

BAB II

TINJAUAN PUSTAKA

2.1. Kajian Pustaka

2.1.1. *Enterprise Architecture*

Enterprise adalah semua kumpulan organisasi yang memiliki tujuan [6]. Juga dapat diartikan setiap kumpulan organisasi yang memiliki seperangkat tujuan, informasi strategis berdasarkan aset yang mendefinisikan misi, kebutuhan informasi untuk melakukan misi, dan proses peralihan untuk mengimplementasikan teknologi baru dalam merespon kebutuhan perubahan misi [7]–[9].

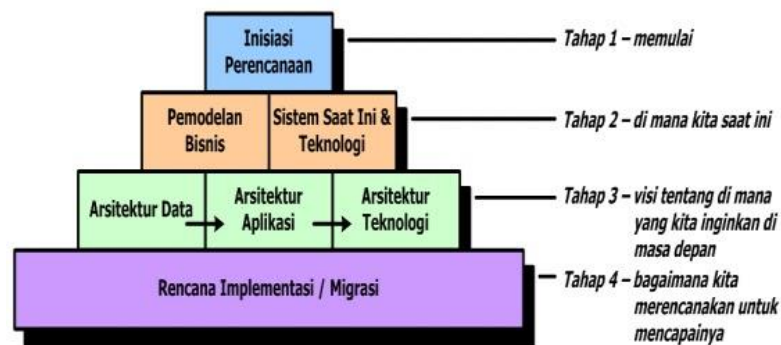
Sedangkan arsitektur merupakan dasar sistem organisasi yang terdiri dari sekumpulan komponen yang memiliki hubungan satu sama lainnya, serta memiliki keterhubungan dengan lingkungan sistem dan aturan untuk perancangan juga evaluasi [6], [10]. Dapat pula diartikan sebagai suatu rancangan dari obyek yang terdiri dari komponen-komponen yang saling berhubungan dalam bentuk *blueprint* untuk dijadikan dasar dalam mewujudkan suatu hasil yang nyata. Juga menyiratkan suatu perencanaan yang diwujudkan dengan model dan gambar dari komponen dari berbagai sudut pandang [11]–[13].

Enterprise architecture sendiri adalah aset informasi strategis yang mendefinisikan misi bisnis, informasi dan teknologi yang diperlukan untuk melakukan visi, dan proses transisi untuk mengimplementasikan teknologi baru dalam menjawab perubahan kebutuhan visi [6], [9], [14]. Tujuan dari *enterprise architecture* untuk mengoptimalkan organisasi dalam menanggapi perubahan

yang dilakukan secara manual ataupun otomatis ke dalam lingkungan yang yang terintegrasi dan mendukung pencapaian strategi bisnis organisasi [7], [15].

2.1.2. *Enterprise Architecture Planning (EAP)*

Enterprise architecture planning (EAP) merupakan metode yang dikembangkan untuk membangun *enterprise architecture*. Tujuan EAP adalah untuk mengoptimalkan seluruh proses yang terjadi pada masing-masing bagian organisasi ke dalam lingkungan terparu yang tanggap terhadap perubahan dan mendukung penyampaian strategi bisnis [6], [10], [16]. Perencanaan dalam EAP ini dilakukan sebagai tahap awal dalam mengembangkan arsitektur dalam mencapai visi dan misi ke depan. Tahapan pembangunan EAP memiliki 4 tahapan yang dapat dilihat pada gambar 2.1.



Gambar 2.1 Lapisan Perencanaan *Enterprise Architecture*

Dalam perancangan EAP ini ada beberapa kerangka kerja yang bisa digunakan. Dengan penggunaan kerangka kerja akan mempercepat dan menyederhanakan pengembangan arsitektur, memastikan cakupan komplit dari solusi desain dan memastikan arsitektur yang terpilih akan memungkinkan pengembangan di masa depan sebagai respon terhadap kebutuhan bisnis [6], [14], [17]. Kerangka kerja EAP didominasi oleh 4 kerangka kerja yang terbesar dan sering digunakan ; *The Zachman Framework for Enterprise Architecture*, *The Open Group Architecture Framework (TOGAF)*, *The Federal Enterprise Architecture (FEA)*, dan *Gartner (Meta Framework)*. Dalam pemilihan sebuah

kerangka kerja EAP terdapat beberapa kriteria berbeda yang bisa dijadikan acuan. Pemetaan kriteria dan kerangka kerja ditunjukkan pada tabel 2.1.

Tabel 2.1 Perbandingan Kerangka Kerja Arsitektur *Enterprise* [7], [18]

	Zachman	FEAF	TOGAF
Definisi arsitektur dan pemahamannya	Parsial	Ya	Ya, pada tahap persiapan
Proses arsitektur yang detail	Ya	Tidak	Ya, ADM dengan 9 tahap yang detail
Dukungan terhadap evolusi arsitektur	Tidak	Ya	Ya, ada perencanaan migrasi
Standarisasi	Tidak	Tidak	Ya, menyediakan <i>technical reference model</i> (TRM), informasi yang standar
<i>Architecture knowledge base</i>	Tidak	Ya	Ya
Pendorong bisnis	Parsial	Ya	Ya
Input teknologi	Tidak	Ya	Ya
Model bisnis	Ya	Ya	Ya
Desain transisional	Tidak	Ya	Ya, hasil dari perencanaan migrasi

Dari pemetaan di atas dapat disimpulkan bahwa untuk kasus yang belum memiliki arsitektur *enterprise* dan memerlukan pengembangan arsitektur *enterprise* yang mudah dan jelas, maka kerangka kerja yang cocok digunakan adalah TOGAF.

2.1.3. *The Open Group Architecture Framework (TOGAF)*

TOGAF (*The Open Group Architecture Framework*), dibuat oleh “*The Open Group*”, merupakan salah satu *framework* arsitektur yang umum digunakan untuk membangun *enterprise architecture* [19]. Secara detail TOGAF dijelaskan

sebagai kerangka kerja yang terperinci dan seperangkat alat dalam pengembangan sebuah *enterprise architecture* yang digunakan untuk mendesain, membangun, dan mengevaluasi suatu arsitektur pada organisasi [3]. Arsitektur dimodelkan dalam 4 tingkat atau domain, yaitu:

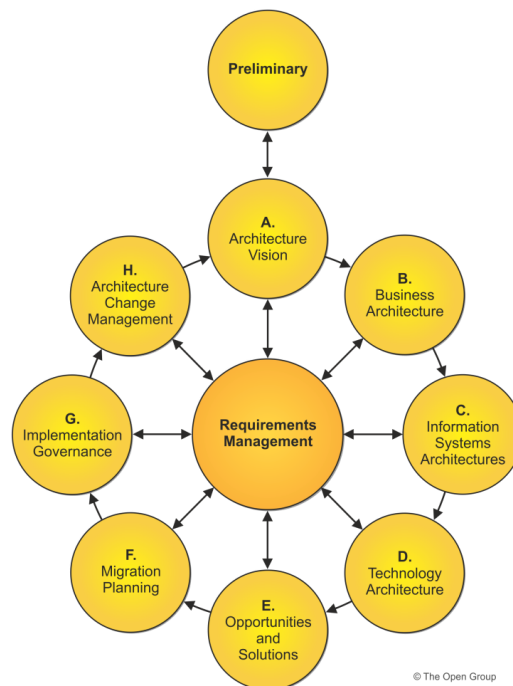
- 1) *Business architecture*, mendefinisikan proses bisnis yang mencakup tujuan, fungsi, strategi, dan proses bisnis dari organisasi
- 2) *Data architecture*, mengidentifikasi pengelolaan, penyimpanan, dan pengaksesan data
- 3) *Application architecture*, menentukan komponen perangkat lunak dan interaksi dengan aplikasi lain
- 4) *Technology architecture*, mengidentifikasi infrastruktur perangkat lunak, perangkat keras, dan jaringan untuk implementasi teknologi

TOGAF menyatukan berbagai domain, untuk memberikan kemampuan bisnis. Dengan menerapkan TOGAF maka dampak arsitektur akan terasa pada seluruh perusahaan tidak hanya sesempit fokus pada implementasi teknologi informasi. [5], [20]

2.1.3.1. TOGAF Architecture Development Method (ADM)

ADM (*Architecture Development Method*) sendiri merupakan metode umum untuk pengembangan arsitektur yang dirancang untuk menangani sebagian besar sistem pada perusahaan [12]. Salah satu hal yang harus dilakukan sebelum menerapkan ADM adalah meninjau komponen-komponennya agar dapat diterapkan, kemudian menyesuaikannya sesuai dengan keadaan masing-masing perusahaan [21].

Secara umum, TOGAF ADM mendefinisikan setiap fase dalam pengembangan arsitektur secara *general* general perusahaan yang mempunyai tujuan yang sama. Sehingga TOGAF ADM berperan secara fleksibel sesuai dengan kebutuhan perusahaan. Fase TOGAF ADM dalam pengembangan arsitektur diantaranya, dan dapat dilihat pada gambar 2.3.



Gambar 2.2 TOGAF ADM Cycle [6]

Pada gambar 2.2 dapat dilihat bahwa ada beberapa tahapan dalam TOGAF ADM. Untuk penjelasan masing-masing fase dapat dilihat pada penjelasan yang dijabarkan berikut ini.

1) ***Preliminary Phase : Framework and Principles***

Fase persiapan ini merupakan tahapan untuk mengkonfirmasi komitmen dari *stakeholder*, menentukan ruang lingkup *enterprise architecture* yang akan dikembangkan serta menentukan komitmen dengan manajemen dalam pengembangan *enterprise architecture* [7].

2) ***Phase A : Architecture Vision***

Fase ini menciptakan keseragaman pandangan mengenai pentingnya arsitektur *enterprise* untuk mencapai tujuan organisasi dengan memvalidasi visi, misi, tujuan organisasi dan mengidentifikasi *stakeholder*. Fase ini menghasilkan persetujuan pengerjaan yang meliputi lingkup dan rencana pengerjaan arsitektur, serta prinsip arsitektur [6].

3) ***Phase B : Business Architecture***

Fase ini mendefinisikan kondisi awal arsitektur bisnis, menentukan model bisnis atau aktivitas bisnis yang diinginkan berdasarkan skenario bisnis. Juga menjelaskan mengenai layanan dan aspek dari organisasi bisnis, fungsional bisnis, proses bisnis, informasi bisnis yang diperlukan perusahaan yang mempengaruhi lingkungan bisnis dari *enterprise* [22].

4) ***Phase C : Information System Architecture***

Fase untuk menekankan aktivitas bagaimana arsitektur sistem informasi dikembangkan. Dalam fase ini melibatkan kombinasi antara *data architecture* dan *application architecture*, komponen sistem informasi. *Data architecture* menentukan tipe dan sumber data yang diperlukan, sedangkan *application architecture* menentukan jenis sistem yang dibutuhkan untuk memproses data dan mendukung proses bisnis [10].

5) ***Phase D : Technology Architecture***

Fase ini menggambarkan kapabilitas perangkat lunak dan perangkat keras yang dibutuhkan untuk implementasi lingkungan teknologi yang mendukung bisnis dan layanan aplikasi [10].

6) ***Phase E : Opportunities and Solution***

Fase ini lebih menekankan pada manfaat yang diperoleh dari arsitektur yang dibangun, sehingga menjadi dasar bagi *stakeholder* untuk memilih dan menentukan arsitektur yang akan diimplementasikan dengan mengevaluasi opsi perencanaan [9].

7) ***Phase F : Migration Planning***

Fase ini berfokus pada rencana migrasi dari arsitektur yang telah dibangun dan dipilih oleh *stakeholder* ke dalam implementasi perusahaan [9]. Dalam rencana migrasi dijelaskan bagaimana cara pemindahan proses pada arsitektur berdasarkan prioritas.

8) ***Phase G : Implementation Governance***

Fase dimana dibuat formulasi rekomendasi setiap implementasi proyek, membuat kontrak arsitektur yang akan menjadi acuan implementasi dan menjaga kesesuaiannya dengan arsitektur yang telah ditentukan [6].

9) ***Phase H : Architecture Change Management***

Fase untuk membentuk skema proses manajemen perubahan arsitektur saat arsitektur terimplementasi pada organisasi, memastikan bahwa arsitektur memenuhi kebutuhan dan memaksimalkan nilai bisnis organisasi [7].

10) ***Requirements Management***

Fase yang bertujuan untuk menyediakan kebutuhan setiap tahap arsitektur sepanjang fase TOGAF ADM, mengidentifikasi kebutuhan *enterprise*, menyimpan lalu memberikan ke fase yang relevan [6].

2.1.3.2. Deliverables, Artifacts, dan Building Blocks

Pada TOGAF tersedia suatu *Architecture Content Framework* yang menyediakan model struktural bagi produk kerja yang diciptakan selama tahap pengerjaan TOGAF secara konsisten didefinisikan, terstruktur, dan disajikan [6]. TOGAF memberikan referensi dan titik awal *content* yang akan dipetakan ke *architecture content* lainnya [9].

Architecture content framework menggambarkan jenis produk kerja arsitektur ke dalam 3 (tiga) kategori berikut [6], [7], [20]:

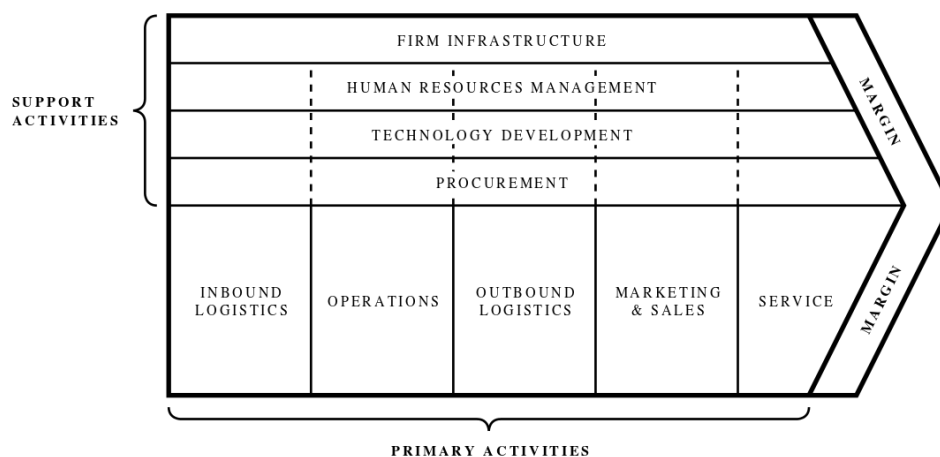
- 1) *Deliverables*, produk kerja yang ditinjau secara resmi, disetujui, dan ditandatangani oleh pemangku kebijakan atau pihak berkepentingan terkait yang berupa dokumentasi dan diarsipkan pada penyelesaian proyek.
- 2) *Artifacts*, produk kerja yang menggambarkan arsitektur dari sudut pandang tertentu secara lebih rinci, dimana diklasifikasikan sebagai katalog (daftar dari aset), matriks, dan diagram.

- 3) *Building blocks*, mewakili komponen dari *enterprise* yang berpotensi dapat digunakan kembali dan dapat dikombinasikan dengan *building blocks* lainnya untuk memberikan solusi arsitektur.

2.1.4. Value Chain

Menurut Michael E. Porter, *value chain* adalah model yang digunakan untuk membantu menganalisis aktivitas spesifik yang dapat menciptakan nilai dan keuntungan kompetitif bagi perusahaan. Aktivitas ini dibagi menjadi dua jenis, yaitu *primary activities* dan *support activities* [15]. Juga dikatakan *value chain* merupakan sebuah perspektif dimana bisnis dipandang sebagai rantai kegiatan dalam mengubah *input* menjadi *output* yang memberikan nilai kepada pelanggan [11].

Sehingga dapat disimpulkan bahwa *value chain* adalah alat bantu untuk menganalisis proses atau layanan yang ada, sehingga dapat menemukan kekurangan atau kelebihan pada layanan perusahaan dalam bentuk deskripsi proses dalam pengembangan dan pemasaran produk atau jasa. Model penggambaran dari *value chain* dapat dilihat pada gambar 2.2.



Gambar 2.3 Model Value Chain [7]



2.1.5. Business Process Modelling Notation (BPMN)



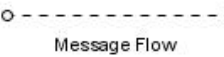
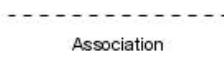
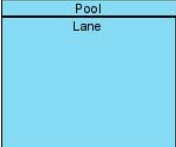


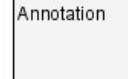
BPMN merupakan metodologi yang dikembangkan oleh *Business Process Modelling Initiative* sebagai suatu standar baru pada pemodelan proses bisnis, juga sebagai alat desain pada sistem yang kompleks seperti sistem *e-Business* yang berbasis pesan [23]. Tujuan utama BPMN adalah menyediakan notasi yang mudah digunakan dan bisa dimengerti oleh semua orang yang terlibat dalam bisnis meliputi bisnis analis yang memodelkan proses bisnis, pengembang yang membangun sistem, dan berbagai tingkatan manajemen yang harus dapat membaca dan memahami proses diagram dengan cepat sehingga dapat membantu dalam pengambilan keputusan [15].

Salah satu kelebihan 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. [7]

Terdapat 4 kategori dari elemen yang ada pada BPMN, kategori tersebut dapat dilihat lebih detail pada tabel di bawah [23].

Tabel 2.2 Kategori dan Simbol BPMN

No	Kategori	Nama	Deskripsi
1	<i>Flow Objects</i>	<i>Events</i>	 <p>Direpresentasikan dengan lingkaran, dapat berupa <i>Start</i>, <i>Intermediate</i>, atau <i>End</i></p>
		<i>Activities</i>	 <p>Memperlihatkan pekerjaan yang harus dilakukan</p>

		<i>Gateways</i>	 Gateway Fork/Join Inclusive Decision/Merge	Memperlihatkan pilihan yang berbeda, juga menjelaskan mengenai percabangan dan penggabungan dari <i>path</i> yang ada
2	<i>Connecting Objects</i>	<i>Sequence Flow</i>	 Sequence Flow	Menjelaskan mengenai urutan aktivitas yang akan dijelaskan
		<i>Message Flow</i>	 Message Flow	Menjelaskan pertukaran pesan yang terjadi
		<i>Association</i>	 Association	Untuk mengasosiasikan sebuah artifak, data, maupun <i>flow object</i>
3	<i>Swimlanes</i>	<i>Pool</i>		Direpresentasikan dengan persegi besar yang di dalamnya berisi <i>flow objects</i> , <i>connecting object</i> , maupun artifak
		<i>Lane</i>		Merupakan bagian lebih mendetail dari <i>pool</i>
4	<i>Artifacts</i>	<i>Data Objects</i>	 Data	Menjelaskan mengenai data yang dibutuhkan atau dihasilkan dari sebuah aktivitas
		<i>Group</i>		Untuk <i>grouping</i> aktivitas
		<i>annotation</i>		Untuk menjelaskan model atau diagram

2.1.6. *Unified Modelling Language (UML)*

UML merupakan Bahasa umum yang digunakan oleh analis, pengembang dan arsitek perangkat lunak untuk mendeskripsikan, menentukan, merancang, dan mendokumentasikan proses bisnis, struktur dan perilaku artifak dari sebuah perangkat lunak [11].

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. [7], [10]

Kesimpulannya UML adalah suatu bahasa pemrograman yang digunakan untuk menggambarkan suatu sistem perangkat lunak antara lain mendefinisikan *Irequirement*, membuat analisis dan desain, serta menggambarkan arsitektur yang ada.

2.1.6.1. **Class Diagram**

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* [7] :

- 1) Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- 2) Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Simbol yang ada pada *class diagram* dapat dilihat pada tabel 2.3.

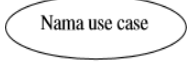
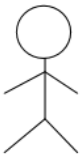

Tabel 2.3 Daftar Simbol *Class Diagram* [18]



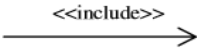
No	Simbol	Nama	Deskripsi
1		<i>Class</i>	Kelas pada struktur sistem
2		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3		<i>Association</i>	Relasi antar kelas dengan makna umum. Asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4		Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7		Agregasi	Relasi antar kelas dengan makna seba-bagian (<i>whole-part</i>)

2.1.6.2. Use Case Diagram

Use Case Diagram adalah pemodelan untuk sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [18]. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol yang digunakan di *use case diagram* dapat dilihat pada tabel di bawah.

Tabel 2.4 Daftar Simbol *Use Case Diagram* [7]

No	Simbol	Nama	Deskripsi
1		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerha diawal frase nama <i>use case</i>
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atay <i>use case</i> memiliki interaksi dengan aktor

4		<i>Extend</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> ditambahkan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5		<i>Generalization</i>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
6		<i>Include</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p>

2.1.7. Analisis Gap

Analisis gap dapat diartikan sebagai perbandingan kinerja aktual dengan kinerja yang diharapkan. Juga merupakan metode yang digunakan untuk mengevaluasi kinerja perusahaan yang menghasilkan masukan berguna bagi perusahaan terutama pada saat perencanaan ataupun evaluasi [24]. Metode ini merupakan salah satu metode yang umum digunakan dalam pengelolaan manajemen internal suatu organisasi. Dimana secara harfiah kata “gap” memiliki arti perbedaan (*disparity*) antara suatu hal dengan hal lainnya [7].

Pada perancangan arsitektur TOGAF ADM perbandingan analisis gap dilakukan pada fase *business architecture*, *data architecture*, *application architecture*, dan *technology architecture*. Yang dibandingkan adalah *existing architecture* dan *target architecture*. Setiap gap yang dianalisis juga dilakukan penentuan masukan untuk fase *opportunities & solutions* dan *migration planning*. [7], [8], [18]

2.2. Penelitian Terdahulu

Sebagai acuan dalam pengerjaan penelitian, terdapat beberapa penelitian terdahulu yang berkaitan dengan penelitian ini. Penelitian terdahulu yang digunakan akan dijabarkan pada uraian di bawah.

- 1) Judul : *Development of Enterprise Architecture in University using TOGAF as Framework*
- Penulis : E Amalia, H Supriadi
- Tahun : 2017
- Permasalahan : Universitas XYZ memiliki fasilitas dan infrastruktur yang mendukung IT infrastruktur, namun dengan menggunakan arsitektur yang ketinggalan zaman. Ditambah dengan adanya sistem di beberapa divisi yang tidak terintegrasi. [14]

- Hasil : Penelitian ini menghasilkan tahapan sebagai berikut:
(1) membentuk desain menggunakan TOGAF dan ADM pada sembilan bidang fungsional bisnis dan mengusulkan 12 calon aplikasi untuk dikembangkan di Universitas XYZ; (2) menghasilkan 11 prinsip pengembangan arsitektur teknologi informasi; (3) membuat portofolio untuk aplikasi masa depan menghasilkan 6 aplikasi di kuadran strategis, 2 operasi aplikasi kuadran, 4 dukungan aplikasi kuadran; dan (4) pemodelan arsitektur enterprise dari penelitian ini yang dapat menjadi acuan dalam membuat cetak biru pengembangan sistem informasi dan teknologi informasi di Universitas XYZ. [14]
- 2) Judul : *TOGAF for Designing The Enterprise Architecture of LAZISMU*
- Penulis : J Fahana, A Azhari
- Tahun : 2018
- Permasalahan : LAZISMU sebagai Lembaga Amil Zakat yang bertugas utama untuk menghimpun dan mengelola zakat masyarakat belum menggunakan sistem informasi yang dapat mendukung proses bisnis. [4]
- Hasil : Penelitian ini menghasilkan aplikasi dan entitas yang dibutuhkan untuk pengembangan LAZISMU sekarang dan di masa depan. Persyaratan aplikasi dan entitas yang ada adalah analisis urutan proses di TOGAF ADM. Beberapa aplikasi yang paling dibutuhkan terdiri dari aplikasi manajemen keuangan, aplikasi pemanfaatan, aplikasi profil web, dan e-marketing. [4]

- 3) Judul : *Design of Architecture Enterprise Model Information System Academic and Student Administrations Bureau using TOGAF ADM*
- Penulis : A Karim, I Gundar
- Tahun : 2019
- Permasalahan : BAAK UNU masih menggunakan aplikasi standar dimana tidak memiliki basis data terkait pengolahan data operasional. [20]
- Hasil : Penelitian ini menghasilkan cetak biru arsitektur sistem informasi berupa aplikasi calon tiap sub organisasi yang disusun berdasarkan tingkat prioritas kebutuhan sehingga proses implementasi sesuai dengan yang diinginkan dan tidak mengganggu kinerja organisasi serta sistem informasi yang sedang berjalan. Dalam penelitian ini terlihat adanya gap pada proses bisnis dimana dalam menjalankan bisnis di BAAK UNU belum menggunakan IT sebagai kebutuhan utamanya serta pengolahan data operasional akademik masih menggunakan aplikasi standar seperti Microsoft Office. [20]
- 4) Judul : *Perencanaan Arsitektur Enterprise Kantor Wilayah Kementerian Agama Menggunakan Framework TOGAF ADM*
- Penulis : Yudi Hermawan
- Tahun : 2015
- Permasalahan : Pemanfaatan sistem informasi belum optimal dalam mendukung tugas dan fungsi organisasi, dimana hanya menunjang sebagian kecil bidang dan tidak terintegrasi antar bidang. [18]

- Hasil : Penelitian ini menghasilkan perancangan untuk sebuah sistem yang terintegrasi secara menyeluruh, yang terdiri dari empat aplikasi yaitu aplikasi Dokumentasi, klipng berita, dan Data Kanwil. Dengan catatan bahwa fasilitas arsitektur Teknologi Informasi sudah cukup memadai, namun tetap dibutuhkan pengembangan teknologi menyesuaikan perubahan zaman. Agar teknologi informasi menjadi faktor penunjang utama dalam menjalankan fungsi bisnis. [18]
- 5) Judul : *Designing Enterprise Architecture for Academics Information System Platform Using the Open Group Architecture Framework Architecture Development Method*
- Penulis : N Zulfarian, I D Rosiyadi
- Tahun : 2020
- Permasalahan : Belum memiliki arsitektur sistem informasi sekolah yang spesifik dan terintegrasi sesuai dengan kebutuhan dan tujuan, karena belum fokus pada proses pengembangan sistem informasi. [25]
- Hasil : Penelitian ini menghasilkan *blueprint* untuk sistem informasi akademik yang terintegrasi, sehingga dapat mempermudah dan mempercepat pelayanan ke seluruh pemangku kepentingan khususnya di bidang layanan informasi. Solusi yang ditawarkan adalah sistem informasi berbasis web, sistem informasi berbasis *mobile*, dengan integrasi menggunakan *platform Service Oriented Architecture*. [25]