BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Beberapa penelitian sebelumnya yang berkaitan dengan penelitian ini, dan dijadikan acuan dalam penelitian ini dituangkan dalam tabel 2.1.

Tabel 2.1 Penelitian Sebelumnya

No	Judul	Deskripsi
1	Perancangan *Enterprise* Arsitektur Sistem Informasi Penjadwalan Menggunakan Kerangka Kerja TOGAF ADM (Studi Kasus : SMK Muhammadiyah 2 Kuningan) Oleh Udin Tahriludin, Magister Sistem Informasi Universitas Komputer Indonesia	Permasalahan dalam penelitian ini adalah masih terdapatnya sumber daya seperti ruang belajar teori dan praktek sebanyak 24 ruangan dari total kebutuhan 34 ruangan, dan kesediaan guru mata pelajaran normatif, adaptif dan produktif yang belum terpenuhi sehinggan proses belajar mengajar tidak berjalan secara optimal. Tahapan TOGAF-ADM yang digunakan untuk mengatasi permasalahan tersebut adalah *architecture vision*, *business architecture*, *system information architecture*, *technology architecture*, *opportunities and solution*, dan *migration planning*. Hasil dari penelitian ini adalah rancangan arsitektur *enterprise* yang dapat dijadikan panduan langkah awal untk melakukan perencanaan *blueprint* perancangan arsitektur *enterprise* sistem informasi penjadwalan.
2	Perancangan *Enterprise Architecture* Sistem Informasi dengan TOGAF ADM 9.1 di CV Cotelligent Indonesia Oleh Raden Sofian Bahri, Magister Sistem Informasi Universitas Komputer Indonesia	Permasalahan dalam penelitian ini adalah perusahaan sudah memiliki sistem-sistem informasi namun tidak ada aliran informasi antar sistem. Selain itu, perusahaan tidak memiliki *blueprint* yang menggambarkan keseluruhan sistem sehingga tidak ada panduan yang bisa digunakan jika terjadi perubahan atau pengembangan. Tahapan TOGAF-ADM yang digunakan untuk mengatasi permasalahan tersebut adalah *architecture vision, business architecture,*

		system information architecture, *technology architecture*, *opportunities and solution*, dan *migration planning*. Hasil dari penelitian ini adalah Sistem yang terpisah yang ada di perusahaan dapat saling berkomunikasi dengan menggunakan webservice yang dibangun menggunakan *Service Oriented Architecture*.
3	Model Arsitektur Bisnis, Sistem Informasi, dan Teknologi di Badan Koordinasi Survey dan Pemetaan Nasional Berbasis TOGAF. Oleh Iyan Supriyana, Badan Koordinasi Survey dan Pemetaan Nasional (BAKORSURTANAL)	Penelitian ini bertujuan untuk menemukan *Enterprise Architecture Framework* yang paling cocok, menyusun rekomendasi model arsitektur bisnis, arsitektur sistem informasi, arsitektur teknologi serta solusi-solusi terbaik yang harus diterapkan dalam pembuatan *blueprint* di Badan Koordinasi Survey dan Pemetaan Nasional. Sedangkan kesimpulan dari penelitian ini yaitu *Enterprise Architecture Framework* yang sesuai untuk BAKORSURTANAL adalah TOGAF.

2.2 *Enterprise*

TOGAF mendefinisikan "*enterprise*" sebagai kumpulan dari organisasi yang memiliki suatu kumpulan tujuan yang sama. Contohnya, suatu enterprise dapat berupa agen pemerintahan, perusahaan secara keseluruhan, suatu divisi dari suatu perusahaan, suatu departemen tunggal, atau suatu rantai dari organisasi yang jauh secara geografis yang dihubungkan bersama dengan kepemilikan yang sama.

Istilah "*enterprise*" dalam konteks "arsitektur *enterprise*" dapat digunakan untuk menggambarkan baik suatu *enterprise* secara keseluruhan - mencakup semua layanan, proses dan infrastruktur informasi dan teknologinya - maupun suatu domain khusus dalam *enterprise*. Dalam kedua kasus tersebut, arsitektur berada dalam beberapa sistem, dan kelompok beberapa fungsional dalam *enterprise*.

(The Open Group : 2011)

2.3 Arsitektur Informasi

Istilah arsitektur telah banyak digunakan dalam konteks teknologi informasi. Istilah ini telah diterapkan untuk sebuah konsep dalam manajemen Teknologi Informasi. Beberapa definisi tentang arsitektur menyatakan sebagai berikut :

1. Dalam TOGAF, "arsitektur" memiliki dua arti yang tergantung pada konteksnya, yaitu:
    a. Deskripsi formal dari suatu sistem, atau rencana rinci dari sistem pada tingkat komponen untuk memandu pelaksanaannya.
    b. Struktur komponen, hubungan antar mereka, dan prinsip-prinsip dan pedoman yang mengatur desain dan evolusi mereka dari waktu ke waktu.

    (The Open Group, 2011)
2. Arsitektur adalah rancangan dari segala jenis struktur, baik fisik maupun konseptual, baik nyata maupun maya (O'Rourke, dkk, 2003).
3. Arsitektur memberikan makna pendekatan yang terencana dan terkontrol, bukan reaktif (Cook, 1996).

Sedangkan arsitektur informasi adalah sebuah sistem, yang mengelola data serta penerapan dari proses bisnis yang telah didefinisikan, sehingga sebelum organisasi mendefinisikan kebutuhan informasi yang akan digunakan untuk menjalankan roda organisasinya, maka terlebih dahulu harus mendefinisikan data, proses bisnis dan sistem aplikasinya. (*IBM Corp. 1981*).

Beberapa definisi lain Arsitektur Sistem Informasi :

1. Pemetaan atau rencana kebutuhan-kebutuhan informasi di dalam suatu organisasi (Turban, McLean, Wtherbe, 1999).
2. Bentuk khusus yang menggunakan teknologi informasi dalam organisasi untuk mencapai tujuan-tujuan atau fungsi-fungsi yang telah dipilih (Laudon & Laudon, 1998).
3. Desain sistem komputer secara keseluruhan (termasuk sistem jaringan) untuk memenuhi kebutuhan-kebutuhan organisasi yang spesifik (Zwass, 1998).

2.4 Arsitektur *Enterprise*

Arsitektur *enterprise* merupakan salah satu disiplin ilmu dalam Teknologi Informasi yang memiliki definisi sebagai berikut :

1. Deskripsi misi para *stakeholder* mencakup parameter informasi, fungsionalitas, lokasi, organisasi, dan kinerja. Arsitektur *enterprise* menjelaskan rencana untuk membangun sistem atau sekumpulan sistem.
2. Pendekatan logis, komprehensif, dan holistik untuk merancang dan mengimplementasikan sistem dan komponen sistem yang bersama.
3. Basis aset informasi strategis, yang menentukan misi, informasi, dan teknologi yang dibutuhkan untuk melaksanakan misi, dan proses transisi untuk mengimplementasikan teknologi baru sebagai tanggapan terhadap perubahan kebutuhan misi.
4. Arsitektur *enterprise* memiliki empat komponen utama yaitu arsitektur bisnis, arsitektur informasi (data), arsitektur teknologi, dan arsitektur aplikasi.

5. Sehubungan dengan keempat komponen ini, produk *Enterprise Architecture* adalah berupa grafik, model, dan/atau narasi yang menjelaskan lingkungan dan rancangan *enterprise*.

(Erwin, 2009)

2.5 *Framework* Arsitektur *Enterprise*

Framework arsitektur *enterprise* mengidentifikasikan jenis informasi yang dibutuhkan untuk mendeskripsikan arsitektur *enterprise*, mengorganisasikan jenis informasi dalam struktur logis, dan mendeskripsikan hubungan antara jenis informasi tersebut. Informasi dalam arsitektur *enterprise* sering dikategorikan dalam model-model atau sudut pandang arsitektural. Dalam mengembangkan arsitektur *enterprise*, perlu diadopsi atau dikembangkan sendiri suatu *framework* untuk arsitektur *enterprise*. Terdapat berbagai macam *framework* yang dapat dimanfaatkan untuk pengembangan arsitektur *enterprise*, seperti *Zachman Framework, Federal Enterprise Architecture Framework* (*FEAF*), *DoD Architecture Framework* (*DoDAF*), *Treasury Enterprise Architecture Framework* (*TEAF*), *The Open Group Architecture Framework* (*TOGAF*), dan lain-lain.

2.5.1 Zachman *Framework*

Salah satu *framework* untuk pengembangan arsitektur *enterprise* adalah *framework* yang diperkenalkan oleh Zachman atau disebut dengan *framework* zachman. *Framework* Zachman merupakan suatu alat bantu yang dikembangkan untuk memotret arsitektur organisasi dari berbagai sudut pandang dan aspek,

sehingga didapatkan gambaran organisasi secara utuh. *Framework* Zachman untuk arsitektur *enterprise* dapat dijelaskan sebagai berikut :

1. **Perencana (*Planner*)** : yang menetapkan objek dalam pembahasan; latar belakang, lingkup, dan tujuan *enterprise*.
2. **Pemilik (*owner*)** : penerima atau pemakai produk/jasa akhir dari *enterprise*.
3. **Perancang (*Designer*)** : perantara antara apa yang diinginkan (pemilik) dan apa yang dapat dicapai secara teknis dan fisik.
4. **Pembangun (*Builder*)** : pengawas/pengatur dalam menghasilkan produk/jasa akhir.
5. **Subkontraktor** : bertanggungjawab membangun dan merakit bagian-bagian dari produk/jasa akhir.
6. ***Functioning enterprise*** : wujud nyata dari produk/jasa akhir.

Karakteristik Zachman *Framework* :

1. Mengkategorikan *deliverables* dari Arsitektur *Enterprise*.
2. Kegunaan Arsitektur *Enterprise* yang terbatas.
3. Banyak diadopsi di seluruh dunia.
4. Perspektif *view* yang kurang menyeluruh.
5. Merupakan *tool* untuk perencanaan.

Kelebihan Zachman *Framework* :

1. Merupakan standar secara *de-facto* untuk mengklasifikasikan artefak arsitektur *enterprise*.

2. Struktur logis untuk analisis dan presentasi artefak dari suatu perspektif manajemen.
3. Menggambarkan secara paralel baik sisi *engineering* yang sudah sangat dimengerti maupun paradigma konstruksi.
4. Dikenal secara luas sebagai tools manajemen untuk memeriksa kelengkapan arsitektur dan *maturity level*.

Kekurangan Zachman *Framework* :

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

2.5.2 *Federal Enterprise Architecture Framework* (*FEAF*)

Federal Enterprise Architecture Framework (*FEAF*) merupakan sebuah *framework* yang diperkenalkan pada tahun 1999 oleh Federal CIO Council. FEAF ini ditujukan untuk mengembangkan *Enterprise Architecture* dalam Federal Agency atau sistem yang melewati batas *multiple inter-agency*. FEAF menyediakan standar untuk mengembangkan dan mendokumentasikan deskripsi arsitektur pada area yang menjadi prioritas utama. FEAF ini cocok untuk mendeskripsikan arsitektur bagi pemerintahan Federal. FEAF membagi arsitektur menjadi area bisnis, data, aplikasi dan teknologi, dimana sekarang FEAF juga mengadopsi tiga kolom pertama pada Zachman *framework* dan metodologi perencanaan *Enterprise*

Architecture oleh Spewak. FEAF menyediakan sebuah struktur untuk mengembangkan, memelihara dan mengimplementasikan lingkungan operasional pada top-level dan mendukung implementasi dari sistem Teknologi Informasi. Pada gambar 2.1 menunjukkan gambaran matriks 5 x 3 FEAF dengan tipe-tipe arsitektur pada sumbu mendatar dan perspektif pada sumbu lainnya. Hubungan antara produk Arsitektur *Enterprise* terdapat pada cells matriks.

	Data Architecture	Application Architecture	Technology Architecture
Planner Perspective	List of Bussiness Objects	List of Bussiness Process	List of Business Locations
Owner Perspective	Semantic Model	Business Process Model	Business Logistics System
Designer Perspective	Logical Data Model	Application Architecture	System Geographic Deployment Architecture
Builder Perspective	Physical Data Model	System Design	Technology Architecture
Subcontractor Perspective	Data Dictionary	Programs	Network Architecture

Gambar 2. 1 Matrik Arsitektur FEAF (Erwin, 2009)

Karakteristik dari FEAF :

1. Merupakan Arsitektur *Enterprise Reference Model*.
2. Standar yang dipakai oleh pemerintahan Amerika Serikat Menampilkan perspektif *view* yang menyeluruh.
3. Merupakan *tools* untuk perencanaan dan komunikasi.

2.5.3 *The Open Group Architecture Framework (TOGAF)*

The Open Group Architecture Framework (TOGAF) merupakan *framework* dan metode yang diterima secara luas dalam pengembangan arsitektur perusahaan. Berawal dari *Technical Architecture for Information Management* (*TAFTM*) di Departemen Pertahanan Amerika Serikat, *framework* itu diadopsi oleh Open Group pada pertengahan tahun 1990-an. Spesifikasi pertama TOGAF diperkenalkan pada tahun 1995. TOGAF merupakan hasil pengembangan forum Open Group yang

merupakan forum kerjasama antara vendor dan pengguna. TOGAF ini digunakan untuk mengembangkan arsitektur *enterprise* yang memiliki metode dan *tools* yang detil untuk mengimplementasikannya. Hal inilah yang membedakan dengan *framework* arsitektur *enterprise* lain misalnya *framework* Zachman. Salah satu kelebihan menggunakan *framework* TOGAF ini adalah karena sifatnya yang fleksibel dan bersifat *open source*. TOGAF memandang arsitektur *enterprise* menjadi empat kategori, yaitu :

1. Arsitektur bisnis, mendeskripsikan tentang bagaimana proses bisnis untuk mencapai tujuan organisasi.
2. Arsitektur aplikasi, merupakan pendeskripsian bagaimana aplikasi tertentu didesain dan bagaimana interaksinya dengan aplikasi lainnya.
3. Arsitektur data, menggambarkan bagaimana penyimpanan, pengelolaan dan pengaksesan data pada perusahaan.
4. Arsitektur teknologi, menggambarkan mengenai infrastruktur hardware dan software yang mendukung aplikasi dan bagaimana interaksinya.

2.5.3.1 Struktur Umum dan Komponen TOGAF

TOGAF secara umum memiliki struktur dan komponen sebagai berikut (Gambar 2.2).

1. ***Architecture Development Method* (ADM)**

   Merupakan bagian utama dari TOGAF yang memberikan rincian bagaimana menentukan sebuah arsitektur *enterprise* secara spesifik berdasarkan kebutuhan bisnisnya.

2. ***Foundation Architecture (Enterprise Continuum)***

   Foundation Architecture merupakan sebuah "*framework-within-a-framework*", dimana di dalamnya tersedia gambaran hubungan untuk pengumpulan arsitektur yang relevan, juga menyediakan bantuan petunjuk pada saat terjadinya perpindahan abstraksi level yang berbeda. *Foundation Architecture* dapat dikumpulkan melalui ADM. Terdapat tiga bagian pada *foundation architecture* yaitu *Technical Reference Model*, *Standard Information* dan *Building Block Information Base*.

3. ***Resource Base***

   Pada bagian ini terdapat informasi mengenai *guidelines*, *templates*, *checklists*, latar belakang informasi dan detil material pendukung yang membantu arsitek di dalam penggunaan ADM.

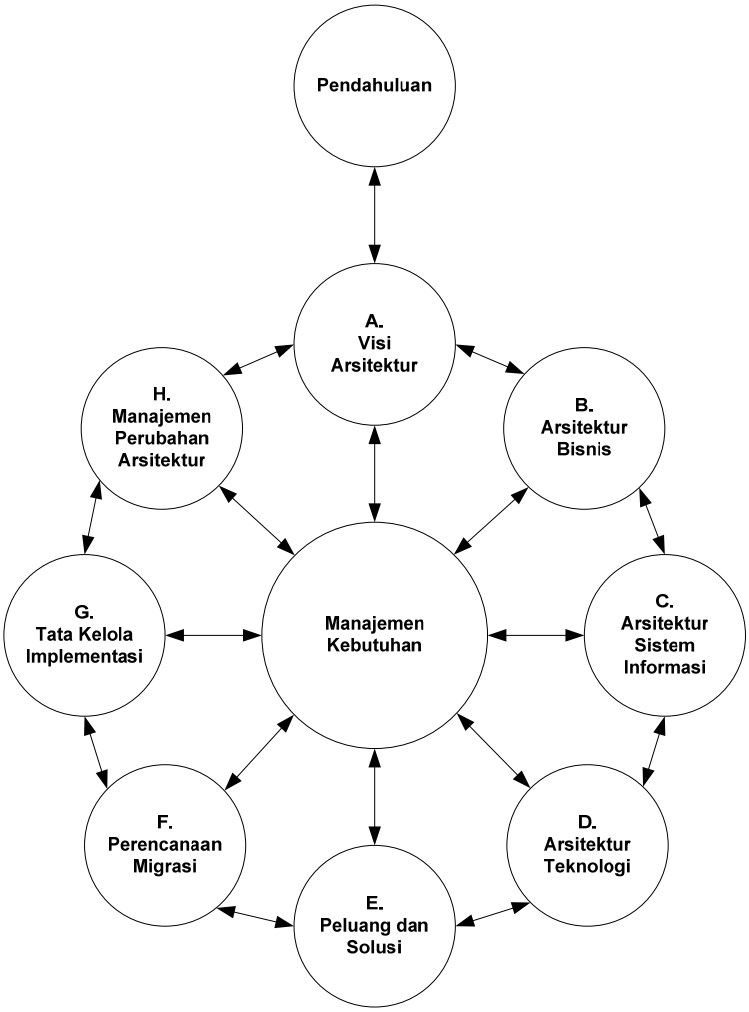


Gambar 2.2 Struktur Umum TOGAF (Erwin, 2009)

2.5.3.2 *Architecture Development Method (ADM)*

TOGAF memberikan metode yang detail mengenai bagaimana membangun, mengelola dan mengimplementasikan arsitektur *enterprise* dan sistem

informasi yang disebut dengan *Architecture Development Method* (ADM), dimana ADM merupakan hasil dari kerjasama praktisi arsitektur dalam Open Group Architecture Forum. ADM merupakan metode generik yang berisikan sekumpulan aktifitas yang mempresentasikan kemajuan dari setiap fase ADM dan model arsitektur yang digunakan dan dibuat selama tahap pengembangan arsitektur *enterprise*. Inti dari ADM adalah pengelolaan kebutuhan, dimana kebutuhan bisnis, sistem informasi, dan arsitektur teknologi selalu diselaraskan dengan sasaran dan kebutuhan bisnis. Gambar 2.3 menunjukan tahapan-tahapan proses pemodelan arsitektur dalam TOGAF ADM.



Gambar 2.3 *TOGAF Architecture Development Method (ADM)*

Langkah-langkah dalam TOGAF ADM adalah sebagai berikut :

1. Pendahuluan

   Menjelaskan aktivitas persiapan dan awal yang dibutuhkan untuk membuat suatu kemampuan arsitektur termasuk kustomisasi dari TOGAF dan definisi dari prinsip-prinsip arsitektur.

2. Manajemen Kebutuhan

   Menguji proses dari pengelolaan kebutuhan arsitektur melalui ADM.

3. Tahap A : Visi Arsitektur (*Architecture Vision*)

   Menciptakan kesamaan padangan mengenai pentingnya arsitektur *enterprise* untuk mencapai tujuan organisasi/perusahaan dan menentukan lingkup dari arsitektur yang akan dikembangkan. Berisikan pertanyaan-pertanyaan yang harus dijawab seperti :

   a. Berapa banyak informasi yang akan diambil;
   b. Bagaimana mengelola informasi tersebut;
   c. Bagaimana organisasi memulai proses pemodelan *enterprise*;
   d. Apakah cukup jika hanya sebagian *enterprise* saja yang ditinjau;
   e. Apakah model arsitektur yang ada dapat digunakan kembali.

4. Tahap B : Arsitektur Bisnis (*Business Architecture*)

   Mendefinisikan kondisi awal arsitektur bisnis, menentukan *Business Art* yang diinginkan, melakukan analisis kesenjangan antara keduanya dan penentuan *tools* serta teknik yang digunakan. Pada tahap ini *tools* dan metode umum seperti *Business Process Model and Notation* (BPMN), *Integrated Definition*

(IDEF) *Modeling Technique*, dan *Unified Modeling Language* (UML) dapat digunakan untuk mengembangkan model yang diperlukan.

5. Tahap C : Arsitektur Sistem Informasi (*Information System Architecture*)

   Membangun arsitektur sistem informasi yang diinginkan, arsitektur ini meliputi 2 (dua) domain yaitu arsitektur data dan arsitektur aplikasi. Arsitektur data lebih memfokuskan pada bagaimana data digunakan untuk kebutuhan fungsi bisnis, proses dan layanan. Teknik yang bisa digunakan yaitu dengan : *ER-Diagram*, *Class Diagram*, dan *Object Diagram*. Pada arsitektur aplikasi lebih menekankan pada bagaimana kebutuhan aplikasi direncanakan dalam mendukung bisnis, serta lebih fokus pada model aplikasi yang akan dirancang. Teknik yang bisa digunakan meliputi : *Application and User Location Diagram*, *Use Case Diagram* dan lainnya.

6. Tahap D : Arsitektur Teknologi (*Technology Architecture*)

   Membangun arsitektur teknologi yang diinginkan, dimulai dari penentuan dasar, alternatif teknologi sampai pelaksanaan analisis kesenjangan. Teknologi direpresentasikan dengan *framework*-nya tersendiri, dengan penjelasan detil penggunaan teknologi dalam organisasi. Dalam tahapan ini juga mempertimbangkan alternatif-alternatif yang diperlukan dalam pemilihan teknologi. Teknik yang digunakan meliputi *Environment and Location Diagram*, *Network Computing Diagram*, dan lainnya.

7. Tahap E : Peluang dan Solusi (*Opportunities and Solution*)

   Mengevaluasi dan memilih alternatif implementasi, identifikasi parameter strategis penilaian keterkaitan, biaya dan manfaat, mendefinisikan strategi

implementasi dan rencana implementasi. Pada tahapan ini lebih menekankan pada manfaat yang diperoleh dari arsitektur *enterprise* yang meliputi arsitektur bisnis, arsitektur data, arsitektur aplikasi dan arsitektur teknologi, sehingga menjadi dasar bagi *stakeholder* untuk memilih dan menentukan arsitektur yang akan diimplementasikan. Untuk memodelkan tahapan ini dalam rancangan bisa menggunakan teknik *Project Context Diagram* dan *Benefit Diagram*.

8. Tahap F : Perencanaan Migrasi (*Migration Planning*)

   Menyusun urutan proyek-proyek berdasarkan prioritas termasuk penilaian kebergantungan, biaya, dan manfaat dari proyek migrasi. Urutan prioritas akan menjadi dasar implementasi proyek. Biasanya pada tahapan ini untuk pemodelannya menggunakan matrik penilaian dan keputusan terhadap kebutuhan utama dan pendukung dalam organisasi terhadap implementasi sistem informasi.

9. Tahap G : Tata Kelola Implementasi (*Implementation Governance*)

   Menyusun rekomendasi untuk setiap implementasi proyek; menyusun kontrak arsitektur dan melaksanakan keseluruhan proses implementasi, menetapkan organisasi pelaksana untuk proses implementasi sistem, memastikan kesesuaian pelaksanaan proyek dengan arsitektur yang dikehendaki. Menyusun rekomendasi untuk pemetaan dari tahapan ini bisa juga dipadukan dengan *framework* yang digunakan untuk tata kelola seperti COBITS dari IT Governance Institute (ITGI) (Open Group, 2009).

10. Tahap H : Manajemen Perubahan Aristektur (*Architecture Change Management*).

Menetapkan proses manajemen perubahan arsitektur untuk arsitektur *enterprise* baru yang telah selesai diimplementasikan; secara berkelanjutan memonitor perkembangan teknologi dan perubahan lingkungan organisasi dan menentukan apakah akan dilakukan siklus pengembangan arsitektur *enterprise* berikutnya.

Prinsip pengembangan arsitektur *enterprise* dengan menggunakan TOGAF-ADM terdiri dari tiga bagian, yaitu :

1. Prinsip-prinsip *enterprise*, mendukung keputusan bisnis di seluruh bagian organisasi/perusahaan.
2. Prinsip-prinsip teknologi informasi, mengarahkan penggunaan sumber daya teknologi informasi di seluruh bagian organisasi/perusahaan.
3. Prinsip-prinsip arsitektur, mengembangkan arsitektur proses organisasi/perusahaan dan arsitektur implementasinya. Prinsip ini dipengaruhi oleh rencana organisasi/perusahaan, strategi, faktor pasar, sistem, dan teknologi yang ada dalam organisasi/perusahaan.

Kelebihan TOGAF

1. Fokus pada siklus implementasi ADM (*Architecture Development Method*).
2. Kaya akan arena teknis arsitektur.
3. *Resource Base* menyediakan banyak material referensi.

Kekurangan TOGAF

1. Tiga layer teratas masih perlu diperkuat

2. Tidak ada template standar untuk seluruh domain (misalnya untuk membuat blok diagram).
3. Tidak ada artefak yang dapat digunakan ulang (*ready made*).

(Surendro, Kridanto, Pengembangan Perencaaan Induk Sistem Informasi, Informatika, 2009).

2.5.4 Pemilihan Arsitektur *Enterprise Framework*

Untuk memilih sebuah arsitektur *enterprise framework* terdapat kriteria yang bermacam-macam yang bisa dijadikan sebagai acuan (Erwin, 2009), yaitu:

1. Tujuan dari arsitektur *enterprise* dengan melihat bagaimana definisi arsitektur dan pemahamannya, proses arsitektur yang telah ditentukan sehingga mudah untuk diikuti, serta dukungan terhadap evolusi arsitektur.
2. *Input* untuk aktivitas arsitektur *enterprise* seperti pendorong bisnis dan *input* teknologi.
3. *Output* dari aktivitas arsitektur *enterprise* seperti model bisnis dan desain transisional untuk evolusi dan perubahan.

Framework merupakan sebuah bagian penting dalam pendesainan arsitektur *enterprise* yang seharusnya memiliki kriteria:

1. *Reasoned*

   Framework yang masuk akal yang dapat memungkinkan pembuatan arsitektur yang bersifat *deterministic* ketika terjadi perubahan batasan dan tetap menjaga

integritasnya walaupun menghadapi perubahan bisnis dan teknologi serta *demand* yang tak terduga.

2. *Cohesive*

   Framework yang kohesif memiliki sekumpulan perilaku yang akan seimbang dalam cara pandang dan ruang lingkupnya.

3. *Adaptable*

   Framework haruslah bisa beradaptasi terhadap perubahan yang mungkin sangat sering terjadi dalam organisasi.

4. *Vendor-independent*

   Framework haruslah tidak tergantung pada vendor tertentu untuk benar-benar memaksimalkan *benefit* bagi organisasi.

5. *Technology-independent*

   Framework haruslah tidak tergantung pada teknologi yang ada saat ini, tapi dapat menyesuaikan dengan teknologi baru.

6. *Domain-neutral*

   Adalah atribut penting bagi *framework* agar memiliki peranan dalam pemeliharaan tujuan organisasi.

7. *Scalable*

   Framework haruslah beroperasi secara efektif pada level departemen, unit bisnis, pemerintahan, level korporat tanpa kehilangan fokus dan kemampuan untuk dapat diaplikasikan.

Perbandingan ketiga *framework* yang banyak digunakan dapat dilihat pada Tabel 2.2. Dalam prakteknya *Enterprise Architecture framework* yang ada tidak ada yang sempurna, masing-masing memiliki kelebihan dan kekurangan. Bahkan penggunaan *Enterprise Architecture framework* di masing-masing *enterprise* bisa menjadi berbeda. Hal ini tergantung dengan karakteristik dari *enterprise* itu sendiri, fokus yang ingin dicapai dan lain-lain.

Tabel 2.2 Perbandingan *Enterprise Architecture Framework*

	FEAF	**Zachman**	**TOGAF**
Definisi arsitektur dan pemahamannya	Ada	Parsial	Pada fase *preliminary*
Proses arsitektur yang detail	Tidak	Ada	Delapan fase detail pada ADM
Support terhadap evolusi arsitektur	Ada	Tidak	Pada fase *migration planning*
Standarisasi	Tidak	Tidak	Ada
Architecture Knowladge Base	Ada	Tidak	Ada
Pendorong Bisnis	Ada	Parsial	Ada
Input teknologi	Ada	Tidak	Ada
Desain tradisional	Ada	Tidak	Pada fase *Migration Planning*
Model bisnis	Ada	Ada	Ada
Menyediakan prinsip arsitektur	Hanya untuk karakteristik FEAF	Tidak	Ada

(Sumber : Erwin, 2009)

Dari hasil pemetaan kriteria tersebut dapat ditarik kesimpulan untuk studi kasus *enterprise* dimana masih belum terdapat arsitektur *enterprise* dan memiliki keperluan untuk pengembangan arsitektur *enterprise* yang mudah dan jelas serta sesuai maka arsitektur *enterprise framework* yang cocok digunakan TOGAF.

(Erwin, 2009)

2.6 Perangkat-perangkat untuk Memodelkan Fase-fase dalam TOGAF

2.6.1 *Value Chain*

Model rantai nilai (*value chain)* menekankan aktivitas khusus pada bisnis dimana strategi kompetitif dapat diterapkan dengan paling baik (Porter, 1985) dan dimana sistem informasi paling mungkin memiliki dampak strategis. Model ini mengenali titik khusus, dan kritis dimana perusahaan dapat menggunakan teknologi informasi paling efektif untuk menggunakan posisi kompetitifnya. Model rantai nilai melihat perusahaan sebagai serangkaian atau rantai aktivitas dasar yang menambahkan margin nilai kepada produk dan jasa perusahaan. Aktivitas ini dapat digolongkan baik sebagai aktivitas utama maupun aktivitas pendukung.

Aktivitas utama (*primary activities)* paling terkait secara langsung dengan produksi dan distribusi produk dan jasa perusahaan, yang menciptakan nilai bagi pelanggan. Aktivitas utama termasuk logistik dari dalam, operasi, logistik dari luar, penjualan dan pemasaran, dan jasa. Logistik masuk termasuk menerima dan menyimpan bahan baku untuk distribusi sampai produksi. Operasi mengubah input menjadi barang jadi. Logistik keluar termasuk menyimpan dan mendistribusikan barang jadi. Penjualan dan pemasaran termasuk mempromosikan dan menjual produk perusahaan. Aktivitas pelayanan termasuk pemeliharaan dan perbaikan barang jasa dan jasa perusahaan.

Aktivitas pendukung (*support activities*) membuat pengiriman aktivitas utama dapat terjadi dan terdiri atas infrastruktur organisasi (administrasi dan manajemen), SDM (perekrutan, penempatan, dan pelatihan karyawan), teknologi (meningkatkan produk dan proses produk), dan pembelian (membeli input).



Gambar 2.4 *Value Chain* (Laudon & Laudon : 1998)

2.6.2 *Business Process Modeling Notation* (BPMN)

BPMN adalah singkatan dari *Business Process Modeling Notation*, yaitu suatu metodologi baru yang dikembangkan oleh *Business Process Modeling Initiative* sebagai suatu standar baru pada pemodelan proses bisnis, dan juga sebagai alat desain pada sistem yang kompleks seperti sistem *e-Business* yang berbasis pesan (*message-based*). Tujuan utama dari BPMN adalah menyediakan notasi yang mudah digunakan dan bisa dimengerti oleh semua orang yang terlibat dalam bisnis, yang meliputi bisnis analis yang memodelkan proses bisnis, pengembang teknik yang membangun sistem yang melaksanakan bisnis, dan berbagai tingkatan manajemen yang harus dapat membaca dan memahami proses diagram dengan cepat sehingga dapat membantu dalam pengambilan keputusan.

Notasi BPMN yang baru juga dirancang untuk sifat sistem berbasis layanan web. BPMN dapat memodelkan pesan kompleks yang dilewatkan diantara pelaku bisnis atau bagian dari pelaku bisnis, kejadian yang menyebabkan pesan dilewatkan, dan aturan bisnis yang membatasi kejadian tersebut. BPMN memugkinkan proses bisnis dipetakan ke bahasa eksekusi bisnis berbasis XML

seperti BPEL4WS (*Business Process Execution Language for Web Service*) dan BPML (*Business Process Modeling Language*). Informasi pada bahasa eksekusi bisnis ini dapat divisualisasikan dengan notasi umum.

Salah satu kelebihan diagram BPMN adalah kemampuan memodelkan aliran pesan. Diagram bisnis proses tradisional mampu memodelkan aliran proses secara sekuensial, dari kejadian awal sampai hasil akhir. Dalam lingkungan *e-commerce*, tentunya orang mengirim pesan kepada yang lain sebagai bagian dari aliran proses. (Rosmala, 2007)

Terdapat 4 kategori dari elemen-elemen dalam BPMN, yaitu:

1. *Flow Objects*
   a. *Events*, sebuah *event* direpresentasikan dengan lingkaran. *Events* dapat berupa Start, Intermediate, atau End.



   b. *Activities*, sebuah aktivitas direpresentasikan dengan persegi dengan sudut melingkar dan memperlihatkan pekerjaan yang harus dilakukan.

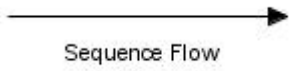


   c. *Gateways*, sebuah *gateway* direpresentasikan dengan belah ketupat dan memperlihatkan pilihan yang berbeda. *Gateway* juga menjelaskan mengenai percabangan dan penggabungan dari path yang ada.

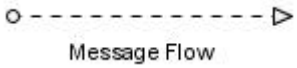


2. *Connecting Objects*

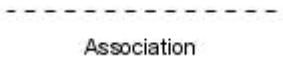
   a. *Sequence Flow, sequence flow* direpresentasikan dengan garis lurus dengan panah tertutup dan menjelaskan mengenai urutan aktivitas yang akan dijalankan.



   b. *Message Flow, message flow* direpresentasikan dengan garis putus-putus dan panah terbuka. *Message flow* menjelaskan pertukaran pesan yang sedang terjadi.



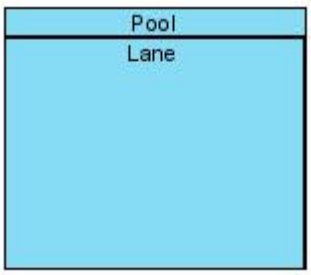
   c. *Association, association* direpresentasikan dengan garis putus-putus. *Association* digunakan untuk mengasosiasikan sebah artifak, data, maupun *flow object*.



3. *Swimlanes*

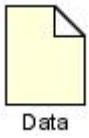
   a. *Pool, pool* direpresentasikan dengan persegi besar yang didalamnya dapat berisi *flow objects, connecting object,* maupun artifak.

   b. *Lane, lane* merupakan bagian lebih mendetail dari *pool*.



4. *Artifacts*

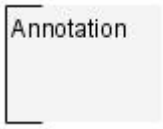
   a. *Data Objects*, *data object* digunakan untuk menjelaskan mengenai data yang dibutuhkan atau dihasilkan dari sebuah aktivitas.



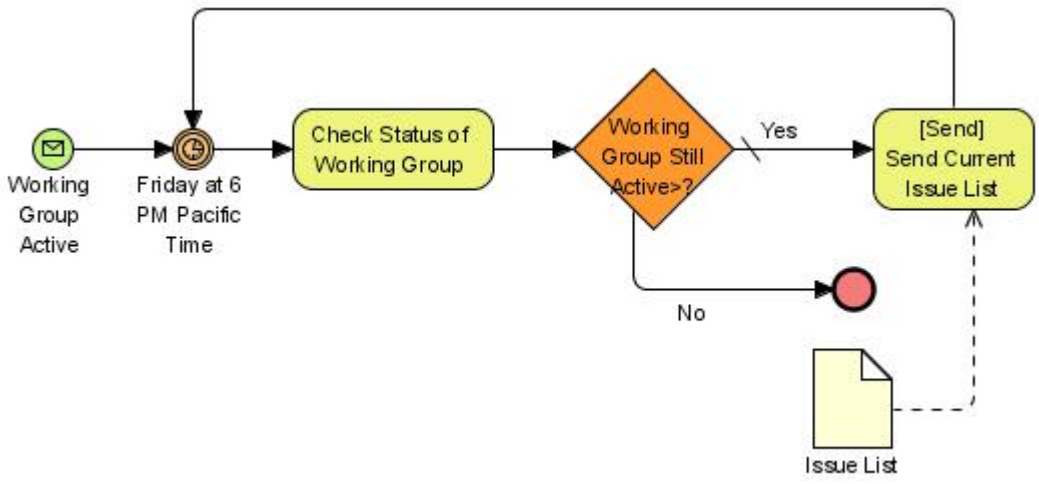
   b. *Group*, *group* direpresentasikan dalam persegi dengan sudut melingkar dan garis luar putus-putus. *Group* untuk melakukan grouping aktivitas.



   c. *Annotation*, *annotation* digunakan untuk menjelaskan model atau diagram.



Business Process dapat digambarkan dalam BPMN secara mudah. Sebagai contoh dapat dilihat pada gambar business process sederhana sebagai berikut:



2.6.3 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya yang dibangun menggunakan program berorientasi objek.

Bahasa pemodelan grafis telah ada di industri perangkat lunak sejak lama, pemicu utama dibalik semuanya adalah bahwa bahasa pemograman berada pada tingkat abstraksi yang tidak terlalu tinggi untuk memfasilitasi diskusi tentang desain.

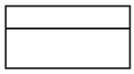
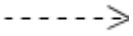
UML lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak kehadirannya pada tahun 1997. (Fowler, 2007)

2.6.3.1 *Class Diagram*

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis. *Class diagram* juga menunjukkan *property* dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. Kotak-kotak yang terdapat di dalam diagram merupakan *class*, yang dibagi menjadi tiga bagian: nama *class*, atributnya, dan operasinya. (Fowler, 2007)

Berikut ini adalah tabel 2.3 yang menerangkan simbol-simbol yang dipakai pada *class diagram*.

Tabel 2.3 Daftar Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.6.3.2 *Use Case* Diagram

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi bagaimana sistem tersebut digunakan.

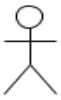
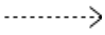
Setiap *Use Case* memiliki aktor utama yang meminta sistem untuk memberi sebuah layanan. Aktor utama adalah aktor dengan tujuan yang akan dipenuhi *Use*

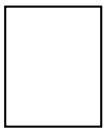
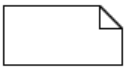
Case, tetapi tidak selalu. Selain itu terdapat banyak aktor lain yang berkomunikasi dengan sistem pada saat menjalankan *Use Case*.

Setiap langkah dalam *Use Case* adalah sebuah elemen dalam interaksi antar aktor dan sistem. Setiap langkah harus berupa pernyataan sederhana dan dengan jelas menunjukkan siapa yang menjalankan langkah-langkah tersebut. Langkah tersebut harus menunjukkan tujuan aktor, bukan mekanisme yang harus dilakukan aktor. (Fowler, 2007)

Berikut ini adalah tabel yang menerangkan simbol-simbol yang dipakai pada *class diagram*.

Tabel 2.4 Daftar Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.

6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi