

## **BAB II**

### **LANDASAN TEORI**

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

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem.

Karakteristik sistem terdiri dari :

##### 1. Komponen Sistem

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

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

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

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

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

#### 6. Keluaran Sistem

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 (Jeperson Hutahaean ; 2014 : 2).

## **II.2. Sistem Pakar**

Sistem pakar menirukan perilaku seorang pakar dalam menangani suatu persoalan. Pada suatu kasus seorang pasien mendatangi dokter untuk memeriksa badannya yang mengalami penyakit kesehatan, maka dokter atau pakar kesehatan akan memeriksa dan melakukan diagnosa. Bila dokter cukup sibuk dan pelaksana diagnosa digantikan oleh sebuah sistem pakar, maka sistem pakar diharapkan dapat membantu memahami dan menganalisa keadaan pasien dan menemukan penyakit yang diderita pasien itu. Sistem pakar diharapkan juga untuk menghasilkan dugaan atau hasil diagnosa yang sama dengan diagnosa yang dilakukan oleh seorang ahli. Tujuan utama sistem pakar bukan untuk menggantikan kedudukan seorang ahli maupun pakar, tetapi untuk memasyarakatkan pengetahuan dan pengalaman pakar-pakar yang ahli di bidangnya (Andri Saputra ; 2011 : 203).

### **II.2.1. Konsep Dasar Sistem Pakar**

Konsep dasar sistem pakar mengandung : keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan. Keahlian adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang termasuk keahlian adalah :

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.

3. Prosedur-prosedur dan aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
4. Strategi-strategi global untuk menyelesaikan masalah.
5. *Meta-knowledge* (pengetahuan tentang pengetahuan) (Andri Saputra, 2011 : 203)

### **II.2.2. Mesin Inferensi (*Inference Engine*) Sistem Pakar**

Inferensi digunakan dalam sistem pakar untuk memperoleh informasi terbaru dari informasi yang sudah ada. Di antaranya :

#### *1. Forward Chaining*

Adalah strategi inferensi yang dimulai dengan sekumpulan fakta, fakta baru yang diperoleh dengan menggunakan rule, dimana alasan yang digunakan sesuai dengan fakta yang ada, dan melanjutkan proses ini sampai goal diraih atau sampai tidak ada rule selanjutnya yang mempunyai alasan yang sesuai dengan fakta yang ada maupun fakta yang diketahui

#### *2. Backwad Chaining*

Adalah strategi inferensi yang diperoleh untuk membuktikan suatu hipotesis dengan dukungan informasi (Rizki Fitriani Maiza ; 2010 : 10).

### **II.2.3. Aktivitas Sistem Pakar**

Bentuk-bentuk ini memungkinkan para ahli untuk dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan ahli. Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan,

mempelajari hal-hal baru seputar topik permasalahan (domain), menyusun kembali pengetahuan jika dipandang perlu, memecah 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 :

1. Tambahan pengetahuan (dari para ahli atau sumber-sumber lainnya).
2. Representasi pengetahuan (ke komputer).
3. Inferensi pengetahuan.
4. Pengalihan pengetahuan ke user (Andri Saputra, 2011 : 203)

#### **II.2.4. Bentuk Sistem Pakar**

Pengetahuan yang disimpan di komputer disebut dengan nama 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 basisdata, maka komputer harus dapat diprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (inference engine). Sebagian besar sistem pakar komersial dibuat dalam bentuk rule-based systems, yang mana pengetahuannya disimpan dalam bentuk aturan-aturan. Aturan tersebut biasanya berbentuk IF-THEN. Fitur lainnya dari sistem pakar adalah kemampuan untuk merekomendasi. Kemampuan inilah yang membedakan sistem pakar dengan sistem konvensional.

Ada 4 bentuk sistem pakar, yaitu :

1. Berdiri sendiri. Sistem pakar jenis ini merupakan software yang berdiri-sendiri tidak tergantung dengan software yang lainnya.
2. Tergabung. Sistem pakar jenis ini merupakan bagian program yang terkandung didalam suatu algoritma (konvensional), atau merupakan program dimana didalamnya memanggil algoritma subrutin lain (konvensional).
3. Menghubungkan ke software lain . Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya DBMS.

Sistem Mengabdi. Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar (Andri Saputra, 2011 : 203).

#### **II.2.5. Struktur Sistem Pakar**

Sistem pakar disusun oleh dua bagian utama, yaitu :

1. Lingkungan pengembangan (*development environment*), digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar.
2. Lingkungan konsultasi (*consultation environment*), digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar (Andri Saputra, 2011 : 204).

### II.3. Penyakit Alzheimer

Penyakit alzheimer ditemukan pertama kali pada tahun 1907 oleh seorang ahli Psikiatri dan neuropatologi yang bernama Alois Alzheimer. Ia mengobservasi seorang wanita berumur 51 tahun, yang mengalami gangguan intelektual dan memori serta tidak mengetahui kembali ketempat tinggalnya, sedangkan wanita itu tidak mengalami gangguan anggota gerak, koordinasi dan reflek. Pada autopsi tampak bagian otak mengalami atrofi yang difus dan simetri, dan secara mikroskopik tampak bagian kortikal otak mengalami neuritis plaque dan degenerasi neurofibrillary.

Secara epidemiologi dengan semakin meningkatnya usia harapan hidup pada berbagai populasi, maka jumlah orang berusia lanjut akan semakin meningkat. Di lain pihak akan menimbulkan masalah serius dalam bidang sosial ekonomi dan kesehatan, sehingga akan semakin banyak yang berkonsultasi dengan seorang neurolog karena orang tua tersebut yang tadinya sehat, akan mulai kehilangan kemampuannya secara efektif sebagai pekerja atau sebagai anggota keluarga. Hal ini menunjukkan munculnya penyakit degeneratif otak, tumor, multiple stroke, subdural hematoma atau penyakit depresi, yang merupakan penyebab utama demensia. Istilah demensia digunakan untuk menggambarkan sindroma klinis dengan gejala menurunnya daya ingat dan hilangnya fungsi intelek lainnya.

Defenisi demensia menurut Unit Neurobehavior pada *Boston Veterans Administration Medical Center (BVAMC)* adalah kelainan fungsi intelek yang didapat dan bersifat menetap, dengan adanya gangguan paling sedikit 3 dari 5

komponen fungsi luhur yaitu gangguan bahasa, memori, visuospasial, emosi dan kognisi. Penyebab pertama penderita demensia adalah penyakit alzheimer (50- 60%) dan kedua oleh cerebrovaskuler (20%). Diperkirakan penderita demensia terutama penderita alzheimer pada abad terakhir ini semakin meningkat jumlah kasusnya sehingga akan mungkin menjadi epidemi seperti di Amerika dengan insidensi demensia 187 populasi /100.000/tahun dan penderita alzheimer 123/100.000/tahun serta penyebab kematian keempat atau kelima (Iskandar Japardi ; 2012 : 1).

### **II.3.1. Gejala Penyakit Alzheimer**

Berikut adalah 10 gejala utama penyakit Alzheimer :

1. Kehilangan memori bertahap. Adalah normal bila sesekali lupa dengan janji atau nama orang dan kemudian mengingatnya lagi. Penderita penyakit Alzheimer lebih sering lupa dengan hal-hal seperti itu dan tidak mengingatnya kembali, terutama bila baru terjadi. Masyarakat awam menyebutnya “pikun”.
2. Kesulitan melaksanakan tugas rutin. Gagal melakukan secara tuntas hal-hal yang biasa dilakukan seperti menyiapkan masakan, mengelola anggaran rumah tangga atau melakukan ibadah.
3. Gangguan penilaian. Kesulitan menilai situasi sehingga, misalnya, tidak mengakui masalah medis yang perlu perhatian atau mengenakan pakaian tebal di hari yang panas.

4. Disorientasi ruang dan waktu. Sering lupa dengan waktu dan mudah tersesat. Ketika berjalan, penderita tidak tahu bagaimana sampai di sana dan caranya pulang ke rumah.
5. Masalah dengan bahasa. Lupa dengan kata-kata sederhana atau kata-kata pengganti sehingga kalimatnya sulit dimengerti.
6. Masalah dengan pemikiran abstrak. Memiliki kesulitan yang signifikan dengan tugas-tugas yang bersifat abstrak seperti menghitung pengembalian uang atau membuat rencana tindakan.
7. Salah menaruh barang. Meletakkan sesuatu di tempat yang tidak semestinya seperti: baju di kulkas atau kacamata di cangkir kopi.
8. Perubahan suasana hati dan perilaku. Memerlihatkan perubahan suasana hati yang bervariasi dari tenang menjadi menangis atau marah tanpa alasan yang jelas.
9. Perubahan kepribadian. Mudah bingung, khawatir, curiga atau menarik diri dari pergaulan, bersikap apatis, atau bertindak di luar karakternya. Dua pertiga penderitanya mengalami gejala depresi dan sekitar 20% menunjukkan agresi, terutama pada pasien laki-laki. Mereka juga dapat mengalami halusinasi, delusi (keyakinan palsu) dan paranoid.
10. Kehilangan inisiatif. Menjadi sangat pasif dan memerlukan isyarat dan dorongan untuk terlibat (Iskandar Japardi ; 2012 : 14).

#### II.4. Metode Teorema Bayes

Probabilitas Bayes merupakan salah satu cara yang baik untuk mengatasi ketidakpastian data dengan menggunakan formula bayes yang dinyatakan dengan rumus :

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

Keterangan :

$P(H | E)$  : probabilitas (tingkat kemungkinan | P) hipotesis (kesimpulan sementara | H) jika diberikan evidence (fakta | 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.

Misalnya seseorang menjalani tes klinik tersebut dan mendapatkan hasil positif, berapakah peluang bahwa ia benar-benar menderita penyakit langka tersebut? Dengan kata lain, kita mencoba untuk mencari peluang dari A, dimana B atau  $P(A | B)$ .

Dari tabel di atas, dapat kita lihat bahwa  $P(A | B)$  adalah peluang dari positif yang benar dibagi dengan peluang positif (benar maupun salah), yaitu  $0,0194 / (0,0194 + 0,0882) = 0,1803$ . Kita dapat juga mendapatkan hasil yang sama dengan menggunakan rumus teorema Bayes di atas:

$$P(A | B) = \frac{P(B \cap A)}{\quad}$$

$$\begin{aligned}
 & P(B) \\
 = & \frac{P(B|A) \times P(A)}{P(B|A)P(A) + P(B|\bar{A})P(\bar{A})} \\
 = & \frac{97\% \times 2\%}{(97\% \times 2\%) + (9\% \times 98\%)} \\
 = & \frac{0.0194}{0.0194 + 0.0882} \\
 = & \frac{0.0194}{0.1076} \\
 P(A|B) = & 0.1803
 \end{aligned}$$

Hasil perhitungan ini sangat berbeda dengan intuisi kita di atas. Peluang bahwa orang yang mendapat hasil tes positif itu benar-benar menderita penyakit langka tidak sebesar yang kita bayangkan. Cuma ada sekitar 18% kemungkinan bahwa dia benar-benar menderita penyakit itu (Sri Rahayu ; 2013 : 131).

#### II.4.1. Langkah - Langkah Metode *Teorema Bayes*

Dalam melakukan perhitungan metode teorema bayes, maka diperlukan beberapa langkah – langkah dalam perhitungan metode teorema bayes seperti berikut:

1. Menentukan kriteria yang akan dinilai terlebih dahulu.
2. Kemudian menentukan data training.
3. Setelah data kriteria dan data training telah ditentukan, maka menentukan probabilitas dari setiap variabel.

4. Dari penilaian probabilitas maka akan dapat ditentukan nilai dari setiap karakter dan kriteria yang akan dinilai tersebut.
5. Kemudian dari nilai kriteria dapat diteruskan dengan mencari likelihood dan probabilitas dari masing – masing kriteria.
6. Adapun perhitungan mencari likelihood dan probabilitas dengan menggunakan rumus  $P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$
7. Kemudian dapat ditentukan nilai tertinggi dari setiap data.

## II.5. Database

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database* (Agustinus Mujilan ; 2012 : 23)

## II.6. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa inggris; *database management system*) data DBMS yang

*multithread*, MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya (Achmad Solichin ; 2010 : 2).

## **II.7. PHP**

Menurut Deni Sutaji (2012) PHP adalah kode atau skrip yang akan dieksekusi pada *server-side*. Skrip PHP akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML, sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server-side* berarti pengerjaan skrip dilakukan di *server* baru kemudian hasilnya dikirimkan ke *browser*

### **II.7.1. Macromedia Dreamweaver**

Macromedia dreamweaver adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman web. Dengan menggunakan Dreamweaver 8, ketika membuat sebuah halaman web, Anda tidak perlu lagi mengetik kodekode HTML atau kode-kode lainnya secara manual. Anda cukup melakukan klik beberapa kali, maka simsalabim, halaman web yang Anda inginkan sudah jadi.

Selain HTML, Dreamweaver 8 juga mendukung *CSS*, *JavaScript*, *PHP*, *ASP*, dan bahasa pemrograman lainnya untuk membuat web. Hal ini akan sangat menguntungkan Anda. Sebagai contoh, jika dahulu Anda harus mengetikkan kode-kode CSS untuk membuat *Style* tertentu, maka dengan Dreamweaver 8, Anda cukup melakukan klik beberapa kali saja. Dreamweaver 8 adalah versi terbaru dari keluarga Dreamweaver. Versi pertamanya sendiri diluncurkan sekitar tahun 1994 oleh Macromedia Inc. Dalam versi terbaru ini, banyak sekali fasilitas baru yang ditambahkan. Contohnya, Anda akan dapat membuat dan menggunakan *Style* dalam CSS dengan mudah dan fleksibel. Panel untuk pengolahan CSS juga sudah diperbarui dan lebih mudah digunakan. Dreamweaver 8 juga menyediakan beberapa template halaman web baru, termasuk *fasilitas Starter Pages*. Akhirnya, selamat memulai mempelajari Dreamweaver 8 (Arief Ramadhan ; 2011 : 2).

### **II.7.2. Pengertian Apache**

Apache adalah paket aplikasi yang digunakan untuk web server yang handal dan stabil. Jika dibandingkan dengan web server lainnya, Apache masih menjadi andalan para webmaster. Perkembangan server ini sangat pesat sehingga hampir semua server web menggunakan Apache. Aplikasi apache dikenal dengan nama httpd. Jika menggunakan linux redhat/Fedora maka paket ini sudah ada dalam bentuk .RPM (Albertus Dwiyoga ; 2012 : 11).

### **II.7.3. Pengertian CSS**

(CSS) *Cascading Style Sheets* adalah suatu bahasa stylesheet yang digunakan untuk mengatur style suatu dokumen. Pada umumnya CSS dipakai

untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML. CSS memungkinkan web developer untuk memisahkan HTML dari aturan-aturan untuk membentuk tampilan sebuah website. CSS diperkenalkan untuk pengembangan website pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi style yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan parent-child pada setiap style (Sulistiyawan ; 2012 : 32).

## **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 tabel 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).

## II.9. 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


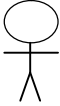
sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.


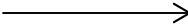
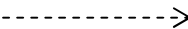
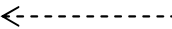
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

#### 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.1. Simbol *Use Case***

| Gambar  | Keterangan   |
|---|--|
|  | <p><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>.</p>  |
|  | <p>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</p> |



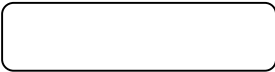
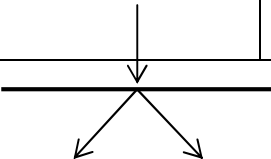
|   |  |
|---|--|
|   | beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .   |
|  | Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan 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.  |

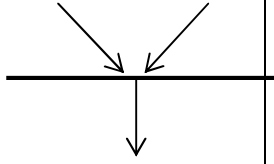
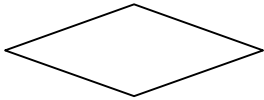

(Sumber : Windu Gata ; 2013 : 4)

## 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.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 |

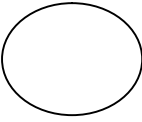
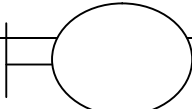
|   |  |
|---|--|
|   | parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.                                     |
|  | <i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.            |
|  | <i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> . |
|  | <i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.               |

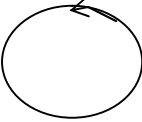
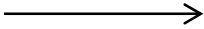
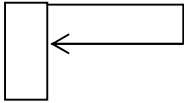

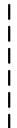
(Sumber : Windu Gata ; 2013 : 6)

### 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.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 : 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.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 : Windu Gata ; 2013 : 9)*