

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Hartatik dan Yasa (2015) mengenai Sistem Pakar Untuk Mendeteksi Hama Tanaman Jahe Menggunakan Teorema Bayes, Kesimpulan yang di dapat adalah teorema Bayes dapat digunakan untuk melakukandiagnosa hama yang menyerang tanaman jahe denganmerunut dari gejala yangdiinputkan oleh *user*. Perhitungan nilai probabilitas penyakit dilakukandengan menghitung gejala yang teramati oleh *user* dan nilai probabilitas yang diberikan oleh pakar padamasing-masing penyakit atau hipotesa.

Berdasarkan penelitian yang dilakukan oleh Jusniwati (2013) mengenai Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Tulang Dengan Menggunakan Metode Bayes, berdasarkan dari hasil penelitian yang sudah dilakukan, kesimpulan yang dapat diambil adalah Dalam mendiagnosa penyakit tulang menggunakan analisa metode bayes, yang berdasarkan pada pengumpulan data yang diberikan, maka tujuan yang diharapkan atau kesimpulan yang akan diperoleh lebih optimal.

II.2. Sistem Pakar

Sistem pakar (*Expert System*) mulai dikembangkan pada pertengahan tahun 1960-an oleh *Artificial Intelligence corporation*. Sistem pakar yang muncul pertama kali adalah *General-Purpose Problem Solve (GPS)* yang

merupakan sebuah *Predecessor* untuk menyusun langkah-langkah yang dibutuhkan untuk mengubah situasi awal menjadi state tujuan yang telah ditentukan sebelumnya dengan menggunakan domain masalah yang kompleks. (Jusniwati, 2013 : 2).

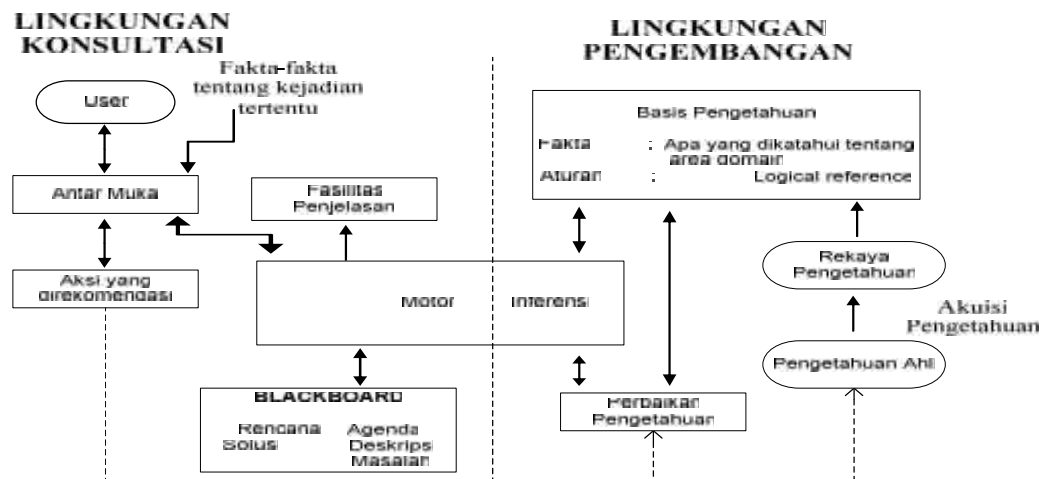
Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti yang layaknya seorang pakar. Selanjutnya, menurut Martin dan Oxman, sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Penyusunannya sistem pakar harus melakukan kombinasi terhadap kaidah-kaidah untuk menarik kesimpulan atau *inference rules* dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu. (Belutowe, 2015 : 52).

II.2.1. Struktur Sistem Pakar

Ada dua bagian dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*).

Lingkungan pengembangan digunakan oleh pembuatan sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi sehingga

pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Adapun struktur pada sistem pakar dapat dilihat pada gambar II.1. sebagai berikut :



Gambar II.1. Struktur Sistem Pakar

Sumber : (Mhd Ihsan, dkk : 2017 : 129)

Keterangan :

1. Akuisisi pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya kedalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber pengetahuan bisa diperbolehkan dari pakar, buku, dokumen, multimedia, basisdata. Laporan riset khusus, dan informasi yang terdapat di *web*.

2. Basis pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar, yaitu :

- a. Fakta, misalnya situasi, kondisi, untuk permasalahan yang ada.
- b. Aturan (*rule*), untuk mengarahkan penggunaan pengetahuan dalam memecahkan masalah.

3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arahan dalam melakukan proses penalaran. Ada tiga teknik pengendalian yang digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik tersebut.

4. Daerah Kerja (*Blackboard*)

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi, sistem pakar membutuhkan *blackboard* yaitu area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard*, yaitu :

- a. Rencana : bagaimana menghadapi masalah.
- b. Agenda : aksi-aksi potensial yang sedang menunggu untuk dieksekusi.
- c. Solusi : calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar, komunikasi ini paling bagus bila disajikan dalam bahasa alami (*naturallanguage*) dan dilengkapi dengan grafik, menu, dan formulir elektronik. Pada bagian ini akan terjadi dialog antara sistem pakar dan pengguna.

6. Subsistem Penjelasan (*Explanation Subsystem/Justifire*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

7. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)

Kemampuan memperbaiki pengetahuan (*knowledge refining system*) dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu. Kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa yang akan datang. Kemampuan evaluasi diri seperti itu diperlukan oleh program agar menganalisa alasan-alasan kesuksesan dan kegagalannya dalam mengambil

kesimpulan. Dengan cara ini basis pengetahuan yang lebih baik dan penalaran yang lebih efektif akan dihasilkan.

8. Pengguna (*User*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dan berbagai permasalahan yang ada.

II.3. Penyakit Vitiligo

Vitiligo adalah gangguan depigmentasi idiopatik didapat yang ditandai dengan gambaran makula putih tidak bersisik yang merupakan hasil dari hancurnya melanosit kulit secara selektif. Studi kasus ini bertujuan untuk mendiagnosis vitiligo dengan akurat sehingga dapat melakukan penatalaksanaan dengan tepat. Pasien seorang laki-laki berusia 25 tahun mengeluh terdapat perubahan warna pada kedua tangan dan kaki sejak 4 tahun yang lalu yang semakin melebar. Perubahan warna tanpa disertai gatal, nyeri, atau mati rasa. (Herperian, 2016 : 1).

II.4. Metode Teorema Bayes

Teorema Bayes merupakan kaidah yang memperbaiki atau merevisi suatu probabilitas dengan cara memanfaatkan informasi tambahan. Maksudnya dari probabilitas awal (*prior probability*) yang belum diperbaiki yang dirumuskan berdasarkan informasi yang tersedia saat ini, kemudian dibentuklah probabilitas berikutnya. (Hartatik, dkk, 2015 : 28).

Teorema Bayes merupakan teorema yang kebanyakan digunakan untuk menghitung peluang-peluang dalam suatu hipotesis tertentu, Teorema Bayes pertama kali diperkenalkan oleh ilmuwan yang bernama Bayes yang ingin memastikan tentang dimana keberadaan Tuhan. Bayes mencari fakta tentang keberadaan Tuhan di dunia dan kemudian mengubahnya dengan nilai probabilitas dan yang akan dibandingkan dengan nilai probabilitas. Theorema Bayes ini juga merupakan dasar dari statistika Bayes yang memiliki penerapan dalam ilmu ekonomi mikro, sains, teori permainan dan kedokteran. (Belutowe, 2015 : 51).

Rumus metode Teorema Bayes adalah sebagai berikut :

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

Keterangan :

$P(A|B)$ = Peluang kejadian A bila B terjadi

$P(B|A)$ = Peluang kejadian B bila A terjadi

$P(A)$ = Probabilitas A

$P(B)$ = Probabilitas B

Kekurangan pada metode Teorema bayes adalah sebagai berikut :

1. Metode Bayes hanya bisa digunakan untuk persoalan klasifikasi yang data-datanya bersifat kategorikal dan dengan metode pembelajaran terawasi.

2. Diperlukan pengetahuan awal bagi Metode Bayes untuk dapat mengambil suatu keputusan. Tingkat keberhasilannya sangat tergantung pada pengetahuan awal yang diberikan. (Belutowe, 2015 : 51).

Studi Kasus :

Reva melakukan diagnosa penyakit vitiligo dengan menjawab pertanyaan sesuai dengan gejala berikut :

$$G1 = 0.9 = P(E|H1)$$

$$G2 = 0.7 = P(E|H2)$$

$$G3 = 0.8 = P(E|H3)$$

$$G4 = 0.5 = P(E|H4)$$

$$G5 = 0.7 = P(E|H5)$$

$$G6 = 0.6 = P(E|H6)$$

$$G7 = 0.5 = P(E|H7)$$

Kemudian mencari nilai semesta dengan menjumlahkan dari hipotesa di atas :

$$k=1$$

$$= G1 + G2 + G3 + G4 + G5 + G6 + G7$$

$$= 0.9 + 0.7 + 0.8 + 0.5 + 0.7 + 0.6 + 0.5$$

$$= 4.7$$

Setelah hasil penjumlahan di atas diketahui, maka didapatlah rumus untuk menghitung nilai semesta adalah sebagai berikut :

$$P(H1) = \frac{H1}{\sum_{k=1}^7 Hk} = \frac{0.9}{4.7} = 0.191$$

$$P(H2) = \frac{H2}{\sum_{k=1}^7 Hk} = \frac{0.7}{4.7} = 0.148$$

$$P(H3) = \frac{H3}{\sum_{k=1}^7 H_k} = \frac{0.8}{4.7} = 0.170$$

$$P(H4) = \frac{H4}{\sum_{k=1}^7 H_k} = \frac{0.5}{4.7} = 0.106$$

$$P(H5) = \frac{H5}{\sum_{k=1}^7 H_k} = \frac{0.7}{4.7} = 0.148$$

$$P(H6) = \frac{H6}{\sum_{k=1}^7 H_k} = \frac{0.6}{4.7} = 0.127$$

$$P(H7) = \frac{H7}{\sum_{k=1}^7 H_k} = \frac{0.5}{4.7} = 0.106$$

Setelah nilai $P(H_i)$ diketahui, probabilitashipotesis H tanpa memandang *evidence* apapun, makalangkah selanjutnya adalah :

$$\begin{aligned} &= P(H1) * P(E|H1) + P(H2) * P(E|H2) + P(H3) * P(E|H3) + P(H4) * P(E|H4) \\ &\quad + P(H5) * P(E|H5) + P(H6) * P(E|H6) + P(H7) * P(E|H7) \\ &= (0.191 * 0.9) + (0.148 * 0.7) + (0.170 * 0.8) + (0.106 * 0.5) + (0.148 * 0.7) + \\ &\quad (0.127 * 0.6) + (0.106 * 0.5) \\ &= 0.1719 + 0.1036 + 0.136 + 0.053 + 0.1036 + 0.0762 + 0.053 \\ &= 0.6973 \end{aligned}$$

Langkah selanjutnya ialah mencari nilai $P(H_i|E)$ atau probabilitas hipotesis H_i benar jika diberikan *evidence* E .

$$P(H1|E) = \frac{0.9 * 0.191}{0.6973} = 0.246522$$

$$P(H2|E) = \frac{0.7 * 0.148}{0.6973} = 0.148573$$

$$P(H3|E) = \frac{0.8 * 0.170}{0.6973} = 0.195038$$

$$P(H4|E) = \frac{0.5 * 0.106}{0.6973} = 0.076007$$

$$0.6973$$

$$P(H5|E) = \frac{0.7 * 0.148}{0.6973} = 0.148573$$

$$P(H6|E) = \frac{0.6 * 0.127}{0.6973} = 0.109278$$

$$P(H7|E) = \frac{0.5 * 0.106}{0.6973} = 0.076007$$

Setelah seluruh nilai $P(H_i|E)$ diketahui, maka jumlahkan seluruh nilai bayesnya

$$= (0.9 * 0.246522) + (0.7 * 0.148573) + (0.8 * 0.195038) + (0.5 * 0.076007) +$$

$$(0.7 * 0.148573) + (0.6 * 0.109278) + (0.5 * 0.076007)$$

$$= 0.2218698 + 0.1040011 + 0.1560304 + 0.0380035 + 0.1040011$$

$$+ 0.0655668 + 0.0380035$$

$$= 0.7274762 * 100\%$$

$$= 72.74762\%$$

II.5. Database

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi. *Database* mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah:

1. Redundansi dan Inkonsistensi data
2. Kesulitan dalam pengaksesan data
3. Isolasi data untuk standarisasi
4. Multi user

5. Keamanan data
6. Integritas data
7. Kebebasan data. (Urva dan Siregar; 2015 :95)

II.6. Normalisasi

Normalisasi merupakan parameter digunakan untuk menghindari duplikasiterhadap tabel dalam basis data dan jugamerupakan proses mendekomposisikansebuah tabel yang masih memiliki beberapa anomali atau ketidakwajaran sehinggamenghasilkan tabel yang lebih sederhanadan struktur yang bagus, yaitu sebuah tabelyang tidak memiliki *data redundancy* danmemungkinkan *user* untuk melakukan *insert*, *delete*, dan *update* pada baris(*record*) tanpa menyebabkan inkonsistensidata. (Triyono, 2011 : 19).

1. *First Normal Form (1 NF)*

Sudah tidak ada *repeating group* yaitupengulangan yang terjadi pada beberapaatribut atau kolom dalam sebuah tabel,dan juga setiap atribut harus bernilaitunggal. Atribut *multivalued*, *composite*, *derive* tidak tunggal. Setiapnilai dari atribut hanya mempunyai nilaitunggal.

2. *Second Normal Form (2 NF)*

Untuk menjadikan tabel normal tingkatke 2 maka sudah 1NF dan setiap atributyang bukan *primary key* sepenuhnya secara *funksional* tergantung padasemua atributpembentuk *primary key*.

3. *Third Normal Form (3 NF)*

Tabel sudah 2NF dan tidak memilikitransitive dependencies, *Transitive dependency* adalah ketika ada atributyang secara tidak langsung

tergantungan pada *primary key* dan atribut tersebut juga tergantung pada atribut lain yang bukan *primary key*.

4. *Boyce-codd Normal Form (BCNF)*

Tabel dalam BCNF jika sudah 3NF dan semua *determinants* adalah *candidate keys*. Perbedaan 3NF dan BCNF adalah untuk *functional dependency* $A \rightarrow B$, 3NF memperbolehkan ketergantungan pada dalam relasi jika B adalah *PrimaryKey* dan A bukan merupakan *candidate key*. Sedangkan BCNF menuntut untuk ketergantungan tetap ada dalam relasi, A harus menjadi *candidate key*.

5. *Fourth Normal Form (4 NF)*

Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalued dependency*.

6. *Fifth Normal Form (5 NF)*

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika tabel tersebut dapat dipecah atau diproyeksikan menjadi beberapa tabel dan dari proyeksi-proyeksi itu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula. Jika penyusunan ini tidak mungkin dilakukan dikatakan pada relasi itu terdapat *join dependencies* dan dikatakan bersifat *lossy join*. (Triyono, 2011 : 20).

II.7. *Visual Basic 2010*

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu *.Net Framework 4.0*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database-database*, baik *standalone* maupun *database server*. (Sari, dkk, 2015 : 2).

II.8. *SQL Server 2008*

SQL Server 2008 adalah sebuah *Relational Database Management System* (RDBMS) yang sangat *powerful* dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbarunya ini, *SQL Server 2008* memiliki banyak fitur yang bisa diandalkan untuk meningkatkan performa *database*.

SQL Server 2008 memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari-hari berkaitan dengan *database*, seperti menulis T-SQL, melakukan *backup* dan *restore database*, melakukan *security database* terhadap aplikasi, dan sebagainya. Pada GUI tersebut kita bisa melakukan settingan terhadap *SQL Server* untuk bekerja lebih optimal. *Settingan* juga bisa dilakukan menggunakan *script* untuk memudahkan *developer* mengubah *Setting Options* pada *SQL Server 2008*. (Ruslan, 2013 : 39).

II.9. *Unified Modeling Language (UML)*

Menurut Windu Gata (2013) 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.


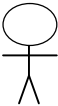

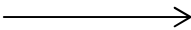
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Urva dan Siregar, 2015 :93).

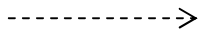
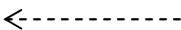
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

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 dapat dilihat pada tabel II.1 dibawah ini:

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 beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>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.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk</p>




	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.

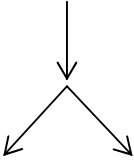
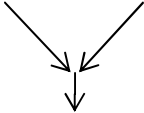
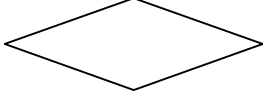

(Sumber:Urva dan Siregar; 2015 : 94)

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* dapat dilihat pada tabel II.2 dibawah ini:

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.

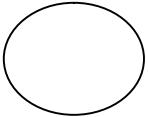
 <hr/>	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.</p>
 <hr/>	<p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

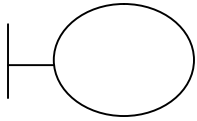
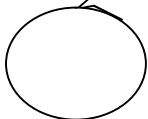

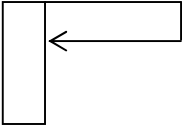


(Sumber : Urva dan Siregar; 2015 : 94)

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* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>EntityClass</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>

	<p><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.</p>
	<p><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.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Urva dan Siregar; 2015 : 95)

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 yang dapat dilihat pada tabel II.4 dibawah ini:

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 : Urva dan Siregar; 2015 : 95)