

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Sistem merupakan suatu jaringan yang kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu (Jepersen Hutahaeen, 2014: 2).

Terdapat dua kelompok pendekatan dalam mendefinisikan sistem. Yaitu:

- a. Pendekatan sistem yang lebih menekankan pada prosedur, mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan. Berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
- b. Pendekatan yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai suatu kumpulan dari elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

Dari beberapa pengertian tersebut dapat diambil kesimpulan bahwa sistem merupakan kumpulan kelompok elemen yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan dan sasaran yang di inginkan (Sutabri, 2012).

#### **II.2. Informasi**

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan (Sutabri 2012:29).

### II.3. Sistem Pendukung Keputusan

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 masalah (Hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan) (Dicky Nofriansyah, 2014:1).

#### II.3.1. Karakteristik Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), karakteristik sistem pendukung keputusan (SPK) yaitu :

- a. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan
- b. Adanya *interface* manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan.
- c. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
- d. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan
- e. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

### II.3.2. Ciri-Ciri Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), ciri-ciri sistem pendukung keputusan adalah sebagai berikut :

- a. Banyak pilihan/alternatif
- b. Ada kendala atau surat
- c. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
- d. Banyak input/variabel
- e. Ada faktor resiko, dibutuhkan kecepatan, ketetapan dan keakuratan.

### II.3.3. Fase Dalam Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), tiga fase dalam pengambilan keputusan yaitu :

a. *Intelligence*

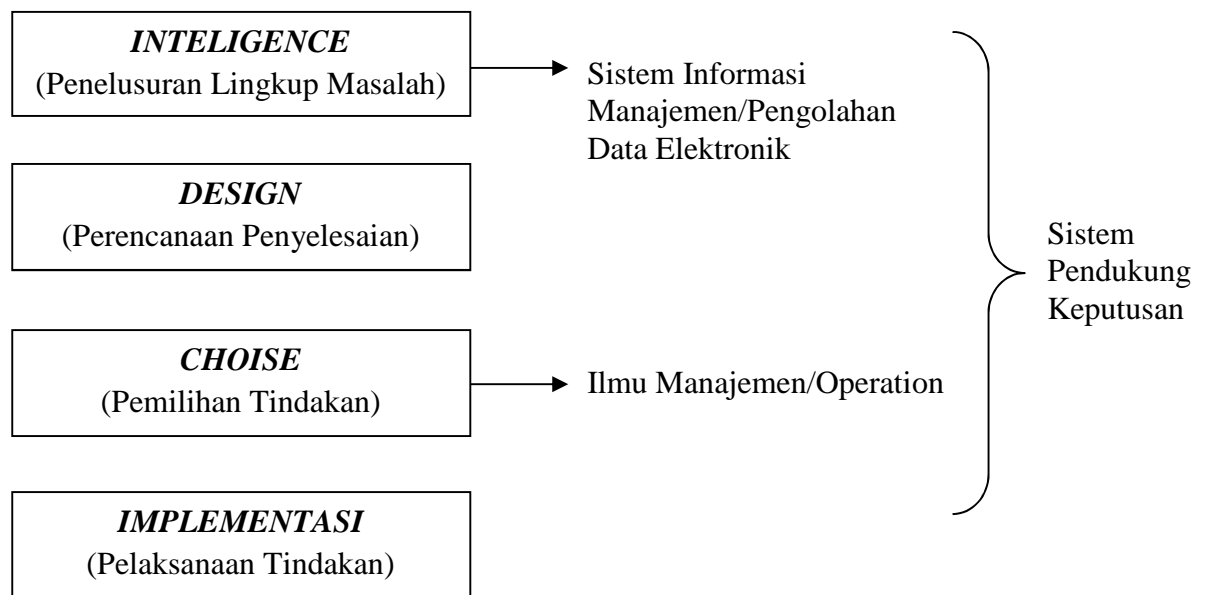
Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematikab secara proses pengenalan masalah. Data masukan diperoleh dan diuji dalam rangka mengidentifikasi masalah.

b. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap melakukan pengujian kelayakan solusi.

c. *Choice*

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



**Gambar II.1. Fase Proses Pengambilan Keputusan**  
Sumber (Dicky Nofriansyah, S.Kom, M.Kom ; 2014)

#### II.3.4. Komponen Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:3), secara garis besar sistem pendukung keputusan dibangun oleh tiga komponen utama yaitu :

a. Subsistem Data (*Database*)

Subsistem data merupakan komponen sistem pendukung keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan untuk diorganisasikan oleh suatu sistem yang disebut dengan Sistem Manajemen Sistem Basis Data (*Database Management System*).

b. Subsistem Model

Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang model adalah bahwa model yang dirancang tidak mampu mencerminkan seluruh variabel alam nyata, sehingga keputusan yg diambil tidak sesuai dengan kebutuhan. Oleh karena itu, dalam menyimpan berbagai model harus diperhatikan dan harus dijaga fleksibilitasnya. Hal lain yg harus diperhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian, keterangan dan penjelasan yang komprehensif mengenai model yg dibuat.

c. Subsistem dialog (*User system Interface*)

Subsistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif, yang dikenal dengan subsistem dialog. Melalui subsistem dialog sistem diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat.

### **II.3.5. Tujuan Sistem Pendukung Keputusan**

Menurut Dicky Nofriansyah (2014:4), tujuan dari sistem pendukung keputusan adalah sebagai yaitu :

- a. Membantu dalam pengambilan keputusan atas masalah yang terstruktur
- b. Memberikan dukungan atas pertimbangan manager dan bukannya dimaksudkan untuk menggantikan fungsi manager.
- c. Meningkatkan efektifitas keputusan yang diambil lebih dari perbaikan efisiensinya.

- d. Kecepatan komputasi komputer memungkinkan para pengambil keputusan untuk banyak melakukan komputasi secara cepat dengan biaya yang sangat rendah.
- e. Peningkatan produktifitas membangun suatu kelompok pengambilan keputusan, terutama para pakar bisa sangat mahal. Sistem pendukung keputusan komputerisasi mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada diberbagai lokasi yang berbeda-beda (menghemat biaya perjalanan). Selain itu produktifitas staf pendukung (misalnya analisis keuangan dan hukum) bisa ditingkatkan. Produktifitas juga ditingkatkan menggunakan peralatan optimalisasi yang menjalankan sebuah bisnis.

#### **II.4. Metode Perbandingan Eksponensial (MPE)**

Metode perbandingan eksponensial (MPE) merupakan salah satu metode untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak. Teknik ini digunakan sebagai pembantu bagi individu pengambil keputusan untuk menggunakan rancang bangun model yang telah terdefinisi dengan baik pada tahapan proses (Sofyansyah, 2014).

##### **II.4.1. Prosedur Metode Perbandingan Eksponensial**

Dalam menggunakan Metode Perbandingan Eksponensial ada beberapa prosedur yang harus dilakukan, yaitu:

1. Menyusun alternatif-alternatif keputusan yang akan dipilih.

2. Menentukan kriteria atau perbandingan keputusan yang penting untuk dievaluasi.
3. Menentukan tingkat kepentingan dari setiap kriteria keputusan.
4. Melakukan penilaian terhadap semua alternatif pada setiap kriteria.
5. Menghitung skor atau nilai total setiap alternatif.
6. Menentukan urutan prioritas keputusan didasarkan pada skor atau nilai total masing-masing alternatif (Sofyansyah, 2014).

#### **II.4.2. Alasan Menggunakan Metode Perbandingan Eksponensial (MPE)**

1. MPE (Metode Perbandingan Eksponensial) mempunyai keuntungan dalam mengurangi bias yang mungkin terjadi dalam analisis.
2. Nilai skor yang menggambarkan urutan prioritas menjadi besar (fungsi eksponensial) ini mengakibatkan urutan prioritas alternatif keputusan lebih nyata.
3. Alasan utama menggunakan Metode Perbandingan Eksponensial (MPE) adalah metode ini cocok dalam penelitian ini dikarenakan metode ini lebih menekankan pada bobot yang telah ditentukan oleh user dan apabila menggunakan ini user dapat langsung mengetahui hasil yang berupa urutan prioritas/peringkat siswa secara nyata (Sofyansyah, 2014).

#### **II.4.3. Formulasi Matematika Metode Perbandingan Eksponensial**

Formulasi perhitungan skor untuk setiap alternatif dalam Metode Perbandingan Eksponensial adalah sebagai berikut :

$$T_{Ni} = \sum_{j=1}^m (R_{ij}) T_{Kj}$$

Keterangan :

$T_{Ni}$  : Total nilai alternatif ke-i

$R_{Kij}$  : Derajat kepentingan relatif kriteria ke-j pada pilihan keputusan i

$TKK_j$ : Derajat kepentingan kriteria keputusan ke-j;  $TKK_j > 0$ ; bulat

m : Jumlah kriteria keputusan

n : Jumlah pilihan keputusan

j : 1,2,3,...,m; m = Jumlah kriteria

i : 1,2,3,...,n; n = Jumlah pilihan alternatif

Penentuan tingkat kepentingan kriteria dilakukan dengan cara wawancara dengan pakar atau melalui kesepakatan curah pendapat. Sedangkan penentuan skor alternatif pada kriteria tertentu dilakukan dengan memberi nilai setiap alternatif berdasarkan nilai kriterianya. Semakin besar nilai alternatif semakin besar pula skor alternatif tersebut. Total skor masing-masing alternatif keputusan akan relatif berbeda secara nyata karna adanya fungsi eksponensial (Sofyansyah, 2014).

## II.5. Bahasa Pemograman

Ada beberapa bahasa pemograman yang digunakan dalam membuat aplikasi diantaranya, Java dan Visual Basic 2010 :



### **II.5.1. Java**

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995 (Eko Sujatmiko, 2012).

### **II.5.2. Visual Basic 2010**

*Visual Basic 2010* adalah inkarnasi dari bahasa visual basic yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat lainnya seperti C++. Anda dapat menggunakan visual basic 2010 untuk membuat aplikasi windows, mobile, web, dan office atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke program lainnya. Visual basic menyediakan berbagai tools dan fitur canggih yang memungkinkan dapat menulis kode, menguji dan menjalankan program tunggal atau terkadang serangkaian program yang terkait dengan satu aplikasi (Christopher Lee:2014:1).

### **II.6. Database**

*Database* adalah sekumpulan file data yang satu sama lainnya saling berhubungan yang diorganisasi sedemikian rupa sehingga memudahkan untuk mendapatkan dan memproses data tersebut (Riyanto, 2016:102).

### **II.6.1. MySQL**

MySQL merupakan salah satu *Relational Database Management System*, bersifat *open source*, MySQL dapat digunakan maupun didistribusikan baik untuk kepentingan individu maupun *corporate* secara gratis, tanpa melakukan lisensi dari pembuatnya (Mukhamad Masrur, 2016:124).

### **II.6.2 SQL Server 2008**

*SQL Server 2008* adalah sebuah *RDBMS (Relational Database Management System)* yang didevelop 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 objek-objek yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, function, stored database, trigger* dan *view*. Selain objek, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data (Elex Media Competindo, 2012).



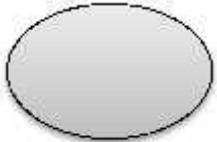


## **II.7. Desain Sistem**

Perancangan desain sistem yang akan dibangun menggunakan pemodelan ERD (*Entity Relationship Diagram*) dan *Unified Modeling System (UML)*. Diagram-diagram yang digunakan *use case diagram, activity diagram*, dan *sequence diagram*.

### II.7.1. ERD (*Entity Relationship Diagram*)

*Entity relationship diagram* adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Adapun simbol-simbol dari ERD yang digunakan, yaitu :

**Tabel II.2 Simbol *Entity Relationship Diagram***

Notasi	Keterangan
	Entitas yaitu kumpulan dari objek yang dapat didefinisikan secara unik
	Relasi yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu kesatu, satu ke banyak, dan banyak ke banyak
	Atribut yaitu karakteristik dari <i>entity</i> atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis hubungan antara <i>entity</i> dengan atributnya dan himpunan entitas.
	<i>Input/Output</i> data, yaitu proses <i>Input/Output</i> data, parameter, informasi.

(Sumber : Tata Sutabri ; 2012)

Pada dasarnya ada tiga komponen yang digunakan, yaitu:

### 1. Entitas

Entitas merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entity ini biasanya digambarkan dengan persegi panjang.

### 2. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol *elips*.

### 3. Hubungan/Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam basis data yaitu:

#### a. Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

#### b. Satu ke banyak (*One to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

#### c. Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

### II.7.2. UML (*Unified Modelling Language*)

*Unified Modelling Language* (UML) menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Artifak dalam UML adalah informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasilkan oleh pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan desain adalah bahwa setiap perubahan terjadi pada suatu artifak yang harus juga dilakukan pada artifak lainnya (Sumber: Evi Triandini Dan I gede Suardika : 2012:15)

Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. *Unified Modeling Language* biasa digunakan untuk :


1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, di buat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.

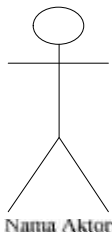
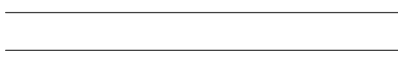
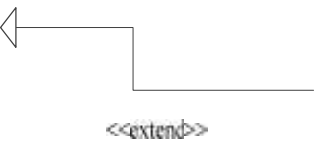
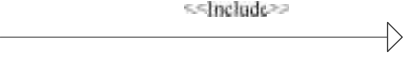
4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionality* dengan *stereotypes* (Yuni Sugiarti; 2013:3)

### II.7.2.1. Use Case Diagram

*Use case* diagrams merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan di buat. Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu *use case*, *actor* dan relasi. Berikut adalah simbol-simbol yang ada pada diagram *use case* (Yuni Sugiarti; 2013:42).

**Tabel II.3. Simbol-Simbol pada Use Case Diagram**

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> ; biasanya ditanyakan dengan menggunakan kata kerja di awal fase nama <i>use case</i> .

<p>Aktor</p> 	<p>Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i>.</p>
<p>Asosiasi</p> 	<p>Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor</p>
<p>Extend</p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, arah panah menunjuk pada use case yg dituju.</p>
<p>Include</p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan.</p>

(Sumber : Yuni Sugiarti : 2013)

### II.7.2.2. Class Diagram




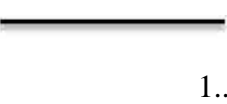


Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas—kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.

3. Operasi atau model adalah fungsi-fungsi yang di miliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Yuni Sugiarti; 2013:57).

**Tabel II.4. Simbol-simbol Class Diagram**

	Simbol	Deskripsi
Package		Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi		Kelas pada struktur sistem
Antarmuka/Interface		Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
Asosiasi		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/directed asosiasi		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi		Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus)
Ketergantungan/defedency		Relasi antar kelas dengan makna ketergantungan antar kelas
Agregasi		Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> ).

(Sumber : Yuni Sugiarti : 2013)

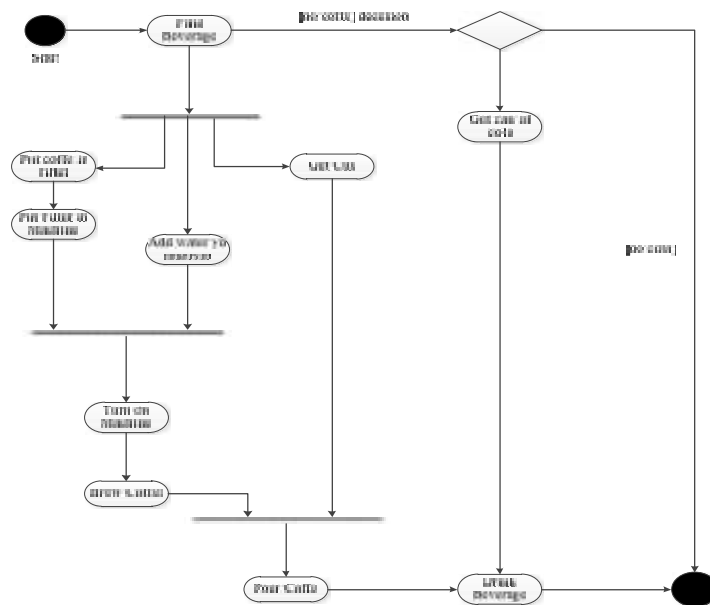


### II.7.2.3. Activity Diagram

Diagram aktivitas atau *Activity Diagram* menngambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis.

*Activity Diagram* merupakan state diagram khusus, dimana sebagian besar sttate adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu acticity diagram menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas (Yuni Sugiarti ; 2013 : 75).



**Gambar II.2. Activity Diagram**  
(Sumber : Yuni Sugiarti : 2013)

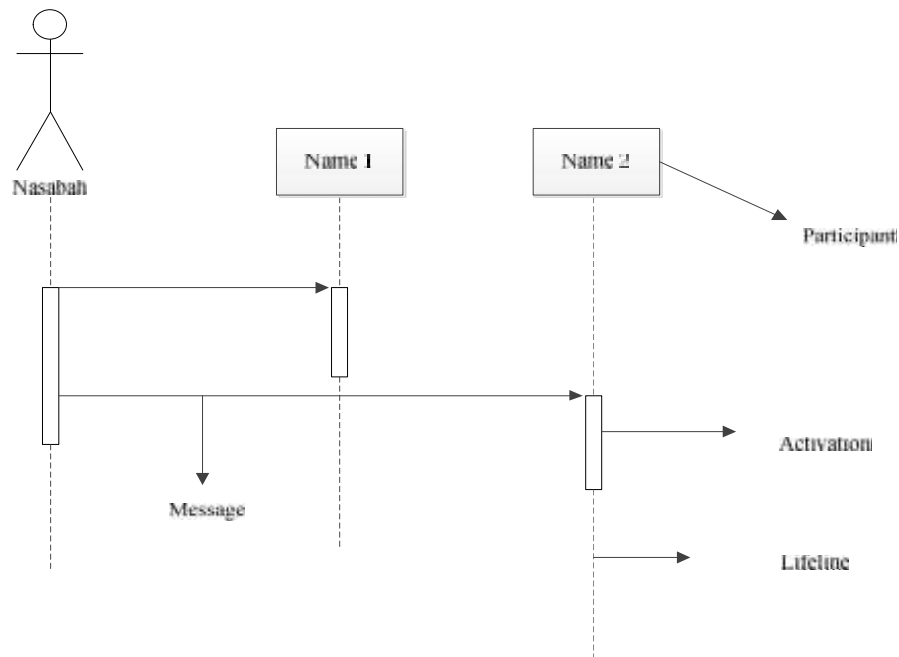
#### II.7.2.4. Sequence Diagram

Diagram sequence menggambarkan kelakuan/pelaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui pula objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Diagram sequence memiliki ciri yang berbeda dengan diagram interaksi pada diagram kolaborasi sebagai berikut :

1. Pada diagram sequence terdapat garis hidup objek. Garis hidup objek adalah garis vertical yang mencerminkan eksistensi sebuah objek sepanjang periode waktu. Sebagian besar objek-objek yang tercakup dalam diagram interaksi akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek-objek itu diletakkan dibagian atas diagram dengan garis hidup tergambar dari atas hingga bagian bawah diagram. Suatu objek lain dapat saja diciptakan, dalam hal ini garis hidup dimulai saat pesan *destroy*, jika kasus ini terjadi, maka garis hidupnya juga berakhir.
2. Terdapat fokus kendali (Focus Of Control), berupa empat persegi panjang ramping dan tinggi yang menampilkan aksi suatu objek secara langsung atau sepanjang sub ordinat. Puncak dari empat persegi panjang adalah permulaan aksi, bagian dasar adalah akhir dari suatu aksi. Pada diagram ini mungkin juga memperhatikan penyaringan (*nesting*) dan fokus kendali yang disebabkan oleh proses rekursif dengan menumpuk fokus kendali yg lain pada induknya. (Yuni Sugiarti; 2013:70).

Berikut simbol-simbol yang ada pada sequence diagram



**Gambar II.3. Simbol Sequence**  
(Sumber : Yuni Sugiarti : 2013)

## II.8. Normallisasi

Normalisasi adalah suatu teknik dengan pendekatan *bottom-up* yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji hubungan, yaitu *functional dependencies* antara atribut. Pengertian lainnya adalah suatu teknik yang menghasilkan sekumpulan hubungan dengan sifat-sifat yang diinginkan dan memenuhi kebutuhan pada perusahaan (Indrajani : 2015 ; 7).

Proses normalisasi merupakan proses pengelompokkan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Proses ini selalu diuji pada beberapa kondisi. Apakah ada kesulitan pada saat menambah (insert), menghapus (delete), mengubah (update), atau membaca (retrieve) pada suatu database. Bila

ada kesulitan pada pengujian tersebut maka relasi dapat dipecah dalam beberapa tabel lagi (Tata Sutabri : 2012 ; 38).

Adapun bentuk-bentuk normalisasi adalah sebagai berikut :

#### 1. Unnormalized Form (UNF)

Merupakan suatu tabel yang berisikan satu atau lebih grup yang berulang. Membuat tabel yang *unnormalized*, yaitu dengan memindahkan data dari sumber informasi.

Contoh : nota penjualan yang disimpan kedalam format tabel dengan baris dan kolom.

#### 2. First Normal Form (1NF)

Merupakan sebuah relasi di mana setiap baris dan kolom berisikan satu dan hanya satu nilai.

Proses UNF ke 1NF

- a. Tentukan satu atau sekumpulan atribut sebagai kunci untuk tabel *unnormalized*.
- b. Identifikasikan grup yang berulang dalam tabel *unnormalized* yang berulang untuk kunci atribut.

#### 3. Second Normal Form (2NF)

Berdasarkan pada konsep *full functional dependency*, yaitu A dan B merupakan atribut sebuah relasi. B dikatakan *fully dependent* terhadap A jika B *functional dependent* pada A tetapi tidak pada propersubset dari A. 2NF merupakan sebuah relasi dalam 1NF dan setiap atribut *non-primary-key* bersifat *fully functionally dependent* pada primary key.

1NF ke 2NF

- a. Identifikasikan *primary key* untuk relasi 1NF.
- b. Identifikasikan *functional dependencies* dalam relasi.
- c. Jika terdapat *partial dependencies* terhadap *primary key*, maka hapus dengan menempatkan dalam relasi yang baru bersama salinan determinannya.

#### 4. Third Normal Form (3NF)

Berdasarkan pada konsep *transitive dependency*, yaitu suatu kondisi di mana A,B dan C merupakan atribut sebuah relasi, maka A – B dan B – C, maka *transitively dependent* pada A melalui B. 3NF adalah sebuah relasi dalam 1NF dan 2NF, dimana tidak terdapat atribut *non primary key* yang bersifat *transitively dependent* pada *primary key*.

2NF ke 3NF

- a. Identifikasikan *primary key* dalam relasi 2NF
- b. Identifikasikan *functional dependencies* dalam relasi.
- c. Jika terdapat *transitive dependency* terhadap *primary key*, maka hapus dengan menempatkan dalam relasi yang baru bersama salinan determinannya.

#### 5. Boyce-code Normal Form (BCNF)

Berdasarkan pada *functional dependencies* yang dimasukkan dalam hitungan seluruh *candidate key* dalam suatu relasi. Bagaimanapun BCNF juga memiliki batasan-batasan tambahan disamakan dengan defenisi umum dari 3NF. Suatu relasi dikatakan BCNF, jika dan hanya jika setiap deteminan merupakan

*candidate key*. Perbedaan antara 3NF dan BCNF yaitu untuk *functional dependent*  $A - B$ , 3NF memungkinkan *dependency* ini dalam suatu relasi jika adalah atribut *primary key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menetapkan dengan jelas bahwa untuk *dependency* ini agar ditetapkan dalam relasi A, maka A harus merupakan *candidate key*. Setiap relasi dalam BCNF juga merupakan 3NF, tetapi relasi dalam 3NF belum tentu termasuk ke dalam BCNF. Dalam BCNF kesalahan jarang sekali terjadi, kesalahan dapat terjadi pada relasi yang

- a. Terdiri atas 2 atau lebih *composite candidate key*
- b. *Candidate key overlap*, sedikitnya satu atribut (Indrajani : 2015 ; 9-10).