BAB II
TINJAUAN PUSTAKA

2.1 Enterprise

Enterprise adalah sekumpulan korporat atau entitas atau lembaga dengan tugas yang mendukung entitas fungsional dan memiliki sekumpulan tujuan atau suatu perintah. Dalam konteks ini, suatu enterprise adalah divisi atau departemen dari suatu perusahaan, sekelompok organisasi yang terpisah secara geografis namun dihubungkan bersama oleh kepemilikan secara administratif, agen pemerintah (atau sekumpulan agen) pada setiap tingkat yuridiksi, sekumpulan agen pemerintah, dan seterusnya.

Organisasi besar dan entitas pemerintah dapat terdiri dari beberapa enterprise. Suatu arsitektur mendukung penyatuan operasi multi organisasional. Framework arsitektural dapat memberikan dasar pengembangan repositori arsitektur untuk penyatuan dan penggunaan kembali model, rancangan data dasar.(Minoli, 2008)

2.2 *Architechture*

Berdasarkan American National Standards Institute/*Institute of Electrical and Electronics Engineers* (ANSI/IEEE) Std 1471-2000, suatu arsitektur adalah dasar organisasi suatu sistem, diwujudkan di dalam komponennya, hubungannya terhadap komponen lain dan lingkungannya, dan prinsip mengatur rancangan dan evolusi sistem tersebut. Arsitektur dapat digambarkan sebagai *blueprint* IT/korporat untuk perencanaan suatu kota atau pembangunan yang besar. (Minoli, 2008) Arsitektur mencakup bagian berikut ini:

1. Arsitektur bisnis: suatu perumusan arsitektural dari fungsi bisnis

2. Arisktektur informasi: suatu perumusan arsitektural dari informasi fungsi melalui model data.
3. Arsitektur sistem/aplikasi: defenisi arsitektural dari sistem/aplikasi fungsi.
4. Arsitektur teknologi infrastruktur : perumusan (penjelasan) arsitektural dari infrastruktur teknologi fungsi.

2.3 *Enterprise Architecture* (EA)

Enterprise Architecture (EA) adalah kumpulan proses bisnis, aplikasi, teknologi, dan data yang mendukung strategi bisnis suatu enterprise. Arsitektur enterprise adalah suatu rencana perekaman, cetak biru dari struktur, susunan, pengaturan, pengelompokan fungsional, antar muka, data, protokol, logika fungsional, integrasi, teknologi, dari sumber IT dan dibutuhkan untuk mendukung fungsi bisnis atau misi korporat atau organisasi.

Tujuan dari EA adalah membuat lingkungan IT yang bersatu (sistem *software* dan *hardware* yang terstandarisasi) melintasi perusahaan atau semua unit bisnis perusahaan, yang dihubungkan kepada sisi bisnis dan strategi organisasi. Secara spesifik, tujuannya adalah untuk mempertimbangkan penjajaran, standarisasi, penggunaan kembali aset IT, dan berbagi metode yang sudah ada untuk pengelolaan proyek dan pembangunan *software* antar organisasi. Hasil akhirnya adalah membuat IT lebih murah, strategis, dan responsif.

Maksud dari EA adalah membuat pemetaan aset IT dan proses bisnis, dan sekumpulan prinsip tata kelola yang mengendalikan strategi bisnis dan bagaimana hal tersebut dapat dijelaskan melalui IT. Hasil dari EA dapat berupa sekumpulan

strategi IT, EA yang baru atau dimodifikasi, standar EA yang baru atau dimodifikasi, *roadmap* yang menjelaskan proyek IT yang dibutuhkan untuk mengimplementasikan arisktektur baru dan mencapai target, dan suatu rencana pengembangan/penyebaran (Minoli, 2008).

2.4 *The Open Group Architecture Framework* (TOGAF)

TOGAF adalah suatu *framework* (rincian metode dan sekumpulan *tools* pendukung) untuk mengembangkan arsitektur enterprise.

TOGAF adalah framework arsitektur yang memberikan metode dan *tools* untuk membantu dalam penerimaan, produksi, penggunaan, dan perawatan arsitektur enterprise. TOGAF didasarkan pada model proses iteratif yang didukung oleh *best practice* dan sekumpulan aset arsitektur yang bisa digunakan kembali.

TOGAF mendefenisikan enterprise sebagai sekumpulan organisasi yang memiliki sekumpulan tujuan (*goal*). Sebagai contoh, enterprise bisa berupa agen pemerintah, suatu perusahaan, suatu bagian dari perusahaan, sautu department, atau organisasi yang terpisah secara geografis namun mempunyai pemilik yang sama.

Istilah enterprise di dalam arsitektur enterprise dapat digunakan untuk menunjukan keseluruhan enterprise – menunjukan seluruh layanan teknologi dan informasi, proses-proses, dan infrastruktur maupun domain tertentu di dalam enterprise. Arsitektur melintasi berbagai sistem dan berbagai kelompok fungsional di dalam enterprise.

Tujuan dari arsitektur enterprise adalah mengoptimasi proses antar enterprise yang terpisah (baik manual atau otomatis) menjadi satu lingkungan terintegrasi yang merespon perubahan dan mendukung strategi bisnis. Arsitektur enterprise yang baik memungkinkan anda untuk mencapai keseimbangan antara efesiensi IT dan inovasi bisnis. Arsitektur enterprsise memungkinan unit bisnis untuk berinovasi untuk mendapatkan keuntungan strategis.

Framework arsitektur merupakan struktur dasar, atau sekumpulan struktur, yang dapat digunakan dalam membangun arsitektur yang berbeda. *Framework* tersebut harus bisa menjelaskan metode perancangan target enterprise di dalam sekumpulan bagian dan menunjukan bagaimana bagian-bagian tersebut saling terhubung. *Framework* mengandung sekumpulan tools dan memberikan penjelasan mengenai istilah yang digunakan dan sekumpulan rekomendasi standar yang bisa digunakan di dalam bagian tersebut.

TOGAF sudah dikembangkan oleh lebih dari 300 perusahaan anggota forum arsitektur yang berasal dari perusahaan dan organisasi besar. Mengembangkan dan mendukung arsitektur enterprise merupakan teknik yang rumit karena mencakup berbagai stakeholder dan proses pengambilan keputusan di dalam organisasi. TOGAF memiliki peran penting di dalam standarisasi dan menjelaskan proses pengembangan arsitektur. TOGAF memberikan praktik kerangka kerja terbaik untuk menambah nilai, dan memungkinkan organisasi untuk membangun solusi yang dapat dilakukan secara ekonomis dan mengakomodasi persoalan dan kebutuhan bisnis. (Harison, 2009)

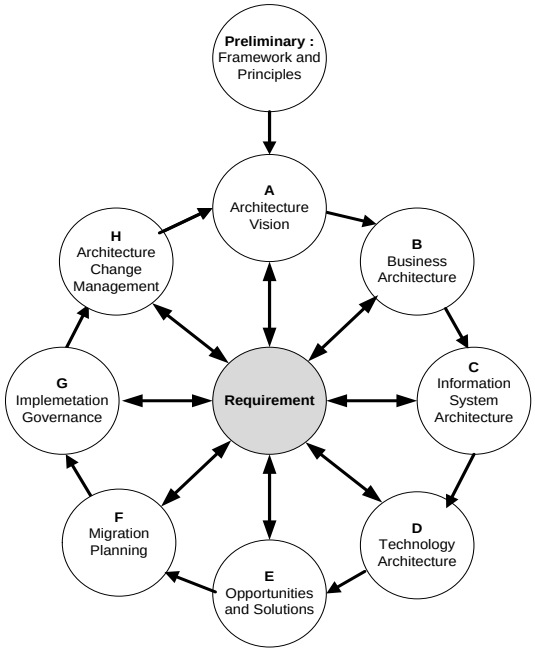
2.4.1 TOGAF ADM

TOGAF *Architecture Development Method* (ADM) memberikan proses teruji dan dapat diulang untuk mengembangkan arsitektur. ADM mencakup penetapan framework arsitektur, membangun konten arsitektur, peralihan, dan mengatur realisasi arsitektur.

Semua aktifitas tersebut ada di dalam suatu siklus iteratif dari defenisi dan realisasi arsitektur yang memungkinkan organisasi mengubah perusahaan menjadi bisa dikontrol yang berhubungan dengan tujuan dan peluang bisnis (Harison, 2009). Fase-fase yang ada di dalam ADM adalah sebagai berikut:

1. Fase *Preliminary* menjelaskan aktifitas persiapan dan permulaan yang dibutuhkan untuk membuat *Architecture Capability* termasuk penyesuaian TOGAF dan pendefinisian prinsip arsitektur.
2. Fase A: *Architecture Vision* menjelaskan fase awal dari siklus pengembangan arsitektur yang mencakup informasi mengenai penentuan ruang lingkup pembangunan arsitektur, identifikasi stakeholder, pembuatan visi arsitektur, dan memperoleh izin untuk membangun arsitektur.
3. Fase B: *Business Architecture* menjelaskan pembangunan Business Architecture untuk mendukung *Architecture Vision* yang sudah disetujui.
4. Fase C: *Information System Architecture* menjelaskan pembangunan arsitektur sistem informasi untuk mendukung *Architecture Vision* yang sudah disetujui.

5. Fase D: *Technology Architecture* menjelaskan pembangunan arsitektur teknologi untuk mendukung *Architecture Vision* yang sudah disetujui.
6. Fase E: O*pportunity and Solution* berhubungan dengan perencanaan implementasi awal dan identifikasi sarana pengiriman untuk arsitektur yang didefenisikan pada fase sebelumnya.
7. Fase F: *Migration Planning* menjelaskan bagaimana bergerak dari garis awal ke target arsitektur dengan mengakhiri rincian implementasi dan rencana perpindahan.
8. Fase G: *Implementation Governance* memberikan kesalahan implementasi yang arsitektural.
9. Fase H: *Architecture ChangeManagement* menetapkan prosedur untuk mengelola perubahan terhadap arsitektur yang baru.
10. *Requirement* memeriksa proses pengelolaan kebutuhan arsitektur melalui ADM.



Gambar 2.1 Tahapan TOGAF ADM

Sumber :Erwin Budi S.,*SNATI 2009* Pemilihan *Ea Framework hal. B-118*

2.5 Pemilihan Architecture Enterprise Framework

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

a. *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.

b. *Cohesive*

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

c. *Adaptable*

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

d. *Vendor-independent*

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

e. *Technology-independent*

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

f. *Domain-neutral*

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

g. *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 2.1. Dalam prakteknya EA *framework* yang ada tidak ada yang sempurna, masing-masing memiliki kelebihan dan kekurangan. Bahkan penggunaan EA *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 2.1 Perbandingan EA 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

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.

2.6 Value Chain

Model rantai nilai (*value chain)* menekankan aktivitas khusus pada bisnis diman 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 TI 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). (Laudon, 2007)



Gambar 2.2 *Value Chain*

Sumber : Kenneth C. Laudon & Jane P. Laudon,

Sistem Informasi Manajemen Mengelola Perusahaan Digital hal. 121

2.7 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 standard 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

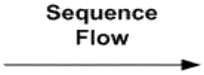
berbagai tingkatan manajemen yang harus dapat membaca dan memahami proses diagram dengan cepat sehingga dapat membantu dalam pengambilan keputusan.

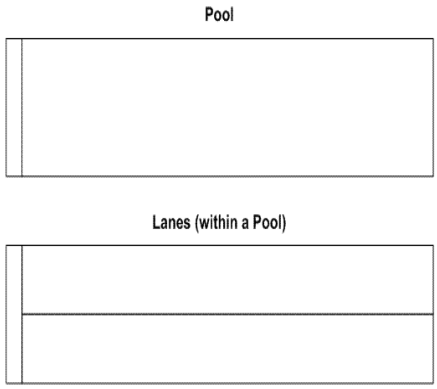
Notasi BPMN yang baru juga dirancang untuk sifat sistem berbasis layanan web. BPMN dapat memodelkan pesan kompleks yang dilewatkan diantara pelaku bisnis atau bagian dari pelaku bisnis, kejadian yang menyebabkan pesan dilewatkan, danaturan bisnis yang membatasi kejadian tersebut. BPMN memugkinkan proses 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)

Berikut ini adalah tabel 2.2 yang menerangkan simbol-simbol yang dipakai pada BPMN.

Tabel 2.2 Daftar Simbol BPMN

Simbol	Keterangan
	1. *Event* adalah sesuatu yang "terjadi" selama jalannya Proses atau Koreografi. Mempengaruhi aliran dari model dan biasanya memiliki penyebab (pemicu) atau dampak (hasil). Ada tiga jenis Event, berdasarkan ketika mereka mempengaruhi aliran: *start*, *intermediate*, dan *end.* 2. *Activities* adalah sebuah istilah umum untuk kerja bahwa perusahaan melakukan Proses. Sebuah Kegiatan dapat menjadi atom atau non-atom. Jenis Kegiatan yang merupakan bagian dari Proses sebuah Model adalah: *Sub-Proses* dan *Task*, yang bulat persegi panjang. 3. *Gateway* digunakan untuk mengontrol perbedaan dan konvergensi dari urutan Arus dalam Proses. Dengan demikian, akan menentukan percabangan, *forking*, penggabungan,dan bergabung dengan jalur. *Internal markers* akan menunjukkan jenis kontrol perilaku.
	Sebuah Arus Urutan digunakan untuk menunjukkan urutank egiatan yang akan dilakukan dalam proses.

	1. *Pool* adalah representasi grafis dari Peserta Kolaborasi. Hal ini juga bertindak sebagai "*swimlane*" dan wadah grafis untuk partisi satu set Pool 2. *Lanes* adalah partisi sub-dalam Proses, kadang-kadang dalam *Pool*, akan memperpanjang seluruhProses yang panjang, baik secara vertikal maupun horizontal. Jalur yang digunakan untuk mengatur dan mengkategorikan Kegiatan.

2.8 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain system 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.

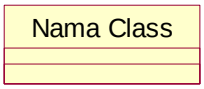
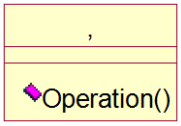
UML lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak kehadirannya pada tahun 1997. (Fowler, 2007)

2.8.1 *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*

Simbol	Keterangan
	Nama class mendeskripsikan nama class.
	Atribut mendeskripsikan properti dengan sebaris teks didalam kotak class tersebut
	Property merupakan suatu kegiatan yang dimengeri oleh sebuah class untuk dilakukan.
	Asosiasi mendeskripsikan hubungan antar class.
	Agregasi merupakan hubungan yang menyatakan bagian.

2.8.2 Use Case Diagram

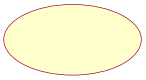
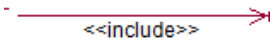
Use case adalah teknik untuk merekam persyaratan fungsional sebuah system. Use case mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi bagaimana sistem tersebut digunakan.

Setiap use case memiliki aktor utama yang meminta sistem untuk memberi sebuah layanan. *Actor* 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*

Simbol	Keterangan
	Aktor merupakan senuah entitas manusia/mesin yang berinteraksi dengan system.
	Use case mereprentasikan fungsionalitas dari suatu system/pekerjaan tertentu pada sistem.
	Asosiasi menjelaskan hubungan antar use case yang berupa pertukaran informasi.
	Dependency include menunjukkan suatu use case haru dipenuhi