



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Sitem Pendukung Keputusan

2.1.1.1 Pengertian Sistem Pendukung Keputusan

Menurut Bonczek, dkk. di dalam buku *Sistem Pendukung Keputusan Teori dan Implementasi* (Latif, dkk., 2018:2) 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).

Menurut Utomo, *et al* (2015), “Sistem Pendukung Keputusan (SPK) adalah suatu sistem informasi yang menggunakan model-model keputusan, basis data, dan pemikiran manajer sendiri, proses *modelling* interaktif dengan komputer untuk mencapai pengambilan keputusan oleh manajer tertentu. Dengan adanya SPK dapat memberikan perangkat interaktif yang memungkinkan *decission maker* melakukan berbagai analisis dari model yang tersedia.”

Dari pendapat diatas disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem informasi yang berbasis komputer yang digunakan untuk membantu seseorang dalam pembuatan keputusan untuk memecahkan permasalahan dari berbagai alternatif yang ada.

2.1.1.2 Komponen Sistem Pendukung Keputusan

Menurut Latif, 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 Systems* (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 *Decision Support System* (DSS) melalui subsistem ini. Subsistem ini berarti menyediakan antarmuka.

4. *Knowledge Management*

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

2.1.1.3 Tahapan Proses Pengambilan Keputusan

Nofriansyah, Dicky (2014:2), menurut Simon ada tiga fase dalam proses pengambilan keputusan diantaranya 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 diimplementasikan dalam proses pengambilan keputusan.



2.1.2 Pengertian Mustahiq Zakat

Mustahiq zakat adalah orang-orang yang berhak menerima zakat, ada delapan asnaf (golongan) yaitu: 1) Orang fakir, yaitu orang yang tidak ada harta untuk keperluan hidup sehari-hari dan tidak mampu bekerja atau berusaha. 2) Orang miskin, yaitu orang yang berpenghasilan sehari-harinya tidak mencukupi kebutuhan hidupnya. 3) Amil zakat, yaitu lembaga atau perorangan yang mengelola zakat. 4) Muallaf, yaitu orang yang baru masuk islam. 5) Riqab, yaitu untuk memerdekakan hamba sahaya. 6) Gharimin, yaitu untuk membebaskan beban orang yang berutang untuk kepentingan kebaikan. 7) Sabilillah, yaitu untuk kepentingan di jalan Allah dan juga untuk keperluan pertahanan islam dan kaum muslimin, fisabilillah juga mencakup kepentingan-kepentingan umum seperti mendirikan sekolah, rumah sakit dan lain-lain. 8) Ibnu Sabil, yaitu orang yang dalam perjalanan yang kehabisan bekal dan perjalanan tersebut untuk tujuan kebaikan, seperti mahasiswa atau santri yang menuntut ilmu di luar kota.

2.1.3 Metode *Profile Matching*

Metode *Profile Matching* merupakan salah satu metode yang sederhana dalam sistem pendukung keputusan dengan membandingkan Gap antara nilai alternatif dan kriteria. Ada beberapa hal yang diketahui tentang analisis *Gap*, salah satu diantaranya adalah tabel nilai bobot Gap. Selain itu, analisis Gap ini juga harus memiliki konsep skala prioritas, karena di dalam pembuatan bobot dengan range 0-5 berdasarkan prioritas setiap kriteria (Latif, Lita Asyriatif, dkk., 2018:38).

Profile Matching adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat minimal yang harus dipenuhi atau dilewati. Langkah-langkah dalam penyelesaian perhitungan dengan menggunakan metode *Profile Matching*. (Juanita, 2015).

1. Aspek Penilaian

Langkah pertama yang harus dilakukan yaitu menentukan aspek-aspek penilaian pada *core factor* (faktor utama) dan *secondary factor* (faktor kedua).



2. Pemetaan GAP Kompetensi

GAP kompetensi adalah perbedaan antara kriteria yang dimiliki seseorang dengan kriteria yang diinginkan. Rumus GAP kompetensi yaitu.

$$GAP = \text{Nilai Kriteria} - \text{Nilai Minimal}$$

3. Pembobotan

Apabila pemetaan GAP sudah selesai dilakukan, maka hasil dari pemetaan tersebut diberi bobot nilai sesuai dengan patokan tabel bobot nilai GAP, seperti yang terlihat pada tabel.

Tabel 2.1 Nilai Pembobotan GAP

No	Selisih	Bobot Nilai	Keterangan
1	0	5	Kompetensi sesuai dengan yang dibutuhkan
2	1	4.5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kurang 1 tingkat/level
4	2	3.5	Kompetensi individu kelebihan 2 tingkat/level
5	-2	3	Kompetensi individu kurang 2 tingkat/level
6	3	2.5	Kompetensi individu kelebihan 3 tingkat/level
7	-3	2	Kompetensi individu kurang 3 tingkat/level
8	4	1.5	Kompetensi individu kelebihan 4 tingkat/level
9	-4	1	Kompetensi individu kurang 4 tingkat/level

Tabel 2.1 berisi ketentuan pembobotan hasil selisih nilai GAP yang dikurangkan dengan nilai minimal yang sudah ditetapkan.



4. Perhitungan dan pengelompokan *Core Factor* dan *Secondary Factor*.

Setelah bobot nilai GAP ditentukan, maka dibagi menjadi 2 kelompok yaitu *Core Factor* (faktor utama) dan *Secondary Factor* (faktor pendukung). Rumus untuk menghitung *core factor*, sebagai berikut.

$$NCF = \frac{\sum NC(asppek)}{\sum IC}$$

Keterangan :

NCF = Nilai rata-rata *core factor*

NC (aspek) = Jumlah total nilai *core factor*

IC = Jumlah item *core factor*

Rumus untuk menghitung *secondary factor*, sebagai berikut.

$$NSF = \frac{\sum NS(asppek)}{IS}$$

Keterangan :

NSF = Nilai rata-rata *secondary factor*

NS (aspek) = Jumlah total nilai *secondary factor*

IS = Jumlah item *secondary factor*

5. Perhitungan Nilai Total

Untuk menghitung nilai total, rumus yang digunakan yaitu.

$$N(asppek) = (X)\% NCF(asppek) + (X)\% NSF(asppek)$$

Keterangan :

NCF (aspek) = nilai rata-rata *core factor*.

NSF (aspek) = nilai rata-rata *secondary factor*.

N(aspek) = nilai total dari aspek



(X)% = nilai persen yang diinputkan

6. Perhitungan Nilai Rangking

Untuk menentukan perangkingan mengacu pada hasil perhitungan menggunakan rumus sebagai berikut.

$$\text{Rangking} = (X)\% NS$$

Keterangan :

NS = Nilai aspek

(X)% = Nilai persen yang diinputkan

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 Budi Raharjo (2015:2) dalam buku yang berjudul *Belajar Otodidak Mysql* menjelaskan bahwa database adalah sekumpulan data yang terintegrasi dan diatur sehingga data tersebut dapat diambil dan dicari secara cepat.

Dari definisi di atas dapat disimpulkan bahwa basis data adalah sekumpulan data yang saling berhubungan dan berkaitan yang disimpan pada sebuah media tertentu untuk memperoleh informasi.

2.2.2 MySQL

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

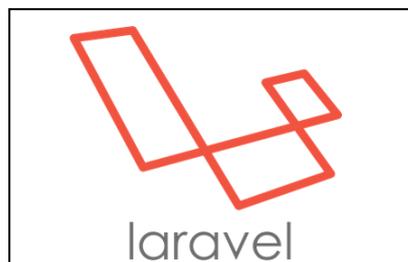
Menurut Budi Raharjo (2015:16) dalam buku yang berjudul *Belajar Otodidak Mysql* menjelaskan bahwa Mysql adalah merupakan software database



yang sangat cepat dalam mengelola, menampung data dalam ukuran besar sehingga dapat diakses oleh banyak pengguna.

Dari pengertian di atas dapat disimpulkan bahwa MySQL merupakan sebuah aplikasi atau yang digunakan untuk menyimpan dan mengolah data dalam bentuk tabel-tabel yang saling berhubungan.

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 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*.

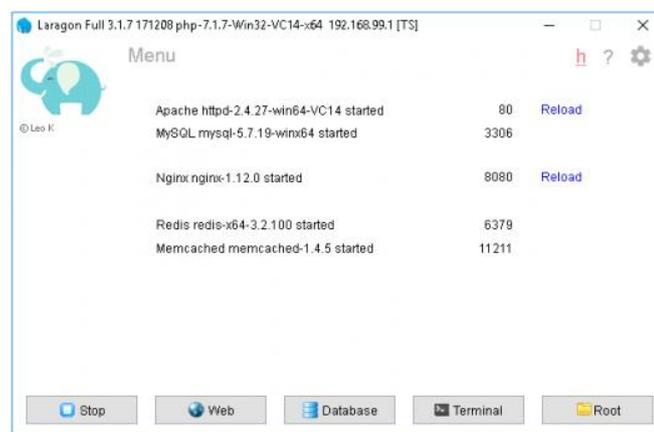
2.2.4 PHP Hypertext Preprocessor (PHP)

Menurut Saputra dan Agustin (2013:2) dalam buku yang judul *Menyelesaikan Website 12 Juta Secara Profesional* menjelaskan bahwa PHP (Hypertext Preprocessor) merupakan bahasa pemrograman berbasis server-side scripting yang dapat menyatu dengan HTML sehingga dapat membangun sebuah website dinamis.

Menurut Setiawan (2017), PHP merupakan kependekan dari Hypertext Preprocessor yang merupakan sebuah bahasa script tingkat tinggi yang dipasang pada dokumen HTML. Sebagian besar sintaks PHP mirip dengan bahasa C, java, dan perl. Namun, PHP terdapat beberapa fungsi yang lebih spesifik. PHP digunakan untuk merancang yang sifatnya dinamis dan dapat bekerja secara otomatis.

Dari definisi di atas dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang digunakan untuk membuat *website* yang bersifat dinamis.

2.2.5 Laragon





Gambar 2.1 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, Cmsder*, dan lainnya. Aplikasi ini sangat cocok digunakan oleh seorang *developer PHP* yang menggunakan *framework* Laravel, karena akan mempermudah dalam melakukan pengembangan aplikasi.

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.



2.2.6 Sublime Text



Gambar 2.3 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.

Berikut beberapa fitur yang diunggulkan dari aplikasi *Sublime Text*:

a. *Goto Anything*

Fitur yang sangat membantu dalam membuka file ataupun menjelajahi isi dari file hanya dengan beberapa keystrokes.

b. *Multiple Selections*



Fitur ini memungkinkan user untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya.

c. *Command Pallete*

Dengan hanya beberapa keystrokes, user dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

d. *Distraction Free Mode*

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

e. *Split Editing*

Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan editing perpecahan. Mengedit sisi file dengan sisi, atau mengedit dua lokasi di satu file. Anda dapat mengedit dengan banyak baris dan kolom yang user inginkan.

f. *Instant Project Switch*

Menangkap semua file yang dimasukkan ke dalam project pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam *project* lainnya dengan cepat.

g. *Plugin API*

Dilengkapi dengan plugin API berbasis Python sehingga membuat aplikasi ini sangat tangguh.

h. *Customize Anything*

Aplikasi ini memberikan user fleksibilitas dalam hal pengaturan fungsional.

i. *Cross Platform*

Aplikasi ini dapat berjalan hampir di semua *operating system modern* seperti *Windows*, *OS X*, dan *Linux based operating system*.



2.3 Teori Khusus

2.3.1 *Unified Modeling Language (UML)*

Menurut Sukamto dan Shalahuddin dalam buku *Rekayasa Perangkat Lunak* (2018:133) 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 Windu Gata (2013), *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

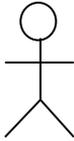
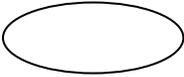
Dalam membangun perancangan sistem dengan alat bantu perancangan *Unified Modeling Language (UML)* ada beberapa tahapan yang akan dilakukan, yaitu sebagai berikut:

2.3.2 *Use Case Diagram*

Menurut Rosa dan Shalahuddin (2018:155), *use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.2 Simbol *Use Case Diagram*

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 anatara sistem dengan actor.
4.		Asosiasi/ Asosiation	Menggambarkan hubungan antar actor dan <i>use case</i> .

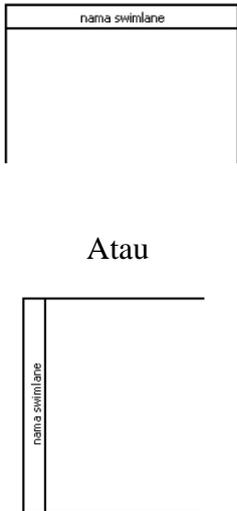
(Sumber : Indrajani, 2015:46)

2.3.3 Activity Diagram (Diagram Aktivitas)

Menurut Rosa dan Shalahuddin (2018:161), *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram aktivitas:



Tabel 2.3 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
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>Atau</p>	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Rosa dan Shalahuddin, 2018:162)



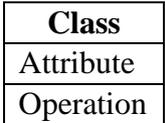
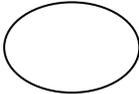
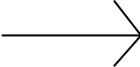
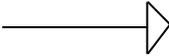
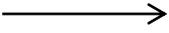
2.3.4 Class Diagram (Diagram Kelas)

Menurut Rosa dan Shalahuddin (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan operasi:

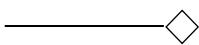
1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.4 Simbol Diagram Kelas

No	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem
2.		Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi obyek
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>	Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertain dengan <i>multiplicity</i>
5.		Generalisasi	Asosiasi antar kelas dengan makna generelisasi spesialisasi (umum – khusus)
6.		Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antarkelas



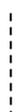
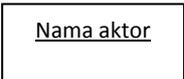
7.		Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)
----	---	----------------------------------	--

(Sumber : Rosa dan Shalahuddin, 2018:146)

2.3.5 *Sequence Diagram* (Diagram Sekuen)

Menurut Rosa dan Shalahuddin (2018:165), *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek tersebut. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel 2.5 Simbol Diagram Sekuen

No	Simbol	Nama	Keterangan
1.		Garis hidup / <i>lifeline</i>	Menyatakan kehidupan suatu objek
2.	 Atau 	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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 di awal frase nama aktor
3.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya

