BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Ada beberapa penelitian terkait dengan perancangan *enterprise architecture* menggunakan Zachman *framework.*

2.1.1 Penerapan Zachman *Framework* Dalam Merancang Sistem Pelaporan Kerusakan Komputer [5]

Penelitian ini menjelaskan tentang perancangan *enterprise architecture planning* menggunakan Zachman *framework* untuk sistem pelaporan kerusakan komputer. Studi kasus ini dilakukan di STMIK AMIKOM Yogyakarta. *Enterprise architecture planning* (EAP) dikembangkan menggunakan perspektif *planner* (*scope), owner* (*busniness model).* Hasil dari penelitian ini adalah *blueprint* yang menjelaskan kebutuhan perencanaan pengembangan sistem. Kekurangan dari penelitian ini adalah *output blueprint* yang dihasilkan hanya berisi informasi perencanaan *enterprise architecture* sebatas dari perspektif *planner* dan *owner.*

2.1.2 Mendefinisikan *Enterprise Architecture Planning* Dalam Perencanaan Integrasi Sistem Informasi Perpustakaan Sekolah [8]

Penelitian ini menjelaskan tentang pembuatan *enterprise architecture* yang berfungsi untuk mengintegrasikan sistem informasi perpustakaan antar sekolah. Studi kasus ini dilakukan di SMUN 1 Tasikmalaya, SMU Siliwangi Tasikmalaya, SMK MJPS Tasikmalaya, SMK Bina Lestari PUI Tasikmalaya. Kelebihan penelitian ini dibandingkan penelitian pada jurnal sebelumnya adalah penggunaan analisis SWOT dan perhitungan nilai IFAS dan EFAS sebelum melakukan pembuatan *enterprise architecture.* Hasil akhir dari penelitian ini adalah berupa *blueprint* arsitektur integrasi sistem informasi perpustakaan. Namun informasi dari

blueprint ini belum cukup untuk dijadikan landasan pengembangan sistem karena hanya berupa rencana (*planning).*

2.1.3 Information Systems Architecture Online Learning in School with the Zachman Framework [9]

Penelitian ini bertujuan untuk mengembangkan *e-learning* yang dapat bermanfaat bagi seluruh sekolah di wilayah Sumatra Selatan. Perencanaan dan analisis sistem *e-learning* dilakukan berdasarkan *enterprise architecture* dengan Zachman *framework.* Studi kasus dilakukan pada SMA, SMK, MA, MAK yang ada di Sumatra Selatan. Hasil akhir dari penelitian dan studi kasus ini adalah tercipta sebuah struktur dasar atau pemodelan sistem informasi yang dapat mendukung integrasi, akses, pengembangan, manajemen, dan perubahan-perubahan yang terjadi pada arsitektur sistem informasi.

2.1.4 Modelling And Design E-Commerce SMI *Sector Using* Zachman *Framework* [10]

Penelitian ini menjelaskan tentang solusi *e-commerce* bagi SMI (industri menengah kecil) yang ada di Bantul, Yogyakarta. Selama ini SMI (industri menengah kecil) yang ada di Bantul sudah memiliki blog, website masing-masing, namun belum ada satu toko online special yang dapat menyatukan mereka dalam memasarkan produknya. Hasil akhir dari penelitian dan studi kasus ini adalah model dan desain sistem *e-commerce*. Tujuan setelah sistem dibuat yaitu penyediaan informasi yang bermanfaat dan dapat diterima oleh organisasi SMI (industri menengah kecil). Hasil tes dengan presentasi 100% untuk tes analisis dan tes manfaat pada prototype menunjukkan bahwa sistem ini sangat direkomendasikan,karena responden tes / *user* menunjukan tingkat kepuasan *user* yang sangat baik dengan nilai yang tertera pada tabel 2.1.

Tabel 2.1*Testing Value*

No	User	Nilai
1	Pembeli produk	3.3

2	Toko SMI (industri menengah kecil)	3.1
3	Karyawan	3.0

Analisis dalam penelitian ini dilakukan berdasarkan perspektif *owner (business model)* dan *designer (system model).* Pembangunan sistem bersifat *non object oriented* sehingga pemodelan sistem dibangun dengan menggunakan context diagram dan relasi data digambarkan dengan menggunakan *entity relationship diagram.* Penelitian ini menampilkan *screenshoot prototype* sistem yang dibangun. Walaupun hasil analisis dapat dijadikan acuan dalam pembangunan *prototype* sistem, namun informasi yang terdapat pada *blueprint* penelitian ini masih kurang lengkap karena untuk membangun sebuah *prototype* seharusnya melibatkan perspektif *planner (scope)* dan *designer (system model)* sebagai dasar analisis *blueprint.*.

2.1.5 Analisis Dan Pengembangan Sistem Informasi Akademik Dengan Permodelan Enterprise Architecture Zachman Framework Pada Politeknik Jambi [11]

Penelitian ini menjelaskan tentang analisis dan pengembangan sistem informasi akademik dengan permodelan enterprise architecture dengan Zachman *framework.* Studi kasus dilakukan pada Politeknik Jambi. Hasil akhir dari penelitian ini adalah pemodelan *enterprise architecture* yang menggambarkan setiap langkah pengerjaan sistem dan dokumen arsitektur sistem informasi yang dapat digunakan sebagai landasan untuk pengembangan selanjutnya sistem informasi di Politeknik Jambi. Penelitian ini juga menampilkan *screenshoot prototype* sistem. Penelitian berbasis *object oriented* sehingga pemodelan sistem dibuat menggunakan UML. Kekurangan dari penelitian ini adalah penggunaan perspektif yang tidak konsisten. Pada kolom *what* berisi perspektif *planner (scope), owner (business model), designer (system model), builder (technology model),* pada kolom *how* dan *where* berisi perspektif *planner (scope), owner (business model), builder (technology model),* kolom *who* berisi perspektif *planner (scope), owner (business model), builder (technology model)* dan *user (functioning),* kolom *why* berisi perspektif

planner (scope), owner (business model), dan kolom *when* berisi perspektif *planner (scope), owner (business model), designer (system model)* dan *user (functioning).*

Tabel 2.2 Penelitian Terkait

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	A. A. Slameto, E. Utami, A. A. Pangera, 2013	Perancang sistem pelaporan kerusakan komputer	Zachman *framework* pada sistem pelaporan kerusakan komputer	Dihasilkan *blueprint* rancangan sistem informasi yang dilihat dari sudut pandang *planner* dan *owner* yang dipetakan dalam matrik Zachman.
2.	Agung Baitul Hikmah, 2014	Mendefinisikan EAP dalam integrasi sistem informasi perpustakaan sekolah	Zachman *framework* pada sistem informasi perpustakaan sekolah	Dihasilkan *blueprint* dan model standar untuk sistem informasi perpustakaan di Tasikmalaya.
3	Yana Hendriana, Rusydi Umar, Andri Pranolo, 2015	Pemodelan dan desain *e-commerce* untuk industri tingkat menengah kebawah	Zachman *framework*	Analisis dalam bentuk bisnis baru ERD, arsitektur informasi, arsitektur data, aliran data diagram, arsitektur aplikasi, desain sistem logistic, jaringan, arsitektur teknologi, struktur organisasi, arsitektur *interface* dan *user interface*.
4	Hatta Wijaya, M. I. Herdiansyah, A. Haidar Mirza, 2016	Perancangan sistem pembelajaran sekolah (SMA, SMK, MAK,	Zachman *framework* pada arsitektur sistem informasi pembelajaran sekolah berbasis online	Dihasilkan *blueprint* dari arsitektur sistem informasi pembelajaran online.

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
		MA) berbasis online		
5	Akhmad Faisal Husni, 2016	Analisis dan pengembangan sistem informasi akademik pada politeknik Jambi	Pemodelan *enterprise architecture* dengan Zachman *framework*	Dihasilkan pemodelan *enterprise architecture* yang menggambarkan setiap langkah pengerjaan sistem dan dokumen arsitektur sistem informasi yang dapat digunakan sebagai landasan untuk pengembangan selanjutnya sistem informasi di Politeknik Jambi.

Berdasarkan beberapa penelitian terkait tersebut, dapat disimpulkan bahwa perancangan *enterprise architecture* dengan menggunakan Zachman *framework* dapat memberikan gambaran mengenai sistem yang akan dibangun, pihak-pihak yang terkait dengan sistem dan hal-hal yang dibutuhkan dalam pembangunan sistem melalui informasi yang tertera dalam matrik dua dimensi dari Zachman *framework* dan menghasilkan *blueprint* sistem yang bermanfaat untuk mengetahui langkah-langkah pembuatan sistem dan berguna sebagai landasan dasar pengembangan sistem selanjutnya. Fungsi dari *blueprint* yang dihasilkan sangat dipengaruhi oleh perspektif yang digunakan saat penelitian. Perbedaan penelitian ini dengan penelitian sebelumnya adalah, dalam penelitian ini digunakan perspektif *planner (scope), owner (business model), designer (system model), builder (technology model)* pada enam kolom *what, how where, who, when* dan *why*, sehingga dihasilkan *blueprint* yang bersifat *principal* dan dapat digunakan sebagai acuan pembuatan *prototype* pengembangan sistem.

2.2 Landasan Teori

Untuk mendukung penelitian yang akan dilakukan, berikut ini merupakan teori-teori pendukung penelitian yang dibutuhkan dari berbagai sumber literatur baik buku, jurnal, ataupun *prosiding.*

2.2.1 *Enterprise Architecture*

Enterprise merupakan organisasi maupun sekumpulan organisasi yang memiliki kesamaan tujuan, seperti organisasi pemerintahan, unit departemen dan lain-lain. Arsitektur adalah ilmu dan seni perancangan atau perencanaan. *Enterprise architecture* merupakan pendekatan yang menjelaskan rencana dalam membangun sistem yang logis dan menyeluruh untuk merancang dan menjalankan sistem secara menyeluruh bersamaan dengan komponen-komponennya [12]. Dalam sumber lain disebutkan bahwa *enterprise architecture* merupakan deskripsi tentang bagaimana merealisasikan tujuan organisasi kedalam proses bisnis, dengan menggunakan bantuan teknologi informasi [13]. Beberapa metodologi pengembangan *enterprise architecture* antara lain :

1. *Enterprise Architecture Planning* (EAP)
2. TOGAF *Architecture Development Method* (TOGAF ADM)
3. *Enterprise Architecture Strategy* (EAS)
4. *Basic Enterprise Architecture Methodology* (BEAM)

Komponen utama *enterprise architecture* terdiri dari :

Tabel 2.3 Komponen *Enterprise Architecture*

No	Komponen Arsitektur	Penjelasan
1.	Arsitektur bisnis	Penentuan proses bisnis yang menjadi motivator untuk komponen lain.
2.	Arsitektur informasi	Sekumpulan entitas yang mendukung proses bisnis.

3.	Arsitektur aplikasi	Penentuan jenis aplikasi utama dan aplikasi pendukung dalam melakukan proses bisnis.
4.	Arsitektur teknologi	Penentuan platform teknologi untuk mempersiapkan lingkungan sistem.

Keuntungan yang dapat diperoleh dari penerapan *enterprise architecture* antara lain proses operasional teknologi informasi menjadi semakin efisien, resiko investasi jangka panjang dapat diminimalisir, proses pengadaan informasi menjadi lebih cepat, mudah dan hemat [14], perusahaan dapat mengatasi dinamika bisnis dengan mengintegrasi, mengatur dan menganalisa sistem [15]

2.2.2 Framework

Framework atau kerangka kerja merupakan kumpulan dari fungsi, prosedur dan *class* yang siap digunakan, bertujuan untuk memberi kemudahan bagi *programmer* sehingga tidak perlu membuat *class* dari awal. Beberapa jenis *framework* antara lain :

a. *The Open Group Archiectural Framework* (TOGAF)
b. *Treasury Enterprise Architecture Framework* (TEAF)
c. *Federal Enterprise Architecture Framework* (FEAF)
d. *DoD Architecture Framework* (DoDAF)
e. Zachman *Framework*

Semua *framework* memiliki karakteristik, kekurangan dan kelebihan masing-masing. Berikut merupakan perbandingan antara kelima *framework* tersebut. Pertama dilihat dari segi *views* atau perspektif atau sudut pandang yang dijabarkan dalam tabel 2.4 berikut :

Tabel 2.4 Perbandingan *Framework* Dilihat Dari *Views* [6]

Framework	***Planner***	***Owner***	***Designer***	***Builder***	***Sub-contractor***	***User***
Zachman	*Scope*	*Bussines s Model*	*Systems Model*	*Technology Model*	*Detailed Representati on*	*Function ing System*
DoDAF	*All View*	*Operatio nal View*	*Systems View*	*Technical View*		
FEAF	*Objective s/Scope Planner' s View*	*Enterpris e Model Owner's View*	*Informatio n Systems Model Designer's View*	*Technology Model Builder's View*	*Detailed Spesification Subcontract or's View*	
TEAF	*Planner*	*Owner*	*Designer*	*Builder*		
TOGAF		*Business Architect ure View*	*Technical rchitecture Views*			

Perbandingan kedua dilihat dari segi *abstraction* yaitu berdasarkan pertanyaan *what, how, where who when why* yang dijabakan dalam tabel 2.5 :

Tabel 2.5 Perbandingan *Framework* Dilihat Dari *Abstraction* [6]

Framework	***What***	***How***	***Where***	***Who***	***When***	***Why***
Zachman	*Data*	*Function*	*Network*	*People*	*Time*	*Motiv ation*
DoDAF	*Data (mission) Logical Data Model*	*Funtion/ Tracebility Functional Effectiveness*	*Physical connectivity plasavailabilit y of off-the-self solutions*	*Organizati onal Relationshi p*		
FEAF	*Data Architectur e*	*Applications Architecture (activities=h ow)*	*Technology Architecture (location=whe re)*			

	(entities=what)					
TEAF	*Information View*	*Functional View*	*Infrastructure View*	*Organizational View*		
TOGAF		*Decision Making guidance*		*IT resource guidance*		

Selanjutnya dilihat dari segi cakupan SDLC yaitu *planning, analysis, design, implementation* sampai kepada tahap *maintenance* atau perawatan, yang dijabarkan dalam tabel 2.6 berikut :

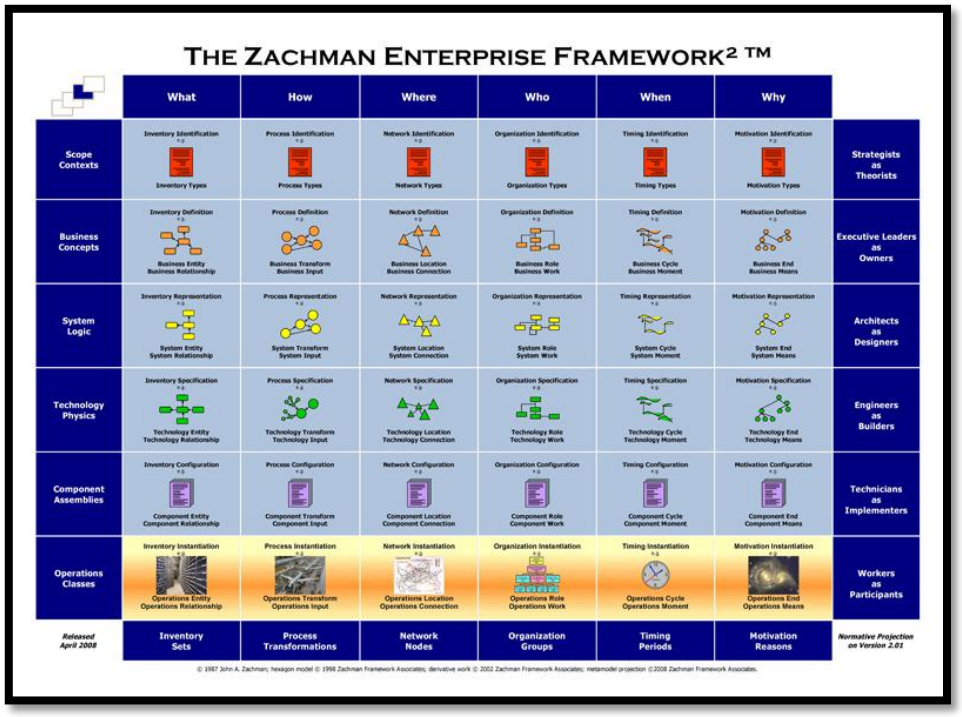
Tabel 2.6 Perbandingan *Framework* Dilihat Dari Cakupan SDLC [6]

Framework	***Planning***	***Analysis***	***Design***	***Implementation***	***Maintenance***
Zachman	*Yes*	*Yes*	*Yes*	*Yes*	*No*
DoDAF	*Yes*	*Yes*	*Yes*	*Describes Final Products*	*No*
FEAF	*Yes*	*Yes*	*Yes*	*Yes*	*Detailed Subcontractor's View*
TEAF	*Yes*	*Owner's Analysis*	*Yes*	*Yes*	*No*
TOGAF		*Principles that support decision making across enterprise, provide guidance of IT resources, support architecture principles for design and implementation.*			

Berdasarkan ketiga perbandingan diatas, dapat ditarik suatu kesimpulan bahwa Zachman *framework* merupakan *framework* yang paling komprehensif dan lengkap dibandingkan dengan *framework* yang lainnya. Akan tetapi, pemilihan *framework* harus disesuaikan dengan kebuthan sistem yang akan dibangun, sehingga *output* dan manfaat dari *framework* dapat lebih optimal.

2.2.2.1 Zachman *Framework*

Zachman *framework* merupakan *framework k*atau kerangka kerja yang diciptakan oleh John Zachman. Konsep fundamental Zachman *framework* dibuat pada bulan Juni tahun 1984, dan selalu mengalami evolusi dari tahun ke tahun [16]. Zachman *framework* didefinisikan dengan matrik dua dimensi, yang terdiri atas enam baris dikalikan enam kolom.



Gambar 2.1 Zachman *Enterprise Framework* [16]

Dimensi pertama Zachman digambarkan dalam baris yang terdiri dari enam perspektif yang dijelaskan pada tabel 2.7 berikut :

Tabel 2.7 Penjelasan Perspektif Pada Dimensi Pertama Zachman *Framework* [17]

No	Perspektif / Sudut Pandang	Penjelasan
1.	*The Planner Perspective (Scope Context)*	Daftar lingkup penjelasan unsur bisnis yang dikenali oleh para ahli strategi sebagai ahli teori yang menetapkan

		objek dalam pembahasan seperti latar belakang lingkup dan tujuan *enterprise*.
2.	*The Owner Perspective (Business Concept)*	Model semantik keterhubungan bisnis antara komponen-komponen bisnis yang didefenisikan oleh pimpinan eksekutif sebagai pemilik, mendefinisikan bentuk dari produk / model bisnis.
3.	*The Designer Perspective (System Logic)*	Model logika yang lebih rinci yang berisi kebutuhan dan desain batasan sistem yang direpresentasikan oleh para arsitek sebagai desainer.
4.	*The Builder Perspective (Technology Physics)*	Model fisik yang mengoptimalkan desain untuk kebutuhan spesifik dalam batasan teknologi spesifik, orang, biaya dan lingkup waktu yang dispesifikasikan oleh *engineer* sebagai *builder*.
5.	*The Implementer Perspective (Component Assemblies)*	Teknologi khusus, tentang bagaimana komponen dirakit dan dioperasikan, dikonfigurasikan oleh teknisi sebagai *implementator*.
6.	*The Participant Perspective (Operation Classes)*	Kejadian-kejadian sistem berfungsi nyata yang digunakan oleh para teknisi sebagai *participant*.

Dimensi kedua Zachman digambarkan dalam kolom yang terdiri dari enam klasifikasi yang dijelaskan dengan *what, how, where, who, when dan why*, berikut merupakan penjelasannya :

1. *Objective* / *Scope* menurut perspektif *planner*
    a. Kolom *What* membahas *high level data class* terkait dengan masing-masing fungsi.
    b. Kolom *How* berisikan daftar semua proses yang diketahui
    c. Kolom *Where* berisikan lokasi yang sangat penting untuk organisasi, bisa menjadi besar dan kecil.
    d. Kolom *Who* berisikan daftar dari semua unit organisasi.
    e. Kolom *Why* menjabarkan tujuan utama organisasi.
    f. Kolom *When* berisikan event-event siklus waktu yang terkait dengan masing-masing fungsi.
2. *Enterprise Model* menurut perspektif *owner*
    a. Kolom *What* berisikan deskripsi pengelolaan material dan hubungannya.
    b. Kolom *How* berisikan deskripsi proses dari *input*, proses, dan *output*.
    c. Kolom *Where* berisikan identifikasi lokasi perusahaan.
    d. Kolom *Who* berisikan identifikasi peran perusahaan.
    e. Kolom *Why* mengidentifikasi tingkatan dan tujuan yang mendukung tujuan utama.
    f. Kolom *When* berisikan identifikasi dan deskripsi kejadian dan siklus yang berhubungan dengan waktu.
3. *System Model* menurut perspektif *designer*
    a. Kolom *What* berisikan deskripsi entitas dan hubungannya tanpa memperhatikan implementadi fisik dan teknis.
    b. Kolom *How* berisikan deskripsi transisi proses, dinyatakan sebagai ungkapan kata kerja tanpa memperhatikan implementasi teknis.
    c. Kolom *Where* berisikan deskripsi lokasi yang digunakan untuk mengakses dan mengubah entitas tanpa memperhatikan implementasi teknis.
    d. Kolom *Who* berisikan identifikasi peran dan hubungannya ke peran lain tanpa memperhatikan implementasi teknis.
    e. Kolom *Why* berisikan berbagai macam *policy* prosedur dan standar yang terkait dengan *business rule.*

f. Kolom *When* berisikan deskripsi keadaan yang berhubungan dengan kejadian yang lain pada *sequence.*

4. *Technology Model* menurut perspektif *builder*
   a. Kolom *What* berisikan tipe kebutuhan sistem manajemen *database* yang sesuai dengan model data.
   b. Kolom *How* berisikan spesifikasi dari aplikas-aplikasi yang beroperasi pada suatu *platform* teknologi tertentu.
   c. Kolom *Where* berisikan deskripsi spesifikasi dari perangkat jaringan dan hubungannya dengan batasan fisik sistem.
   d. Kolom *Who* berisikan identifikasi hak akses masing-masing *user*.
   e. Kolom *Why* identifikasi aturan nama dan logika terstruktur untuk pengujian aturan sistem.
   f. Kolom *When* berisikan transformasi keadaan-keadaan terhadap minat perusahaan.
5. *As Build* menurut perspektif *programmer / sub-contractor*
   a. Kolom *What* berisikan definisi data yang sesuai dengan model logika.
   b. Kolom *How* berisikan fungsi program yang di *coding* untuk dioperasikan pada suatu *platform* teknologi.
   c. Kolom *Where* berisikan konfigurasi perangkat jaringan untuk disesuaikan dengan spesifikasi node.
   d. Kolom *Who* berisikan identifikasi hak akses masing-masing *user*.
   e. Kolom *Why* berisi berbagai macam *business rule* yang sesuai dengan standar teknologi tertentu.
   f. Kolom *When* berisikan pendefinisian *timing* untuk menentukan urutan aktivitas proses.
6. *Functioning Enterprise* menurut perspektif *user*
   a. Kolom *What* berisikan konten dan nilai data yang tersimpan di *database.*
   b. Kolom *How* berisikan instruksi manual untuk menjalankan perangkat komputer dan sistem informasi.
   c. Kolom *Where* berisikan pesan-pesan yang terkirim dan yang diterima.

d. Kolom *Who* berisikan berbagai macam personel dan *stakeholder* yang bekerja sesuai dengan *job desc* nya.
e. Kolom *Why* berisi informasi karakteristik operasi pada suatu teknologi berdasarkan standar tertentu.
f. Kolom *When* berisikan pendefinisian waktu melakukan aktivitas berdasarkan urutan waktu tertentu.

Zachman *framework* diperkenalkan sebagai standar *framework* kelas dunia dan telah digunakan oleh banyak organisasi besar di dunia seperti Johnson and Johnson, Hewlett-Packard, Microsoft dan lainnya [18]. Kelebihan dari *framework* ini dibandingkan dengan *framework* lain adalah:

1. Zachman *framework* mengklasifikasikan kebutuhan arsitektur *enterprise system* dengan sangat detail, dan merupakan yang paling detail dibandingkan dengan *framework* lain [7].
2. Zachman *framework* merupakan standar secara *de-facto* untuk mengklasifikasikan artefak arsitektur *enterprise.*
3. Zachman *framework* menggambarkan secara parallel baik dari sisi *engineering* maupun konstruksi.
4. Zachman *framework* dikenal secara luas sebagai *tool* manajemen untuk memeriksa kelengkapan arsitektur dan *maturity level*.

2.2.3 Perancangan Sistem

Alat pemodelan yang berfungsi untuk mendokumentasikan suatu sistem berbasis obyek atau object-oriented, dan sekaligus menjadi standar pemodelan sistem saat ini adalah Unified Modeling Language atau yang sering disingkat UML [19]. Merupakan sebuah bahasa yang berdasarkan grafik atau gambar (pemodelan) untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah pengembangan sistem [20]. Pemodelan digunakan untuk menyederhanakan permasalahan yang kompleks sehingga lebih mudah dipelajari. Sumber lain menyebtkan bahwa alat pemodelan yang berfungsi untuk mendokumentasikan suatu sistem berbasis obyek dan sekaligus menjadi standar pemodelan sistem saat ini adalah *Unified Modeling Language* atau yang sering disingkat UML.

UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu.

2.2.3.1 Jenis UML

Unified Modeling Language (UML) terdiri dari empat belas jenis diagram, beberapa jenis diagram UML dijelaskan dalam tabel 2.8. Penelitian ini menggunakan empat jenis UML yaitu *use case, activity diagram, sequence diagram* dan *class diagram* untuk menggambarkan pemodelan sistem.

Tabel 2.8 *View* dan Diagram UML [20]

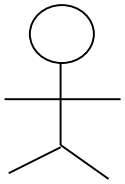
Major Area	***View***	***Diagrams***	***Main Concepts***
Structural	*Static view*	*Class diagram*	*Class, association, generalization, dependency, realization, interface*
	Use case view	*Use case diagram*	*Use case, actor, association, extend, include, use case generalization*
	Implementation view	*Component diagram*	*Component, interface, dependency, realization*
	Deployment view	*Deployement diagram*	*Node, component, dependency, location*
Dynamic	*State machine view*	*Statechart diagram*	*State, event, transition, action*

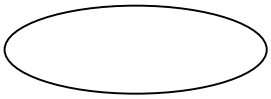
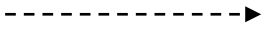
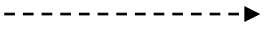
	Activity view	*Activity diagram*	*State, activity, completion transition, fork, join*
	Interaction view	*Sequence diagram*	*Interaction, object message, activation*
		Collaboration diagram	*Collaboration, interaction, collaboration role, message*
Model management	*Model management view*	*Class diagram*	*Package, subsystem, model*
Extensibility	*All*	*All*	*Constraint, stereotype, tagged value*

2.2.3.2 Use Case Diagram

Diagram pertama yang digunakan dalam penelitian ini adalah *use case. Use case diagram* menunjukkan interaksi antara sistem dan lingkungannya. *Usecase diagram* adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Simbol-simbol *use case* dijelaskan dalam tabel 2.9.

Tabel 2.9 Simbol *Use Case Diagram* [21]

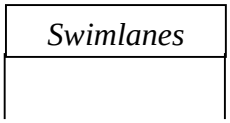
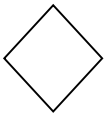
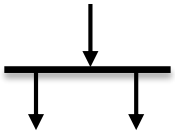
Simbol	Keterangan
	***Actor*** Menunjukkan user yang akan berinteraksi dengan sistem informasi yang akan dibuat.

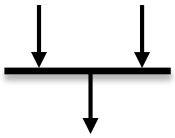
	***Use Case*** Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor.
	Generalization Menunjukkan hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	Association Komunikasi antara dua aktor dan use case yang berpartisipasi pada use case.
<< include>> 	***Include*** Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.
<<extend>> 	***Extend*** Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan.

2.2.3.3 Activity Diagram

Diagram kedua yang digunakan dalam penelitian ini adalah *activity diagram. Activity diagram* menggambarkan aktivitas yang terlibat dalam sebuah proses atau dalam pemrosesan data. Tabel 2.10 berisi penjelasan simbol-simbol yang digunakan pada *activity diagram.*

Tabel 2.10 Simbol *Activity Diagram* [21]

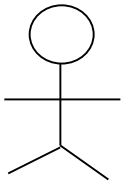
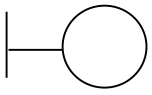
Simbol	Keterangan
	***Starting activity (Pseudo)*** Menunjukkan awal dari suatu aktivitas pada diagram
	***Ending activity (Pseudo)*** Menunjukkan akhir dari suatu aktivitas pada diagram
	Transition arrow Menunjukkan kondisi transisi antar aktivitas
	***Swimlane heading*** Menunjukkan siapa saja aktor yang terlibat pada aktivitas diagram
	***Activity*** Menunjukkan aktivitas apa saja yang terjadi dalam diagram
	***Decision activity*** Menunjukkan pengecekan terhadap suatu kondisi.
	***Synchronizarion bar (Split)*** Untuk menghubungkan satu proses sebelumnya dengan dua atau lebih proses setelahnya.

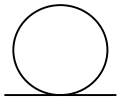
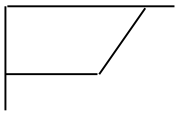
	***Synchronizarion bar (Join)*** Untuk menghubungkan dua atau lebih proses sebelumnya dengan satu proses setelahnya

2.2.3.4 Sequence Diagram

Diagram selanjutnya yang digunakan dalam penelitian ini adalah *sequence diagram*. *Sequence diagram* adalah diagram yang digunakan untuk mendefinisikan *input* dan *output* serta urutan interaksi antara pengguna dan sistem. Simbol-simbol *sequence diagram* dijelaskan dalam tabel 2.11.

Tabel 2.11 Simbol *Sequence Diagram* [21]

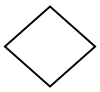
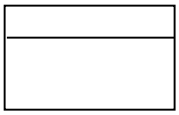
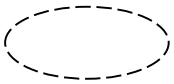
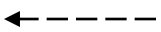
Simbol	Keterangan
	***Aktor*** Menunjukkan user yang akan berinteraksi dengan sistem.
	Lifeline Lifeline mengindikasikan keberadaan sebuah objek dalam basis waktu
	***Activation*** Activation dinotasikan sebagai sebuah persegi panjang yang digambar pada lifeline. Activation mengindikasikan sebuah objek yang akan melakukan aksi.
	***Boundary*** Boundary terletak diantara sistem. Semua form, laporan-laporan, antarmuka ke perangkat keras seperti printer dan antarmuka ke sistem lainnya adalah termasuk dalam kategori.

	***Control*** Control berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, dan penanganan kesalahan.
	***Entity*** Digunakan menangani informasi yang mungkin akan disimpan secara permanen. Entity bisa juga merupakan sebuah tabel pada struktur basis data.
	***Message*** Pesan digambarkan dengan anak panah horizontal yang menghubungkan antara activation. Pesan mengindikasikan komunikasi antar objek.
	***Self-Message*** Panggilan mandiri yang mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.
	***Loop*** *Operator loop* adalah fragmen yang dapat mengeksekusi berulang kali dan penjaga menunjukkan iterasi.

2.2.3.5 *Class Diagram*

Diagram terakhir yang digunakan dalam penelitian ini adalah *class diagram.* Diagram yang menunjukkan sekumpulan *class, interface* dan *collaboration* serta *relationships*. Diagram ini biasa ditemukan pada pemodelan *object oriented* system. *Class diagram* menunjukkan *view* desain statik dari sebuah sistem. *Class diagram* yang termasuk *active class* menunjukkan view proses statik sebuah sistem. Tabel 2.12 berisi keterangan simbol-simbol yang digunakan pada *class diagram.*

Tabel 2.12 Simbol *Class Diagram* [21]

Simbol	Keterangan
	Generalization Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
	***Nary Association*** Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	***Class*** Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	***Collaboration*** Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan sesuatu yang terukur bagi aktor.
	***Realization*** Operasi yang benar-benar dilakukan oleh suatu objek.
	Depedency Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

_____	***Association*** Apa yang menghubungkan objek satu dengan objek lainnya.

2.2.4 Jaringan

Jaringan merupakan penghubung antara banyak komputer dengan perangkat peripheral lainnya. Sebelum ada jaringan, komputer hanya dapat digunakan secara *independent* sehingga proses pertukaran informasi menjadi sangat terbatas. Dengan adanya jaringan, proses pertukaran informasi menjadi lebih mudah tak terbatas ruang dan waktu [22]. Dilihat dari pola pengoperasian, jaringan komputer dibagi menjadi *peer to peer* dan *client server*. Pada jaringan *peer to peer*, seluruh komputer bisa menjadi *server* sekaligus *client*, sedangkan pada jaringan *client server* ada satu atau lebih komputer yang digunakan sebagai *server*, dan komputer lainnya sebagai *client* [23].

2.2.5 Konsep Dasar *Database*

Database atau basis data merupakan struktur yang terbagi atas *database* flat dan *database* relasional. *Database* flat merupakan *database* dua dimensi yang sering disebut sebagai file flat. Contoh dari *database* flat antara lain dokumen Word dan *spreadsheet* Excel.

Database relasional lebih mudah dipahami karena memiliki bentuk yang lebih sederhana dengan pengoprasian data yang lebih mudah dibandingkan *database* flat. *Database* relasional terdiri atas tabel-tabel yang berisikan kolom dan baris. Terdapat sebuah kolom yang berfungsi untuk mendefinisikan jenis informasi yang harus disimpan.

2.2.5.1 Kelebihan dan Kekurangan *Database*

Kelebihan dari *database* yaitu:

1. Data independen, sehingga jika dilakukan perubahan pada aplikasi maka data tidak akan ikut berubah
2. Data lebih konsisten dan meminimalisir *redundancy* atau data rangkap [24].

3. Memungkinkan data sharing atau berbagi penggunaan data tanpa mengurangi konsistensi dari data tersebut karena perubahan pada basis data tersedia untuk semua aplikasi yang terhubung dengan basis data tersebut [25].

Kekurangan dari *database* yaitu:

1. Memanfaatkan *database* berarti akan mengurangi ruang penyimpanan dalam *harddisk.*
2. Kecepatan pemrosesan data akan berkurang.

2.2.5.2 Komponen-Komponen *Database*

Komponen-komponen yang terdapat dalam *database* antara lain :

1. Entitas

   Merupakan hal yang menyatakan objek atau kejadian. Di dalam tabel, entitas berfungsi sebagai nama tabel.

2. Atribut

   Merupakan item yang menjadi bagian dari entitas. Di dalam tabel, atribut berfungsi sebagai *header* tabel.

3. Relasi

   Merupakan hubungan yang terjadi antar entitas dalam suatu *database*. Relasi ditentukan oleh entitas yang terkait. Tipe relasi terdiri dari relasi *One to One*, relasi *One to Many* dan relasi *Many to Many*.

   Relasi *One to One* menghubungkan satu anggota entitas A dengan satu anggota entitas B. Contoh dari relasi ini adalah relasi antara Penduduk dan KTP dimana satu penduduk hanya boleh memiliki satu buah KTP.

   Relasi *One to Many* menghubungkansatu anggota dari entitas A dengan banyak anggota dari entitas B. Contoh dari relasi ini adalah relasi antara dosen wali dan mahasiswa, dimana satu dosen wali dapat mengampu banyak mahasiswa dalam perwaliannya, namun satu mahasiswa hanya diijinkan memiliki satu dosen wali saja.

Sedangkan relasi *Many to Many* menghubungkan banyak anggota dari relasi A dengan banyak anggota dari relasi B. Contoh dari relasi ini adalah relasi antara matakuliah dan mahasiswa, dimana banyak matakuliah dapat diambil oleh banyak mahasiswa.