

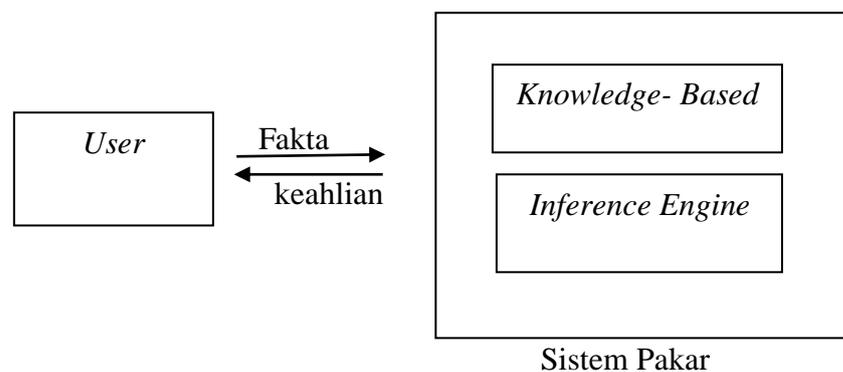
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Pakar

##### 2.1.1. Definisi Sistem Pakar

Menurut Arhami, M (2005) Sistem pakar adalah salah satu cabang dari AI (*Artificial Intelligenci*) penggunaan secara luas pengetahuan (*knowledge*) yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Sistem ini bekerja dengan menggunakan pengetahuan (*knowledge*) dan metode analisis yang telah didefinisikan terlebih dahulu oleh pakar yang sesuai dengan bidang keahliannya. Sistem ini disebut sistem pakar karena fungsi dan perannya sama seperti seorang ahli yang harus memiliki pengetahuan, pengalaman dalam memecahkan suatu persoalan. Sistem biasanya berfungsi sebagai kunci penting yang akan membantu suatu sistem pendukung keputusan atau sistem pendukung eksekutif. Sistem pakar terdiri dari dua komponen utama yaitu : basis pengetahuan (*knowledge base*) yang berisi pengetahuan dan *mesin inferensi* yang menggambarkan kesimpulan. Biasa pengetahuan didapat dari akumulasi pengetahuan pakar pada bidang tertentu.

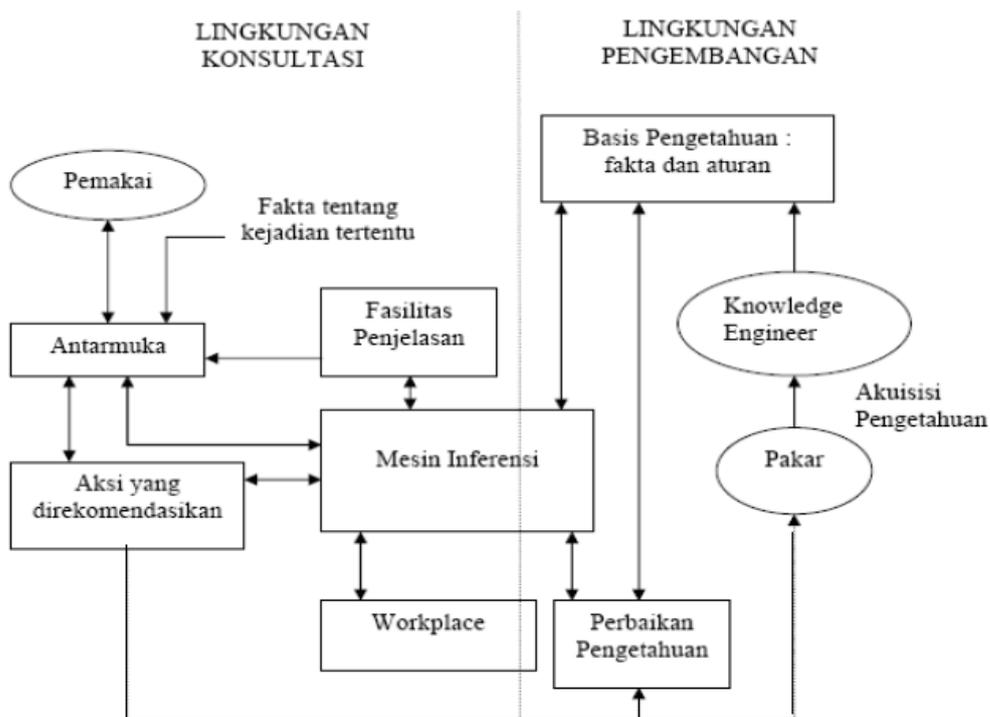


**Gambar 2.1** Konsep Dasar Fungsi Sistem Pakar

Pengetahuan disini didefinisikan sebagai kumpulan data dan himpunan aturan untuk memanipulasi atau mengolah data untuk menjadi pengetahuan baru. Basis pengetahuan merupakan komponen penting dari suatu sistem pakar, besar kecilnya kemampuan sistem pakar biasanya ditentukan oleh kapasitas dari basis pengetahuannya, sedangkan mesin pengambil keputusan adalah aplikasi yang membantu dan memandu pengguna sistem pakar dalam memanipulasi data dan memilih pengetahuan yang sesuai untuk mendapatkan kesimpulan.

### 2.1.2 Struktur Sistem Pakar

Komponen komponen yang terdapat dalam struktur sistem pakar ini yaitu *user interface*(antarmuka pengetahuan), basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan.



**Gambar 2.2** Struktur Sistem Pakar

1. Antarmuka Pengguna ( *User Interface* )  
Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem.
2. Basis Pengetahuan  
Mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan.
3. Akuisisi Pengetahuan  
Akuisisi Pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer.
4. Mesin Inferensi  
Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah.
5. *Workplace* (Tempat Kerja)  
Merupakan area dari sekumpulan memori kerja (*working memory*). Digunakan untuk merekam hasil antara dan kesimpulan yang dicapai.
6. Fasilitas Penjelasan  
Komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggunakan penalaran sistem kepada pemakai.
7. Perbaikan Pengetahuan  
Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya.

### 2.1.2 Karakteristik Sistem Pakar

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum berikut ini :

- a. Kinerja sangat baik
- b. Waktu respon yang baik
- c. Dapat diandalkan
- d. Dapat dipahami
- e. Fleksibel

## 2.2 Kanker Prostat

Menurut Sari,N.A (2007) Kelenjar prostat adalah salah satu kelenjar seks pada pria yang berukuran kecil, terletak di bawah kandung kemih dan mengelilingi saluran urin. Kelenjar ini berperan dalam membuat senyawa yang penting bagi pembentukan cairan semen. Selain itu kerja kelenjar prostat bersama dengan otot-otot kandung kemih adalah mengontrol aliran urin melewati uretra. Kanker prostat adalah penyakit yang menyerang kelenjar prostat. Sel-sel kelenjar prostat tumbuh abnormal dan tidak terkendali, sehingga mendesak dan merusak jaringan sekitarnya bahkan dapat mengakibatkan kematian.

Kadar *Prostate Specific Antigen* (PSA), usia, *Prostate Volume* (PV), kadar hormon testosteron, ras, dan riwayat keluarga dapat menjadi faktor resiko yang mempengaruhi terjadinya penyakit kanker prostat.

### 2.2.1 Faktor Resiko Kanker Prostat

Berikut ini adalah factor factor yang dapat mempengaruhi terjadinya kanker prostat :

- a. Kadar *Prostate Specific Antigen* (PSA)  
PSA adalah kadar protein spesifik yang terdapat pada kadar PSA yang tinggi dapat menjadi factor kemungkinan terjadinya kanker prostat.

b. *Prostate Volume (PV)*

Penderita kanker prostat memiliki *prostate volume* yang lebih tinggi dari pada keadaan normal. Hal ini disebabkan tumbuhnya sel-sel kelenjar prostat yang tumbuh secara abnormal.

c. Usia

Semakin lanjut usia seorang pria, maka resiko mengalami kanker prostat akan meningkat. Pada usia 80 tahun sebanyak 60-70% pria memiliki gambaran histology kanker prostat.

d. Faktor Hormonal

Peningkatan hormonal testosterone pada pria juga menjadi salah satu factor terjadinya kanker prostat.

e. Pola Makan

Perkembangan kanker diduga bisa dipengaruhi oleh pola makan seseorang. Namun hal ini belum bisa dijelaskan dengan pasti disebabkan oleh banyaknya factor lain.

f. Riwayat Keluarga

Resiko yang lebih besar bisa terjadi 2-3 kali lipat lebih besar pada pria yang satu generasi yang menderita kanker prostat.

### **2.2.2 Gejala Kanker Prostat**

Beberapa gejala yang ditimbulkan apabila menderita kanker prostat adalah :

- a. Terasa nyeri saat berkemih
- b. Terasa nyeri saat ejakulasi
- c. Terasa nyeri pada punggung
- d. Hematuria (ada darah pada air kemih)
- e. Nokturia (berkemih pada malam hari)

### 2.2.3 Diagnosa

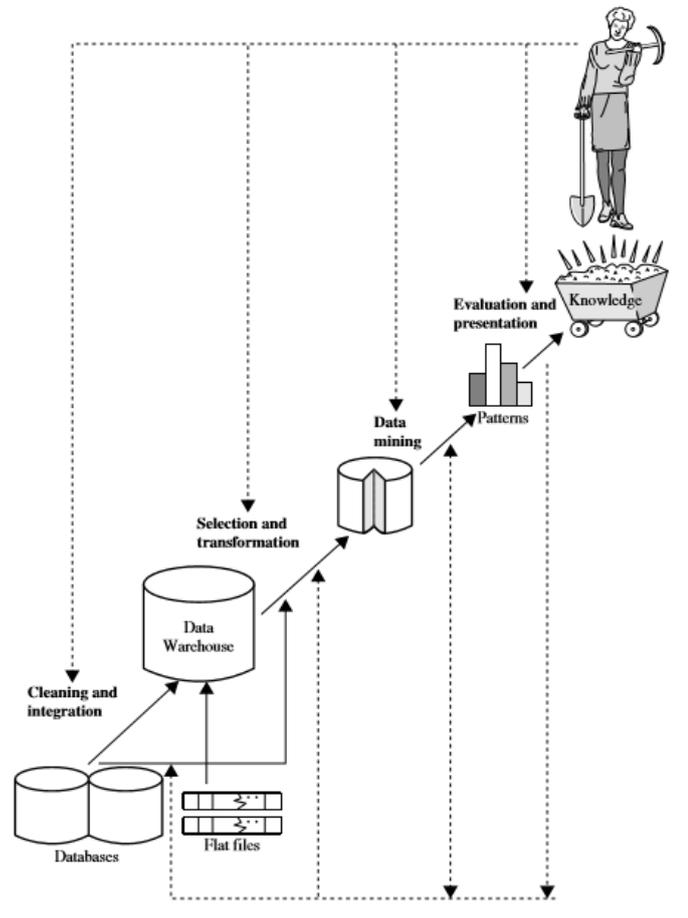
Diagnosa yang dapat dilakukan untuk mengetahui adanya penyakit kanker prostat yaitu dengan melakukan pemeriksaan PSA (*Prostate specific Antigen*) atau pemeriksaan colok dubur. PSA adalah pemeriksaan kadar PSA yang merupakan protein dalam darah. Protein tersebut dihasilkan oleh sel-sel kelenjar pada prostat. Kemungkinan menderita kanker prostat lebih besar apabila PSA semakin tinggi. Kadar normal PSA sedalam darah yaitu 0-4 ng/ml (nanogram per mililiter), dan kadar PSA untuk menentukan terjadinya kanker prostat pada seseorang adalah lebih dari 10 ng/ml. Selain PSA diagnosa dapat dilakukan dengan pemeriksaan colok dubur pada penderita kanker prostat yang dapat diamati dari adanya benjolan keras yang tidak beraturan.

### 2.3 Data Mining

Menurut Sulistiyanto (2018) Data mining merupakan sebuah istilah yang digunakan untuk menggambarkan sebuah kegiatan penemuan pengetahuan didalam sebuah kumpulan data. Data mining merupakan proses yang menggunakan teknik statistik, matematika, kecerdasan buatan dan pembelajaran berbasis mesin untuk mengekstrak dan mengidentifikasi informasi yang berguna beserta pengetahuan dari sebuah data. Setiap aplikasi kelas data mining dibantu oleh pendekatan sebuah algoritma untuk mengekstrak suatu hubungan dalam sebuah data. Perbedaan kelas pendekatan algoritma tersebut dapat membantu dalam menyelesaikan masalah yang ditemui. Beberapa kelas tersebut adalah:

1. *Clasifikasi* Mendefinisikan karakteristik dari sebuah grub atau membagi kedalam beberapa jenis yang diketahui.
2. *Clustering* Mengelompokkan data yang tidak diketahui label kelasnya kedalam sejumlah kelompok tertentu sesuai ukuran kemiripannya.
3. *Association* Mengidentifikasi hubungan antara kejadian yang terjadi dalam satu waktu. Pendekatan ini didasarkan pada analisis keranjang belanja.
4. *Sequencing* Sama dengan pendekatan association, namun yang membedakan hanya di waktu terjadinya. Sequence mengukur waktu berdasarkan periode tertentu.

5. *Regression* menemukan suatu fungsi yang memodelkan data dengan kesalahan prediksi (galat) seminimal mungkin.



**Gambar 2.3** Proses Data Mining Untuk Mendapatkan Pengetahuan

### 2.3.1 Algoritma C4.5

Menurut Sulistiyanto (2018) *Algoritma C4.5* atau sering disebut juga *Decision Tree* merupakan metode klasifikasi dan prediksi yang sangat terkenal. Metode ini mengubah fakta yang sangat besar menjadi pohon keputusan yang representasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Proses pada decision tree adalah mengubah bentuk data (tabel) menjadi model pohon, mengubah model pohon menjadi rule dan menyederhanakannya. Saat menyusun sebuah *decision tree* pertama yang harus dilakukan adalah menentukan atribut mana yang akan menjadi simpul akar dan atribut mana yang akan menjadi simpul selanjutnya. Pemilihan atribut yang baik adalah atribut yang memungkinkan untuk mendapatkan *decision tree* yang paling kecil ukurannya atau atribut yang bisa memisahkan objek menurut kelasnya. Untuk dapat membuat sebuah pohon keputusan, maka nilai entropinya harus di cari dahulu menggunakan persamaan:

$$Entropy(S) = - \sum_{i=1}^n p_i * \log_2 p_i$$

Keterangan:

- $S$  : himpunan kasus
- $n$  : jumlah partisi  $S$
- $p_i$  : proporsi dari  $S_i$ , terhadap  $S$

Contoh Perhitungan :

$$\begin{aligned} Entropy(S) &: \left(-\frac{11}{20}\right) \times \log_2 \left(-\frac{11}{20}\right) + \left(-\frac{9}{20}\right) \times \log_2 \left(-\frac{9}{20}\right) \\ &: (-0,55) \times \log_2 (0,55) + (-0,45) \times \log_2 (0,45) \\ &: 0,99 \end{aligned}$$

Setelah itu mencari nilai *information gain*. *Information Gain* merupakan kriteria yang paling populer untuk pemilihan atribut. *Information gain* dapat dihitung dari output data atau variabel dependen  $y$  yang dikelompokkan berdasarkan atribut  $A$ , dinotasikan dengan  $gain(y,A)$ .  $Gain(y,A)$  dari atribut  $A$  relatif terhadap output data  $y$  adalah:

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i)$$

Keterangan:

$S$  : himpunan kasus

$A$  : atribut

$n$  : jumlah partisi atribut  $A$

$|S_i|$  : jumlah kasus pada partisi ke- $i$

$|S|$  : jumlah kasus dalam  $S$

Contoh Perhitungan :

$$\begin{aligned} \text{Information Gain} & : 0,9927 - \left( \frac{10}{20} \times 0,7219 \right) + \left( \frac{10}{20} \times 0,8812 \right) \\ & : 0,9927 - ( 0,5 \times 0,7219 ) + ( 0,5 \times 0,8812 ) \\ & : (0,9927 - 0,36095) + 0,4406 \\ & : 1,07 \end{aligned}$$

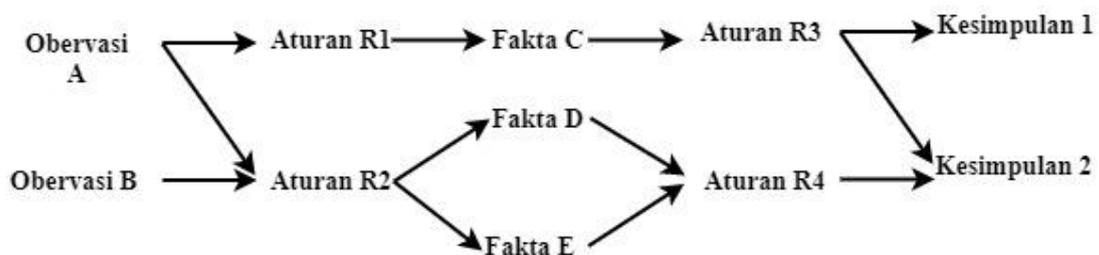
Setelah semua atribut dihitung menggunakan persamaan diatas, maka atribut yang memiliki nilai informasi tertinggi dibanding atribut yang lain dijadikan sebagai *node* (akar).

- a. Buat cabang, setelah diperoleh atribut yang mempunyai nilai *gain* tertinggi, maka atribut tersebut digunakan sebagai *node*. *Node* ini memiliki *instance* sehingga *instance* dijadikan sebagai cabang dari *node*.
- b. Bagi kasus dalam cabang, setiap nilai pada *instance* memiliki nilai yang berbeda. Nilai *instance* tersebut agar menjadi lebih sederhana. Tetapi, jika nilai *instance* tidak dapat disederhanakan lagi maka perlu melakukan perhitungan lebih lanjut.
- c. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

## 2.4 Mesin Inferensi

- a. Metode Inferensi Forward Chaining.

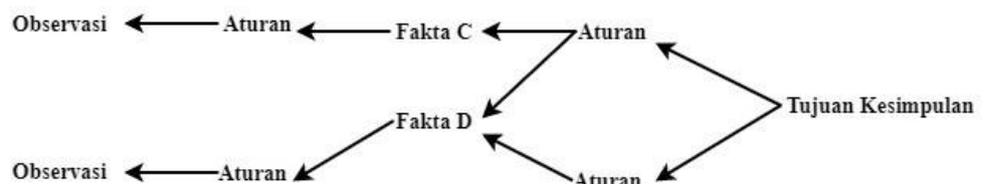
Metode Forward Chaining adalah suatu metode pengambilan keputusan yang aman digunakan dalam sistem pakar. Proses pencarian dengan metode Forward Chaining berangkat dari kiri ke kanan, yaitu dari premis menuju kepada kesimpulan akhir. Metode ini sering disebut data driven, yaitu pencarian dikendalikan oleh data yang diberikan.



**Gambar 2.4** Proses Forward Chaining

b. Metode Inferensi Backward Chaining.

Menurut Widyawati,A.C (2017) Metode *Backward Chaining* sesuai dengan namanya bekerja mundur dari asumsinya. Jika asumsinya q benar maka tak ada yang perlu dikerjakan selanjutnya. Proses pencarian dengan metode backward chaining berangkat dari kanan ke kiri, yaitu dari kesimpulan sementara menuju kepada premis, metode ini sering disebut gol driven pencarian dikendalikan oleh tujuan yang diberikan. Selain itu algoritmanya akan mencari implikasi-implikasi di dalam basis pengetahuan atau *Knowledge Based* (KB) yang kesimpulannya q. Jika premis-premis dari salah satu implikasi tersebut bisa dibuktikan benar, maka q adalah benar.



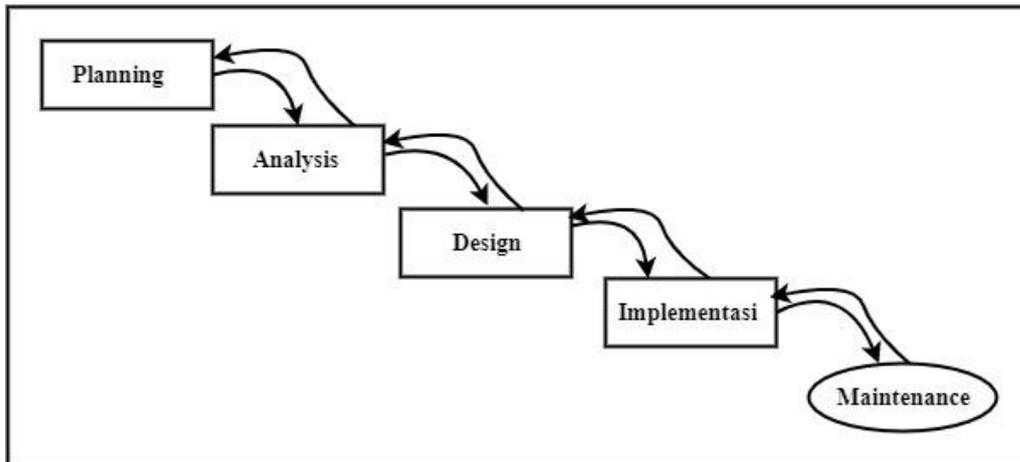
**Gambar 2.5** Proses *Backward Chaining*

## 2.5 Metode Pengembangan Perangkat Lunak

Metodelogi yang digunakan untuk membangun sistem ini adalah Metode *Waterfall* sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematis, dengan beberapa tahapan yaitu : *Planning, Analysis, Design, Implementation, dan maintenance*.

### 2.5.1 Waterfall

Menurut A.S., Rosa dan Shalahuddin, M. (2015) menjelaskan bahwa Model SDLC air terjun (*waterfall*) seing juga disebut model esksekunsiallinier aturan alur hidup klasik. Model air terjun menyediakan alur hidup perangkat lunak secara sekuensial atau tururt. Berikut ini adalah gambar model air terjun :



**Gambar 2.6** Sistem *Waterfall*

Penjelasan dari gambar 2.6 di atas adalah sebagai berikut :

a. Perencanaan (*Planning*)

Tahapan perencanaan merupakan proses penting untuk mengetahui mengapa sistem harus dibuat dan menentukan bagaimana cara membangun sistem tersebut.

b. *Analysis*

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user.

c. *Design*

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean.

d. Pengujian (*Implementation*)

Merupakan tahap berikutnya untuk menerjemahkan data atau pemecahan masalah yang telah dirancang kedalam bahasa pemrograman computer yang telah di tentukan.

e. Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

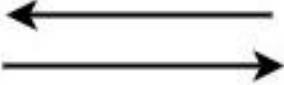
## **2.6 Unified Modeling Language (UML)**

Menurut Bee, Defit et al (2016) *Unified Modelling Language* (UML) merupakan bahasa untuk visualisasi, spesifikasi, konstruksi serta dokumentasi. UML juga merupakan bahasa yang sifatnya standart untuk menggambarkan suatu proses, seperti proses bisnis, dan lain-lain. Dalam kerangka visualisasi, para pengembang menggunakan UML untuk mengkomunikasikan idenya kepada para pemrogram atau calon pengguna sistem atau aplikasi tertentu. Ada banyak jenis model proses perangkat lunak yang menyarankan bagaimana untuk membangun perangkat lunak. model proses yang akan digunakan pada penelitian ini adalah *linear sequential*. *Linear Sequential* Model ini juga dikenal sebagai model air terjun atau siklus hidup klasik. Ini adalah model pertama yang pernah diformalkan, dan model proses lain didasarkan pada pendekatan pembangunan. Ini menunjukkan pendekatan yang sistematis dan sekuensial untuk pengembangan perangkat lunak.

### **2.6.1 Use Case Diagram**

Menurut Yuliawati,D,Saleh,S., dan Indera,I (2015) Use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Diagram use case bersifat statis, diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas).

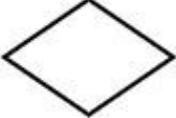
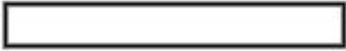
Tabel 2.1 Simbol Use Case Diagram

SIMBOL	KETERANGAN
 Actor	Mendefinisikan entitas diluar sistem yang memakai sistem
 Use Case	Gambaran fungsionalitas dari suatu sistem, sehingga pengguna mengerti kegunaan sistem yang akan dibangun
 Relasi	Menceritakan hubungan antara actor dan use case diagram dapat dipahami

### 2.6.2 Activity Diagram

Menurut Yuliawati,D,Saleh,S., dan Indera,I (2015) Aktivitas merupakan kumpulan aksi- aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah lagi. Aktivitas dapat mengakses atribut dan operasi dan operasi classifier, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku.

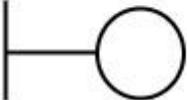
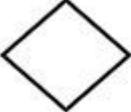
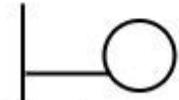
Tabel 2.2 Simbol Activity Diagram

SIMBOL	KETERANGAN
 Partition	Memperlihatkan dimana aliran berwal
 Relasi	Merupakan langkah atau aksi-aksi yang terjadi
 Decision	Memperlihatkan dimana keputusan perlu diambil selama terjadi aliran kerja
 Frok	Merupakan percabangan dari sebuah aksi-aksi yang terjadi
 Final Node	Memperlihatkan dimana aliran itu berakhir

### 2.6.3 Class Diagram

Menurut Dona Yuliawati, Sushanty Saleh dan Indera (2015) Class Diagram bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.

Tabel 2.3 Simbol Class Diagram

SIMBOL	KETERANGAN
 Generalization	Menggambarkan relasi generalisasi
 Realize	Menggambarkan relasi relasi
 Interface	Menambahkan kelas antarmuka (interface) pada diagram
 Return message	Menggambarkan pengembalian dari pemanggilan prosedur
 Aggregation	Menggambarkan relasi agregasi
 Actor	Menggambarkan aktor pada diagram kelas
 Boundary	Menambahkan kelas batasan (boundary) pada diagram

## **2.7 Perangkat Lunak Pengembangan Sistem**

### **2.7.1 Android Studio**

Menurut Juansyah, A (2015) Android studio adalah IDE (*Integrated Development Environment*) resmi untuk Jurnal Ilmiah Komputer dan Informatika. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (*Android Development Tools*).

Android studio memiliki fitur :

- a.) Proyek berbasis pada Gradle Build
- b.) Refactory dan pembenahan bug yang cepat
- c.) Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d.) Mendukung Proguard And App-signing untuk keamanan.
- e.) Memiliki GUI aplikasi android lebih mudah
- f.) Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan.

### **2.7.2 Android**

Menurut Sari, Y.P (2016) Android adalah sebuah sistem operasi untuk smartphone dan Tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasinya yang tersedia pada *device*.

## **2.8 Black-Box Testing (Pengujian Kotak Hitam)**

Menurut Pressman (2012) Metode pengujian pada aplikasi ini menggunakan *Black BoxTesting* yaitu yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black BoxTesting* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal hal yang tidak dicakup oleh *White Box Testing*.

*Black Box Testing* cenderung untuk menemukan hal hal berikut :

1. Fungsi tidak benar atau hilang.
2. Kesalahan interface atau antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan ferformansi.
5. Kesalahan inisialisasi dan terminasi.

## 2.9 Penelitian Terkait

Penelitian terdahulu terkait yang telah dilakukan dengan system pakar dengan *Algoritma C4.5* dan *Backward Chaining*.

**Table 2.3** Penelitian Terkait

No	Judul	Nama, Tahun	Keterangan
1	Diagnosa Penyakit Tanaman Kopi Dengan Menggunakan Backward Chaining Pada Kabupaten Tanggamus	(IIB DARMAJAYA Widyawati, A.C 2018.)	Penelitian ini bertujuan untuk menghasilkan system diagnosa sebagai alternative untuk mendiagnosa penyakit tanaman kopi di Kabupaten Tanggamus dengan Metode Backward Chaining. Penelitian ini menghasilkan sautu system pakar untuk mendiagnosa tanaman penyakit kopi, sehingga bisa mambantu warga Kabupaten Tanggamus untuk mengetahui tanaman kopinya terkena penyakit atau tidak.  Metode yang digunakan dalam penelitian ini adalah Backward

Lanjutan Tabel 2.3

			Chaining dengan proses pencarian berangkat dari kanan ke kiri, yaitu dari kesimpulan sementara menuju kepada premis.
2	Rancang Bangun Sistem Pakar untuk Deteksi Dini Katarak Menggunakan Algoritma C4.5	(Ivana Herliana W. Jayawardanu, Seng Hansun, 2015)	Penelitian ini bertujuan untuk mengembangkan system pakar untuk membantu meminimalisir ketidaktahuan tersebut. Dengan membangun sebuah sistem yang dapat mendeteksi katarak secara dini tanpa harus mengunjungi klinik mata. Metode yang digunakan dalam penelitian ini adalah algoritma C4.5 dari metode learning decision tree. Sistem pakar ini dibangun berdasarkan data data yang didapatkan dari form diagnose yang telah dikumpulkan dari dokter spesialis mata.
3	Sistem Pakar Diagnosa Penyakit Prostat Menggunakan Metode Certainty factor	(Prastika, I. A., Mulyanto, E., Informatika, J. T., & Nuswantoro, 2016)	Penelitian ini bertujuan untuk mengembangkan system untuk mendiagnosa penyakit sebagai media konsultasi untuk membantu masyarakat mendiagnosa gejala gejala awal dari penyakit kanker prostat.

Lanjutan table 2.3

			<p>Metode yang di gunakan peneliti adalah Metode Certainty Factor yang dapat digunakan untuk mendiagnosa penyakit prostat dengan menggunakan gejala – gejala yang akan menghasilkan prosentase nilai kepastian terhadap penyakit prostat yang di derita.</p>
4	<p>Penerapan Algoritma C4.5 Untuk Prediksi Penyakit Hepatitis</p>	<p>(Wisti Dwi Septiani, Maret 2014)</p>	<p>Penelitian ini bertujuan untuk mengembangkan system untuk memprediksi penyakit Hepatitis yang merupakan penyakit peradangan hati karena infeksi virus yang menyaerang kerusakan pada sel – sel dan fungsi organ hati. Metode yang di gunakan peneliti adalah Algoritma C4.5 merupakan salah satu metode dalam <i>decision tree</i>. <i>Decision tree</i> yang mengubah fakta yang besar menjadi pohon keputusan yang mempresnetasikan aturan, dalam penelitian ini akan dilakukan perhitungan dengan menentukan information gain atau entropy.</p>