

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Dalam melakukan Penelitian ini, peneliti merujuk kepada beberapa sumber ilmiah yang dijadikan sebagai bahan acuan yang terkait penerapan algoritma C4.5 (*Decision Tree*). Berikut beberapa penelitian yang sudah dilakukan sebelumnya :

Choirul Anam dan Harry Budi Santoso (2018) dengan judul penelitiannya “Perbandingan Kinerja Algoritma C4.5 dan Naïve Bayes untuk Klasifikasi Penerima Beasiswa”. Dalam penelitian ini dilakukan perbandingan kinerja Algoritma C4.5 dan *Naïve Bayes* yang bertujuan untuk mengukur tingkat akurasi dan lama waktu proses (*execution time*) dari masing – masing algoritma untuk mendapatkan algoritma terbaik yang akan diterapkan dalam membantu proses penentuan penerima beasiswa. Penelitian menggunakan data sekunder berupa daftar pemohon dan penerima beasiswa sebagai *dataset* yang memiliki enam faktor penentu, yaitu semester, IPK, prestasi ko/ekstra kurikuler, penghasilan orang tua, beban biaya listrik dan tanggungan orang tua. Pengujian dengan *10-fold cross validation* sekaligus evaluasi kinerja model menggunakan *tool RapidMiner*. Hasil penelitian menunjukkan tingkat akurasi model algoritma C4.5 sebesar 96,40% lebih baik dari tingkat akurasi model algoritma *Naïve Bayes* sebesar 95,11%, sedangkan waktu proses dari kedua model algoritma yang di teliti menunjukkan hasil 0s [1].

Khairul Zaman (2016) dengan judul penelitiannya "Penerapan Data Mining Menggunakan Algoritma C4.5 Untuk Menentukan Kelayakan Penerima Bantuan Rehabilitas Sosial Rumah Tidak Layak Huni (Studi Kasus Di Pemerintah Kabupaten Solok Selatan)". Pada penelitian ini, menggunakan parameter keputusan calon penerima bantuan rumah tidak layak huni yaitu status, pekerjaan, pendapatan per bulan, jumlah tanggungan, lantai rumah, bentuk fisik rumah, sarana air bersih dan MCK dengan menggunakan 30 data. Dapat disimpulkan bahwa implementasi algoritma C4.5 untuk melakukan prediksi terhadap calon penerima bantuan rehabilitas sosial rumah tidak layak huni (Studi Kasus di Pemerintah Kabupaten Solok Selatan) telah berhasil dilakukan dan sangat membantu serta mempercepat pengambilan keputusan dalam menentukan kelayakan penerima bantuan RSRTLH [2].

Dalam penelitian Erni Ermawati dan Taufik Hidayatulloh (2016) dengan judul "Penerapan Algoritma C4.5 Pada Sistem Penunjang Keputusan Penentuan Penerima Raskin ( Beras Masyarakat Miskin)". Dari penelitian yang telah dilakukan dapat ditarik kesimpulan mengenai penentuan penerima raskin dengan menggunakan algoritma C4.5 dengan atribut – atribut penentu yaitu luas lantai bangunan, jenis lantai bangunan, jenis dinding bangunan, fasilitas buang air besar, sumber penerangan, sumber air minum, bahan bakar masak, frekuensi konsumsi daging atau susu atau ayam perminggu, frekuensi pembelian pakaian baru, frekuensi makan dalam sehari, tidak mampu membayar untuk berobat, pekerjaan dengan pendapatan di bawah Rp 600.000,- /bulan, pendidikan KRT, tidak memiliki aset/tabungan/barang yang mudah di jual. Sistem pendukung keputusan

membantu para pengguna khususnya kepala seksi kesejahteraan rakyat (KASIKESRA) dan kepala desa (KADES) serta para perangkat desa dalam pengambilan keputusan tanpa harus melakukan rembuk desa yang berulang dan berkepanjangan. Hasil dari penggunaan algoritma C4.5 dalam penelitian ini, memprediksi masyarakat penerima RASKIN dengan entropy terbesar 0,9954 yakni petani dan gain terbesarnya adalah bahan bakar memasak 0,5000 serta dilengkapi dengan hasil uji kelayakan dengan begitu para pengguna akan lebih cepat dan tepat sasaran dalam menentukan penerima RASKIN [3]



**Tabel 2.1 Perbandingan Penelitian**

No.	Penelitian	Judul Penelitian	Tahun	Persamaan	Perbedaan
1.	Choirul Anam dan Harry Budi Santoso	Perbandingan Kinerja Algoritma C4.5 dan Naive Bayes untuk Klasifikasi Penerima Beasiswa	2018	- Implementasi Algoritma C4.5 sebagai algoritma pembentuk pohon keputusan ( <i>decision tree</i> ).	- Pada permasalahan di penelitian sebelumnya dilakukan perbandingan Algoritma C4.5 dan <i>Naive Bayes</i> untuk klasifikasi penerima beasiswa. - Pengujian dengan <i>10-fold cross validation</i> sekaligus evaluasi kinerja model menggunakan tool <i>RapidMiner</i> .
2.	Khairul Zaman	Penerapan Data Mining Menggunakan Algoritma C4.5 Untuk Menentukan Kelayakan Penerima Bantuan Rehabilitasi Sosial Rumah Tidak Layak Huni (Studi Kasus Di Pemerintah Kabupaten Solok Selatan)	2016	- Menggunakan Penerapan Algoritma C4.5 sebagai algoritma pembentuk pohon keputusan ( <i>decision tree</i> ).	- Pada penelitian sebelumnya menyelesaikan permasalahan klasifikasi untuk menentukan kelayakan penerima bantuan RSRTLH pada Dinas Sosial Ketenagakerjaan dan Transmigrasi Kabupaten Solok Selatan. - Peneliti sebelumnya menggunakan <i>Software Weka</i> dan <i>RapidMiner</i> untuk proses perhitungan atau mengolah data.

3.	Erni Ermawati dan Taufik Hidayatulloh	Penerapan Algoritma C4.5 Pada Sistem Penunjang Keputusan Penentuan Penerima Raskin ( Beras Masyarakat Miskin)	2016	<ul style="list-style-type: none"> <li>- Implementasi Algoritma C4.5 sebagai algoritma pembentuk pohon keputusan (decision tree)</li> <li>- Pada pengujian akurasi algoritma menggunakan model <i>Confusion Matrix</i></li> </ul>	<ul style="list-style-type: none"> <li>- Pada studi kasus penelitian sebelumnya yaitu pengelompokan calon siswa baru berdasarkan penjurusan dengan output prediksi masuk jurusan RPL dan TKJ, dengan menggunakan sampel data sebanyak 30 calon siswa dan kriteria parameter kelulusannya menggunakan tes minat dan bakat, sedangkan pada penelitian penulis outputnya terbagi menjadi dua pengelompokan kelayakan yaitu diterima dan tidak diterima</li> </ul>
----	---------------------------------------	---	------	---	--

## 2.2 Dasar Teori

### 2.2.1 Konsep Dasar Sistem

Ada dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih mengedepankan pada prosedur mendefinisikan bahwa sistem adalah suatu jaringan kerja pada prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

Pendekatan sistem yang lebih mengedepankan pada elemen atau komponennya mendefinisikan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu [4].

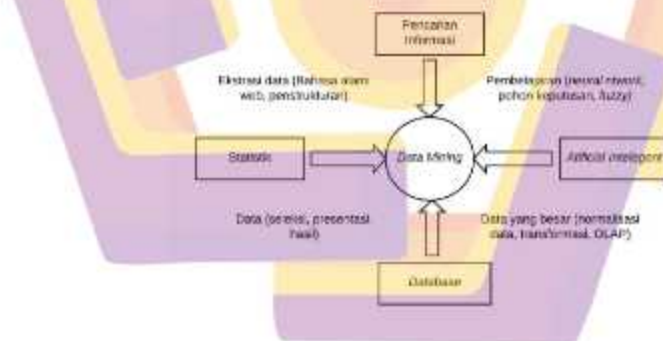
### 2.2.2 Pengertian Sistem Informasi

Informasi dapat diperoleh dari sistem informasi (*Information System*) atau disebut juga dengan *Processing System* atau *Information Generating System*. Menurut Robert A. Leitch dan K. Roscoe Davis sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [4].

### 2.2.3 Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database. Data mining adalah proses menggunakan teknik statistik, matematika, kecerdasan buatan dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar. Dapat dilihat digambar 2.1 Bidang ilmu data mining [6].

Menurut Garnet Group Data Mining adalah suatu proses menemukan hubungan yang berarti, pola dan kecenderungan dengan memeriksa dengan sekumpulan besar data yang tersimpan dalam penyimpanan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika. [6].



Gambar 2.1 Bidang Ilmu Data Mining

### 2.2.4 Tahapan Proses KDD dalam Data Mining

Istilah *data mining* dan *knowledge discovery in database (KDD)* sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi

tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan dengan satu sama lain, dimana salah satu tahapan dalam keseluruhan proses KDD adalah data mining. Proses KDD dijelaskan sebagai berikut: [6]

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, kita memilih data-data seperti apa saja yang kita butuhkan untuk diproses lebih lanjut dan kemudian data disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing (Cleaning)*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data dan informasi eksternal.

3. *Transformation*



Pada tahap ini dilakukan pemilihan data yang diperlukan oleh teknik *data mining* yang dipakai. Transformasi dan pemilihan data ini juga menemukan kualitas dari hasil *data mining* nantinya karena ada beberapa karakteristik dari teknik-teknik *data mining* tertentu. Contohnya *coding*, *coding* adalah proses transformasi pada data yang dipilih, sehingga data tersebut sesuai dengan proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

#### 4. *Data Mining*

*Data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. teknik, metode dan algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan. Semakin banyak data yang digunakan semakin akurat dan semakin sedikit kesalahan (*error*) maka semakin bagus model yang dihasilkan.

#### 5. *Interpretation (Evaluation)*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau

informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

### 2.2.5 Pengelompokan Data Mining

*Data Mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu [6] :

#### 1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

#### 2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variable target estimasi lebih kearah numerik daripada kearah kategori. Model dibangun menggunakan record lengkap yang menyediakan nilai dari variable target sebagai nilai prediksi.

#### 3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

#### 4. Klasifikasi

Dalam klasifikasi, terdapat target variable kategori. Contoh klasifikasi dalam bisnis dan penelitian sebagai berikut :

1. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi curang atau bukan
2. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk
3. Mediagnosis penyakit seorang pasien untuk mendapatkan termasuk dalam kategori penyakit apa

#### 5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan untuk membentuk kelas-kelas objek yang memiliki kemiripan yang satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain.

#### 6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

Contoh asosiasi dalam bisnis dan penelitian adalah

1. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
2. Menemukan barang didalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

### 2.2.6 Pohon Keputusan (Decision Tree)

Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Dan bisa diekspresikan dalam bentuk Bahasa basis data seperti *Structured Query language* untuk mencari *record* pada kategori tertentu [6].

Pohon keputusan berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon *variable* input dengan sebuah *variable* target. Karena pohon keputusan memadukan antara eksplorasi data dan pemodelan, maka sangat bagus sebagai langkah awal dalam proses pemodelan bahkan ketika dijadikan sebagai akhir dari beberapa teknik lain.

Sebuah pohon keputusan adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Dengan masing-masing rangkaian pembagian, anggota himpunan hasil menjadi mirip satu dengan yang lainnya [6].

### 2.2.7 Algoritma C4.5

Algoritma C4.5 merupakan algoritma yang digunakan untuk membentuk pohon keputusan [7]. Didalam algoritma C4.5 ini, pohon-pohon keputusan yang dibentuk berdasarkan kriteria-kriteria pembentuk keputusan. Didalam penyelesaian sebuah kasus menggunakan algoritma C4.5 ada beberapa elemen yang harus dipahami yaitu:

1. *Entropy*

2. *Info Gain*
3. *Split Info*
4. *Gain Ratio*

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut :

1. Pilih atribut sebagai akar
2. Buat cabang untuk tiap-tiap nilai
3. Bagi kasus dalam cabang
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *Gain Ratio* tertinggi dari atribut-atribut yang ada. Untuk menghitung nilai *Gain Ratio*, digunakan rumus seperti pada persamaan berikut.

*Entropy* digunakan untuk menentukan seberapa informatif sebuah atribut *input* untuk menghasilkan atribut *output*.

Perhitungan nilai *entropy* dapat dilihat dari persamaan berikut :

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

(2.1)

Keterangan :

S : himpunan kasus

A : fitur

$n$  : jumlah partisi  $S$

$P_i$  : proporsi dari  $S_i$  terhadap  $S$

Untuk menghitung *gain* digunakan rumus seperti tertera dalam persamaan berikut :

$$Gain(S, A) = Entropy(s) - \sum_{i=1}^n \frac{|S_i|}{|S|} \cdot Entropy(S_i) \quad (2.2)$$

Keterangan :

$S$  : himpunan kasus

$A$  : atribut

$n$  : jumlah partisi atribut  $A$

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

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

Untuk mencari nilai *Split Info* digunakan rumus berikut ini.

$$Split\ Info(S, A) = \sum_{i=1}^n \frac{s_i}{s} \cdot \log_2 \frac{s_i}{s_i} \quad (2.3)$$

Untuk mencari nilai *Gain Ratio* digunakan rumus berikut ini.

$$Gain\ Ratio(A) = \frac{Gain(A)}{Split\ Info(A)} \quad (2.4)$$

Untuk memudahkan penjelasan mengenai algoritma C4.5, berikut ini disertakan contoh kasus yang dituangkan dalam Table 2.2.

**Tabel 2.2 Keputusan bermain Tenis**

NO	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	Sunny	Hot	High	False	No

2	Sunny	Hot	High	True	No
3	Cloudy	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	Yes
7	Cloudy	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Cloudy	Mild	High	True	Yes
13	Cloudy	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Berikut ini adalah penjelasan lebih terperinci mengenai tiap-tiap langkah dalam pembentukan pohon keputusan dengan menggunakan algoritma C4.5 untuk menyelesaikan permasalahan pada table 2.2.

Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus dari keputusan *No*, dan *Entropy* dari semua kasus. Kasus yang dibagi berdasarkan atribut *Outlook*, *Temperature*, *Humidity* dan *Windy*. Setelah itu lakukan perhitungan *Gain* untuk setiap atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.3.

**Tabel 2.3 Perhitungan Node 1**

Node	Jml kasus (S)	Tidak Ya ( $S_2$ ) (Si)	Entropy	Gain
------	---------------	-------------------------	---------	------

1	Total	14	4	10	0.863120569	
	Outlook					0.258521037
	Cloudy	4	0	4		
	Rainy	5	1	4	0.721928095	
	Sunny	5	3	2	0.970950594	
	Temperature					0.183850925
	Cool	4	0	4	0	
	Hot	4	2	2	1	
	Mild	6	2	4	0.918295834	
	Humidity					0.370506501
	High	7	4	3	0.985228136	
	Normal	7	0	7	0	
	Windy					0.005977711
	False	8	2	6	0.811278124	
	True	6	4	2	0.918295834	

Baris Total kolom *Entropy* pada Table 2.3 dihitung dengan persamaan sebagai berikut:

$$Entropy(Total) = \left(-\frac{4}{14} \cdot \log_2 \left(\frac{4}{14}\right)\right) + \left(-\frac{10}{14} \cdot \log_2 \left(\frac{10}{14}\right)\right)$$

$$Entropy(Total) = 0.863120569$$

Sementara itu, nilai *Gain* pada baris *Outlook* dihitung dengan menggunakan persamaan sebagai berikut:

$$Gain(Total, Outlook) = Entropy(Total) - \sum_{i=1}^n \frac{|Outlook_i|}{|Total|} \cdot Entropy(Outlook_i)$$

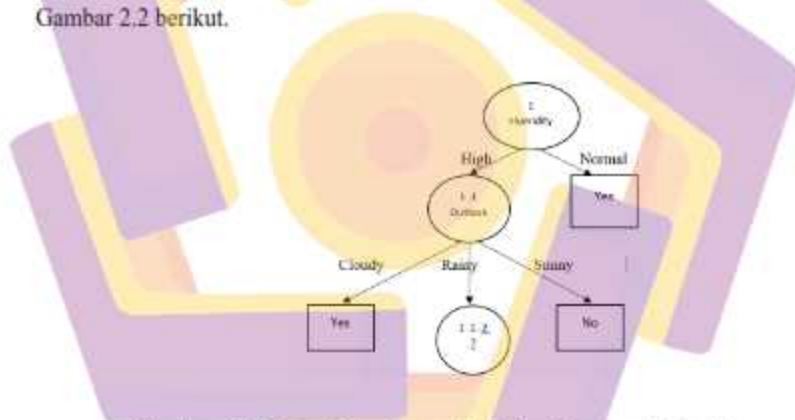
$$Gain(Total, Outlook) = 0.863120569 - \left(\left(\frac{4}{14}\right) \cdot 0\right) + \left(\frac{5}{14} \cdot 0.723\right) + \left(\frac{5}{14} \cdot 0.97\right)$$



$$Gain(Total, Outlook) = 0.23$$

Dari hasil pada Table 2.3 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah *Outlook*, yaitu sebesar 0.67, dengan demikian tiga nilai atribut dari *Outlook*, yaitu *Cloudy*, *rainy*, dan *Sunny*. Dari ketiga nilai atribut tersebut, nilai atribut *Cloudy* sudah mengklasifikasikan kasus menjadi 1, yaitu keputusan *yes* dan nilai atribut *Sunny* sudah mengklasifikasikan kasus menjadi satu dengan keputusan *No*, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut *Rainy* masih perlu dilakukan perhitungan lagi.

Pohon keputusan yang terbentuk sampai tahap ini ditunjukkan pada Gambar 2.2 berikut.



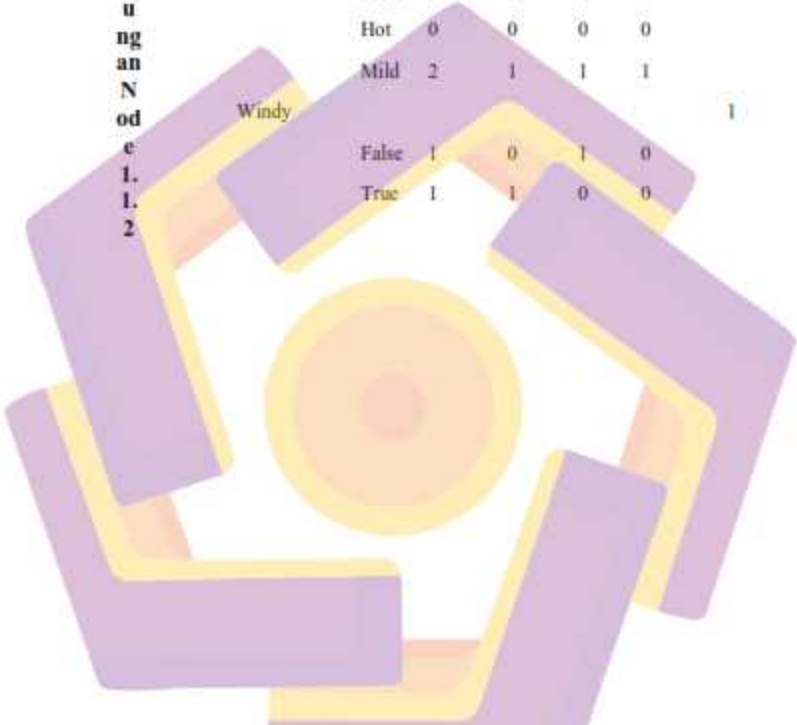
**Gambar 2.2 Pohon Keputusan Hasil Perhitungan Node 1.1**

Menghitung jumlah kasus, jumlah kasus untuk keputusan *Yes*, jumlah kasus untuk keputusan *No*, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut *Temperature* dan setelah itu dilakukan perhitungan *Gain* untuk tiap-tiap atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.4.

Node	Jumlah kasus	Tidak (S1)	Ya (S2)	Entropy	Gain
------	--------------	------------	---------	---------	------

T  
a  
b  
e  
l  
2.  
4  
P  
e  
r  
h  
i  
t  
u  
n  
g  
a  
n  
N  
o  
d  
e  
1.  
1.  
2

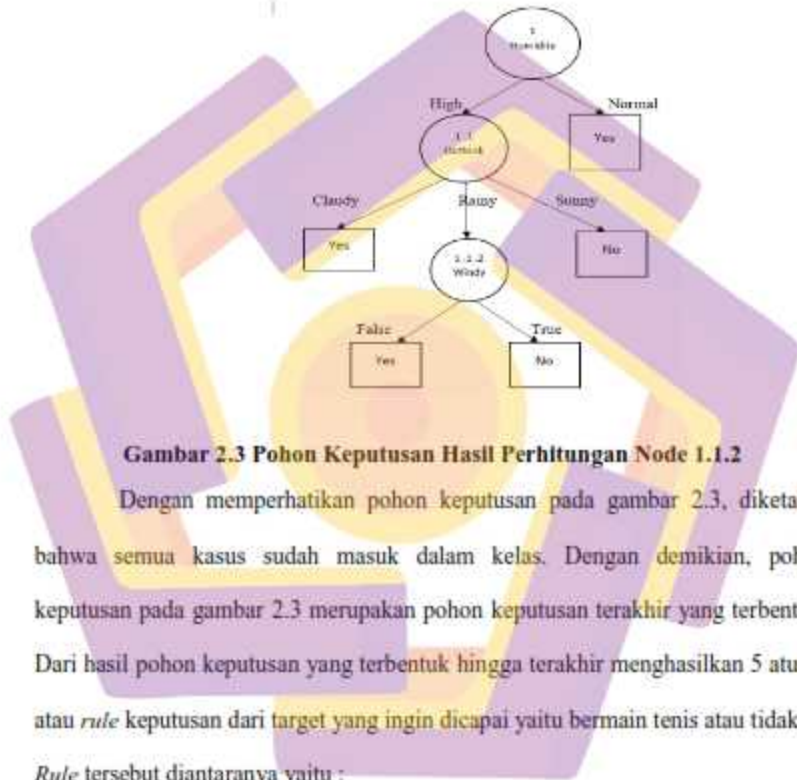
		(s)				
1.1.2	Humidity-high dan Outlook-Rainy	2	1	1	1	
	Temperature					0
	Cool	0	0	0	0	
	Hot	0	0	0	0	
	Mild	2	1	1	1	
	Windy					1
	False	1	0	1	0	
	True	1	1	0	0	



Dari hasil pada Tabel 2.4 dapat diketahui bahwa atribut dengan Gain tertinggi adalah *Windy*, yaitu sebesar 1. Dengan demikian *Windy* dapat menjadi node cabang dari nilai atribut *Rainy*. Ada dua nilai atribut dari *Windy*, yaitu *False* dan *True*. Dari kedua nilai atribut tersebut nilai atribut *False* sudah mengklasifikasikan kasus menjadi 1, yaitu keputusannya *Yes* dan nilai atribut

*True* sudah mengklasifikasikan kasus menjadi satu dengan keputusan *No*, sehingga tidak perlu dilakukan perhitungan lebih lanjut untuk nilai atribut ini.

Berikut hasil pohon keputusan perhitungan Node 1.1.2 ditunjukkan pada gambar 2.3.



**Gambar 2.3 Pohon Keputusan Hasil Perhitungan Node 1.1.2**

Dengan memperhatikan pohon keputusan pada gambar 2.3, diketahui bahwa semua kasus sudah masuk dalam kelas. Dengan demikian, pohon keputusan pada gambar 2.3 merupakan pohon keputusan terakhir yang terbentuk. Dari hasil pohon keputusan yang terbentuk hingga terakhir menghasilkan 5 aturan atau *rule* keputusan dari target yang ingin dicapai yaitu bermain tenis atau tidak. *Rule* tersebut diantaranya yaitu :

1. Jika humidity normal maka bermain tenis.
2. Jika humidity high dan outlook sunny maka tidak bermain tenis.
3. Jika humidity high dan outlook cloudy maka bermain tenis.
4. Jika humidity high dan outlook rainy dan angin kecil maka bermain tenis.

5. Jika humidity high, outlook rainy dan windy true maka tidak bermain tenis.

Dari aturan-aturan yang telah dihasilkan nantinya akan digunakan untuk mencocokkan kasus pada atribut target yang diinputkan. Tiap record akan dikoreksi statusnya dengan aturan yang telah ada sehingga dapat diklasifikasikan dalam satu kelas berdasarkan target yang ingin diketahui.

## **2.2.8 Model Pengembangan Waterfall**

### **2.2.8.1 Pengertian Model Waterfall**

Waterfall model merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, dan evolusi, dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi, pengujian unit, integrasi sistem, pengujian sistem, operasi dan pemeliharaan [8].

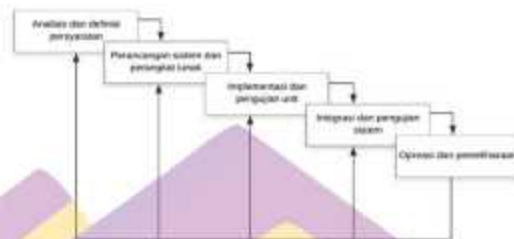
### **2.2.8.2 Tahapan Pengembangan Waterfall**

Tahapan pada waterfall model mengambil kegiatan dasar yang digunakan dalam hampir semua pengembangan perangkat lunak, sehingga dapat lebih mudah untuk dipahami terlebih bila hanya digunakan dalam mengembangkan perangkat lunak yang tidak begitu besar dan kompleks.

Tahap-tahap utama dari waterfall model memetakan kegiatan-kegiatan pengembangan dasar, yaitu:[8]

1. Analisis dan definisi persyaratan, merupakan proses mendefinisikan secara rinci mengenai fungsi-fungsi batasan dan tujuan dari perangkat lunak sebagai spesifikasi sistem yang akan dibuat.
2. Perancangan sistem dan perangkat lunak adalah proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya.
3. Implementasi dan pengujian unit, pada tahap ini perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Kemudian pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya.
4. Integrasi dan pengujian sistem merupakan unit program/program individual yang diintegrasikan menjadi sebuah kesatuan sistem dan kemudian dilakukan pengujian. Dengan kata lain, pengujian ini ditujukan untuk menguji keterhubungan dari tiap-tiap fungsi perangkat lunak untuk menjamin bahwa persyaratan sistem telah terpenuhi. Setelah pengujian sistem selesai dilakukan, perangkat lunak dikirim ke pelanggan/user.
5. Operasi dan pemeliharaan adalah tahap sistem diterapkan (diinstall) dan dipakai. Pemeliharaan mencakup koreksi dari beberapa kesalahan yang tidak diketemukan pada tahapan sebelumnya, perbaikan atas

implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan.





**Gambar 2.4 Tahap Pengembangan Waterfall**

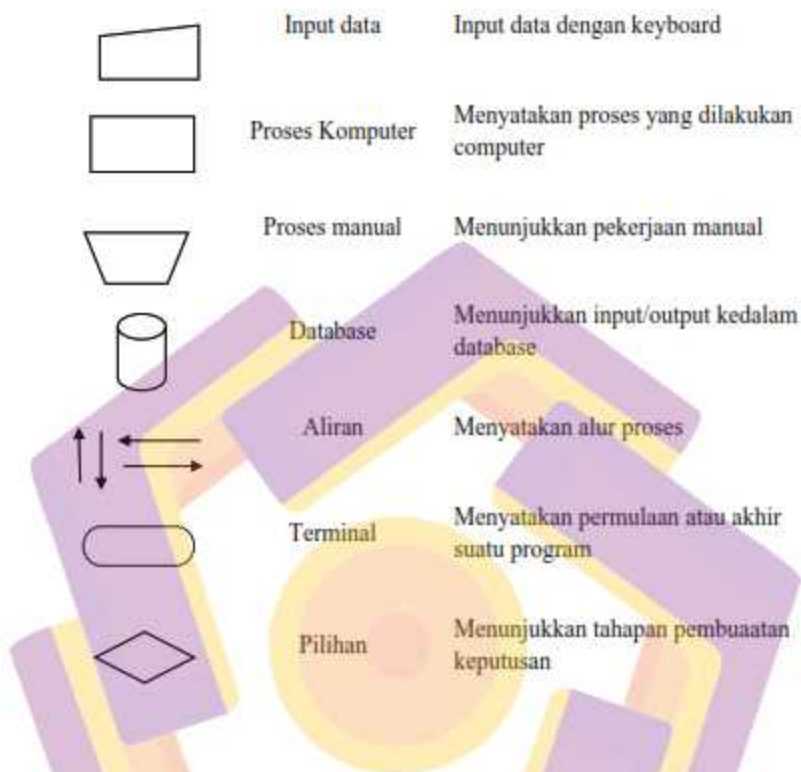
## 2.2.9 Perancangan Sistem

### 2.2.9.1 Flowchart

*Flowchart* atau bagan alur merupakan metode untuk menggambarkan tahap - tahap penyelesaian masalah (prosedur) beserta aliran data dengan simbol - simbol standar yang mudah dipahami. Tujuan utama *flowchart* adalah untuk menyederhanakan rangkaian proses atau prosedur untuk memudahkan pemahaman pengguna terhadap informasi tersebut. Untuk itu simbol - simbol *flowchart* dilihat pada tabel 2.5 [9].

**Tabel 2.5 Simbol Flowchart**

Simbol	Nama	Keterangan
	Dokumen	Digunakan untuk mencetak keluaran dalam bentuk dokumen
	Input/output	Digunakan untuk mewakili data input/output



### 2.2.9.2 Data Flow Diagram (DFD)

*Data Flow Diagram* adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. Simbol *Data Flow Diagram* menurut Jourdan dan DeMarco dapat dilihat pada tabel 2.6 [9].

Tabel 2.6 Simbol Data Flow Diagram (DFD)

Simbol	Nama Simbol	Keterangan
	Terminator	Mewakili entitas luar yang berkomunikasi dengan sistem yang dikembangkan
	Proses	Menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.
	Data Store	Digunakan untuk membuat model sekumpulan paket data dan diberi nama dengan kata benda jamak, misalnya Mahasiswa
	Alur Data	Menunjukkan arah menuju ke dan keluar dari suatu proses.

#### 2.2.10 Sistem Basis Data

Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional (dengan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses / pekerjaan tertentu [10].

Basis data adalah kumpulan data yang saling berhubungan yang disimpan diorganisasi secara bersama, dalam bentuk sedemikian rupa dan tanpa redundansi (pengulangan) yang tidak perlu, supaya dapat dimanfaatkan kembali dengan cepat dan mudah untuk memenuhi berbagai kebutuhan. Penggunaan basis data ditujukan untuk mempermudah penyimpanan, pencarian, pengelompokan data,



keamanan, ketersediaan data dan kebersamaan pemakai (*shareability*) suatu informasi [10].

Sistem basis data dapat terbagi dalam beberapa komponen penting, yaitu [10]:

1. Data

Merupakan informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.

2. *Hardware*

Merupakan perangkat keras berupa komputer dengan media penyimpanan sekunder yang digunakan untuk menyimpan data karena pada umumnya basis data memiliki ukuran yang besar.

3. Sistem Operasi

Program yang mengaktifkan / memfungsikan sistem komputer, mengendalikan seluruh sumber daya komputer dan melakukan operasi-operasi dasar dalam komputer yang meliputi operasi input / output, pengelolaan file dan sebagainya.

4. Basis Data

Basis data sebagai inti dari sistem basis data. Basis data menyimpan data serta struktur sistem basis data baik untuk entitas maupun objek-objeknya secara detail.

5. *Database Management System (DBMS)*

Merupakan perangkat lunak yang digunakan untuk melakukan pengelolaan basis data.

6. *User*

Merupakan pengguna yang menggunakan data yang tersimpan dan dikelola. *User* dapat berupa seseorang yang mengelola basis data tersebut, yang disebut *Database Administrator (DBA)*, bisa juga *end user* yang mengambil hasil dari pengelolaan basis data melalui *query*.




#### 7. Aplikasi lain

Program yang dibuat untuk memberikan *interface* kepada *user* sehingga lebih mudah dan terkontrol dalam mengakses basis data.

#### 2.2.10.1 Entity Relationship Diagram (ERD)

Menurut Al Fatta, *Entity Relationship Diagram* adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Adapun simbolnya sebagai berikut:[9]

Tabel 2.7 Simbol – Simbol ERD

Simbol	Nama	Keterangan
	Entitas	Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi	Untuk menunjukkan adanya hubungan antara sejumlah entitas yang ada
	Atribut	Atribut merupakan sebutan untuk mewakili suatu entitas.
	Garis	Sebagai penghubung antar relasi dengan entitas, relasi dan entitas dengan atribut.

Dalam *Entity Relation Diagram* hubungan (relasi) dapat terdiri dari sejumlah entitas yang disebut dengan derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas sedangkan derajat minimum disebut dengan modalitas. Jadi kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain.

Hubungan antara dua entitas dapat dikategorikan menjadi tiga macam, sebagai berikut:[9]

1. *One to One Relationship*

Hubungan antara entitas pertama dan kedua adalah satu berbanding satu.

2. *One to Many Relationship*

Hubungan antara entitas pertama dan kedua adalah satu berbanding banyak.

3. *Many to Many Relationship*

Hubungan kedua entitas adalah banyak berbanding banyak.

### 2.2.11 Confusion Matrix 2.2.9

Pada data mining untuk mengukur kinerja dari model yang dihasilkan salah satunya menggunakan *Confusion Matrix*. *Confusion Matrix* adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep *data mining*. Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya diprediksi positif secara benar [6].

Tabel 2.8 Model Confusion Matrix

Predicted Class	Actual Class	
	+	-
+	True Positives (TP)	False Positives (FP)
-	False Negatives (FN)	True Negatives (TN)

Perhitungan akurasi dengan tabel *Confusion Matrix* adalah sebagai berikut.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN}$$

(2.5)

Presisi didefinisikan sebagai rasio item relevan yang dipilih terhadap semua item yang terpilih. Presisi dapat diartikan sebagai kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut. Rumus presisi adalah:

$$Presisi = \frac{TP}{TP+FP}$$

(2.6)

Recall didefinisikan sebagai rasio dari item relevan yang dipilih terhadap total jumlah item relevan yang tersedia. Recall dihitung dengan rumus:

$$Recall = \frac{TP}{TP+FN}$$

(2.7)

Presisi dan Recall dapat diberi nilai dalam bentuk angka dengan menggunakan perhitungan persentase (1-100%) atau dengan menggunakan bilangan antara 0-1. Sistem rekomendasi akan dianggap baik jika nilai presisi dan recallnya tinggi.

### 2.2.12 Pengertian Web

Web adalah kumpulan halaman yang diletakkan dalam satu tempat /situs/site yang di dalamnya terdapat gambar video, dan file digital lain yang diletakkan dalam satu tempat yang didefinisikan melalui nam domain (*domain name*) dan alamat IP (*IP address*). WWW memiliki 3 bagian yang digunakan, diantaranya adalah sebagai berikut [11]:

1. Adanya penggunaan HTTP (*Hypertext Transfer Protocol*) sehingga antar jaringan yang satu dengan yang lainnya dapat melakukan komunikasi serta melakukan transfer data.
2. Adanya URL (*Uniform Resource Locatur*) yang digunakan untuk menentukan penamaan dari alamat yang ada di web atau dikenal dengan istilah *adres* (alamat di web).
3. Menggunakan HTML (*Hypertext Markup Language*) yaitu Bahasa pemrograman yang umum digunakan untuk mendukung akses web melalui internet dikenal dengan nama dokumen html dan berektensi dot html (.html).

### 2.2.13 Pengertian PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Disebut bahasa pemrograman *script server-side* karena php diproses pada komputer server. PHP banyak dipakai untuk membuat proram situs web dinamis [11].

PHP bersifat *open source* yang dirilis dalam lisensi PHP *license*, sedikit berbeda dengan lisensi GNU *General Public License* (GPL) yang bisa dignakan

dengan proyek *Open Source*. Kemudahan dan kepopuleran PHP sudah menjadi standar bagi pemrograman web di seluruh dunia.

Salah satu kelebihan PHP adalah kemudahannya untuk berinteraksi dengan database. PHP dapat mendukung beberapa *database* secara langsung tanpa menginstal konektor. Dengan demikian, PHP sangat fleksibel berhubungan dengan berbagai database [11].

#### 2.2.14 MySQL

MySQL adalah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*) [12].

Selain itu MySQL memiliki kelebihan dibandingkan dengan database lain, di antaranya adalah [12] :

1. MySQL sebagai *Database Management System* (DBMS).
2. MySQL sebagai *Relation Database Management System* (RDBMS).
3. MySQL adalah sebuah *Software Database* yang *Open Source*, artinya program ini bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatannya.
4. MySQL merupakan sebuah *database server*, jadi dengan menggunakan *database* ini anda dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
5. MySQL merupakan sebuah *database client*. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *query*

yang mengakses database pada server. Jadi MySQL juga dapat berperan sebagai *Client*.

6. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.

