

BAB 2

LANDASAN TEORI

2.1. Pengertian Sistem

Suatu sistem adalah sekelompok komponen yang saling berhubungan yang berfungsi bersama untuk mencapai hasil yang diinginkan. Sebagai contoh, terdapat sebuah *home theater* yang terdiri dari pemutar DVD, *receiver*, *speaker*, dan layar monitor. (Whitten & Bentley, 2007:6)

Dari pengertian diatas dapat diambil kesimpulan bahwa suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya antara satu dengan yang lainnya, yang berfungsi untuk mencapai sebuah tujuan tertentu.

2.2. Pengertian Informasi

Informasi merupakan sekumpulan data yang telah diolah dan disajikan sehingga memiliki nilai dan arti bagi si penerima (Cegielski & Rainer, 2011:10). Informasi merupakan sekumpulan data yang telah diubah dalam bentuk yang bermakna dan berguna bagi manusia (Laudon & Laudon, 2014:89)

Dari pengertian diatas dapat disimpulkan bahwa informasi merupakan sekumpulan data yang berisi fakta yang telah diolah dan dibentuk dalam suatu bentuk yang berguna, bermakna dan dapat dimengerti oleh manusia.

2.3. Pengertian Sistem Informasi

Sistem informasi merupakan sekumpulan komponen komputer yang saling terkait yang mengumpulkan, memproses, menyimpan dan menyediakan *output* informasi yang diperlukan untuk menyelesaikan tugas bisnis (Satzinger, Jackson & Burd, 2012:4).

Sistem informasi dalam organisasi menangkap dan mengelola data untuk menghasilkan informasi yang berguna yang mendukung organisasi, karyawan, pelanggan, pemasok dan mitra. (Whitten & Bentley, 2007:6)

2.4. Pengertian Administrasi

Administrasi didefinisikan sebagai suatu usaha dan kegiatan yang meliputi penetapan tujuan serta penetapan cara-cara yang berkaitan dengan penyelenggaraan pembinaan organisasi (Safroedin Bahar:2008).

Administrasi adalah proses penyelenggaraan kegiatan organisasi untuk mencapai tujuan yang telah ditetapkan dengan menggunakan sumber daya manusia.

Dari teori yang telah dikemukakan di atas dapat dikatakan bahwa administrasi adalah suatu kegiatan atau usaha untuk mengarahkan semua kegiatan dalam mencapai suatu tujuan dengan memanfaatkan sumber daya manusia serta sarana dan prasarana.

2.5. Pengertian Sistem Informasi Administrasi

Dari teori-teori yang telah diuraikan di atas tentang sistem, informasi, sistem informasi dan administrasi, maka dapat dikatakan bahwa sistem informasi administrasi adalah kumpulan dari komponen-komponen atau elemen-elemen yang saling berhubungan untuk melakukan proses pencatatan, pengaturan, pengalokasian suatu kegiatan untuk mencapai tujuan tertentu dengan menggunakan sarana dan perlengkapan yang tersedia.

2.6. Pengertian Sistem Informasi Manajemen Laboratorium

Menurut McLeod (2001), sistem informasi manajemen adalah sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Para pemakai biasanya membentuk suatu entitas organisasi formal perusahaan atau sub unit dibawahnya. Informasi menjelaskan perusahaan atau salah satu sistem utamanya mengenai apa yang terjadi dimasa lalu, apa yang sedang terjadi sekarang dan mungkin apa yang terjadi dimasa depan. Informasi tersebut tersedia dalam bentuk laporan periodik, laporan khusus, dan *output* dari simulasi matematika. *Output* informasi digunakan oleh manajer maupun nonmanajer dalam perusahaan saat mereka membuat keputusan untuk memecahkan masalah.

Dari pengertian diatas dapat disimpulkan bahwa sistem informasi manajemen laboratorium merupakan sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa diantaranya dekan, ketua program studi, kepala laboratorium, laboran, asisten dosen dan mahasiswa.

2.7. *Interface*

Interface adalah tampilan aplikasi untuk pengguna yang harus terstruktur dengan baik dan mudah digunakan. Ketika desain aplikasi telah dibentuk dan komponen-komponennya telah dikenali, desainer interaksi mempersiapkan desain antarmuka pengguna dan alur navigasi yang menunjukkan bagaimana pengguna berpindah dari satu layar ke layar lain untuk mengakses layanan.

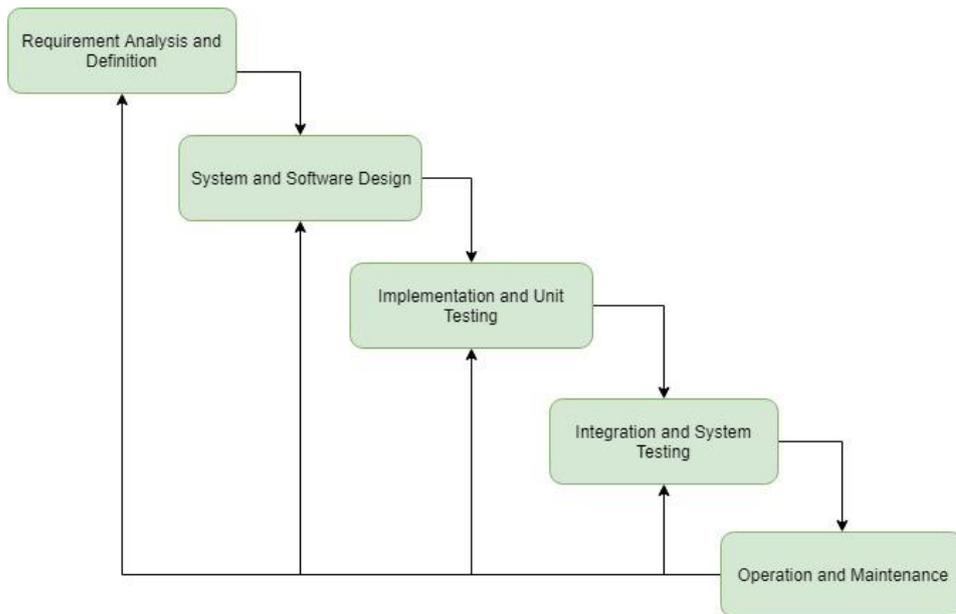
Berikut ini adalah beberapa panduan sederhana yang dapat membantu di dalam merancang antarmuka pengguna yang lebih efektif.

- a. Menjaga antarmuka aplikasi agar cukup sederhana sehingga jarang menggunakan petunjuk pengguna untuk melakukan suatu tugas.
- b. Mengurangi jumlah informasi yang ditampilkan.
- c. Membuat masukan yang ringkas sehingga pengguna dapat memenuhi tugas dengan jumlah tombol klik minimum.
- d. Antarmuka tidak tergantung pada ukuran layar manapun.

Keluaran dari tahap ini adalah petunjuk pengguna yang menggambarkan tampilan aplikasi dan alur navigasi. (Janner Simarmata, 2010:375)

2.8. *Metode Waterfall*

Model pertama yang diterbitkan dari proses pengembangan perangkat lunak berasal dari proses rekayasa sistem yang lebih umum (Royce, 1970). Model ini diilustrasikan pada Gambar 2.1. Karena prosesnya berjalan dari satu fase ke fase lain, model ini dikenal sebagai *waterfall model* atau *software development life cycle*. *Waterfall model* adalah contoh dari proses yang digerakkan oleh rencana pada prinsipnya, harus merencanakan dan menjadwalkan semua proses kegiatan sebelum mulai bekerja. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirements analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance*. (Sommerville, 2011:30)



Gambar 2.1 Waterfall Model

(Sumber : Sommerville (2011:30))

Berikut ini penjelasan tahapan-tahapan dari *waterfall model* :

1. Requirement Analysis and Definition

Tahapan melakukan identifikasi masalah, analisa kebutuhan dan tujuan sistem melalui konsultasi dengan pengguna sistem.

2. System and Software Design

Tahapan melakukan desain sistem berdasarkan kebutuhan yang telah ditetapkan pada tahapan sebelumnya.

3. Implementation and Unit Testing

Tahapan hasil dari desain perangkat lunak untuk direalisasikan sebagai satu set program atau unit program.

4. Integration and System Testing

Tahapan mengintegrasikan setiap unit program satu sama lain dan diuji sebagai satu system yang utuh untuk memastikan system sudah memenuhi persyaratan yang sudah ditetapkan.

5. Operation and Maintenance

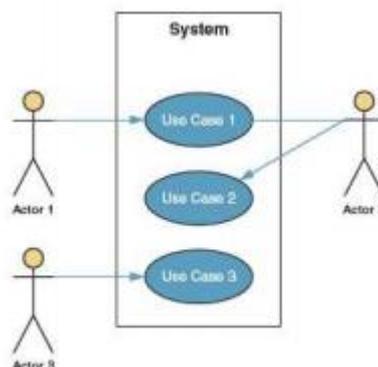
Tahapan instalasi dan penerapan system. Pada tahap ini juga dilakukan pengujian pada saat system dijalankan untuk menemukan dan memperbaiki *error* yang tidak ditemukan pada tahap pembuatan.

2.9. Unified Modelling Language (UML)

Unified Modeling Language (UML) suatu kumpulan rancangan dari model yang digunakan untuk menggambarkan sistem dari sisi perangkat lunak dan dari sisi segi objek. Perkembangan UML berawal dari pengembangan sistem berorientasi objek yang dikembangkan sebelumnya oleh James Rumbaugh dan Grady Booch dimana bertujuan untuk menciptakan standar tunggal dalam proses pengembangan sistem berorientasi objek di tahun 1994. Kemudian melalui kerja sama antara James Rumbaugh, Grady Booch dan Ivar Jacobson mengembangkan standar baru didalam bahasa pemodelan yang berorientasi pada objek. Dan pada tahun 1997, UML versi 1.0 resmi diluncurkan. Melalui pengembangan yang lebih lanjut, saat ini bahasa pemodelan berorientasi objek yang digunakan menjadi UML versi 2.0. (Whitten & Bentley, 2007:371)

2.9.1. Use Case Diagram

Use case menjelaskan fungsi sistem dari perspektif pengguna yang dapat dimengerti. *Use case* mewakili satu tujuan dari sistem dan menjelaskan urutan kegiatan dan interaksi pengguna untuk mencapai tujuan pembuatan sistem. (Whitten & Bentley, 2007:246)



Gambar 2.2 Use Case

(Sumber : Whitten & Bentley (2007(246))

Sebuah *use case* diagram digambarkan dengan komponen sebagai berikut:

1. Use Case

Use case digambarkan dalam bentuk *elips* yang merupakan aktivitas sistem yang dilakukan oleh *actor*.



Gambar 2.3 Use Case

(Sumber : Whitten & Bentley (2007(246))

2. *Actor*

Actor merupakan pelaku yang melakukan interaksi dengan sistem

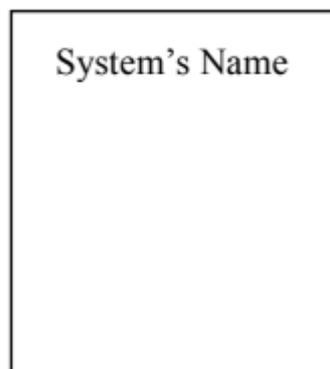


Gambar 2.4 Actor

(Sumber : Whitten & Bentley (2007(247))

3. *Boundary*

Boundary merupakan garis batas antara *use case* dengan *actor*. Biasanya *boundary* memiliki nama untuk mengklasifikasikan proses *use case diagram*.



Gambar 2.5 Boundary

(Sumber : Whitten & Bentley (2007(246))

4. *Relationship*

Relationship digambarkan dalam bentuk tanda panah yang menghubungkan antara *actor* dengan *use case*. Terdapat beberapa bentuk *relationship* yaitu :

a. *Association*

Relationship antara *actor* dengan sebuah *use case*.

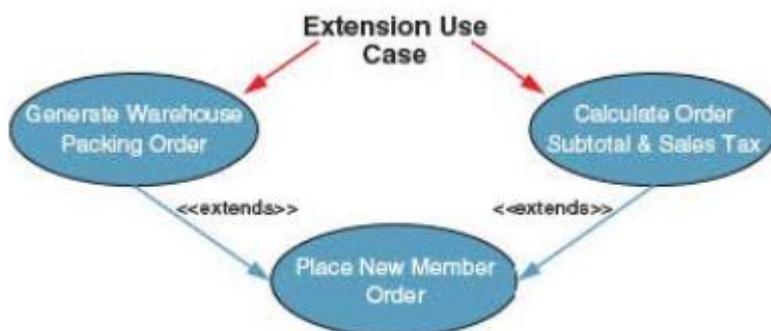


Gambar 2.6 Association Relationship

(Sumber : Whitten & Bentley (2007(248))

b. *Extends Use Case*

Sebuah *use case* yang mencakup langkah-langkah ekstraksi dari *use case* yang lebih kompleks dengan tujuan untuk menyederhanakan *use case* aslinya.

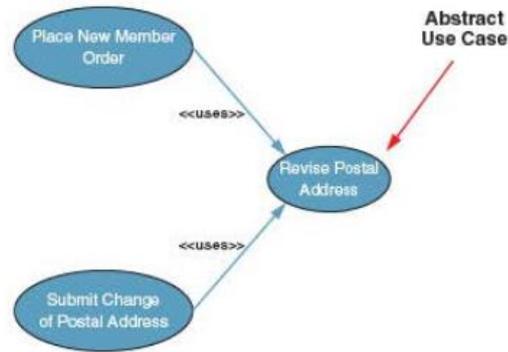


Gambar 2.7 Extends Use Case

(Sumber : Whitten & Bentley (2007(249))

c. *Uses* atau *Include*

Uses atau *include* berfungsi untuk menjelaskan hubungan ketergantungan yang terjadi pada *use case*.



Gambar 2.8 Uses atau Include

(Sumber : Whitten & Bentley (2007(249))

2.9.2. Use Case Narrative

Menurut Whitten & Bentley (2007:256) penjelasan secara narasi akan membantu mempercepat pemahaman terhadap sistem karena keterangannya yang bersifat *high-level* (gambaran tingkat tinggi dan secara luas).

Member Services System	
Author (s): _____ 1	Date: _____ 2 Version: _____ 3
Use-Case Name: Place New Order 4	Use-Case Type Business Requirements: 5
Use-Case ID: MSS-BUC002.00 6	
Priority: High 7	
Source: Requirement — MSS-R1.00 8	
Primary Business Actor: Club member 9	
Other Participating Actors:	<ul style="list-style-type: none"> Warehouse (external receiver) Accounts Receivable (external server) 10
Other Interested Stakeholders:	<ul style="list-style-type: none"> Marketing — Interested in sales activity in order to plan new promotions. Procurement — Interested in sales activity in order to replenish inventory. Management — Interested in order activity in order to evaluate company performance and customer (member) satisfaction.
Description: 12	This use case describes the event of a club member submitting a new order for SoundStage products. The member's demographic information as well as his or her account standing is validated. Once the products are verified as being in stock, a packing order is sent to the warehouse for it to prepare the shipment. For any product not in stock, a back order is created. On completion, the member will be sent an order confirmation.

Gambar 2.9. High-Level Use Case Narrative

(Sumber : Whitten & Bentley (2007(257))

Penjelasan dari gambar *use case narrative* diatas adalah sebagai berikut:

1. *Author*: diisi dengan nama seseorang yang menulis *use case* dan bersedia untuk dihubungi untuk semua yang informasi dari *use case* yang dibuat.

2. *Date*: tanggal terakhir *use case* tersebut dimodifikasi.
3. *Version*: versi *use case* yang digunakan.
4. *Use-case name*: nama dari sebuah *use case*, harus menggambarkan tujuan yang ingin dicapai dari *use case* tersebut. Sebaiknya nama *use case* menggunakan kata kerja.
5. *Use-case type*: merupakan penggambaran keinginan dari perilaku sistem.
6. *Use-case ID*: identitas yang unik dari *use case*.
7. *Priority*: tingkat prioritas dari *use case* terdapat high, medium atau low.
8. *Source*: mendefinisikan entitas yang memicu pembuatan *use case*.
9. *Primary business actor*: *stakeholder* yang paling diuntungkan dari terlaksananya *use case* dengan menerima sesuatu yang dapat diukur.
10. *Other participating actor*: *actor* lain yang terlibat dalam *use case* tersebut, *actor* inisitor, *actor* fasilitator, *actor* sampingan ataupun *actor* penerima.
11. *Interested stakeholder*: pemangku kepentingan yang bertanggung jawab dalam proses pengembangan dan pengolahan sistem *software*.
12. *Description*: ringkasan singkat yang terdiri dari beberapa kalimat yang menggambarkan tujuan dari *use case* beserta aktivitasnya.

Selain *use case narrative* diatas, terdapat *high-level use case narrative* bisa dikembangkan menjadi lebih luas dan memiliki pembahasannya yang lebih detail pada setiap tahapnya. Terdapat beberapa tambahan, antara lain:

Member Services System					
Author (s): _____	Date: _____				
	Version: _____				
Use-Case Name:	Place New Order				
Use-Case ID:	MSS-BUC002.00				
Priority:	High				
Source:	Requirement — MSS-R1.00				
Primary Business Actor:	Club member				
Other Participating Actors:	<ul style="list-style-type: none"> Warehouse (external receiver) Accounts Receivable (external server) 				
Other Interested Stakeholders:	<ul style="list-style-type: none"> Marketing — Interested in sales activity in order to plan new promotions. Procurement — Interested in sales activity in order to replenish inventory. Management — Interested in order activity in order to evaluate company performance and customer (member) satisfaction. 				
Description:	This use case describes the event of a club member submitting a new order for SoundStage products. The member's demographic information as well as his or her account standing is validated. Once the products are verified as being in stock, a packing order is sent to the warehouse for it to prepare the shipment. For any product not in stock, a back order is created. On completion, the member will be sent an order confirmation.				
Precondition:	1 The party (individual or company) submitting the order must be a member.				
Trigger:	2 This use case is initiated when a new order is submitted.				
Typical Course of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Actor Action</th> <th style="width: 50%; text-align: center;">System Response</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">3 Step 1: The club member provides his or her demographic information as well as order and payment information.</td> <td style="vertical-align: top;"> Step 2: The system responds by verifying that all required information has been provided. Step 3: The system verifies the club member's demographic information against what has been previously recorded. Step 4: For each product ordered, the system validates the product identity. Step 5: For each product ordered, the system verifies the product availability. Step 6: For each available product, the system determines the price to be charged to the club member. Step 7: Once all ordered products are processed, the system determines the total cost of the order. Step 8: The system checks the status of the club member's account. Step 9: The system validates the club member's payment if provided. Step 10: The system records the order information and then releases the order to the appropriate distribution center (warehouse) to be filled. Step 11: Once the order is processed, the system generates an order confirmation and sends it to the club member. </td> </tr> </tbody> </table>	Actor Action	System Response	3 Step 1: The club member provides his or her demographic information as well as order and payment information.	Step 2: The system responds by verifying that all required information has been provided. Step 3: The system verifies the club member's demographic information against what has been previously recorded. Step 4: For each product ordered, the system validates the product identity. Step 5: For each product ordered, the system verifies the product availability. Step 6: For each available product, the system determines the price to be charged to the club member. Step 7: Once all ordered products are processed, the system determines the total cost of the order. Step 8: The system checks the status of the club member's account. Step 9: The system validates the club member's payment if provided. Step 10: The system records the order information and then releases the order to the appropriate distribution center (warehouse) to be filled. Step 11: Once the order is processed, the system generates an order confirmation and sends it to the club member.
Actor Action	System Response				
3 Step 1: The club member provides his or her demographic information as well as order and payment information.	Step 2: The system responds by verifying that all required information has been provided. Step 3: The system verifies the club member's demographic information against what has been previously recorded. Step 4: For each product ordered, the system validates the product identity. Step 5: For each product ordered, the system verifies the product availability. Step 6: For each available product, the system determines the price to be charged to the club member. Step 7: Once all ordered products are processed, the system determines the total cost of the order. Step 8: The system checks the status of the club member's account. Step 9: The system validates the club member's payment if provided. Step 10: The system records the order information and then releases the order to the appropriate distribution center (warehouse) to be filled. Step 11: Once the order is processed, the system generates an order confirmation and sends it to the club member.				
Alternate Courses:	4 Alt-Step 2: The club member has not provided all the information necessary to process the order. The club member is notified of the discrepancy and prompted to resubmit. Alt-Step 3: If the club member information provided is different from what was previously recorded, verify what was recorded is current, then update the club member information accordingly. Alt-Step 4: If the product information the club member provided does not match any of SoundStage's products, notify the club member of the discrepancy and request clarification. Alt-Step 5: If the quantity ordered of the product is not available, a back order is created. Alt-Step 8: If the status of the club member's account is not in good standing, record the order information and place it in hold status. Notify the club member of the account status and the reason the order is being held. Terminate use case. Alt-Step 9: If the payment the club member provided (credit card) cannot be validated, notify the club member and request an alternative means of payment. If the club member cannot provide an alternate means, cancel the order and terminate the use case.				
Conclusion:	5 This use case concludes when the club member receives a confirmation of the order.				
Postcondition:	6 The order has been recorded and if the ordered products were available, they were released. For any product not available a back order has been created.				
Business Rules:	7 <ul style="list-style-type: none"> The club member responding to a promotion or a member using credits may affect the price of each ordered item. Cash or checks will not be accepted with the orders. If provided, they will be returned to the club member. The club member is billed for products only when they are shipped. 				
Implementation Constraints and Specifications:	8 <ul style="list-style-type: none"> GUI to be provided for Member Services associate, and Web screen to be provided for club member. 				
Assumptions:	9 Procurement will be notified of back orders by a daily report (separate use case).				
Open Issues:	10 <ol style="list-style-type: none"> Need to determine how distribution centers are assigned. 				

Gambar 2.10 Expanded High-Level Use Case Narrative

(Sumber : Whitten & Bentley, (2007(259))

Berikut penjelasan gambar di atas :

1. *Precondition*: merupakan keadaan sistem sebelum pengekseskuan *use case*. Biasanya mengacu pada *use case* lain yang harus dieksekusi terlebih dahulu.
2. *Trigger*: merupakan kejadian yang menjadi pemicu untuk pengekseskuan *use case*.
3. *Typical course of event*: merupakan urutan aktivitas antara *actor* dan sistem untuk memenuhi tujuan *use case*.
4. *Alternate course*: merupakan alternatif perilaku lain dari *use case* bila terdapat pengecualian atau pemilihan dari kejadian yang ada.
5. *Conclusion*: menspesifikasikan ketika *use case* selesai dijalankan. Dengan kata lain ketika *primary business actor* menerima sesuatu yang dapat diukur.
6. *Postcondition*: merupakan keadaan dari sistem setelah pengekseskuan *use case* selesai.
7. *Business rules*: spesifikasi kebijakan dan prosedur dari proses bisnis yang harus ada di sistem yang baru.
8. *Implementation constraints and spesification*: menspesifikasikan non *functional requirement* yang mempengaruhi realisasi dari *use case* tersebut, dan mungkin membantu dalam perencanaan arsitektural dan pembatasan ruang lingkup.
9. *Assumptions*: semua asumsi yang dibuat oleh penyusun ketika mendokumentasikan *use case*.
10. *Open issues*: semua pertanyaan atau permasalahan yang butuh penyelesaian atau investigasi sebelum *use case* dapat diakhiri.

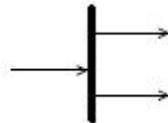
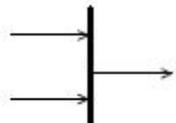
2.9.3. Activity Diagram

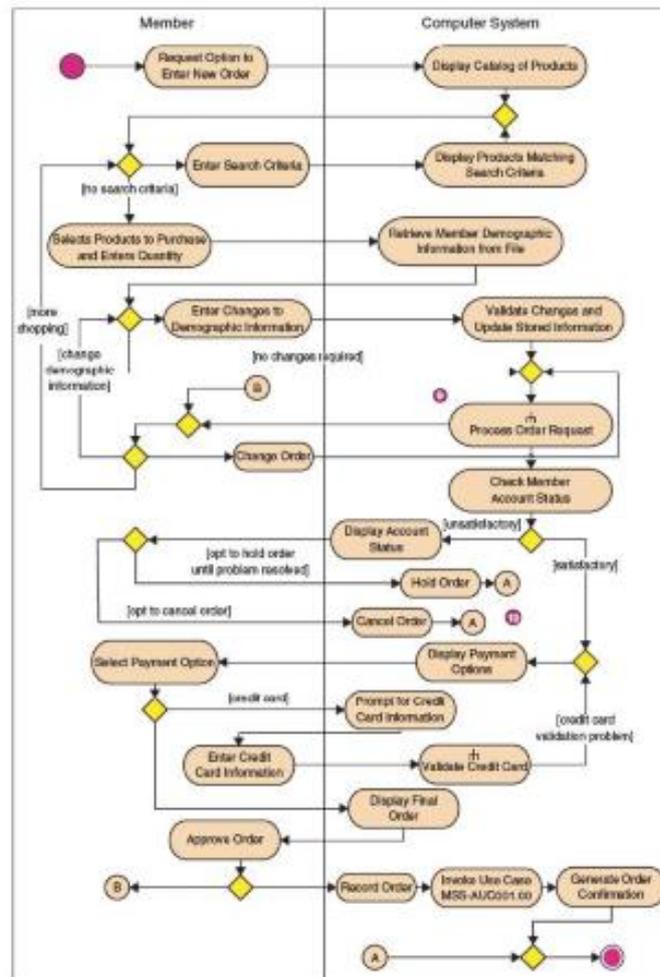
Activity Diagram digunakan untuk membuat model dari tahapan proses atau aktivitas dari sebuah sistem. *Activity diagram* berbeda dengan *flowchart* karena dalam *activity diagram* menyediakan sebuah mekanisme untuk menggambarkan aktivitas yang terjadi secara paralel. *Activity diagram* dapat dibuat untuk setiap *use case*. (Whitten & Bentley, 2007:391).

Sebuah *activity diagram* digambarkan dengan komponen sebagai berikut:

Tabel 2.1 Simbol – simbol pada *Activity Diagram*

(Sumber : Whitten & Bentley (2007(392))

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
2		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
3		<i>Activity Final Node</i>	Bagaimana objek dan dihancurkan
4		<i>Decision</i>	Pilihan untuk mengambil keputusan
5		<i>Fork Percabangan</i>	menunjukkan adanya percabangan secara paralel dari aktivitas.
6		<i>Fork Gabungan</i>	menunjukkan adanya percabangan secara paralel dari aktivitas.
7	Swimlanes		menggambarkan obyek mana yang bertanggung jawab terhadap suatu aktivitas tertentu.



Gambar 2.11 Activity Diagram

(Sumber : Whitten & Bentley (2007(393))

2.9.4. Class Diagram

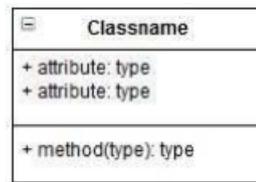
Class Diagram menggambarkan struktur sistem obyek yang menunjukkan kelas-kelas yang telah disusun oleh sistem. (Whitten & Bentley, 2007:400)

Class diagram menghubungkan antara entitas orang, benda, dan data untuk menciptakan hubungan dalam sistem.

Sebuah *class diagram* digambarkan dengan komponen sebagai berikut :

1. Class

Class digambarkan sebagai kotak yang berisi tiga bagian yaitu *class*, *attribute*, dan *method*



Gambar 2.12 Class Diagram

(Sumber : Whitten & Bentley (2007(650))

2. Relationship

Gambaran relasi dari sebuah *class diagram* yang dideskripsikan melalui garis yang menghubungkan satu kelas dengan kelas lainnya. Beberapa relasi dalam *class diagram*, yaitu :

a. Associations

Associations merupakan hubungan dimana suatu *method* didalam sebuah kelas yang memanggil *method* dari objek kelas lain. *Association* digambarkan melalui garis lurus tanda panah yang mengarah dari satu kelas ke kelas lainnya.



Gambar 2.13 Associations

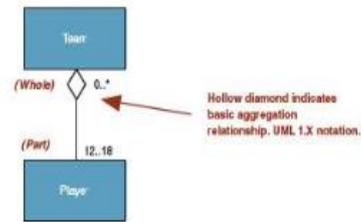
(Sumber : Whitten & Bentley (2007(380))

b. Aggregation

Aggregation merupakan hubungan suatu objek pada kelas berbeda yang merupakan bagian dari kelas lain, namun objek dari kelas tersebut dapat berdiri sendiri. Hubungan antar kelas digambarkan

melalui garis lurus pada sebuah kelas yang mengarah pada kelas lain dengan bentuk segi belah ketupat pada kelas yang dituju.

(a)

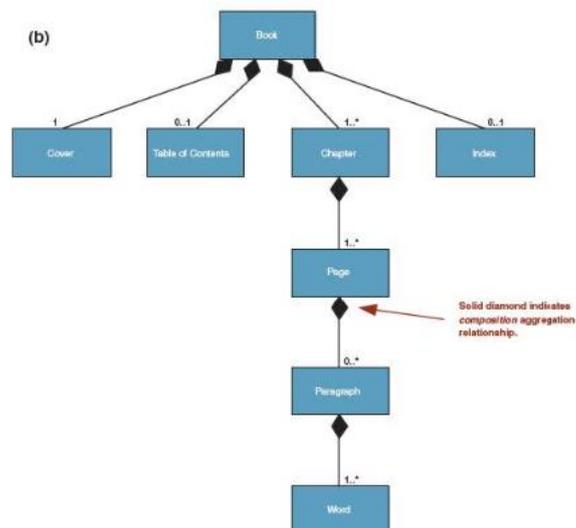


Gambar 2.14 Aggregation

(Sumber : Whitten & Bentley (2007(379)))

c. *Composition*

Composition memiliki hubungan yang hampir sama dengan *aggregations*, namun hubungan *composition* menyatakan bahwa bagian dari objek suatu kelas tidak dapat dipisahkan dari kelas yang memiliki kaitan dengan kelas yang dituju. Hubungan antar kelas digambarkan melalui garis lurus pada sebuah kelas yang mengarah pada kelas lain dengan bentuk segi belah ketupat penuh.



Gambar 2.15 Composition

(Sumber : Whitten & Bentley (2007(379)))

3. Multiplicity

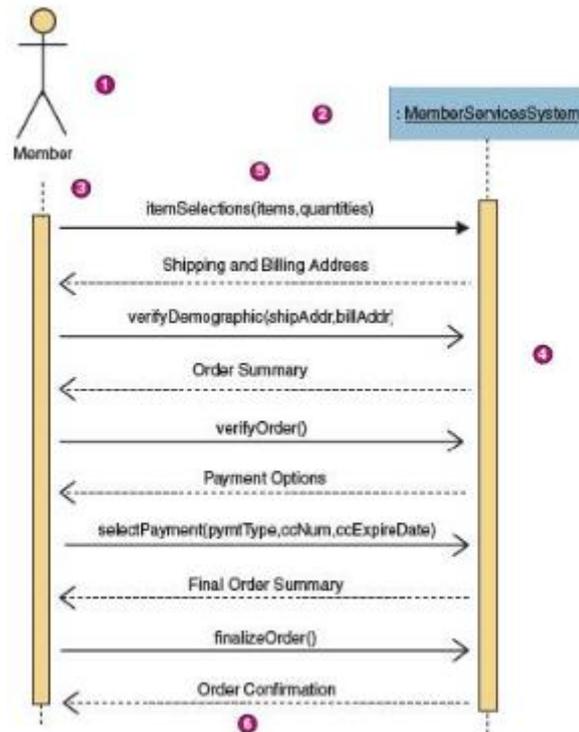
Multiplicity menunjukkan berapa banyak objek dari suatu kelas yang saling terelasi dengan kelas lainnya. *Multiplicity* akan memberikan keterangan pada salah satu jenis hubungan class diagram yang disebut *association*.

Tabel 2.2 *Multiplicity*
(Sumber : Whitten & Bentley (2007(377))

Multiplicity	UML Multiplicity Notation	Association with Multiplicity	Association Meaning
Exactly 1	1 -- or -- leave blank		An employee works for one and only one department.
Zero or 1	0..1		An employee has either one or no spouse.
Zero or more	0..* -- or -- *		A customer can make no payment up to many payments.
1 or more	1..*		A university offers at least 1 course up to many courses.
Specific range	7..9		A team has either 7, 8, or 9 games scheduled.

2.9.5. Sequence Diagram

Sequence Diagram merupakan diagram yang menggambarkan interaksi antara satu actor dan sistem untuk skenario *use case*. Satu *sequence diagram* kemungkinan memiliki beberapa diagram untuk satu buah *use case*. Menurut Whitten & Bentley (2007:394) sebuah *sequence diagram* digambarkan dengan komponen sebagai berikut :

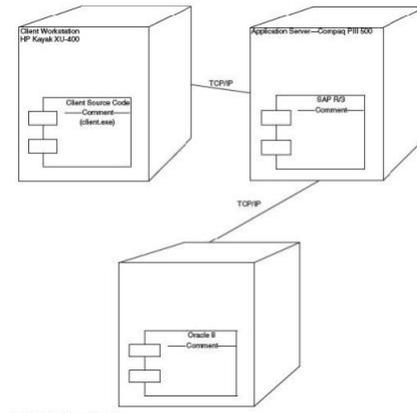


Gambar 2.16 Sequence Diagram

(Sumber : Whitten & Bentley (2007(395))

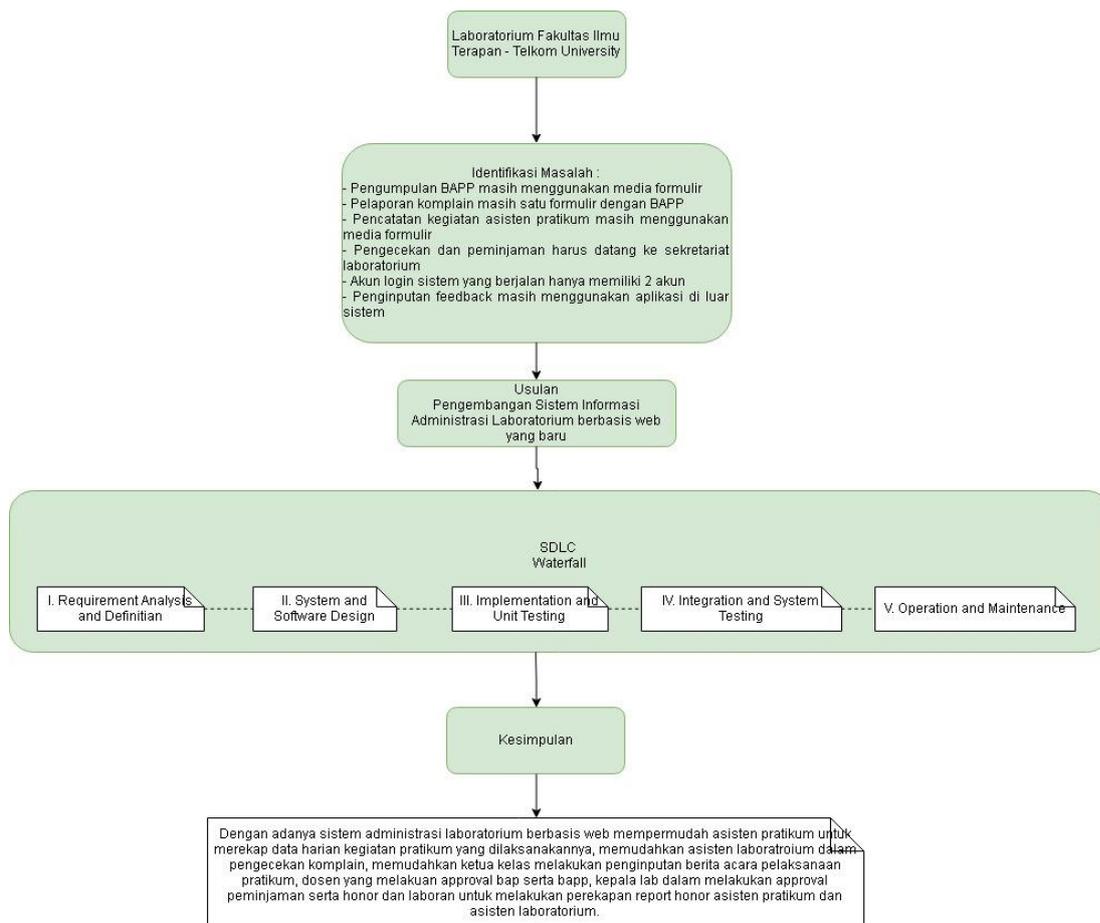
2.9.6. Deployment Diagram

Deployment diagram adalah diagram yang digunakan memetakan *software* ke *processing node*. Menunjukkan konfigurasi elemen pemroses pada saat *run time* dan *software* yang ada di dalamnya. Diagram Ini adalah salah satu diagram paling penting dalam tingkat implementasi perangkat lunak dan kadang-kadang ditulis sebelum coding. Dengan menggunakan *deployment diagram*, kita dapat menentukan ruang yang tersedia dan waktu eksekusi yang tersedia oleh perangkat keras. Menurut Whitten & Bentley (2010:673) *deployment diagram* adalah tipe implementasi diagram yang mendeskripsikan arsitektur fisik antara *hardware* dan *software* dalam sebuah sistem. Memisahkan antara komponene *software*, prosesor, dan arsitekur yang membangun sistem.



Gambar 2.17 Deployment Diagram
(Sumber : Whitten & Bentley (2007(673)))

2.10. Kerangka Berpikir



Gambar 2.18 Kerangka Berpikir