



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Sistem Pendukung Keputusan

Menurut Nofriansyah dan Defit (2017:1) mengatakan bahwa Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditunjukkan untuk membantu manajemen dalam kegiatan mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur.

Sedangkan menurut Bonczek, dkk. di dalam buku *Sistem Pendukung Keputusan Teori dan Implementasi* (Latif, Lita Asyriatif, dkk., 2018) mengatakan bahwa Sistem Pendukung Keputusan adalah sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi meliputi sistem bahasa (mekanisme untuk memberikan komunikasi antar pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (repositori pengetahuan domain masalah yang ada baik sebagai data atau sebagai prosedur), sistem pemrosesan masalah (hubungan antara komponen lainnya terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

Dari dua definisi diatas dapat disimpulkan bahwa Sistem Pendukung Keputusan (*Decision Support Systems*) adalah teknik dalam pengambilan keputusan yang berbasis computer baik untuk individu maupun kelompok yang memiliki kriteria yang memiliki nilai-nilai atau bobot yang harus dimiliki oleh setiap alternatif, dimana sistem ini memberikan pilihan pada pengambilan keputusan yang lebih baik dan lebih konsisten dan lebih cepat.

Menurut Nofriansyah dan Defit (2017:2) terdapat empat komponen dari sistem pendukung keputusan yaitu sebagai berikut.

a. *Data Management.*

Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).



b. *Model Management.*

Melibatkan model finansial, statistikal, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analistis, dan manajemen *software* yang diperlukan.

c. *Communication (dialog subsystem).*

User dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.

d. *Knowledge Management.*

Subsistem *optional* ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

Menurut Simon dalam buku *Sistem Pendukung Keputusan Teori dan Implementasi* (Latif, Lita Asyriatif, dkk., 2018), terdapat tiga tahapan dalam proses pengambilan keputusan yaitu sebagai berikut.

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, diproses, dalam diuji dalam rangka mengidentifikasi masalah.

2. *Design*

Tahap ini merupakan proses menemukan, mengembangkan, dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan dan solusi.

3. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil penelitian tersebut kemudian diimplementasikan dalam proses pengambilan keputusan..

2.1.2 Pengertian Permukiman Kumuh

Menurut Aminudin Nur, dkk. (2017:206), mengatakan Perumahan dan permukiman merupakan salah satu kebutuhan dasar manusia dan merupakan faktor penting dalam peningkatan harkat serta martabat manusia. Perumahan dan permukiman juga merupakan bagian dari pembangunan nasional yang harus



ditingkatkan dan dikembangkan secara terpadu, terarah, terencana, dan berkesinambungan. Sedangkan arti dari "permukiman" itu sendiri adalah bagian dari lingkungan hidup di luar kawasan lindung, dapat merupakan kawasan perkotaan dan perdesaan, berfungsi sebagai lingkungan tempat tinggal/hunian dan tempat kegiatan yang mendukung kehidupan dan penghidupan. Sedangkan kata kumuh dapat diartikan sebagai sebagai kotor atau cemar, dimana kondisi tersebut memiliki tingkat hunian dan bangunan yang padat.

Permukiman juga merupakan suatu lingkungan hidup yang dapat diartikan sebagai kawasan perkotaan dan perdesaan, yang berfungsi sebagai lingkungan tempat tinggal/hunian dan tempat kegiatan yang mendukung kehidupan dan penghidupan. Sedangkan kumuh adalah suatu kondisi yang kotor atau cemar dan mengakibatkan lingkungan tidak nyaman. Sedangkan permukiman kumuh merupakan suatu kondisi kawasan yang buruk, kotor, serta tingkatan hunian dan kepadatan bangunan yang tinggi dan jumlah penduduk yang padat.

2.1.3 Metode *Simple Multi Attribute Rating Technique* (SMART)

menurut Nofriansyah dan Defit (2017:27), mengatakan bahwa SMART (*Simple Multi Attribut Rating Technique*) merupakan metode dalam pengambilan keputusan multiatribut yang dikembangkan oleh Edward pada tahun 1997. Teknik pembuatan keputusan multiatribut ini digunakan untuk mendukung pembuat keputusan dalam memilih antara beberapa alternatif. Setiap pembuat keputusan harus memilih sebuah alternatif yang sesuai dengan tujuan yang telah dirumuskan. Setiap alternatif terdiri dari sekumpulan atribut dan setiap atribut mempunyai nilai atau bobot dan setelah itu di rata-rata dengan skala tertentu.

Jadi, dapat disimpulkan bahwa metode SMART merupakan teknik dalam pengambilan keputusan yang bertujuan membantu pembuat keputusan dalam memilih dari beberapa alternatif yang ada, dan di setiap alternatif terdapat kriteria yang memiliki nilai-nilai atau bobot.

Model yang digunakan dalam SMART (*Simple Multi Attribut Rating Technique*) (Nofriansyah dan Defit, 2017:27) yaitu:

$$U(a_i) = \sum_{j=1}^m W_j U_i(a_i)$$

Keterangan:

1. W_j = Nilai Pembobotan Kriteria ke- j dan K- kriteria
2. $U(a_i)$ = nilai Utility kriteria ke-I untuk kriteria ke-i
Dimana $i= 1,2,\dots,m$

Adapun algoritma penyelesaian dari Metode SMART (*Simple Multi Attribut Rating Technique*) yaitu sebagai berikut:

1. Langkah 1 : Menentukan Jumlah Kriteria dari Keputusan yang akan diambil
2. Langkah 2 : Sistem secara default memberikan nilai 0-100 berdasarkan prioritas dengan melakukan normalisasi ($W_j / \sum W_j$)
3. Langkah 3 : Memberikan nilai kriteria untuk setiap alternatif
4. Langkah 4 : Menghitung nilai Utility untuk setiap kriteria masing-masing

$$u_i(a_i) = 100 \frac{C_{max} - C_{out}^i}{C_{max} - C_{min}} \%$$

Dimana:

- 1) $u_i(a_i)$ adalah nilai utiliti kriteria ke-1 untuk kriteria ke-I,
- 2) C_{max} adalah nilai kriteria maksimal
- 3) C_{min} adalah nilai kriteria minimal
- 4) C_{out}^i adalah nilai kriteria ke- i.
5. Langkah 5 : Menghitung nilai akhir dan melakukan Perangkingan

2.2 Teori Program

2.2.1 Basis Data (*Database*)

Menurut Winarno (2014:102) dalam buku *Pemrograman Web Berbasis HTML5, PHP, dan JavaScript* mengatakan bahwa basis data merupakan sebuah tempat untuk menyimpan data yang jenisnya beraneka ragam.



Menurut Pamungkas (2017:2) dalam buku *Pengantar dan Implementasi Basis Data*, “Basis data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu.

Sedangkan menurut Jubilee (2017:1) dalam buku *Otodidak MySQL untuk Pemula* mengatakan bahwa basis data adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap basis data mempunyai perintah tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data yang ada di dalamnya.

Dari ketiga definisi di atas dapat disimpulkan bahwa basis data adalah sekumpulan data yang jenisnya beraneka ragam yang saling berhubungan dan disimpan secara bersama-sama dalam sebuah media tertentu guna memperoleh informasi dari basis data tersebut.

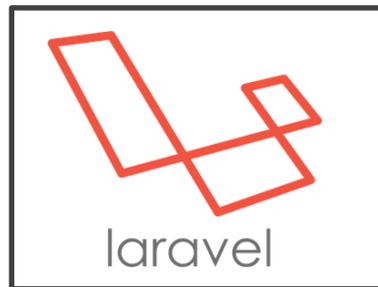
2.2.2 Web Service

Menurut Kreger (2001) dalam jurnal yang berjudul *Implementasi Web-Service Pada Aplikasi Pengisian Kartu Rencana Studi Mahasiswa* mengatakan bahwa *Web-service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet dalam bentuk pesan XML

Menurut Siswoutom (2004) dalam jurnal yang berjudul *Implementasi Web-Service Pada Aplikasi Pengisian Kartu Rencana Studi Mahasiswa* mengatakan bahwa Platform dasar *Web-Service* adalah XML ditambah HTTP. HTTP adalah protokol yang dapat berjalan di mana saja di Internet. Sedangkan XML dapat dibangun aplikasi dengan bahasa apapun untuk berkomunikasi dengan aplikasi lain yang dibangun dengan bahasa yang bisa saja sama sekali berbeda.

Jadi, berdasarkan pengertian diatas dapat disimpulkan bahwa *Web-Service* adalah XML yang ditambah HTTP, sedangkan HTTP merupakan link yang bisa diambil untuk mengetahui sebuah halaman *Protocol*, dan juga sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan

2.2.3 Framework Laravel



Gambar 2.1 Logo Laravel

Menurut Naista (2017) *framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan.

Laravel adalah *framework* berbasis *PHP* yang sifatnya *open source*, dan menggunakan konsep *model – view – controller*. Laravel berada di bawah lisensi *MIT License* dengan menggunakan Github sebagai tempat berbagi *code* menjalankannya (Naista, 2017).

Menurut Abdulloh (2017:3) terdapat beberapa keunggulan dari *framework* Laravel yaitu sebagai berikut.

1. Laravel memiliki banyak fitur yang tidak dimiliki oleh *framework* lain.
2. Laravel merupakan *framework* *PHP* yang ekspresif, artinya sintaks pada Laravel menggunakan bahasa yang mudah dimengerti sehingga *programmer* pemula sekalipun akan mudah paham kegunaan suatu sintaks walaupun belum mempelajarinya.
3. Laravel memiliki dokumentasi yang cukup lengkap, bahkan setiap versinya memiliki dokumentasi tersendiri mulai dari cara instalasi hingga penggunaan fitur-fiturnya.
4. Laravel digunakan oleh banyak *programmer* sehingga banyak *library* yang mendukung Laravel yang diciptakan para *programmer* pecinta Laravel.

5. Laravel didukung oleh Composer sehingga *library-library* diperoleh dengan mudah dari internet menggunakan Composer.
6. Laravel memiliki *template engine* tersendiri yang diberi nama *blade* yang memudahkan kita menampilkan data pada *template* HTML.

Ada beberapa fitur yang dimiliki oleh *framework* Laravel yaitu sebagai berikut (Aminudin, 2015:5).

- a) **Bundles** yaitu sebuah fitur dengan sistem pengemasan modular dan berbagai *bundle* telah tersedia untuk digunakan dalam aplikasi.
- b) **Eloquent ORM** merupakan penerapan PHP lanjutan dari pola *active record* menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*.
- c) **Application Logic** merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *controllers* maupun sebagai bagian dari deklarasi *route*. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh *framework* Sinatra.
- d) **Reverse Routing**, mendefinisikan hubungan antara *link* dan *route*. Sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *link* yang dibuat dengan menggunakan nama-nama dari *route* yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
- e) **Restful Controller**, memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.
- f) **Class Auto Loading**, menyediakan otomatis *loading* untuk kelas-kelas PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang tidak perlu.
- g) **View Composers** adalah kode unit *logical* yang dapat dijalankan ketika sebuah *view* di *load*.
- h) **IoC Container** memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip *control* pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai Singletons.



- i) **Migrations** menyediakan versi sistem *control* untuk skema *database*, sehingga memungkinkan untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak *database*, mempermudah dalam penempatan dan memperbarui aplikasi.
- j) **Unit Testing** mempunyai peran penting dalam *framework* Laravel, dimana *unit testing* ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. *Unit testing* dapat dijalankan melalui fitur “*artisan command-line*”.
- k) **Automatic Pagination** menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

2.2.4 Android

Menurut Ismail (2017) dalam jurnal yang berjudul *Aplikasi Android sebagai pengontrol Jarak Jauh Smartphone dengan Koneksi Jaringan Internet* mengatakan bahwa Android adalah sistem informasi berbasis Linux yang di modifikasi untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.

Android adalah suatu sistem operasi berbasis Linux yang dirancang untuk perangkat yang bergerak layar sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005 (Yudhanto dan Wijayanto, 2017:1)

Jadi dapat disimpulkan bahwa yang dimaksud dengan android adalah sebuah sistem yang dirancang pada perangkat yang menggunakan bahasa program berupa bahasa java dalam membangun aplikasinya seperti telepon pintar dan komputer tablet..

2.2.5 MySQL

Menurut Nugroho (2019), “MySQL merupakan database yang paling digemari kalangan Programmer Web, dengan alasan bahwa program ini merupakan database yang sangat kuat dan stabil”. Sedangkan menurut Winarno



(2014:102), “MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan.

Jadi dapat disimpulkan, MySQL merupakan sistem basis data yang bisa mengakses database yang bersifat jaringan, dan dapat digunakan untuk melakukan perintah-perintah SQL dalam mendukung pengelolaan data yang baik, dan mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system (DBMS)*.

2.2.6 Hypertext Preprocessor (PHP)

Menurut Yudhanto dan Prasetyo (2018:7), “PHP adalah bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web. Bahasa pemrograman ini dirancang untuk para pengembang web agar dapat menciptakan suatu halaman web yang bersifat dinamis.

Menurut Jubilee (2017:1) mengatakan bahwa PHP adalah bahasa pemrograman yang digunakan untuk membuat *website* dinamis dan interaktif. Dinamis artinya *website* tersebut bisa berubah-ubah tampilan kontennya sesuai kondisi tertentu, dan interaktif artinya dapat memberi *feedback* bagi *user*.

Sedangkan menurut Winarno (2014:49), “PHP adalah sebuah bahasa pemrograman web berbasis *server (server-side)* yang mampu mem-*parsing* kode PHP dari kode web dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis disisi *client (browser)*.

Dari ketiga definisi di atas dapat disimpulkan bahwa PHP adalah bahasa pemrograman web berbasis *server side* yang digunakan untuk membuat *website* yang bersifat dinamis dan tampilan kontennya sesuai kondisi tertentu, dan interaktif artinya dapat memberi *feedback* bagi *user*.

2.2.7 Laragon

Laragon adalah sebuah aplikasi mirip seperti XAMPP, namun didesain untuk kebutuhan *developer PHP* yang menggunakan *framework* Laravel. *Service* yang *include* dalam Laragon seperti: *Apache, MySQL, PHP Server, Memcached*,



Redis, Composer, Xdebug, PhpMyAdmin, Cmdr, dan lainnya. Aplikasi ini sangat cocok digunakan oleh seorang *developer PHP* yang menggunakan *framework* *Laravel*, karena akan mempermudah dalam melakukan pengembangan aplikasi.

2.2.8 *Sublime Text*



Gambar 2.2 Logo *Sublime Text*

Sublime Text Editor adalah *text editor* yang di buat untuk mempermudah pekerjaan *programmer*. *Sublime* merupakan *text editor* yang digunakan untuk banyak sekali bahasa pemrograman dan bahasa *markup*. *Sublime text editor* juga mendukung penambahan *plugin*. *Sublime* dibangun dengan menggunakan *python*.

Sublime Text memiliki banyak kelebihan diantaranya:

1. *Multiple Selection*, mempunyai fungsi untuk melakukan perubahan pada sebuah kode dalam waktu yang sama dan dalam baris yang berbeda.
2. *Command Pallete*, mempunyai fungsi yang berguna untuk mengakses file *shortcut* dengan mudah, untuk mencari file tersebut dengan menekan CTRL+SHIFT+P.
3. *Distraction free mode*, fitur ini sangat dibutuhkan oleh pengguna yang sedang fokus dalam pekerjaan, yaitu dapat merubah tampilan layar menjadi penuh dengan menekan SHIFT+F11.
4. *Find in project*, kita dapat mencari dan memiih file dalam *project* dengan mudah, dengan menekan SHIFT+P.
5. *Multi platform*, *Sublime Text* sudah tersedia dalam berbagai *platform* sistem operasi seperti Windows, Linux, Mac os.

2.3 Teori Khusus

2.3.1 *Unified Modeling Language (UML)*

Menurut Mustakim et.al (2016:4), “UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.”

Menurut Nugroho (2005:16), “UML adalah notasi-notasi yang digunakan sebagai pemodelan visual untuk sistem informasi atau perangkat lunak. Pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati”.

Beberapa diagram dalam UML yang akan digunakan dalam pembuatan laporan kerja praktik ini antara lain sebagai berikut:

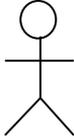
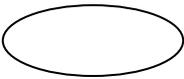
2.3.1.1 *Usecase Diagram*

Menurut Indrajani (2015:45), “*Usecase diagram* adalah suatu diagram yang berisi *usecase*, *actor* serta *relationship* yang merupakan titik awal dalam menganalisis kebutuhan sistem pada saat perancangan.”

Menurut Nugroho (2015:19), “*Usecase diagram* bersifat statis yang memperlihatkan himpunan *usecase* dan *actor* untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan dan diharapkan pengguna.”

Diagram *usecase* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*, sedangkan *usecase* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata pengguna dan aktor mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. *Usecase diagram* dapat digunakan untuk menentukan kebutuhan apa saja yang diperlukan suatu sistem, jadi dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan sesuatu. Berikut merupakan simbol-simbol diagram *usecase*:

Tabel 2.1 Simbol-simbol Diagram *Usecase*

No	Gambar	Nama	Keterangan
1.		Aktor/ Actor	Aktor menggambarkan pengguna sistem, dapat berupa manusia atau sistem <i>terotomatisasi</i> lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
2.		Use Case	Simbol ini menggambarkan interaksi antara actor dengan software aplikasi tersebut.
3.		System Boundary	Menggambarkan batasan antara sistem dengan actor.
4.		Asosiasi/ Asosiation	Menggambarkan hubungan antar actor dan <i>use case</i> .

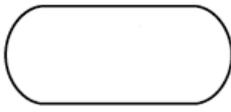
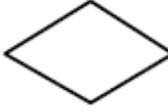
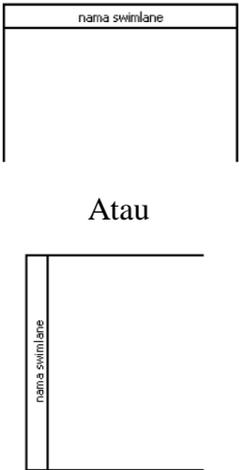
(Sumber : Indrajani 2015:46)

2.3.1.2 Activity Diagram

Menurut Nugroho (2015:61), "*Activity Diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *Usecase*. *Activity diagram* digunakan untuk memodelkan aspek dinamis dari sistem. *Activity diagram* secara esensial mirip diagram alir (*flowchat*), memperlihatkan aliran kendali dari suatu aktivitas ke aktivitas yang lainnya."

Menurut Rosa dan Salahudin (2014:161), "Diagram aktivitas (*activity diagram*) menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem." Berikut merupakan simbol-simbol yang digunakan dalam Diagram Aktivitas yang dapat dilihat pada tabel berikut:

Tabel 2.2 Simbol-simbol Diagram dengan Aktivitas

No	Simbol	Nama	Deskripsi
1.		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan / <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5.	 Atau	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Rosa dan Salahudin, 2014:161)

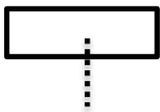
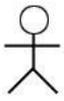
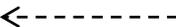
2.3.1.3 Sequence Diagram

Diagram sekuensial adalah interaction diagram yang memperlihatkan event-event yang berurutan sepanjang berjalannya waktu, dibaca dari atas ke bawah. Diagram sekuen akan menggambarkan aliran-aliran pada suatu usecase.

Indrajani (2015:50), menjelaskan tentang sequence diagram merupakan suatu diagram yang menggambarkan bagaimana objek-objek berpartisipasi dalam bagian interaksi dan pesan yang ditukar dalam urutan waktu. Kegunaannya untuk

menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem waktu atau urutan gunakan Sequence Diagram. Berikut simbol-simbolnya:

Table 2.3 Simbol-simbol Diagram Sekuensial

No	Simbol	Nama	Keterangan
1.		Lifeline	<i>Object entity</i> , yang saling berinteraksi.
2.		Aktor	Menggambarkan <i>actor</i> yang terlibat.
3.		Message (call)	Menggambarkan alur <i>message</i> yang merupakan kejadian objek pengirim <i>lifeline</i> ke objek penerima <i>lifeline</i> .
4.		Message (return)	Menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.
5.		Activation	Menggambarkan hubungan antara <i>object</i> dengan <i>message</i> .

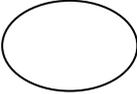
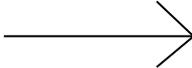
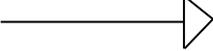
(Sumber: Indrajani 2015:51)

2.3.1.4 Class Diagram

Menurut Nugroho (2005:110), "*Class diagram* adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang dikembangkan. Dalam class diagram memberi gambaran diagram statis tentang sistem atau perangkat lunak dengan relasi-relasi yang ada didalamnya.

Menurut Indrajani (2014:49): "*Class diagram* adalah diagram yang menggambarkan perbedaan antara class-class hubungan antar class dan dinamakan sub sistem class, pada class diagram terdapat nama class, atribut dan operation. Berikut merupakan simbol Diagram Kelas:

Tabel 2.4 Simbol-simbol Diagram Kelas

No	Simbol	Nama	Keterangan			
1.	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">Class</td> </tr> <tr> <td style="text-align: center;">Attribute</td> </tr> <tr> <td style="text-align: center;">Operation</td> </tr> </table>	Class	Attribute	Operation	Kelas	Kelas pada struktur sistem
Class						
Attribute						
Operation						
2.		Antarmuka/ Interface	Sama dengan konsep interface dalam pemrograman berorientasi obyek.			
3.		Asosiasi/ Association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.			
4.		Asosiasi berarah/directed	Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multiplicity.			
5.		Spesialisasi	Asosiasi antar kelas dengan makna generalisasi spesialisasi (umum – khusus)			
6.		Kebergantungan/ dependency	Ketergantungan antarkelas.			
7.		Agregasi	Relasi antar kelas dengan makna umum semua bagian (whole-part)			

(Sumber: Rosa dan Salahudin, 2014:141)