

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Triara Puspitasari (2016) dengan judul Implementasi Metode Dempster-Shafer Dalam Sistem Pakar Diagnosa Anak Tunagrahita Berbasis Web. Sistem pakar ini dibangun dengan menggunakan bahasa pemrograman PHP dan database MySQL. Output dari sistem ini adalah hasil diagnosis berupa nilai kesimpulan dan klasifikasi tunagrahita yang dialami oleh penderita. Hasil analisis dan pengujian menunjukkan bahwa ditinjau dari segi akurasi diagnosis dengan Dempster-Shafer mampu menghasilkan diagnosis yang akurat. Dari uji kelayakan sistem yang dilakukan menggunakan kuesioner diperoleh baik dengan persentase variabel 52,50%, variabel kemudahan penggunaan 41,67%, variabel kinerja sistem 45,00%, dan variabel isi 48,33%. Sistem ini sudah dapat diakses di www.sistempakar-anaktunagrahita.id

Penelitian yang dilakukan oleh Endang Lestari (2018) dengan judul Sistem Pakar Dengan Metode Dempster Shafer Untuk Diagnosis Gangguan Layanan Indihome Di Pt Telkom Magelang. Hasil penelitian ini adalah sebuah sistem pakar yang mendiagnosis gangguan layanan Indihome. Sistem secara otomatis memberikan hasil diagnosis dengan menampilkan jenis gangguan beserta solusinya berdasarkan gejala gangguan yang dialami. Kesimpulan penelitian ini adalah bahwa sistem pakar dengan menggunakan metode Dempster Shafer untuk

menangani ketidakpastian data saat diagnosis gangguan layanan Indihome sangat membantu dalam mengatasi masalah kualitas pelayanan yang menurun.

Penelitian yang dilakukan oleh Puri Yatna (2019) dengan judul Sistem Pakar Mendiagnosis Penyakit Pada Ikan Gurami Menggunakan Metode Dempster Shafer. Dalam sistem pakar terdapat beberapa mesin inferensi yang dapat digunakan, salah satunya dengan menggunakan Dempster Shafer yang merupakan teori matematika untuk pembuktian berdasarkan belief functions (fungsi kepercayaan) dan plausible reasoning (penalaran yang masuk akal). Digunakan untuk mengkalkulasi kemungkinan dari suatu peristiwa yang didapat dari kombinasi potongan informasi (fakta) yang terpisah. Dalam penelitian ini dibuat aplikasi sistem pakar diagnosa penyakit pada ikan gurami menggunakan mesin inferensi Dempster Shafer menghasilkan perhitungan tingkat keyakinan terhadap penyakit ikan gurame paling tinggi 99.85% pada penyakit Penyakit Merah (*Motile Aeromonas Septocemia*) dan paling rendah 90% pada penyakit Bintik putih (*White spot*).

Penelitian yang dilakukan oleh Syailendra Orthege (2017) dengan judul Implementasi Metode Dempster-Shafer untuk Mendiagnosa Penyakit Tanaman Padi. Pengumpulan data dalam penelitian ini menggunakan studi literatur, metode wawancara dan metode observasi. Hasil dari penelitian ini adalah sebuah sistem untuk mendiagnosa penyakit tanaman padi menggunakan metode Dempster-Shafer yang memuat berbagai gejala, penyebab, solusi dan hasil diagnosis yang berdasarkan basis pengetahuan para pakar atau para ahli di bidang pertanian. Dari kasus uji yang telah dilakukan, hasil dari pengujian akurasi yaitu 90% yang

menunjukkan bahwa aplikasi berfungsi dengan baik sesuai dengan metode Dempster-Shafer.

Penelitian yang dilakukan oleh Nita Sari Br Sembiring (2017) dengan judul Penerapan Metode Dempster Shafer Untuk Mendiagnosa Penyakit Dari Akibat Bakteri *Treponema Pallidum*. Berdasarkan gejala yang tampak dan untuk membantu masyarakat untuk dapat mengetahui lebih dini mengenai bakteri *Treponema Pallidum* dan tingkatan penyakit yang ditimbulkannya. Metode penelitian yang digunakan dalam penelitian ini diadaptasi dari kerangka berpikir metodologi penelitian sistem informasi yaitu dimulai dari tahap eksplorasi konsep, analisis, pengumpulan dan analisis data serta penarikan kesimpulan

Penelitian yang dilakukan oleh Yudi (2018) dengan judul Sistem pakar Diagnosa Penyakit Yang Disebabkan Oleh Hewan Protozoa Metode Dempster Shafer. Dalam hal ini, penulis memberikan Informasi Eksekutif dalam mendiagnosis penyakit yang disebabkan oleh hewan protozoa. Oleh sebab itu maka penulis sangat tertarik untuk mempermudah masyarakat dalam mendapatkan informasi mengenai diagnosis penyakit yang disebabkan oleh hewan protozoa.

Penelitian yang dilakukan oleh Wirhan Fahrozi (2018) dengan judul Penerapan Metode Dempster Shafer Untuk Mendiagnosa Penyakit Ikan Koi. Dalam penelitian ini penulis membantu para peternak dapat mendiagnosis penyakit yang terjadi pada ikan koi dan mengetahui cara penanganan penyakit dengan menjawab pertanyaan pertanyaan yang diajukan oleh sistem.

Penelitian yang dilakukan oleh Adil Setiawan (2018) dengan judul Sistem Pakar Mendiagnosa Penyakit Bebek Menggunakan Metode Dempster Shafer. Dalam penelitian ini penulis memberikan kemudahan pada user untuk melakukan diagnosa mendiagnosa penyakit bebek dan akan memberikan solusi berupa penanganan awal serta pola penanganan penyakit bebek. Hasil pengujian Sistem Pakar Mendiagnosa Penyakit Bebek Menggunakan Metode Dempster Shafer menunjukkan bahwa sistem pakar ini dapat mendiagnosa penyakit penyakit bebek sesuai dengan jawaban yang diberikan oleh user dengan nilai keakuratan 100%.

II.2. Landasan Teori

II.2.1. Sistem

Menurut Hesty dkk (2017:66), “Sistem adalah serangkaian subsistem yang saling terkait dan tergantung satu sama lain, bekerja bersama-sama untuk mencapai tujuan dan sasaran yang sudah ditetapkan sebelumnya”.

Menurut Maniah dan Dini Hamidini (2017:1), mengatakan bahwa “Sistem adalah kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware maupun software yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan/sasaran tertentu yang sama”.

II.2.2. Sistem Pakar

Menurut (Charles Jhony Mantho Sianturi, 2017) Sistem pakar adalah suatu sistem yang mengelola pengetahuan seorang ahli atau manusia yang dimasukkan ke dalam sistem komputer sehingga membuat sistem atau komputer itu sendiri layaknya seorang ahli.

Menurut (Yudi, 2018) Sistem pakar dapat diartikan sebagai sistem yang mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh pakar, sehingga sistem pakar dapat menjadi asisten dari seorang pakar. Aplikasi yang dapat membantu mendiagnosis suatu penyakit berbasis pengetahuan biasa disebut kecerdasan buatan.

II.2.3. Dempster Shafer

Menurut Nita Sari Br Sembiring (2017:182) teori ini memiliki beberapa karakteristik yang secara instutitif sesuai dengan cara berfikir seorang pakar, namun dasar matematika yang kuat. *Belief* (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 (nol) maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. Menurut Giarratano dan Riley fungsi *belief* dapat diformulasikan sebagai:

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \dots \dots \dots (1)$$

sedangkan *Plausibility* (Pls) dinotasikan sebagai :

$$Pls(X) = 1 - Bel(X') = 1 - \sum_{Y \subseteq X'} m(Y) \dots \dots \dots (2)$$

dimana:

$$Bel(X) = Belief(X)$$

$$Pls(X) = Plausibility(X)$$

$$m(X) = mass\ function\ dari\ (X)$$

$$m(Y) = mass\ function\ dari\ (Y)$$

Plausibility juga bernilai 0 sampai 1, jika kita yakin akan X' maka dapat dikatakan $Belief(X') = 1$ sehingga dari rumus di atas nilai $Pls(X) = 0$.

Pada teori *Dempster-Shafer* juga dikenal adanya *frame of discernment* yang dinotasikan dengan Θ . FOD ini merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* (Adrian O'Neill, 2000), dimana:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_n\} \dots \dots \dots (3)$$

dimana:

$$\Theta = FOD\ atau\ environment$$

$$\theta_1 \dots \theta_n = elemen/unsur\ bagian\ dalam\ environment$$

Environment mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan $P(\Theta)$, setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.

$$m = P(\Theta) \rightarrow [0,1]$$

sehingga dapat dirumuskan:

$$\sum_{X \in P(\Theta)} m(X) = 1 \approx \sum_{X \in P(\theta)} m(X) = 1 \dots \dots \dots (4)$$

dengan $P(\Theta)$ = *power set* dan $m(X)$ = *mass function* dari (X)

sebagai contoh:

$$P(\text{hostile}) = 0,7$$

$$P(\text{non-hostile}) = 1 - 0,7 = 0,3$$

Pada contoh di atas *belief* dari *hostile* adalah 0,7 sedangkan *disbelief hostile* adalah 0,3. dalam teori *Dempster-Shafer*, *disbelief* dalam *environment* biasanya dinotasikan $m(\theta)$.

Sedangkan *mass function* (m) dalam teori *Dempster-Shafer* adalah tingkat kepercayaan dari suatu *evidence* (gejala), sering disebut dengan *evidence measure* sehingga dinotasikan dengan (m).

Pada aplikasi sistem pakar dalam satu penyakit terdapat sejumlah *evidence* yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk diagnosa suatu penyakit. Untuk mengatasi sejumlah *evidence* tersebut pada teori *Dempster-Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination* .

$$m1 \oplus m2(Z) = \sum_{X \cap Y = Z} m1(X)m2(Y) \dots \dots \dots (5)$$

dimana:

$$m1 \oplus m2(Z) = \text{mass function dari evidence (Z)}$$

$$m1(X) = \text{mass function dari evidence (X)}$$

$$m2(Y) = \text{mass function dari evidence (Y)}$$

\oplus = operator *direct sum*

secara umum formulasi untuk *Dempster's Rule of Combination* adalah:

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - k} \dots\dots\dots(6)$$

dimana: k = Jumlah *evidential conflict*.

Besarnya jumlah *evidential conflict* (k) dirumuskan dengan:

$$k = \sum_{X \cap Y = \emptyset} m1(X)m2(Y) \dots\dots\dots(7)$$

sehingga bila persamaan (7) disubstitusikan ke persamaan (6) akan menjadi:

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - \sum_{X \cap Y = \emptyset} m1(X)m2(Y)} \dots\dots\dots(8)$$

dimana: $m1 \oplus m2(Z)$ = *mass function* dari *evidence* (Z)

$m1(X)$ = *mass function* dari *evidence* (X)

$m2(Y)$ = *mass function* dari *evidence* (Y)

k = jumlah *evidential conflict*

Contoh:

$$\Theta = \{Ko, Kr, Ks, Kv\}$$

dimana: Ko = Kanker *Ovarium*

Kr = Kanker Rahim

Ks = Kanker *Serviks*

Kv = Kanker *Vulva*

Andaikan seorang pasien mengalami 2 gejala :

1. Tercium bau amis pada vagina dengan nilai probabilitas 0,8. Gejala tersebut adalah gejala dari penyakit : Kanker *Ovarium*, Kanker Rahim dan Kanker *Serviks*.

$$m1 \{Ko,Kr,Ks\} = 0.8$$

$$m1(\theta) = 1 - 0.8 = 0.2$$

2. Nyeri pinggang dengan nilai probabilitas 0.9. Gejala tersebut adalah gejala dari penyakit : Kanker Rahim, Kanker *Serviks* dan Kanker *Vulva*.

$$m2 \{Kr,Ks,Kv\} = 0.9$$

$$m2(\theta) = 1 - 0.9 = 0.1$$

Dengan munculnya gejala kedua yaitu nyeri pada pinggang, maka harus dilakukan penghitungan densitas baru untuk beberapa kombinasi ($m3$).

Untuk memudahkan perhitungan maka himpunan-himpunan bagian yang terbentuk dimasukkan ke dalam tabel. Kolom pertama diisi dengan gejala yang pertama ($m1$). Sedangkan baris pertama diisi dengan gejala yang kedua ($m2$). Sehingga diperoleh nilai $m3$ sebagai hasil kombinasi $m1$ dan $m2$.

Tabel 4. Tabel perhitungan dua gejala

		$\{Kr,Ks,Kv\}$ (0.9)	θ (0.1)
$\{Ko,Kr,Ks\}$ (0.8)		$\{Kr,Ks\}$ (0.72)	$\{Ko,Kr,Ks\}$ (0.08)
θ (0.2)		$\{Kr,Ks,Kv\}$ (0.18)	θ (0.02)

Sehingga dapat dihitung :

$$m3 \{Kr,Ks\} = \frac{0.72}{1-0.02} = 0.73$$

$$1-0.02$$

$$m3 \{Kr,Ks,Kv\} = \frac{0.18}{1-0.02} = 0.18$$

$$1-0.02$$

$$m3 \{Ko,Kr,Ks\} = \frac{0.08}{1-0.02} = 0.08$$

$$m_3 \{ \theta \} = \frac{1-0.02}{1-0.02} = 0.02$$

Dari hasil perhitungan nilai densitas m_3 kombinasi di atas dapat dilihat bahwa terdapat gejala pertama yaitu $\{K_o, K_r, K_s\} = 0.8$, namun setelah ada gejala kedua yaitu $\{K_r, K_s, K_v\} = 0.9$. Sedangkan nilai $\{K_r, K_s\} = 0.73$. Jadi dari perhitungan di atas, nilai densitas yang paling tinggi adalah $m_3\{K_r, K_s\} = 0.73$

Jika kemudian terdapat gejala lain yaitu :

3. Siklus menstruasi tidak teratur dengan nilai probabilitas 0,6. Gejala tersebut merupakan gejala penyakit Kanker *Serviks*.

$$m_4 = 0.6$$

$$m(\theta) = 1 - 0.6 = 0.4$$

Dengan munculnya gejala baru yaitu menstruasi tidak teratur, maka harus dilakukan perhitungan densitas baru, untuk beberapa kombinasi (m_5). Untuk memudahkan perhitungan maka himpunan-himpunan bagian yang terbentuk dimasukkan ke dalam tabel. Kolom pertama berisi semua himpunan bagian pada m_3 (1) sebagai fungsi densitas. Sedangkan baris pertama berisi semua himpunan bagian pada gejala menstruasi yang tidak teratur dengan m_4 sebagai fungsi densitas. Sehingga diperoleh nilai m_5 sebagai hasil m kombinasi (Edi Faizal ; 2013 : 58-62).

II.2.5. Sql Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang database. SQL Server adalah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

Microsoft merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
2. Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7 (Iskandar Zulkarnaen ; 2018 : 3)

II.2.6. Teknik Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal / data mentah biasa disebut juga *unnormalized form*. Masing – masing level normalisasi mempunyai aturan tersendiri.

1. *First Normal Form*

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan array.
- c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. *Second Normal Form*

Suatu tabel dikatakan dalam keadaan *Second Normal Form* (2NF) jika tabel tersebut sudah dalam keadaan *First Normal Form* (1NF) dan jika semua atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form*

Suatu tabel dikatakan dalam keadaan *third normal form* (3NF) jika tabel tersebut sudah dalam keadaan *second normal form* (2NF) dan jika tidak

terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

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

Tabel dikatakan dalam keadaan *boyce-codd normal form (BCNF)* jika tabel tersebut dalam keadaan *third normal form (3NF)* dan setiap determinan adalah kunci kandidat.

5. *Fourth Normal Form (4NF)*

Suatu tabel dikatakan dalam keadaan *fourth normal form (4NF)* jika tabel tersebut dalam keadaan *boyce-codd normal form (BCNF)* dan jika tidak terdapat ketergantungan nilai ganda.

6. *Fiveth Normal Form (5NF)*

Tabel dikatakan dalam keadaan *Fiveth Normal Form (5NF)* jika tabel tersebut dalam keadaan *fourth normal form (4NF)* dan jika setiap ketergantungan dalam join ada pada tabel sudah konsekuen dengan kunci kandidat pada tabel tersebut (Ema Utami ; 2015 : 73-76).

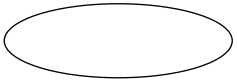
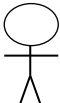


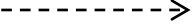
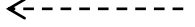
II.2.7. Unified Modelling Language (UML)

Unifed Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Indra Griha Tofik Isa ; 2017 : 141)

a. Use case Diagram

Use case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). sehingga pembuatan *use case diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem.

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 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 : Indra Griha Tofik Isa ; 2017 : 141)

b. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Tabel II.2. Multiplicity Class Diagram




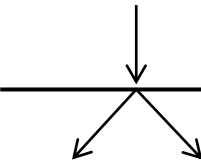
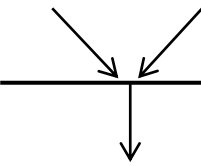
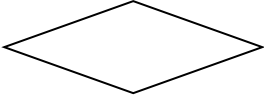

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Indra Griha Tofik Isa ; 2017 : 141)

c. Activity Diagram

Activity Diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *Activity Diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

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 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> .
 New Swimline	<i>Swimlane</i> , pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.

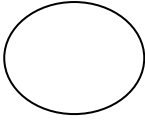
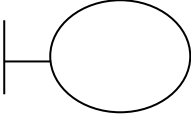
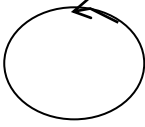
(Sumber : Indra Griha Tofik Isa ; 2017 : 141)


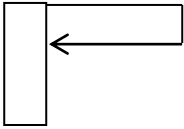


d. *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

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> .

	
	<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 : Indra Griha Tofik Isa ; 2017 : 141)