

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Terdapat beberapa penelitian terkait mengenai permodelan *Enterprise Architecture* dengan menggunakan Framework Zachman, yaitu:

Penelitian yang dilakukan oleh Antonius Wahyu Sudrajat[2] menemukan adanya masalah pada sistem manajemen penyusunan anggaran yang masih menggunakan cara konvensional sehingga terdapat banyak masalah seperti keterlambatan membuat laporan pelaksanaan, mencairkan dana kegiatan, membuat kontrak, kesalahan nomor pada kode rekening, dan lain-lain. Untuk mengatasi permasalahan tersebut dibuat suatu perancangan arsitektur Sistem Manajemen Penyusunan Anggaran Keuangan Daerah dalam skala nasional dengan menggunakan Framework Zachman yang terfokus pada baris *Scope* (Perspektif *Planning*), *Business Model* (Perspektif *Owner*), dan *System Model* (Perspektif *Builder*) dengan kolom *What* (*Data*), *How* (*Function*), dan *Where* (*Network*). Hasil dari penelitian ini adalah rancangan arsitektur Sistem Manajemen Penyusunan Anggaran Keuangan Daerah.

Sementara penelitian yang dilakukan oleh Roni Yunis dan Theodora[3] menemukan adanya permasalahan yaitu pengembangan bisnis dan TI pada perusahaan besar yang tidak dilakukan sesuai kebutuhan. Penelitian ini juga membahas mengenai beberapa framework yang sering digunakan seperti TOGAF, FEAF, TEAF, dan Zachman serta menerapkan Framework Zachman dalam memodelkan EA yang terfokus pada baris *Scope* (Perspektif *Planning*), *Business Model* (Perspektif *Owner*), dan *System Model* (Perspektif *Builder*) dengan kolom *What* (*Data*), dan *How* (*Function*). Hasil dari penelitian ini yaitu penjelasan berbagai macam framework yang dapat digunakan sesuai kebutuhan serta jenis *enterprise architecture* seperti apa yang ingin dimodelkan.

Tabel 2.1 Penelitian Terkait

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	Antonius Wahyu Sudrajat, 2015	Sistem manajemen penyusunan anggaran yang masih menggunakan cara konvensional seperti keterlambatan membuat laporan pelaksanaan, mencairkan dana kegiatan, membuat kontrak, kesalahan nomor pada kode rekening, dan lain-lain.	Framework Zachman yang terfokus pada baris <i>Scope</i> (Perspektif <i>Planning</i>), <i>Business Model</i> (Perspektif <i>Owner</i>) , dan <i>System Model</i> (Perspektif <i>Builder</i>) dengan kolom <i>What (Data)</i> , <i>How (Function)</i> , dan <i>Where (Network)</i> .	Rancangan arsitektur Sistem Manajemen Penyusunan Anggaran Daerah
2.	Roni Yunis dan Theodora, 2012	Pengembangan bisnis dan TI pada perusahaan besar yang tidak dilakukan sesuai kebutuhan.	Framework Zachman yang terfokus pada baris <i>Scope</i> (Perspektif <i>Planning</i>), <i>Business Model</i> (Perspektif <i>Owner</i>) , dan <i>System Model</i> (Perspektif <i>Builder</i>) dengan	Penjelasan berbagai macam framework yang dapat digunakan sesuai kebutuhan serta jenis <i>enterprise architecture</i> seperti apa

			kolom <i>What</i> yang ingin (<i>Data</i>), dan <i>How</i> dimodelkan. (<i>Function</i>).
--	--	--	---

Dari penelitian yang telah dilakukan di atas maka dapat disimpulkan bahwa Framework Zachman dapat digunakan sebagai solusi pemecahan masalah dalam memodelkan sistem informasi. Oleh sebab itu, peneliti tertarik menggunakan Framework Zachman untuk membangun Sistem Informasi Akademik di SMA PL Don Bosko Semarang dengan membatasi penelitian pada Perspektif *Scope*, *Business Model*, dan *System Model* agar menghasilkan *blueprint* (cetak biru) pemodelan Sistem Informasi Akademik yang baik, terstruktur, dan terintegrasi.

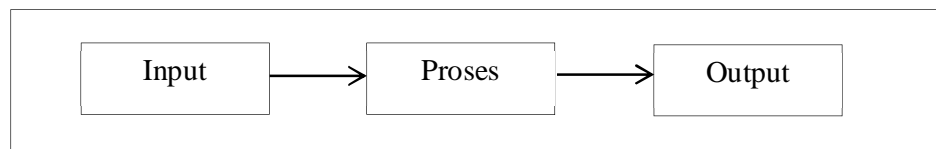
2.2 Konsep Dasar Sistem Informasi dan Sistem Informasi Akademik

2.2.1 Sistem

Sistem dalam suatu organisasi sangatlah penting karena memberikan aturan atau intruksi secara runtut sehingga segala proses yang terjadi di dalamnya berjalan dengan baik dan lancar. Sistem dapat menyelesaikan suatu sasaran yang ingin dicapai dan melakukan berbagai kegiatan karena terdiri dari jaringan kerja yang terbentuk dari prosedur dan saling berhubungan satu dengan yang lainnya[4].

Sistem sendiri juga dapat dikatakan sebagai hubungan yang tidak dapat dipisahkan antara unit satu dengan unit lainnya sehingga apabila suatu unit mengalami gangguan atau terganggu oleh sesuatu maka tentu akan mengganggu unit lainnya dalam mencapai tujuan yang telah ditetapkan dari awal[5].

Model umum dari suatu sistem terdiri dari masukan (*input*), proses (*process*), dan keluaran (*output*) seperti yang ditunjukkan pada gambar di bawah ini:



Gambar 2.1 Model Umum Sistem[6]

Di mana input merupakan kegiatan memasukan data mentah ke dalam sistem, kemudian data mentah tersebut diolah dan di proses agar mengeluarkan output berupa informasi penting. Model sistem di atas merupakan contoh dari sebuah sistem yang sangat sederhana karena sistem memiliki berbagai macam masukan dan memberikan hasil berupa berbagai keluaran dalam satu proses sekaligus.

2.2.2 Informasi

Tidak ada organisasi yang tidak membutuhkan informasi. Informasi merupakan salah satu aset penting dan berharga untuk dapat menentukan strategi organisasi selanjutnya secara tepat. Informasi bersumber dari hasil pengolahan sedemikian rupa atau perubahan bentuk dari data yang belum memiliki makna menjadi sesuatu atau informasi yang memiliki nilai dan memberikan pengetahuan bagi penerimanya.

Dapat dikatakan jika informasi merupakan hasil dari analisis terhadap berbagai macam data. Informasi memiliki fungsi sebagai dasar untuk pengambilan keputusan[6]. Sehingga, informasi dapat diartikan sebagai data yang diatur ke bentuk yang sesuai dengan kebutuhan organisasi tertentu dan dapat digunakan sebagai dasar pengambilan keputusan.

2.2.3 Sistem Informasi

Sistem informasi merupakan sistem yang terdapat pada organisasi yang menjembatani antara kebutuhan untuk mengolah transaksi harian, memberikan dukungan operasi yang sifatnya manajerial, dan menyediakan berbagai laporan yang dibutuhkan oleh organisasi. Fungsi dari sistem informasi yaitu untuk memberikan kemudahan dalam melaksanakan kegiatan utama organisasi. Sistem informasi terdiri dari berbagai elemen yang menjadi satu kesatuan dan saling berinteraksi secara teratur serta sistematis yang digunakan untuk menghasilkan aliran informasi sebagai pendukung dalam pengambilan keputusan sekaligus melakukan kontrol pada jalannya proses bisnis pada organisasi[7].

2.2.4 Sistem Informasi Akademik

Sistem informasi akademik dapat diibaratkan seperti tiang utama sebagai pengatur segala hal yang terkait dengan penyelenggaraan kegiatan akademik. Sistem Informasi Akademik merupakan sistem yang dikumpulkan dari berbagai macam data yang dikelola secara otomatis dengan menggunakan alat dan metode tertentu sehingga menghasilkan suatu informasi yang diperlukan untuk mendukung terlaksananya kegiatan akademis di suatu sekolah. Sistem Informasi akademik dapat berupa suatu *software* atau perangkat lunak yang digunakan untuk memberikan informasi dengan kegiatan akademis yang diharapkan kegiatan yang berhubungan dengan kegiatan administrasi akademis dapat dikelola dengan baik serta informasi yang dibutuhkan dapat didapatkan dengan mudah dan lebih cepat[8].

2.3 Konsep Dasar *Enterprise Architecture*

2.3.1 *Enterprise*

Enterprise terdiri dari sekumpulan organisasi yang berorientasi pada keuntungan (profit) maupun yang tidak (non profit atau nirlaba) seperti lembaga pemerintahan dan lembaga pendidikan yang setiap kegiatan atau aktivitasnya memiliki tujuan tertentu, mendukung cakupan bisnis serta misi yang ditetapkan dari awal[9].

2.3.2 *Architecture*

Architecture terdiri dari sekumpulan komponen yang saling terhubung dengan komponen lainnya yang menjadi dasar dari sistem suatu organisasi sebagai perancangan dan evaluasi. *Architecture* memberikan gambaran yang lebih jelas mengenai bagaimana detail bentuk konstruksi suatu sistem yang akan dibangun, bagaimana menyusun setiap komponen sistem, dan bagaimana mengintegrasikan seluruh komponen tersebut melalui aturan dan *interface* yang telah direncanakan[9].

2.3.3 Enterprise Architecture

Enterprise Architecture merupakan sekumpulan metode dan model yang logis dan digunakan oleh organisasi untuk mendesain serta menjalankan struktur organisasi *enterprise*, sistem informasi, infrastruktur, dan proses bisnisnya[9]. *Enterprise Architecture* memberikan jaminan pada organisasi bahwa sumber daya informasi yang mereka miliki akan digunakan sebagai pendukung strategi organisasi tersebut. Kebutuhan informasi dan komunikasi yang diuraikan dari sasaran dan tujuan organisasi merupakan dasar untuk membuat rencana *enterprise architecture*.

Fungsi utama dari *enterprise architecture* bagi organisasi yaitu sebagai pemandu dalam memberikan informasi sekaligus memberikan batasan pada keputusan yang berkaitan dengan penginvestasian pada teknologi informasi. Selain itu *enterprise architecture* juga dapat digunakan untuk meningkatkan efisiensi TI pada saat inovasi bisnis dikembangkan dalam organisasi serta membantu dalam mengembangkan aplikasi sistem informasi yang tepat.

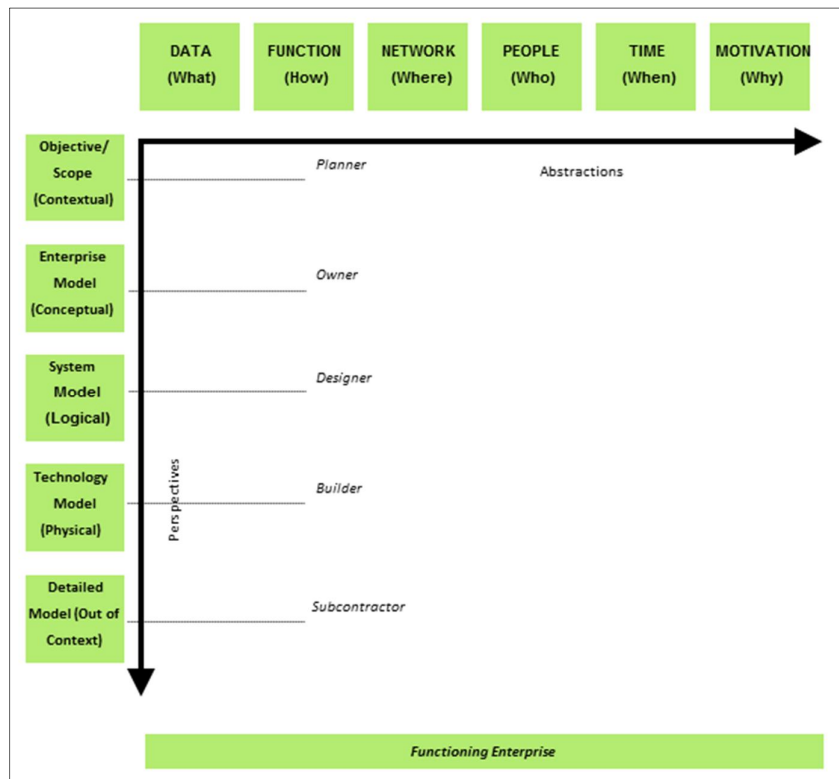
Karakteristik utama pada *enterprise architecture* yaitu menyediakan cara pandang yang menyeluruh mengenai sebuah *enterprise*. Agar dapat mengimplementasikan *enterprise architecture* maka organisasi harus mengadopsi sebuah metode atau framework yang dapat digunakan dalam mengembangkan *enterprise architecture* tersebut. Sehingga dengan adanya metode *enterprise architecture* diharapkan dapat mengelola sistem yang kompleks dan menyelaraskan bisnis serta TI yang akan diinvestasikan[10]. Salah satu metode atau framework *enterprise architecture* yang sering digunakan yaitu Framework Zachman.

2.4 Framework Zachman

Framework Zachman memungkinkan untuk dapat membuat struktur dan dokumentasi mengenai berbagai macam aspek yang berkaitan dengan manajemen dan pembangunan sistem dalam suatu *enterprise*. Framework Zachman yang dicetuskan oleh John Zachman pada tahun 1987 ini dapat dikatakan sebagai kunci untuk dapat memahami *architecture enterprise* dan kedinamisan organisasi.

Framework Zachman merupakan salah satu framework *enterprise architecture* yang menyajikan cara pandang sekaligus memberikan definisi *enterprise* secara formal dan terstruktur dari berbagai perspektif atau sudut pandang.

Pendekatan Framework Zachman secara luas digunakan untuk pengembangan dan atau mendokumentasikan *architecture enterprise* secara praktis, di mana dalam framework ini terdapat dua sumbu utama yaitu sumbu vertikal dan sumbu horizontal. Sumbu vertikal menyajikan berbagai perspektif dari keseluruhan arsitektur sedangkan sumbu horizontal merupakan abstraksi klasifikasi berbagai artifak (model, gambaran, diagram, maupun dokumen) dari suatu arsitektur.

































Gambar 2.2 Deskripsi Konseptual Framework Zachman[9]

2.4.1 Komponen Framework Zachman

Terdapat 6 komponen pada framework Zachman yaitu *data*, *function* (fungsi), *network* (jaringan), *people* (personil), *time* (waktu), dan *motivation* (motivasi). Setiap komponen yang ada digambarkan dan dilihat berdasarkan 6 perspektif yang berbeda yaitu perspektif *planner*, *owner*, *designer*, *builder*, dan *subcontractor*.

Berbagai perspektif ini memberikan syarat dan batasan pada arsitektur sistem informasi yang merupakan representasi lengkap mengenai sistem informasi yang bersama-sama menghasilkan deskripsi lengkap mengenai *enterprise architecture*.

Framework Zachman dapat digunakan sebagian-sebagian sesuai dengan kebutuhan berdasarkan skala prioritas dan tidak harus menggunakan keseluruhan framework tersebut karena akan memakan waktu serta biaya[11].

	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	
	What	How	Where	Who	When	Why	
Objective/Scope: Contextual	List of Things Important in the Business	List of Core Business Processes	List of Business Locations	List of Important Organizations	List of Significant Events	List of Business Code	Objective/Scope: Contextual
Role: Planner							Role: Planner
Enterprise Model Conceptual	Conceptual Data/Object Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan	Enterprise Model Conceptual
Role: Owner							Role: Owner
System Model Logical	Logical Data/Class Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Role Model	System Model Logical
Role: Designer							Role: Designer
Technology Model Physical	Physical Data/Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design	Technology Model Physical
Role: Builder							Role: Builder
Detailed Representations Out of Context	Data Definitions	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification	Detailed Representations Out of Context
Role: Programmer							Role: Programmer
Functioning Enterprise	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy	Functioning Enterprise
Role: User							Role: User

Gambar 2.3 Framework Zachman[12]

Masing-masing perspektif yang digambarkan sebagai keseluruhan baris dari matriks dapat dijelaskan sebagai sebuah model berikut[9]:

1. Konteks Bisnis

Seseorang yang membangun wacana, latar belakang, ruang lingkup, dan tujuan perusahaan. Perspektif ini menjelaskan mengenai arah dan tujuan bisnis dari sebuah perusahaan. Baris ini berhubungan dengan sesuatu yang menjelaskan mengenai misi perusahaan yang nantinya akan menghasilkan konteks kebutuhan perusahaan tersebut. Perspektif ini dibutuhkan agar dapat menetapkan konteks bagi setiap usaha pengembangan suatu sistem.

2. Model Bisnis

Memberikan penjelasan mengenai perusahaan dari sisi perspektif bisnis. Baris ini memberikan gambaran mengenai situasi bisnis yang sebenarnya meliputi struktur, fungsi, dan organisasi yang ada di perusahaan dalam konteks bisnis.

3. Model Sistem

Desainer, perancang atau arsitek yang berperan untuk memberikan penjabaran mengenai keinginan *owner* dalam tataran teknis, menjelaskan mengenai sistem aplikasi dari perspektif kebutuhan, dan menguraikan secara rinci perubahan bentuk data yang terjadi pada masing-masing fungsi bisnis.

4. Model Teknologi

Kontraktor yang membangun produk akhir dan layanan, menjelaskan bagaimana teknologi dapat digunakan untuk memenuhi kebutuhan pengolahan informasi sekaligus dapat menyelesaikan permasalahan bisnis perusahaan.

5. Penggambaran Detail

Sebagai bagian yang bertanggung jawab dalam membangun dan merakit bagian atau komponen untuk menjadi produk akhir atau layanan. Artifak-artifak ini meliputi kode sumber program dan skema basis data yang digunakan oleh pengembang untuk membangun sistem yang diinginkan. Baris ini memperlihatkan *listing* program, spesifikasi basis data, jaringan komputer, dan komponen.

6. Sistem Terpakai

Sistem atau produk akhir yang diimplementasikan oleh organisasi yang terdiri dari program dan *database*. Perspektif ini berisi produk akhir yang diimplementasikan.

Berdasarkan hasil perspektif yang telah didefinisikan tersebut, berikut ini dapat disimpulkan inti dari isi perspektif yang ada pada baris Framework Zachman.

Tabel 2.2 Hasil Perspektif Framework Zachman[9]

Perspektif	Tujuan	Hasil	Batasan
<i>Planner</i>	Mendefinisikan lingkup/ <i>scope</i> .	Definisi dari lingkup / <i>scope</i> .	Keuangan dan pengaturannya.
<i>Owner</i>	Mendeskripsikan bentuk dari produk.	Model bisnis.	Kebijakan dan penggunaannya.
<i>Designer</i>	Mendeskripsikan bentuk logika / abstrak dari produk.	Model sistem.	Lingkungan dan teknologi yang akan digunakan.
<i>Builder</i>	Mendeskripsikan pengembangan produk dan penerapannya.	Model teknologi.	Mengembangkan dan menyiapkan kebutuhan akan teknologi yang diperlukan.
<i>Subcontractor</i>	Mendeskripsikan komponen yang diperlukan.	<i>Out-of-context models</i> .	Penerapan dan integrasi.

Setiap kolom dalam Framework Zachman menggambarkan fokus komponen sistem informasi yang berbeda namun secara bersama-sama saling berinteraksi dalam menguraikan keseluruhan *enterprise architecture*. Berikut merupakan penjelasan lebih lanjut mengenai tiap kolom yang ada di Framework Zachman[9]:

1. Kolom Data (*What*)

Kolom ini berfokus pada hubungan antarentitas dengan menjabarkan relasi data yang satu dengan yang lain agar dapat menggambarkan kebutuhan perusahaan yang harus dijaga.

2. Kolom Fungsi (*How*)

Kolom ini berfokus pada proses dan fungsi yang dihasilkan dengan menjelaskan keseluruhan proses yang terjadi di dalam organisasi, proses kegiatan dalam pemenuhan kebutuhan stakeholder, dan proses *input* serta *output* yang terjadi pada organisasi.

3. Kolom Jaringan (*Where*)

Kolom ini berfokus pada berbagai node dan link yang menjelaskan lokasi operasional dari organisasi, struktur bangunan hingga peta instalasi jaringan yang dimiliki oleh organisasi tersebut.

4. Kolom Orang (*Who*)

Menurut struktur dan tanggung jawab yang ada di organisasi kolom ini berfokus pada peran dan tanggung jawab dalam mendeskripsikan alokasi sumber daya manusia atau orangnya.

5. Kolom Waktu (*When*)

Berfokus pada siklus waktu yang berguna untuk mendeskripsikan waktu terjadinya proses dalam organisasi yang memiliki relasi dalam membangun kriteria kinerja dan tingkat kualitatif sumber daya organisasi.

6. Kolom Motivasi (*Why*)

Berfokus pada visi, misi, dan tujuan organisasi yang menjabarkan motivasi dan tujuan akhir dari organisasi tersebut beserta strategi dan metode pencapaian yang digunakan organisasi.

2.4.2 Kelebihan dan Kelemahan Framework Zachman

Framework Zachman memiliki kelebihan yaitu[13]:

1. Sangat mudah dipahami karena mengacu pada organisasi secara umum dan menggambarkan *tools* serta metode secara independen.
2. Semua komponen yang ada pada framework tersebut dapat dipetakan agar menemukan kondisi yang paling cocok dengan organisasi.

3. Adanya klasifikasi yang memungkinkan untuk pengidentifikasian seluruh bagian yang berbeda dari infrastruktur TI melalui sudut pandang dan logika yang berbeda sehingga akan lebih banyak daftar komponen untuk dianalisa secara lebih teratur.

Sedangkan kelemahan dari Framework Zachman yaitu[13] :

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

2.4.3 Artifak Pada Framework Zachman

Pada dasarnya Framework Zachman untuk *architecture enterprise* adalah sebuah skema yang digunakan untuk mengelola artifak suatu *enterprise*. Pengertian artifak di sini dapat berupa model, gambar, diagram, atau dokumen. Tidak semua artifak yang dimiliki perusahaan berguna untuk mengelola sistem informasinya.

Dengan bantuan Framework Zachman perencanaan sistem informasi suatu perusahaan dapat difokuskan pada artifak yang berguna dan menghindari adanya *artifact of noise* yang sering mengalihkan perhatian dalam perencanaan untuk mengidentifikasi masalah yang sesungguhnya.

Artifak yang benar-benar digunakan untuk perencanaan sistem informasi disebut dengan *artifact of value* yang membantu mengelola perusahaan dan membuat pengoperasian artifak tersebut lebih baik. Framework Zachman membantu dalam menentukan kegunaan dari suatu artifak. *Artifact of Value* membantu dalam membahas suatu topik penting bagi perusahaan. Bila pendefinisian suatu artifak tidak berguna dalam membahas isu penting dalam perusahaan, maka artifak tersebut dapat dianggap tidak berpengaruh besar pada proses perubahan dalam perusahaan.

Di bawah ini merupakan serangkaian kemungkinan artefak atau *tools* yang dapat mewakili isi sel pada Framework Zachman.

Tabel 2.3 Artefak Pada Framework Zachman[14][15]

	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>Time</i>	<i>Why</i>
<i>Scope</i>	Data yang berkaitan dengan sistem.	Proses utama yang terjadi pada sistem.	Lokasi tempat sistem berada.	Orang yang berperan penting pada proses tersebut.	Kegiatan atau kejadian yang berkaitan dengan sistem.	Visi dan Misi secara umum organisasi.
<i>Business/Enterprise Model</i>	<ul style="list-style-type: none"> • <i>Flowmap</i> • <i>Use Case System</i> • <i>Entites Relationship Diagram (ERD)</i> • <i>UML / UMEL Diagram</i> • <i>Entities Dictionary</i> • <i>IDEF1X</i> 	<ul style="list-style-type: none"> • <i>Physical Data Flow</i> • <i>Sys/Sub.sys. vs. Processes Matrix</i> • <i>Processes Dictionary</i> • <i>Activity Diagram</i> • <i>Work Flow</i> • <i>UML Act. Diag.</i> • <i>GRAI Nets</i> 	<ul style="list-style-type: none"> • Desain jaringan yang telah ada pada organisasi. • <i>Functional Logistic System Decomposition</i> • <i>UML Packages</i> 	<ul style="list-style-type: none"> • Orang-orang yang ditugaskan membangun dan mengelola sistem informasi. • <i>Organization Chart</i> • <i>GRAI Grid</i> • <i>Processes vs. Organization Matrix</i> • <i>Use Case</i> 	<ul style="list-style-type: none"> • <i>Time Schedule pembangunan proyek sistem dengan GANTT Chart.</i> • <i>IDEF3 OSTN (Object State Transition Network)</i> 	<ul style="list-style-type: none"> • Alasan pengadaan sistem informasi. • <i>Business Table</i>
<i>System Model</i>	<ul style="list-style-type: none"> • <i>Class Diagram</i> • Logical ERD/ERM/IDE F1X 	<ul style="list-style-type: none"> • <i>Sequence Diagram</i> • <i>Detailed Use Case</i> • <i>Processes vs. Entities Matrix</i> • <i>Sys. Process Dictionary</i> • <i>Use Case</i> 	<ul style="list-style-type: none"> • Desain jaringan usulan. • <i>Systems Diagram</i> • <i>UML Component</i> 	<ul style="list-style-type: none"> • Rancangan manual interface aplikasi. • <i>Roles Matrix</i> • <i>GRAI grid</i> • <i>Use Case</i> 	<ul style="list-style-type: none"> • Detail jadwal perancangan model sistem dengan GANTT Chart. • <i>State Diagram</i> • <i>IDEF3</i> 	<ul style="list-style-type: none"> • Aturan dalam pembuatan model. • <i>FOL (First Order Logic)</i> • <i>Decision Table</i>

	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>Time</i>	<i>Why</i>
Technology Model	<ul style="list-style-type: none"> • IDEF1X (<i>Inf. Model</i>) • Relasi Antar Tabel 	<ul style="list-style-type: none"> • <i>Sequence Diagram.</i> • IDEF0 • <i>Activity Struct. Chart</i> • UML/UEML <i>Class</i> 	<ul style="list-style-type: none"> • Lokasi dan teknologi yang digunakan. • UML <i>Deployment</i> 	<ul style="list-style-type: none"> • Gambar interface aplikasi. • <i>Use Case</i> 	<ul style="list-style-type: none"> • Detail jadwal perancangan sistem informasi dengan GANTT Chart. • UML <i>Sequence & collaboration Diag.</i> 	<ul style="list-style-type: none"> • Aturan dalam pembuatan sistem. • FOL (<i>First Order Logic</i>) • <i>Decision Table</i>
Detailed Description	<ul style="list-style-type: none"> • Algoritme pembuatan tabel entitas. • DB <i>Schema</i> • SQL 	<ul style="list-style-type: none"> • <i>Algoritma proses pelaporan.</i> • <i>Programming Languages</i> 	<ul style="list-style-type: none"> • Konfigurasi setingan router jaringan. • URL • IP • TCP/IP 	Algoritma konfigurasi akses aplikasi tiap user.	Detail jadwal proses coding yang dibutuhkan dengan GANTT Chart.	Aturan-aturan dalam proses coding.
Functioning Enterprise	Contoh dari data yang berkaitan dengan sistem diisikan pada masing-masing tabel.	Contoh hasil printout aplikasi yang sudah dibuat dalam proses sistem.	Infrastruktur jaringan yang dibutuhkan.	User yang akan menggunakan sistem informasi.	Time schedule proses perancangan, desain, dan implementasi.	SOP penggunaan sistem informasi.

2.5 Unified Modeling Language (UML)



Unified Modeling Language (UML) merupakan sebuah bahasa pemodelan visual yang digunakan membangun dan mendokumentasi berbagai artefak dari sebuah sistem perangkat lunak. UML membantu dalam menangkap keputusan dan memberikan pemahaman mengenai sistem yang harus dibangun.

UML memiliki beberapa diagram, yaitu:


2.5.1 Use Case Diagram

Use Case merupakan deskripsi dari serangkaian urutan aksi, termasuk kegiatan yang dilakukan oleh sistem agar dapat memberikan hasil berupa nilai yang dapat diobersvasi oleh para aktor[18]. *Use case* digunakan untuk memberikan gambaran mengenai sistem dari sudut pandang *user* sehingga dapat menghadirkan interaksi antara aktor dan sistem.

Tabel 2.4 Komponen Use Case Diagram[18]

Notasi	Keterangan	Simbol
<i>Actor</i>	Menggambarkan seseorang, perangkat, atau sistem lain yang berinteraksi dengan sistem yang dibuat.	
<i>Use Case</i>	Menggambarkan kegiatan apa yang dikerjakan oleh sistem, bukan bagaimana sistem mengerjakannya. Penamaan pada <i>use case</i> menunjukkan hal yang dicapai dari interaksinya dengan aktor.	

Tabel 2.5 Jenis Hubungan Pada Use Case Diagram[18]

Relasi	Fungsi	Notasi
Asosiasi	Jalur komunikasi antara aktor dan <i>use case</i> yang berpartisipasi di dalamnya.	


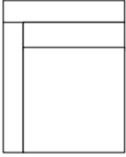
<i>Extend</i>	Menunjukkan jika <i>use case</i> hanya dieksekusi pada keadaan tertentu.	«extend» ----->
<i>Include</i>	Menunjukkan jika <i>use case</i> harus terpenuhi agar sebuah <i>event</i> dapat terjadi dan menggambarkan bahwa suatu <i>use case</i> termasuk di dalam <i>use case</i> lain.	«include» ----->
<i>Generalization</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> yang lebih spesifik di mana <i>use case</i> mewarisi <i>behaviour</i> dari <i>parent use case</i> .	→

2.5.2 Activity Diagram

Activity Diagram adalah grafik node dan arus yang menunjukkan objek yang melakukan dan menerima aktivitas serta menampilkan aliran aktivitas di dalam sistem secara *sequential* maupun bercabang[18]. Pada dasarnya *activity diagram* memiliki peran hampir seperti dengan *flowchart* namun *activity diagram* dapat mendukung perilaku paralel yang tidak dapat dilakukan oleh *flowchart*.

Tabel 2.6 Komponen *Activity Diagram*[17]

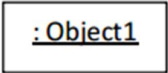



Notasi	Keterangan	Simbol
<i>Initial</i> (Status Awal)	Titik awal yang digunakan untuk memulai sebuah aktivitas.	●
<i>Final</i> (Status Akhir)	Titik akhir yang digunakan untuk mengakhiri sebuah aktivitas.	○
<i>Activity</i>	Aktivitas yang dikerjakan oleh sistem.	□

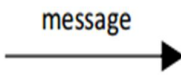

<i>Fork (Join)</i>	Menunjukkan kegiatan yang dilakukan secara paralel untuk menggabungkan dua kegiatan paralel menjadi satu.	
<i>Swimlane</i>	Memisahkan organisasi yang bertanggung jawab terhadap aktivitas yang terjadi pada sistem.	

2.5.3 Sequence Diagram

Sequence Diagram menjabarkan *behavior* sebuah skenario tunggal. Diagram ini menunjukkan sejumlah objek dan pesan yang melwati objek-objek ini di dalam *use case*. *Sequence Diagram* menunjukkan interaksi dengan menampilkan setiap partisipan dengan garis alir secara vertikal dan pengurutan pesan dari atas ke bawah[19].

Tabel 2.7 Komponen *Sequence Diagram*[17]

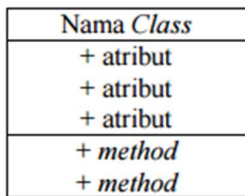


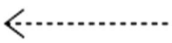
Notasi	Keterangan	Simbol
<i>Object</i> (Partisipan)	<i>Instance</i> dari sebuah class dan dituliskan tersusun secara horizontal	
<i>Actor</i>	Menggambarkan seseorang, perangkat, atau sistem lain yang berinteraksi dengan sistem yang dibuat.	
<i>Lifeline</i>	Keberadaan sebuah object dalam basis waktu.	
<i>Actiation</i>	Objek yang akan melakukan sebuah aksi.	


<i>Message</i>	Komunikasi antarobjek-objek	
<i>Self-message</i>	Komunikasi kembali ke dalam sebuah objek itu sendiri.	

2.5.4 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas yang akan dibuat untuk membangun sistem sehingga antara dokumen perancangan dan perangkat lunak yang dibuat sinkron[17]. *Class Diagram* menunjukkan hubungan antarkelas yang di dalamnya terdapat atribut dan fungsi dari suatu objek.

Tabel 2.8 Komponen Class Diagram[17]

Notasi	Keterangan	Simbol
<i>Class</i>	Blok pembangunan pada pemrograman berorientasi objek, Bagian atas adalah bagian nama dari <i>class</i> , bagian tengah mendefinisikan atribut <i>class</i> , dan bagian akhir mendefinisikan berbagai <i>method</i> dari <i>class</i> .	
<i>Assosiation</i>	Relasi antarkelas yang memiliki makna kelas yang satu digunakan oleh kelas yang lain. Biasanya relasi ini disertai dengan <i>multiplicity</i> .	
<i>Composition</i>	Relasi pada <i>class</i> yang tidak dapat berdiri sendiri dan merupakan bagian dari <i>class</i> lain terhadap <i>class</i> tempat bergantung.	
<i>Dependency</i>	Kebergantungan antarkelas.	

<i>Aggregation</i>	Relasi yang mengindikasikan keseluruhan bagian dari hubungan dan biasanya disebut sebagai relasi yang mempunyai sebuah atau bagian dari.	
--------------------	--	---

Tabel 2.9 Simbol *Relationship Multiplicity*[19]

Indikator	Arti
0...1	Kosong atau satu.
0! *	Nol atau lebih.
1	Hanya satu.
1...*	Satu atau lebih
*	Banyak atau <i>Many</i> .

2.6 *Gantt Chart*


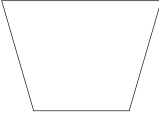



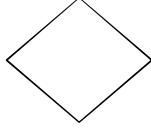
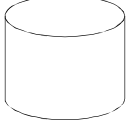

Gantt Chart menyediakan suatu format *standard* untuk menggambarkan informasi mengenai jadwal proyek dengan menampilkan kegiatan proyek, jadwal mulai dan jadwal selesai dalam format kalender[20]. *Gantt Chart* membantu penggunaannya untuk memastikan bahwa semua kegiatan telah direncanakan, urutan kinerja telah diperhitungkan, perkiraan waktu kegiatan telah tercatat, dan keseluruhan waktu proyek telah dibuat[21].

Gantt Chart memiliki kelebihan yaitu dapat menunjukan waktu kegiatan dan urutan kegiatan namun memiliki kelemahan yaitu tidak memperlihatkan hubungan dan ketergantungan antar kegiatan sehingga sulit diantisipasi jika terjadi keterlambatan suatu kegiatan terhadap jadwal keseluruhan proyek.

2.7 *Flowmap*

Flowmap merupakan campuran dari peta dan *flow chart* yang menunjukkan pergerakan beda dari suatu lokasi ke lokasi lain. *Flomap* menolong analisis dan programmer untuk memecahkan masalah ke dalam segmen yang lebih kecil dan menolong dalam mengenalaisis alternatif lain dalam pengoperasian.

Tabel 2.10 Simbol *Flowmap*[22]

Notasi	Keterangan	Simbol
Dokumen	Menunjukkan dokumen <i>input</i> / <i>output</i> baik untuk proses manual, mekanik, atau komputer	
<i>Manual Operation</i>	Menunjukkan pekerjaan yang ditangani dengan cara manual tanpa menggunakan proses komputerisasi.	
Proses	Menunjukkan kegiatan proses <i>input</i> atau <i>output</i> data yang diproses dari operasi program komputer.	
<i>Keyboard</i>	Menunjukkan kegiatan <i>input</i> yang menggunakan <i>on-line keyboard</i> .	
Garis Alir	Menunjukkan arus dari proses.	
Keputusan (<i>Decision</i>)	Keputusan yang dilakukan secara komputer atau manual.	
<i>Database</i>	Penyimpanan data secara komputer.	
Arsip	Simpanan data non komputer atau informasi file pada proses manual.	

2.8 *Blueprint*

Blueprint adalah sebuah rancangan yang dirumuskan untuk memberikan arahan terhadap kegiatan perusahaan yang dilakukan secara berkesinambungan sehingga setiap kegiatan memiliki kesesuaian dengan tuntutan, tantangan, dan kebutuhan di lingkungan sekitar perusahaan[23]. Fungsi *blueprint* adalah sebagai strategi perencanaan dan pengembangan dalam sistem informasi di suatu perusahaan yang merupakan bagian terintegrasi dengan perencanaan korporat (*corporate business plan*) yang menyatakan bahwa keberadaan sistem informasi merupakan bagian dari strategi perusahaan dalam usaha pencapaian visi dan misi yang dimiliki oleh perusahaan.