

## B A B 2

### LANDASAN TEORI

#### 2.1 Pengertian Sistem Informasi

##### 2.1.1 Pengertian Sistem

Sistem memiliki berbagai arti menurut beberapa buku, diantaranya :

1. Menurut Mulyadi (2016:5) Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan.
2. Menurut Romney dan Steinbart (2015: 3) sistem adalah suatu rancangan yang terdiri dari dua atau lebih komponen yang berhubungan dan saling berinteraksi untuk mencapai suatu tujuan.
3. Sebuah sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem - subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien. (Eka Iswandi 2016)

Berdasarkan pengertian di atas dapat disimpulkan bahwa sistem adalah suatu rancangan yang terdiri dari dua atau lebih komponen yang dibuat menurut pola yang terpadu agar dapat mencapai suatu tujuan.

##### 2.1.2 Pengertian Informasi

Informasi merupakan suatu hal penting di dalam sebuah perusahaan dalam suatu pengambilan keputusan. Informasi berasal dari Bahasa Perancis yaitu *Informacion* yg diambil dari bahasa Latin *Informationem* yang artinya garis besar, konsep atau ide. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. (Rini Asmara 2016).

Menurut Rainer, Prince dan Cegielski (2014:12) Informasi adalah data yang sudah diatur sehingga mempunyai arti dan nilai bagi penerimanya.

Pengertian informasi menurut Gordon B Davis yang dikutip oleh Rini Asmara (2016) adalah data yang telah diolah menjadi suatu bentuk yang

penting bagi penerima dan mempunyai nilai nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang

Berdasarkan pengertian-pengertian di atas dapat disimpulkan bahwa informasi adalah data yang dapat diolah menjadi suatu bentuk yang dapat digunakan untuk suatu tujuan tertentu misalnya dalam pengambilan sebuah keputusan.

### 2.1.3 Pengertian Sistem Informasi

Eka Iswandi (2015:72) mengutip pernyataan Suryadi Ph.D yang menjelaskan “Pengertian sistem informasi dapat dilihat dari segi fisik dan fungsinya. Dari segi fisiknya dapat diartikan susunan yang terdiri dari perangkat keras, perangkat lunak dan tenaga pelaksanaannya yang secara bersama-sama saling mendukung untuk menghasilkan suatu produk”.

Menurut Kadir (2014:9) Sistem Informasi adalah sebuah rangkaian formal dimana data dikelompokkan, diproses menjadi informasi dan disitribusikan kepada pemakai.

Menurut Satzinger (2015:4) Sistem Informasi adalah kumpulan seperangkat komponen yang saling berkaitan yang mengumpulkan, memproses, menyimpan dan menyediakan sebagai output informasi yang dibutuhkan untuk menyelesaikan sebuah proses bisnis. Sistem informasi selalu mencakup orang-orang yang mengoperasikan sistem dan melakukan beberapa pekerjaan.

Menurut Rainer (2014:30) Sistem Informasi dibagi menjadi 6 komponen yaitu :

#### 1. Komponen Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) terdiri dari beberapa perangkat seperti misalnya prosesor, layar monitor, keyboard, mouse, printer dan lain sebagainya.

#### 2. Perangkat Lunak (*Software*)

Perangkat Lunak (*Software*) adalah program atau sekumpulan program yang memungkinkan perangkat keras (*hardware*) untuk memproses suatu data.

### 3. Penyimpanan Data (*Database*)

Penyimpanan Data atau *database* adalah kumpulan file atau tabel yang berisi data.

### 4. Jaringan (*Network*)

Jaringan adalah sistem penghubung (*wireline* atau *wireless*) yang memungkinkan komputer yang berbeda untuk saling berbagi sumber daya.

### 5. Prosedur

Prosedur adalah instruksi untuk menggabungkan komponen-komponen di atas untuk memproses informasi dan menghasilkan output yang diinginkan.

### 6. Manusia

Manusia adalah individual yang menggunakan perangkat keras dan perangkat lunak, berinteraksi dengan itu dan memanfaatkan hasilnya.

Menurut definisi tersebut maka dapat disimpulkan bahwa sistem informasi adalah sebuah rangkaian data yang dapat disusun atau dikelompokkan untuk kemudian didistribusikan kepada pengguna.

## 2.2 Android

Menurut Kasman (2015:12) Android merupakan sebuah sistem operasi telepon seluler dengan komputer tablet layar sentuh yang berbasis Linux. Android dikembangkan oleh Android Inc. dengan dukungan finansial dari Google. Google kemudian mengakuisisi Android Inc. pada tanggal 17 Agustus 2005 sebagai anak perusahaannya. Android, Inc. didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White untuk mengembangkan 'perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunanya'. Android berisi sistem operasi, *middleware* dan aplikasi-aplikasi dasar. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi (*open source*). Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel serta pengembang aplikasi. Android memungkinkan penggunanya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti Google Play, Amazon Appstore, atau dengan mengunduh dan memasang

berkas APK dari situs pihak ketiga. Basis *Operating System* Android adalah kernel linux 2.6 yang telah dimodifikasi untuk *mobile device*. Ponsel Android yang pertama kali dijual adalah HTC Dream yang diluncurkan pada 22 Oktober 2008.

### 2.3 *System Development Life Cycle (SDLC)*

SDLC adalah pendekatan bertahap untuk analisa dan perancangan yang menyatakan bahwa sistem paling baik dikembangkan melalui penggunaan siklus yang spesifik oleh aktivitas analis dan pengguna (Kendall 2014:32).

Menurut Satzinger (2015:7) *System Development Life Cycle (SDLC)* adalah sebuah *framework* yang mengidentifikasi segala jenis aktivitas yang dibutuhkan untuk meneliti, membangun, menyebarkan dan memelihara sebuah sistem informasi. Biasanya SDLC mencakup seluruh aktivitas yang dibutuhkan untuk merancang, menganalisa sistem, merancang sistem, membangun sistem, menguji sistem dan tahapan pelatihan pengguna pengembangan sistem informasi serta segala jenis kegiatan manajemen proyek lainnya yang diperlukan untuk menerapkan sistem informasi yang baru. Banyak jenis pendekatan SDLC termasuk variasi khusus untuk jenis proyek tertentu. Namun demikian setiap SDLC memiliki proses utama yang selalu dibutuhkan meskipun banyak jenis SDLC yang digunakan. Berikut adalah 6 proses utama dalam sebuah pengembangan sistem :

1. Mengidentifikasi masalah atau kebutuhan dan dapatkan persetujuan untuk memproses proyek tersebut.
2. Merencanakan dan mengawasi proyek tersebut seperti apa yang harus dilakukan, bagaimana cara melakukannya dan siapa yang melakukannya.
3. Temukan dan pahami detail masalah atau kebutuhan (apa yang dibutuhkan).
4. Rancang komponen sistem yang akan menyelesaikan permasalahan atau penuhi kebutuhan (bagaimana cara kerja sistem tersebut).
5. Rancang, uji dan integrasikan proyek tersebut.
6. Selesaikan pengujian sistem dan distribusikan sistem tersebut kepada pengguna.

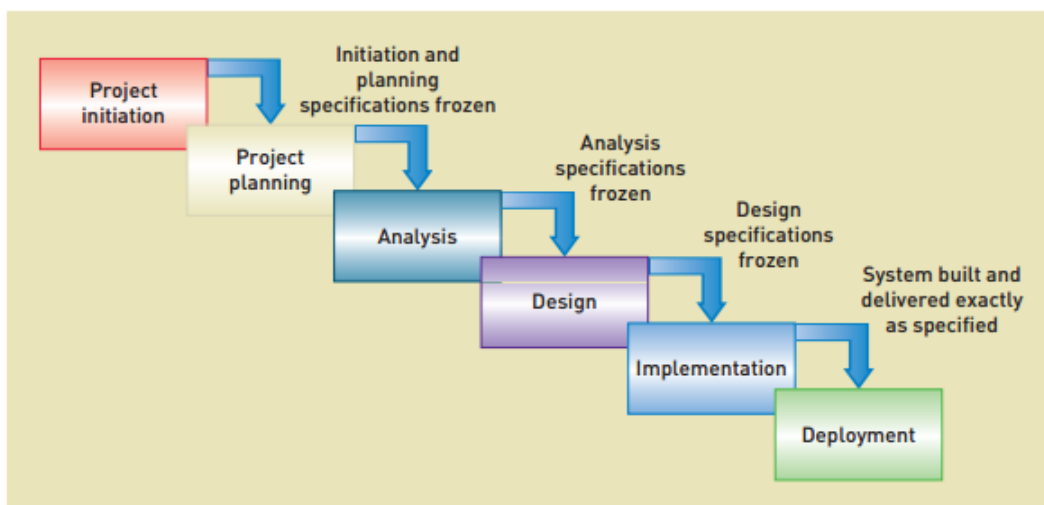
Model *waterfall* seringkali disebut sebuah siklus yang klasik, menyarankan pendekatan sistematis dan berurutan untuk mengembangkan perangkat lunak

yang dimulai dengan spesifikasi kebutuhan pelanggan dan berkembang melalui perencanaan, pemodelan, konstruksi, dan penyebaran yang berpuncak pada dukungan berkelanjutan dari sebuah pengembangan proyek perangkat lunak yang telah selesai (Pressman 2010:9).

Model SDLC yang paling sederhana adalah model *Waterfall* dengan fase proyek mengalir ke bawah (satu demi satu). Model ini mengasumsikan bahwa fase ini dapat dilakukan dan diselesaikan secara berurutan. Pertama, rencana terperinci dikembangkan, kemudian kebutuhan ditentukan secara menyeluruh, lalu sistem dirancang hingga algoritma terakhir, dan kemudian diprogram, diuji, dan dipasang. Setelah sebuah proyek sudah melaju ke fase berikutnya maka jika ada penambahan hanya dapat dilakukan setelah sistem sudah selesai (tidak dapat kembali ke proses sebelumnya). Dalam praktiknya, model *waterfall* mengasumsikan perencanaan yang kaku dan pengambilan keputusan akhir pada setiap langkah proyek pengembangan (Satzinger 2016:299).

Adapun tahapan SDLC *Waterfall* menurut Satzinger adalah sebagai berikut:

1. *Project initiation* (Inisiasi Proyek)
2. *Project planning* (Perencanaan Proyek)
3. *Analysis* (Analisa)
4. *Design* (Perancangan)
5. *Implementation* (Implementasi)
6. *Deployment* (Penyebaran/Pendistribusian)



Gambar 2.1 Model Waterfall Menurut Satzinger

Sumber Satzinger (2015:299)

### 2.3.1 *Project Initiation (Inisiasi Proyek)*

Fase pertama dalam SDLC adalah *Project Initiation* atau Inisiasi Proyek. Dalam fase ini yang dilakukan adalah mengidentifikasi masalah atau peluang bisnis, mendefinisikan solusi dan membentuk tim untuk membangun atau mengembangkan suatu proyek. Atau dengan kata lain *Project Initiation* dilakukan untuk mengidentifikasi masalah dan mendapatkan persetujuan untuk mengembangkan sistem (Satzinger 2015:228).

### 2.3.2 *Project Planning (Perencanaan Sistem)*

Perencanaan sistem berfokus pada bagaimana cara mengembangkan suatu sistem. Hal-hal yang perlu diperhatikan dalam merencanakan sebuah pengembangan sistem diantaranya adalah :

1. Pembentukan tim serta mendiskusikan dengan tim pengembang.
2. Menentukan tujuan dan ruang lingkup pengembangan.
3. Mengidentifikasi permasalahan-permasalahan yang dapat diselesaikan melalui pengembangan sistem.
4. Menentukan dan mengevaluasi strategi yang akan digunakan dalam pengembangan sistem.

### 2.3.3 Analisis Sistem

Terdapat 2 pengertian analisis sistem menurut KBBI yaitu :

1. Prosedur atau proses sistematis yang memungkinkan pengombinasian pertimbangan para pakar dari berbagai bidang ilmu sehingga diperoleh hasil yang sempurna dari kegunaan tiap disiplin.
2. Pengamatan mengenai suatu kegiatan, metode, prosedur, atau teknik untuk menentukan manfaat kegiatan tersebut dan cara terbaik untuk memperolehnya.

Sedangkan menurut Kendall (2014:30) analisa dan perancangan sistem berusaha untuk mengerti apa yang dibutuhkan manusia untuk menganalisa penguasaan data atau aliran data secara sistematis, memproses atau mengubah data, menyimpan data, dan menghasilkan informasi dalam konteks organisasi atau perusahaan tertentu.

Dengan melakukan analisa, analisis berusaha mengidentifikasi dan memecahkan masalah dengan benar. Selanjutnya analisa dan perancangan

sistem digunakan untuk menganalisa, merancang dan mengimplementasikan perbaikan untuk mendukung pengguna dan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi yang terkomputerisasi. Memasang sebuah sistem tanpa perencanaan yang memadai dapat menyebabkan ketidakpuasan pengguna dan perlahan akan menyebabkan sistem lambat laun tidak digunakan. Analisa dan perancangan sistem dapat dianggap sebagai serangkaian proses yang dilakukan secara sistematis untuk meningkatkan bisnis melalui penggunaan sistem informasi yang terkomputerisasi. Analisa dan perancangan sistem melibatkan pengguna sistem informasi saat ini dan pengguna baru untuk mendukung mereka dalam bekerja dengan teknologi dalam pengaturan organisasi (Kendall 2014:30).

Analisa sistem terdiri dari berbagai aktivitas yang memungkinkan seseorang untuk memahami dan menentukan apa yang harus diselesaikan sebuah sistem baru. Analisis sistem lebih dari sekadar sebuah pernyataan singkat suatu masalah. Misalnya sistem manajemen pelanggan harus melacak pelanggan, mendaftarkan produk, memantau jamINAN, dan melacak tingkat layanan, di antara banyak fungsi lainnya - yang semuanya memiliki banyak detail apa yang harus dilakukan sistem untuk memenuhi kebutuhan atau menyelesaikan masalah (Satzinger 2015:4).

### **2.3.3.1 System Requirement**

*System requirement* adalah semua aktivitas yang harus dipenuhi atau didukung oleh sebuah sistem baru. Pada dasarnya analisis membagi *System requirement* dalam 2 kategori yaitu :

#### **1. Functional Requirement**

*Functional Requirement* adalah segala jenis aktivitas yang harus dipenuhi oleh sistem (bisnis menggunakan sistem yang akan diterapkan). Sebagai contoh apabila kita sedang membangun sebuah sistem *payroll* karyawan, yang dibutuhkan oleh perusahaan tersebut mungkin mencakup fungsi-fungsi seperti menghasilkan transfer dana elektronik, menghitung jumlah komisi, menghitung pajak,

memelihara informasi yang bergantung pada karyawan, dan melaporkan pengurangan pajak ke IRS. Sistem yang baru harus menangani semua fungsi ini. Mengidentifikasi dan menggambar semua penggunaan bisnis ini memerlukan banyak waktu dan upaya karena daftar fungsi dan ketergantungannya bisa menjadi sangat kompleks.

*Functional Requirement* ditentukan oleh daftar kasus penggunaan untuk Sistem Tradeshow, tetapi *Functional Requirement* juga didasarkan pada prosedur dan aturan yang digunakan organisasi untuk menjalankan bisnisnya. Maka dari itu detail *Use Case* harus ditangkap dan dimodelkan.

## 2. *Nonfunctional Requirements*

*Nonfunctional requirement* adalah karakteristik sebuah sistem selain aktivitas-aktivitas diatas yang selalu harus didukung. Tidak mudah untuk membedakan *functional requirement* dan *nonfunctional requirement*. Salah satu cara membedakannya adalah menggunakan kerangka kerja untuk mengidentifikasi dan mengklasifikasikan *requirement*. Dari beberapa kerangka kerja yang berkembang seiring waktu, kerangka kerja yang paling sering digunakan adalah FURPS adalah akronim yang berarti *Functionality*, *Usability*, *Reliability*, *Performance* dan *Security*. "F" di FURPS setara dengan *functional requirement* yang ditentukan sebelumnya. Kategori FURPS yang tersisa menjelaskan *nonfunctional requirement*:

- a. *Usability requirements* menjelaskan karakteristik operasional yang berhubungan dengan pengguna (seperti antarmuka pengguna), berhubungan dengan prosedur kerja, bantuan online dan dokumentasi.
- b. *Reliability requirements* menjelaskan ketergantungan terhadap sebuah sistem



(menggambarkan ketergantungan suatu sistem (seberapa sering suatu sistem menunjukkan perilaku seperti pemadaman layanan dan salah memproses dan bagaimana mendeteksi dan mengatasi dari permasalahan tersebut).

- c. *Performance requirements* menjelaskan karakteristik operasional yang berhubungan menggambarkan karakteristik operasional terkait dengan ukuran beban kerja, seperti *throughput* dan waktu respons. Sebagai contoh, bagian klien dari suatu sistem mungkin diharuskan memiliki waktu respons satu setengah detik untuk semua penekanan tombol, dan server mungkin perlu mendukung 100 sesi klien secara bersamaan (dengan waktu respons yang sama).
- d. *Security requirements* menjelaskan bagaimana akses ke aplikasi tersebut terkontrol dan bagaimana data akan dilindungi selama penyimpanan dan pengiriman. Misalnya, aplikasi mungkin dilindungi kata sandi, mengenkripsi data yang disimpan secara lokal dengan kunci 1024-bit dan menggunakan HTTP aman untuk komunikasi antara klien dan server node.

Requirement categories	FURPS + categories	Example requirements
Functional	Functions	Business rules and processes
Nonfunctional	Usability Reliability Performance Security + Design constraints Implementation Interface Physical Support	User interface, ease of use Failure rate, recovery methods Response time, throughput Access controls, encryption Hardware and support software Development tools, protocols Data interchange formats Size, weight, power consumption Installation and updates

Gambar 2.2 Kategori *System Requirements*

Sumber Satzinger (2015:160)

#### 2.3.4 Perancangan Sistem

Perancangan sistem terdiri dari kegiatan-kegiatan yang memungkinkan seseorang untuk menjelaskan secara rinci bagaimana sistem informasi akan benar-benar dilaksanakan untuk memberikan solusi yang dibutuhkan. Dengan kata lain, perancangan sistem menggambarkan bagaimana sistem akan benar-benar bekerja. Ini menentukan secara rinci semua komponen sistem solusi dan bagaimana mereka bekerja bersama (Satzinger 2015:5).

Perancangan sistem adalah upaya membangun model, seperti halnya analisis sistem. Ketika keputusan rancangan dibuat, terutama pada level detail, keputusan tersebut diambil dan didokumentasikan oleh pembuatan model. Seperti yang ditunjukkan sebelumnya, model mungkin cukup informal, tetapi mereka adalah inti dari desain. Misalnya, dalam desain basis data, kami mengidentifikasi tabel mana yang akan diperlukan dan bidang apa yang akan berada di tabel mana sebelum kami mulai membuat tabel dengan pernyataan SQL. Dalam desain perangkat lunak, kami memutuskan kelas mana yang merupakan kelas inti dan mana kelas utilitas dan apa tanggung jawab (metode) yang akan dimiliki masing-masing kelas. Desain antarmuka pengguna sering membutuhkan storyboard atau model visual lainnya untuk membuat keputusan alur kerja yang efisien. Semua tugas desain sistem ini adalah tugas membangun model (Satzinger 2015:161).

Perancangan sistem berfungsi untuk memahami kebutuhan atau persyaratan informasi dari organisasi secara menyeluruh. Kemudian informasi akan didokumentasikan dalam serangkaian spesifikasi. Tujuan dari perancangan sistem adalah sebagai cetak biru yang mendefinisikan, mengatur, dan menyusun komponen dari solusi yang ditawarkan.

Tahapan dalam merancang sistem terdiri dari 6 bagian yaitu:

1. *Design the environment.*
2. *Design application architecture and software.*
3. *Design user interfaces.*
4. *Design system interfaces.*
5. *Design the database.*
6. *Design system controls and security.*

Design activity	Key question
Design the environment	Have we specified in detail the environment and all the various options in which the software will execute?
Design application architecture and software	Have we specified in detail all the elements of the software and how each use case is executed?
Design system interfaces	Have we specified in detail how the system will communicate with all other systems inside and outside the organization?
Design user interfaces	Have we specified in detail how users will interact with the system to carry out all their tasks (use cases)?
Design the database	Have we specified in detail all the information storage requirements, including all the schema elements?
Design system controls and security	Have we specified in detail all the elements to ensure the system and the data are secure and protected?

Gambar 2.3 Tahapan Perancangan Sistem

Sumber Satzinger (2015:161)

Dari keenam tahapan pengembangan *software* yang diidentifikasi di atas, penulis akan mengerjakan sampai tahapan ketiga yaitu menggambarkan sistem secara keseluruhan sehingga dapat diketahui secara detail apa saja yang sistem dapat lakukan. Kemampuan sistem dapat dijelaskan dalam bentuk *Unified Modeling Language* (UML) yang bertujuan untuk merepresentasikan sistem yang akan dirancang dalam bentuk visual.

#### 2.3.4.1 *Unified Modeling Language (UML)*

*Unified Modeling Language* yang sering disingkat dengan UML adalah seperangkat standar model konstruksi dan notasi yang didefinisikan oleh *Object Management Group (OMG)*, yang merupakan organisasi standart dalam pengembangan sistem (Satzinger, 2015:59). UML dapat digunakan untuk menganalisa model *Use Case* untuk mendapatkan objek sistem dan interaksinya satu sama lain dan dengan pengguna sistem. Dengan menggunakan teknik UML, Anda menganalisis objek lebih lanjut dan interaksinya untuk mendapatkan perilaku, atribut, dan hubungan objek. (Kendall 2014:287). Dengan menggunakan UML analis dan pengguna mampu menggambarkan dan memahami berbagai diagram yang spesifik yang digunakan dalam sebuah proyek pengembangan sistem.

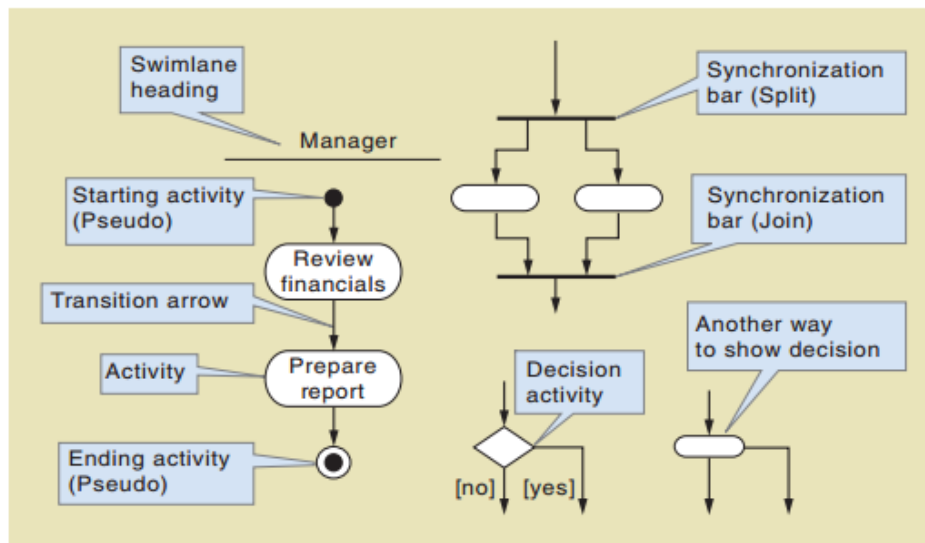
##### 2.3.4.1.1 *Activity Diagram*

*Activity Diagram* menunjukkan urutan kegiatan dalam suatu proses, termasuk kegiatan dan keputusan berurutan dan paralel. *Activity Diagram* biasanya dibuat untuk satu kasus penggunaan dan dapat menunjukkan skenario yang berbeda (Kendall 2014:290).

*Activity Diagram* menjelaskan berbagai aktivitas pengguna (atau sistem), orang atau komponen yang melengkapi setiap aktivitas, dan aliran berurutan dari aktivitas ini. (Satzinger 2015:60). *Activity Diagram* adalah diagram yang menggambarkan aliran proses bisnis, tahapan pada suatu *Use Case*, atau logika dari karakteristik suatu objek.

Gambar di bawah ini menunjukkan simbol dasar yang digunakan dalam diagram aktivitas. Oval pipih mewakili aktivitas individu dalam alur kerja. Panah penghubung mewakili urutan antara aktivitas. Lingkaran hitam menunjukkan awal dan akhir alur kerja. Berlian adalah titik keputusan di mana aliran proses akan mengikuti satu jalur atau lainnya. Garis padat yang berat adalah bilah

sinkronisasi, yang membagi jalur menjadi beberapa jalur konkuren atau menggabungkan kembali jalur konkuren. *Swim lane* mewakili agen yang melakukan kegiatan. Ini disebut *swim lane* karena masing-masing agen mengalirkan jalur sejajar dengan agen lain dalam alur kerja, seperti halnya perenang di kolam renang (Satzinger 2015:60).







Gambar 2.4 Notasi *Activity Diagram*  
 Sumber Satzinger (2015:60)

2.3.4.1.2 *Use Case Diagram*

*Use Case* adalah aktivitas yang dilakukan sistem, biasanya sebagai respons terhadap permintaan oleh pengguna. *Use Case* diagram merupakan alat yang dapat digunakan untuk menganalisis dalam menemukan dan mengidentifikasi skenario bisnis serta reaksi sistem terhadap *event* tersebut.

Berikut adalah komponen yang digunakan dalam *Use Case* diagram :

	System Boundary menggambarkan batasan antara sistem dengan actor.
	Simbol ini menggambarkan interaksi antara actor dengan software aplikasi tersebut.
	Actor menggambarkan pengguna dari sistem, dapat berupa manusia atau sistem terotomatisasi lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
	Menggambarkan hubungan antar actor dan use case.

Gambar 2.5 Komponen *Use Case diagram*

Sumber Indrajani (2015:5)

#### 2.3.4.1.3 *Use Case Description*

Daftar *Use Case* dan *Use Case Diagram* memberikan gambaran umum dari semua kasus penggunaan untuk suatu sistem. Informasi terperinci tentang setiap *Use Case* dijelaskan dengan deskripsi *Use Case*. *Use Case Description* adalah sebuah model tekstual yang mendaftar dan menjelaskan detail pemrosesan untuk sebuah *Use Case*. (Satzinger 2015:133)

*Use Case Description* adalah deskripsi yang menjelaskan proses detail untuk masing-masing *Use Case*. Setiap detail informasi dalam *Use Case* digambarkan dengan deskripsi *Use Case*. *Use Case description* terdiri dari 3 level yaitu :

1. *Brief description Use Case description* menjelaskan proses detail di dalam *Use Case* dimana tingkat detailnya hanya sebatas deskripsi *Use Case*. *Brief description* umumnya digunakan untuk proses yang tidak terlalu kompleks

2. *Intermediate description* adalah perluasan dari *briefdescription* yang memasukkan arus aktivitas-aktivitas internal untuk suatu *Use Case*. Semua aktivitas harus dideskripsikan untuk setiap skenario dan jika diperlukan dapat juga mendokumentasikan kondisi-kondisi pengecualian.
  
3. *Fully developed description* adalah metode yang paling formal dalam mendokumentasikan sebuah *Use Case*. *Fully develop description* menjelaskan detail lebih banyak sehingga dapat memberikan gambaran bagaimana internal flow dari suatu aktivitas terjadi. *Use Case description* yang *fully develop* sulit dibuat jika *user requirement* tidak tersedia. Namun *fully develop description* memiliki kelebihan dapat memberikan pengertian menyeluruh dari proses bisnis dan menjelaskan bagaimana cara sistem dapat mendukung proses tersebut.

#### 2.3.4.1.4 *Class diagram*

*Class diagram* digunakan untuk menunjukkan kelas objek untuk suatu sistem. Salah satu jenis *class diagram* yang menunjukkan hal-hal dalam domain masalah pengguna disebut *class diagram model domain*. Jenis lain dari *class diagram* disebut *design class diagram*, dan digunakan saat merancang kelas perangkat lunak (Satzinger 2015:101).

#### 2.3.4.1.5 *Sequence Diagram*

*System Sequence Diagram* (SSD) digunakan untuk menggambarkan aliran informasi masuk dan keluar dari sistem otomatis. Dengan demikian, SSD mendokumentasikan input dan output dan

mengidentifikasi interaksi antara aktor dan sistem. SSD adalah jenis diagram interaksi (Satzinger 2015:126).

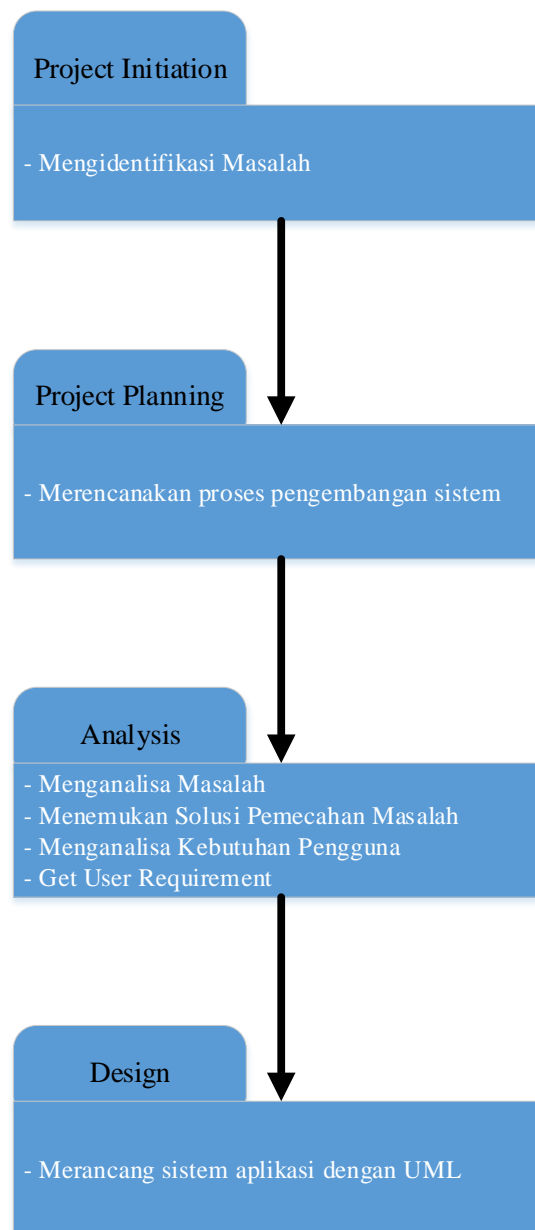
#### **2.3.4.1.6 User Interface**

*User Interface* atau antarmuka pengguna adalah input dan output yang berhubungan langsung dengan pengguna. User Interface dapat digunakan untuk kebutuhan internal atau eksternal. Desain mereka sangat bervariasi tergantung pada faktor-faktor seperti tujuan antarmuka, karakteristik pengguna, dan karakteristik perangkat antarmuka tertentu. Misalnya, meskipun semua antarmuka pengguna harus dirancang untuk kemudahan penggunaan maksimal, pertimbangan lain, seperti efisiensi operasional, mungkin penting bagi pengguna internal yang dapat dilatih untuk menggunakan antarmuka spesifik yang dioptimalkan untuk perangkat perangkat keras tertentu (misalnya, *keyboard*, *mouse*, dan layar besar beresolusi tinggi). Sebaliknya, antarmuka pengguna yang sangat berbeda mungkin dirancang untuk sistem yang menghadapi pelanggan yang menganggap telepon seluler sebagai perangkat input/output. Dalam sebagian besar proyek pengembangan sistem, analisis memisahkan desain antarmuka sistem dari desain antarmuka pengguna karena masing-masing membutuhkan keahlian dan teknologinya sendiri. Tetapi seperti halnya dengan desain komponen sistem apa pun, diperlukan koordinasi yang besar. (Satzinger 2015:189)



## 2.4 Kerangka Berpikir

Berikut ini adalah gambar kerangka berpikir Sistem Aplikasi *Mobile Reminder Monitoring* Gangguan dan Keluhan Pelanggan :



Gambar 2.6 Kerangka Berpikir

