



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Sistem

Menurut Kusri (2007:11) berpendapat bahwa sistem adalah kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*).

Menurut Nurhasanah (2017:61) mendefinisikan bahwa sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan.

Menurut Latief, dkk. (2018:1) mengungkapkan bahwa sistem merupakan kumpulan sub-sub sistem (elemen) yang saling berkorelasi satu dengan yang lainnya untuk mencapai suatu tujuan tertentu.

Berdasarkan ketiga definisi sistem diatas dapat disimpulkan bahwa sistem adalah kumpulan elemen-elemen atau bagian-bagian yang saling berinteraksi atau berhubungan untuk mencapai suatu tujuan yang telah disepakati bersama.

2.1.2 Pengertian Sistem Pendukung Keputusan

Menurut Bonczek, dkk. (dalam Nofriansyah, dkk., 2017:2) mendefinisikan bahwa sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi meliputi sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (respositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

Menurut Moore dan Chang (dalam Nofriansyah, dkk., 2017:2) berpendapat bahwa konsep struktur pada definisi awal sistem pendukung keputusan (bahwa



sistem pendukung keputusan dapat menangani situasi semistruktur dan tidak terstruktur), sebuah masalah dapat dijelaskan sebagai masalah terstruktur dan tidak terstruktur hanya dengan memperhatikan si pengambil keputusan atau suatu spesifik. Jadi mereka mendefinisikan DSS sebagai sistem yang dapat diperluas untuk mampu mendukung analisis data ad hoc dan pemodelan keputusan, berorientasi terhadap perencanaan masa depan, dan digunakan pada interval yang tidak reguler dan tak terencana.

Sedangkan menurut Litte (dalam Latief, dkk., 2018:2) mendefinisikan sistem pendukung keputusan sebagai suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur maupun tidak terstruktur dengan menggunakan data dan model.

Dari ketiga pendapat di atas dapat disimpulkan bahwa sistem pendukung keputusan merupakan suatu sistem informasi berbasis komputer yang digunakan untuk membantu seseorang dalam pengambilan keputusan mengenai masalah terstruktur, semi terstruktur maupun tidak terstruktur berdasarkan kriteria tertentu.

Menurut Latief, dkk. (2018:4) terdapat empat komponen dari sistem pendukung keputusan yaitu sebagai berikut.

1. *Data Management*

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

2. *Model Management*

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

3. *Communication (dialog subsystem)*

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



4. *Knowledge Management*

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

Menurut Simon terdapat tiga fase dalam proses pengambilan keputusan adalah sebagai berikut.

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendekteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, diproses dan 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 solusi.

3. *Choice*

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

2.1.3 Pengertian Relawan

Relawan adalah orang yang tanpa dibayar menyediakan waktunya untuk mencapai tujuan organisasi, dengan tanggung jawab yang besar atau terbatas, tanpa atau dengan sedikit latihan khusus, tetapi dapat pula dengan latihan yang sangat intensif dalam bidang tertentu, untuk bekerja sukarela membantu tenaga profesional. Dalam *kamus bahasa indonesia* relawan sepadan dengan kata sukarelawan yang artinya adalah orang yang melakukan sesuatu dengan sukarela (tidak karena diwajibkan atau dipaksakan). (serikatnews.com)

Berdasarkan pemaparan di atas, relawan adalah sekelompok orang atau seseorang yang tidak memiliki kewajiban menolong seseorang tetapi memiliki dorongan untuk membantu seseorang atau dalam kegiatan tertentu yang secara sukarela tanpa mengharapkan imbalan.



2.1.4 Pengertian Website

Menurut Raharjo (2011:2) berpendapat bahwa *World Wide Web* (WWW), sering disingkat dengan *web*, merupakan suatu layanan di dalam jaringan internet yang berupa ruang informasi. Sedangkan menurut Sibero (2013:11) mengatakan bahwa *World Wide Web* (W3) atau yang dikenal juga dengan istilah *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet.

Dari kedua definisi diatas dapat disimpulkan bahwa *Website* adalah kumpulan halaman-halaman yang dapat menampilkan teks, gambar, animasi, video, suara yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.1.5 Pengertian Metode

Dalam *Kamus Besar Bahasa Indonesia* metode adalah cara teratur yang digunakan untuk melaksanakan suatu pekerjaan agar tujuan tercapai sesuai yang dikehendaki. Metode merupakan suatu proses atau cara sistematis yang digunakan untuk mencapai tujuan tertentu dengan efisiensi, biasanya dalam urutan langkah-langkah tetap yang teratur. (kanalinfo.web.id)

Berdasarkan pemaparan diatas dapat disimpulkan bahwa metode adalah suatu cara kerja yang teratur yang digunakan untuk melaksanakan atau melakukan suatu kegiatan tertentu yang berguna untuk mencapai tujuan yang dikehendaki.

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

Metode SMART merupakan metode pengambilan keputusan yang multi atribut yang dikembangkan oleh Edward pada tahun 1977. Teknik pembuatan keputusan multi atribut 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-nilai, nilai ini dirata-rata dengan skala tertentu. Setiap atribut mempunyai bobot yang menggambarkan seberapa penting dibandingkan dengan atribut lain.



Pembobotan pada SMART menggunakan skala antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif.

Model yang digunakan dalam SMART adalah seperti berikut:

$$u(a_i) = \sum_{j=1}^k w_j \cdot u_j(a_i)$$

Keterangan:

- 1) $u(a_i)$ = nilai total alternatif
- 2) w_j = hasil dari normalisasi bobot kriteria
- 3) $u_j(a_i)$ = hasil dari penentuan nilai *utility*

Adapun langkah-langkah dalam penyelesaian metode SMART yaitu sebagai berikut.

4. Menentukan jumlah kriteria dari keputusan yang akan diambil.
5. Sistem secara *default* memberikan nilai 0-100 berdasarkan prioritas dengan melakukan normalisasi.

$$nw_j = \frac{w_j}{\sum w_j}$$

Keterangan:

- 1) nw_j adalah normalisasi bobot kriteria ke- j
- 2) w_j adalah nilai bobot kriteria ke- j
6. Menghitung nilai *utility* untuk setiap kriteria.

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



Keterangan:

- 1) $u_i(a_i)$ = nilai *utility* kriteria ke-1 untuk kriteria ke-i
 - 2) c_{\max} = nilai kriteria maksimal.
 - 3) c_{\min} = nilai kriteria minimum.
 - 4) $c_{\text{out } i}$ = nilai kriteria ke-i.
7. Menghitung nilai akhir dan melakukan perangkingan.

2.2 Teori Program

2.2.1 Basis Data (*Database*)

Menurut Enterprise (2017:1) mendefinisikan *Database* adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap *database* mempunyai perintah tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data yang ada di dalamnya.

Menurut Fathansyah (2007:1) mengatakan bahwa Basis Data sendiri terdiri dari dua kata, yaitu Basis dan Data. Basis berarti markas atau gudang, tempat bersarang atau berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek.

Dari kedua definisi diatas dapat disimpulkan bahwa basis data adalah kumpulan data yang disimpan di dalam suatu tempat tertentu secara sistematis untuk memperoleh informasi dari basis data tersebut.

2.2.2 MySQL

Menurut Enterprise (2017:3) mendefinisikan bahwa MySQL adalah *database* yang cukup terkenal karena hampir sebagian besar aplikasi berbasis *website*, seperti *WordPress* dilengkapi dengan MySQL.

Menurut Sibero (2013:97) berpendapat bahwa MySQL atau dibaca “*My Sekuel*” dengan suatu RDBMS (*Relational Database Management System*) merupakan aplikasi sistem yang menjalankan fungsi pengolahan data. MySQL sendiri pertama dikembangkan oleh MySQL AB yang kemudian diakuisi oleh *Sun Microsystem* dan terakhir dikelola oleh *Oracle Corporation*.



Dari kedua definisi diatas dapat disimpulkan bahwa MySQL merupakan salah satu sistem manajemen *database* (DBMS) berbasis SQL yang bersifat *open source* dan paling populer saat ini.

2.2.3 PHP Hypertext Preprocessor (PHP)

2.2.3.1 Pengertian PHP Hypertext Preprocessor (PHP)

Menurut Sidik (2017:4) berpendapat bahwa PHP secara umum dikenal 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*.

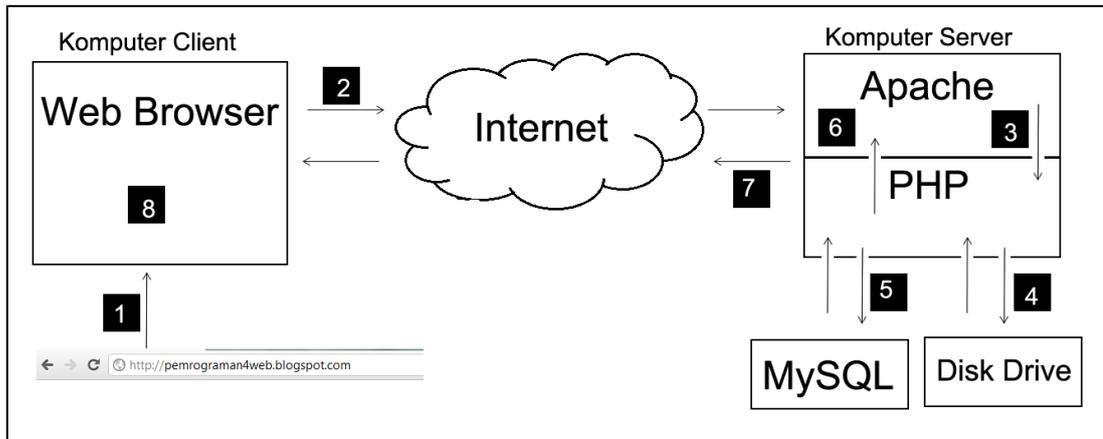
Menurut Winarno, dkk. (2014:49) mendefinisikan bahwa PHP adalah sebuah bahasa pemrograman *web* berbasis *server* (*server-side*) yang mampu memarsing kode PHP dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client* (*browser*).

Menurut Raharjo (2016:38) mengungkapkan bahwa PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*.

Dari definisi diatas dapat disimpulkan bahwa PHP *Hypertext Preprocessor* dikenal sebagai bahasa pemrograman *server side* yang membuat dokumen HTML secara *on the fly* yang dieksekusi dari *server web* yang difungsikan untuk membangun suatu *web* dinamis atau aplikasi berbasis *web*.

2.2.3.2 Cara Kerja PHP

Menurut Raharjo (2016:39) menyatakan bahwa cara kerja aplikasi *web* yang ditulis dengan PHP dapat diilustrasikan dengan gambar di bawah ini.



Gambar 2.1 Cara Kerja PHP

Berikut adalah keterangan dari gambar di atas:

1. *User* menulis `http://pemrograman4web.blogspot.com` ke dalam *address bar* dari *web browser* (IE, Mozilla Firefox, Opera, dll)
2. *Web browser* mengirimkan pesan di atas ke komputer *server* (`http://pemrograman4web.blogspot.com`) melalui internet, meminta halaman `index.php` (halaman default dari *web*).
3. *Web server* (misalnya *Apache*), program yang berjalan di komputer *server*, akan menangkap pesan tersebut, lalu meminta *interpreter* PHP (program lain yang juga berjalan di komputer *server*) untuk mencari file `index.php` dalam *disk drive*.
4. *Interpreter* PHP membaca file `index.php` dari *disk drive*.
5. *Interpreter* PHP akan menjalankan perintah-perintah atau kode PHP yang ada dalam file `index.php`. jika kode dalam file `index.php` melibatkan akses terhadap *database* (misalnya MySQL) maka *interpreter* PHP juga akan berhubungan dengan MySQL untuk melaksanakan perintah-perintah yang berkaitan dengan *database*.
6. *Interpreter* PHP mengirimkan halaman dalam bentuk HTML ke *Apache*.
7. Melalui internet, *Apache* mengirimkan halaman yang diperoleh dari *interpreter* PHP ke komputer user sebagai respon atas permintaan yang diberikan.



8. *Web browser* dalam komputer *user* akan menampilkan halaman dikirm oleh *Apache*.

2.2.4 Laragon

Laragon merupakan universal development environment bagi PHP, Java, Ruby, Node.js, Go dan Python yang dapat digunakan secara portable, terisolasi, cepat, mudah digunakan dan cukup ringan. Aplikasi ini sendiri dapat digunakan sebagai pengganti XAMPP. (logique.co.id)

Aplikasi ini mempunyai beberapa keunggulan jika dibandingkan menggunakan XAMPP. Beberapa keunggulan tersebut antara lain yaitu:

1. *Pretty URLs*

Jika Anda menggunakan Laragon, maka Anda dapat mengakses *project* melalui *app.test* saja tanpa harus menggunakan *localhost/app*.

2. *Portable*

Salah satu keunggulan lainnya jika dibandingkan XAMPP adalah foldernya yang mampu dipindahkan tanpa perlu merusak sistem.

3. *Isolated*

Sistem Laragon terisolasi langsung dari sistem operasi yang Anda miliki sehingga apapun yang Anda lakukan terhadap aplikasi ini tidak akan mempengaruhi komputer lokal Anda.

4. *Easy Operation*

Aplikasi ini juga telah mengotomatiskan banyak konfigurasi yang kompleks sehingga dapat Anda gunakan dengan sangat mudah. Anda juga dapat menambahkan versi terbaru dari PHP, Java, Go, Apache, Python, Ruby, MySQL, MongoDB, Nginx, PostgreSQL dan lainnya tanpa kesulitan sama sekali serta tidak akan merusak sistem yang telah ada.

5. *Modern & Powerfull*

Keuntungan lain dari aplikasi ini, yaitu mempunyai arsitektur modern yang cocok untuk digunakan dalam mengembangkan aplikasi *web modern*, dimana Anda dapat menggunakan Apache & Nginx sekaligus.



2.2.5 *Sublime Text*

Menurut Faridi (2015:3) mendefinisikan bahwa *Sublime Text* adalah *text editor* berbasis *Python*, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di kalangan *developer* (pengembang), penulis dan *desainer*. Para *programmer* biasanya menggunakan *sublime text* untuk menyunting *source code* yang sedang ia kerjakan.



Gambar 2.2 Logo *Sublime Text*

Menurut Faridi (2015:3-6) terdapat beberapa fitur yang diunggulkan dari aplikasi *Sublime Text*:

a. *Multiple Selection*

Fitur ini berfungsi untuk membuat perubahan pada sebuah kode pada waktu yang sama dan dalam baris yang berbeda.

b. *Command Pallete*

Fitur ini berguna untuk mengakses file *shortcut* dengan mudah.

c. *Distraction Free Mode*

Bila user memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu user dengan memberikan tampilan layar penuh dengan menekan Shift + F11.

d. *Find In Project*

Dengan fitur ini kita dapat mencari dan membuka file di dalam sebuah *project* dengan cepat dan mudah. Hanya dengan menekan Ctrl+P anda dapat mencari file yang diinginkan.



e. *Plugin API Switch*

Sublime text mempunyai keunggulan dengan *plugin* yang berbasis *python plugin API*. Teks editor ini juga mempunyai *plugin* yang sangat beragam, dan ini dapat memudahkan pengguna dalam mengembangkan *software* nya.

f. *Drag and Drop*

Dalam teks editor ini pengguna dapat menyeret dan melepas file teks ke dalam editor yang akan membuka tab baru secara otomatis.

g. *Split Editing*

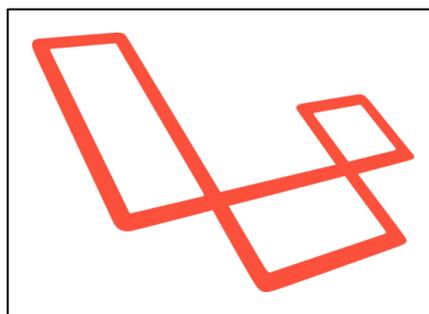
Di dalam fitur ini pengguna dapat mengedit file secara berdampingan dengan klik File->New menu into file.

h. *Multiple Platform*

Sublime text sudah tersedia dalam berbagai *platform* sistem operasi seperti *Windows, Linux, dan MacOS*.

2.2.6 Laravel

Menurut Chen et al (2017) Laravel adalah *web framework* PHP yang bersifat *open source* dan gratis yang dibuat oleh Taylor Otweel yang dapat digunakan dalam mengembangkan *web applications* dengan menggunakan arsitektur MVC (*Model-View-Controller*). *Framework* Laravel mudah dipahami dan memudahkan dalam hal *authentication, routing, session manager, caching*, dan beberapa kegunaan lain dari komponen – komponen di Laravel. Laravel juga menyediakan fitur seperti *database migration* dan integrasi *unit testing support* yang memudahkan *developer* untuk membangun aplikasi yang kompleks.



Gambar 2.3 Logo Laravel



Menurut Mediana, dkk. (2018:76-77) terdapat enam dasar dalam Laravel:

1. *Artisan*

Artisan adalah *command line* atau perintah yang dijalankan melalui terminal dan disediakan beberapa perintah perintah yang dapat digunakan selama melakukan pengembangan dan pembuatan aplikasi. Salah satu fungsi dari php artisan yaitu “php artisan serve”. Php artisan serve berfungsi untuk membuka *website* yang telah dibuat tanpa menggunakan web server lokal.

2. *Routing*

Routing adalah suatu proses yang bertujuan agar suatu item yang diinginkan dapat sampai ke tujuan. Dengan menggunakan *routing* dapat ditentukan halaman halaman yang akan muncul ketika dibuka oleh *user*. Pengaturan *routing* di laravel biasanya terletak di file *web.php*. File *web.php* terletak di dalam folder *routes*.

3. *Controller*

Controller adalah suatu proses yang bertujuan untuk mengambil permintaan, menginisialisasi, memanggil model untuk dikirimkan ke *view*. Ada dua cara membuat *controller* di laravel. Cara pertama adalah dibuat file *controller* secara manual dan dituliskan *code extends controller* di dalamnya. Cara kedua adalah dibuat file *controller* menggunakan *command line* dengan menuliskan “php artisan make controller nama_file_controller”. Permintaan yang dibuat dalam laravel harus berada di dalam *controller*, kemudian dilempar melalui *routing* untuk mendapat permintaan yang diinginkan.

4. *View (blade templating)*

Blade adalah *template engine* bawaan dari laravel. *Blade* memiliki kode kode yang lebih mudah untuk menghasilkan laravel. Cara membuat file.blade dilakukan secara manual dengan membuat nama_file.php.blade di dalam folder *views*. Di dalam *blade* dapat dibuat *template* master dan *template inheritance*. Pembuatan *template* master dan turunannya ini bertujuan agar elemen yang sama tidak ditulis secara berulang-ulang. Pada *template inheritance* diberikan kode “*extend (nama_layout) dan section (nama_content)*”.



5. *Middleware*

Middleware adalah penengah Antara request yang masuk dengan *controller* yang dituju. Cara membuat *middleware* menggunakan artisan dengan mengetikkan “php artisan make:middleware nama_file”. File *middleware* berada di dalam folder *middleware*.

6. *Session*

Session adalah sebuah cara yang digunakan untuk penyimpanan pada *server* dan penyimpanan tersebut digunakan pada beberapa halaman termasuk halaman itu sendiri. Dalam menggunakan *session* ada dua cara. Cara yang pertama *session* dapat dibuat menggunakan *Request*. Cara yang kedua dapat digunakan fungsi *global helper session*.

2.3 Teori Khusus

2.3.1 *Unified Modeling Language (UML)*

Menurut Sukamto dan Shalahuddin (2018:133) menyatakan bahwa UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Menurut Pender (2002) mengungkapkan bahwa UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasi, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*).”

Beberapa diagram dalam UML yang akan digunakan dalam pembuatan tugas akhir ini antara lain sebagai berikut:

2.3.1.1 *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2018:124-125) menyatakan bahwa *use case* atau diagram *use case* merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam

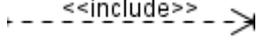


sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut merupakan simbol-simbol diagram *use case*:

Tabel 2.1 Simbol-simbol Diagram *Use case*

No	Gambar	Nama	Keterangan
1.		Aktor/ <i>Actor</i>	Aktor menggambarkan pengguna sistem, dapat berupa manusia atau sistem terotomatisasi lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
2.		<i>Use Case</i>	Simbol ini menggambarkan interaksi antara <i>actor</i> dengan <i>software</i> aplikasi tersebut.
3.		Asosiasi/ <i>Asosiation</i>	Menggambarkan hubungan antar aktor dan <i>use case</i> .
4.		Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.		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 walau tanpa <i>use case</i> tambahan itu.

Lanjutan **Tabel 2.1** Simbol-simbol Diagram *Use case*

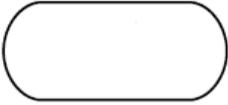
No	Gambar	Nama	Keterangan
6.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

(Sumber : Sukamto dan Salahudin, 2018:156-158)

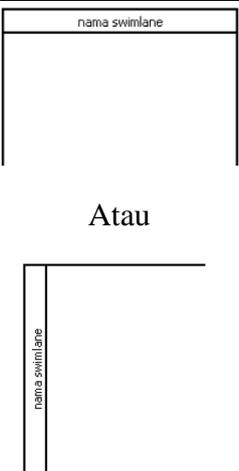
2.3.1.2 Activity Diagram

Menurut Sukamto dan Shalahuddin (2018:161) menyatakan bahwa 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 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

Lanjutan **Tabel 2.2** Simbol-simbol Diagram dengan Aktivitas

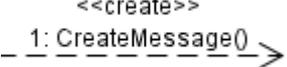
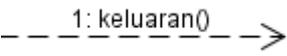
No	Simbol	Nama	Deskripsi
4.		Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	 <p style="text-align: center;">Atau</p>	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Sukamto dan Salahudin, 2018:162-163)

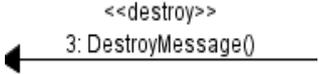
2.3.1.3 *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2018:165) menyatakan bahwa sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut simbol-simbolnya:

Table 2.3 Simbol-simbol Diagram Sekuensial

No	Simbol	Nama	Keterangan
1.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
2.		Aktor	Menggambarkan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
5.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
6.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
7.		Pesan tipe <i>return</i>	Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

Lanjutan **Table 2.3** Simbol-simbol Diagram Sekuensial

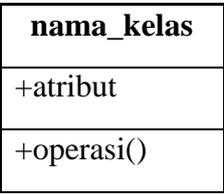
No	Simbol	Nama	Keterangan
8.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
9.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber: Sukamto dan Salahudin, 2018:165-167)

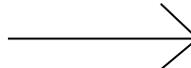
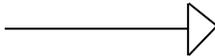
2.3.1.4 Class Diagram

Menurut Sukamto dan Shalahuddin (2018:141) menyatakan bahwa diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut dengan atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan metode atau operasi merupakan fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.4 Simbol-simbol Diagram Kelas

No	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem
2.	 nama_interface	Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi obyek.

Lanjutan **Tabel 2.4** Simbol-simbol Diagram Kelas

No	Simbol	Nama	Keterangan
3.		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		Asosiasi berarah/ <i>directed</i> <i>Association</i>	Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.		Generalisasi	Relasi antar-kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.		Kebergantungan/ <i>dependency</i>	Ketergantungan antarkelas.
7.		Agregasi	Relasi antar kelas dengan makna umum semua bagian (<i>whole-part</i>)

(Sumber: Sukamto dan Salahudin, 2018:146-147)

