

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Marini Utami Putri dan rekan-rekannya, dipelajari mengenai Prediksi Penjualan Laptop menggunakan Metode pemecahan C4.5. Dalam penelitian tersebut, Marini Utami Putri dan timnya melakukan analisis terhadap data transaksi di toko Computer Square dengan tujuan untuk melakukan prediksi penjualan laptop di masa mendatang menggunakan metode pemecahan C4.5. Dalam mengumpulkan data, Marini Utami Putri dan rekan-rekannya melakukan observasi eksklusif dan wawancara pribadi dengan pihak terkait, termasuk pimpinan kawasan perjuangan dan karyawan, menggunakan pendekatan kualitatif. Dalam implementasi sistemnya, digunakan bahasa pemrograman *Python* dengan menggunakan *Anaconda Prompt* dan *Text Editor Jupyter Notebook*. Fokus utama dari penelitian yang dilakukan oleh Marini Utami Putri dan timnya adalah membantu toko atau organisasi dalam mengestimasi permintaan produk mereka [1].

Penelitian selanjutnya telah dilakukan oleh Aprilianus Kristianus Lalo dan rekan-rekannya yang membahas mengenai Implementasi metode C4.5 untuk pembagian terstruktur dalam konteks Penjualan Barang di Toko pasar swalayan Dutalia. Terdapat enam atribut yang digunakan, yaitu jenis barang, harga, jumlah penjualan, waktu penjualan, dan status penjualan sebagai atribut target. Metode pengumpulan data yang digunakan dalam penelitian ini adalah metode lapangan yang juga dapat disebut sebagai metode kualitatif. Berdasarkan pengujian

menggunakan data pelatihan dan data prediksi di aplikasi *RapidMiner Studio* Versi 9.7, diperoleh tingkat akurasi perhitungan sebesar 100%. Akibatnya, sistem yang dirancang mampu digunakan untuk memprediksi penjualan barang yang akan datang [2].

Persamaan penelitian yang dilakukan oleh penulis dengan penelitian yang dilakukan oleh Marini Utami Putri dkk dan penelitian yang dilakukan oleh Aprilianus Kristianus Lalo dkk adalah metode pengumpulan data yang dilakukan menggunakan metode kualitatif. Selain itu persamaan lainnya terdapat pada tujuan pada penelitian, yaitu untuk membantu pihak perusahaan dalam menjalankan proses bisnisnya. Sedangkan perbedaan penelitian yang dilakukan penulis dengan penelitian yang dilakukan oleh Marini Utami Putri dkk yaitu penulis menggunakan bahasa pemrograman *Python* menggunakan *Anaconda Prompt* serta menggunakan *Text Editor Jupyter Notebook*, sedangkan penelitian yang dilakukan oleh Aprilianus Kristianus Lalo dkk menggunakan *RapidMiner Studio* Version 9.7 [2]. Pada tujuan penelitian terdapat perbedaan, jika pada Marini Utami Putri dkk serta penelitian yang dilakukan Aprilianus Kristianus Lalo dkk hanya terdapat tujuan untuk mengolah data transaksi penulis memiliki tujuan pada penelitian yang lain seperti membuat aplikasi prediksi penjualan *jeans* berbasis *web*.

2.2. Teori Pendukung

Teori pendukung merupakan istilah dalam penelitian yang dapat digunakan sebagai acuan untuk memahami penelitian yang dilakukan oleh penulis.

2.2.1. Penjualan

Menurut Mulyadi penjualan merupakan kegiatan yang dilakukan oleh perusahaan untuk menjual barang atau jasa kepada pelanggan dengan tujuan memperoleh pendapatan. Penjualan melibatkan interaksi antara penjual dan pembeli dalam mencapai transaksi yang saling menguntungkan [3]. Ketatnya persaingan pada sektor penjualan, membuat pelaku usaha harus pintar dalam menganalisis pasar. Selain itu, ketersediaan barang yang dijualpun menjadi salah satu faktor yang harus dianalisis supaya kebutuhan pelanggan dapat terpenuhi. Untuk mendukung hal itu, data penjualan sebelumnya dapat dimanfaatkan untuk menganalisis pasar dan kebutuhan pelanggan. [4]

2.2.2. Data Mining

Data mining adalah proses untuk mendapatkan informasi yang berguna dari basis data yang besar dan perlu diekstraksi agar menjadi informasi baru dan dapat membantu dalam pengambilan keputusan [5].

Data mining adalah proses menganalisa data dari yang berbeda dan menyimpulkannya menjadi informasi atau pengetahuan atau pola yang penting untuk meningkatkan keuntungan, memperkecil biaya pengeluaran, atau bahkan keduanya [6].

Menurut Hermawati, *Data mining* adalah proses yang memperkerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis [7].

Berdasarkan definisi-definisi di atas tentang, dapat disimpulkan bahwa *data mining* merupakan proses pencarian otomatis untuk menemukan pola atau model yang signifikan dari *database* yang besar.

2.2.2.1. Operasi Data Mining

Operasi *data mining* dapat dibagi menjadi beberapa jenis berdasarkan sifatnya:, yaitu :

1. Prediksi (*prediction driven*) operasi ini digunakan untuk menjawab pertanyaan "apa yang akan terjadi?" dan berfokus pada penciptaan model prediktif. Tujuannya adalah untuk membuat prediksi yang valid dan akurat terhadap fenomena atau peristiwa yang akan datang. Contoh penggunaan operasi prediksi meliputi validasi hipotesis, *querying*, dan pelaporan.
2. Penemuan (*discovery driven*) operasi ini bersifat transparan dan digunakan untuk menjawab pertanyaan "mengapa hal itu terjadi?" atau "apa yang membuatnya terjadi?". Fokusnya adalah pada pemahaman dan penemuan pola yang tersembunyi dalam data. Contoh penggunaan operasi penemuan meliputi analisis eksploratif, pemodelan prediktif, segmentasi *database*, analisis keterkaitan (*link analysis*), dan deteksi deviasi [7].

2.2.2.2. Teknik Data Mining

Beberapa teknik dan sifat *data mining* adalah sebagai berikut :

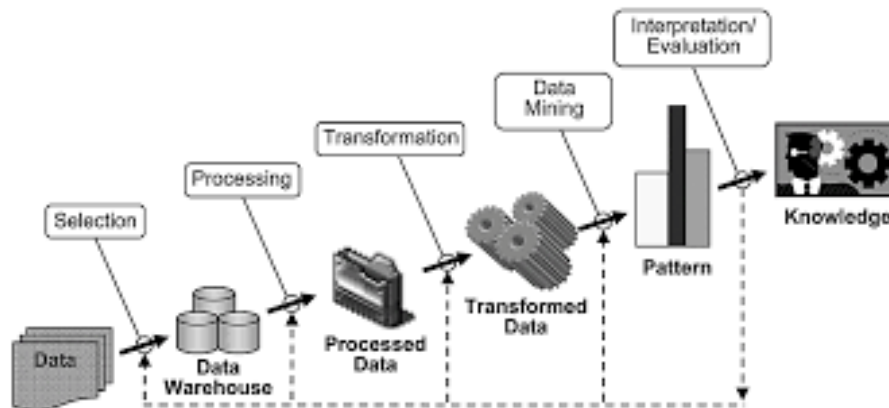
1. Klasterisasi adalah mempartisi *dataset* menjadi beberapa *sub-net* atau kelompok sedemikian rupa sehingga elemen-elemen dari suatu kelompok tertentu memiliki *set property* yang di *share* bersama, dengan tingkat

similaritas yang tinggi dalam suatu kelompok yang rendah. Disebut juga dengan “*unsupervised learning*”.

2. Regresi adalah memprediksi nilai dari suatu variabel kontinu yang diberikan berdasarkan nilai dari variabel yang lain, dengan mengasumsikan sebuah model ketergantungan linier atau non linier.
3. Klasifikasi adalah menentukan sebuah *record* data baru ke salah satu dari beberapa kategori (kelas) yang telah didefinisikan sebelumnya dan disebut juga dengan “*supervised learning*”.
4. Kaidah Asosiasi (*association rule*) adalah mendeteksi kumpulan atribut-atribut yang muncul bersamaan (*co-occur*) dalam frekuensi yang sering dan membentuk sejumlah kaidah dari kumpulan-kumpulan tersebut [7].

2.2.3. Knowledge Discovery in Database

Knowledge Discovery in Database (KDD) adalah proses menentukan informasi yang berguna serta pola-pola yang ada dalam data. Informasi ini terkandung dalam basis data yang berukuran besar yang sebelumnya tidak diketahui dan potensial bermanfaat. Data Mining merupakan salah satu langkah dari serangkaian proses *iterative* KDD [8]. Berikut tahapan proses KDD dapat dilihat pada gambar.



Gambar 2.1 Tahapan Dalam KDD
(Sumber : Algoritma Data Mining [8][8])

Menurut Amalia, KDD atau *Knowledge Discovery in Database* adalah sekumpulan proses yang digunakan untuk menemukan pengetahuan yang meliputi :

1. *Data Selection.*

Hanya data yang relevan yang akan dianalisis dan diambil dari *database*, sehingga tidak semua data yang ada dalam *database* akan digunakan. Sebagai contoh, dalam kasus analisis *market basket* yang mempelajari faktor kecenderungan pelanggan, tidak diperlukan pengambilan nama pelanggan, tetapi hanya perlu menggunakan ID pelanggan.

2. *Pre-processing.*

Pada tahap *pre-processing*, dilakukan penghapusan data yang tidak konsisten dan mengandung *noise*, penghapusan duplikasi data, perbaikan kesalahan data, serta penambahan data eksternal yang relevan.

3. *Transformation.*

Transformasi data adalah proses mengubah dan menggabungkan data ke dalam format yang ditentukan. Sebelum diterapkan dalam *data mining*, data perlu diubah menjadi format khusus. Sebagai contoh, metode standar seperti analisis asosiasi dan *clustering* hanya dapat menerima *input* data yang bersifat kategorikal. Oleh karena itu, jika data berupa angka numerik dengan sifat kontinu, perlu dibagi menjadi beberapa interval. Proses ini umumnya dikenal sebagai transformasi data.

4. Data Mining.

Proses penambangan data juga dikenal sebagai *data mining*. Ini merupakan tahap utama di mana metode digunakan untuk menemukan pengetahuan berharga yang tersembunyi dalam data.

5. *Interpretation/Evaluasi*.

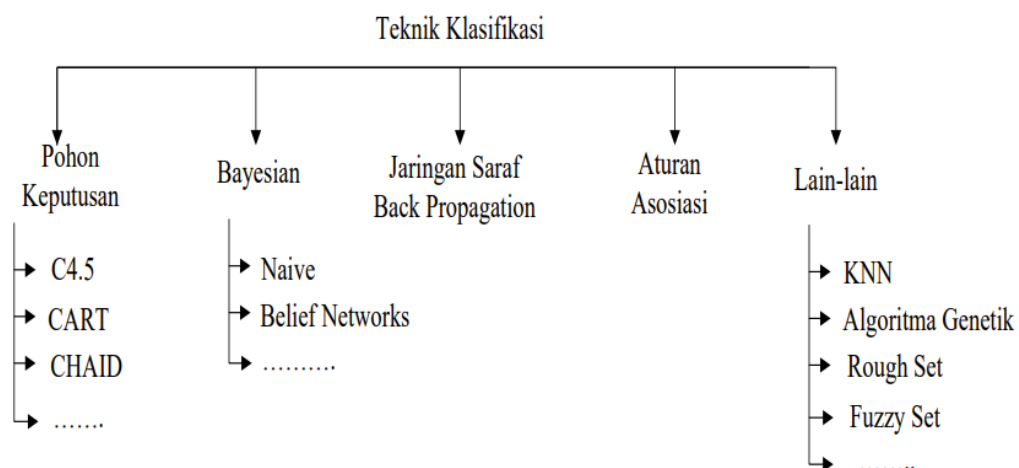
1. Evaluasi Pola (*Pattern Evaluation*) memiliki peran dalam mengidentifikasi pola-pola menarik dalam *knowledge base* yang telah ditemukan. Pada tahap ini, pola-pola khas dihasilkan dari model klasifikasi yang dievaluasi untuk menilai keberhasilan hipotesis yang ada. Jika hasilnya tidak sesuai dengan hipotesis, ada beberapa alternatif yang dapat diambil, seperti menggunakan hasil tersebut sebagai umpan balik untuk meningkatkan proses *data mining*, atau mencoba metode *data mining* lain yang lebih sesuai.
2. Presentasi Pengetahuan (*Knowledge Presentation*) melibatkan visualisasi dan penyajian informasi mengenai metode yang digunakan untuk mendapatkan pengetahuan atau informasi yang telah ditemukan

oleh pengguna. Tahap terakhir dalam proses *data mining* adalah merumuskan keputusan berdasarkan hasil analisis yang diperoleh. Proses ini melibatkan menerjemahkan pola-pola yang dihasilkan dari *data mining* dan menguji kecocokannya dengan fakta atau hipotesis sebelumnya. Pengetahuan yang diperoleh dari pola-pola tersebut kemudian dipresentasikan dalam bentuk visualisasi [9].

2.2.4. Klasifikasi

Menurut Hermawati, Klasifikasi merupakan proses pembelajaran suatu fungsi tujuan (target) f yang memetakan tiap himpunan atribut x ke satu dari label kelas yang didefinisikan sebelumnya. Fungsi target disebut juga model klasifikasi [7].

Terdapat beberapa teknik yang digunakan dalam klasifikasi, di antaranya sebagai berikut:



Gambar 2.2 Teknik Klasifikasi
(Sumber : Data Mining [7])

2.2.5. Prediksi

Prediksi numerik adalah langkah yang dilakukan dalam memprediksi nilai yang kontinu (atau berurutan) untuk nilai yang diberikan. Menurut Han (2007) sejauh ini pendekatan yang paling banyak digunakan untuk prediksi numerik (yang kemudian disebut dengan prediksi saja) adalah regresi, sebuah metodologi statistik yang dikembangkan oleh Sir Frances Galton, seorang ahli matematika yang juga merupakan sepupu dari Charles Darwin. [10].

2.2.5.1. Teknik Prediksi

Teknik prediksi dalam *data mining* mencakup penggunaan berbagai metode untuk mengembangkan model prediktif berdasarkan data historis. Beberapa teknik prediksi umum yang digunakan dalam *data mining* meliputi:

1. Regresi (*Regression*)

Teknik regresi digunakan untuk memprediksi nilai kontinu berdasarkan hubungan statistik antara variabel *input* dan *output*.

2. Pohon Keputusan (*Decision Trees*)

Teknik pohon keputusan digunakan untuk membangun model prediktif berupa struktur pohon yang memecah data menjadi kelompok-kelompok yang semakin spesifik berdasarkan aturan keputusan.

3. Jaringan Saraf Tiruan (*Artificial Neural Networks*)

Teknik jaringan saraf tiruan digunakan untuk membangun model prediktif yang terinspirasi dari sistem saraf manusia, dengan neuron-neuron yang saling terhubung dan dapat mempelajari pola-pola kompleks dalam data.

4. *K-Nearest Neighbors* (KNN)

Teknik KNN digunakan untuk memprediksi kelas atau nilai berdasarkan kumpulan data terdekat dalam ruang fitur.

5. *Naive Bayes*

Teknik *Naive Bayes* menggunakan teorema Bayes untuk memprediksi kelas atau nilai berdasarkan perhitungan probabilitas dari kombinasi fitur yang ada [7].

2.2.6. Pohon Keputusan

2.2.6.1. Pengertian Pohon Keputusan

Pohon (*tree*) adalah sebuah struktur data yang terdiri dari simpul (*node*) dan rusuk (*edge*). Simpul pada sebuah pohon dibedakan menjadi tiga, yaitu simpul akar (*root/node*), simpul percabangan/internal (*branch/internal node*) dan simpul daun (*leaf node*) [11].

Pohon keputusan merupakan representasi sederhana dari teknik klasifikasi untuk sejumlah kelas berhinga, dimana simpul *internal* maupun simpul akar ditandai dengan nama atribut, rusuk-rusuknya diberi label nilai atribut yang mungkin dan simpul daun ditandai dengan kelas-kelas yang berbeda [11].

2.2.6.2. Model Pohon Keputusan

Model pohon keputusan merupakan metode prediksi yang menggunakan struktur pohon atau hierarki. *Decision tree* adalah salah satu metode klasifikasi yang paling populer yang digunakan dalam *data mining*. Beberapa model pohon keputusan yang sering digunakan dalam *data mining* adalah sebagai berikut :

1. Model CART (*Classification and Regression Trees*)

Model pohon keputusan CART dikembangkan oleh Breiman dkk. Model ini dapat digunakan untuk klasifikasi (*classification trees*) dan regresi (*regression trees*) tergantung pada tipe variabel target yang diprediksi. Model CART bekerja dengan membagi *dataset* berdasarkan nilai-nilai atribut yang memiliki hubungan dengan variabel target, sehingga membentuk struktur pohon yang dapat digunakan untuk prediksi. *Internal Node*, merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.

2. Model C4.5

Model pohon keputusan C4.5 dikembangkan oleh Quinlan sebagai pengembangan dari model pohon keputusan sebelumnya, ID3 Model C4.5 memiliki kemampuan untuk mengatasi variabel numerik dan hilangnya data (*missing values*). Model ini menggunakan ukuran keuntungan informasi (*information gain*) untuk memilih atribut terbaik dalam membagi *dataset* dan membangun pohon keputusan.

3. Model ID3

Model pohon keputusan ID3 dikembangkan oleh Quinlan. Model ini menggunakan ukuran keuntungan informasi (*information gain*) untuk memilih atribut terbaik dalam membagi *dataset*. ID3 hanya cocok untuk atribut dengan tipe data kategori diskrit dan tidak dapat mengatasi variabel numerik atau *missing values* [7].

2.2.7. Model C4.5

2.2.7.1. Pengenalan Metode C4.5

Metode C4.5 yaitu sebuah metode yang digunakan untuk membentuk *decision tree* (pohon keputusan). Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode C4.5 mengkonstruksi pohon keputusan dari data pelatihan, yang berupa kasus-kasus atau *record-record* (*tupel*) dalam basis data. Setiap kasus berisikan nilai dari atribut-atribut untuk sebuah kelas. Setiap atribut dapat berisi data diskrit atau kontinu (numerik). Metode C4.5 juga menangani kasus yang tidak memiliki nilai untuk sebuah atau lebih atribut. Akan tetapi, atribut kelas hanya bertipe diskrit dan tidak boleh kosong [12].

Metode C4.5 merupakan pengembangan dari metode ID3 yang telah diperbaiki (Santosa, ID3). Metode C4.5 melakukan perbaikan dengan memperhatikan *missing value*, data kontinu, *pruning*, dan penggunaan aturan. Tahapan-tahapan yang diikuti oleh metode C4.5 dalam membentuk pohon keputusan dapat dijelaskan sebagai berikut :

1. Tahap awal pembentukan pohon dimulai dengan menciptakan sebuah *node* yang mewakili sampel pelatihan.
2. Apabila semua sampel pada *node* tersebut memiliki kelas yang identik, maka *node* tersebut akan diubah menjadi *leaf node* dengan label kelas yang sesuai.
3. Apabila sampel-sampel pada *node* tersebut memiliki kelas yang berbeda, metode akan mencari atribut yang memiliki *gain ratio* tertinggi dari atribut yang tersedia. Hal ini bertujuan untuk memilih atribut yang paling

berpengaruh dalam memisahkan sampel pada *training sample*. Jika atribut tersebut memiliki nilai yang kontinu, maka perlu dilakukan proses diskritisasi terlebih dahulu.

4. Cabang-cabang pada setiap *node* akan dibentuk berdasarkan nilai-nilai atribut pengujian.
5. Proses pembentukan pohon keputusan akan dilakukan secara rekursif untuk setiap sampel pada setiap cabang.
6. Proses rekursif akan berhenti jika salah satu dari kondisi berikut terpenuhi:
 1. Semua sampel pada *node* berasal dari kelas yang sama.
 2. Tidak ada atribut lain yang dapat digunakan untuk mempartisi sampel lebih lanjut.
 3. Tidak ada sampel yang memenuhi kondisi *test-attribute* = nilai tertentu.

Dalam kasus ini, *node* akan diubah menjadi *leaf node* dan diberi label kelas yang paling umum (*majority voting*).

Metode untuk membangun pohon keputusan dapat ditemukan pada gambar yang ada pada penjelasan berikut :

Algorithm : Generete_decision_tree.

Narrative : Generate a decision tree from the given training data.

Input : The training samples, samples, represented by discrete valued attribute, the set of candidate attributes, attribute-list.

Output : A dicsion tree.

Method

- 1) Craete a node N;
- 2) If samples are all of the same class, C then
- 3) Return N as a leaf node labeled with class C
- 4) If attribute-list is empty then

- 5) Return N as leaf node labeled with the most common class in samples; // majority voting
- 6) Select test-attribute, the attribute among attribute-list with the highest gain ratio;
- 7) Label node N with last attribute;
- 8) For each known value a, of test-attribute
- 9) Grow a branch from node N for the condition test-attribute = α_i
- 10) Let be the set of samples in samples for which test-attribute = // a partition
- 11) If is empty then
- 12) attach a leaf labeled with the most common class in sample;
- 13) Else attach the node returned by Generate_decision_tree (S_i , attribute-list-list-attribute);

Pada tahap pembelajaran metode C4.5 memiliki 2 prinsip kerja

yaitu:

1. Proses konstruksi pohon keputusan. Tujuan dari metode penginduksi pohon keputusan adalah membangun struktur pohon yang dapat digunakan untuk memprediksi kelas dari sebuah kasus baru yang belum diketahui kelasnya. Metode C4.5 menggunakan metode *divide and conquer* dalam pembangunan pohon keputusan. Pada awalnya, hanya dibuat satu *node* akar dengan menerapkan metode *divide and conquer*. Metode ini memilih pemisahan kasus yang paling optimal dengan menghitung dan membandingkan *gain ratio*. Setelah itu, proses *divide and conquer* akan digunakan kembali pada setiap *node* yang terbentuk di level berikutnya hingga daun-daun terbentuk.
2. Proses pembuatan aturan (*rule set*). Aturan-aturan yang terbentuk dari pohon keputusan akan membentuk kondisi dalam bentuk *if-then*. Aturan-

aturan ini diperoleh dengan menelusuri pohon keputusan dari akar hingga daun. Setiap *node* dan kondisi percabangan akan membentuk suatu *if*, sementara nilai-nilai yang terdapat pada daun akan membentuk suatu hasil atau *then*.

Untuk memudahkan penjelasan mengenai metode C4.5 berikut ini disertakan contoh kasus yang dituangkan dalam tabel berikut:

Tabel 2.1 Contoh Tahapan Pohon Keputusan 1

(Sumber : Algoritma Data Mining [8][8])

<i>No</i>	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
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

Pada Tabel 2.1, akan dilakukan pembangunan pohon keputusan untuk menentukan apakah akan bermain tenis atau tidak berdasarkan kondisi cuaca (*outlook*), temperatur, kelembaban (*humidity*), dan keadaan angin (*windy*). Secara umum, metode C4.5 yang digunakan untuk membangun pohon keputusan dapat dijelaskan sebagai berikut:

1. Memilih atribut sebagai akar pohon.

2. Membuat cabang untuk setiap nilai yang mungkin pada atribut tersebut.
3. Membagi kasus ke dalam cabang berdasarkan nilai atribut..
4. Mengulangi proses untuk setiap cabang hingga semua kasus pada cabang memiliki kelas yang sama.

Dalam pemilihan atribut sebagai akar, digunakan kriteria *gain ratio* tertinggi dari atribut yang tersedia.

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

Dengan S : Himpunan kasus,

A : Atribut, n : Jumlah partisi atribut A,

|S_i| : Jumlah kasus pada partisi ke-i,

|S| : Jumlah kasus dalam S. Sedangkan perhitungan nilai *entropy* dapat dilihat pada rumus kedua berikut:

$$Entropy(S) = \sum_{i=1}^n - p_i \times \log_2 p_i \quad \dots\dots\dots \text{persamaan (2)}$$

Dengan p_i : Proporsi dari S_i terhadap S.

Berikut ini adalah penjelasan rinci mengenai setiap langkah dalam pembentukan pohon keputusan menggunakan metode C4.5 untuk memecahkan permasalahan.

Menghitung jumlah kasus, jumlah kasus dengan keputusan "Yes", jumlah kasus dengan keputusan "No", dan menghitung Entropi dari seluruh kasus dan kasus yang dibagi berdasarkan atribut *Outlook*, *Temperature*, *Humidity*, dan *Windy*. Selanjutnya, dilakukan perhitungan Gain untuk setiap atribut. Hasil perhitungan ini ditampilkan dalam tabel berikut:

Tabel 2.2 Contoh Tahapan Pohon Keputusan 2

(Sumber : Algoritma Data Mining [8])

		S	No(S ₁)	Yes(S ₂)	Entropy	Gain
Total		14	4	10	0,863120569	
Outlook						0,258521037
	Cloudy	4	0	4	0	
	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						
	False	8	2	6	0,811278124	
	True	6	4	2	0,918295834	

Baris Total kolom Entropy diperoleh sebagai berikut :

$$Entropy (Total) = \left(-\frac{4}{14} \times \log_2 \left(\frac{4}{14}\right)\right) + \left(-\frac{10}{14} \times \log_2 \left(\frac{10}{14}\right)\right) = 0,863120569$$

Sedangkan nilai Gain pada baris Outlook diperoleh sebagai berikut

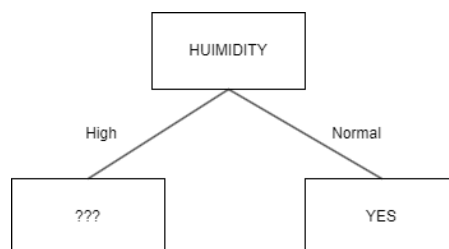
:

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

Sehingga didapat Gain (Total, Outlook) = 0,258521037

Berdasarkan hasil yang diberikan di atas, dapat disimpulkan bahwa atribut dengan Gain tertinggi adalah *Humidity*, dengan nilai 0,37. Oleh karena itu, *Humidity* akan menjadi *node* akar dalam pembentukan pohon keputusan. Terdapat 2 nilai atribut untuk *Humidity*, yaitu *High* dan *Normal*. Dari kedua nilai atribut tersebut, nilai

atribut Normal telah mengklasifikasikan kasus menjadi "Yes" sebanyak 1 kali, sehingga tidak perlu dilakukan perhitungan lebih lanjut. Namun, untuk nilai atribut *High*, masih perlu dilakukan perhitungan lanjutan. Berdasarkan informasi ini, dapat digambarkan pohon keputusan sementara dalam bentuk gambar.



Gambar 2.3 Contoh Pohon Keputusan 1
(Sumber : Algoritma Data Mining [8][8])

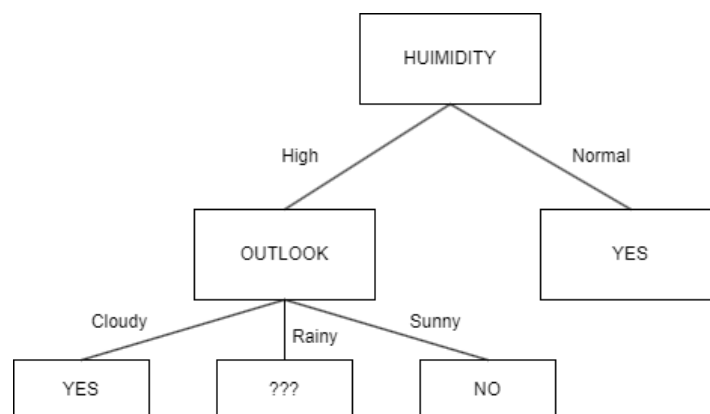
Proses perhitungan meliputi penghitungan jumlah kasus, jumlah kasus untuk keputusan "Yes", jumlah kasus untuk keputusan "No", dan perhitungan *Entropy* dari seluruh kasus serta kasus yang dibagi berdasarkan atribut Outlook, *Temperature*, dan Windy yang dapat menjadi *node* akar dari nilai atribut *High*. Setelah itu, dilakukan perhitungan *Gain* untuk setiap atribut. Hasil perhitungan ini disajikan dalam tabel berikut.

Tabel 2.3 Contoh Tahapan Pohon Keputusan 3
(Sumber : Algoritma Data Mining [8])

		S	No(S ₁)	Yes(S ₂)	Entropy	Gain
Humidity =		7	4	3	0,985228136	
High						
Outlook						0,69951385
	Cloudy	2	0	2	0	
	Rainy	2	1	1	1	
	Sunny	3	3	0	0	

Temperature						0.020244207
	Cool	0	0	0	0	
	Hot	3	2	1	0,918295834	
	Mild	4	2	2	1	
Windy						0.020244207
	False	4	2	2	1	
	True	3	2	1	0,918295834	

Berdasarkan hasil perhitungan, ditemukan bahwa atribut dengan *Gain* tertinggi adalah *Outlook* dengan nilai 0,67. Oleh karena itu, *Outlook* akan menjadi cabang *node* dari nilai atribut *High*. Terdapat tiga nilai atribut dari *Outlook*, yaitu *Cloudy*, *Rainy*, dan *Sunny*. Dari ketiga nilai atribut tersebut, nilai atribut *Cloudy* sudah dapat mengklasifikasikan kasus menjadi "Yes" dan nilai atribut *Sunny* sudah mengklasifikasikan kasus menjadi "No", sehingga tidak diperlukan perhitungan lebih lanjut. Namun, untuk nilai atribut *Rainy*, perhitungan masih diperlukan. Pohon keputusan yang terbentuk hingga tahap ini dapat dilihat pada gambar berikut.



**Gambar 2.4 Contoh Pohon Keputusan 2
(Sumber : Algoritma Data Mining [8])**

Melakukan perhitungan jumlah kasus, jumlah kasus untuk keputusan "Yes", jumlah kasus untuk keputusan "No", dan *Entropi* dari seluruh kasus serta kasus yang dibagi

berdasarkan atribut *Temperature* dan *Windy*, yang dapat menjadi cabang *node* dari nilai atribut *Rainy*. Kemudian, dilakukan perhitungan *Gain* untuk setiap atribut. Hasil perhitungan tersebut ditampilkan dalam tabel berikut.

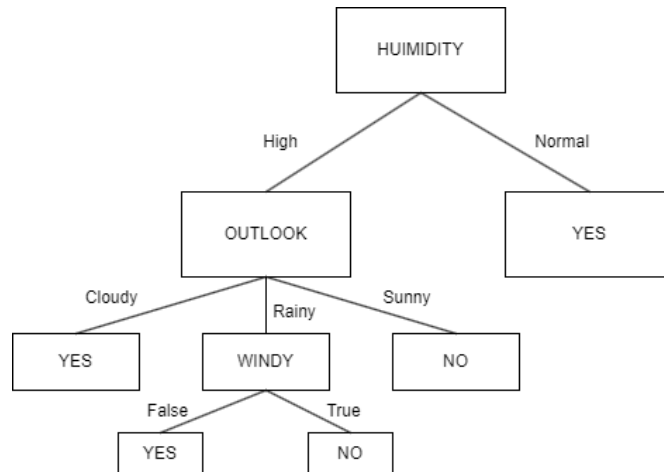
Tabel 2.4 Contoh Tahapan Pohon Keputusan 4

(Sumber : Metode Data Mining [8])

		S	No(S ₁)	Yes(S ₂)	Entropy	Gain
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	

Berdasarkan hasil tersebut, dapat disimpulkan bahwa atribut dengan *Gain* tertinggi adalah *Windy*, dengan nilai *Gain* sebesar 1. Oleh karena itu, *Windy* dapat menjadi cabang *node* dari nilai atribut *Rainy*. Terdapat 2 nilai atribut dari *Windy*, yaitu *False* dan *True*. Dari kedua nilai atribut tersebut, nilai atribut *False* sudah mengklasifikasikan kasus menjadi satu kategori dengan keputusan *Yes*, sedangkan nilai atribut *True* sudah mengklasifikasikan kasus menjadi satu kategori dengan keputusan *No*. Oleh karena itu, tidak perlu dilakukan perhitungan lebih lanjut untuk

nilai atribut ini. Pohon keputusan yang terbentuk sampai tahap ini dapat dilihat pada gambar berikut [13].



**Gambar 2.5 Contoh Pohon Keputusan 3
(Sumber : Algoritma Data Mining [8])**

2.3. Piranti Pendukung

Piranti pendukung merupakan alat yang digunakan untuk membantu penulis dalam menyelesaikan penelitian.

2.3.1. Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* bisa dibilang bahasa pemrograman dengan tujuan umum yang dikembangkan secara khusus untuk membuat *source code* mudah dibaca. *Python* juga memiliki *library* yang lengkap sehingga memungkinkan *programmer*

untuk membuat aplikasi yang mutakhir dengan menggunakan *source code* yang tampak sederhana [14].



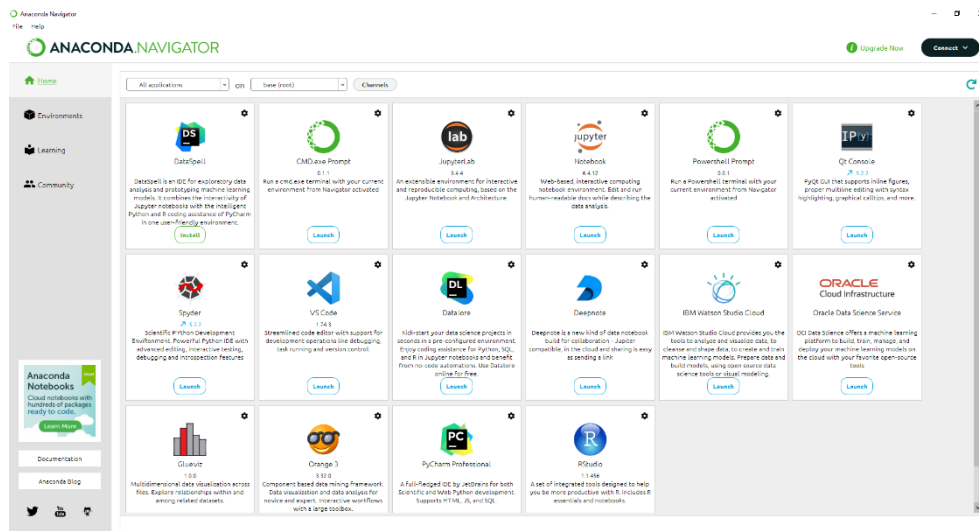
Gambar 2.6 Logo Python
(Sumber : python.org [15])

2.3.2. Anaconda Navigator

Anaconda Navigator adalah aplikasi desktop yang disediakan oleh *Anaconda Distribution*, yang merupakan platform distribusi *Python* dan lingkungan pemrograman yang populer di kalangan ilmu data dan pemrosesan data. *Anaconda Navigator* menyediakan antarmuka grafis (GUI) yang memudahkan pengguna dalam mengelola dan menggunakan berbagai paket dan lingkungan *Python* yang terpasang dalam *Anaconda*. Dalam *Anaconda Navigator* terdapat beberapa aplikasi salah satunya adalah *Jupyter* [16].



**Gambar 2.7 Logo Anaconda
(Sumber : anaconda.com [17])**



Gambar 2.8 Tampilan Anaconda

2.3.2.1. Jupyter

Jupyter adalah lingkungan komputasi interaktif yang populer di kalangan ilmu data dan pemrosesan data. *Jupyter* memungkinkan pengguna untuk membuat dan berbagi dokumen yang menggabungkan kode, teks naratif, visualisasi, dan elemen-elemen interaktif lainnya. Dokumen-dokumen ini disebut sebagai "*notebook*" dan dapat dijalankan di dalam *Jupyter Notebook* atau *JupyterLab*, yang merupakan antarmuka pengguna untuk lingkungan *Jupyter* [18].



**Gambar 2. 9 Logo Jupyter Notebook
(Sumber : jupyter.org [19])**

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, f1_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
from sklearn import tree
import pickle

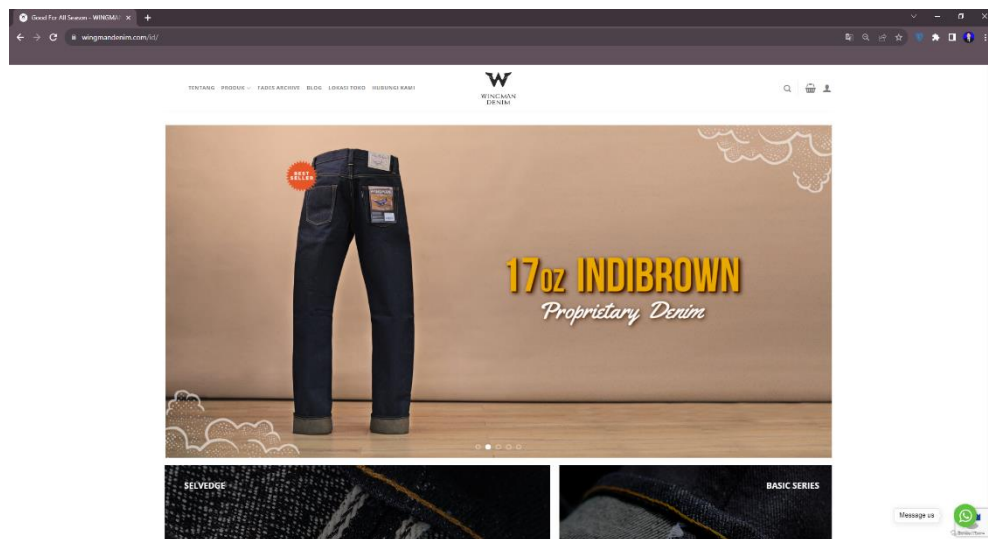
def read_file(dataset):
    df = pd.read_excel (dataset)
    print(df.shape)
    df['STATUS'] = df['STATUS'].apply(lambda x: x.strip().lower())
    return df
```

Gambar 2.10 Contoh Coding Jupyter Notebook

2.3.3. Website

Menurut Rangga Sidik dan Deni Hamdani, *website* adalah sejumlah halaman *web* yang memiliki topik saling terkait, terkadang disertai pula dengan

berkas - berkas gambar, video, atau jenis-jenis berkas lainnya. Sebuah *website* biasanya ditempatkan setidaknya pada sebuah *server web* yang dapat diakses melalui jaringan seperti internet, ataupun jaringan wilayah lokal (LAN) melalui alamat internet yang dikenali sebagai *URL*. Gabungan atas semua situs yang dapat diakses publik di internet disebut pula sebagai *World Wide Web (WWW)* [20].



Gambar 2.11 Contoh Website
(Sumber : wingmandenim.com [21])

2.3.4.1. HTML (*HyperText Markup Language*)

Menurut Winarno dan Utomo yang disitasi oleh Agus Prayitno dan Yulia Safitri “HTML singkatan dari *Hypertext Markup Language* dan berguna untuk menampilkan halaman *web*”. Kode HTML ini digunakan sebagai bahan untuk melakukan *rendering* halaman *web*. Karena berbasis teks murni (*plain text*) sehingga ukurannya kecil dan tidak memboroskan *bandwidth* apabila ditransfer melalui jaringan internet. Dalam pemrograman HTML dikenal adanya istilah *Tag*. *Tag* adalah sintak dari HTML yang ditulis diantara dua tanda lebih kecil dan lebih besar “<>” [22].



Gambar 2.12 Logo HTML
(Sumber : w3schools.com [23])

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Title goes here</title>
6   </head>
7   <body>
8
9   </body>
10 </html>
```

Gambar 2.13 Contoh Coding HTML

2.3.4.2. CSS (*Cascading Style Sheet*)

Menurut Adhi Prasetyo yang disitasi oleh Raden Shafira Annisa Ridmadhani, dkk menjelaskan bahwa CSS (*Cascading Style Sheet*) adalah salah satu bahasa desain *web* yang digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa *markup*, seperti HTML. CSS berfungsi untuk memisahkan konten utama dengan tampilan visualnya. CSS menciptakan fleksibilitas dalam mengontrol spesifikasi tampilan suatu halaman *web*. CSS dibuat dan dikembangkan oleh W3C (*World Wide Web Consortium*) pada tahun 1996 [24]. Pada saat ini

perkembangan dari CSS itu sudah mencapai CSS3, dimana CSS1 dibuat pada tahun 1996, sesuai dengan perilisan pertama CSS. Lalu selanjutnya terdapat CSS2 dirilis pada tahun 1998 dan CSS3 dirilis pada tahun 1999.



Gambar 2.14 Logo CSS
(Sumber : hostinger.co.id [25])

```

34  /* A reference to a type */
35  span.ts span.type-ref {
36      color: ■rgb(175, 0, 219) !important;
37  }
38
39  /* Signature details */
40  div.signature > table {
41      border-collapse: collapse;
42      border: thin ■darkgray solid;
43      width: 60%;
44  }

```

Gambar 2.15 Contoh Coding CSS

2.3.4.3. PHP (*Hypertext Preprocessor*)

Menurut Alexander F. K. Sibero yang disitasi oleh Harri Hidayat, dkk menjelaskan bahwa PHP (*Hypertext Preprocessor*) adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh

prosesnya dijalankan pada *server* tidak dijalankan pada *client*. PHP merupakan suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode fungsi PHP dengan kebutuhannya [26].



Gambar 2.16 Logo PHP
(Sumber : php.net [27])

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Belajar PHP</title>
5 </head>
6 <body>
7   <?php
8     echo "Nama saya Mohammad Nur Fawaiq";
9   ?>
10 </body>
11 </html>
```

Gambar 2.17 Contoh Coding PHP