

BAB II

TINJAUAN PUSTAKA

II.2. Landasan Teori

Landasan teori merupakan dasar dari literature yang membantu penelitian dalam menyelesaikan masalah.

II.2.1 Penelitian Terkait

Untuk mendukung keberhasilan peneliti ini, penyusun melakukan pendekatan teoritis melalui beberapa literature yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu :

Penelitian yang dilakukan oleh Edi Faizal, dkk (2015; 26) yang berjudul *“Integrasi Case-Based Reasoning Dan Rule-Based Reasoning Untuk Pengembangan Sistem Pendeteksi Dini Gangguan Tumbuh Kembang Anak”*. Penelitian tersebut telah menghasilkan sebuah aplikasi yang dapat melakukan pengembangan sistem pendeteksi gangguan/penyakit untuk perkembangan pertumbuhan anak. Adapun penelitian yang akan dilakukan adalah sebuah sistem pakar tentang penyakit ikan Hias yang memperhatikan harga dan kualitas ikan Hias tersebut dengan menerapkan metode *Case Based Reasoning*.

Penelitian lain yang dilakukan oleh Billy Kadmiel, dkk (2016; 220) dengan judul *“Implementasi Case Based Reasoning Untuk Menentukan Tujuan Wisata”*. Penelitian ini menghasilkan sebuah implementasi Metode *Case Based Reasoning* dan teknik similarity dapat memberikan hasil rekomendasi yang akurat kepada pengguna serta bobot akhir dari setiap hasil rekomendasi yang diberikan. Bobot akhir yang tertinggi akan menjadi rekomendasi terbaik kepada pengguna

aplikasi. Output yang dihasilkan berupa value atau nilai bobot dan rekomendasi tempat wisata, serta keterangan dari tempat wisata tersebut. Adapun yang akan membedakan dari sistem aplikasi yang akan dibangun penulis adalah sebuah sistem pakar penyakit ikan Hias dengan menerapkan Metode *Case Based Reasoning*.

Penelitian lain yang dilakukan oleh Arno Reza Pahlawan, dkk (2017; 215) dengan judul "*Implementasi Case Based Reasoning Untuk Sistem Diagnosis Hama Dan Penyakit Tanaman Cabe Merah Menggunakan Algoritma Similitas Neyman*". Penelitian ini menghasilkan sebuah aplikasi dirancang untuk meminimalkan kesalahan dengan pengujian hipotesis *biner*, dan seseorang harus memutuskan antara hipotesis nol dan alternatif hipotesis. Sebuah tingkat pengujian dikenakan pada probabilitas *false alarm* dan tetap mencari tes yang memenuhi hambatan ini sambil memperkecil kehilangan probabilitas serta memaksimalkan ekuivalen deteksi probabilitas. Yang membedakan dari sistem aplikasi yang akan dibangun penulis sebuah sistem pakar penyakit ikan Hias dengan menerapkan sedangkan peneliti diatas sebuah aplikasi sistem pakar yang mendiagnosa penyakit hama dan penyakit cabe merah dengan menggunakan Metode yang sama (*Case Based Reasoning*).

Berdasarkan hasil penelitian yang terdahulu, maka dibuatlah kesimpulan untuk merancang sebuah aplikasi untuk mengimplementasikan sebuah aplikasi yang belum pernah dibuat atau dibangun oleh orang lain. Sehingga pada penulisan skripsi ini dibuatlah sebuah judul "*Sistem Pakar Penerapan Metode Case Based Reasoning Diagnosa Penyakit Ikan Hias Air Tawar*". Berdasarkan judul tersebut

nantinya akan dihasilkan sebuah aplikasi Sistem Pakar Diagnosa Penyakit Ikan Hias Air Tawar dengan menerapkan metode *Case Based Reasoning*.

II.2. Sistem Pakar

Sistem pakar merupakan sebuah perangkat lunak yang kemampuannya dibuat mirip layaknya seorang pakar, dimana sistem pakar berusaha untuk mengimplementasikan kemampuan seorang pakar untuk menyelesaikan suatu permasalahan, sesuai dengan pengetahuan yang dimilikinya (Sharda et al., 2014). Sistem pakar menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan dari Sistem Pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mensubstitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak. Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Kusrini, 2008). Sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti biasa dilakukan oleh para ahli (Sandi Kosasi: 2015; 17).

II.3. Diagnosa

Diagnosa adalah Proses menemukan kelemahan atau penyakit apa yang dialami seseorang dengan melalui pengujian dan studi yang seksama mengenai gejala - gejalanya. Diagnosa memiliki proses tidak langsung ketika mengidentifikasi jenis penyakit dengan cara mengetahui jenisnya. Sehingga kita

dapat dikatakan bahwa itu penyakit yang sedang di alami (Asri Azis Iskandar; 2020; 127).

II.4. Penyakit

Penyakit dapat diartikan sebagai organisme yang hidup dan berkembang di dalam tubuh sehingga organ tubuh terganggu, jika salah satu atau sebagian organ tubuh terganggu, akan terganggu pula pada seluruh jaringan tubuh, lingkungan, dan patogen. Dalam kondisi tubuh yang buruk, sangatlah besar kemungkinan terserang penyakit. Sebaliknya jika kondisi tubuhnya baik, sangat kecil kemungkinan terserang penyakit. Kondisi perubahan lingkungan secara mendadak yang membuat tubuh mengalami kondisi keadaan tidak siap dalam menghadapi suatu kondisi tertentu (Arga Dian Setyo Wicaksono; 2016; 7).

II.5. Ikan Hias

Ikan hias merupakan salah satu hewan peliharaan yang digemari oleh berbagai kalangan dikarenakan harganya yang mahal dan bentuknya yang cantik. Ikan hias yang banyak dipelihara adalah ikan hias jenis air tawar karena mudah dalam perawatan terutama segi air yang digunakan. Ada beberapa aspek yang harus dijaga dan diperhatikan dalam pemeliharaan ikan hias. Aspek yang mempengaruhi tumbuh kembang dan kesehatan ikan hias antara lain adalah kualitas air, tingkat keasaman (pH) air, suhu air dan intensitas pemberian pakan. Pemilik ikan hias biasanya kurang memperhatikan aspek-aspek tersebut karena biasanya pemilik ikan sibuk dengan segala aktivitas dan meninggalkan akuariumnya di rumah tanpa pengawasan dan pengontrolan. Oleh karena itu,

kualitas ikan hias yang dipelihara tidak mampu bertahan hidup lama. Hal itu menjadikan kerugian yang besar bagi pemilik ikan hias (Agung Gumilar Putra, dkk; 2019; 28).

II.6. Ikan Maanvis

Ikan Maanvis merupakan komoditas perikanan yang populer di masyarakat, karena ikan ini merupakan jenis ikan hias yang banyak diminati. Bentuknya yang pipih dari samping, gerakannya anggun serta mempunyai daya tarik karena memiliki sirip punggung maupun sirip perut yang membentang ke arah ekor, warna yang mencolok, bentuk yang indah serta tingkah laku yang tenang (Budi Setiawan; 2019; 49).

II.7. Ikan Mas Koki Ryukin

Ikan mas koki ryukin adalah jenis ikan mas koki yang mempunyai sirip yang sangat besar. Bahkan, ukuran siripnya dua kali lipat dari ukuran badannya. Pada umumnya, ikan ini berwarna merah. Namun, masih ada beberapa variasi warna yang tidak kalah cantik dengan warna merah (Ade Tri Bowo; 2015; 4).

II.8. Ikan Mas Koki

Ikan maskoki merupakan ikan hias air tawar yang banyak ditemukan di Indonesia salah satunya yaitu daerah Jawa Timur. Ikan maskoki memiliki beberapa keunggulan diantaranya dapat dijadikan ikan hias yang jinak, dan memiliki warna yang indah dan bentuk tubuh unik. Selain itu ikan maskoki mudah dipelihara (Ade Tri Bowo; 2016; 2).

II.9. Metode *Case Based Reasoning* (CBR)

Metode *case based reasoning* merupakan metode yang menerapkan 4 tahapan proses, yaitu *retrieve*, *reuse*, *revise*, dan *retain*. Cara kerja metode *case based reasoning* secara umum adalah berpedoman pada basis pengetahuan yang dimiliki oleh sistem yang bersumber dari kasus-kasus yang pernah terjadi pada masa yang lalu yang kemudian dihitung tingkat kemiripannya dengan kasus baru yang dimasukan pengguna. Berdasarkan tingkat kemiripan kasus inilah sistem akan mengeluarkan kemungkinan hama yang menyerang dengan pestisida yang direkomendasikan (Yaseruddin, dkk; 2015: 73-74).

Contoh Perhitungan Kasus

Kasus baru

Gejala: Terdapat lubang kecil pada daun Terdapat <i>larva</i> pada batang
Usia: 2-3 Minggu
Cuaca: Hujan
Kondisi Tanah: Lembab

Kasus 88

Gejala: Terdapat lubang kecil pada daun Daun terlihat rusak tinggal tulang daun Terdapat pupa pada tanah
Usia: 2-3 Minggu
Cuaca: Hujan
Kondisi Tanah: Lembab
Solusi: Tanaman jagung terserang hama Ulat Grayak, maka pestisida yang direkomendasikan ialah <i>Starmek 18EC</i>

Kasus 49

Gejala: Terdapat lubang kecil pada daun Terdapat larva pada batang Terdapat lubang gorokan pada batang
Usia: >3 Minggu
Cuaca: Hujan
Kondisi Tanah: Lembab
Solusi: Tanaman jagung terserang hama Lalat Bibit, maka pestisida yang direkomendasikan ialah <i>Lannate 40SP</i>

Berikut adalah contoh perhitungan kedekatan kasus yang baru dengan kasus lama yaitu:

1. Proses *Retrieve* Penerapan Algoritma *Nearest Neighbor retrieval* didapatkan hasil sebagai berikut:

Kondisi	Kasus Baru	Kasus Lama	
		Kasus 88	Kasus 49
Cuaca	Hujan	Hujan	Hujan
Usia Jagung	3-Feb Minggu	3-Feb minggu	>3 Minggu
Kondisi Tanah	Lembab	Lembab	Lembab
Terdapat lubang kecil pada daun	√	√	√
Terdapat larva pada batang	√	-	√

Daun terlihat rusak tinggal tulang daun	-	√	
pupa pada tanah	-	√	
Terdapat lubang gorokan pada batang	-	-	√

Proses perhitungan pada tahap *retrieve* adalah sebagai berikut:

Perhitungan Kasus Baru Terhadap Kasus 88

Kedekatan kriteria usia adalah 1 dengan bobot 80.

Kedekatan kriteria kondisi tanah adalah 1 dengan bobot 60.

Bobot pada gejala Terdapat lubang kecil pada daun adalah 65. Bobot pada gejala

Terdapat *larva* pada batang adalah 25. Dengan menggunakan persamaan (1)

diperoleh :

Similarity (88,baru)

$$\begin{aligned}
 &= \frac{(1 \times 80) + (1 \times 60) + (1 \times 65) + (0 \times 25)}{80 + 60 + 65 + 25} \\
 &= \frac{205}{230} \\
 &= 0,891304
 \end{aligned}$$

Jadi nilai kedekatan kasus 88 dengan kasus baru adalah 0,891304.

Kasus Baru Terhadap Kasus 49

Kedekatan kriteria usia adalah 1 dengan bobot 80 Kedekatan kriteria kondisi tanah adalah 1 dengan bobot 60 Bobot pada gejala Terdapat lubang kecil pada daun adalah 65 Bobot pada gejala Terdapat larva pada batang adalah 25.

Dengan menggunakan persamaan (1) diperoleh :

Similarity (49,Baru)

$$\begin{aligned}
 &= \frac{(1 \times 80) + (1 \times 60) + (1 \times 65) + (1 \times 25)}{80 + 60 + 65 + 25} \\
 &= \frac{230}{230} \\
 &= 1
 \end{aligned}$$

Jadi nilai kedekatan kasus 49 dengan kasus baru adalah 1.

2. Proses *Reuse* Dari hasil perhitungan tersebut, kasus yang memiliki bobot kemiripan paling rendah adalah pada kasus 88 dengan nilai 0, 891304. Kasus 49 memiliki nilai kedekatan 1. Pada proses *Reuse* solusi yang diberikan adalah solusi dengan bobot kemiripan kasus lama dengan kasus baru yang paling tinggi. Dalam contoh kasus ini kasus 49 adalah kasus yang paling mirip dengan hasil perhitungan nilai 1. Jadi kemungkinan hama yang menyerang pada kasus baru ialah hama lalat bibit, maka pestisida yang direkomendasikan adalah *lannate 40SP*.

3. Proses *Revise*

Proses *revise* adalah proses peninjauan kembali kasus dan solusi yang diberikan. Pada aplikasi penentuan pestisida pada tanaman Jagung Bisi 2, pada proses *revise* dilakukan oleh admin dimana data baru yang dimasukkan akan

disimpan pada database setelah itu akan dievaluasi dan diperbaiki oleh pakar untuk mencocokkan solusinya.

4. Proses *Retain*

Setelah proses *revise* selesai dan sudah ditemukan solusi yang benar-benar tepat barulah admin menambah kasus baru yang sudah ditemukan solusinya tersebut kedalam database yang nantinya dapat digunakan untuk kasus berikutnya yang memiliki permasalahan yang sama (Yaseruddin, dkk; 2015: 73-74).

II.10. Basis Data (*Database*)

Pengertian *database* adalah merupakan kumpulan data yang saling berhubungan satu dengan yang lain nya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* juga merupakan salah satu komponen yang penting di sistem infomasi, karena berfungsi sebagai penyedia informasi bagi para pemakainya. (Leonard Tambunan; 2018: 132).

II.11. Bentuk-bentuk Normalisasi

Normalisasi adalah proses pengelompokkan atribut data yang membentuk entittas sederhana, fleksibel, dan mudah beradaptasi, sehingga dapat dipastikan bahwa database yang dibuat berkualitas baik.

Adapun bentuk-bentuk normalisasi sebagai berikut :

1. Bentuk normal tahap pertama (1” Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok

bahan bakar minyak berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

Contoh normalisasi 1NF adalah seperti pada table berikut :

No Penyewa	No_Properti	Alamat Properti	Tgl Mulai	Tgl Akhir	Sewa Perbulan	No Pemilik	Nama Pemilik
S001	PR 1	Jln. Kebun Jeruk No.1	01/1/2008	01/12/2008	500,000	PP99	Matius
S001	PR 4	Jl. Gatot Subroto No. 100	01/1/2009	01/06/2009	1.000.000	PP77	Wahyu
S003	PR 4	Jl. Gatot Subroto No. 100	01/1/2008	01/12/2009	1.000.000	PP77	Wahyu
S003	PR 1	Jln. Kebun Jeruk No.1	01/1/2007	01/12/2007	500,000	PP99	Matius
S003	PR 2	Jln. Sudirman No.2	01/1/2008	01/12/2008	700,000	PP99	Matius

Tabel II.1. Tabel Bentuk Normal Pertama (1NF)

2. Bentuk normal tahap kedua (2nd 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.

Tabel II.2. Tabel Bentuk Normal Kedua (2NF)

Penyewa

No Penyewa	Nama Penyewa
S001	Indrajani
S003	Indrajani

Sewa Rumah

No Penyewa	No Property	Tgl Mulai Sewa	Tgl Akhir Sewa
S001	PR1	01/1/2008	01//12/2008
S001	PR4	01/1/2009	01//06/2009
S003	PR4	01/1/2008	01//12/2009
S003	PR1	01/1/2007	01//12/2007
S003	PR2	01/1/2008	01//12/2008

Properti Pemilik

No Properti	Alamat Properti	Sewa Per Bulan	No Pemilik	Nama pemilik
PR1	Jl. Kebun Jeruk No. 1	500,000	PP99	Matius
PR2	Jl. Sudirman No. 2	750,000	PP99	Matius
PR4	Jl. Gatot Subroto No. 100	1,000,000	PP77	Wahyu

3. Bentuk normal tahap ketiga (3rd 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.

Tabel II.3. Tabel Bentuk Normal Ketiga (3NF)

Properti

No Properti	Alamat	Sewa Per Bulan	No Pemilik
PR1	Jl. Kebun Jeruk No. 1	500,000	PP99
PR2	Jl. Sudirman No. 2	750,000	PP99
PR4	Jl. Gatot Subroto No. 100	1,000,000	PP77

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.

Tabel II.4. Tabel Bentuk Normal BCNF

Interview

<i>ClientNo</i>	<i>InterviewDate</i>	<i>InterviewTime</i>	<i>StaffNo</i>
CR76	13-May-05	10:30	SG5

CR56	13-May-05	12:00	SG5
CR74	13-May-05	12:00	SG37
CR56	01-Jul-05	10:30	SG5

StaffRoom

<i>StaffNo</i>	<i>InterviewDate</i>	<i>RoomNo</i>
SG5	13-May-05	G101
SG37	13-May-05	G102
SG5	01-Jul-05	G102

5. Bentuk Normal Keempat (4NF)

Sebuah tabel rasional 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 ketergantungan *multivalued* tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

Tabel II.5. Tabel Bentuk Normal Keempat (4NF)

BranchStaffOwner

BranchNo	sName	Oname
B003	Ann Beech	Carol Farel
B003	David Ford	Carol Farel
B003	Ann Beech	Tina Murphy
B003	David Ford	Tina Murphy

BranchStaff

BranchOwner

branchNo	SName
----------	-------

branchNo	SName
B003	Ann Beech
B003	David Ford

B003	Carol Farel
B003	Tina Murphy

6. Bentuk Normal Kelima

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*) (Indrajani; 2017: 58).

Tabel II.6. Tabel Bentuk Normal Kelima (5NF)

<i>ItemDescription</i>	<i>SupplierNo</i>
<i>Bed</i>	S1
<i>Chair</i>	S2
<i>Bed</i>	S2

<i>PropertyNo</i>	<i>SupplierNo</i>
PG4	S1
PG4	S2
PG16	S2

II.12. MySQL Server

MySQL adalah salah satu jenis database server yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses databasenya. *MySQL* termasuk jenis RDBMS (*Relational Database Management System*). Pada *MySQL*, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom (Priyo S; 2016: 25).

II.13. PHP

Pengertian PHP menurut Anhar (2010:23) “PHP adalah (PHP *Hypertext Preprocessor*) adalah bahasa pemrograman web berupa script yang dapat diintegrasikan dengan HTML” (Agus Prayitno, dkk ; 2015 : 2).

II.14. Unified Modeling Language (UML)


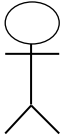


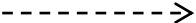
Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam membentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar ; 2018 : 49).

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 tipikal interaksi antara (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem yang dipakai (Munawar ; 2018 : 89).

Tabel II.2. Simbol *Use Case Diagram*

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 mengidentifikasikan 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 <i>control</i> 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 mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><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.</p>

←-----	<i>Extend</i> , merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.
--------	---

(Sumber : Munawar ; 2018 : 93)

2. *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 merupakan diagram statis dari suatu aplikasi. *Class Diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak (Munawar ; 2018 : 101).

Tabel II.3. Simbol *Class Diagram*




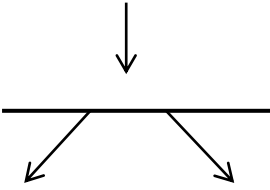
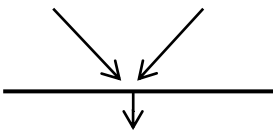
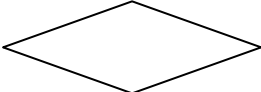
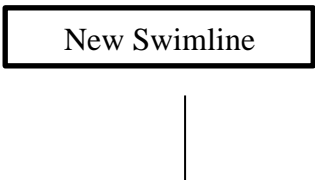
<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Munawar ; 2018 : 101)

3. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram* (Munawar ; 2018 : 137).

Tabel II.4. Simbol Diagram Aktivitas

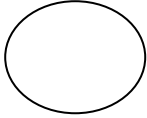
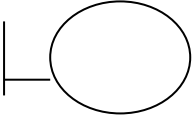
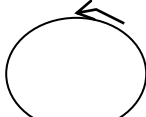

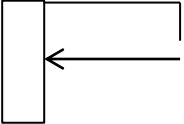

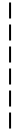
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau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

(Sumber : Munawar ; 2018 : 137)

4. Diagram Urutan (*Sequence Diagram*)

Sequence diagram adalah salah satu *interaction diagram*. Karena *sequence diagram* mengacu kepada obyek, maka sbelum membuat diagram ini class diagram sudah harus teridentifikasi (Munawar ; 2018 : 186).

Tabel II.5. Simbol Diagram Urutan

Gambar	Keterangan
	<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.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> 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> , 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 : Munawar ; 2018 : 186