BAB II
LANDASAN TEORI

2.1 *The Open Group Architecture Framework* (TOGAF)

Menurut Suhendro [12] TOGAF adalah satu kerangka terperinci dan alat pendukung untuk mengembangkan satu *Enterprise Architecture (*EA) yang dipergunakan dengan bebas oleh apapun organisasi yang mengembangkan untuk mendesain, evaluasi, dan membangun *blueprint* TI. Metodologi untuk desain arsitektur didalam TOGAF disebut *architecture development method* (ADM) yaitu suatu proses yang menyeluruh, terintegrasi untuk mengembangkan dan memelihara suatu EA.

EA dapat didefinisikan sebagai sebuah *blueprint* yang menjelaskan bagaimana semua elemen teknologi informasi (TI) dan manajemen bekerja bersama dalam satu kesatuan dan memberikan gambaran eksplisit mengenai hubungan antara proses manajemen dengan TI yang sekarang dan yang diharapkan [13]. Jika dikaitan dengan *enterprise,* maka *EA* harus memberikan strategi yang memungkinkan organisasi mendukung keadaan yang sekarang dan juga bertindak sebagai *roadmap* menuju lingkungan yang ditargetkan.

Arsitektur merupakan satu praktek manajemen untuk memaksimalkan kontribusi dari sumber daya perusahaan, investasi TI, dan aktivitas pembangunan sistem untuk mencapai tujuan kinerjanya. Guna mencapai misi organisasi melalui[12] kinerja optimal dari proses bisnis dengan efisiensi lingkungan TI maka penerapan *framework* harus dimasukkan kedalam *roadmap* dari perusahuaan. Arsitektur sistem terintegrasi TI menyediakan konteks strategis bagi evolusi sistem TI dalam menanggapi kebutuhan yang terus berubah di lingkungan bisnis. Arsitektur harus sejalan dengan TI dan bisnis. Hal ini memungkinkan unit bisnis untuk berinovasi mencapai keunggulan kompetitif, secara bersamaan, mendorong sinergi di seluruh unit bisnis perusahaan. Keuntungan dari arsitektur perusahaan yang baik adalah sebagai berikut.

a. Operasi TI lebih efisien.
b. Investasi yang menguntungkan.
c. Mengurangi risiko dalam hal penyimpangan terhadap aturan.
d. Lebih cepat, sederhana, dan operasi bisnis lebih efisien.

TOGAF secara umum memiliki struktur dan komponen sebagai berikut.

a. ADM

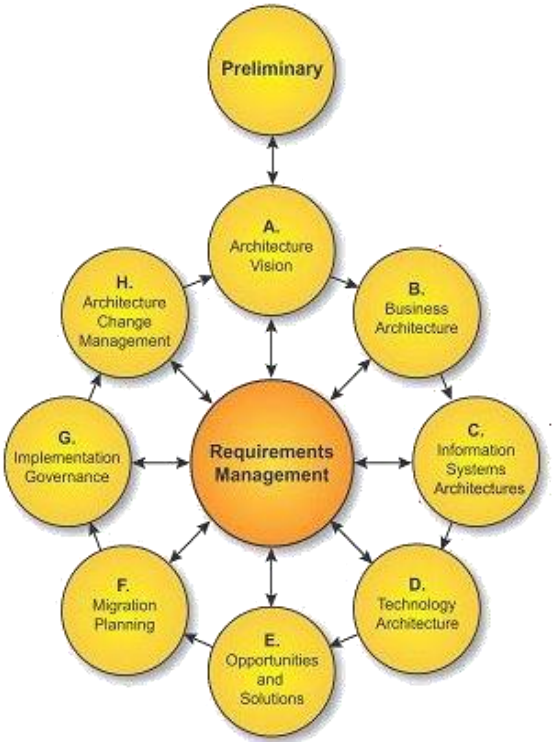
   Memberikan gambaran rinci bagaimana menentukan sebuah *Architecture* secara spesifik berdasarkan kebutuhan bisnisnya.

b. *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 f*oundation architecture* yaitu *Technical Reference Model, Standard Information dan Building Block Information Base*

c. *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. Dari ketiga komponen TOGAF tersebut, ADM merupakan elemen kunci dari TOGAF[4]. ADM adalah fitur penting yang memungkinkan perusahaan mendefinisikan kebutuhan bisnis dan membangun arsitektur spesifik untuk memenuhi kebutuhan itu. ADM memiliki tahapan-tahapan dasar seperti yang terlihat pada gambar 2.1 berikut ini:



Gambar 2.1 Tahapan TOGAF 9.1 ADM [4]

Sebagai komponen inti, TOGAF ADM menyediakan serangkaian proses iteratif mulai dari menyusun arsitektur, transisi, hingga mengelola proses realisasi arsitektur. Berikut adalah penjelasan dari kesepuluh fase TOGAF ADM.

a. *Preliminary Phase*

Fase ini mencakup aktivitas persiapan untuk menyusun kapabilitas arsitektur termasuk kustomisasi TOGAF dan mendefinisikan prinsip-prinsip arsitektur. Tujuan fase ini adalah untuk menyakinkan setiap orang yang terlibat di dalamnya bahwa pendekatan ini untuk mensukseskan proses arsitektur. [4]

Pada fase ini harus menspesifikasikan *who, what, why, when,* dan *where* dari arsitektur itu sendiri.[4]

1. *What* adalah ruang lingkup dari usaha.

2. *Who* adalah siapa yang akan memodelkannya, siapa orang yang akan bertanggung jawab untuk mengerjakan arsitektur tersebut, dimana mereka akan dialokasikan dan bagaimana peranan mereka.

3. *How* adalah bagaimana mengembangkan EA, menentukan *framework* dan metode apa yang akan digunakan untuk menangkap informasi.

4. *When* adalah kapan tanggal penyelesaian arsitektur.

5. *Why* adalah mengapa arsitektur ini dibangun. Hal ini berhubungan dengan tujuan organisasi yaitu bagaimana arsitektur dapat memenuhi tujuan organisasi.

b. *Phase A: Architecture Vision*

Menciptakan keseragaman pandangan mengenai pentingnya arsitektur *enterprise* untuk mencapai tujuan organisasi yang dirumuskan dalam bentuk strategi serta menentukan lingkup dari arsitektur yang akan dikembangkan. Fase ini berisikan pertanyaan-pertanyaan yang diajukan untuk mendapatkan arsitektur yang ideal. Fase ini merupakan fase inisiasi dari siklus pengembangan arsitektur yang mencakup pendefinisian ruang lingkup, identifikasi *stakeholders*, penyusunan visi arsitektur, dan pengajuan persetujuan untuk memulai pengembangan arsitektur. Beberapa tujuan dari fase ini adalah sebagai berikut.

1. Menjamin evolusi dari siklus pengembangan arsitektur mendapat pengakuan dan dukungan dari *enterpise management*.

2. Mengesahkan prinsip bisnis, tujuan bisnis dan pergerakan strategis bisnis organisasi.

3. Mendefinisikan ruang lingkup dan melakukan identifikasi serta memprioritaskan komponen dari arsitektur saat ini.

4. Mendefiniskan kebutuhan bisnis yang akan dicapai dalam usaha arsitektur ini dan batasannya.

5. Menghasilkan visi arsitektur yang menunjukkan respon terhadap kebutuhan dan batasannya.

Beberapa langkah yang dilakukan pada fase ini adalah sebagai berikut.

1. Menentukan / menetapkan proyek.

2. Mengindentifikasi tujuan dan pergerakan bisnis. Jika hal ini sudah didefinisikan, pastikan definisi ini masih sesuai dan lakukan klarifikasi terhadap bagian yang belum jelas.

3. Meninjau prinsip arsitektur termasuk prinsip bisnis. Meninjau ini berdasarkan arsitektur saat ini yang akan dikembangkan. Jika hal ini sudah didefinisikan, pastikan definisi ini masih sesuai dan lakukan klarifikasi terhadap bagian yang belum jelas.

4. Mendefinisikan apa yang ada di dalam dan di luar rungan lingkup usaha saat ini.

5. Mendefinisikan batasan-batasan seperti waktu, jadwal, sumber daya dan sebagainya.

6. Mengindentifikasikan stakeholder, kebutuhan bisnis dan visi arsitektur.

7. Mengembangkan *statement of architecture work.*

c. *Phase B: Business Architecture*

Mendefinisikan kondisi awal arsitektur bisnis, menentukan model bisnis atau aktivitas bisnis yang diinginkan berdasarkan skenario bisnis. Fase ini mencakup pengembangan arsitektur bisnis untuk mendukung visi arsitektur yang telah disepakati. Pada tahap ini *tools* dan *method* umum untuk pemodelan seperti: *Integration DEFinition* (IDEF) dan *Unified Modeling*

Language (UML) bisa digunakan untuk membangun model yang diperlukan. Beberapa tujuan dari fase ini adalah sebagai berikut:

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

Beberapa langkah yang dilakukan di fase ini adalah sebagai berikut.

1. Mengembangkan deskripsi asitektur bisnis saat ini untuk mendukung arsitektur bisnis target.
2. Mengindentifikasi *reference model*, sudut pandang dan *tools.*
3. Melengkapi arsitektur bisnis
4. Melakukan *gap analysis* dan membuat laporan

d. *Phase C: Information Systems Architectures*

Pada tahapan ini lebih menekankan pada aktivitas bagaimana arsitektur sistem informasi dikembangkan. Pendefinisian arsitektur sistem informasi dalam tahapan ini meliputi arsitektur data dan arsitektur aplikasi yang akan digunakan oleh organisasi. Arsitektur data lebih memfokuskan pada bagaimana data

digunakan untuk kebutuhan fungsi bisnis, proses dan layanan. Teknik yang bisa digunakan dengan yaitu: *ER-Diagram*, *Class Diagram*, dan *Object Diagram*. Tujuan dari fase ini adalah mengembangkan arsitektur tujuan dalam domain data dan aplikasi. Ruang lingkup dari proses bisnis yang didukung dalam fase C dibatasi pada proses-proses yang didukung oleh TI dan *interface* dari proses-proses yang berkaitan dengan non-TI. Implementasi dari arsitektur ini mungkin tidak perlu dalam urutan yang sama, diutamakan terlebih dahulu yang begitu sangat dibutuhkan. Tujuan dari arsitektur data adalah untuk mendefinisikan tipe dan sumber utama data yang diperlukan untuk mendukung bisnis dengan cara yaitu dapat dipahami oleh *stakeholder*, lengkap, konsisten, dan stabil. Penting untuk diketahui bahwa arsitektur ini tidaklah memperhatikan perancangan *database*. Tujuannya adalah untuk mendefinisikan entitas data yang relevan dengan *enterprise,* bukanlah untuk merancang sistem penyimpanan fisik dan logik. Beberapa langkah yang diperlukan untuk membuat arsitektur data adalah sebagai berikut.

1. Mengembangkan deskripsi arsitektur data dasar.
2. Review dan validasi prinsip, *reference model*, sudut pandang dan *tools*.
3. Membuat model arsitektur.
4. Memilih arsitektur data.
5. Melengkapi arsitektur data.
6. Melakukan *gap analysis* arsitektur data saat ini dengan arsitektur data target dan membuat laporan.

Tujuan dari arsitektur aplikasi adalah untuk mendefinisikan jenis-jenis utama dari sistem aplikasi yang penting untuk memproses data dan mendukung bisnis. (*Open Group*, 2009). Penting untuk diketahui bahwa arsitektur aplikasi ini tidaklah memperhatikan perancangan sistem aplikasi. Tujuannya adalah untuk mendefinisikan jenis-jenis sistem aplikasi yang relevan dengan *enterprise* dan aplikasi apa saja yang diperlukan untuk mengatur data dan menghadirkan informasi kepada aktor manusia dan komputer di *enterprise*. Aplikasi tidak diuraikan sebagai sistem komputer tetapi sebagai grup logik dari kemampuan

untuk mengatur objek data dalam arsitektur data dan mendukung fungsi-fungsi bisnis dalam arsitektur bisnis. Aplikasi dan kemampuan didefinisikan tanpa mereferensikan ke teknologi khusus. Suatu aplikasi bersifat stabil dan relatif tidak berubah sepanjang waktu sedangkan teknologi yang digunakan untuk mengimplementasikannya akan barubah sepanjang waktu, berdasarkan pada teknologi yang sekarang tersedia dan perubahan kebutuhan bisnis.

Beberapa langkah yang diperlukan untuk membuat arsitektur aplikasi adalah sebagai berikut.

1. Mengembangkan deskripsi arsitektur aplikasi dasar
2. *Review* dan validasi prinsip, *reference model*, sudut pandang dan *tools*.
3. Membuat model arsitektur
4. Identifikasi sistem aplikasi kandidat
5. Melengkapi arsitektur aplikasi
6. Melakukan *gap analysis* dan membuat laporan

e. *Phase D: Technology Architecture*

Membangun arsitektur teknologi yang diinginkan, dimulai dari penentuan jenis kandidat teknologi yang diperlukan dengan menggunakan *Technology Portfolio Catalog* yang meliputi perangkat lunak dan perangkat keras. Dalam tahapan ini juga mempertimbangkan alternatif-alternatif yang diperlukan dalam pemilihan teknologi. Beberapa langkah yang diperlukan untuk membuat arsitektur teknologi yaitu sebagai berikut.

1. Membuat deskripsi dasar dalam format TOGAF.
2. Mempertimbangkan *reference model* arsitektur yang berbeda, sudut pandang dan *tools*.
3. Membuat model arsitektur.
4. Memilih *services* portfolio yang diperlukan
5. Mengkonfirmasi bahwa tujuan bisnis tercapai
6. Menentukan kriteria pemilihan spesifikasi

7. Melengkapi definisi arsitektur
8. Melakukan *gap analysis* antara arsitektur teknologi saat ini dengan arsitektur teknologi target.

f. *Phase E: Opportunities and Solutions*

Pada tahap ini akan dievaluasi model yang telah dibangun untuk arsitektur saat ini dan tujuan, indentifikasi proyek utama yang akan dilaksanakan untuk mengimplementasikan arsitektur tujuan dan klasifikasikan sebagai pengembangan baru atau penggunaan kembali sistem yang sudah ada. Pada fase ini juga akan direview *gap analysis* yang sudah dilaksanakan pada fase D. Tujuan dari fase ini adalah sebagai berikut.

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

g. *Phase F: Migration and Planning*

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

h. *Phase G: Implementation Governance*

Fase ini mencakup pengawasan terhadap implementasi arsitektur. Tujuan dari fase ini adalah sebagai berikut.

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

i. *Phase H: Architecture Change Management*

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

j. *Requirements Management*

Menguji proses pengelolaan *architecture requirements* sepanjang siklus ADM berlangsung.

Berdasarkan penjelasan di atas diketahui bahwa TOGAF ADM merupakan metode generik yang berisikan sekumpulan aktivitas yang digunakan dalam memodelkan pengembangan arsitektur *enterprise*. Metode ini juga bisa digunakan sebagai panduan atau alat untuk merencanakan, merancang,

mengembangkan dan mengimplementasikan arsitektur sistem informasi untuk organisasi [14]. TOGAF ADM merupakan *framework* yang memungkinkan dan banyak digunakan untuk diimplementasikan diberbagai bentuk dan bidang organisasi seperti perbankan, industri manufaktur dan juga pendidikan. TOGAF ADM membentuk sebuah siklus yang iteratif untuk keseluruhan proses, antar fase, dan dalam tiap fase dimana pada tiap-tiap iterasi keputusan baru harus diambil. Keputusan tersebut dimaksudkan untuk menentukan luas cakupan *enterprise*, level kerincian, target waktu yang ingin dicapai dan aset arsitektural yang akan digali dalam *enterprise continuum.* ADM merupakan metode yang umum sehingga jika diperlukan pada prakteknya ADM dapat disesuaikan dengan kebutuhan spesifik tertentu, misalnya digabungkan dengan *framework* yang lain sehingga ADM menghasilkan arsitektur yang spesifik terhadap organisasi.

TOGAF dibagi kedalam empat bagian pengembangan arsitektur. Pengembangan sistem dimulai dari mendefinisikan arsitektur bisnis yang ada dalam organisasi, mendefinisikan arsitektur data yang akan digunakan, mendefinisikan arsitektur aplikasi yang akan dibangun serta mendefinisikan arsitektur teknologi[5]. TOGAF memiliki metode dan *tools* yang detil untuk mengimplementasikannya, dan hal inilah yang membedakan TOGAF dengan *Framework* EA lainnya. Kelebihan menggunakan TOGAF adalah karena sifatnya yang fleksibel dan bersifat *open source.* TOGAF ADM merupakan metode yang fleksibel yang dapat mengantifikasi berbagai macam teknik pemodelan yang digunakan dalam perancangan, karena metode ini bisa disesuaikan dengan perubahan dan kebutuhan selama perancangan dilakukan. TOGAF ADM juga menyatakan visi dan prinsip yang jelas tentang bagaimana melakukan pengembangan arsitektur *enterprise*, prinsip tersebut digunakan sebagai ukuran dalam menilai keberhasilan dari pengembangan arsitektur *enterprise* oleh organisasi (*Open Group*, 2009). Prinsip-prinisip tersebut dapat dijelaskan sebagai berikut.

a. Prinsip *Enterprise*

   Pengembangan arsitektur yang dilakukan diharapkan mendukung seluruh bagian organisasi, termasuk unit-unit organisasi yang membutuhkan.

b. Prinsip Teknologi Informasi (TI)

   Lebih mengarahkan konsistensi penggunaan TI pada seluruh bagian organisasi, termasuk unit- unit organisasi yang akan menggunakan.

c. Prinsip Arsitektur

   Merancang arsitektur sistem berdasarkan kebutuhan proses bisnis dan bagaimana mengimplementasikannya.

Langkah awal yang perlu diperhatikan pada saat mengimplementasikan TOGAF ADM adalah mendefinisikan persiapan-persiapan yaitu dengan cara mengidentifikasi konteks arsitektur yang akan dikembangkan, kedua adalah mendefenisikan strategi dari arsitektur dan menetapkan bagian-bagian arsitektur yang akan dirancang, yaitu mulai dari arsitektur bisnis, arsitektur sistem informasi, arsitektur teknologi, serta menetapkan kemampuan dari arsitektur yang akan dirancang dan dikembangkan (*Open Group*, 2009).

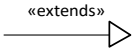
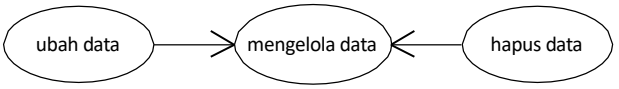
2.2 Alat (*Tools*) dan Teknik yang Digunakan

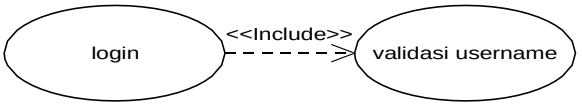
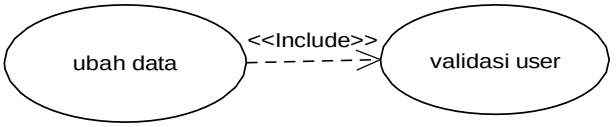
Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami [15]. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang dibuat aktor dan *use case*.

a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2.1 Simbol *Use Case* Diagram [15]

Keterangan	Simbol	Deskripsi
Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama *use case*
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi		Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
Ekstensi		Relasi use case tambahan ke sebuah *use case*, dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misal Arah panah mengarah pada *use case* yang ditambahkan.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum).

Menggunakan /*include/uses*	<<Include>> «uses»	Ada dua sudut pandang yang cukup besar mengenai *include* di *use case* : a. Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, misal pada kasus berikut :  b. Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan, misal pada kasus berikut :  Ke dua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

Sumber [15]

2.3 *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas [15].

Kelas-kelas yang ada pada struktur sistem, harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

a. Kelas main

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

b. Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

c. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi-fungsi yang baru ada diambil dari pendefinisian *use case.*

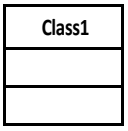
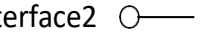
d. Kelas yang diambil dari pendefinisian data

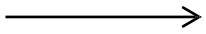
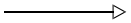
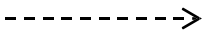
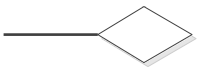
   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas tersebut juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks dan lainnya.

Dalam mengidentifikasikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain, sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum, maka sebuah metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah. Simbol-simbol yang ada pada diagram kelas adalah seperti pada Tabel 2.2.

Tabel 2.2. Simbol *Use Case* Diagram

Simbol	**Deskripsi**
Kelas 	Kelas pada struktur sistem.
Natarmuka/*interface* 	Sama dengan konsep *interface* dalam pemograman berorientasi objek.
Asosiasi 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.

Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan maknasemua bagian (*whole-part*).

Sumber [15]

2.4 Observasi

Menurut Jogiyanto dalam Faris [16] observasi (observation) merupakan teknik atau pendekatan untuk mendapatkan data primer dengan cara mengamati langsung obyek datanya. Kegiatan observasi ini dimulai dengan melakukan pengamatan langsung terhadap proses bisnis dan strategi bisnis di organisasi, mengamati apa saja yang menjadi dukungan agar proses bisnis dapat berjalan sesuai dengan yang diinginkan oleh instansi, melihat III - 3 sejauh mana pemanfaatan sistem dan teknologi informasi di organisasi tersebut, serta mencari data dan informasi terkait lainnya yang diperlukan untuk penelitian. Teknik observasi dilakukan untuk mempelajari dan memperoleh gambaran nyata dari setiap aktivitas bisnis, serta dukungan apa saja yang sudah ada agar aktivitas-aktivitas bisnis tersebut dapat berjalan sesuai dengan tujuan organisasi.

2.5 Wawancara

Teknik ini dilakukan untuk mendapatkan informasi yang berkaitan dengan kegiatan-kegiatan utama (aktivitas bisnis) di organisasi. Dalam hal ini wawancara dilakukan dengan pihak yang dianggap mengetahui semua hal yang bertujuan untuk

mendapatkan data dan informasi yang berkaitan dengan proses bisnis yang berjalan di organisasi. Penelitian ini menggunakan wawancara untuk mengumpulkan data yang diperlukan berkaitan dengan proses bisnis organisasi dan aliran input-proses-output untuk mempertahankan dan mengembangkan bisnis serta untuk meningkatkan profit organisasi.

2.6 Penelitian Sebelumnya

Berikut ini beberapa penelitian terkait yang digunakan sebagai acuan pada penelitian yang dilakukan.

Tabel 2.1 Penelitian Terdahulu

Dari penelitian terkait tersebut, didapatkan informasi mengenai metodologi seperti

NO	Penulis	Judul	Tahun	Hasil
1	Yeni Kustiyahningsih	Perencanaan Arsitektur *Enterprise* menggunakan Metode TOGAF ADM (Studi Kasus : RSUD Dr.Soegiri Lamongan)	2013	Penelitian ini memberikan solusi bisnis untuk RSUD Dr. Soegiri Lamongan dengan perencanaan arsitektur *enterprise* menggunakan metode TOGAF ADM. Tahapan metode TOGAF ADM yang dilakukan dimulai pada fase *Preliminary* hingga fase *Architecture Change Management*. Hasil dan pembahasan berisi fungsi bisnis RSUD Dr.Soegiri yang digambarkan menggunakan *value chain* sedangkan

apa yang digunakan, beberapa contoh *tools* yang digunakan dalam perancangan *Enterprise Architecture* menggunakan TOGAF, serta mengetahui perbedaan dari

				pemodelan proses bisnis digambarkan mengunakan UML yaitu *Use Case Diagram*.
2	Guntur Ardi Suryonugroho, Eko Darwiyanto dan Gede Agung Ary W.	Perencanaan Strategis Sistem Informasi Dinas Kesehatan Kota Bandung menggunakan *The Open Group Architecture Framework* (TOGAF) dan *Architecture Development Method* (ADM)	2015	Penelitian ini bertujuan untuk melakukan perencanaan strategis SI pada Dinas Kesehatan Kota Bandung menggunakan TOGAF dan ADM. Perencanaan strategis ini harus dilakukan melalui berbagai tahapan. Dimulai dengan studi pustaka dan pengumpulan data mengenai masalah masalah yang berkaitan dengan proses bisnis di Dinas Kesehatan Kota Bandung
3	Husain, Pulung Nurtantio Andono dan M. Arif Soeleman	Perspektif Baru *Enterprise Architecture* Pemerintahan Kota Mataram Berbasis TOGAF ADM	2017	Penelitian ini merupakan penelitian deskriptif kualitatif
4	Kholid Haryono	Model Arsitektur Sistem dan Teknologi Informasi	2015	Penelitian ini menggambarkan model arsitektur *enterprise*

5	Cipta Muhamad Firmansyah dan Yoanes Bandung	*Designing an Enterprise Architecture Government Organization Based on TOGAF ADM and SONA*	2016	Penelitian ini menjelaskan bahwa untuk menciptakan infrastruktur TI yang berkelanjutan dan fleksibel, perlu untuk menyelaraskan antara kebutuhan TI dengan bisnis organisasi, sehingga kedua entitas tersebut akan saling mendukung. *Enterprise Architecture* dapat membantu membangun keselarasan itu dan pendekatan arsitektur yang berorientasi pada layanan akan membantu merancang infrastruktur
6	Ian Yosef Matheus Edward, Wervyan Shalannanda, Aldo Agusdian dan Susmini Indriani Lestariningati	*E-Government Master Plan Design with TOGAF*	2014	Penelitian ini berisi tentang bagaimana menerapkan *framework* TOGAF dalam merancang *master plan* atau rancangan induk *E-Government* untuk pemerintahan Kota Payakumbuh
7	Anfusa Gandri Herucakra, Ari Fajar dan Ridha Hanafi	Analisis dan Perancangan *Enterprise Architecture* untuk II-42	2015	Penelitian ini bertujuan untuk melakukan perencanaan

		Mendukung Fungsi Terkait *System Online Payment Point* menggunakan *Framework* TOGAF ADM pada PT. Pos Indonesia		*Enterprise Architecture* yang dapat menjamin penerapan sistem secara optimal dan dapat memenuhi seluruh kebutuhan bisnis PT.Pos Indonesia. Metode perancangan *Enterprise Architecture* yang digunakan ialah TOGAF ADM. Tahapan TOGAF ADM yang dilakukan yaitu dimulai dari fase *Preliminary* sampai dengan *Phase D*
8	Aradea Dipaloka, Husni Mubarok, dan Andi Rosandi	*Blueprint* Teknologi Informasi untuk Mengintegrasikan Sistem Informasi Perguruan Tinggi	2013	membuat sebuah arsitektur *enterprise* untuk STIKes Muhammadiyah Tasikmalaya, yang berupa *blueprint* IT yang digunakan untuk mencapai visi dan misi dari organisasi tersebut. Perencanaan arsitektur *enterprise* tersebut dilakukan dengan menggunakan metodologi EAP (*Enterprise Architecture Planning*).

setiap penelitian terkait. Selain itu berdasarkan penelitianpenelitian tersebut, dapat diketahui bahwa penelitan yang dilakukan memiliki topik yang sama yaitu mengenai perancangan *Enterprise Architecture* menggunakan *framework* TOGAF dengan metode ADM, namun dengan tempat studi kasus yang berbeda yaitu dilakukan di Dinas Perpustakaan dan Kearsipan Daerah Kota Tasikmalaya. Selain itu, penelitian dilakukan dengan menggunakan *tools* yang lebih banyak dibandingkan dengan penelitian-penelitian terkait tersebut.