

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem**

Menurut Sutanto dalam Djahir dan Pratita (2015:6) mengemukakan bahwa “sistem adalah kumpulan/grup dari subsistem/bagian/komponen apapun, baik fisik ataupun nonfisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu”. Sedangkan menurut Mulyani (2016:2) menyatakan bahwa “sistem bisa diartikan sebagai sekumpulan sub sistem, komponen yang saling bekerja sama dengan tujuan yang sama untuk menghasilkan output yang sudah ditentukan sebelumnya”.

#### **2.2 Keputusan**

Menurut Nadwa dalam Jurnal Pendidikan Islam (2014) Keputusan (decision) secara harfiah berarti pilihan (choice). Pilihan yang dimaksud disini adalah pilihan dari dua atau lebih kemungkinan, atau dapat dikatakan pula sebagai keputusan dicapai setelah dilakukan pertimbangan dengan memilih satu kemungkinan pilihan. Definisi diatas mengandung pengertian, dalam keputusan yaitu : 1. Ada pilihan atas dasar logika atau pertimbangan. 2. Ada beberapa alternatif yang harus dipilih salah satu yang terbaik. 3. Ada tujuan yang ingin dicapai dan keputusan itu makin mendekatkan pada tujuan tersebut.

#### **2.3 Sistem Pendukung Keputusan ( Decision Support System )**

Sistem adalah kumpulan dari obyek-obyek seperti manusia, resources, konsep, dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau

memenuhi suatu tujuan. Ludwig Von Bertalanfy berpendapat bahwa sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur 17 tersebut dengan lingkungan. Sedangkan menurut Jogiyanto sistem adalah komponen- komponen yang saling berhubungan untuk mencapai satu tujuan tertentu. Suatu sistem dapat dirumuskan sebagai setiap kumpulan bagian-bagian atau subsistem- subsistem yang disatukan, yang dirancang untuk mencapai suatu tujuan. Kata keputusan (decision) berarti pilihan (choice), yaitu pilihan dari dua atau lebih kemungkinan. Keputusan juga dapat berarti kegiatan memilih suatu strategi atau tindakan dalam pemecahan suatu masalah. Tindakan memilih strategi atau aksi yang diyakini manajer akan memberikan solusi terbaik atas sesuatu itu disebut pengambilan keputusan (ma'ruf, 2016).

Sistem Pendukung Keputusan (SPK) didefinisikan sebagai sebuah sistem yang bertujuan untuk mendukung seorang manajer mengambil sebuah keputusan dalam kondisi permasalahan yang semi terstruktur. SPK difungsikan sebagai fasilitas yang dapat memperkuat kapabilitas sang pengambil keputusan, namun tidak sepenuhnya menggantikan peran pengambil keputusan tersebut. SPK digunakan pada pengambilan keputusan yang melibatkan pertimbangan dari manajer, atau pada pengambilan keputusan yang tidak sepenuhnya dapat diselesaikan dengan perhitungan. Komponen Sistem Pendukung Keputusan (SPK) terdiri dari: (ma'ruf, 2016).

1. Subsistem Manajemen Data (Data-management Subsystem).
2. Subsistem Manajemen Model (Model Management Subsystem).
3. Subsistem Antarmuka Pengguna (User Interface Subsystem).

4. Subsistem Manajemen Berbasis Pengetahuan (Knowledge-based Management Subsystem) (ma'ruf, 2016).

## **2.4 Mobil Bekas**

Mobil bekas adalah kendaraan yang telah dipakai oleh pemilik mobil dalam jangka waktu tertentu sehingga menimbulkan penyusutan. Nilai penyusutan mobil bekas yang membuat harganya turun. Jual beli mobil bekas telah berlangsung lama dan menjadikan ceruk bisnis tersendiri. Sekarang ini sudah banyak sekali showroom mobil bekas yang menawarkan harga dan kualitas yang sepadan. Bisnis mobil bekas yang semakin pesat merupakan pesaing baru bagi dealer-dealer mobil baru yang dapat merebut pasarnya. Pasar mobil bekas yang besar juga membuat para perusahaan pembiayaan berlomba-lomba membiayai konsumen secara kredit dengan bunga yang kompetitif yang tidak kalah dengan mobil baru.

Kualitas produk mobil bekas sangat mempengaruhi minat konsumen untuk membeli. Kualitas produk mobil bekas antara lain dapat diukur dari tahun perakitan, kilometer, kondisi fisik interior dan eksterior, kondisi mesin, kondisi ban, warna serta atribut-atribut lainnya yang membuat harga mobil tersebut dapat berubah-ubah. Kualitas produk sebuah mobil bekas menjadi tolak ukur konsumen dalam menilai kelayakan mobil untuk dibeli. (Indra Sasmita et al : 2013)

## **2.5 Metode SMART**

**SMART** (*Simple Multi Attribute Rating Technique*) merupakan metode pengambilan keputusan yang multi-atribut. 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 skala tertentu dan tiap atribut mempunyai bobot yang menggambarkan seberapa pentingkah suatu atribut dibandingkan dengan atribut lain. Pembobotan dan pemberian peringkat ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif yang terbaik.

**SMART** menggunakan linier adaptif model untuk meramal nilai setiap alternatif.

**SMART** lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisis yang terbaik adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan. Pembobotan pada **SMART** menggunakan skala 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif. Model yang digunakan dalam **SMART** ada beberapa tahapan sebagai berikut (Goodwin and Wright 2004) :

### **1. Menentukan Kriteria**

Menentukan kriteria yang digunakan dalam menyelesaikan masalah pengambilan keputusan. Untuk menentukan kriteria-kriteria apa saja yang digunakan dalam sistem pengambilan keputusan ini diperlukan data-data dari pengambil keputusan atau pihak yang berwenang/kompeten terhadap masalah yang akan diselesaikan.

### **2. Menentukan Bobot Kriteria**

Memberikan bobot kriteria pada masing-masing kriteria dengan menggunakan interval 1-100 untuk masing-masing kriteria dengan prioritas terpenting.

### **3. Normalisasi Bobot Kriteria**

Menghitung normalisasi bobot dari setiap kriteria dengan membandingkan nilai bobot kriteria dengan jumlah bobot kriteria

misalkan nilai untuk kriteria harga sudah dapat dipastikan berbentuk kuantitatif sedangkan nilai untuk kriteria fasilitas bisa jadi berbentuk kualitatif (sangat lengkap, lengkap, kurang lengkap). Apabila nilai kriteria berbentuk kualitatif maka kita perlu mengubah ke data kuantitatif dengan membuat parameter nilai kriteria, misalkan sangat lengkap artinya 3, lengkap artinya 2 dan tidak lengkap artinya 1.

### **4. Menentukan Nilai Utility**

Menentukan nilai *utility* dengan mengkonversikan nilai kriteria pada masing-masing kriteria menjadi nilai kriteria data baku. Nilai *utility* ini tergantung pada sifat kriteria itu sendiri.

- Kriteria Biaya (*Cost Criteria*)

Kriteria yang bersifat “lebih diinginkan nilai yang lebih kecil” kriteria seperti ini biasanya dalam bentuk biaya yang harus dikeluarkan (misalkan kriteria harga, kriteria penggunaan bahan bakar per kilometer untuk pembelian mobil, periode pengembalian modal dalam suatu usaha, kriteria waktu pengiriman) dapat dihitung dengan menggunakan persamaan:

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

Gambar 2.1 Kriteria Biaya

### Keterangan

- **ui(ai)** : nilai utility kriteria ke-i untuk alternatif ke-i
- **cmax** : nilai kriteria maksimal
- **cmin** : nilai kriteria minimal
- **cout** : nilai kriteria ke-i
- Kriteria Keuntungan (*Benefit Criteria*)

Kriteria yang bersifat “lebih diinginkan nilai yang lebih besar”, kriteria seperti ini biasanya dalam bentuk keuntungan (misalkan kriteria kapasitas tangki untuk pembelian mobil, kriteria kualitas dan lainnya)

### 5. Menentukan Nilai Akhir

Menentukan nilai akhir dari masing-masing dengan mengalikan nilai yang didapat dari normalisasi nilai kriteria data baku dengan nilai normalisasi bobot kriteria

$$u(a_i) = \sum_{j=1}^m w_j u_j(a_i), \quad i = 1, 2, \dots, m$$

Gambar 2.2 Rumus Nilai akhir

### Keterangan:

- **u(ai)** : nilai total untuk alternatif ke-i
- **wj** : nilai bobot kriteria ke-j yang sudah ternormalisasi
- **uj(ai)** : nilai utility kriteria ke-j untuk alternatif ke-i

## 6. Perangkingan

Hasil dari perhitungan Nilai akhir kemudian diurutkan dari nilai yang terbesar hingga yang terkecil, alternatif dengan nilai akhir yang terbesar menunjukkan alternatif yang terbaik

### 2.6 Logika Fuzzy

Menurut Agung Setiawan dkk (2018, h. 2), secara umum *fuzzylogic* adalah sebuah metode “berhitung” dengan variabel kata-kata, sebagai pengganti berhitung dengan bilangan. Memang kata-kata yang digunakan dalam *fuzzy* tidak setepat bilangan, namun kata yang digunakan lebih dekat dengan intuisi manusia, seperti kata “merasakan”, “kira-kira”, “lebih kurang”, dan sebagainya.

Sesuai dengan perkembangan daya pikir manusia, maka logika fuzzy ini menjadi populer untuk digunakan dalam riset, karena kemampuannya dalam menjembatani bahasa mesin yang serba tepat dengan bahasa manusia yang cenderung tidak tepat. Biasanya diistilahkan dengan kata signifikan

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*, misalnya umur, temperatur, dan lain-lain.

Fungsi keanggotaan atau membership function adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antara 0 sampai 1.

Beberapa alasan mengapa orang menggunakan *fuzzy logic*, yaitu:

1. Konsep *fuzzy logic* mudah dimengerti. Konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
2. *Fuzzy logic* sangat fleksibel.
3. *Fuzzy logic* memiliki toleransi terhadap data-data yang sangat tepat.
4. *Fuzzy logic* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. *Fuzzy logic* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. *Fuzzy logic* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.

### **2.6.1 Fuzzy Tahani**

Fuzzy Tahani adalah salah satu cabang dari logika fuzzy, yang merupakan salah satu metode fuzzy yang menggunakan basis data standar. Tahani mendeskripsikan suatu metode pemrosesan query fuzzy, dengan didasarkan atas manipulasi bahasa yang dikenal dengan nama SQL (Structured Query Language), sehingga model fuzzy Tahani sangat tepat digunakan dalam proses pencarian data yang tepat dan akurat (Anggraeni, Indarto, Kusumadewi, 2004 dalam Amalia, L., Fananie, Z. B., Utama, D. N., 2010).

Berikut ini adalah tahapan logika fuzzy model Tahani (Kahar, 2013) :

1. Menggambarkan fungsi keanggotaan (membership function) untuk setiap kriteria atau variabel fuzzy, yaitu suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (derajat keanggotaan) yang memiliki interval antara 0 sampai 1, salah satu cara yang dapat digunakan adalah dengan pendekatan fungsi. Pendekatan fungsi keanggotaan berbentuk segitiga.

2. Fuzzifikasi adalah fase pertama dari perhitungan fuzzy yaitu pengubahan nilai tegas ke nilai fuzzy. Dimana setiap variabel fuzzy dihitung nilai derajat keanggotaannya terhadap setiap himpunan fuzzy.

3. Fuzzifikasi query diasumsikan sebuah query konvensional (non fuzzy) DBMS yang akan mencoba membuat dan menerapkan sebuah sistem dasar logika fuzzy query atau disebut juga dengan pembentukan query dengan menggunakan relasi dasar. Operator yang digunakan untuk relasi dasar dalam pembentukan query pada himpunan fuzzy yaitu sebagai berikut (Kahar, 2013):

- a. Interseksi, operator ini berhubungan dengan operasi interseksi pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan dengan persamaan berikut:

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y))$$

- b. Union, operator ini berhubungan dengan operasi union pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan dengan persamaan berikut :

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y))$$

- c. Komplemen, operator ini berhubungan dengan operasi komplemen

pada himpunan. A-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1 dengan persamaan berikut ini :  $\mu = 1 - \mu A(x)$  4. Setelah diperoleh hasil operasi relasi dari pembentukan query, maka data hasil rekomendasi baik operator AND atau OR adalah nilai rekomendasi  $> 0$

## **2.7 Perangkat lunak yang digunakan dalam membuat aplikasi**

### **2.7.1 Website**

Dimuat dalam jurnal Guntur Wibisono, Wahyu Eko Susanto (2015) bahwa menurut pendapat Arief (2011:7), “Web adalah salah satu aplikasi yang berisikan dokumen–dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (hypertext transfer protokol) dan untuk mengakses menggunakan perangkat lunak yang disebut browser”. Fungsi website diantaranya :

1. Media Promosi
2. Media Pemasaran
3. Media Informasi
4. Media Pendidikan
5. Media Komunikasi

### **2.7.2 HTML**

Menurut Anhar (2010), HTML (*Hyper Text Markup Language*) merupakan bahasa pemrograman web yang memiliki sintak atau aturan

tertentu dalam menuliskan script atau kode-kode, sehingga browser dapat menampilkan informasi dengan membaca kode-kode HTML.

### **2.7.3 CSS**

Menurut Ummy Gusti Salamah, S.ST.,MIT (2021), CSS (*Cascading Style Sheet*) adalah bahasa yang dapat digunakan untuk mendefinisikan bagaimana suatu bahasa *markup* ditampilkan pada suatu media dimana bahasa *markup* ini salah satunya adalah *HTML*.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antarparagraf, spasi antar tekps, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

### **2.7.4 PHP**

Menurut (Jubilee Enterprise, 2017) dalam Buku “PHP Komplit”. PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis website. Sebagai sebuah aplikasi, website tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis artinya, website tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu (misalnya, menampilkan produk yang berbeda – beda untuk setiap pengunjung). Interaktif artinya, website tersebut dapat memberi feedback bagi user (misalnya, menampilkan hasil pencarian produk).

### **2.7.5 JavaScript**

JavaScript berfokus pada proses pengolahan data di sisi client dan

menyajikan komponen web yang lebih interaktif serta berfungsi untuk menambah fungsionalitas dan kenyamanan halaman web (Solichin, 2016:11). Menurut Sibero (2013:150) mengatakan bahwa “Javascript adalah suatu bahasa pemrograman yang di kembangkan untuk dapat berjalan pada web browser”.

Berdasarkan kutipan di atas, dapat disimpulkan bahwa javascript merupakan bahasa pemrograman yang berbasis client dan script untuk tampilan pendukung pada website sehingga membuat halaman menjadi lebih interaktif.

#### **2.7.6 Bootstrap**

Menurut (Nugroho & Setiyawati, 2019), bootstrap adalah framework css untuk membuat tampilan web. Bootstrap menyediakan class dan komponen yang sudah siap dipakai.

#### **2.7.7 JQuery**

Menurut (Wahyudi, 2017), jquery merupakan sekumpulan kode Javascript yang dibuat dalam berbagai modul dan digunakan sesederhana mungkin.

#### **2.7.8 MySQL**

MySQL bekerja menggunakan SQL Language (Structure Query Language), yang dapat diartikan bahwa MySQL merupakan standar penggunaan database di dunia untuk pengolahan data. Kelebihan yang dimiliki MySQL yaitu bersifat open source, yang memiliki kemampuan untuk dikembangkan lagi. (Wahyudi, 2017).

## 2.8 Pemodelan Objek

### 2.8.1 UML (*Unified Modeling Language*)

Menurut Sukamto dan Shalahuddin (2018:133), mendefinisikan bahwa “UML merupakan sebuah standar Bahasa yang digunakan untuk menganalisis dan merancang serta menggambarkan arsitektur program dalam pemrograman *object oriented*”.

UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi *object* dan *design* berorientasi *object* (OOAD&D/*object oriented analysis and design*) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode *Booch*, *Rumbaugh* (OMT) dan *Jacobson*. Tetapi UML mencakup lebih luas daripada OOAD. Pada pertengahan saat pengembangan UML, dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML bakal menjadi bahasa standar pemodelan pada masa yang akan datang (yang sekarang sudah banyak dipakai oleh berbagai kalangan) (Pratama 2017).

Diagram – Diagram UML

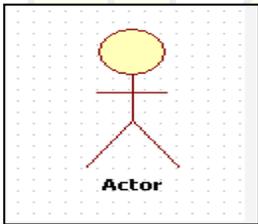
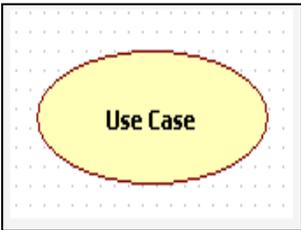
Ada beberapa diagram yang disediakan UML antara lain:

#### 2.8.2 Diagram Use Case

Menurut Tohari dalam Tabrani dan Aghniya (2019:46) menyimpulkan bahwa, “*use case* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor”.

Menurut Pratama (2019b), “*Use case diagram* adalah gambaran grafis dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detil tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Di dalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat”. Berikut adalah beberapa komponen dalam Diagram Use Case:

**Tabel 2.1** Komponen *use case Diagram*.  
Pratama (2019b)

<b>Komponen Use Case</b>	<b>Penjelasan</b>
	<p>Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem.</p>
	<p><i>Use case</i> adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.</p>

Ada beberapa relasi yang terdapat dalam *use case*, antara lain:

**Tabel 2.2** Relasi Tabel *use case diagram*.  
Pratama (2019b)

<b>Relasi Use Case</b>	<b>Penjelasan</b>
  Relasi <i>Association</i>	<i>Association</i> , menghubungkan link antara element.
  Relasi <i>Generalization</i>	<i>Generalization</i> disebut juga <i>inheritance</i> (pewarisan), sebuah elemen dapat merupakan spesialisasi dari element lainnya.
  Relasi <i>Depedency</i>	<i>Dependency</i> , sebuah elemen bergantung dalam beberapa cara ke element lainnya.

Tipe relasi / *stereotype* yang mungkin terjadi pada *use case diagram*:

**Tabel 2.3** Tabel *stereotype* yang mungkin terjadi pada *use case diagram*. Pratama (2019b)

<b>Relasi/ Stereotype</b>	<b>Penjelasan</b>
<b>&lt;&lt;include&gt;&gt;</b>	Kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
<b>&lt;&lt;extends&gt;&gt;</b>	Kelakuan yang hanya berjalan dibawah kondisi tertentu seperti menggerakkan alarm.
<b>&lt;&lt;Communicates&gt;&gt;</b>	Ditambahkan untuk asosiasi yang mungkin menunjukkan asosiasinya adalah <i>communicates association</i> . Ini merupakan pilihan selama asosiasi hanya tipe <i>relationship</i> yang dibolehkan antara <i>actor</i> dan <i>use case</i> .

### 2.8.3 Diagram Activity

Menurut (Penerbit Andi, 2014). *Activity Diagram* adalah teknik untuk menggambarkan logika procedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung behavior paralel. Activity Diagram menggambarkan aliran fungsionalitas sistem. Dapat juga digunakan untuk menggambar aliran kejadian dalam use case. Berikut beberapa komponen yang terdapat dalam activity diagram:

**Tabel 2.4** Komponen *activity diagram*. (Andi, 2014)

<i>Activity Diagram</i>	Penjelasan
 <i>Start State</i>	<i>Start State</i> , sebagai tanda awal proses dari <i>activity diagram</i> .
 <i>State</i>	State, berfungsi menampung <i>event</i> dalam <i>activity diagram</i> .
 <i>Activity</i>	<i>Activity</i> , memiliki fungsi yang sama dengan <i>state</i> . Menampung <i>event</i> atau aktifitas pada proses sistem.

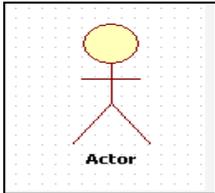
 <p style="text-align: center;"><i>State Transition</i></p>	<p><i>State Transition</i>, berfungsi untuk menunjukkan aliran atau urutan dari <i>event</i> atau aktifitas pada diagram.</p>
 <p style="text-align: center;"><i>Transition to self</i></p>	<p><i>Transition to self</i>, berfungsi untuk menunjukkan transisi sebuah <i>event</i> yang mengarah ke <i>event</i> itu sendiri.</p>
 <p style="text-align: center;"><i>Horizontal Synchronization</i></p>	<p><i>Horizontal Synchronization</i>, berfungsi untuk mensinkronisasikan 2 cabang <i>event</i> yang posisinya horizontal.</p>
 <p style="text-align: center;"><i>Vertical Synchronization</i></p>	<p><i>Vertical Synchronization</i>, berfungsi untuk mensinkronisasikan 2 cabang <i>event</i> yang posisinya vertikal.</p>
 <p style="text-align: center;"><i>Decision</i></p>	<p><i>Decision</i>, digunakan ketika terjadi pemilihan 2 kondisi <i>event</i> pada diagram.</p>

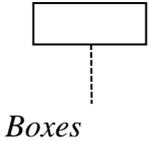
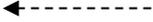
 End State	End State, sebagai tanda akhir dari <i>activity diagram</i> .
------------------------------------------------------------------------------------------------	---------------------------------------------------------------

#### 2.8.4 Sequence Diagram

Menurut (Ade Handini, 2016) dalam Jurnal Khatulistiwa Informatika Vo.1 IV No. 2 yang berjudul "*Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang*". *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antarobjek.

**Tabel 2.5** Komponen *sequence diagram*.  
(Ade Handini, 2016)

<i>Sequence Diagram</i>	<b>Penjelasan</b>
	<i>Actor</i> , menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem.

 <p style="text-align: center;"><i>Boxes</i></p>	<p><i>Boxes</i>, sebuah kontak yang tampil pada posisi paling atas diagram, yang mewakili <i>object</i>, <i>use case</i>, <i>class</i>, dan <i>actor</i>.</p>
 <p style="text-align: center;"><i>Return Message</i></p>	<p><i>Return Message</i>, menggambarkan pesan atau hubungan antara obyek yang menunjukkan urutan kejadian yang terjadi.</p>
<p style="text-align: center;"><i>Lifeline</i></p>	<p><i>Lifeline</i>, eksekusi obyek selama <i>sequence</i> (<i>message</i> dikirim atau diterima dan aktifasinya)</p>
 <p style="text-align: center;"><i>Message to Self</i></p>	<p><i>Message to Self</i>, menggambarkan pesan atau hubungan obyek itu sendiri yang menunjukkan urutan kejadian yang terjadi.</p>
 <p style="text-align: center;"><i>Object Message</i></p>	<p><i>Object Message</i>, menggambarkan pesan atau hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.</p>