

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem**

Pengertian sistem menurut beberapa ahli yaitu, Menurut (Tata Sutabri, 2012) pada buku Analisis Sistem Informasi, pada dasarnya sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Menurut Gelinas dan Dull (2012), Sistem merupakan seperangkat elemen yang saling bergantung yang bersama-sama mencapai tujuan tertentu. Dimana sistem harus memiliki organisasi, hubungan timbal balik, integrasi dan tujuan pokok. Dengan demikian dapat disimpulkan bahwa sistem merupakan seperangkat elemen yang saling berhubungan yang bersama-sama mencapai suatu tujuan tertentu dalam proses yang teratur yang dapat mendukung sistem yang lebih besar dan saling memiliki ketergantungan untuk mencapai tujuan tertentu.

#### **2.2 Informasi**

Menurut McLeod dikutip oleh Yakub (2012) pada buku Pengertian Sistem Informasi, Informasi adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Sedangkan Menurut (Tata Sutabri, 2012) pada buku Analisis Sistem Informasi, Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

#### **2.3 Sistem Informasi**

Menurut Satzinger, Jackson, dan Burd (2012), Sistem informasi merupakan kumpulan dari komponen-komponen yang mengumpulkan, memproses, menyimpan, dan menyediakan *output* dari setiap informasi yang dibutuhkan dalam proses bisnis serta aplikasi yang digunakan melalui perangkat lunak, *database* dan bahkan proses manual yang terkait.

Menurut (Stair and Reynolds, 2012), Sistem Informasi adalah suatu sekumpulan elemen atau komponen berupa orang, prosedur, database dan alat yang saling terkait untuk memproses, menyimpan serta menghasilkan informasi untuk mencapai suatu tujuan (goal). Menurut Gelinas dan Dull (2012) Sistem Informasi adalah sistem yang di buat secara umum berdasarkan seperangkat komputer dan komponen manual yang dapat dikumpulkan, disimpan dan diolah untuk menyediakan output kepada user. Dengan demikian dapat disimpulkan bahwa sistem informasi adalah suatu kombinasi modul yang terorganisir yang berasal dari komponen-komponen yang terkait dengan *hardware, software, people* dan *network* berdasarkan seperangkat komputer dan menghasilkan informasi untuk mencapai tujuan.

## 2.4 Website

Menurut (Arief, 2011) *website* adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, animasi, video) didalamnya yang menggunakan protokol HTTP (*Hypertext Transfer Protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*. Menurut (Sibero, 2011) *Website* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan *internet*. *Website* adalah sering juga disebut *Web*, dapat diartikan suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau *hyperlink*.

Atau definisi *website* adalah kumpulan dari berbagai macam halaman situs, yang terangkum didalam sebuah domain atau juga subdomain, yang lebih tepatnya berada di dalam WWW (*World Wide Web*) yang tentunya terdapat di dalam Internet. Halaman *website* biasanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language* (HTML), yang bisa diakses melalui HTTP, HTTP adalah suatu

protokol yang menyampaikan berbagai informasi dari server website untuk ditampilkan kepada para user atau pemakai melalui web browser berada pada layar sebuah *smartphone* dan tentu juga di pengaruhi oleh keterbasan ukuran layar dan jenis layar sentuhnya.

Di dalam *web* mempunyai prinsip dalam sistemnya. yakni *develope once run everywhere* yang dapat berartikan bahwa web ini beroperasi dalam lintas platform untuk sekali pengembangan. jadi kesimpulanya *web* lebih relatif tidak membutuhkan banyak biaya dalam proses pengembangan jika di bandingkan dengan mobile aplikasi di sebabkan dapat berjalannya dengan baik di semua browser dan platform. web juga dapat di distribusikan tanpa harus mendapatkan approve dari pihak tertentu serta pengguna tidak perlu susah payah mencari di *web*. namun dengan demikian bukan berarti web ini tidak memiliki kekurangan. Tetap saja *web* ini mempunyai kekurangan namanya juga sistem pasti tidak lepas dengan kekurangan meskipun tidak membuat rugi pengguna secara financial atau material. dan kekurangan web ini terdapat pada domain yang dimana pengguna web harus memiliki 2 buah domain dan bila ada gangguan atau masalah maka akan rumit dalam memperbaikinya

## 2.5 PHP (*Pear Hypertext Preprocessor*)

PHP singkatan dari *Pear Hypertext Preprocessor* yaitu bahasa pemrograman web *server-side* yang bersifat open source. Menurut Betha Sidik, dalam bukunya yang berjudul Pemrograman Web Dengan PHP (Betha Sidik, 2012), menyebutkan bahwa PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman *script – script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*". PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta

oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

Keunggulan PHP :

- A. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- B. Dalam sisi pengembangannya lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
- C. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
- D. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin ( *linux, unix, windows* ) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.
- E. PHP memiliki tingkat akses yang lebih cepat.
- F. PHP memiliki tingkat keamanan yang tinggi.
- G. PHP mampu berjalan di beberapa *server* seperti *Apache, Microsoft IIS, PWS* dan *Xitami*.
- H. PHP mendukung akses ke beberapa *database* baik yang bersifat *free/gratis* atau *komersial*, seperti *MySQL, mSQL* dan *Microsoft SQL Server*.
- I. PHP bersifat *free/gratis*.

## 2.6 MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user. Menurut (Betha Sidik, 2012), dalam bukunya yang berjudul Pemrograman Web Dengan PHP, menyebutkan bahwa “MySQL merupakan software database yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari database nya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah”.MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuat oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris.

MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL(*General Public License*) namun tidak boleh menjadikan produk turunan yang bersifat komersial. Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. SQL (*Structured Query Language*) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangkan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni. SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemograman seperti C, dan Delphi

## 2.7 DataBase

Menurut Sutarman (2012), *Database* sekumpulan file yang saling berhubungan dan terorganisasi atau kumpulan record-record yang menyimpan data dan hubungan diantaranya. Menurut (Ladjamudin, 2013), *Database* adalah sekumpulan data store (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya. Dari pengertian diatas dapat disimpulkan Database adalah sekumpulan file yang saling berhubungan untuk menyimpan data dan tersimpan dalam sebuah media penyimpanan.

## 2.8 Dokumen

Dokumen adalah suatu rekaman yang menyimpan berbagai informasi penting dan merupakan pertanggungjawaban kerja yang harus selalu dipelihara, dirawat dan dilindungi keberadaannya (Hidayatul, Yeni, 2013). Istilah dokumen dalam tiga pengertian. Pertama dalam arti luas, yaitu yang meliputi semua sumber, baik sumber

tertulis maupun sumber lisan. Kedua dalam arti sempit, yaitu yang meliputi semua sumber tertulis saja. Ketiga dalam arti spesifik, yaitu hanya yang meliputi surat-surat resmi dan suratsurat negara, seperti surat perjanjian, undang-undang, konsesi, hibah dan sebagainya (Natalina Nilamsari, 2014).

## 2.9 E-dokumen

Dokumen elektronik atau yang biasa disebut dengan e-document adalah setiap konten elektronik (selain program komputer atau file sistem) yang dimaksudkan untuk digunakan baik dalam bentuk elektronik atau dicetak sebagai output. Secara teknis, proses terciptanya sebuah e-dokumen tidak terlalu rumit. Manfaat e-dokumen salah satunya adalah untuk meminimalisasi tempat karena setumpukan data atau buku bisa dirangkum menjadi sebuah disket kecil. Untuk mencapai hasil tersebut harus melalui beberapa proses (Hidayatul, Yeni, 2013).

## 2.10 Metode Pengembangan Sistem

Dalam penelitian ini penulis menggunakan metode pendekatan yaitu *Object Oriented Analysis and Design (OOAD)* digunakan untuk memodelkan system yang dikembangkan . *Object Oriented Analysis and Design (OOAD)* dapat di definisikan menjadi dua bagian yaitu:

1. *Object Oriented Analysis* adalah metode analisis yang memeriksa *requirement* (syarat atau keperluan yang harus di penuhi sistem) di lihat dari kelas dan objek yang ditemui dalam pengembangan sistem.
2. *Object Oriented Design* adalah metode untuk mengarahkan arsitektur software sesuai dengan manipulasi sistem maupun subsistem.

## 2.11 Alat Dan Pengembangan Sistem

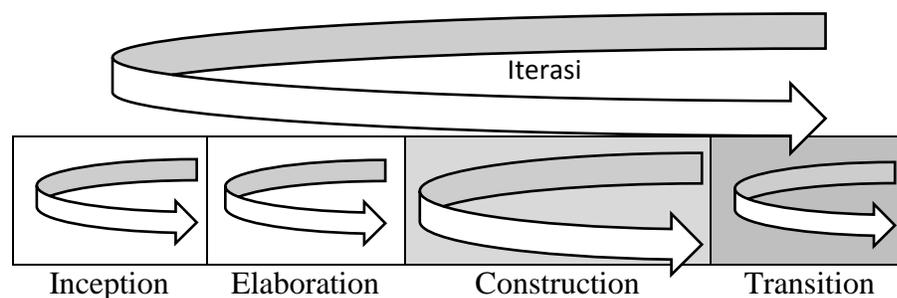
### 2.11.1 Pemodelan

Pada dunia pembangunan perangkat lunak sistem informasi juga diperlukan permodelan. Permodelan perangkat lunak digunakan untuk mempermudah langkah berikutnya dari pengembangan sebuah sistem informasi sehingga lebih terencana. Seperti halnya maket (tiruan), permodelan pada pembangunan perangkatan lunak digunakan untuk memvisualkan perangkat lunak yang akan dibuat (Rosa & Shalahuddin, 2018)

### 2.11.2 RUP

*Unified Process* atau dikenal juga dengan proses iteratif dan inkremental merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan inkremental (bertahap dengan proses menaik). Iteratif bisa dilakukan di dalam setiap tahap atau iteratif tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental, dimana setiap iterasi akan memperbaiki iterasi berikutnya (Rosa & Salahuddin, 2018). Salah satu *Unified Process* yang terkenal adalah RUP (*Rational Unified Process*).

RUP adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang, fokus pada arsitektur, lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik dan penstrukturan yang baik. RUP memiliki empat buah tahap fase, yaitu seperti pada Gambar 2.1.



Gambar 2.1 Alur Hidup RUP (Sumber : Rosa, 2018)

a. *Inception* (permulaan)

Tahap ini lebih pada memodelkan bisnis yang dibutuhkan dan mendefinisikan kebutuhan akan sistem yang akan dibuat. Tahap yang dibutuhkan pada permulaan ini adalah :

1. Memahami ruang lingkup dari proyek (termasuk biaya, waktu, kebutuhan, resiko dan lainnya).
2. Membangun kasus bisnis yang dibutuhkan.

Hasil yang diharapkan pada tahap ini adalah memenuhi *lifecycle objective milestone* (batas/tonggak objektif dari siklus) dengan kriteria berikut :

1. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya dan perkiraan jadwal.
2. Kebutuhan dimengerti dengan pasti dan sejalan dengan kasus primer yang dibutuhkan.
3. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, risiko dan proses pengembangan.
4. Ruang lingkup purwarupa (*prototype*) yang akan dikembangkan.
5. Membangun garis dasar dengan membandingkan perencanaan aktual dengan perencanaan yang direncanakan.

Jika pada akhir tahap ini target yang diinginkan tidak dicapai maka dapat dibatalkan atau diulang kembali setelah dirancang ulang agar kriteria yang diinginkan dapat dicapai.

b. *Elaboration* (perluasan atau perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*). Hasil yang diharapkan pada tahap ini adalah

memenuhi *lifecycle objective milestobe* (batas/tonggak objektif dari siklus) dengan kriteria berikut :

1. Model kasus yang digunakan (*use case*) dimana kasus dan aktor yang terlihat telah didefinisikan dan sebagian besar kasus harus dikembangkan.
2. Deskripsi dari arsitektur perangkat lunak telah dibuat.
3. Rancangan arsitektur yang dapat diimplementasikan dan mengimplementasikan *use case*.
4. Kasus bisnis atau proses bisnis dan daftar resiko yang sudah mengalami perbaikan.
5. Rencana pengembangan untuk seluruh proyek telah dibuat.
6. Purwarupa (*prototype*) yang dapat didemonstrasikan untuk mengurangi setiap resiko teknis yang diidentifikasi.

Jika pada akhir tahap ini target yang diinginkan tidak dicapai, maka dapat dibatalkan atau diulang kembali.

c. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak atau kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

d. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau inisialisasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem.

### 2.11.3 UML (Unified Modeling Language)

*Unified Modeling Language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa & Shalahudin, 2018).

Start UML (*Unified Modeling Language*) yang merupakan *tools open sources* yang dapat bebas dikembangkan oleh semua pengguna, UML sendiri merupakan alat bantu analisis dan perancangan system perangkat lunak berorientasi objek.

Ada tiga komponen pembentuk dalam Pengembangan Perangkat Lunak dengan UML, yaitu :

1. *Things*

*Things* sendiri mempunyai pembagian yang spesifik, yaitu:

- a. Struktur

Merupakan bagian yang relative statis dalam UML, biasanya berupa elemen fisik atau konseptual.

- b. *Behavior*

Merupakan bagian yang paling dinamis dalam UML, biasanya berupa kata kerja dan perilaku.

- c. *Grouping*

Merupakan bagian pengorganisasian dalam UML, biasanya digunakan untuk menyederhanakan model.

d. *Annotational*

Merupakan bagian yang memperjelas model dalam UML, biasanya berupa komentar atau pesan.

2. Relasi

Ada 4 (empat) macam relational dalam UML, yaitu:

a. Ketergantungan

Merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) mempengaruhi elemen yang tidak mandiri.

b. Asosiasi

Suatu yang menghubungkan antara objek satu dengan objek lainnya, atau disebut agregasi yaitu hubungan antara suatu objek dengan bagian-bagiannya.

c. Generalisasi

Merupakan hubungan dimana objek anak (*descendent*) berbagi perilaku dengan objek di atasnya atau objek induk (*ancestor*).

d. Realisasi

Merupakan operasi yang benar-benar dilakukan oleh suatu objek.

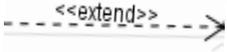
3. Diagram

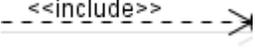
Berikut adalah macam-macam diagram dalam UML, yaitu :

a. *Use Case Diagram*

*Use case diagram* Pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *use case* mendeskripsikan sebuah interaksi satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa, Salahuddin, 2018). Dalam konteks UML, tahap awal atau konseptual dilakukan dengan pembuatan *usecase* diagram yang merupakan deskripsi peringkat tinggi bagaimana perangkat lunak. Penggunaan *use case* menjadi sangat penting karena digunakan dalam tahap analisis, perancangan, dan pencarian kelas-kelas yang terlibat dalam aplikasi, untuk melakukan pengujian (*testing*). Simbol-simbol

yang ada pada *UseCase* diagram dapat dilihat pada tabel 2.1 berikut :

No	Simbol	Keterangan
1.	<i>UseCase</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2.	<i>Actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor adalah orang namun aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<i>Association</i> 	Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada usecase atau <i>Use Case</i> memiliki interaksi dengan aktor.
4.	Ekstensi/ <i>Extend</i> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>Use Case</i> tambahan itu.
5.	<i>Generalization</i> 	Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya

6.	<i>Include</i>  	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini.
----	---	--

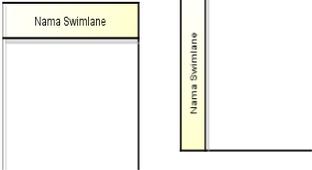
Tabel 2.1 Simbol – simbol pada *use case diagram*

b. *Activity Diagram*

*Activity diagram* Menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa & Salahuddin, 2018). Sebelum membuat *activity diagram* terlebih dahulu harus dideskripsikan dengan skenario sesuai dengan aktivitas yang terjadi, pada dasarnya *activity diagram* hampir mirip dengan diagram alir (*flow chart*) yang membedakan hanya *swimlane* yang menunjukkan suatu *state* berbeda pada objek/kelas tertentu. Simbol – simbol yang ada pada *activity diagram* dapat dilihat pada tabel berikut ini:

Simbol	Keterangan
Status awal  	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas  	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i>  	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

Simbol	Keterangan
--------	------------

Simbol	Keterangan
Penggabungan/ join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Tabel 2.2 Simbol-simbol pada *activity diagram*

c. *Class Diagram*

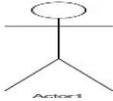
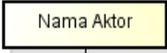
*Class diagram* Menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system (Rosa & Salahuddin, 2018). Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Simbol – simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.3 berikut ini :

<p>Kelas/Class</p> <table border="1" style="margin-left: 20px;"> <tr> <td>Nama_kelas</td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>+operasi()</td> </tr> </table>	Nama_kelas	+atribut	+operasi()	<p>Kelas pada struktur system</p>
Nama_kelas				
+atribut				
+operasi()				
<p>Antar muka/<i>interface</i></p> <p style="text-align: center;">○</p>	<p>Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek</p>			
<p>Asosiasi/<i>association</i></p> <p style="text-align: center;">—————</p>	<p>Relasi antar kelas dengan makna umum ,asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>			
<p>Asosiasi berarah/<i>directed</i></p> <p style="text-align: center;">—————&gt;</p>	<p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>			
<p>Generalisasi</p> <p style="text-align: center;">—————▷</p>	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)</p>			
<p>Kebergantungan atau</p> <p><i>Dependence</i></p> <p style="text-align: center;">.....→</p>	<p>Relasi antar kelas dengan makna kebergantungan antar kelas.</p>			
<p>Agregasi/<i>aggregation</i></p> <p style="text-align: center;">—————◊</p>	<p>Relasi antar kelas dengan makna semua-bagian(whole-part)</p>			

Tabel 2.3 Simbol-simbol pada *class diagram*

d. *Sequence Diagram*

*Sequence diagram* menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek atau *message* yang dikirimkan dan diterima antar objek system (Rosa & Salahuddin, 2018). Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas. Simbol – simbol yang ada pada *sequence diagram* dapat dilihat pada tabel 2.4 berikut ini :

Simbol	Keterangan
<p>Aktor</p>  <p>atau </p> <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.</p>
<p>Garis hidup</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatukan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe <i>create</i></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah</p>

Simbol	Keterangan
<<create>>	mengarah pada objek yang dibuat.

Tabel 2.4 Simbol-simbol pada *sequence diagram*

## 2.12 Teknik Pengumpulan Data

Hal pertama yang dilakukan dalam analisis sistem adalah melakukan pengumpulan data (Rosa & Salahuddin, 2018). Ada beberapa teknik pengumpulan data yang sering dilakukan yaitu sebagai berikut.

- a. Teknik Wawancara
- b. Teknik Observasi
- c. Teknik Kuisioner

### 2.12.1 Wawancara

Pengumpulan data dengan menggunakan wawancara mempunyai beberapa keuntungan sebagai berikut (Rosa & Salahuddin, 2018).

- a. Lebih mudah dalam menggali bagian sistem mana yang dianggap baik dan bagian mana yang dianggap kurang baik.
- b. Jika ada bagian tertentu yang menurut anda perlu untuk digali lebih dalam, anda dapat langsung menanyakan pada narasumber.
- c. Dapat menggali kebutuhan user secara lebih bebas.
- d. User dapat mengungkapkan kebutuhannya secara lebih bebas.

### 2.12.2 Observasi

Pengumpulan data dengan menggunakan observasi mempunyai keuntungan yaitu (Rosa & Salahuddin, 2018).

- a. Analis dapat melihat langsung bagaimana sistem lama berjalan.

- b. Mampu menghasilkan gambaran lebih baik jika dibanding dengan teknik lainnya.

### **2.12.3 Kuisisioner**

Pengumpulan data dengan menggunakan kuisisioner mempunyai keuntungan yaitu (Rosa & Salahuddin, 2018).

- a. Hasilnya lebih objektif, karena kuisisioner dapat dilakukan kepada banyak orang sekaligus.
- b. Waktunya lebih singkat.