

BAB II

TINJAUAN PUSTAKA

II.1. Teori Sistem

Menurut Tata Sutabri (2005:2), Suatu Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling bergantung satu sama lain, dan terpadu. Secara garis besar terdapat dua kelompok pendekatan yang dilakukan yaitu:

1. Pendekatan sistem yang lebih menekan pada elemen-elemen atau kelompoknya, didalam hal ini sistem ini didefenisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. Prosedur didefenisikan sebagai urutan operasi kerja (tulis-menulis), yang bisanya melibatkan beberapa orang didalam satu atau lebih departement yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefenisikan sistem sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan sistem dapat tercapai.

II.1.1. Karakteristik Sistem

Menurut Tata Sutabri (2005:11), sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain :

1. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupaya suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batasan sistem (*Boundary*)

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

3. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun 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*)

Merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukkan sistem (*input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukkan dapat berupa masukkan perawatan (*maintenance input*) dan masukkan sinyal (*signal input*).

6. Keluaran sistem (*output*)

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

7. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya, suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.2. Klasifikasi Sistem

Menurut Tata Sutabri (2005:13), sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya sebagai berikut:

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. misalnya, sistem teknologi yang berisi gagasan tentang hubungan manusia dan Tuhan.

Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia).

Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tertentu

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat.

Misalnya, sistem komputer. Sistem tak tertentu adalah sistem yang tidak dapat diramal dengan pasti karena mengandung unsur probabilitas.

Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungannya. Misalnya, reaksi kimia dalam tabung terisolasi.

Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan diperbaharui oleh lingkungan.

II.2. Konsep Dasar Informasi

Tata sutabri (2005:23), Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan sistem mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya. Teori informasi lebih tepat disebut teori matematik, komunikasi yang juga memberikan beberapa pandangan yang berguna bagi sistem informasi. Sumber sistem informasi adalah data, data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata merupakan suatu bentuk yang masih mentah yang belum menghasilkan suatu informasi.

II.2.1. Kualitas Informasi

Informasi yang berkualitas memiliki 3 kriteria, yaitu:

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.

Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*Timelines*)

Informasi yang datang pada sipenerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan dalam pengambilan keputusan.

3. Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakaian. Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas.

II.3. Konsep Dasar Sistem Informasi

Tata Sutabri (2005:41), Sistem Informasi adalah suatu sistem didalam sistem suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak tertentu dengan laporan-laporan yang diperlukan. Defenisi umum sistem informasi adalah “ Sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan”.

II.3.1. Komponen Sistem Informasi

Tata Sutabri (2005:42), dalam suatu sistem informasi terdapat komponen-komponen yang saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Blok Masukkan (*input block*)

Input mewakili data yang masuk kedalam sistem informasi. Input termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basisdata dengan cara sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*output Block*)

Merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakaian sistem.

4. Blok teknologi (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian diri dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan diprangkat keras komputer menggunakan perangkat lunak untuk memanipulasi.

6. Blok kendali (*control block*)

Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem seperti, kecurangan-kecurangan, kegagalan sistem itu sendiri, debu dan air, dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung diatasi.

II.4. Sistem Pendukung Keputusan (SPK)

Sistem pendukung keputusan (SPK) biasanya dibangun untuk mendukung solusi atau 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 dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur (Dicky Nofriansyah : 2014)

II.4.1. Pengertian Sistem Pendukung Keputusan

Menurut Bonczek dkk, (1980) dalam buku "*Decission Support System and Intelligent System*" (Turban 2005:137) mendefenisikan Sistem pendukung keputusan (SPK) 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 (repository pengetahuan domain masalah yang ada pada Sistem pendukung keputusan (SPK) 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).

II.4.2. Karakteristik Sistem Pendukung Keputusan

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

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan.
2. Adanya *interfce* manusia/mesin dimana manusia (*user*) tetap memegang kontrol proses pengambilan keputusan .
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki sub sistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.

II.4.3. Ciri – ciri Sistem Pendukung Keputusan

Menurut Dicky Nofriansyah (2014:2), kriteria atau ciri - ciri sistem pendukung keputusan (SPK) yaitu :

1. Banyak pilihan / alternatif.
2. Ada kendala / surat.
3. Mengikuti suatu pola atau model tingkah laku, baik yang terstruktur maupun tidak terstruktur.
4. Banyak input / variabel.
5. Ada faktor resiko, dibutuhkan kecepatan, ketepatan dan keakuratan.

II.5. Minyak Mentah Kelapa Sawit

Tanaman kelapa sawit (*Elaeis guinensis*) dapat menghasilkan dua jenis minyak, yakni minyak kelapa sawit mentah (CPO) yang diekstraksi dari mesokarp buah kelapa sawit, dan minyak inti kelapa sawit (PKO) yang diekstraksi dari biji atau inti kelapa sawit. Bagian yang paling utama untuk diolah dari kelapa sawit adalah buahnya, biasanya disebut Tandan Buah Segar(TBS). Bagian daging buah menghasilkan minyak kelapa sawit mentah (CPO) melalui proses ekstraksi. Buah diproses dengan membuat lunak bagian daging buah dengan temperature 90°C. Daging yang telah melunak dipaksa untuk berpisah dengan bagian inti dan cangkang untuk berpisah dengan bagian inti dan cangkang dengan *pressing* pada mesin slinder berulang.

Minyak kelapa sawit mentah hasil proses ekstraksi tersebut masih mengandung bahan ikatan seperti kadar asam lemak bebas, kadar air, kadar kotoran dan bau. Biasanya proses ekstraksi minyak kelapa sawit ini dilanjutkan

dengan proses *bleaching* (pemutihan) dan *deodorized* (penghilang bau) agar minyak tersebut menjadi jernih, bening dan tidak berbau, yang biasa disebut *refined, bleaching and deodorized* (RBD) atau disebut juga proses penyulingan. Dan proses penyulingan CPO memerlukan adsorben (zat padat) untuk menyerap kotoran dalam CPO (Yustina dkk : 2014).

II.6. Metode TOPSIS (*Technique For Order Preference By Similarity To Ideal Solition*)

Pada dasarnya, proses pengambilan keputusan adalah memilih suatu alternatif. Menurut Hwang dan Zeleny dalam (Kusumadewi, 2006), *TOPSIS* (*Technique For Order Preference By Similarity To Ideal Solition*) didasarkan pada konsep dimana alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif (A+) namun juga memiliki jarak terpanjang dari solusi ideal negatif (A-).

Konsep dari *TOPSIS* yang tidak hanya mencari jarak terpendek dari A+ tetapi juga jarak terpanjang dari A-, diharapkan satu-satunya kemungkinan solusi ideal terbaik. Karena apabila dicari jarak terpanjang dari A+ dari jarak terpendek dari A- alternatif keputusan terbaik akan dihasilkan.

Menurut Hwang, Liang dan Yeh dalam (Kusumadewi, 2006) konsep ini banyak digunakan pada beberapa model MAD (*Multi-Attribute Decision Making*) untuk menyelesaikan masalah keputusan secara praktis. Hal ini disebabkan konsepnya sederhana, mudah dipahami, komputasinya efisien, dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana (Asep Hendar Rustiawan : 2010).

Secara umum, prosedur atau langkah-langkah dalam metode TOPSIS (*Technique For Order Preference By Similarity To Ideal Solution*) meliputi :

1. Membuat matriks keputusan yang ternormalisasi.
2. Membuat matriks keputusan yang ternormalisasi terbobot.
3. Membuat matriks solusi ideal positif dan matriks solusi ideal negatif.
4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif.
5. Menentukan nilai nilai preferensi untuk setiap alternatif.

TOPSIS membutuhkan rating kinerja setiap alternatif A_i pada setiap kriteria C_j yang ternormalisasi, yaitu :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad \dots \dots \dots (1)$$

dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

Solusi ideal positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan rating bobot ternormalisasi (y_{ij}) sebagai:

$$y_{ij} = w_i r_{ij} \quad \dots \dots \dots (2)$$

dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

$$A^+ = (y_{1^+}, y_{2^+}, \dots, y_{n^+}); \quad \dots \dots \dots (3)$$

$$A^- = (y_{1^-}, y_{2^-}, \dots, y_{n^-}); \quad \dots \dots \dots (4)$$

dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$

Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan sebagai:

$$D_i^+ = \sqrt{\sum_{i=1}^n (y_i^+ - y_{ij})^2}; \quad \dots \dots \dots (5)$$

Dengan $i = 1, 2, \dots, m$

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_{i^-})^2}; \quad \dots\dots\dots (6)$$

Dengan $i = 1, 2, \dots, m$

Nilai dari preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}; \quad \dots\dots\dots (7)$$

dengan $i = 1, 2, \dots, m$

Nilai V_i yang lebih besar menunjukkan bahwa alternatif A_i lebih dipilih.

II.7. Microsoft Visual Studio 2010

Platform Microsoft.Net merupakan model untuk *development* dimana *platform* dan aplikasi bias dibuat dan dijalankan tanpa bergantung pada alat (*device*) yang dipakai. Teknologi ini memungkinkan beberapa aplikasi bekerja sama. *Visual Basic.Net 2010* merupakan *core* dari pembuatan aplikasi berbasis *.Net*. yang merupakan lingkungan pemrograman yang mempermudah tahapan desain, *development*, *debugging*, dan *deployment* dari aplikasi berbasis *.Net* dan *XML web service* meningkatkan efisiensi *developer* dengan menyediakan lingkungan pemrograman yang sudah biasa digunakan (Suharli, 2005:13). *.NET Framework* adalah teknologi inti yang menyediakan berbagai *library* untuk digunakan oleh aplikasi di atasnya. Komponen inti *.NET Framework* adalah *Common Language Runtime (CLR)* yang menyediakan run time environment untuk aplikasi yang dibangun menggunakan *Visual Studio .NET*, terlepas dari jenis bahasa pemrogramannya (Benardo, dkk; Jurnal Media Informatika Vol 11 No. 1 ; 2015).

II.8. Microsoft SQL Server 2008

SQL Server merupakan suatu *Relational Database Management Systems* (RDBMS) yang digunakan untuk menyimpan data. Data yang disimpan pada *database* bisa dalam skala kecil maupun besar. Selain itu, penyajiannya merupakan penyajian pada level fisik karena kita akan menyimpan langsung data pada *database* dengan kondisi yang sebenarnya, yaitu disimpan pada *table* apa, kolom mana, dan menggunakan tipe data saat penyimpanan.

Database merupakan suatu tempat untuk menyimpan data. Pada sebuah *database* biasa terdapat satu atau lebih tabel dan *query*. Operasi yang biasanya dilakukan *database* berhubungan erat dengan pengaksesan tabel atau *query*. Pada *SQL Server 2008* terdapat fitur-fitur yang dapat mengembangkan performa dari *database* tersebut (Benardo, dkk: 2015). Beberapa fitur tersebut, yaitu :

1) *Date Data Type*:

Digunakan untuk menyimpan data tanggal saja sehingga akan menghemat space pada server.

2) *Data Compression*:

Digunakan untuk melakukan compress data sehingga ukuran data yang disimpan dalam hal *space hardisk* akan lebih kecil.

3) *Sparse Column*:

Digunakan untuk menyimpan data yang memiliki lebih banyak data NULL dengan lebih efisien.

4) *Row Constructor*:

Digunakan untuk melakukan insert beberapa data sekaligus dengan satu perintah INSERT.

5) *Table-Valued Parameter*:

Digunakan untuk melakukan parsing array pada bahasa pemrograman, dimana satu variable diberikan data-data yang akan diproses setelahnya.

II.8.1 Konsep Database

Pangkalan data atau basis data, atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system, DBMS*). Sistem basis data dipelajari dalam ilmu informasi.

Database merupakan suatu tempat untuk menyimpan data. Pada sebuah *database* bisa terdapat satu atau lebih tabel dan *query*. Operasi yang biasanya dilakukan *database* berhubungan erat dengan pengaksesan tabel atau *query*. Selain itu juga dalam mengambil data dari *server* lain akan mengalami penurunan performa. Tetap dengan menggunakan terdistribusi, bisa dengan cepat melakukan akses untuk data pada *database server* yang didistribusikan. Sedangkan untuk tersentralisasi, karena databasenya hanya satu dan terpusat (misalnya di *head office*) maka seluruh client dari manapun akan mengambil data tersebut dari satu *database*. Dengan demikian data yang diambil tidak akan bermasalah dalam hal

konsistensi karena berada dalam satu sumber, tetapi akan membutuhkan hardware yang jauh lebih besar dan *bandwidth* yang lebih tinggi. Hal ini dikarenakan *server* tersebut berfungsi untuk menampung penggunaan *connection* yang sangat banyak.

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update yang rumit. Untuk penyimpanan *database*, biasanya digunakan relational *database*, yaitu suatu mekanisme penyimpanan data pada suatu *table* tertentu yang terhubung antara tabel yang satu dengan tabel lainnya dengan menggunakan references data. Data tersebut berupa *field* atau kolom pada tabel yang menghubungkan tabel yang satu dengan tabel yang lain (Benardo, dkk: 2015).

II.9. Unified Modelling Language (UML)

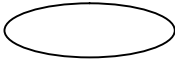



Menurut (Windu Gata ; 2013 : 4) hasil pemodelan pada OOAD (*Object Oriented Analysis And Design*) terdokumentasikan dalam bentuk *Unified Modeling Language*(UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem

berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut (Ade Hendini : 2016).

a. *Usecase Diagram*

Usecase diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Ade Hendini : 2016).

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



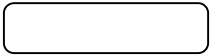
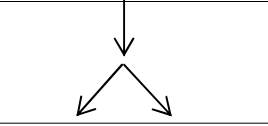
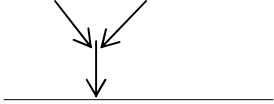
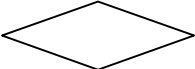

	panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
----->	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Ade Hendini : 2016)

b. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Ade Hendini : 2016).

Tabel II.2. Simbol Activity Diagram

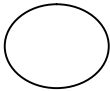
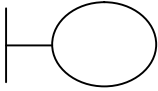





Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel 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</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Ade Hendini : 2016)

c. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Ade Hendini : 2016).

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 : Ade Hendini : 2016)

d. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem (Ade Hendini : 2016).

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
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 : Ade Hendini : 2016)