

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem**

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem. Sebagai contoh, raket dan pemukul bola kasti (masing-masing sebagai elemen) tidak bisa membentuk sebuah sistem, karena tidak ada system permainan olahraga yang memadukan kedua peralatan tersebut (Kadir ; 2014 : 61).

##### **II.1.1 Karakteristik Sistem**

Supaya sistem itu dikatakan baik memiliki karakteristik yaitu:

1. **Komponen**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. **Batasan Sistem (*Boundary*)**

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini

memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain.

5. Masukan Sistem (*Input*)

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

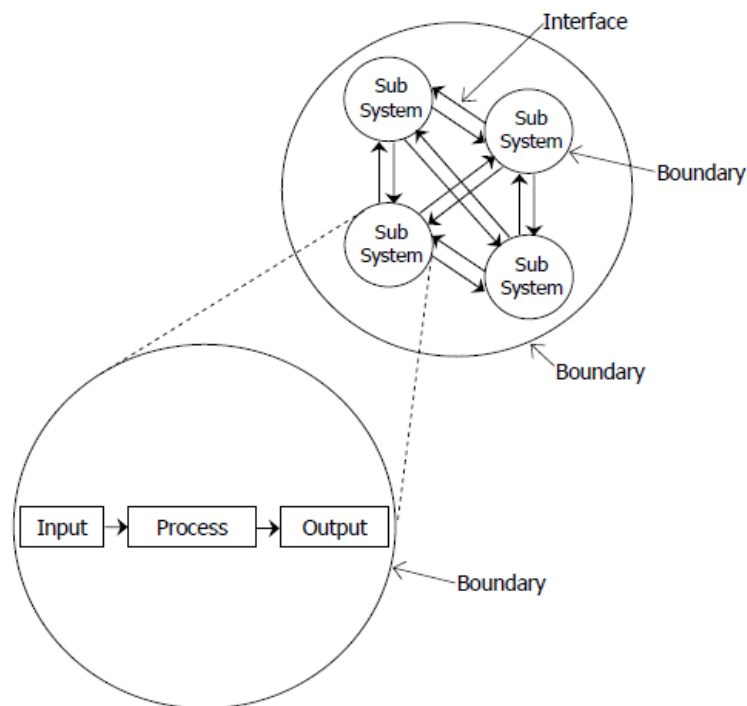
Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

## 8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Hutahaean ; 2014 : 3-5).



**Gambar II.1 Karakteristik Sistem**  
(Sumber : Hutahaean ; 2014 : 5)

## II.2. Sistem Pendukung Keputusan

*Decision support system* atau sistem pendukung keputusan secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semi terstruktur. Secara khusus, Sistem pendukung keputusan didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manager maupun sekelompok manager dalam memecahkan masalah semi terstruktur

dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu (Ritonga ; 2014 : 19).

Sistem pendukung keputusan (SPK) biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk suatu peluang. Aplikasi sistem pendukung keputusan (SPK) digunakan dalam pengambilan keputusan. Aplikasi sistem pendukung keputusan (SPK) menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, 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 atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan (Nofriansyah ; 2014 : 1).

### **II.2.1. Karakteristik Sistem Pendukung Keputusan**

Berikut ini adalah uraian beberapa karakteristik SPK, yaitu:

#### **1. Kapabilitas interaktif**

Yaitu SPK memberi pengambil keputusan akses cepat ke data dan informasi yang dibutuhkan.

## 2. Fleksibilitas

Yaitu SPK dapat menunjang para manajer pembuat keputusan di berbagai bidang fungsional (keuangan, pemasaran, operasi produksi, dan lain-lain).

## 3. Kemampuan menginteraksi model

Yaitu SPK memungkinkan para pembuat keputusan berinteraksi dengan model-model, termasuk memanipulasi model- model tersebut sesuai dengan kebutuhan.

## 4. Fleksibilitas output

Yaitu SPK mendukung para pembuat keputusan dengan menyediakan berbagai macam output, termasuk kemampuan grafik menyeluruh atas pertanyaan- pertanyaan pengandaian (Suhermin ; 2012 : 4).

### **II.2.2. Komponen Sistem Pendukung Keputusan**

Sistem pendukung keputusan dibangun oleh lima komponen yaitu subsistem manajemen data, subsistem antarmuka pengguna, dan subsistem manajemen berbasis pengetahuan. Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama yaitu subsistem manajemen model, subsistem manajemen moden dan antarmuka pengguna. Sistem Pendukung Keputusan disusun dari beberapa subsistem yaitu :

#### 1. Subsistem manajemen data

Basis data yang relevan untuk suatu situasi dan dikelola oleh *database management system* (DBMS). Subsistem ini dapat diinterkoneksi dengan data *warehouse* perusahaan yang relavan untuk pengambilan keputusan.

2. Subsistem manajemen model

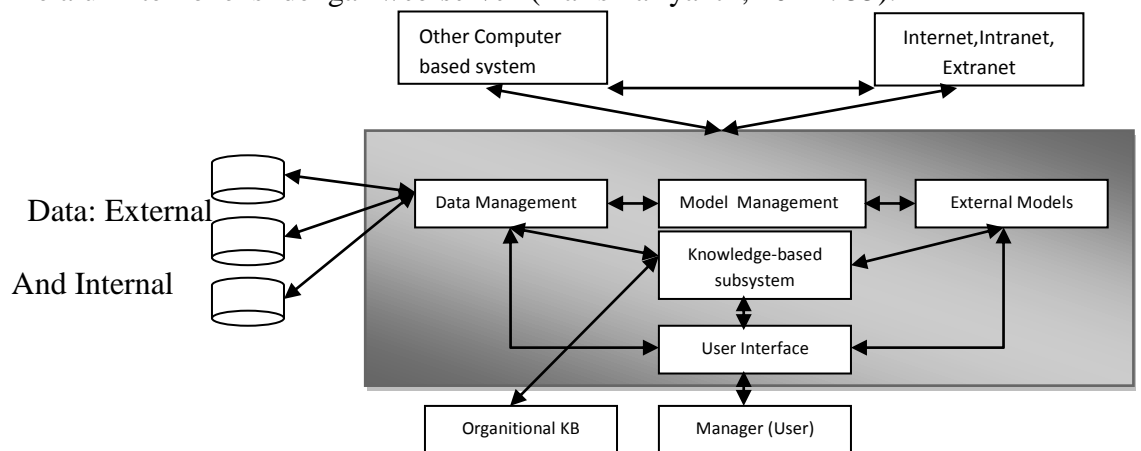
Merupakan paket perangkat lunak yang menyimpan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kemampuan analitik yang tepat. Perangkat lunak ini sering disebut *modelbase management system* (MBMS) dan dapat diimplementasikan pada sistem pengembangan web untuk berjalan pada *server* aplikasi.

3. Subsistem antarmuka pengguna

Merupakan dukungan komunikasi antara dengan pengguna. *Web browser* menjadi salah satu antarmuka yang menampilkan dalam bentuk grafis dan interaktif dengan pengguna.

4. Subsistem manajemen

Berbasis pengetahuan bertindak sebagai komponen independen yang memberikan kemampuan intelegensi untuk memperbesar pengetahuan pengambil keputusan. Perusahaan memiliki sistem manajemen pengetahuan. Keterhubungan subsistem ini dengan sistem pendukung keputusan dapat melalui interkoneksi dengan *web server* (Karismariyanti ; 2011 : 55).



**Gambar II.2 Komponen Sistem Pendukung Keputusan**  
(Sumber : Karismariyanti ; 2011 : 55)

### **II.3. Produksi**

Bagi orang awam, yang dimaksud dengan kegiatan produksi adalah kegiatan untuk menghasilkan barang. Misalnya, kegiatan membuat kursi, membuat baju, atau merakit mobil. Pengertian tersebut sebenarnya tidak salah, hanya saja kurang lengkap. Selain menghasilkan barang atau jasa, kegiatan produksi juga harus menambah manfaat atau nilai guna suatu barang dan jasa. Dengan demikian, produksi adalah kegiatan yang dilakukan oleh orang atau lembaga untuk menghasilkan atau menambah manfaat (nilai guna) suatu barang dan jasa. Orang atau lembaga yang melakukan kegiatan produksi disebut produsen. Untuk mendapatkan definisi produksi yang lebih lengkap, hal-hal berikut perlu diperhatikan.

- a. Kegiatan produksi dilakukan oleh perusahaan.
- b. Tujuan perusahaan adalah untuk memperoleh laba.
- c. Perusahaan mengombinasikan seluruh sumber daya.
- d. Barang atau jasa yang dihasilkan perusahaan ditujukan untuk memenuhi kebutuhan manusia.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa produksi adalah kegiatan yang dilakukan perusahaan untuk memperoleh laba dengan mengombinasikan seluruh sumber daya ekonomi untuk menghasilkan barang atau jasa yang ditujukan untuk memenuhi kebutuhan manusia (Deliarnov ; 2012 : 45).

#### II.4. Metode Fuzzy Tsukamoto

Tsukamoto yaitu setiap konsekuen pada aturan berbentuk IF-THEN harus dipresentasikan dengan suatu himpunan fuzzy, dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan dengan berdasarkan predikat (fire strength). Hasil akhir diperoleh dengan menggunakan rata-rata terbobot. Misalkan ada 2 variabel input, yaitu x dan y serta satu variabel output z. Variabel x terbagi atas dua himpunan yaitu A1 dan A2, sedangkan variabel y terbagi atas himpunan B1 dan B2. Variabel z juga terbagi atas dua himpunan yaitu C1 dan C2. Tentu saja himpunan C1 dan C2 harus merupakan himpunan yang bersifat monoton. Ada 2 aturan yang digunakan, yaitu:

$$[R1] \text{IF}(x \text{ is } A_1) \text{And}(y \text{ is } B_2) \text{THEN}(z \text{ is } C_1)(5)$$

$$[R1] \text{IF}(x \text{ is } A_2) \text{And}(y \text{ is } B_2) \text{THEN}(z \text{ is } C_2)(6)$$

Keterangan:

R1 : Aturan fuzzy

x : variabel pinjaman

$\alpha_1$  : Himpunan pinjaman tertinggi

$\alpha_2$  : Himpunan pinjaman terendah

And : Operator yang digunakan

y : Variabel jaminan

B1 : Himpunan jaminan tertinggi

B2 : Himpunan jaminan terendah

THEN : Operator yang digunakan

Z : Variabel Penghasilan (nilai crisp)

C1 : Himpunan penghasilan tertinggi (harus monoton)

C2 : Himpunan penghasilan terendah (harus monoton) (Tri Murti ; 2015 : 253-254).

#### II.4.1. Studi Kasus Perhitungan Fuzzy Tsukamoto :

Penyelesaian masalah untuk kasus pengadaan barang menggunakan Logika Fuzzy metode Tsukamoto adalah sebagai berikut :

1. Menentukan Variabel yang terkait dalam proses yang akan ditentukan dan fungsi fuzzyfikasi yang sesuai. Pada kasus ini, ada 3 variabel yang akan dimodelkan, yaitu :

a. Variabel penjualan [x], terbagi menjadi 2 himpunan fuzzy, yaitu naik dan turun. Berdasarkan dari data penjualan terbesar dan terkecil tahun 2013, maka fungsi keanggotaan dirumuskan sebagai berikut :

- Fungsi keanggotaan, untuk Variabel Penjualan pada himpunan Turun

$$\mu_{Turun}[X] = \begin{cases} 1; & ; x \leq a \\ (b - x)/(b - a); & a \leq x \leq b \\ 0; & ; x \geq b \end{cases}$$

- Fungsi keanggotaan, untuk Variabel Penjualan pada himpunan Naik

$$\mu_{Naik}[X] = \begin{cases} 0; & ; x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & ; x \geq b \end{cases}$$

$$\begin{aligned} \mu_{Naik}[30] &= (x - 20)/(45 - 20) \\ &= (30-20)/(45-30) \\ &= 0.40 \end{aligned}$$

b. Variabel persediaan [y], terbagi menjadi 2 himpunan fuzzy, yaitu Sedikit dan Banyak. Berdasarkan dari data Persediaan terbesar dan terkecil tahun 2013, maka fungsi keanggotaan dirumuskan sebagai berikut :

- Fungsi keanggotaan, untuk Variabel Persediaan pada himpunan Sedikit

$$\mu_{Banyak}[y] = \begin{cases} 0; & ; y \leq a \\ (y - a)/(b - a); & a \leq y \leq b \\ 1; & ; y \geq b \end{cases}$$

$$\begin{aligned} \mu_{Sedikit}[8] &= (15 - y)/(15 - 5) \\ &= (15-8)/(15-5) \\ &= 0.70 \end{aligned}$$

- Fungsi keanggotaan, untuk Variabel Persediaan pada himpunan Banyak

$$\mu_{Banyak}[y] = \begin{cases} 0; & ; y \leq a \\ (y - a)/(b - a); & a \leq y \leq b \\ 1; & ; y \geq b \end{cases}$$

$$\begin{aligned} \mu_{Banyak}[8] &= (y - 150)/(15 - 5) \\ &= (8-5)/(15-5) \\ &= 0.30 \end{aligned}$$

c. Variabel Pengadaan (z) terbagi menjadi 2 himpunan fuzzy, yaitu berkurang dan bertambah. Berdasarkan dari data Pengadaan terbesar dan terkecil tahun 2013, maka fungsi keanggotaan dirumuskan sebagai berikut :

- Fungsi keanggotaan, untuk Variabel Pengadaan pada himpunan Berkurang

$$\mu_{Berkurang}[z] = \begin{cases} 1; & ; z \leq a \\ (b - z)/(b - a); & a \leq z \leq b \\ 0; & ; z \geq b \end{cases}$$

- Fungsi keanggotaan, untuk Variabel Pengadaan pada himpunan Berkurang

$$\mu_{\text{Berkurang}}[z] = \begin{cases} 1; & ; z \leq a \\ (b - z)/(b - a); & a \leq z \leq b \\ 0; & ; z \geq b \end{cases}$$

$$\mu_{\text{Berkurang}}[z] = \begin{cases} 1; & ; z \leq 25 \\ (55 - z)/(55 - 25); & 25 \leq z \leq 55 \\ 0; & ; z \geq 55 \end{cases}$$

- Fungsi keanggotaan, untuk Variabel Pengadaan pada himpunan Bertambah

$$\mu_{\text{Bertambah}}[z] = \begin{cases} 0; & ; z \leq a \\ (z - a)/(b - a); & a \leq z \leq b \\ 1; & ; z \geq b \end{cases} \quad (\text{Ula ; 2014 : 43-44}).$$

## II.5. Visual Basic

VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada *platform* .NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer Visual Basic. Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas *platform Windows*. Fitur baru bahasa VB.NET dibandingkan Visual Basic bahwa bahasa VB.NET mendukung *object-oriented* dan juga *dynamics programming*. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET (Kurniawan ; 2013 : 10).

## **II.6. SQL Server 2008**

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang di develop oleh Microsoft, yang digunakan untuk menyimpan dan mengolah data. Pada SQL Server 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL Server 2008, kita bisa membuat object yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, function, stored procedure, trigger* dan *view*. Selain object, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data (Solution ; 2010 : 101).

## **II.7. UML (*Unified Modeling Language*)**

Hasil pemodelan pada OOAD terdokumentasi dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan system 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 industry perangkat lunak dan pengembangan sistem ( Gata ; 2013 : 4 ).

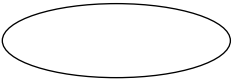
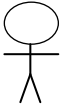

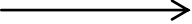
Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut :

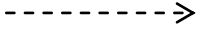
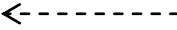
## 1. Use Case Diagram

*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 didalam system informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *Use Case* Diagram, yaitu :

**Tabel II.1. Simbol Use Case**

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.




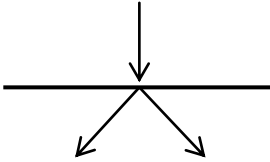
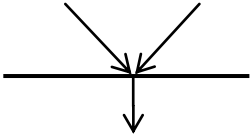
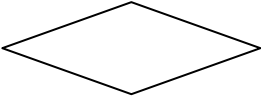

	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gata ; 2013 : 4-6)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu :

**Tabel II.2. Diagram Aktivitas**

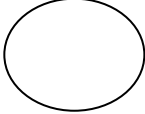
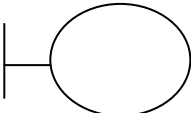


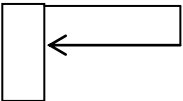

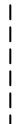
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Gata ; 2013 : 6-7)

### 3. Sequence Diagram

*Sequence* diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence* diagram, yaitu :

**Tabel II.3 Simbol *Sequence Diagram***

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gata ; 2013 : 7-8)

#### 4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu system, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class* diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

*Class* diagram secara khas meliputi : Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau kardinaliti

**Tabel II.4. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gata ; 2013 : 9)

## **II.8. Teknik Normalisasi**

Salah satu topik yang cukup kompleks dalam dunia manajemen *database* adalah proses untuk menormalisasi tabel-tabel dalam *database relasional*. Dengan normalisasi kita ingin mendesain *database relasional* yang terdiri dari tabel-tabel berikut :

1. Berisi data yang diperlukan.
2. Memiliki sesedikit mungkin redundansi.
3. Mengakomodasi banyak nilai untuk tipe data yang diperlukan.
4. Mengefisienkan update.
5. Menghindari kemungkinan kehilangan data secara tidak disengaja/tidak diketahui.

Alasan utama dari normalisasi *database* minimal sampai dengan bentuk normal ketiga adalah menghilangkan kemungkinan adanya “*insertion anomalies*”, “*deletion anomalies*”, dan “*update anomalies*”. Tipe-tipe kesalahan tersebut sangat mungkin terjadi pada *database* yang tidak normal.

### **II.8.1. Bentuk-bentuk Normalisasi**

#### **1. Bentuk tidak normal**

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

## 2. Bentuk normal tahap pertama (1<sup>st</sup> Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

## 3. Bentuk normal tahap kedua (2<sup>nd</sup> normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada primary key secara utuh.

## 4. Bentuk normal tahap ketiga (3<sup>rd</sup> normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi  $X \rightarrow A$ , dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- a. X haruslah superkey pada tabel tersebut.
- b. Atau A merupakan bagian dari primary key pada tabel tersebut.

## 5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian,

terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

#### **6. Boyce Code Normal Form (BCNF)**

- a. Memenuhi 1<sup>st</sup> NF
- b. Relasi harus bergantung fungsi pada atribut superkey (Kusrini ; 2010 : 39-43).