

## **BAB II**

### **LANDASAN TEORI**

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

Sistem pakar (*expert system*) adalah salah satu teknik kecerdasan buatan yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan sistem pakar ini, masyarakat awam pun dapat menyelesaikan masalah yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman (Sri Kusumadewi; 2003: 109).

Ada beberapa definisi tentang sistem pakar antara lain :

1. Menurut Durkin dalam buku Sri Kusumadewi: 2003:109). Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar.
2. Menurut Ignizio dalam buku Sri Kusumadewi: 2003:109). Sistem pakar adalah suatu model dan prosedur yang berkaitan, dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.

3. Menurut Giarratano dan Riley dalam buku Sri Kusumadewi: (2003:109). Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar.

### **II.1.1 Konsep Dasar Sistem Pakar**

Konsep dasar sistem pakar mengandung keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan. Keahlian adalah suatu kelebihan penguasaan pengetahuan dibidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan, menyusun kembali pengetahuan jika diapandang perlu, memecahkan aturan – aturan jika dibutuhkan, dan menentukan relevan tidaknya keahlian mereka. Pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan 4 aktivitas yaitu : tambahan pengetahuan (dari para ahli atau sumber – sumber lainnya), representasi pengetahuan (komputer), inferensi pengetahuan dan pengalihan pengetahuan ke user. Pengetahuan yang disimpan di komputer disebut basis pengetahuan. Ada 2 tipe pengetahuan, yaitu : fakta dan prosedur (biasanya berupa aturan).

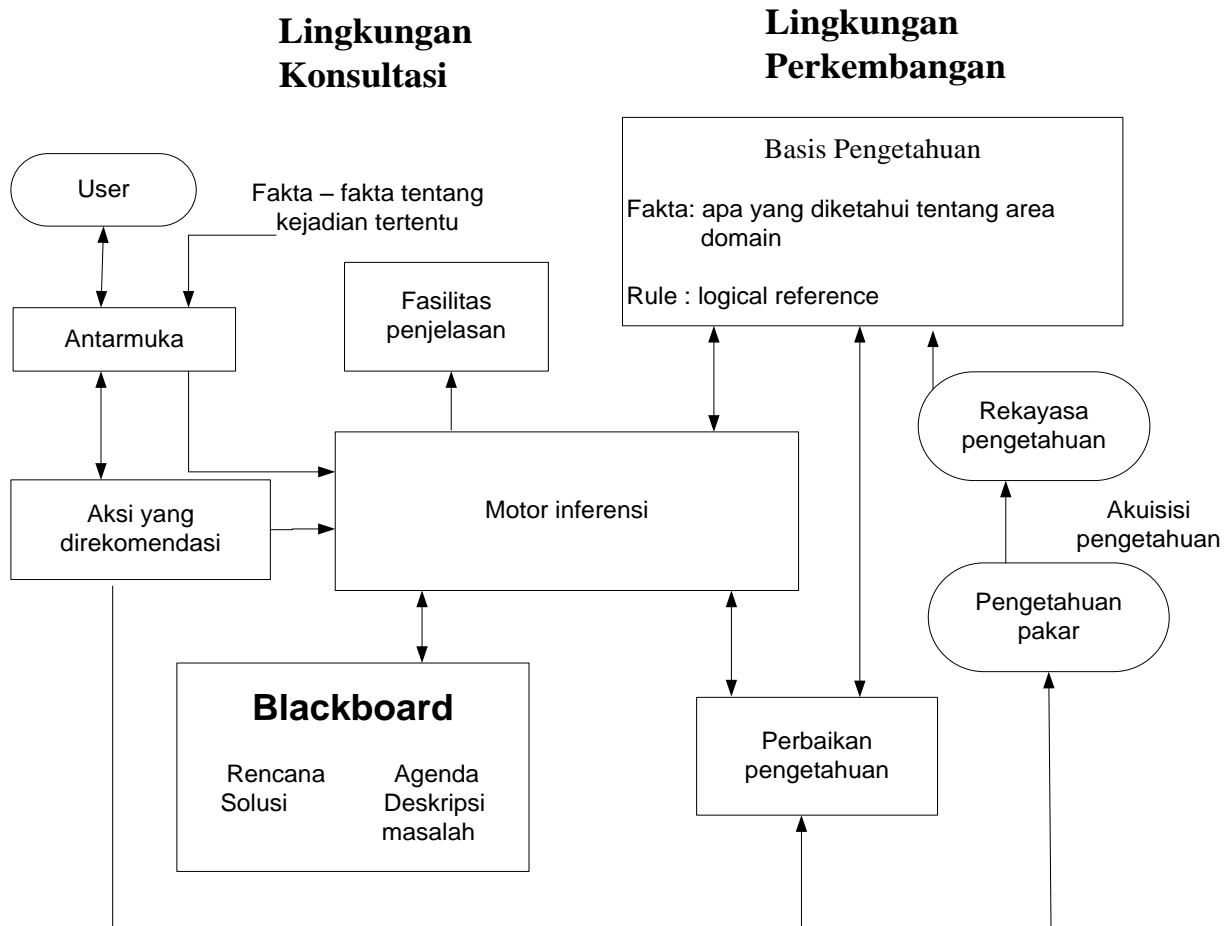
Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar. Jika keahlian – keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basis data, maka komputer harus dapat deprogram untuk membuat inferensi. Proses inferensi

ini dikemas dalam bentuk inferensi (*inference engine*). Sebagian besar sistem pakar komersial dibuat dalam bentuk *rule based systems*, yang nama pengetahuan disimpan dalam bentuk aturan – aturan. Aturan tersebut biasanya berbentuk IF-THEN (Sri Kusumadewi; 2003: 109).

### **II.1.2 Struktur Sistem Pakar**

Sistem pakar terdiri dari dua bagian pokok, yaitu lingkungan pengembangan (*development envirointment*) dan lingkungan konsultasi (*consultation envirointment*). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi.

Komponen-komponen sistem pakar dalam kedua bagian tersebut terlihat dalam gambar II.1 :



**Gambar II.1 : Struktur Sistem Pakar**

**Sumber : (Sri Kusumadewi; 2003: 114)**

Keterangan :

1. Subsystem penambahan pengetahuan. Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu bisa berasal dari: ahli, buku, basisdata, penelitian, dan gambar.
2. Basis pengetahuan. Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

3. Motor inferensi (*inference engine*). Program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan blackboard, serta digunakan untuk memformulasi konklusi. Ada 3 elemen utama dalam motor inferensi yaitu:
  - a. Interpreter : mengeksekusi item-item agenda yang terpilih dengan menggunakan aturan-aturan dalam basis pengetahuan yang sesuai.
  - b. Scheduler : akan mengontrol agenda.
  - c. Consistency enforcer: akan berusaha memelihara kekonsistenan dalam merepresentasikan solusi yang bersifat darurat.
4. Blackboard. Merupakan area dalam memori yang digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara. Ada 3 tipe keputusan yang dapat direkam, yaitu :
  - a. Rencana : bagaimana menghadapi masalah.
  - b. Agenda : aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi.
  - c. Solusi : calon aksi yang akan dibangkitkan.
5. Antarmuka. Digunakan untuk media komunikasi antara user dan program.
6. Subsistem penjelasan. Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:
  - a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
  - b. Bagaimana konklusi dicapai?
  - c. Mengapa ada alternatif yang dibatalkan?

- d. Rencana apa yang digunakan untuk mendapatkan solusi?
7. Sistem penyaring pengetahuan. Sistem ini digunakan untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa mendatang.

### II.1.3 Ciri – Ciri Sistem Pakar

Sistem pakar yang baik harus memenuhi ciri-ciri sebagai berikut :

1. Memiliki fasilitas informasi yang handal.
2. Mudah dimodifikasi.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki kemampuan untuk belajar beradaptasi.

### II.2. Teorema Bayes

Probabilitas *Bayes* merupakan salah satu cara untuk mengatasi ketidakpastian data dengan menggunakan formula *Bayes* yang dinyatakan dalam rumus sebagai berikut :

$$P(H | E) = \frac{P(E | H) \cdot P(H)}{P(E)}$$

Keterangan :

$P(H | E)$  : probabilitas hipotesis H jika diberikan *evidence* E

$P(E | H)$  : probabilitas munculnya *evidence* apapun

$P(E)$  : probabilitas *evidence* E

Dalam bidang kedokteran teorema *Bayes* sudah dikenal tapi teorema ini lebih banyak diterapkan dalam logika kedokteran modern. Teorema ini lebih

banyak diterapkan pada hal-hal yang berkenaan dengan probabilitas serta kemungkinan dari penyakit dan gejala-gejala yang berkaitan. (Sri Kusumadewi; 2003: 96):

### **II.3. Centripugal Pump**

Centripugal pump adalah suatu alat / mesin untuk memindahkan cairan dengan memanfaatkan gaya sentrifugal yang dihasilkan oleh putaran impeller. Pompa sentrifugal mengubah energi kecepatan menjadi energi tekanan. Ada juga yang menyebutkan sebagai mesin kecepatan karena semakin cepat putaran pompanya maka akan semakin tinggi tekanan yang dihasilkan.

Keuntungan dan kerugian pompa sentripugal :

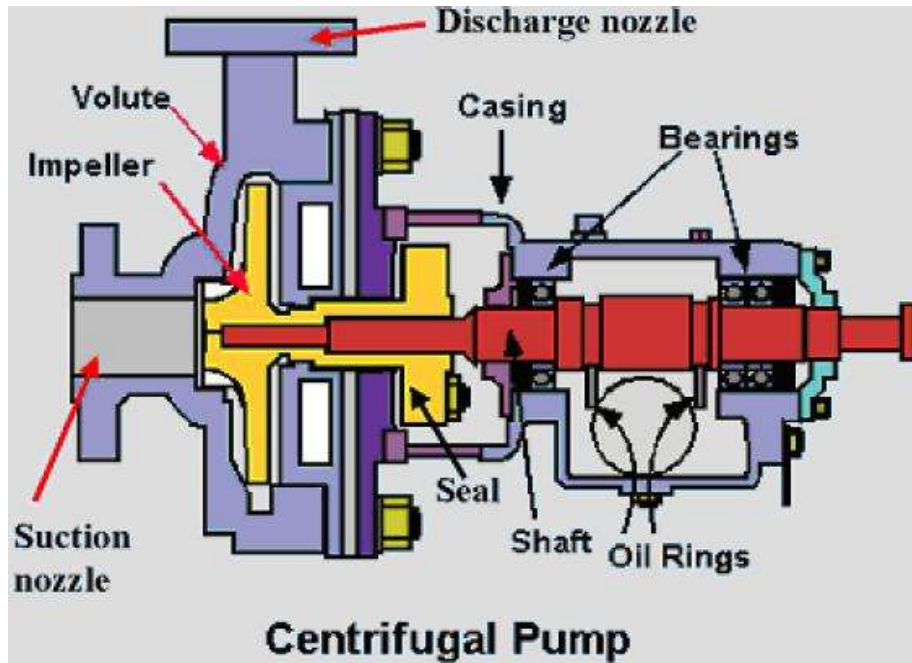
Keuntungan :

- Merupakan jenis yang paling umum / banyak digunakan, Kontruksinya sederhana, Operasinya handal, Harganya murah, Kapasitasnya besar, Efisiensinya bagus, Dapat digunakan untuk suhu tinggi.

Kerugiannya :

- Tidak cocok untuk cairan yang viskositasnya rendah, Tidak *self priming* , walaupun dengan desain khusus dapat dibuat menjadi *self priming*, Tidak cocok untuk kapasitas yang kecil.

Adapun bagian – bagian dari centripugal pump dapat dilihat pada gambar II.2. centripugal pump sebagai berikut:



**Gambar II.2. Centripugal pump**

**Sumber PT. Musim Mas KIM II Medan**

Fungsi dari setiap bagian tersebut yaitu:

1. *Impeller* : suatu bagian terpenting didalam pompa centripugal karena berfungsi sebagai pemindah material dari *suction nozzle* ke *discharge nozzle*
2. *Volute* : berfungsi sebagai wadah dimana material akan melalui alur *volute* untuk ke *discharge nozzle* yang diputarakan oleh *impeller*
3. *Bearings* : berfungsi sebagai penyeimbang shaft agar putaran pompa dapat stabil
4. *Shaft* : tanpa adanya *shaft* suatu pompa tidak akan terbentuk karena semua bagian menempel di area *shaft*.

5. Seal : berfungsi sebagai perisai agar material tidak keluar dari chasing pompa yang mengakibatkan kebocoran.
6. Oil Rings : berfungsi sebagai pengaduk oli untuk melumasi area *bearing* sehingga *bearing* tidak mudah rusak.
7. Suction Nozzle : dimana jalur awal material dimulai
8. Discharge Nozzle : akhir jalur material setelah pemindahan material

#### **II.4. Pengertian Visual Basic**

Visual Basic dibuat Microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, Visual Basic juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi *database*. Visual Basic Merupakan bahasa perograman *event drive*, di mana program akan menunggu samapai respons dari *user/pemakai* program aplikasi yang dapat berupa kejadian atau *event*, misalnya ketika *user* mengklik tombol tau menekan enter. Jika kita membuat aplikasi dengan Visual Basic maka kita akan mendapatkan file yang menyusun aplikasi tersebut, yaitu :

1. File Project (\*.vbp)

File ini merupakan kumpulan dari aplikasi yang kita buat, file project bisa berupa file\*.frm,\* .dsr atau file lainnya.

2. File From (\*.frm)

File ini merupakan file yang berfungsi untuk menyimpan informasi tentang bentuk *form* maupun *interface* yang kita buat.

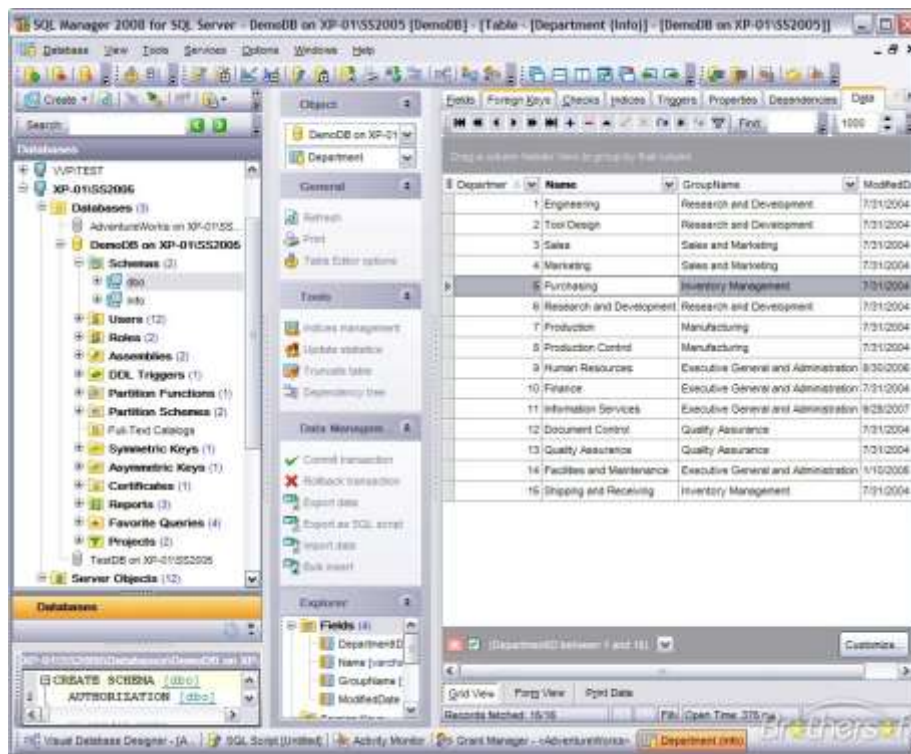
Untuk menjalankan Visual Basic sanagt mudah :

1. Dari menu Start, pilih program selanjutnya pilih Microsoft Visual Basic 2010 jika anda menggunakan Visual Basic Versi Microsoft Visual Basic 2010
2. Ketika pertama kali dijalankan maka akan muncul kotak dialog yang memiliki 3 tabulasi yaitu :
  - a. *New*, digunakan untuk membuat project baru
  - b. *Existing*, untuk membuka project yang pernah kita buat.

*Recent*, untuk membuka project yang pernah kita buka.(Edy Winarno ; 2010 : 83).

## **II.5. Pengertian SQL Server**

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-object yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, funcation, stored procedure, trigger* dan *view*. Selain *object*, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data. (*Cybertron Solution ; 2010 : 101-102*).



**Gambar II.2. Tampilan SQL Server**  
 (Sumber : Cybertron Solution ; 2010 : 101-102)

## II.6. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

## II.6.1. Bentuk-bentuk Normalisasi

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

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

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

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

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

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

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

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF

sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

## **5. Bentuk Normal Tahap Keempat dan Kelima**

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

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

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


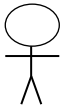

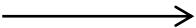
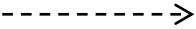
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

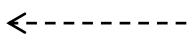
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

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 di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu :

**Tabel II.2. 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



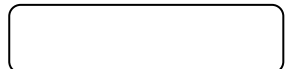
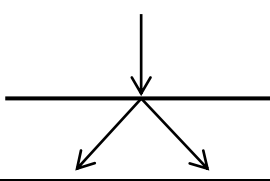
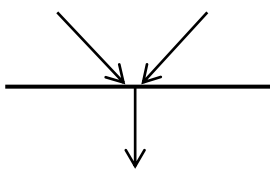
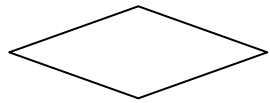

	program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ; 2013 : 4)

## 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.3. Simbol *Activity Diagram***

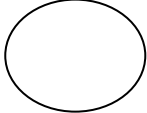
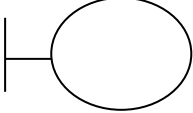
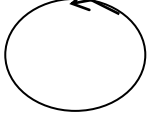

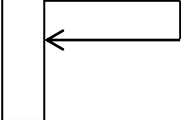

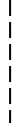
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan 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 : Windu Gata ; 2013 : 6)

### 3. Diagram Urutan (*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.4. 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 : Windu Gata ; 2013 : 7)

#### 4. *Class Diagram* (Diagram Kelas)

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.

*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*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.5. *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 : Windu Gata ; 2013 : 9)