

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Teori Dasar**

##### **2.1.1 Kecerdasan Buatan (Artificial Intelligence)**

Cabang dari ilmu pengetahuan yang berkaitan dengan memanfaatkan mesin dalam pemecahan persoalan yang rumit dengan cara yang lebih manusiawi dimana mesin tersebut mampu melakukan pekerjaan sebaik yang dilakukan oleh manusia bahkan lebih.

Kecerdasan buatan atau AI (*Artificial Intelligence*) merupakan ilmu yang mempelajari tentang bagaimana membangun suatu sistem komputer yang menunjukkan kecerdasan dalam berbagai cara (Andhik Ampuh Yunanto, 2015). *Artificial Intelligence* merupakan sebuah area penelitian yang bersifat dinamik yang membahas riset ilmu komputer. Telah banyak penelitian mengenai perkembangan *Artificial Intelligence* diantaranya *neural networking*, *evolutionary computing*, *machine learning*, *natural language processing*, dan *object oriented programming* (OOP).

Kecerdasan yang dimiliki manusia mampu menyelesaikan permasalahan karena manusia memiliki pengetahuan dan pengalaman. Namun pengetahuan saja tidak cukup, manusia juga harus mampu memahami setiap permasalahan yang timbul. Tanpa adanya pemahaman dan penalaran yang baik, maka permasalahan tidak akan

terselesaikan dengan baik. Dengan kedua hal tersebut manusia akan mampu menciptakan mesin cerdas yang dapat bekerja seperti layaknya manusia sehingga dapat menghasilkan informasi baru yang dapat diserap sebagai pengetahuan.

Dalam penelitiannya, (Prasetyo & Retnowati, 2015) menerangkan bahwa kecerdasan buatan adalah bagian dari bidang ilmu yang dapat menduplikasi perilaku dan fungsional dari otak manusia. Karena kecerdasan buatan mengadaptasi perilaku dan fungsional dari otak manusia, sehingga muncul istilah cerdas.

Kombinasi antara *Artificial Intelligence* dengan beberapa cabang pada bidang ilmu yang lainnya dapat melahirkan subdisiplin ilmu dalam *Artificial Intelligence*. Beberapa diantara subdisiplin ilmu lainnya antara lain Logika Fuzzy (*Fuzzy Logic*), Jaringan Saraf Tiruan (*Artificial Neural Network*), dan Sistem Pakar (*Expert System*) (Sutojo, Mulyanto, & Suhartono, 2011).

### **2.1.2 Logika Fuzzy (*Fuzzy Logic*)**

Dalam sebuah sistem, terdapat bahan masukan (*input*) dan akan menghasilkan sebuah hasil keluaran (*output*). *Fuzzy logic* adalah sebuah metode dalam memetakan sebuah sistem dari masukan hingga menghasilkan keluaran. *Output* dari *fuzzy logic* ini biasanya adalah sebuah keputusan. (Rohayani, 2013)

Logika *fuzzy* digunakan untuk mendekati atau mencocokkan bahasa yang digunakan komputer dengan bahasa manusia. Logika *fuzzy* adalah sebuah metodologi dari sistem kontrol yang digunakan dalam *problem solving* yang cocok diimplementasikan didalam sebuah sistem, mulai dari sistem yang sederhana,

*embedded system*, jaringan PC, *multichannel* atau *workstation* berbasis akuisisi data dan sistem kontrol.

*Fuzzy Logic* digunakan oleh beberapa orang dengan alasan tertentu, alasan penggunaan *fuzzy logic* antara lain (Sutojo, Mulyanto, & Suhartono, 2011):

1. Konsep dari *fuzzy logic* mudah dimengerti. Konsep matematika yang mendasari dalam penalaran *fuzzy logic* sangat simpel dan dapat dengan mudah untuk dimengerti.
2. *Fuzzy logic* sangat fleksibel.
3. *Fuzzy logic* memiliki toleransi terhadap data-data yang kurang tepat.
4. *Fuzzy logic* dapat merepresentasikan fungsi-fungsi nonlinier yang sangat kompleks.
5. *Fuzzy logic* dapat membuat dan mengaplikasikan pengalaman-pengalaman dari para pakar secara langsung tanpa harus melalui proses *training*.
6. *Fuzzy logic* dapat berkolaborasi dengan teknik-teknik kendal secara konvensional.
7. *Fuzzy logic* didasarkan pada bahasa alami.

Dalam *fuzzy logic* terdapat beberapa metode yang sering digunakan oleh penggunanya, berikut adalah beberapa metode yang digunakan dalam *fuzzy logic*.

(Sutojo, Mulyanto, & Suhartono, 2011)

1. Tsukamoto *Method*
2. Mamdani *Method*
3. Sugeno *Method*

### **2.1.3 Jaringan Saraf Tiruan (*Artificial Neural Network*)**

Jaringan saraf tiruan (*artificial neural network*) adalah salah satu cabang ilmu dari bidang ilmu Kecerdasan Buatan, merupakan alat untuk memecahkan masalah terutama di bidang-bidang yang melibatkan pengelompokan dan pengenalan pola.

*Artificial neural network* adalah sebuah paradigma dalam mengolah sebuah informasi yang sistemnya terinspirasi oleh sistem saraf manusia secara biologis, seperti proses penyampaian informasi dalam otak manusia. Elemen utamanya adalah struktur dari sistem pengolahan informasi yang terdiri dari jumlah elemen pemrosesan yang paling berhubungan (neuron), serentak untuk menyelesaikan masalah tertentu. Beberapa contoh implementasi jaringan syaraf tiruan yaitu pada bidang ilmu kedokteran, diagnosis sistem kardiovaskular, dokter instan.

### **2.1.4 Sistem Pakar (*Expert System*)**

Teknik dalam *artificial intelligent* dalam menduplikasi dari pengetahuan manusia adalah sistem pakar. Dalam memecahkan masalah-masalah yang terbilang kompleks biasanya dilakukan oleh seorang pakar yang terkait dalam bidangnya. *Expert system* mencoba untuk mencari solusi yang memuaskan, yaitu sebuah solusi yang cukup bagus agar sebuah *task* dapat berjalan walaupun itu bukan solusi yang optimal (Sutojo, Mulyanto, & Suhartono, 2011).

*Expert System* sebagai salah satu hasil dari perkembangan ilmu komputer, khususnya di bidang *Artificial Intelligence*, dapat memberikan solusi untuk mengatasi masalah tersebut. *Expert system* adalah sebuah sistem yang berusaha mengadopsi

pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti layaknya para pakar (*expert*). Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para pakar/ahli. Dengan pengembangan sistem pakar, diharapkan bahwa orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli.

*Expert system* merupakan bagian dari *Artificial Intelligence* yang cukup tua karena *expert system* ini mulai dikembangkan pada pertengahan 1960. *Expert system* ini bekerja untuk menduplikasi pengetahuan manusia kedalam sistem komputer dengan menggabungkan *knowledge base* untuk menggantikan seorang pakar dalam penyelesaian suatu masalah tertentu. *Expert system* berasal dari istilah *knowledge base expert system* (Sutojo, Mulyanto, & Suhartono, 2011). *Expert system* adalah sebuah sistem yang dirancang untuk dapat menyelesaikan suatu permasalahan tertentu dengan menduplikasi kerja dari para *expert* dalam menjawab pertanyaan yang diberikan dan memecahkan suatu permasalahan. Dengan *expert system* ini orang yang masih awam pun dapat menyelesaikan permasalahan yang cukup rumit, dimana yang sebenarnya hanya dapat diselesaikan dengan bantuan para *expert*. Bagi para ahli *expert system* ini juga membantu aktivitasnya sebagai asisten yang sangat berpengalaman.

### **A. Kategori Permasalahan Sistem Pakar**

Aplikasi dalam *expert system* mencakup dalam beberapa area permasalahan, permasalahan yang tercakup didalam *expert system* adalah berikut ini (Sutojo, Mulyanto, & Suhartono, 2011):

1. Interpretasi, adalah membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
2. Prediksi, adalah memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.
3. Diagnosa, adalah menentukan sebab malfungsi dalam situasi yang didasarkan pada gejala-gejala yang teramati.
4. Desain, adalah menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu.
5. Perencanaan, adalah merencanakan serangkaian tindakan yang dapat mencapai sejumlah tujuan dengan kondisi awal tertentu.
6. Debugging dan Repair, adalah menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.
7. Instruksi, adalah mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek.
8. Pengendalian, adalah mengatur tingkah laku suatu lingkungan yang kompleks.

## **B. Keuntungan dan Kelemahan Sistem Pakar**

Penggunaan *expert system* memiliki beberapa keuntungan, keuntungan yang didapat diantaranya adalah:

1. Memungkinkan pekerjaan seorang *expert* dapat diselesaikan dengan orang biasa (awam).
2. *Automaticly process*.
3. Dapat menyimpan pengetahuan serta keahlian dari seorang *expert*.
4. Meningkatkan kualitas konsultasi dengan cara memberi nasehat (masukan) yang konsisten serta dapat mengurangi kesalahan.
5. Meningkatkan hasil dan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari manusia.
6. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
7. Memiliki kemampuan untuk mengakses pengetahuan.
8. Meningkatkan kapabilitas dalam penyelesaian masalah.
9. Menghemat waktu dalam pengambilan keputusan.

Kelemahan sistem pakar :

1. Biaya yang diperlukan untuk membuat, memelihara, dan mengembangkannya sangat mahal.
2. Sulit dikembangkan, hal ini erat kaitannya dengan ketersediaan pakar di bidangnya.

3. Sistem pakar tidak 100% benar karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan.
4. Kepakaran sangat sulit diekstrak dari manusia. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.
5. Sangat sulit bagi seorang pakar untuk mengabstraksi atau menjelaskan langkah mereka dalam menangani masalah.
6. Sistem pakar bekerja baik untuk suatu bidang yang sempit.
7. Istilah dan jargon yang dipakai oleh pakar dalam mengekspresikan fakta seringkali terbatas dan tidak mudah dimengerti oleh orang lain.
8. Transfer pengetahuan dapat bersifat subyektif dan bias.

### **C. Komponen Sistem Pakar**

Menurut (Sutojo, Mulyanto, & Suhartono, 2011) *expert system* sebagai sebuah aplikasi yang dapat difungsikan dalam menirukan keahlian dari manusia harus bisa menyelesaikan *task* yang dapat dikerjakan oleh seorang *expert*. Agar dapat menciptakan sebuah sistem yang seperti itu, maka dibutuhkan beberapa komponen.

Komponen tersebut adalah sebagai berikut:

1. Antarmuka pengguna sistem (*user interface*)
2. Basis pengetahuan (*knowledge base*)
3. Mesin inferensi (*inference machine*)

4. Memori kerja (*working memory*)

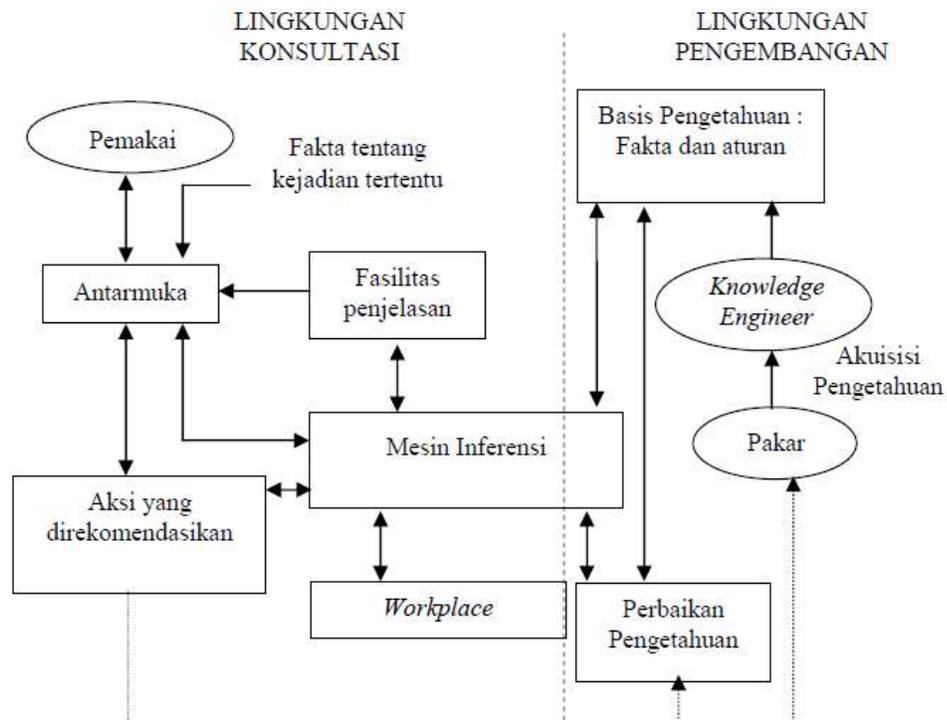
Sedangkan untuk menjadikan *expert system* menjadi lebih menyerupai seorang *expert* yang berinteraksi dengan pemakai, maka dilengkapi dengan fasilitas berikut:

1. Memfasilitas dalam melakukan penjelasan (*explanation facility*)
2. Memfasilitas dalam mengakuisisi pengetahuan (*knowledge acquisition facility*)

**D. Struktur Sistem Pakar**

Sistem pakar memiliki dua bagian utama, yaitu :

1. Lingkungan pengembangan (*development environment*), yaitu bagian yang digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar.
2. Lingkungan konsultasi (*consultation environment*), yaitu bagian yang digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan.



**Gambar 2.1** Komponen Dalam *Expert System*  
 Sumber Kecerdasan Buatan (Sutojo, Mulyanto, & Suhartono, 2011)

### E. Mesin Inferensi (*Inference Engine*)

Dalam penjelasannya (Sutojo, Mulyanto, & Suhartono, 2011) *inference engine* berperan sebagai otak dari sebuah *expert system* yang berupa *software* yang melakukan *inference task* dalam melakukan penalaran *expert system*, biasa dikatakan sebagai mesin pemikir (*Thinking Machine*). Pada dasarnya *inference engine* inilah yang akan mencari solusi dari suatu *problem*. Dalam *expert system*, proses inferensi dilakukan dengan suatu modul yang disebut Mesin Inferensi (*Inference Engine*). Ada beberapa metode inferensi yang penting dalam *expert system*, yaitu *Forward*

*Chaining* (runut maju) dan *Backward Chaining* (runut mundur), *Cased Based Reasoning*, *Certainy Factor*.

## **F. Metode Sistem Pakar**

### **1. *Forward Chaining* (Runut Maju)**

Menurut (Sutojo, Mulyanto, & Suhartono, 2011) *Forward Chaining* merupakan sebuah teknik dalam melakukan pencarian yang dimulai dengan fakta-fakta yang diketahui, kemudian mencocokkan fakta tersebut dengan bagian *IF* dan *rules IF-THEN*. Bila terdapat fakta yang cocok dengan bagian *IF*, maka *rule* dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru bagian (*THEN*) dimasukkan kedalam *database*. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dijalankan sebanyak sekali saja. Proses pencocokan fakta yang ada akan terhenti apabila tidak ada lagi *rule* yang dapat dieksekusi.

*Forward chaining* melakukan proses peruntan (penalaran) dimulai dari premis-premis atau informasi *input (IF)* terlebih dahulu kemudian menuju konklusi atau *drived information (THEN)*. Konsep ini dapat dimodelkan sebagai berikut:

*IF* (informasi masukan)

*THEN* (konklusi).

Informasi *input* dapat berupa suatu pengamatan sedangkan *conclusion* dapat berupa diagnosis sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari informasi *input*. Didalam metode *forward chaining* ini, sistem tidak melakukan praduga apapun terhadap *conclusion*, namun sistem tersebut akan menerima semua

gejala masukan yang diberikan pengguna lalu sistem akan memeriksa gejala masukan tersebut dan selanjutnya mencocokkan dengan *conclusion* yang sesuai (Sutojo, Mulyanto, & Suhartono, 2011).

## 2. ***Backward Chaining (Runut Mundur)***

*Backward chaining* sendiri merupakan metode yang berlawanan dengan *forward chaining* di mana mulai dengan sebuah hipotesis sebuah objek, setelah itu meminta sebuah informasi untuk menyakinkan atau mengabaikan *backward chaining inference engine* sering disebut *Object – Driven / Goal-Driven*.

## 3. ***Case Based Reasoning***

Metode *Cased based reasoning* merupakan metode yang menerapkan empat tahap proses, yaitu *retrieve, reuse, revise, dan retain*. Cara kerja dari sistem yang menggunakan metode *cased based reasoning* secara umum berpedoman pada *knowledge base* yang dimiliki oleh sistem yang bersumber dari *case* yang pernah ditangani oleh seorang *expert* yang kemudian dihitung tingkat kecocokannya dengan *case* baru yang dimasukkan oleh *user*.

## 4. ***Certainy Factor***

Metode *certainty factor* digunakan ketika menghadapi permasalahan yang jawabannya tidak tentu. Ketidakpastian ini bisa merupakan *probabilitas*. *Certainy factor* ini diperkenalkan Shortlife Buchanan pada tahun 1970-an. Beliau menggunakan *certainty factor* saat melakukan diagnosa dan terapi terhadap penyakit meningitis dan *blood infection*. Tim pengembang dari *certainty factor* mencatat

bahwa, dokter sering kali menganalisa informasi yang ada dengan pernyataan seperti “mungkin”, “hampir pasti”.

*Certainty factor* mirip dengan *fuzzy logic*, karena ketidakpastian direpresentasikan dengan *trustment degree* sedangkan perbedaannya adalah pada *fuzzy logic* saat perhitungan untuk *rule* yang premisnya lebih dari satu, *fuzzy logic* tidak memiliki *trustment value* untuk *rule* tersebut sehingga perhitungannya hanya melihat nilai terkecil untuk operator AND atau nilai terbesar untuk operator OR dari setiap premis yang pada *rule* tersebut berbeda dengan *certainty factor* yaitu setiap *rule* memiliki *trustment value* sendiri tidak hanya premis-premisnya saja yang memiliki nilai keyakinan. *Certainty factor* menunjukkan ukuran kepastian terhadap suatu *fact* atau *rule*.

Menurut (Khairina Eka Setyaputri, 2018) metode CF ini menuntut ketidakpastian terhadap fakta atau suatu aturan yang ada. CF sendiri merupakan sebuah indikator klinis dari ketidakpastian yang diberikan oleh *MYCIN* untuk menunjukkan besarnya kepercayaan pada suatu *fact* atau *rule* yang ada. Metode CF sendiri memiliki kelebihan dalam menghitung suatu *fact* yang pasti dan tidak pasti dalam mengambil keputusan khususnya dipergunakan dalam pembuatan *expert system*. Berikut adalah rumus dasar *MYCIN* pada metode *certainty factor*.

$$CF(h, e) = MB(h, e) - MD(h, e)$$

**Rumus 2.1** *MYCIN*

Keterangan:

$CF(h, e)$	=	<i>Certainty Factor</i> (ketidak pastian) dalam suatu hipotesis $h$ didalam gejala $e$
$MB(h, e)$	=	<i>Measure of Believe</i> (Ukuran Kepercayaan) dalam suatu hipotesis $h$ didalam gejala $e$
$MD(h, e)$	=	<i>Measure of Disbelive</i> (Ukuran Tidak Percaya) dalam suatu hipotesis $h$ didalam gejala $e$
$h$	=	hipotesis
$e$	=	<i>evidence</i> (gejala)

## 2.2 Variabel

Variabel merupakan sesuatu yang menjadi *object* pengamatan dalam penelitian, sering juga disebut sebagai faktor yang berperan dalam penelitian atau gejala yang akan diteliti. Menurut Kerlinger (Kerlinger, 2006) *variable* merupakan sebuah konstruk atau sifat yang akan dipelajari yang mempunyai nilai yang bervariasi.

Dalam penelitian ini *variable* yang dibahas adalah penyakit dalam yang timbul yang diakibatkan oleh gadget. Berikut adalah variabel yang dijadikan fokus dalam penelitian ini.

1. Rabun Jauh (*Miopy*)
2. Rabun Dekat (*Hypermetropi*)
3. *Computer Vision Syndrom* (CSV)

### 2.2.1 Rabun Jauh (*Myopi*)

Miopia merupakan keadaan dimana bayangan dari obyek yang jauh difokuskan di depan retina oleh mata yang tidak berakomodasi. Hal ini disebabkan mata memiliki kekuatan optik yang terlalu tinggi karena kornea yang terlalu cembung atau panjang aksial bola mata yang terlalu besar. Penggunaan smartphone yang monoton pada jarak dekat dan pengaturan intensitas cahaya yang tidak normal misalnya terlalu redup atau terlalu terang dapat beresiko terhadap terjadinya rabun jauh (miopia). Kondisi ruangan gelap dan layar smartphone yang dekat dengan mata dapat mempengaruhi intensitas dan kuantitas cahaya yang diterima mata sehingga menyebabkan kelainan refraksi pada mata.

#### A. Gejala Penyakit

- Penglihatan bura, ketika melihat benda dalam jarak yang jauh
- Harus menyipitkan mata
- Sakit kepala
- Kesulitan melihat saat berkendara , terutama pada malam hari
- Mengedipkan mata secara berlebihan
- Terus menggosok mata

#### B. Solusi

- Penggunaan kaca mata khusus rabun jauh
- Penggunaan lensa korektif
- Operasi refraktif

### 2.2.2 Rabun Dekat (*Hypermetropi*)

Rabun dekat adalah penyakit yang mengakibatkan penderitanya tidak dapat atau memiliki kelemahan dalam melihat *object* yang letaknya jauh dari pandangan mata. Berbeda dengan *miopy*, penderita diharuskan untuk menggunakan kacamata yang berlensa cembung.

#### A. Gejala Penyakit

- Anak sering menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas
- Sakit kepala
- Mengedipkan mata secara berlebihan
- Sakit pada bagian leher dan puncak
- Penglihatan tidak fokus ketika melihat objek yang dekat
- Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis dan membaca

#### B. Solusi

- Menggunakan kacamata khusus penderita rabun dekat
- Penggunaan lensa kontak
- Melakukakan operasi laser

### **2.2.3 Computer Syndrome Vision (CSV)**

Kondisi dimana pengguna terlalu sering menggunakan *gadget* dengan jarak pandang mata yang terlalu dekat, hal ini menyebabkan atau mengarah pada kelelahan mata.

#### **A. Gejala Penyakit**

- Terus menggosok mata
- Mata cepat merasa lelah dan pegal
- Mata terasa tegang

#### **B. Solusi**

- Sesuaikan cahaya lingkungan sekitar
- Gunakan obat tetes air mata buatan
- Atasi kondisi mata yang lain
- Berhadapan dengan gadget tidak boleh lebih dari 10 menit

## **2.3 Software Pendukung**

### **2.3.1 UML (*Unified Modeling Language*)**

Menurut penjelasan dari (Sukamto, 2016) UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML menyediakan serangkaian

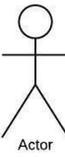
gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori objectoriented dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar team programmer maupun dengan pengguna.

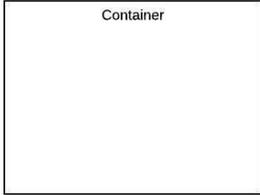
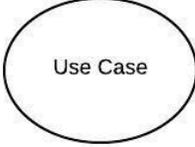
#### A. *Use Case diagram*

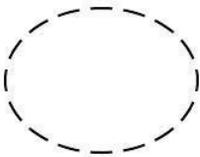
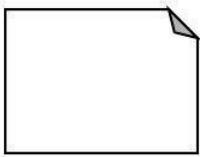
Rosa dan Shalahuddin (2013:155), Use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.

Komponen pembentuk diagram *use case* adalah: 1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem. 2. *Use case*, aktivitas / sarana yang disiapkan oleh bisnis / sistem. 3. Hubungan (link), aktor mana saja yang terlibat dalam *use case* ini.

**Tabel 2.1** Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		Aktor	Merupakan orang atau proses yang ada di dalam sistem
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada <i>dependent element</i> akan mempengaruhi <i>independent element</i>

3		<i>Generalization</i>	hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> )
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

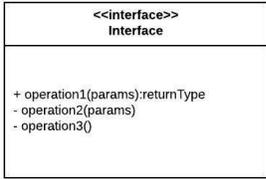
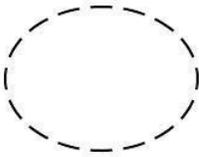
Sumber Data Penelitian 2019

### B. *Class Diagram*

Menurut (Sukamto, 2016) class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Tabel 2.2 Simbol Class Diagram

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).

2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Yang menghubungkan antara objek satu dengan objek lainnya

Sumber Data Penelitian 2019

### C. *Activity Diagram*

Menurut (Sukamto, 2016) activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

**Tabel 2.3** Simbol Activity Diagram

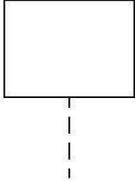
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Intial Node</i>	Bagaimana objek dibentuk atau diawali
4		<i>Activiy Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber Data Penelitian 2019

#### D. *Sequence Diagram*

Menurut (Sukanto, 2016) secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram. Dalam *sequence* diagram terdapat simbol-simbol yang merepresentasikan kegiatan yang digambarkan, berikut adalah penjelasan dari simbol-simbol yang ada di dalam *sequence* diagram.

**Tabel 2.4** Simbol *Sequence* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber Data Penelitian 2019

#### 2.3.2 PHP (*Personal Home Page*) *Programming Language*

PHP merupakan salah satu *programming language* yang sering digunakan oleh para *programmer*. PHP sering dikombinasikan dengan bahasa pemrograman HTML 5

dalam membuat sebuah *website*. Menurut (Enterprise, 2018) membedakan HTML dan PHP dari penjelasannya, dimana HTML adalah bahasa pemrograman yang digunakan untuk membuat sebuah *website* dimana HTML bukan sebuah *programming language* yang standalone sehingga tidak dapat berdiri sendiri tanpa bantuan CSS atau *script* lainnya. Sedangkan PHP adalah *programming language* yang digunakan dalam pembuatan sebuah *dynamic website* yang interaktif. PHP memiliki keuntungan dalam pembuatan *dynamic website* dibandingkan dengan HTML.

### 2.3.3 Notepad

Dalam penjelesannya (Asropudin, 2013) menjelaskan bahwa *application* atau aplikasi merupakan *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan *task* tertentu, misalnya Microsoft Word, Microsoft Excel. (Sutarman, 2012) menjelaskan bahwa *application* adalah program-program yang dibuat oleh suatu perusahaan *computer* untuk *user* yang beroperasi dalam bidang-bidang umum, seperti toko, penerbitan, komunikasi, penerbangan, perdagangan, dan sebagainya. Berdasarkan pengertian diatas peneliti menyimpulkan bahawa aplikasi atau program aplikasi adalah software atau program-program yang dibuat untuk membantu para pemakai atau user dalam mengerjakan tugas-tugas tertentu.

### 2.3.4 HTML (*Hyper Text Markup Language*)

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML biasa disebut *programming language* yang paling dasar dan penting yang digunakan dalam menampilkan dan mengelola *interface* pada *website page*.

Saat ini HTML versi 5 merupakan versi yang paling ramai diperbincangkan didunia maya, HTML5 layaknya sebuah HTML biasa yang sering kita gunakan dalam membangun sebuah *website page*, hanya saja HTML5 ini memiliki beberapa kelebihan dibandingkan versi sebelumnya.

### 2.3.5 XAMPP

XAMPP adalah sebuah *server* yang *standalone (localhost)* yang terdiri beberapa program seperti *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang dituliskan dengan *programming language* PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat *operating system* apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Publik license* dan bebas, merupakan *website server* yang mudah untuk digunakan yang dapat menampilkan *website page* yang *dynamic* secara *free* dan mendukung instalasi di *Linux* dan *Windows*.

### 2.3.6 MySQL Server

MySQL dapat diartikan sebagai sebuah *server* yang bertugas dalam pelayanan pusat olah data atau *database*. MySQL biasa digunakan oleh *programmer* dalam pembuatan serta pengolahan *database*. Di dalam MySQL terdapat fasilitas *query*

(perintah) SQL. *Database* sendiri sangat dibutuhkan apabila ingin peng *input* an suatu data dari *user* dengan menggunakan *form* HTML yang nantinya akan diproses oleh PHP dan di simpan di dalam *database* MySQL.

Instruksi SQL (*Structure Query Language*) dikelompokan berdasarkan jenis dan fungsinya masing-masing. Terdapat 3 perintah dasar dari SQL diantaranya adalah :

1. *Data Definition Language*
2. *Data Manipulation Language*
3. *Data Control Language*

#### **2.4 Penelitian Terdahulu**

1. Ruhul Amin dan Pipit Pitriani (2018). “**Penerapan Metode *Forward Chaining* Untuk Diagnosa Penyakit Insomnia**”. E-ISSN : 2527-6514. Sekolah swasta di beberapa kota besar, beberapa diantaranya melaksanakan *main task* pada siang hari, hal ini disebabkan keterbatasan sarana dan prasana yang dimiliki oleh sekolah. Proses pembelajaran yang dilakukan pada siang hari akan berdampak pada pola tidur pada siswa-siswi. Sekolah SMK PGRI Pinang di Tangerang melaksanakan kegiatan belajar mengajar pada siang hari. Pola tidur akan mempengaruhi kesehatan seseorang, pola tidur yang baik menjaga badan kita tetap sehat, sebaliknya kekurangan tidur akan menyebabkan *imune* tubuh kita berkurang sehingga mudah terserang penyakit. Tujuan dari penelitian ini adalah memudahkan mendiagnosa penyakit insomnia berdasarkan gejala-gejala yang

dimiliki oleh siswa SMK PGRI Pinang Tangerang. Penelitian ini menggunakan *forward chaining* mendapatkan sebuah kesimpulan dari gejala-gejala insomnia yang dimiliki oleh pasien. Penelitian ini menghasilkan kesimpulan tentang seorang siswa yang terkena penyakit insomnia, selain itu sistem pakar juga memberikan solusi untuk penyembuhan dari pasien, sehingga hal ini bisa menjadi acuan seorang siswa sebelum konsultasi lebih lanjut ke dokter.

2. Syamsudin, Usman dan Selviana (2017). “**Aplikasi Sistem Pakar Diagnosa Penyakit Pernapasan Menggunakan Metode *Case-Based Reasoning***”. ISSN : 1979-9292. Sebuah aplikasi telah dianalisis, dirancang, dan diimplementasi, aplikasi ini dirancang dengan menggunakan Metode *Cased-Based Reasoning*. Tujuan Aplikasi ini adalah untuk mengidentifikasi 8 Jenis Penyakit Pernapasan pada Manusia dengan 34 gejala. Untuk pengembangan sistem Pakar Menggunakan beberapa fase yaitu: *Planing*, *Analisis*, *Desain* dan *Implementasi*, dimana pengetahuan berbasis Pakar dan Pengetahuan Berbasis Kasus, mencocokkan kasus yang baru dengan kasus yang lama menggunakan teknik *similarity*. Aplikasi sistem pakar ini sudah sesuai dengan pakar penyakit pernapasan karena sudah diuji oleh pakar dan pengguna.
3. Nanda Jarti dan Roden Trisno (2017). “**Sistem Pakar Diagnosa Penyakit Alergi Pada Anak Berbasis *Web* Dengan Metode *Forward Chaining* di Kota Batam**”. ISSN : 2407-0491. Sistem pakar merupakan sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran untuk memecahkan masalah yang hanya dapat dipecahkan oleh seorang pakar. Alergi

merupakan respon *abnormal* dari sistem kekebalan tubuh. Orang yang mengalami alergi memiliki sistem kekebalan tubuh yang bereaksi terhadap suatu zat yang biasanya tidak berbahaya di lingkungan. Pembatasan pada penelitian ini terdapat empat penyebab alergi yaitu, alergi makanan laut, alergi debu, alergi susu sapi, dan alergi hewan peliharaan. Dalam perancangan sistem pakar ini merupakan sistem yang digunakan untuk mendiagnosa penyakit alergi pada anak. Proses diagnosa dimulai dengan cara *user* menjawab pertanyaan-pertanyaan berupa gejala yang diderita oleh *user*. Metode inferensi yang digunakan adalah runut maju (*Forward chaining*). *Forward chaining* merupakan teknik pencarian yang dimulai dengan fakta-fakta yang diketahui kemudian dicocokkan menggunakan *rule IF-Then*. Sistem pakar ini berbasis *web* sehingga dapat diakses kapanpun dan di manapun oleh *user* selama mereka terhubung dengan internet dan sistem ini dibangun dengan menggunakan bahasa pemrograman php, *database* MySQL, dan prosesnya menggunakan adobe dreamweaver CS6. Kesimpulan yang dapat dari hasil penelitian ini antara lain aplikasi sistem pakar yang dirancang mampu mendiagnosa penyakit alergi pada anak berdasarkan gejala-gejala yang diinput oleh *user*.

4. Minarni dan Anisah Fadhillah (2017). **“Expert System in Detecting Rice Plant Deseases Using Certainy Factor”**. e-ISSN: 2502-0692. Kehidupan di Indonesia sangat menitik beratkan dengan pengkonsumsian nasi. Produksi dari beras akan menurun apabila tanaman diserang penyakit. Penyakit dapat dikenali sejak tanaman masih muda apabila petani dapat mengetahui bagaimana

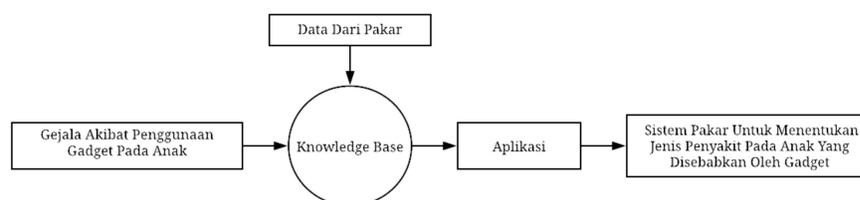
penanganannya. Faktanya, petani banyak yang tidak mengerti tentang penyakit yang ada pada tanaman padi karena kurangnya pelatihan dan kurangnya waktu pelatihan. Untuk menangani permasalahan tersebut, salah satu solusi terbaiknya adalah menciptakan sebuah aplikasi sistem pakar yang mengadopsi pengetahuan dari pakar tanaman padi. Penggunaan metode certainty factor untuk mengasumsi pengetahuan pada tanaman padi. Hasil dari sistem pakar ini akan menampilkan gambar, solusi, dan besar presentasi tanaman padi yang terkena penyakit. Hasil sistem pakar tersebut didapat dari hasil konsultasi dari pemakai. Pemakai diharuskan menjawab pertanyaan-pertanyaan tentang gejala dari penyakit. Dalam mendesain sistem pakar ini, PHP digunakan sebagai bahasa pemrograman dan database yang digunakan dibuat dengan MYSQL. Basis aplikasi ini adalah sebuah website. Penelitian ini memiliki hasil sebesar 60% pengguna menyatakan bahwa aplikasi ini sangat berguna dan sebesar 78% pengguna menyatakan bahwa aplikasi ini memiliki fitur yang menarik.

5. Randa Elqassas, Samy S. Abu-Naser (2018). *“Expert System for the Diagnosis of Mango Disease”*. ISSN: 2000-001X. Mangga adalah salah satu tipe buah yang tidak memiliki inti buah. Buah mangga adalah salah satu dari turunan mangier. Spesies ini mengandung berbagai macam tipe buah, salah satunya adalah buah tropis. Tanaman spesies ini dikenal sebagai tanaman budidaya. Mangga bukanlah tanaman yang baru ditemukan, melainkan mangga adalah tanaman yang sudah dikenal sejak dahulu sekitar ribuan tahun yang lalu yang ditemukan di Arab dengan nama *Anabj*. Akar tanaman ini dapat mencapai 6

meter, dan memiliki tinggi sekitar 35 meter dan 40 meter. Daunnya bewarna hijau dengan panjang daun sekitar 15 cm sampai 35 cm. Tanaman ini memiliki berbagai macam penyakit yang dapat mengancam produksinya. Dalam penelitian ini, mereka bertujuan untuk membangun sebuah sistem pakar untuk diagnosa penyakit pada tanaman mangga. Sistem pakar ini didesain dengan cara mengimplementasikan CLIPS dan menggunakan bahasa pemrograman Delphi. Beberapa petani, dan orang-orang yang tertarik dengan budidaya mangga dan pakar tanaman telah mencoba sistem pakar ini dan memberikan dampak yang positif.

## 2.5 Kerangka Pemikiran

Kerangka pemikiran adalah suatu diagram yang menjelaskan secara garis besar alur logika berjalannya sebuah penelitian. Kerangka pemikiran dibuat berdasarkan pertanyaan penelitian (*research question*), dan merepresentasikan suatu himpunan dari beberapa konsep serta hubungan diantara konsep-konsep tersebut (Polancik, 2009).



**Gambar 2.2** Kerangka Pemikiran  
Sumber Data Penelitian 2019

1. Peneliti menemukan gejala yang ditimbulkan akibat penggunaan gadget yang berlebihan pada penyakit mata. Data-data gejala tersebut dijadikan bahan masukan dalam penelitian ini.
2. Hasil input dari data tersebut akan diproses dalam knowledge base dengan menggunakan metode certainty factor. Tahap knowledge base juga dibantu dengan beberapa data yang didapat dari pakar spesialis mata.
3. Hasil dari penelitian ini adalah berupa sebuah aplikasi website yaitu sistem pakar menentukan jenis penyakit pada anak yang disebabkan oleh gadget.