

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem Pendukung Keputusan**

Sistem Pendukung Keputusan (SPK) merupakan aplikasi interaktif berbasis komputer yang mengkombinasikan data dan model matematis untuk membantu proses pengambilan keputusan dalam menangani suatu masalah (Vercellis, 2009:36) dalam (Taufiq & Sari, 2019).

Sistem Pendukung Keputusan adalah sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan (Turban, 2005:137) dalam (Andika *et al.*, 2019). Ada beberapa karakteristik dari sistem pendukung keputusan adalah sebagai berikut:

1. Sistem pendukung keputusan memberikan dukungan bagi pengambil keputusan pada situasi semi terstruktur dan tak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi.
2. Sistem pendukung keputusan memberikan dukungan bagi pengambil keputusan pada situasi semi terstruktur dan tak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi.
3. Dukungan di semua fase proses pengambilan keputusan, yaitu intelligence, design, choice dan implementation.
4. Peningkatan efektivitas dari pengambilan keputusan daripada efisiensi.
5. Pengguna akhir bisa mengembangkan dan memodifikasi sendiri sistem sederhana.
6. Dapat digunakan sebagai standalone oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di suatu organisasi secara keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.

##### **2.1.1 Karakteristik Sistem Pendukung Keputusan**

Menurut (Saliman, 2011:8) dalam (Sofiah & Septiana, 2017), karakteristik dari Sistem Pendukung Keputusan (*Decision Support System*) diantaranya:

1. Sistem pendukung keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang bersifat semi terstruktur ataupun tidak terstruktur;

2. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan modelmodel / teknik-teknik analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari / interogasi informasi;
3. Sistem pendukung keputusan dirancang sedemikian rupa, sehingga dapat digunakan dengan mudah oleh orang yang tidak memiliki dasar kemampuan pengoperasian komputer yang tinggi;
4. Sistem pendukung keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi, sehingga mudah disesuaikan dengan kebutuhan pemakai.

## **2.2 Logika Fuzzy**

Logika fuzzy merupakan generalisasi dari logika klasik yang hanya memiliki dua nilai keanggotaan antar 0 dan 1. Logika fuzzy adalah salah satu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output (Kusumadewi & Purnomo, 2010). Dengan teori himpunan fuzzy, suatu objek dapat menjadi anggota dari banyak himpunan dengan derajat keanggotaan yang berbeda dalam masing-masing himpunan. Dengan memperluas konsep fungsi karakteristik itu, Zadeh mendefinisikan Jurnal Teknologi Informasi Vol. 6 No. 2 himpunan kabur dengan menggunakan apa yang disebutnya fungsi keanggotaan (membership function), yang nilainya berada dalam selang tertutup  $[0,1]$ . Jadi keanggotaan dalam himpunan kabur tidak lagi merupakan sesuatu yang tegas atau crisp (yaitu anggota atau bukan anggota), melainkan sesuatu yang berderajat atau bergradasi secara berkelanjutan.

### **2.2.1 Himpunan Fuzzy**

Prinsip dasar dan persamaan matematika dari teori himpunan fuzzy adalah sebuah teori pengelompokan objek dalam batas yang samar. Himpunan tersebut dikaitkan dengan suatu fungsi yang menyatakan derajat kesesuaian unsur-unsur dalam semestanya dengan konsep yang merupakan syarat keanggotaan himpunan tersebut. Fungsi itu disebut fungsi keanggotaan dan nilai fungsi itu disebut derajat keanggotaan suatu unsur dalam himpunan itu, yang selanjutnya disebut himpunan kabur (fuzzy set). Dengan demikian setiap unsur dalam semesta wacananya mempunyai derajat keanggotaan tertentu dalam himpunan tersebut. Derajat keanggotaan dinyatakan dengan suatu bilangan real dalam selang tertutup  $[0, 1]$ . Nilai fungsi sama dengan satu menyatakan keanggotaan penuh, dan nilai fungsi sama dengan nol menyatakan sama sekali bukan anggota himpunan kabur tersebut. Maka himpunan tegas (crisp) juga dapat dipandang sebagai kejadian khusus dari himpunan kabur,

yaitu himpunan kabur yang fungsi keanggotaannya hanya bernilai satu atau nol saja. Himpunan fuzzy memiliki dua atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MAHAL, SEDANG, MURAH dan sebagainya.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti 100 juta, 200 juta, 500 juta dan lain sebagainya (Umar & Widarti, 2015).

### 2.2.2 Fungsi Keanggotaan

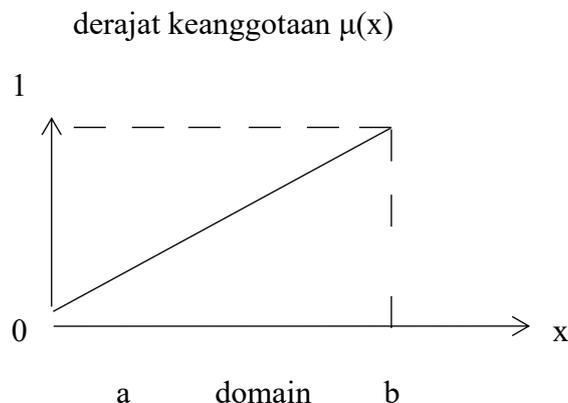
Menurut (Saleh, 2016) fungsi keanggotaan merupakan suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval 0 sampai 1.

#### a) Fungsi Himpunan *Fuzzy*

Fungsi Keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam keanggotaannya yang memiliki interval antara 0 sampai 1. Salah satu cara untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi (Kusumadewi, 2002: 8). Ada 2 keadaan himpunan fuzzy yang linear, yaitu :

#### 1. Representasi Linear Naik

Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.



Gambar 2.1 Representasi Linear Naik

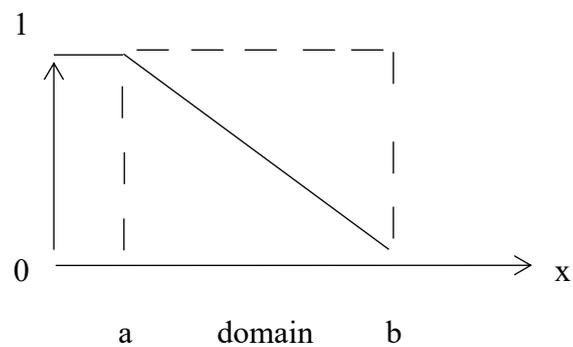
Rumus fungsi keanggotaan linear naik dinyatakan dengan :

$$\mu(x) = \begin{cases} 0; x < a \\ \frac{x-a}{b-a}; a \leq x \leq b \\ 1; x > b \end{cases}$$

## 2. Representasi Linear Turun

Penurunan himpunan dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.

derajat keanggotaan  $\mu(x)$



Gambar 2.2 Representasi Linear Turun

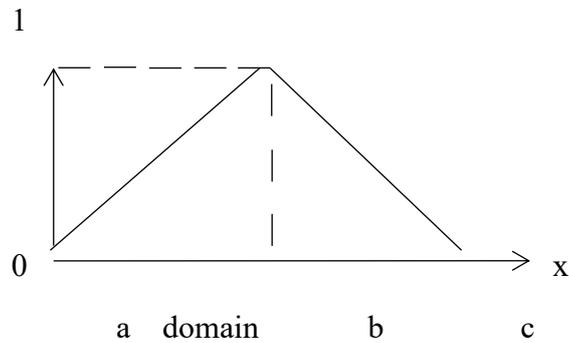
Rumus fungsi keanggotaan linear turun dinyatakan dengan :

$$\mu(x) = \begin{cases} 1; x < a \\ \frac{b-x}{b-a}; a \leq x \leq b \\ 0; x \geq b \end{cases}$$

### b) Representasi Kurva Segitiga

Fungsi keanggotaan fuzzy ini merupakan gabungan dari fungsi keanggotaan linear naik dan fungsi keanggotaan linear turun.

derajat keanggotaan  $\mu(x)$



Gambar 2.3 Representasi Kurva Segitiga

Rumus fungsi keanggotaan segitiga dinyatakan dengan :

$$\mu(x) = \begin{cases} 0; x < a \text{ atau } x > c \\ \frac{b-x}{b-a}; a \leq x \leq b \\ \frac{c-x}{c-b}; b \leq x \leq c \end{cases}$$

### c) Representasi Kurva Trapesium

Fungsi keanggotaan yang merepresentasikan kurva trapesium adalah :

$$\mu(x) = \begin{cases} 0; x < a \text{ atau } x \geq d \\ \frac{x-a}{b-a}; a \leq x \leq b \\ 1; b \leq x \leq c \\ \frac{d-x}{d-c}; x \geq d \end{cases}$$

### d) Representasi Kurva Bahu

Kurva bahu terdiri dari bahu kanan dan bahu kiri. Kurva bahu kiri merepresentasikan kondisi konstan dari kiri dengan nilai keanggotaan 1 kemudian turun dengan nilai

keanggotaan menuju ke 0. Sedangkan kurva bahu kanan merepresentasikan keadaan yang linier naik menuju nilai keanggotaan 1 secara konstan kekanan.

Fungsi keanggotaan yang merepresentasikan kurva bahu kiri :

$$\mu(x) = \begin{cases} 1; x \leq a \\ \frac{(b-x)}{(b-a)}; a \leq x \leq b \\ 0; x \geq b \end{cases}$$

Fungsi keanggotaan yang merepresentasikan kurva bahu kanan :

$$\mu(x) = \begin{cases} 0; x \leq b \\ \frac{(x-b)}{(c-b)}; b \leq x \leq c \\ 1; x \geq c \end{cases}$$

#### e) Representasi Kurva-S

Kurva-S atau sigmoid berhubungan dengan kenaikan dan penurunan permukaan secara tak linear. Kurva-S yang menunjuk pertumbuhan adalah kurva S-MF. Fungsi keanggotaan S-MF adalah :

$$\mu(x) = \begin{cases} 0; x \leq a \\ 2 \left( \frac{x-a}{y-a} \right)^2; a \leq x \leq \frac{a+y}{2} \\ 1 - 2 \left( \frac{y-x}{y-a} \right)^2; \frac{a+y}{2} \leq x \leq y \\ 1; x \geq y \end{cases}$$

Adapun kurva penyusutan merupakan kurva-S yang berhubungan dengan penurunan adalah kurva Z-MF dengan fungsi keanggotaan :

$$\mu(x) = \begin{cases} 1; x \leq a \\ 1 - 2 \left( \frac{x-a}{y-a} \right)^2; a \leq x \leq \frac{a+y}{2} \\ 2 \left( \frac{x-y}{y-a} \right)^2; \frac{a+y}{2} \leq x \leq y \\ 0; x \geq y \end{cases}$$

### 2.2.3 Sistem Inferensi Fuzzy Metode Sugeno

Menurut (Lodewyik Rahakbauw Dorteus, 2015), penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output (konsekuen) sistem tidak berupa himpunan Fuzzy, melainkan berupa konstanta atau persamaan linear. Michio Sugeno mengusulkan penggunaan singleton sebagai fungsi keanggotaan dari konsekuen. Singleton adalah sebuah himpunan Fuzzy dengan fungsi keanggotaan yang pada titik tertentu mempunyai sebuah nilai dan 0 di luar titik tersebut. Ada 2 aturan dasar *fuzzy* metode Sugeno yaitu sebagai berikut:

- a. Model fuzzy sugeno orde nol Secara umum bentuk fuzzy sugeno orde nol adalah :

$$\text{IF } (x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots (x_N \text{ is } A_N) \text{ THEN } z = k$$

Dengan  $A_i$  adalah himpunan fuzzy ke  $-i$  sebagai anteseden dan  $k$  adalah konstanta tegas sebagai konsekuen.

- b. Model *fuzzy* sugeno orde satu secara umum bentuk fuzzy sugeno orde satu adalah :

$$\text{IF } (x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ \dots (x_N \text{ is } A_N) \text{ THEN } z = P_1 * x_1 + \dots + P_N * x_n + q$$

Dengan  $A_i$  adalah himpunan fuzzy ke  $-i$  sebagai anteseden,  $p_i$  konstanta tegas ke- $i$  dan  $q$  konstanta pada konsekuen.

Selanjutnya, tahapan yang dapat dilakukan untuk melakukan implementasi *Fuzzy* Sugeno adalah sebagai berikut:

#### 1. Tahap fuzzifikasi

Fuzzifikasi merupakan proses mentransformasikan data pengamatan kedalam bentuk himpunan fuzzy.

#### 2. Pembentukan Fungsi Impikasi

Fungsi implikasi merupakan tahapan yang berfungsi demi memperoleh hasil keluaran berdasarkan IF THEN rule base. Fungsi implikasi dapat berupa Min bila dalam kasus yang ditemui yaitu menggunakan operator AND dan berupa max bila menggunakan operator OR.

#### 3. Komposisi aturan (*Rule Base Evaluation*)

Apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan kombinasi antar aturan yang nantinya digunakan dalam

proses inferensi. Pada tahap ini, kombinasi aturan didapatkan dari nilai fuzzyfikasi yang didapatkan.

#### 4. Penegasan (defuzzifikasi)

Pada proses ini output berupa bilangan crisp. Defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya yaitu :

$$z = \frac{\sum_{r=1}^R \alpha_r z_r}{\sum_{r=1}^R \alpha_r}$$

### 2.3 Bisnis Startup

Bisnis startup adalah sebuah perusahaan rintisan yang dirancang untuk menemukan model bisnis yang tepat untuk perusahaannya agar dapat bertahan ditengan ketidakpastian yang ekstrem (Jaya *et al.*, 2017). *Bisnis Startup* merupakan sebuah perusahaan rintisan yang didesain untuk berkembang dengan cepat. Perlu diingat bila *startup* tidak harus bergerak di bidang teknologi, memperoleh dana dari investor, dan melakukan exit. Hal yang paling penting adalah berkembang dengan cepat. Semua hal yang berkaitan dengan *startup* akan mengikuti perkembangan perusahaan tersebut. Menurut Rei Inamoto dalam (Milivojević & Manić, 2019), dalam *startup*, terdapat 3 talent yang dibutuhkan yaitu Hipster, Hacker, Hustler.

#### 2.3.1 Talent Hipster

Talent Hipster adalah desainer yang memiliki cita rasa dalam membuat tampilan dan user interface di web atau aplikasi menjadi menarik. Hipster menetapkan praktik yang terbaik dalam marketing, membangun identitas brand, pengalaman konsumen, serta melihat dan merasakan produk. Talent hipster juga berperan sebagai copywriter. Seorang hipster berperan dalam estetika sehingga seorang hipster memiliki jiwa kreatifitas, memiliki selera yang baik dan people-oriented.

#### 2.3.2 Talent Hacker

Talent Hacker merupakan seorang developer yang mempunyai keterampilan utama dalam membangun startup bagian coding. Hacker memanfaatkan teknologi untuk membuat produk, membangun algoritma, kekayaan intelektual, dan menghasilkan teknologi yang membangun building blocks untuk bisnis. Hacker berperan penting dalam membangun platform, baik web-based atau aplikasi sebagai awal membangun startup. Seorang hacker memiliki sifat yang analitis, tersruktur, dan problem solver.

### **2.3.3 Talent Hustler**

Talent Hustler adalah seorang pebisnis yang memastikan perusahaan menghasilkan uang. Selain itu, hustler juga membangun tim, menyalakan rasa semangat, mengatur proyek, memperbarui model bisnis, membangun kerja sama, dan membimbing tim dan memperkuat budaya. Hustler memiliki passion dalam menjual, memasarkan, dan memperkenalkan produk startup-nya. Selain itu,

hustler mempunyai sifat rajin dan giat, disiplin, networker serta berpikir jangka panjang.

### **2.4 Inkubator Bisnis**

Inkubator bisnis adalah merupakan lembaga intermediasi yang bertujuan untuk menumbuhkan kembangkan wirausaha pemula berbasis teknologi selama periode waktu tertentu. Dan lembaga inkubator bisnis yang berada di bawah pengelolaan perguruan tinggi menggunakan konsep link and match (Lutfiani *et al.*, 2020). Dalam inkubator bisnis terdapat proses inkubasi yang berperan untuk melakukan

pendampingan terhadap tenant dengan tujuan agar tenant dapat menyelesaikan masalahnya dengan baik sehingga jiwa kompetensi atau bersaing dalam usaha dapat bertahan dan berkembang pesat (Siregar *et al.*, 2019). Berdasarkan literatur tersebut dapat disimpulkan bahwa dalam inkubator bisnis merupakan suatu lembaga yang memproses calon bisnis rintisan atau yang baru saja tumbuh untuk di-inkubasi berupa pendampingan terhadap calon tenant baik di perguruan tinggi maupun pada masyarakat yang mempunyai UMKM serta menyediakan fasilitas berupa pendampingan teknologi, manajemen dan modal dengan tujuan tenant dapat menghasilkan produk atau jasa dari inovasi dan kreativitas yang dimiliki sehingga mampu mandiri dan dapat melihat peluang dalam membuka usaha dengan memanfaatkan teknologi dalam melakukan pemasarannya.

### **2.5 Perangkat Lunak**

Dalam pengembangan perangkat lunak, komputer membutuhkan perangkat lunak baik sistem operasi maupun perangkat lunak aplikasi. Untuk itu, dalam merancang dan membangun perangkat lunak dalam penelitian ini, dibutuhkan beberapa perangkat lunak penunjang dalam pengembangannya, diantaranya sebagai berikut:

### **2.8.1 PHP ( *PHP Hypertext Preprocessor* )**

Menurut (Andi, 2014) pada (Novendri *et al.*, 2019), Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat website yang bersifat server-side scripting. PHP bersifat dinamis. PHP dapat dijalankan pada berbagai macam sistem operasi seperti Windows, Linux, dan Mac Os. Selain Apache, PHP juga mendukung beberapa web server lain, seperti Microsoft ISS, Caudium, dan PWS. PHP dapat memanfaatkan database untuk menghasilkan halaman web yang dinamis. Sistem manajemen database yang sering digunakan bersama PHP adalah MYSQL. Namun, PHP juga mendukung sistem manajemen Database Oracle, Microsoft Access, Interbase, d-Base, dan PostgreSQL.

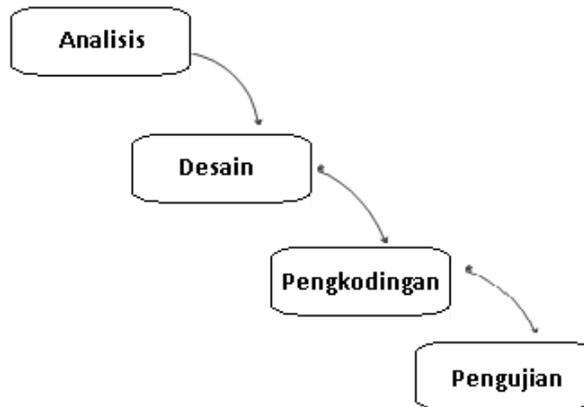
### **2.8.2 MySQL**

MYSQL merupakan sistem manajemen database yang bersifat relational. Artinya, data yang dikelola dalam database yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. MYSQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai yang sangat besar. SQL juga dapat diartikan sebagai antar muka standar untuk sistem manajemen relasional, termasuk sistem yang beroperasi pada komputer pribadi. (Andi,2014) dalam (Novendri *et al.*, 2019).

## **2.7 Metode Pengembangan Perangkat Lunak**

Model pengembangan perangkat lunak atau biasa dikenal dengan *software development life cycle* (SDLC) yang digunakan untuk mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang dipergunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Model yang cocok digunakan untuk spesifikasi sistem yang jarang berubah adalah model air terjun (*waterfall*).

Menurut Rosa dan Shalahuddin (2013:28) dalam jurnal (Dermawan & Hartini, 2017) model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (support). Model *waterfall* sangat cocok digunakan untuk pengembangan perangkat lunak yang jarang berubah-ubah. Berikut adalah gambar model dari SDLC air terjun:



Gambar 2.4 SDLC Metode Air Terjun (*Waterfall*)

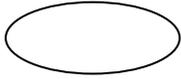
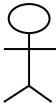
## 2.8 Unified Modeling Language (UML)

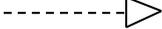
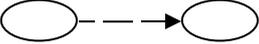
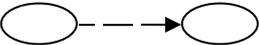
“Unified Modeling language (UML) merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek”. UML memiliki banyak diagram diantaranya :

### 2.8.1 Use Case Diagram

Use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Komponen pembentuk diagram use case yaitu (a) Aktor, menggambarkan pihak-pihak yang berperan dalam sistem. (b) Use case, aktivitas/sarana yang disiapkan sistem. (c) Hubungan (link), aktor mana saja yang terlibat dalam use case ini. (Rosa dan Salahuddin 2015:155) dalam (Nirsal *et al.*, 2019)

Tabel 2.1 Simbol *Use Case Diagram*

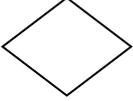
Simbol	Keterangan
<i>Use Case</i> 	Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.

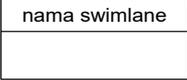
Asosiasi 	Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Generalisasi 	Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> .
<<Include>> 	<i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.
<<Extend>> 	<i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.

### 2.8.2 Activity Diagram

*Activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2 Simbol *Activity Diagram*

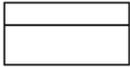
Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu.
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.

### 2.8.3 Class Diagram

*Class diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Tabel 2.3 Simbol *Class Diagram*

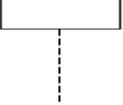
SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

#### 2.8.4 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Tabel 2.4 *Symbol Class Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas.

## 2.9 Blackbox Testing

Metode Blackbox Testing adalah sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail software. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing masing. Tidak ada upaya untuk mengetahui kode program apa yang output pakai (Latif, 2015). Proses Black Box Testing dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan (Wahyudi, Utami, & Arief, 2016). Salah satu jenis pengujian Blackbox Testing adalah dengan menggunakan teknik Equivalence Partitioning (EP) yang penulis gunakan untuk menguji masukan serta membagi masukan kedalam kelompok-kelompok berdasarkan fungsinya. Sehingga didapatkan sebuah test case yang akurat (Aristoteles, Wardiyanto, & Pratama, 2015) dalam (Ningrum *et al.*, 2019).