

Arsitektur Enterprise Menggunakan TOGAF

From these questions, TOGAF recommends that the position of each stakeholder be clarified, notably his or her degree of involvement. The Figure presents these different degrees. Each stakeholder will be positioned using these degrees of involvement, which determine the relationships to develop and the level of involvement in architecture project steering committees.

Naturally, key players play a determining role and must be on the front line in all areas of decision making.



Arsitektur Enterprise Menggunakan TOGAF

This qualification is cross-referenced with the role played in the context of the current project:

- The executive management, who defines strategic goals
- The client, who is responsible for the allocated budget, with regard to the expected goals
- The user, who directly interacts with the system in the course of his or her activities
- The provider, who delivers the component elements of the architecture, notably its software components
- The sponsor, who drives and guides the work
- The enterprise architect, who turns business goals into reality within the structure of the system

Arsitektur Enterprise Menggunakan TOGAF

2. Transformation Readiness Assessment

- Is the organization ready for the envisaged change? This question may seem incongruous, but how many projects have finished unsuccessfully because this dimension was not taken into account? In just a few years, “change management” has become a discipline in its own right, producing a large number of articles and seminars.
- TOGAF is entirely dedicated to this question, which is widely discussed in the description of the ADM phases. The identification of change resistance risks and the definition of actions to take to limit these risks are essential tasks that must be carried out before embarking upon a transformation project. This is particularly important for an operation covering a broad scope and resulting in significant restructuring.

Arsitektur Enterprise Menggunakan TOGAF

- While it is not possible to provide turnkey solutions on such a subject, it is possible to use certain techniques that will help reduce this type of risk:
 1. A clear presentation of the impacts of changes made, notably on an organizational level
 2. A concrete view of the expected business benefits, in the form of “business cases”
 3. An objective assessment of the enterprise’s IT, business, and financial aptitudes, with no over estimation of its real capacities
 4. An executive management team recognized as being able to defend the project in the long term
 5. High-quality communication, which aims to promote a common understanding of the stakes and the solutions to implement

Arsitektur Enterprise Menggunakan TOGAF

3. Views and viewpoints

- If a message is to be successfully understood, the most important aspect to consider is that its content and form must be tailored to the intended recipient.
- For this, TOGAF uses the concept of viewpoints. A viewpoint designates the most appropriate perspective for a given participant, and is materialized by a certain number of views of the architecture, in the form of diagrams, documents, or other elements. For example, executive management will be more interested in a high-level description, while communication with operational staff will require much more detailed representations.

Arsitektur Enterprise Menggunakan TOGAF

3. Views and viewpoints

- This is a critical point, one that will condition the quality of communication and that will be encountered during each phase of the ADM cycle. Consequently, it is imperative that views and viewpoints be defined for each stakeholder before beginning work on the four architecture domains (business, data, application, and technology).

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

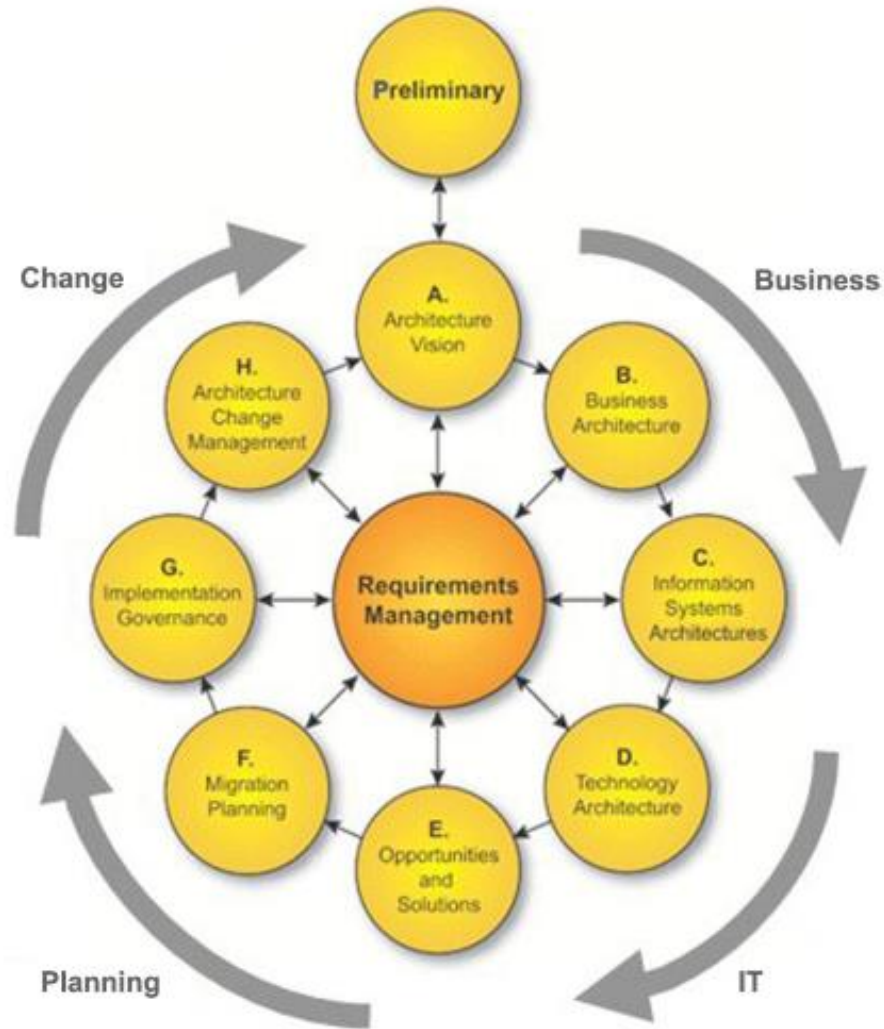
Metode ADM

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

Metode ADM

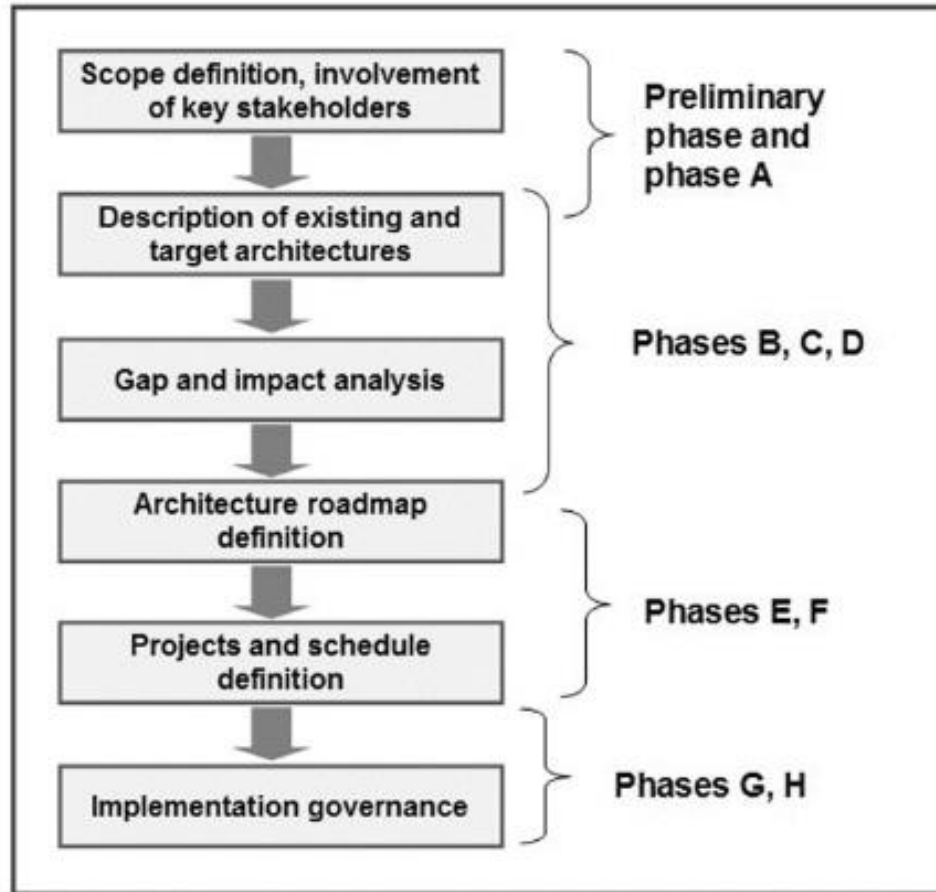


Gambar 1. Architecture development method (ADM)—TOGAF9

Metode ADM

- 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.

Metode ADM



Gambar 2. Typical path of an ADM cycle.

- Figure presents an overview of the progression of an ADM cycle, from the preliminary phase right through to phase H. This typical path is guided by one major goal: the need to obtain the expected result by mastering each step of the process. This goal requires rigorous preparation, a description of the target with regard to what already exists for all facets (business, information system, and technology), precise evaluation of the gaps and risks determining the choice of trajectory, and finally evaluation of the results and careful management of any adjustments made.
- TOGAF describes each step of each phase in detail. This does not mean that the realization of each of these steps is systematic. Some results are already available, either because they are produced by other entities or because they are linked to more general activities. For example, if architecture principles are recorded as results from phase A, they can simply be checked where they already exist. The main thing here is that the presence and conformity of each result must be checked, just like a checklist associated with each phase.

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.

Fase-Fase Pada TOGAF ADM

2. Phase A (vision)

Phase A is the first phase of the ADM cycle, triggered by the validation of the “Request for Architecture Work” document. Phase A has two main goals:

- First, phase A further develops and enriches elements resulting from the preliminary phase, such as architecture principles, key indicators, and the organization or planning of elaboration work.
- Second, phase A prepares subsequent phases by providing a general representation of the baseline and target architectures. At this stage, these remain high-level representations, whose goal is to highlight structuring points and typical solutions.

Communication plays a key role during this phase. All stakeholders must have the same understanding, in order to obtain consensus on orientations and expected results. Other points are also dealt with, such as the identification of fundamental requirements, their links to strategic goals, or risk management. The “Architecture vision” document constitutes the main output of this phase.

To sum up, at the end of phase A we have a common vision of:

- 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

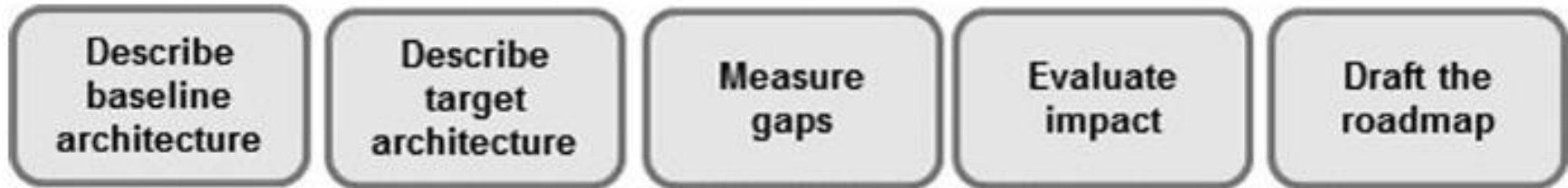
In other words, we know where we’re going, how we’re getting there, and with whom. Note that at this stage the perspective is horizontal, and covers all architecture domains (business, information system, and technology), unlike the following three phases, which operate vertically, focusing on one particular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

Fase-Fase Pada TOGAF ADM

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.

Fase-Fase Pada TOGAF ADM



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.

Fase-Fase Pada TOGAF ADM

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.

Fase-Fase Pada TOGAF ADM

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 discussed 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.

Fase-Fase Pada TOGAF ADM

Phase D (technology architecture)

- Unsurprisingly, the role of phase D is to establish the technological and physical correspondence of the elements developed during the previous phases. In particular, technology architecture defines the platforms and execution environments on which the applications run and the data sources are hosted for use.
- 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.

Fase-Fase Pada TOGAF ADM

Phase D (technology architecture)

- 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.

Fase-Fase Pada TOGAF ADM

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

- At this point of the ADM cycle, the operational realization of architecture transformation truly begins: projects are set up, schedules defined, resources identified, and operational monitoring put in place. The previous phases have provided the target, an overall roadmap, and now their concrete implementation has to be defined.
- Phases E and F look at the scheduling and organization of the implementation of new architecture. Emphasis is placed on building the migration path, which must bring true business benefit to each step.
- During phase E, the results of the elaboration phases (B, C, and D) are consolidated: architectures, requirements, and gaps. This consolidation constitutes the raw material used to define transition architectures, while bearing in mind the enterprise's capability for change (for example, new applications to develop and evolutions of existing applications, according to the coverage of the business functions). Technical and organization feasibility, compromises between requirements and costs, and integration constraints are also studied.
- Phase F precisely establishes migration scheduling, as well as the constitution of implementation projects and their organization, goals, and costs.

Fase-Fase Pada TOGAF ADM

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

- Phase F precisely establishes migration scheduling, as well as the constitution of implementation projects and their organization, goals, and costs.

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

- Phase G establishes the definitive version of architecture contracts with implementation projects, including recommendations from the architecture board. These signed contracts constitute the basis for conformity reviews of implementation projects.
- Phase H handles the management of the deployed architecture: change management, including the evaluation of change requests that impact the architecture. It should be noted that certain evolution requests can lead to new ADM cycles.

UML UNTUK PERMODELAN TOGAF

UML DAN BPMN UNTUK PERMODELAN TOGAF

- TOGAF merujuk beberapa kali ke bahasa pemodelan UML dan BPMN untuk mendukung pemodelan arsitektur enterprise. Namun, beberapa penjelasan masih diperlukan tentang bagaimana standar ini harus digunakan untuk menerapkan TOGAF, dan pada bagian model mana yang akan ditampilkan untuk setiap viewpoint.
- UML dan BPMN akan digunakan untuk mewakili objek arsitektur dasar, yang didefinisikan dalam metamodel TOGAF. Ini berarti kita harus memutuskan elemen UML atau BPMN mana yang akan digunakan untuk mewakili setiap objek TOGAF. Dengan cara ini, proses BPMN secara alami akan mewakili proses TOGAF, aktor UML akan mewakili aktor TOGAF tetapi juga peran TOGAF, dan kelas UML akan mewakili entitas bisnis TOGAF juga produk TOGAF.

UML DAN BPN UNTUK PERMODELAN TOGAF

- Model harus disesuaikan untuk memfasilitasi komunikasi antara participants. UML dan BPMN, keduanya standar yang sangat kaya, harus difilter menurut sudut pandang sehingga dapat menyajikan konsep yang berguna dan juga harus disesuaikan agar sesuai dengan terminologi TOGAF (seperti peran, entitas bisnis, dll.). Kadang-kadang mereka harus diperluas untuk mendukung konsep yang tidak ada dalam bahasa ini (misalnya, fungsi, unit organisasi, tujuan).

The UML Standard

- UML was standardized in 1997, and a major new version was published in 2005. UML groups together a large number of modeling techniques that were previously scattered among different domains (entity relationship, object model, state diagram, sequence diagram, process modeling, etc.). It is widely accepted and used in the modeling of software systems.
- UML enables data to be modeled through class diagrams. Behavior is modeled through object modeling (object behaviors, operations, etc.) and the support of sequence diagrams, state diagrams, and activity diagrams. Systems and architectures are also modeled using the concept of components and component assembly techniques.

The UML Standard

- UML is used for a wide variety of domains and targets and must therefore be adapted to correspond to the concerns and concepts specific to each target. For example, the concept of persistence must be introduced when modeling database schemas, while concepts specific to programming languages must be added for targets. Similarly, the concept of service is required when modeling SOA architectures, while the concept of the system is central to large system modeling.
- A UML profile is a set of extensions brought to UML to adapt it to a particular target. For example, SoaML or SysML (Figure 1) are UML profiles standardized by the OMG.

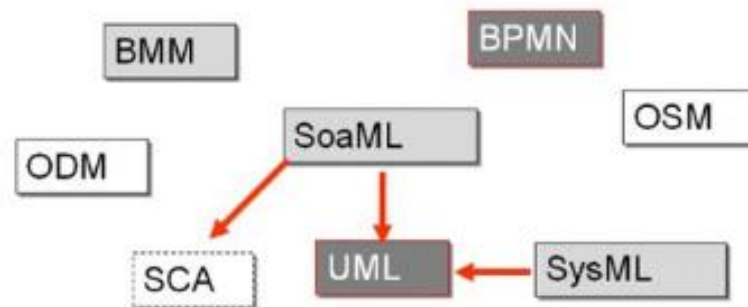


Figure 1

System Analysis and Design with UML

1. System Analysis

1. Business Process Identification
 - **Use Case Diagram**
2. Business Process Modeling
 - **Activity Diagram** or Business Process Modeling Notation (BPMN)
3. Business Process Realization
 - **Sequence Diagram**

2. System Design

1. Program Design
 1. **Class Diagram**
 2. **Package Diagram**
 3. **Deployment Diagram**
2. **User Interface Design**
3. **Entity-Relationship Model**

STUDI KASUS: ATM SYSTEM

ATM System



Main menu

- 1 - View my balance
- 2 - Withdraw cash
- 3 - Deposit funds
- 4 - Exit

Enter a choice:

1 2 3

4 5 6

7 8 9

0 Enter

Take cash here

Insert deposit envelope here

ATM System

Layar

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Masukkan PIN:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Menu Utama

1. Melihat Saldo
2. Mentransfer Uang
3. Mengambil Uang
4. Logout

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Menu Melihat Saldo

1. Saldo anda adalah

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Menu Mentransfer Uang

1. No Account Penerima:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Menu Mentransfer Uang

1. Jumlah uang yang dikirim:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Menu Mentransfer Uang

1. Uang berhasil terkirim

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Menu Mengambil Uang

1. Jumlah uang yang diambil:

Kotak Uang

Kotak Kartu

Kotak Kuitansi

Menu Mengambil Uang

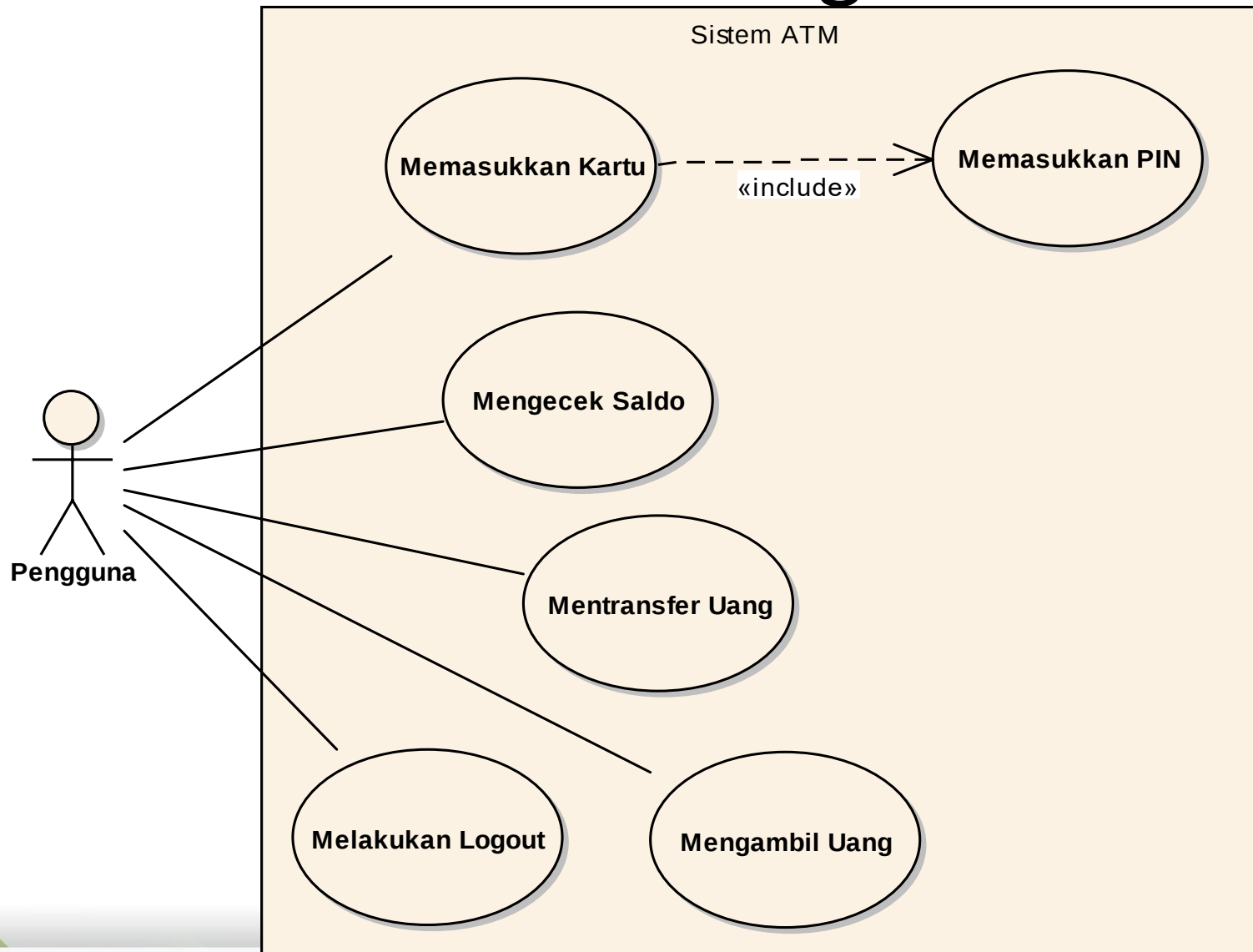
Uang berhasil diambil

Kotak Uang

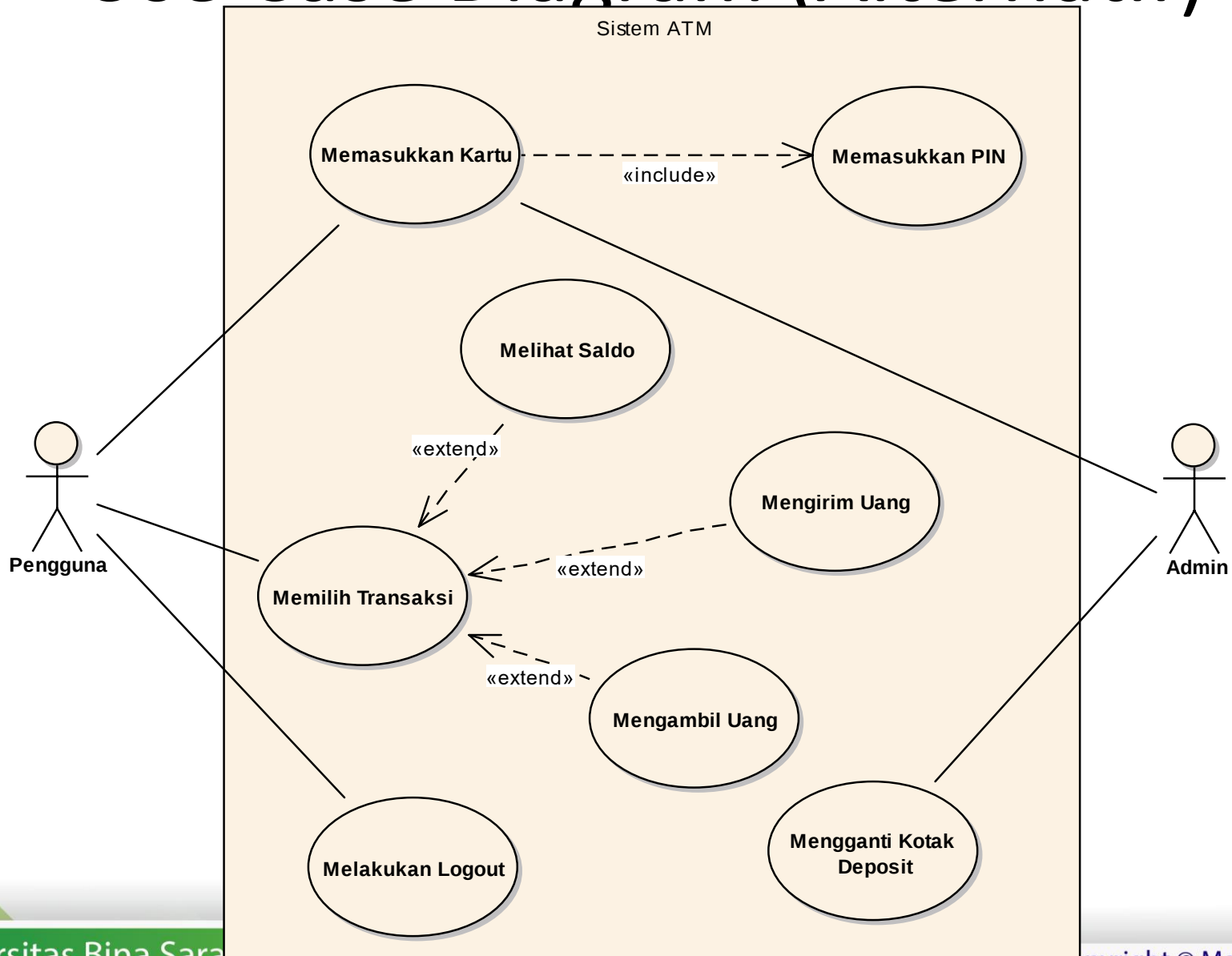
Kotak Kartu

Kotak Kuitansi

Use Case Diagram



Use Case Diagram (Alternatif)



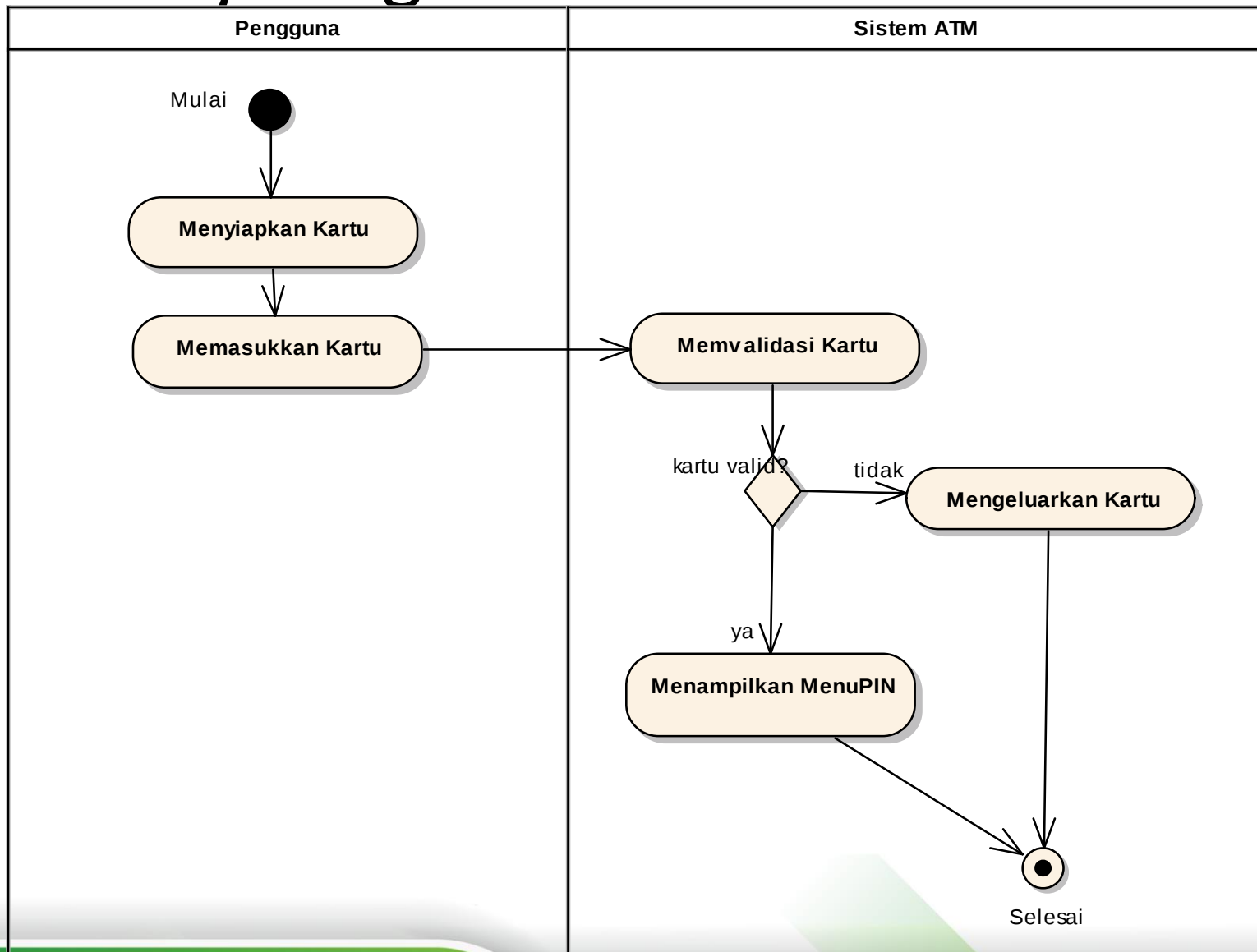
BUSINESS PROCESS MODELING: ACTIVITY DIAGRAM

BPM With Activity Diagrams

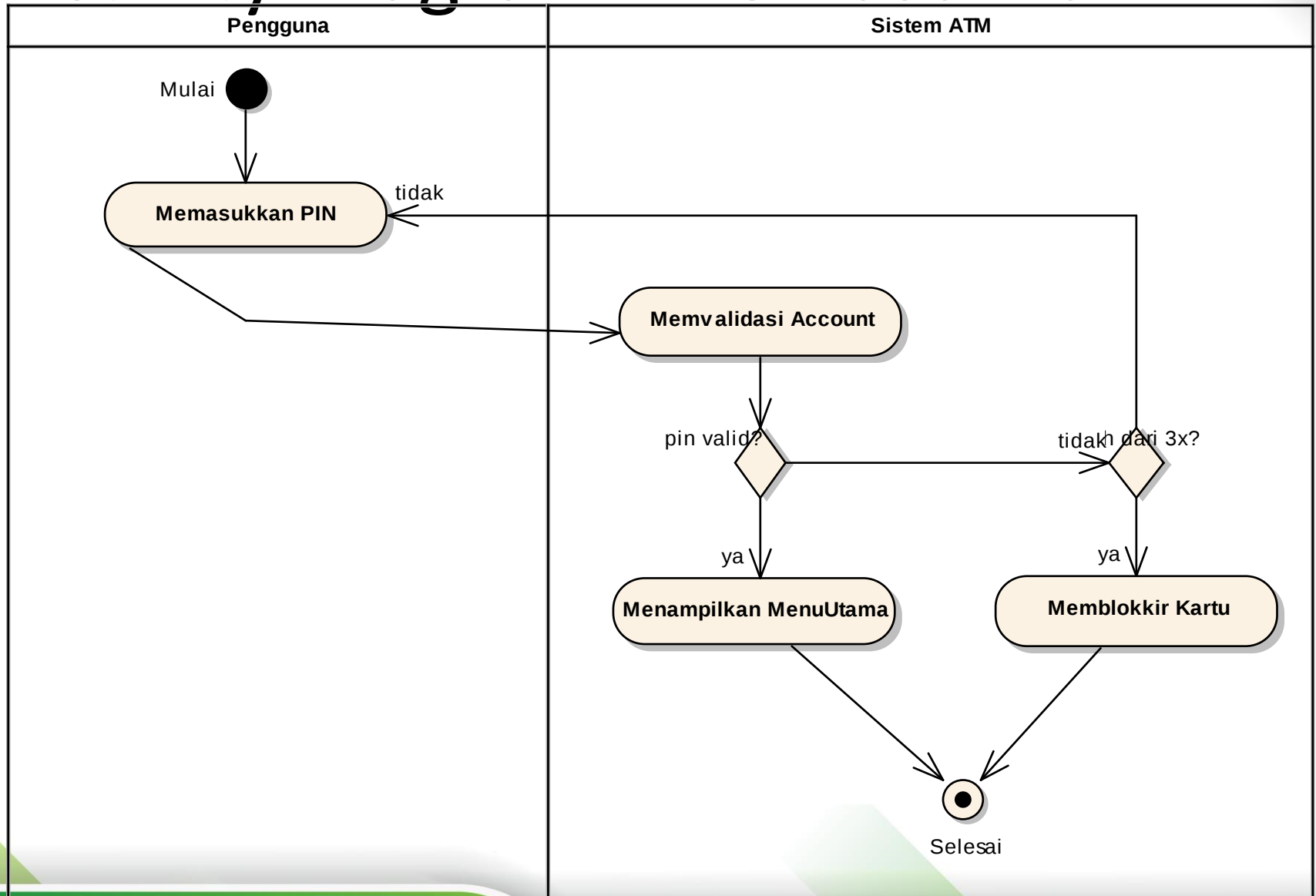
- A number of activities support a business process across several departments
- Activity diagrams model the behavior in a business process

STUDI KASUS: ATM SYSTEM

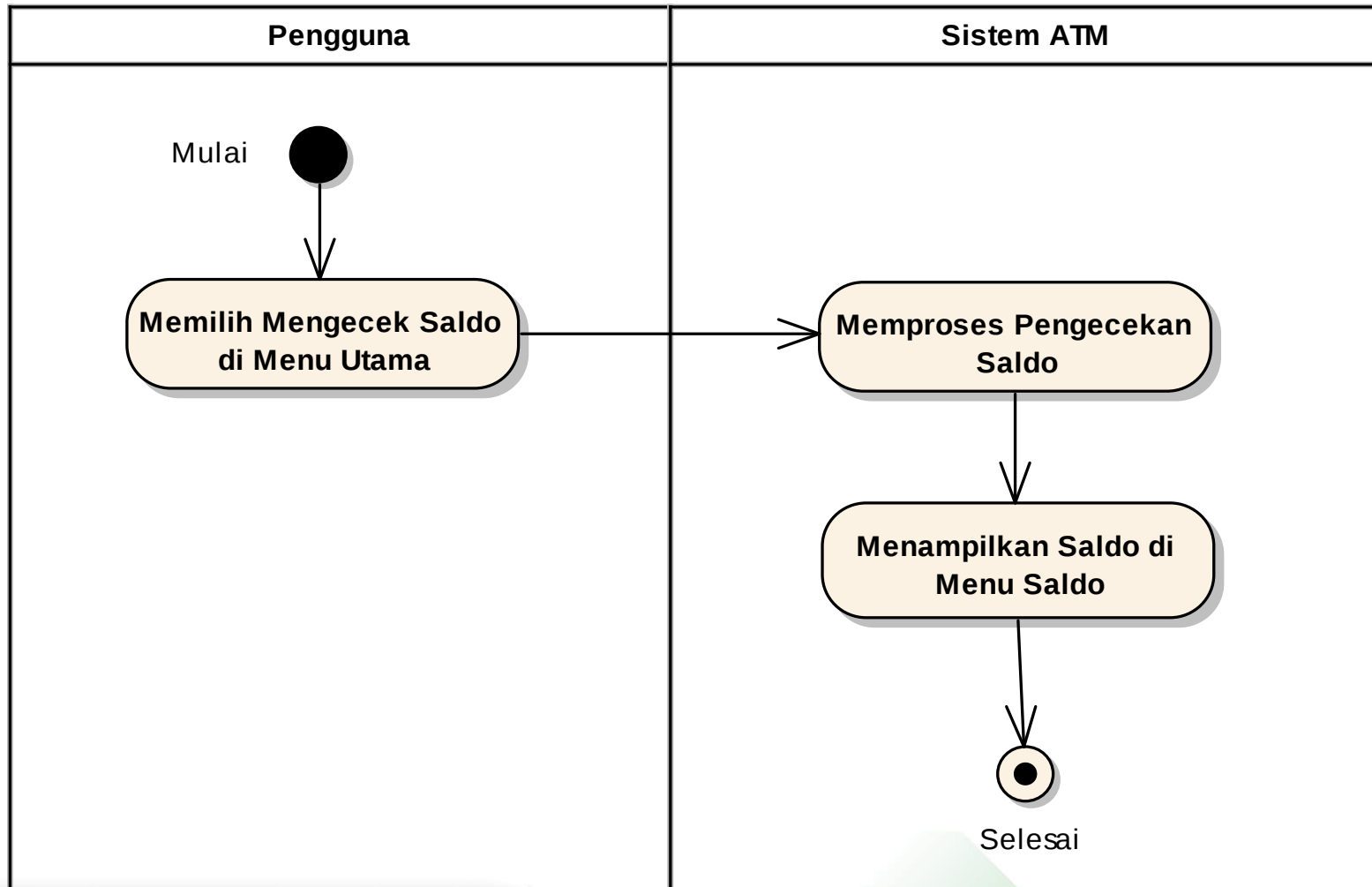
Activity Diagram: Memasukkan Kartu



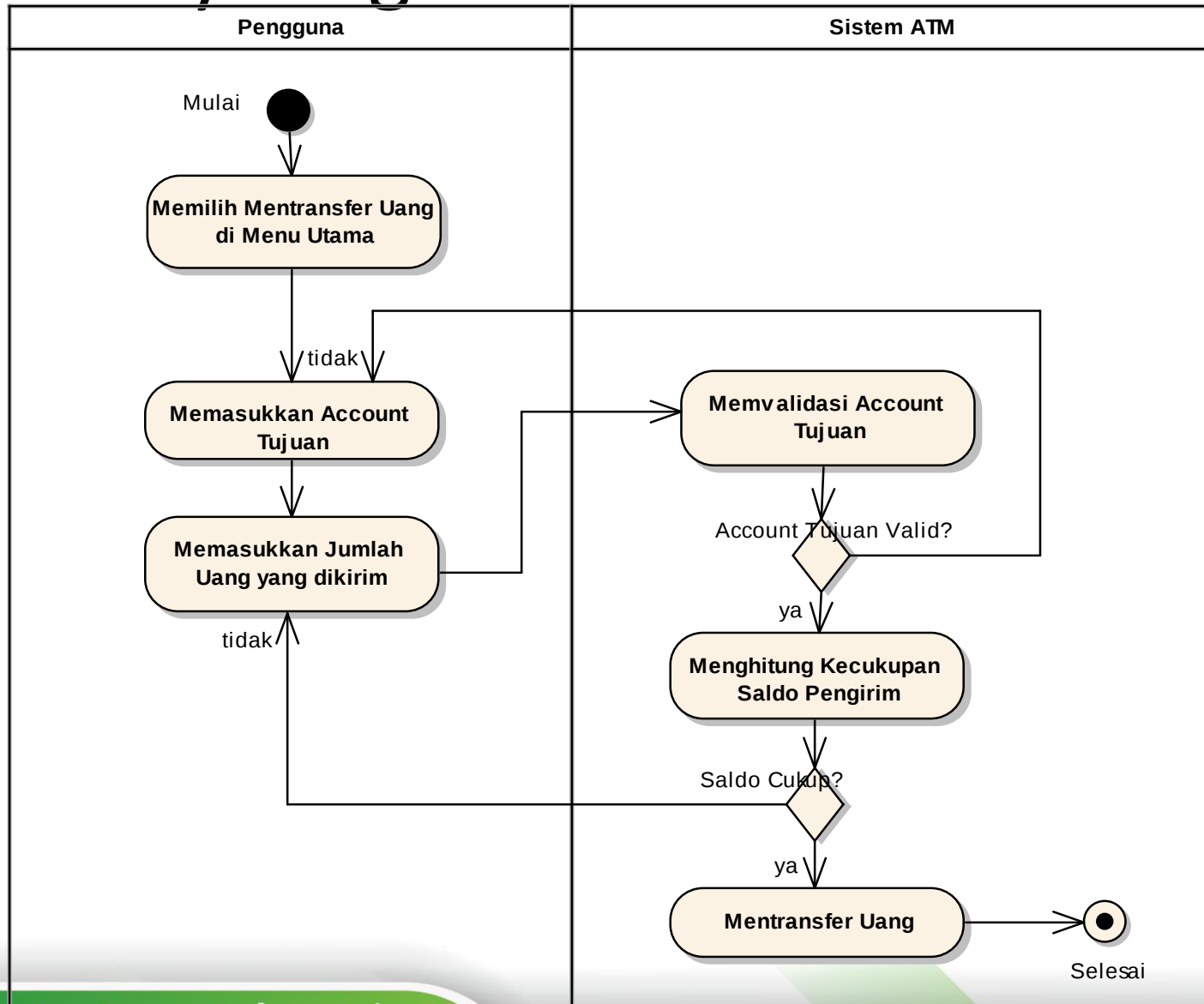
Activity Diagram: Memasukkan PIN



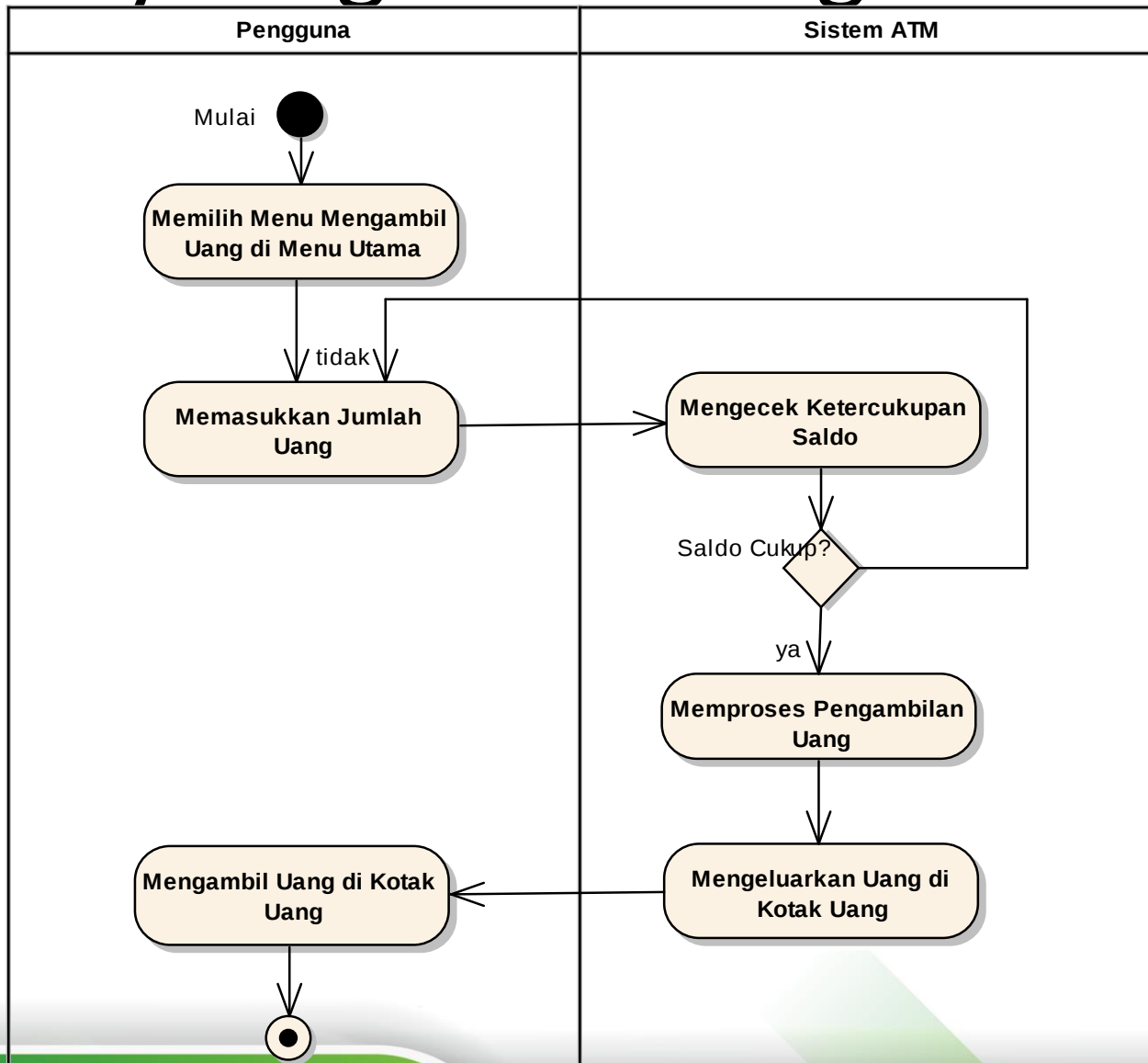
Activity Diagram: Mengecek Saldo



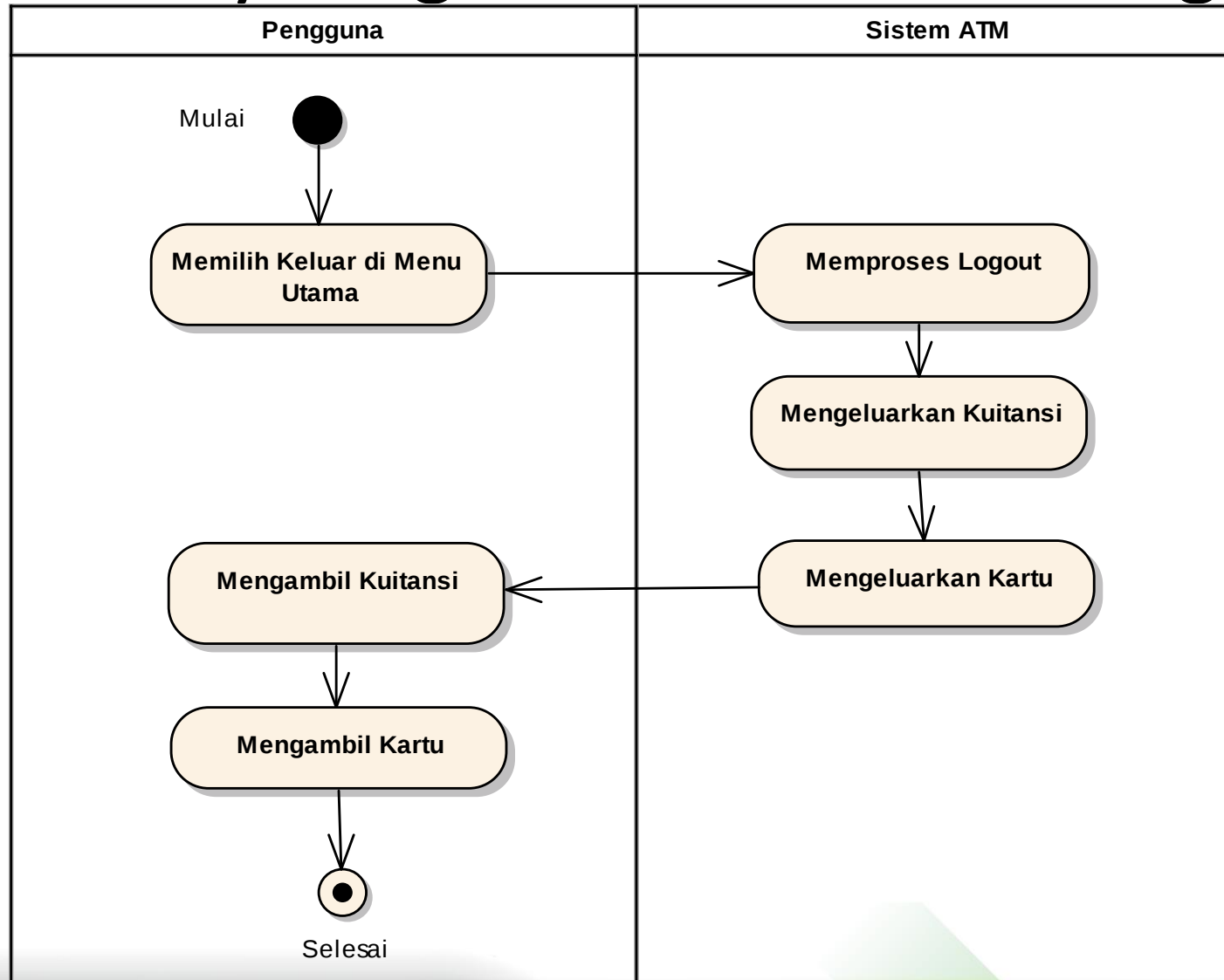
Activity Diagram: Mentransfer Uang



Activity Diagram: Mengambil Uang



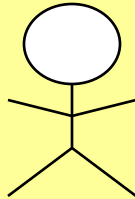
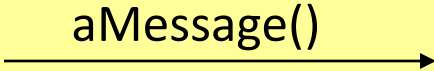
Activity Diagram: Melakukan Logout



Sequence Diagrams

- Illustrate the **objects that participate** in a use case
- Show the **messages that pass between objects** for a particular use-case over time

Sequence Diagram Syntax

AN ACTOR	
AN OBJECT	
A LIFELINE	
A FOCUS OF CONTROL	
A MESSAGE	
OBJECT DESTRUCTION	

Sequence Diagram

1. Susun Sequence Diagram **untuk setiap Use Case** yang dibuat
2. Mulai dari menarik Actor yang ada di Use Case Diagram, lanjutkan dengan membuat sequence detail dari berjalannya Use Case

Catatan: **Objek** dari **Lifeline di Sequence Diagram** akan menjadi **kandidat Class**

Jenis Class

1. Boundary Class:

1. Class yang berinteraksi dengan aktor langsung (user interface)
2. Form, input, UI ini masuk di sini

2. Control Class:

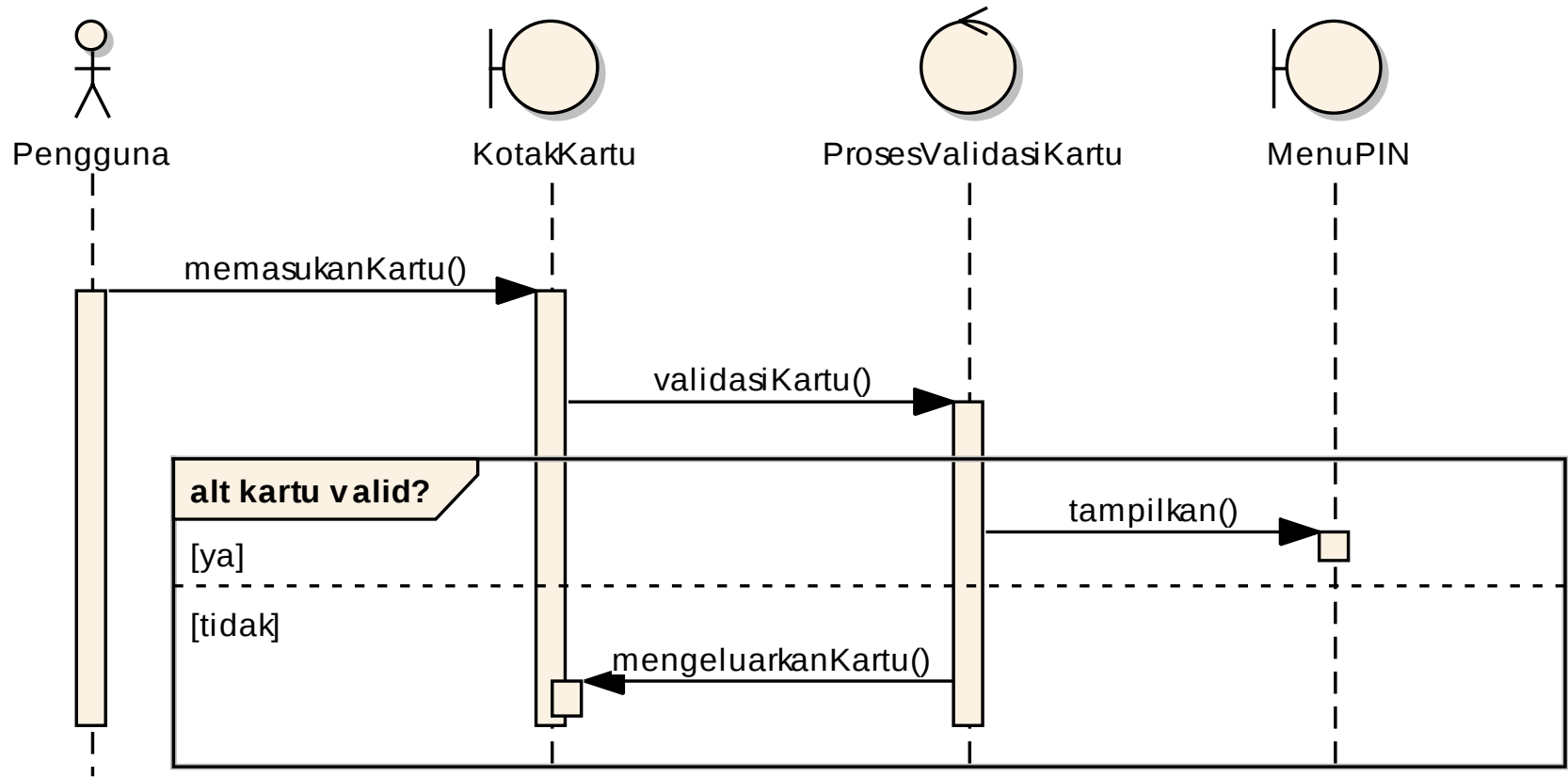
1. Class yang berhubungan dengan pemrosesan, penghitungan, kalkulasi, komputasi, query, dst

3. Entity Class:

1. Class yang berhubungan dengan data, penyimpanan data/file

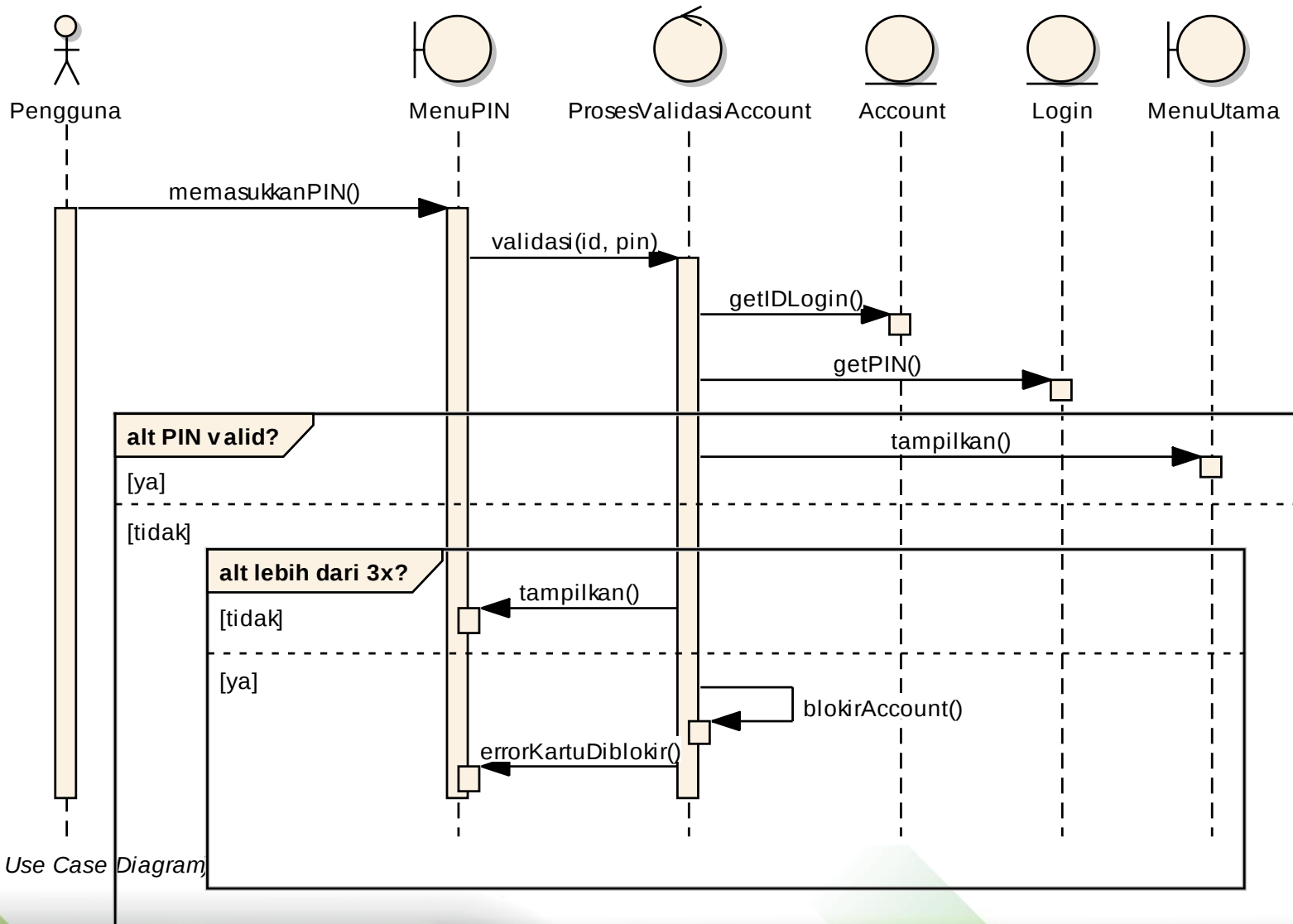
STUDI KASUS: ATM SYSTEM

Sequence Diagram: Memasukkan Kartu

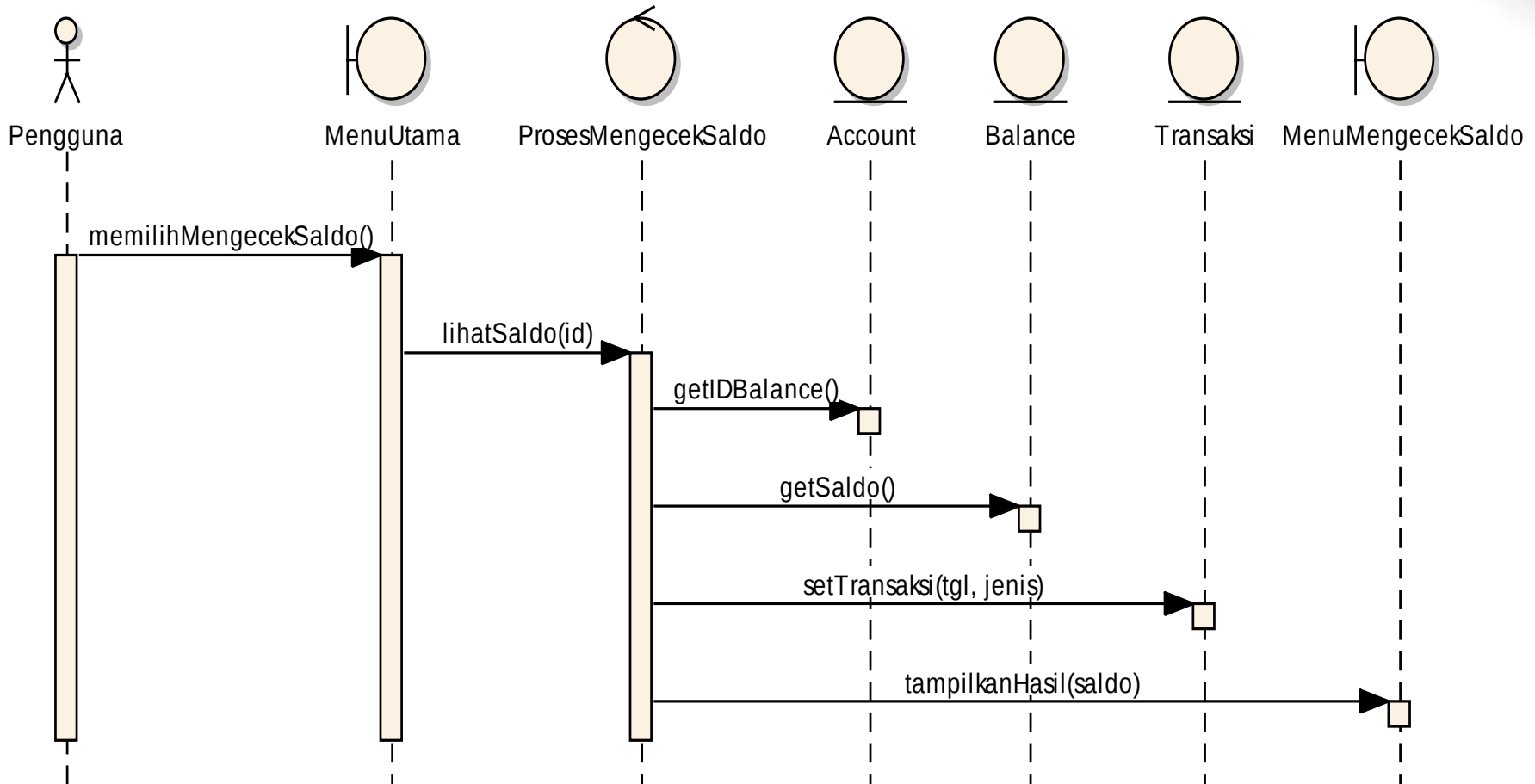


(from 1 Use Case Diagram)

Sequence Diagram: Memasukkan PIN

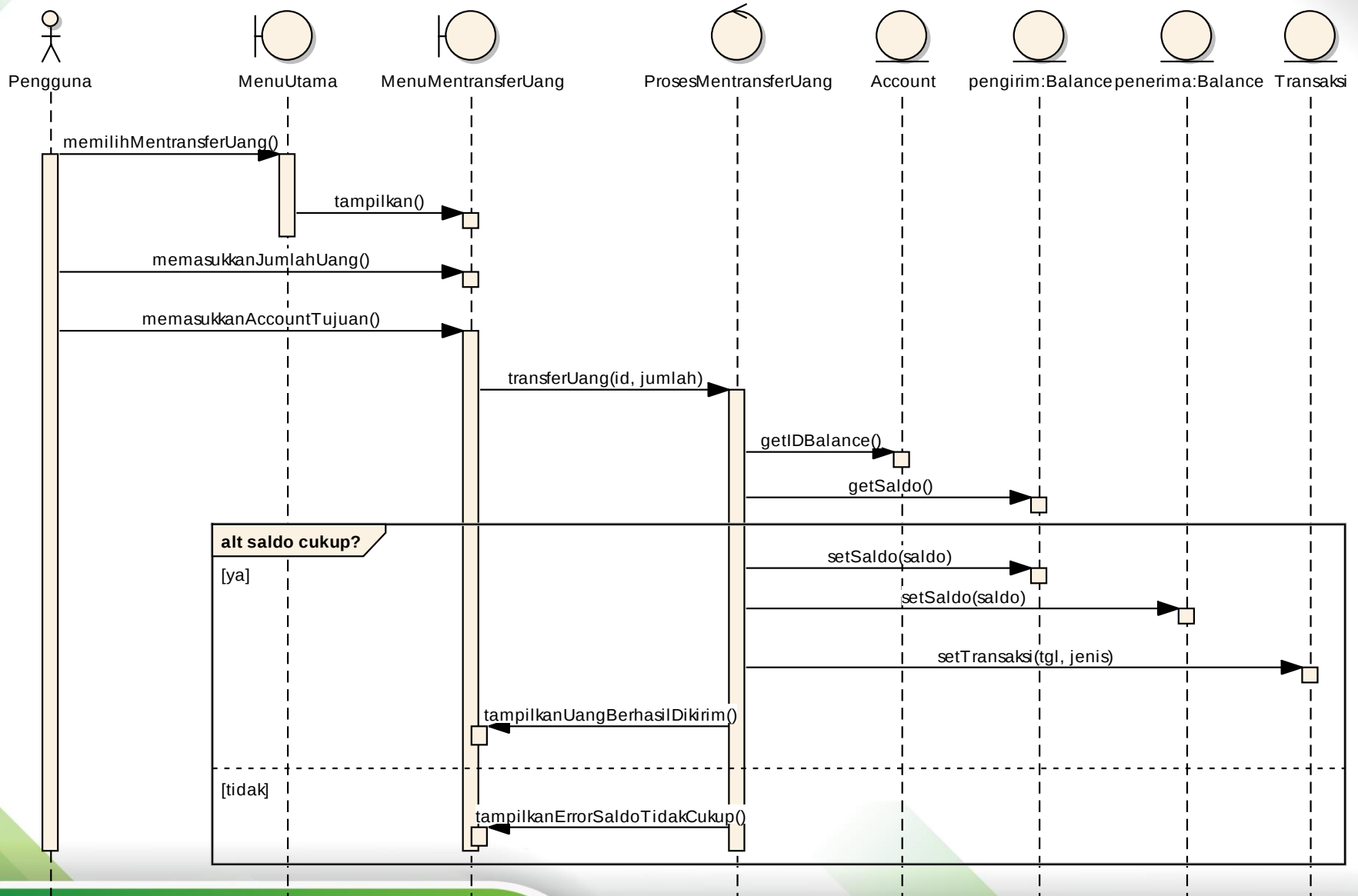


Sequence Diagram: Mengecek Saldo

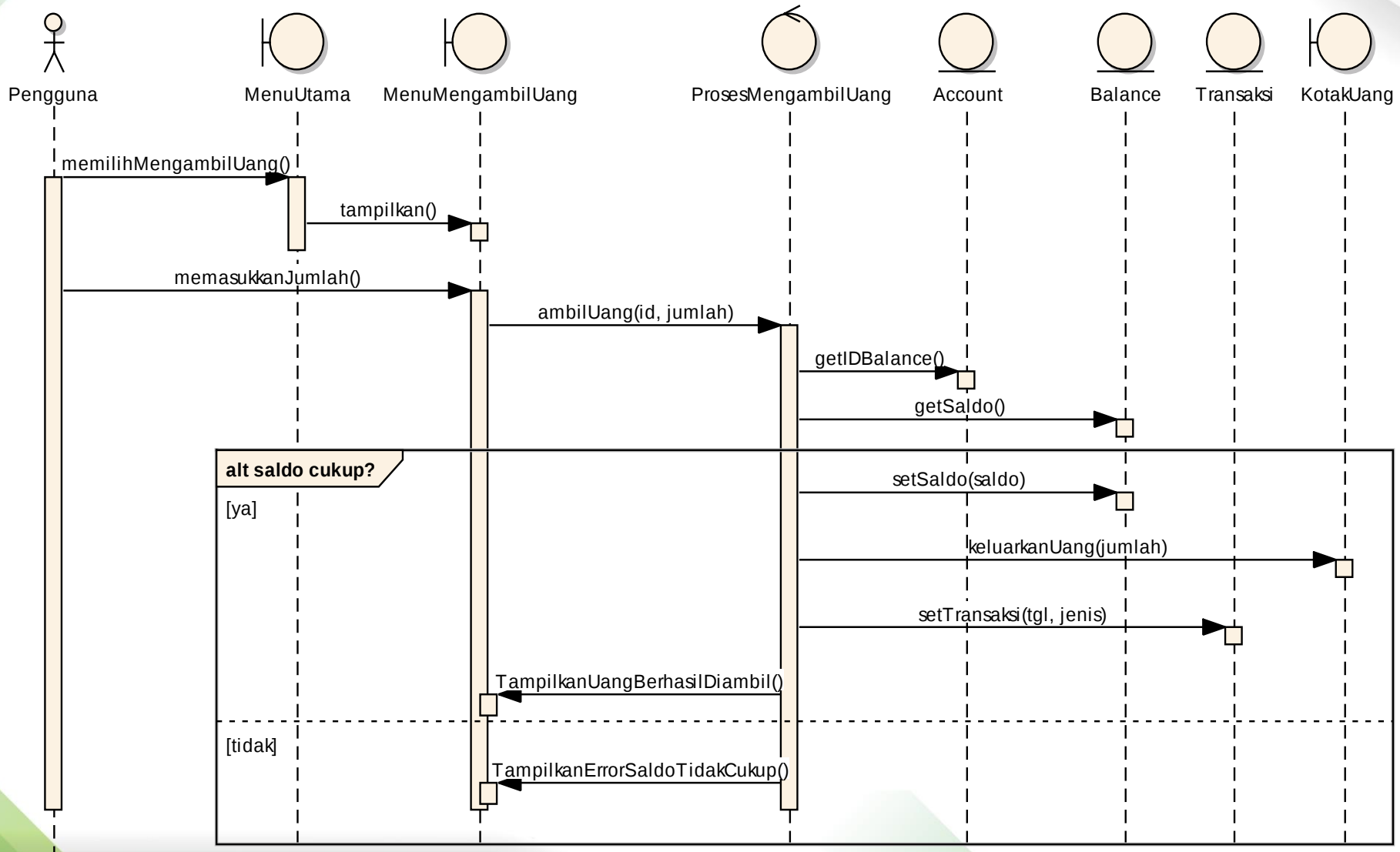


(from 1 Use Case Diagram)

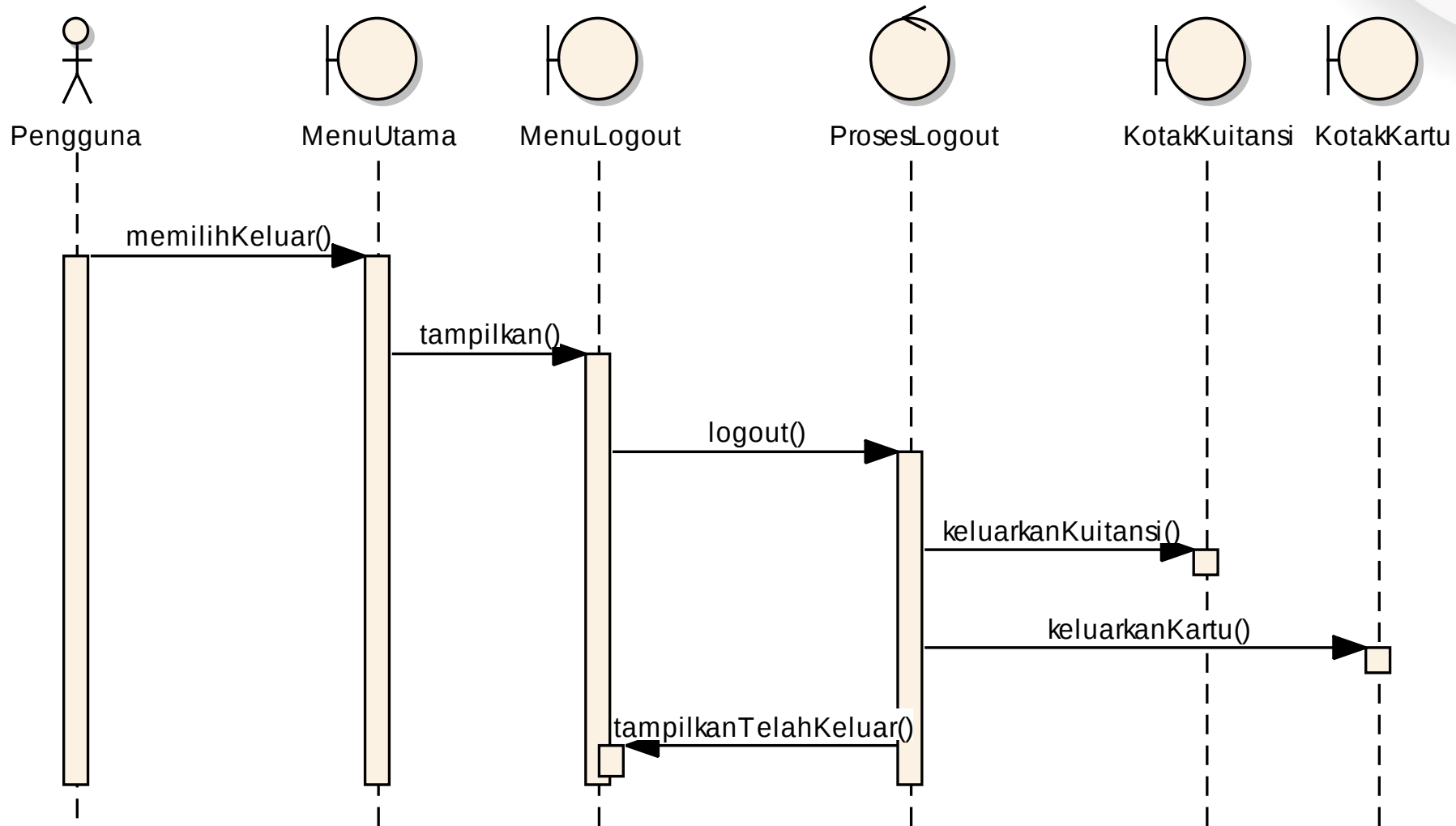
Sequence Diagram: Mentransfer Uang



Sequence Diagram: Mengambil Uang



Sequence Diagram: Melakukan Logout



(from 1 Use Case Diagram)

CLASS DIAGRAM

Class Diagram Elements

1. Classes
2. Attributes
3. Operations
4. Relationships

Classes

- **Templates** for creating instances or objects
- All objects of a class have same structure and behavior, but contain **different attributes**
 1. **Concrete**: used to create actual objects
 2. **Abstract**: extended to create other classes

Attributes

- Units of information relevant to the **description** of the class
- Only **attributes important** to the task should be included
- Attributes should be **primitive types** (integers, strings, doubles, date, time, Boolean, etc.)

Operations (Methods)

- Defines the **behavior** of the class
- **Action** that instances/objects can take
- Focus on relevant problem-specific operations (at this point)

Relationships

- **Generalization**

- “Is-A” relationship
- Enables inheritance of attributes & oper's
- Subclasses and superclasses
- Principle of substitutability
 - Subclass be substituted for superclass

- **Aggregation**

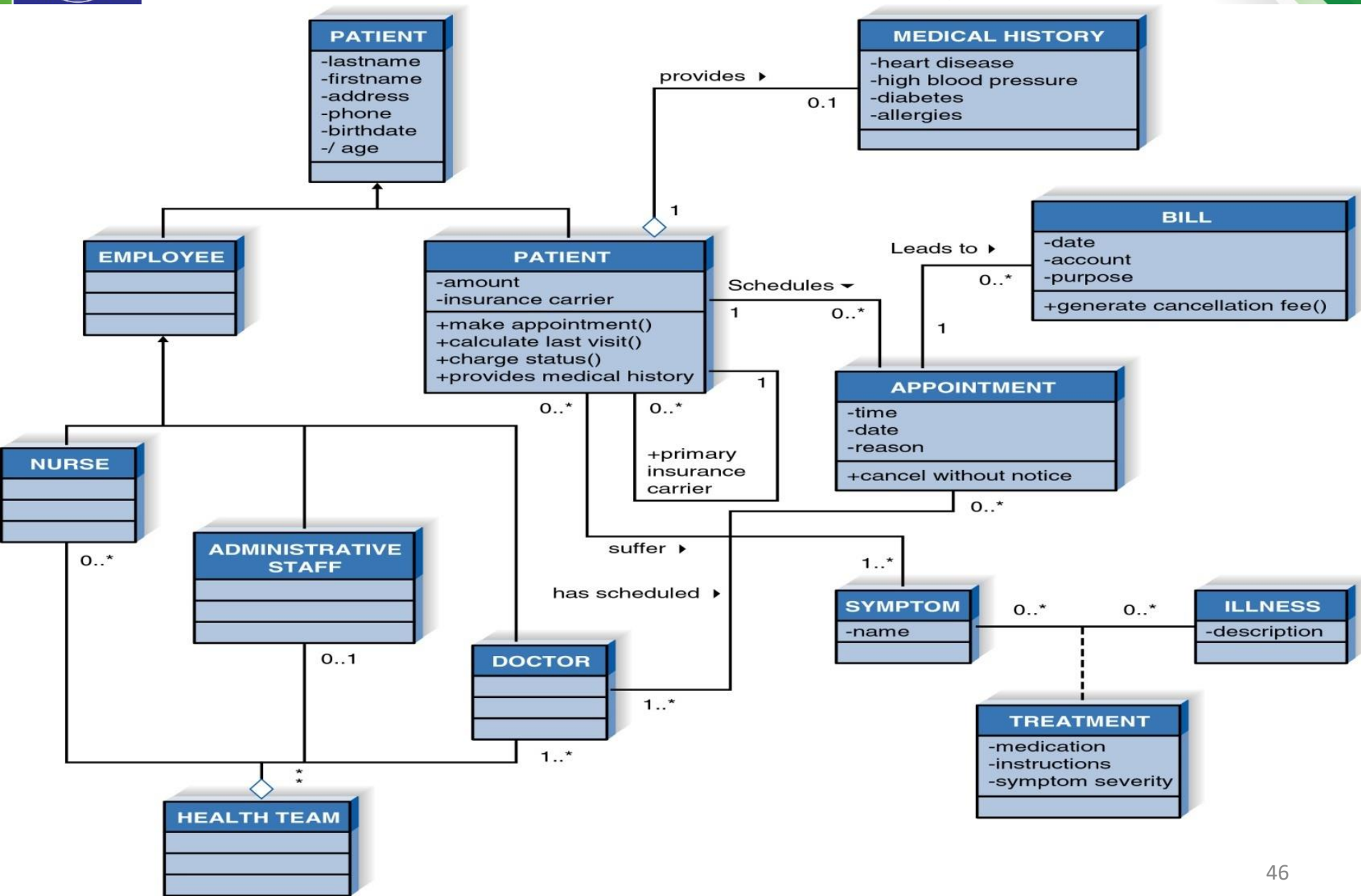
- “Has-A” relationship
- Relates parts to wholes
- Uses decomposition

- **Association**

- Relationships that don't fit “Is-A” or “Has-A”
- Often a weaker form of “Has-A”
- Miscellaneous relationships between classes
- Example:
 - Patient schedules an appointment
 - So the appointment has a patient

- This is weak

Example Class Diagram



More on Attributes

- **Derived attributes**
 - /age, for example can be calculated from birth date and current date
- **Visibility of attributes**
 - **+Public**: not hidden from any object
 - **#Protected**: hidden from all but immediate subclasses
 - **-Private**: hidden from all other classes
- **Default is private**

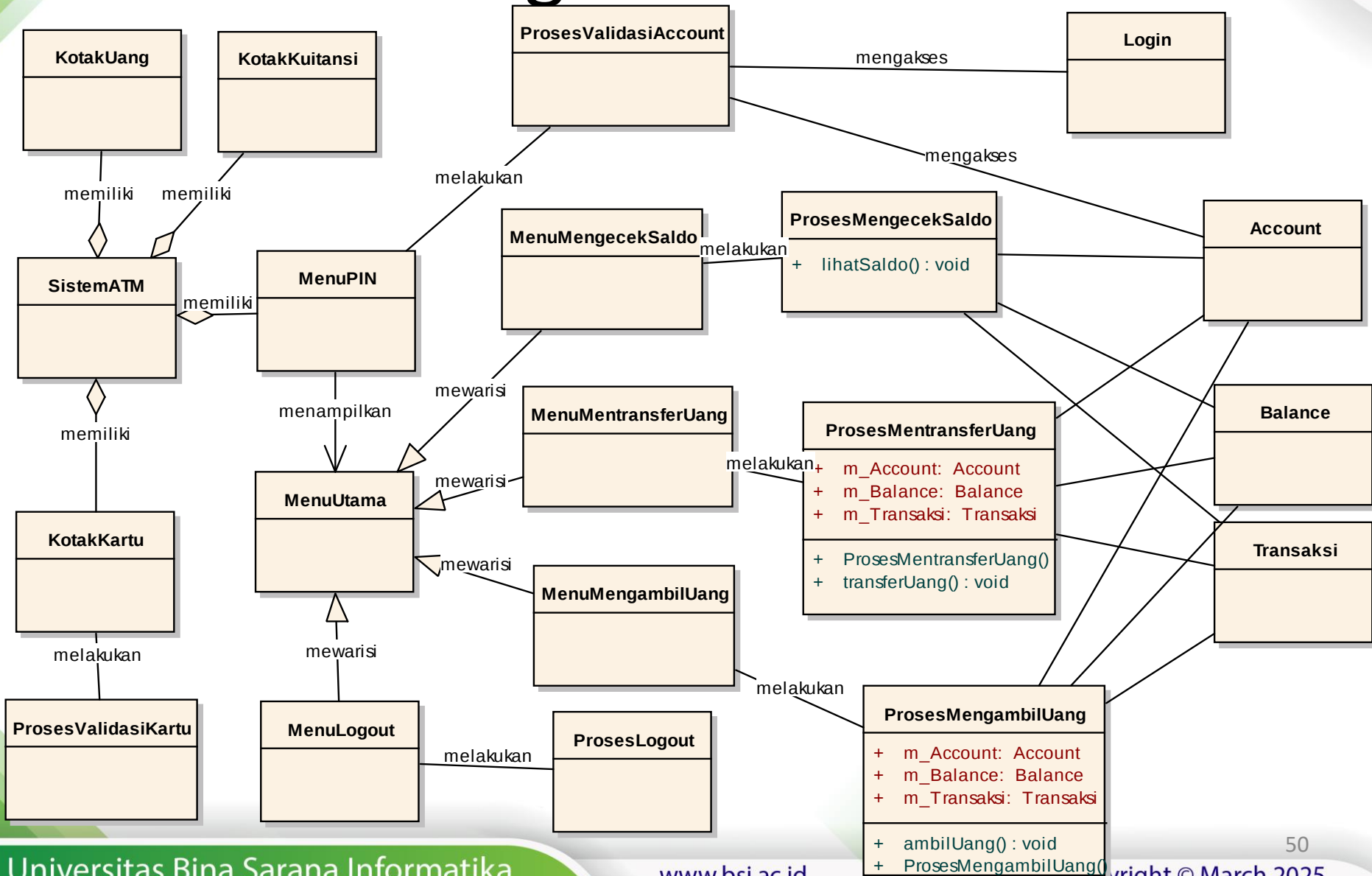
More on Operations

- **Constructor**: creates object
- **Query**: see class state
- **Update**: change attribute values
- Operations can also be public, protected, or private
 - Default for operations is public

More on Relationships

- A primary purpose of class diagrams is to show relationships, or *associations*, between classes
- Class can be related to itself (role)
 - Use a "+" sign to show it's a role and not the name of a relationship

Class Diagram: Sistem ATM



DEPLOYMENT DIAGRAM

Deployment Diagram

- **Servers**
 - Mainframes, Minis, Micros
- **Clients**
 - Input/Output HW used by users
 - Terminals, PCs, special purpose HW
- **Network**
 - HW and SW to connect clients to servers
- **Nodes**
 - Any piece of hardware in the model
 - A computational resource
 - Labeled by its name
 - Stereotype to label the type of node
- **Artifacts**
 - Piece of the information system, such as software or a database table
- **Node with Deployed Artifact**
 - Shows artifact placed on a physical node
 - Good for showing distribution data or software
- **Communication paths**
 - Links between nodes of the network

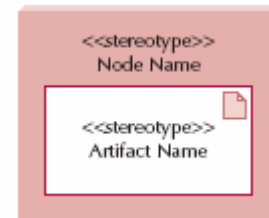
Node



Artifact



Node with
Deployment Artifact



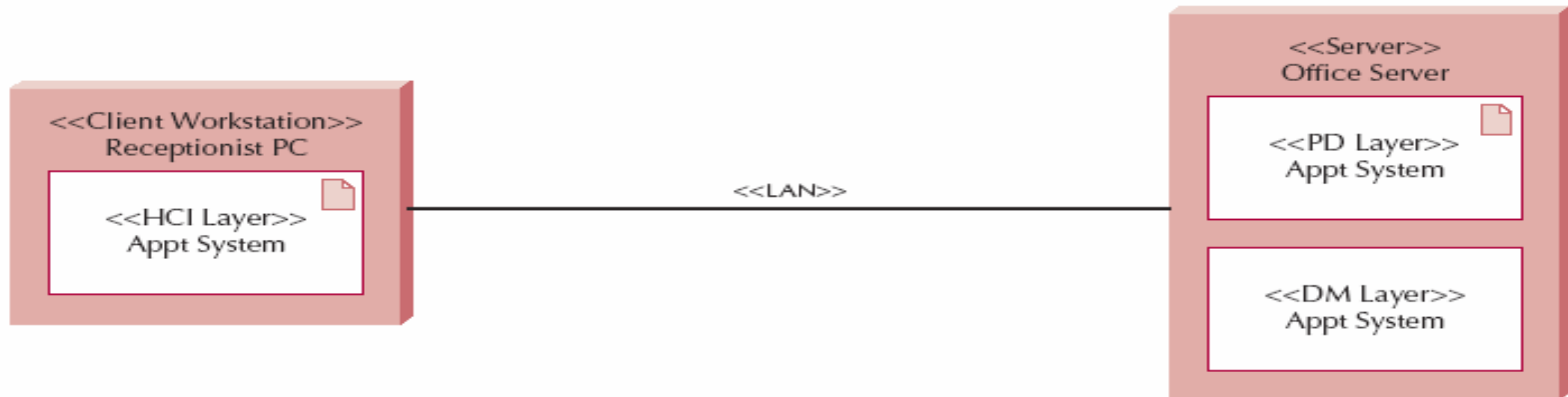
Communication
Path



Diagram Examples



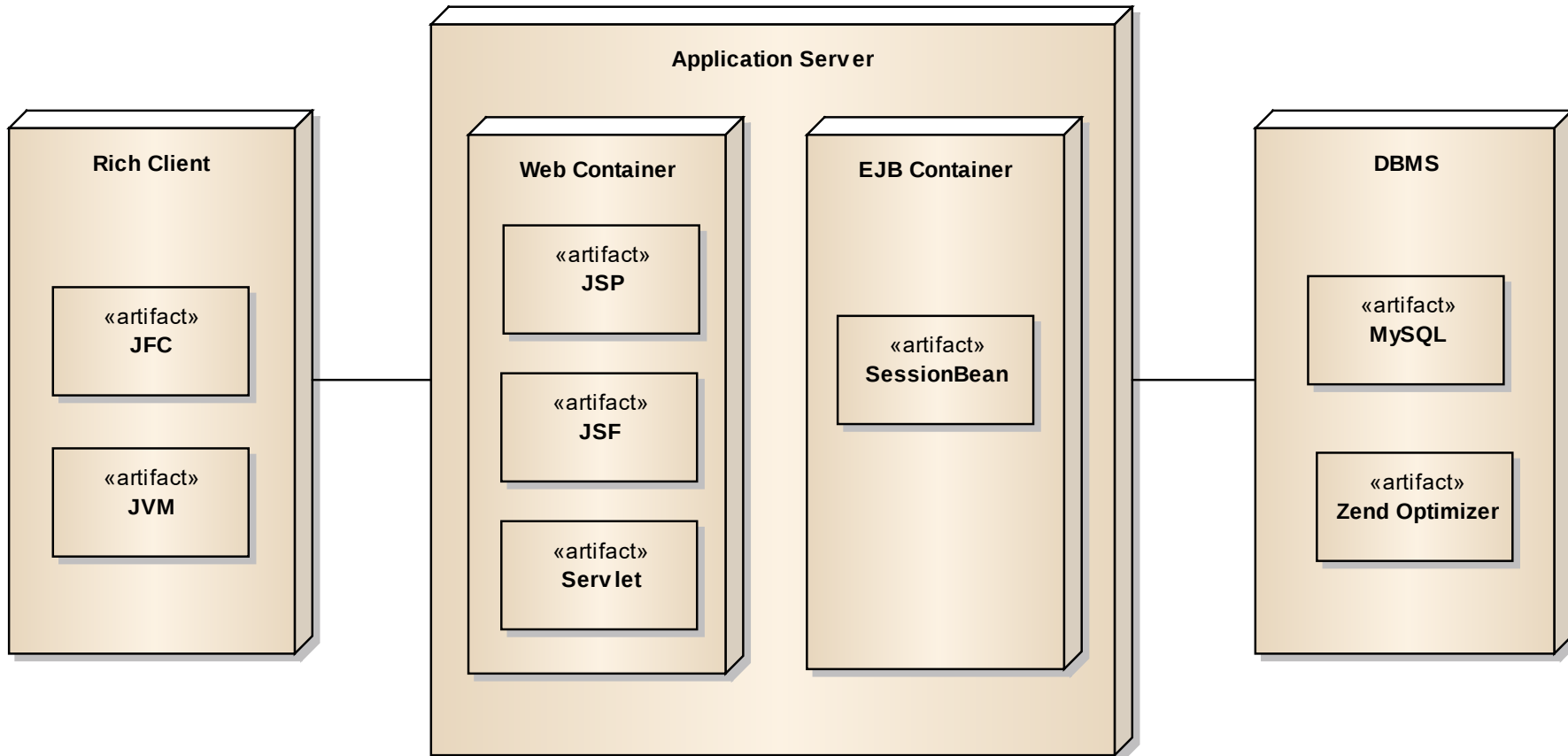
(A)



(B)



Deployment Diagram (3 Tier)



Pengembangan Permodelan Arsitektur Teknologi Menggunakan Framework TOGAF

Studi kasus Pengembangan Permodelan Arsitektur Teknologi Menggunakan Framework TOGAF yang dibahas berdasarkan penelitian yang dilakukan oleh Yunis dkk pada Tahun 2016.

Tahapan TOGAF ADM terdiri dari (Yunis dkk, 2016):

- 1) architecture vision,
- 2) business architecture,
- 3) information system architecture,
- 4) Technology architecture,
- 5) opportunities and solution,
- 6) Migration planning,
- 7) implementation governance, dan
- 8) architecture change management.

Model Arsitektur Bisnis

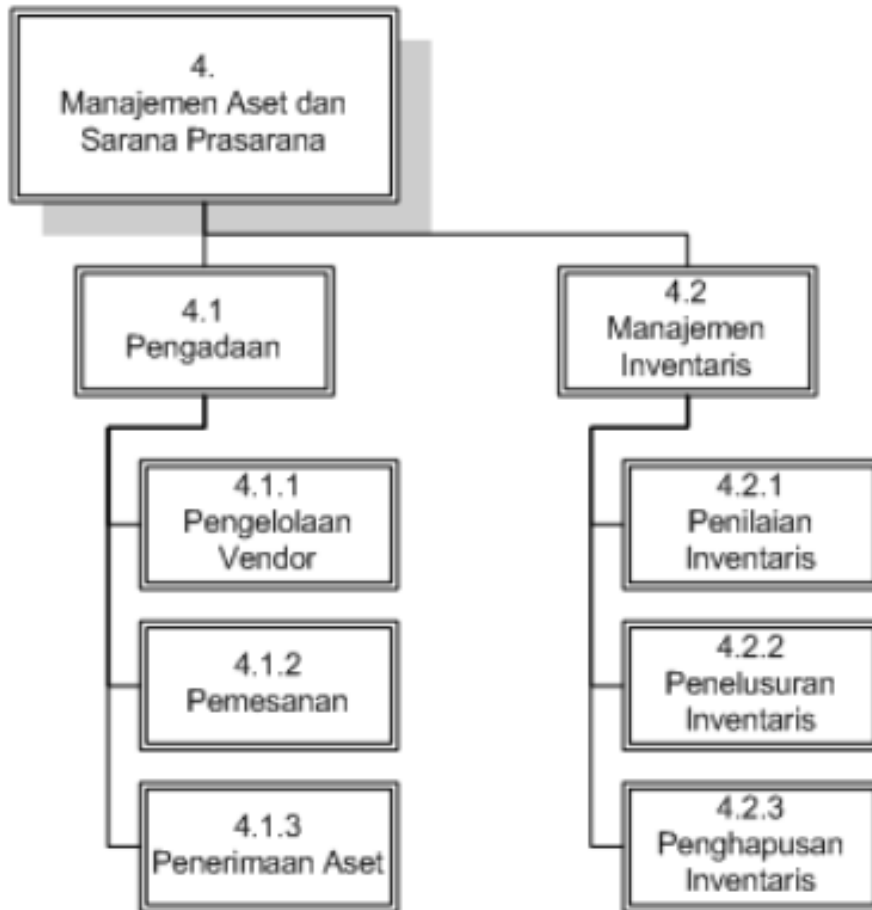
- Sebelum memulai pemodelan bisnis ini, maka hal yang harus dilakukan pertama kali adalah mendefinisikan suatu pertanyaan, yang berhubungan dengan pedoman yang digunakan perguruan tinggi dalam menjalankan bisnisnya.
- Kegiatan Tridharma Perguruan Tinggi merupakan aktivitas inti dalam sebuah perguruan tinggi. Aktivitas inti ini tidak akan bisa dijalankan dengan baik, apabila tidak ada dukungan dari aktivitas lainnya, seperti: manajemen keuangan, manajemen sarana dan prasarana serta manajemen sumber daya manusia dan lainnya.
- Untuk mengidentifikasi area fungsional utama dan pendukung dari perguruan tinggi digambarkan dalam bentuk rantai nilai seperti diilustrasikan pada Gambar 1.
- Masing masing kegiatan utama dan pendukung dapat diuraikan sebagai berikut:
Fungsi Bisnis Utama terdiri dari tiga kegiatan yaitu: Penerimaan Mahasiswa, Operasional Akademik, dan Pengelepasan Akademik.
Fungsi Bisnis Pendukung terdiri dari tiga kegiatan yaitu: Manajemen Aset dan Prasarana, Manajemen Sumber Daya Manusia, dan Manajemen Keuangan

uc Rantai Nilai Diagram



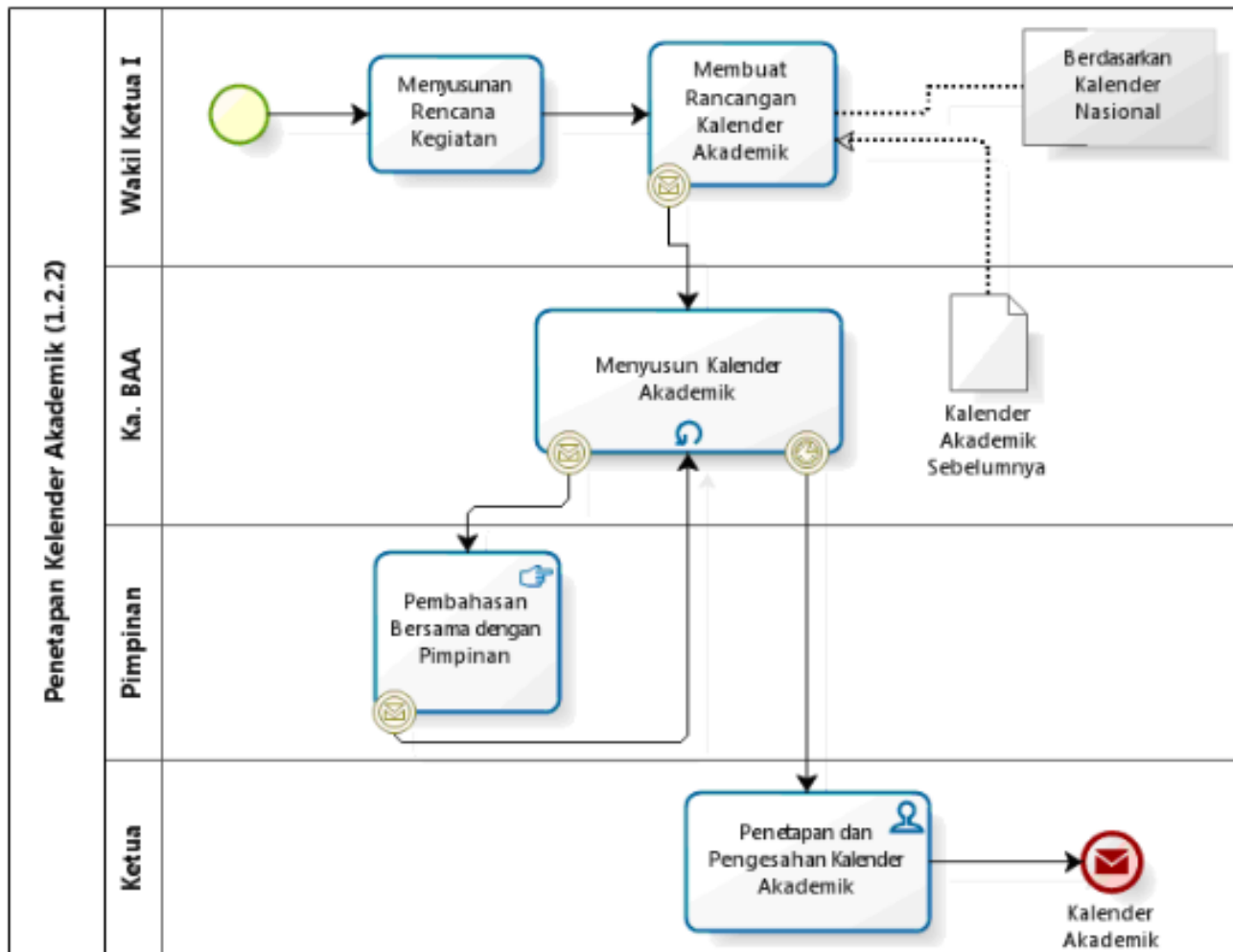
Gambar 1. Rantai Nilai Perguruan Tinggi

- Untuk memodelkan arsitektur bisnis, terdapat berbagai teknik yang bisa digunakan, seperti Business Process Modeling Notation (BPMN), Functional Decomposition Diagram (FDD), diagram Unified Modeling Language (UML), dan Integration Definition for Function Modeling (IDEF0). Pada tahapan kedua dari TOGAF ADM yaitu Arsitektur Bisnis (Business Architecture) juga menyediakan teknik yang bisa digunakan untuk memodelkan arsitektur bisnis tersebut, antara lain BPMN.
- Sebelum memodelkan arsitektur bisnis yang ada di perguruan tinggi, berikut akan dilakukan terlebih dahulu perumusan turunan fungsi bisnis dan fungsi pendukung berdasarkan rantai nilai dari Gambar 1 yang sudah dijelaskan sebelumnya. Adapun bentuk turunan dari fungsi bisnis utama dan pendukung akan digambarkan dengan Functional Decomposition Diagram (FDD). Salah satu contoh FDD bisa dilihat pada Gambar 2.



Gambar 2. FDD Manajemen Aset Sarana dan Prasarana

- Agar model arsitektur bisnis dapat dipahami dengan baik maka fungsi-fungsi bisnis yang telah didefinisikan dapat digambarkan dengan BPMN. Model proses BPMN merupakan representasi grafis mengenai satu atau beberapa aspek sistem manajemen dari suatu organisasi, karena secara langsung juga memberikan gambaran yang jelas tentang siapa pembuat keputusan untuk setiap proses. Salah satu contoh BPMN untuk fungsi penetapan kalender akademik perguruan tinggi ditunjukkan pada Gambar 3. Disamping hasil analisis dan dokumentasi proses bisnis yang sudah dibuat, sebaiknya juga diperlukan evaluasi terhadap proses bisnis sehingga menghasilkan proses bisnis yang lebih efektif dan efisien.



Gambar 3. BPMN Penetapan Kalender Akademik

Model Arsitektur Data

- Arsitektur data haruslah dapat mengidentifikasi data yang mendukung fungsi-fungsi bisnis seperti yang terdefinisi dalam model bisnis. Untuk mendefinisikan arsitektur data, pertama sekali didaftarkan kandidat entitas data dengan melakukan brainstorming terhadap orang, tempat, dan kejadian yang memiliki makna (informasi) sehubungan dengan model bisnis perguruan tinggi.
- Untuk mendefinisikan arsitektur data, TOGAF ADM merekomendasikan sebuah katalog (Data Component Catalog) yang berisikan kumpulan data yang ada dalam organisasi. Rekomendasi lainnya adalah Business Function Matrix yang menggambarkan hubungan relasi fungsi bisnis dengan entitas data yang ada dalam perguruan tinggi. Berikut akan dijelaskan tentang sebagian Data Component Catalog dalam sebuah perguruan tinggi.

Copyright © March 2025

Entitas-entitas Operasional Akademik termasuk diantaranya adalah :

1. Entitas Program Studi
2. Entitas Jurusan
3. Entitas Sarana Prasarana (Ruang Kuliah)
4. Entitas Semester
5. Entitas Kurikulum
6. Entitas Kalender Akademik
7. Entitas Herregistrasi (Registrasi Kuliah)
8. Entitas Dosen
9. Entitas Mahasiswa
10. Entitas Matakuliah
11. Entitas Jadwal Kuliah
12. Entitas Berita Acara Kuliah
13. Entitas Absensi Kuliah
14. Entitas Jadwal Ujian
15. Entitas Beritas Acara Ujian (UTS & UAS)
16. Entitas Nilai (KHS, Transkrip)
17. Entitas KRS
18. Entitas Cuti Akademik
19. Entitas Registrasi Sidang Akademik
20. Entitas Registrasi Judul PI
21. Entitas Registrasi Judul TA
22. Entitas Evaluasi

Berdasarkan kandidat entitas data yang sudah didefinisikan tersebut langkah selanjutnya adalah menggambarkan arsitektur data yang dibutuhkan dalam fungsi dan model bisnis yang sebelumnya sudah dijelaskan. Arsitektur data akan disajikan dalam bentuk diagram E-R, diagram E-R dipilih mengingat arsitektur data hanyalah mengidentifikasi dan mendefinisikan entitas data.

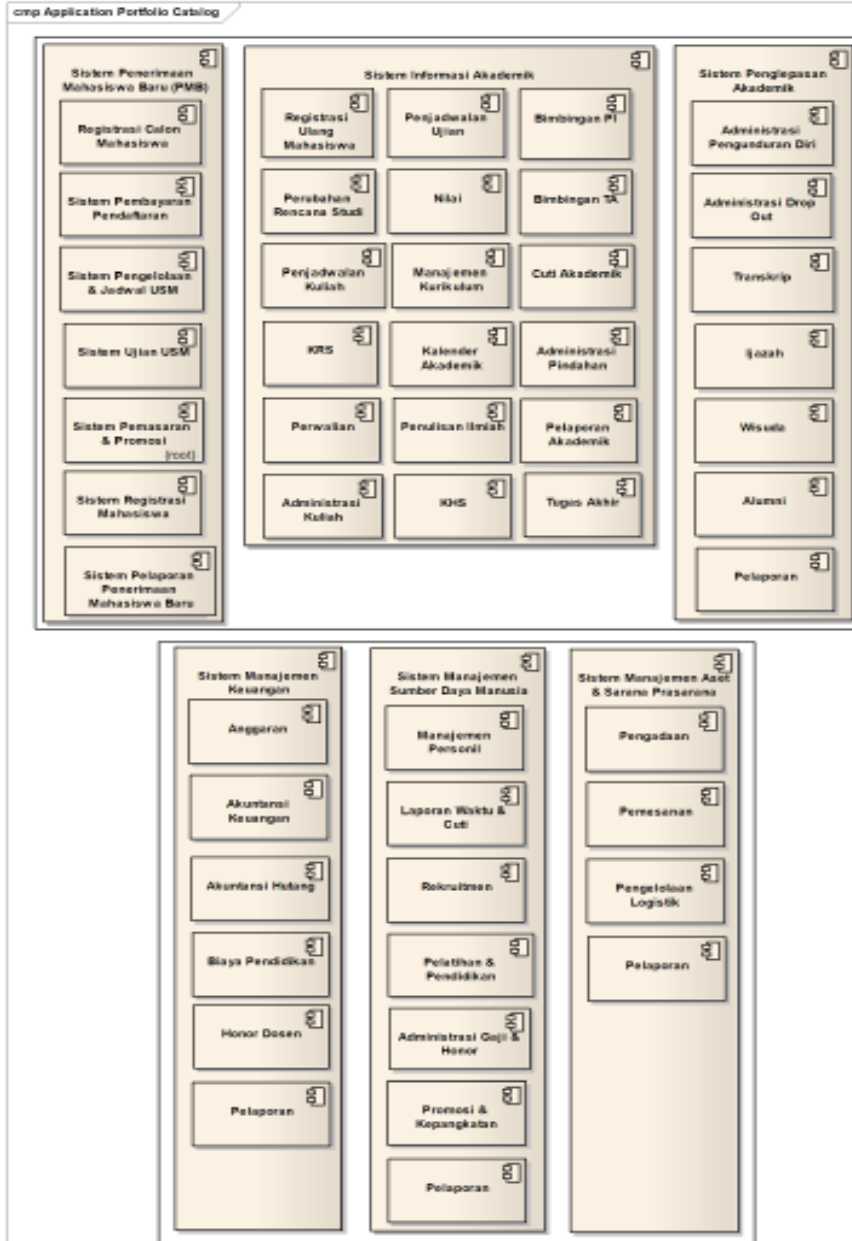
Untuk keperluan memvisualkan entitas-entitas data tersebut beserta makna konseptualnya, diagram E-R sudah cukup baik dan memadai. Pada Gambar 4 merupakan salah satu contoh dari Diagram ER untuk Fungsi Operasional Akademik.

Model Arsitektur Aplikasi

Arsitektur Aplikasi mempunyai tujuan untuk mendefinisikan aplikasi-aplikasi yang diperlukan untuk mengelola data dan mendukung fungsi-fungsi bisnis yang ada dalam perguruan tinggi. Model arsitektur aplikasi merupakan definisi tentang hal yang dilakukan aplikasi untuk mengelola data dan menyediakan informasi bagi pelaksana fungsifungsi bisnis. Arsitektur aplikasi dapat didefinisikan dengan teknik-teknik Application Portfolio Catalog, System -/Function Matrix dan Application Communication Catalog.

Tahapan Application Portfolio Catalog bertujuan untuk mendefinisikan semua daftar aplikasi yang digunakan oleh perguruan tinggi, arsitektur aplikasi mempunyai tujuan untuk mengelola data-data dan fungsi bisnis yang ada dalam perguruan tinggi. Contoh dari Application Portfolio Catalog aplikasi dalam perguruan tinggi dapat dilihat pada Gambar 6.

Gambar 5: Application Portfolio Catalog

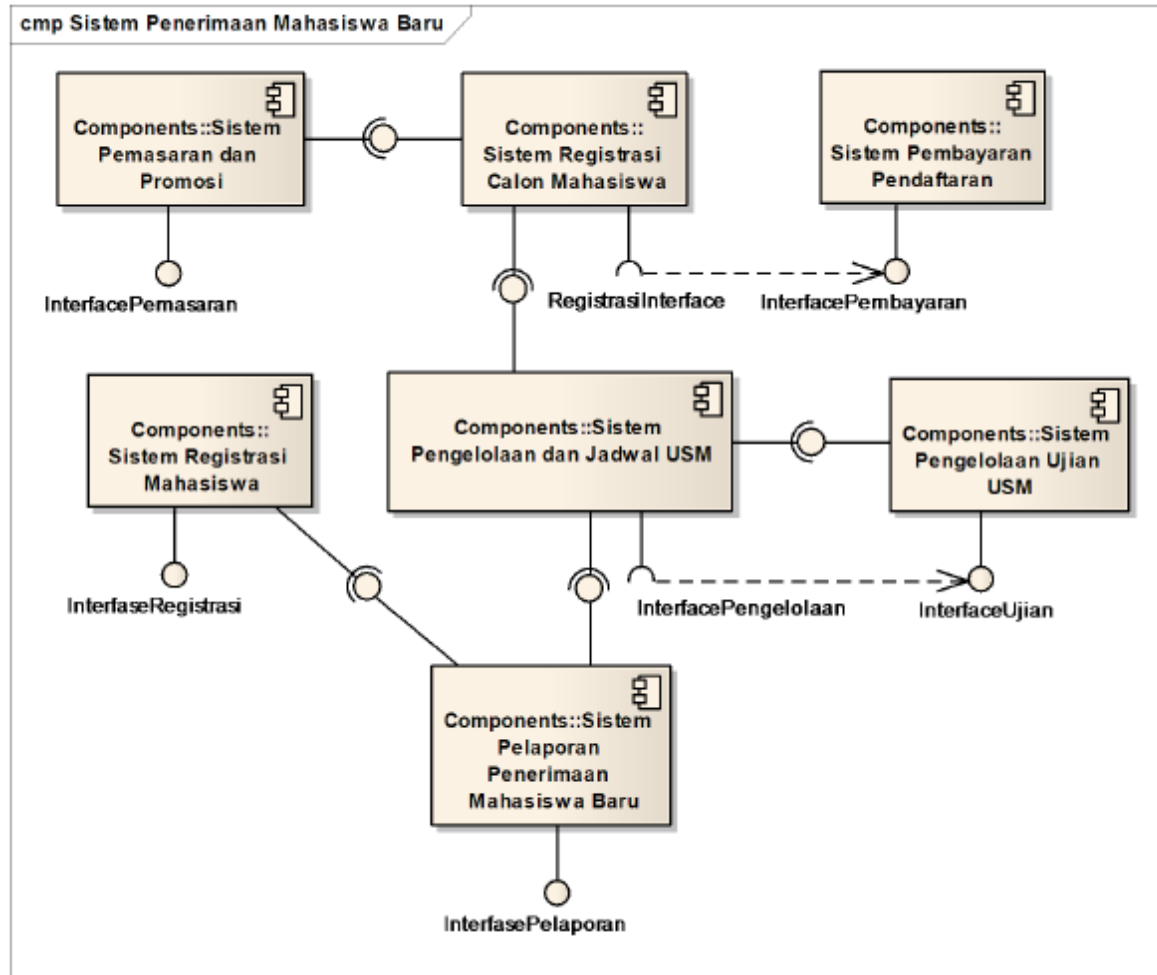


Kemudian tahap System/Function Matrix bertujuan untuk mengidentifikasi fungsi-fungsi bisnis yang secara langsung didukung atau dilakukan oleh aplikasi. Secara umum langkah pemetaan hubungan aplikasi dengan fungsi bisnis adalah dengan cara: 1) menetapkan penggunaan aplikasi yang digunakan fungsi bisnis organisasi berdasarkan kebutuhan dari fungsi bisnis dengan meninjau proses bisnis dan data yang digunakan, 2) menentukan kebijakan penggunaan dan bagaimana layanan yang ada dalam aplikasi yang mendukung fungsi bisnis, 3) mendukung analisis kesenjangan antara peranan aplikasi dalam mendukung proses bisnis organisasi, 4) menentukan peranan aplikasi dalam mendukung fungsi bisnis dan mengidentifikasi kebutuhan untuk perubahan aplikasi kedepannya. Contoh dari matriks relasi aplikasi dengan fungsi bisnis dapat dilihat pada Gambar 6.

Tahap terakhir dalam model arsitektur aplikasi adalah Application Communication Catalog yang bertujuan untuk menggambarkan semua model dan pemetaan yang berhubungan dengan bentuk komunikasi antara aplikasi yang digunakan oleh Perguruan Tinggi. Diagram ini juga akan menunjukkan aplikasi dan interface antara komponen aplikasi yang akan digunakan. Gambaran komunikasi aplikasi ini digambarkan secara logis dengan harapan dapat menunjukkan arsitektur aplikasi yang relevan untuk dirancang dan dikembangkan. Contoh dari diagram ini, bisa dilihat pada Gambar 7.

APLIKASI	FUNGSI	FUNGSI PENDUKUNG																														
		Manajemen Aset dan Sarana Prasarana									Manajemen Sumber Daya Manusia										Manajemen Keuangan											
		Pengelolaan Vendor	Pengelolaan Pemesanan Barang/Jasa	Pengelolaan Penerimaan Aset	Pengelolaan Pembelian Inventaris	Pelaksanaan Pemeliharaan Inventaris	Pengelolaan Penghapusan Inventaris	Pengelolaan laporan dan evaluasi sarana prasarana	Pengelolaan Potongan	Pengelolaan Cuti	Pengelolaan dan Pengumpulan Laporan Waktu	Pengelolaan Tunjangan	Pengelolaan Perhitungan Honor	Pengelolaan dan Perhitungan Gaji	Pengelolaan Informasi Personil	Pengelolaan dan Pelaksanaan Rekrutmen	Peminaan Personil	Pengembangan Personil	Pengelolaan pelaporan dan evaluasi SDM	Perencanaan dan Pelaporan Anggaran	Penyusunan Anggaran	Pengawasan Anggaran	Pelaksanaan Revisi Anggaran	Menetapkan Alokasi Anggaran	Pelaksanaan Monitoring dan Evaluasi Anggaran	Menetapkan Akuntansi Keuangan	Pengelolaan Pelaporan Keuangan	Pengelolaan Penerimaan Keuangan	Pengelolaan Akuntansi Hutang dan Pembayaran	Pengelolaan Biaya Pendidikan	Pengelolaan Pembayaran Honor Dosen	
Sistem Manajemen Personil																																
Sistem Manajemen Laporan Waktu & Cuti																																
Sistem Manajemen Rekrutmen																																
Sistem Manajemen Pelatihan dan Pendidikan																																
Sistem Administrasi Honor, Gaji dan Potongan																																
Sistem Manajemen Promosi dan Kepangkatan																																
Sistem Pelaporan Manajemen Sumber Daya Manusia																																
Sistem Pengadaan																																
Sistem Pemesanan																																
Sistem Pengelolaan Logistik																																
Sistem Pelaporan Manajemen Aset & Sarana Prasarana																																
Sistem Anggaran																																
Sistem Akuntansi Keuangan																																
Sistem Akuntansi Hutang dan Pembayaran																																
Sistem Pengelolaan Biaya Pendidikan																																
Sistem Pengelolaan Honor Dosen																																
Sistem Pelaporan Keuangan																																

Gambar 6: Matrik Relasi Aplikasi dengan Fungsi Bisnis



Gambar 7: Diagram Komunikasi Aplikasi Penerimaan Mahasiswa

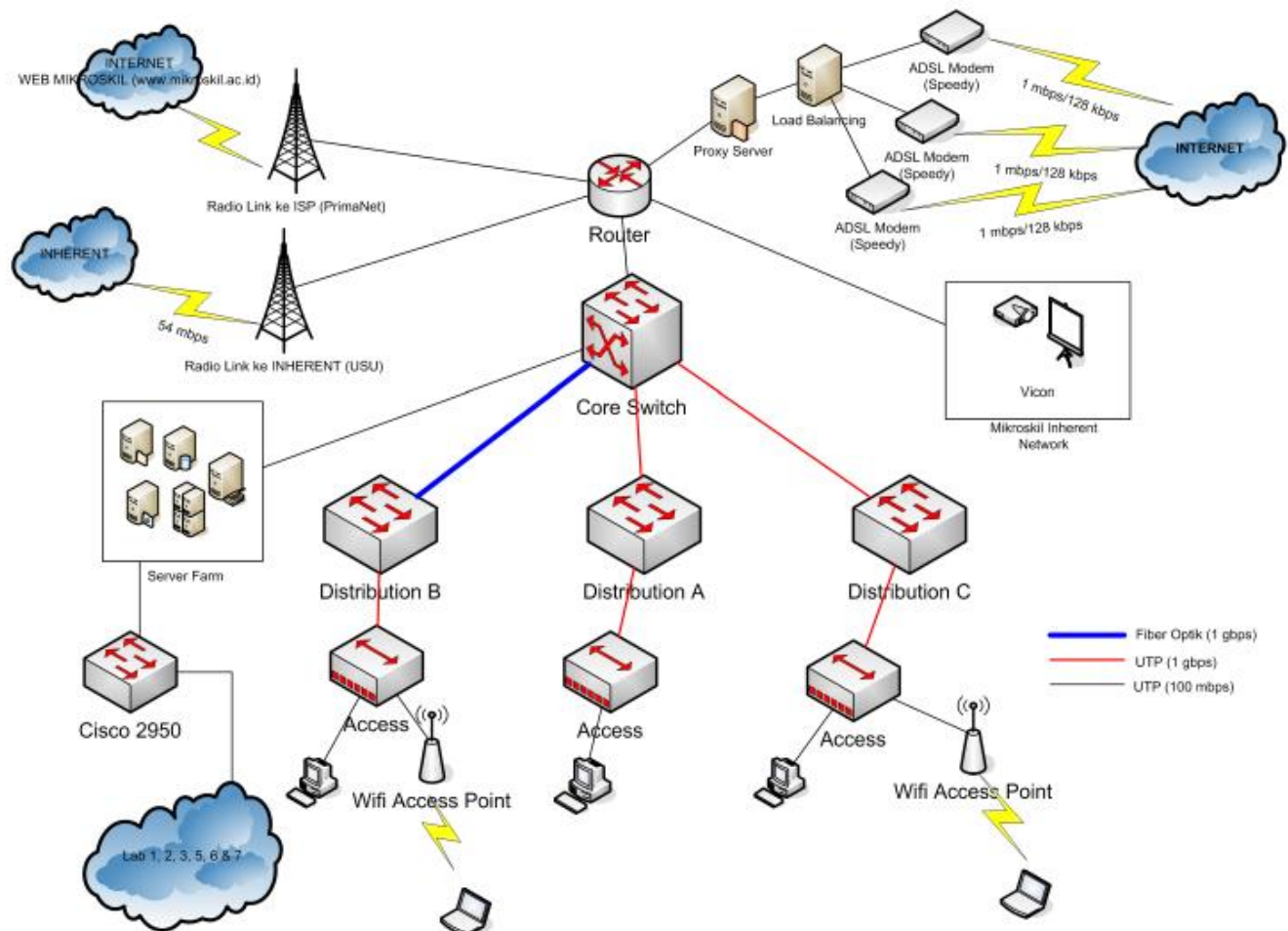
Model Arsitektur Teknologi

Tujuan dari tahapan ini adalah untuk membangun arsitektur teknologi yang diinginkan, dimulai dari penentuan konsep dasar teknologi sampai alternatif teknologi yang diperlukan. Arsitektur teknologi dipandang sebagai pendefinisian platform teknologi yang akan digunakan untuk penyediaan lingkungan aplikasi dalam mengelola data dan sebagai alat dalam mendukung fungsi bisnis yang ada dalam perguruan tinggi. Langkah pertama yang dilakukan adalah penentuan prinsip platform teknologi untuk mendefinisikan prinsip-prinsip teknologi yang akan digunakan. Tujuannya agar pendefinisian teknologi lebih jelas dan terukur maka perlu didefinisikan prinsip platform teknologi yang meliputi perangkat keras, perangkat lunak, dan komunikasi.

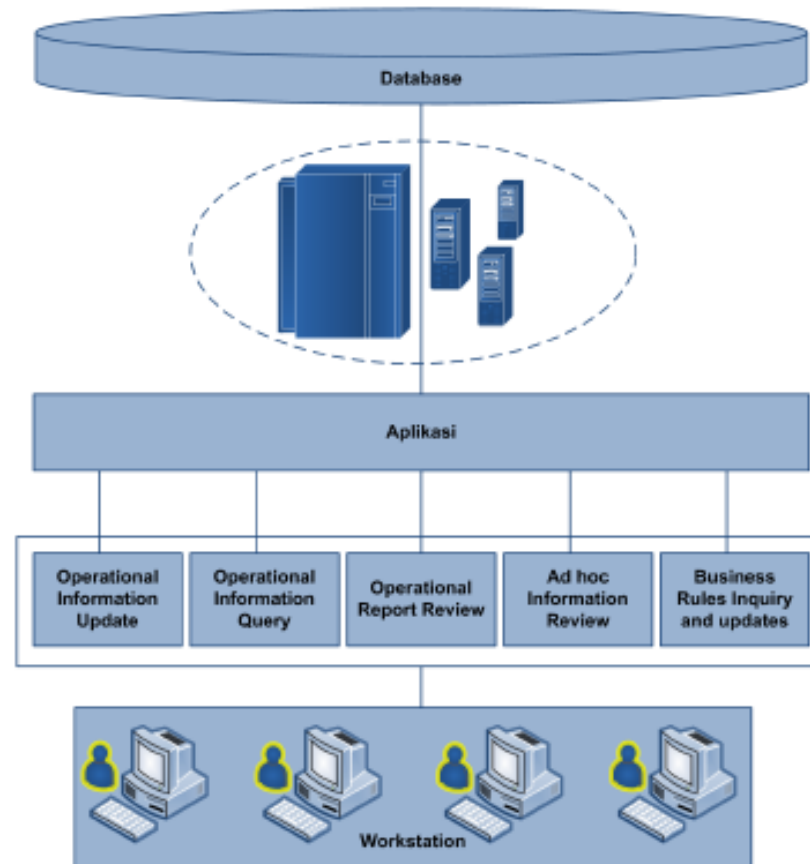
Arsitektur teknologi yang dihasilkan bersifat konseptual sehingga bukan merupakan analisis kebutuhan secara detail melainkan hanya memberikan gambaran umum dan perlu ditinjau ulang saat melakukan implementasi. Arsitektur didefinisikan untuk memastikan bahwa teknologi tersebut reasonable, feasible, dan consistent dengan arsitektur bisnis, data dan aplikasi. Secara umum konfigurasi konseptual teknologi tersebut dapat dibagi atas 2 bagian, yaitu: (1) Jaringan Perguruan Tinggi Konseptual, dan (2) Arsitektur Sistem Bisnis. Masing-masing dari konfigurasi dari konseptual teknologi ini bisa dilihat pada Gambar 8 dan Gambar 9.



Pada prinsip-prinsip teknologi teridentifikasi bahwa teknologi yang diperlukan adalah teknologi jaringan yang menghubungkan suatu bagian dengan bagian lain. Oleh karena itu dalam menentukan platform teknologi, hal yang perlu diperhati-kan adalah lokasi bisnis yang akan menjadi area penempatan infrastruktur teknologi.



Gambar 8: Jaringan Perguruan Tinggi



Gambar 9: Arsitektur Sistem Bisnis

Model arsitektur enterprise yang dikemukakan dalam pertemuan ini adalah model dasar bagi institusi perguruan tinggi didalam pengembangan arsitektur enterprise. TOGAF ADM merupakan metoda pengembangan arsitektur enterprise yang bisa disesuaikan dengan kebutuhan, dengan syarat bahwa institusi mempunyai aturan dan prosedur yang jelas tentang proses bisnis untuk mendukung proses pengembangan sistem informasi terintegrasi.

PRESENTASI TUGAS KELOMPOK

PRESENTASI TUGAS KELOMPOK