Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TELAAH LITERATUR

Tinjauan pustaka berisikan teori–teori yang digunakan untuk memperjelas penelitian ini. Teori-teori yang dimaksud adalah teori *data mining*. Teori-teori yang dibahas pada skripsi ini berasal dari buku-buku dan makalah-makalah yang relevan.

Menurut Tan (2006), *data mining* merupakan pengekstrakan informasi baru yang diambil dari bongkahan data besar yang membantu dalam pengambilan keputusan. Pola yang disajikan haruslah mudah dipahami, berlaku untuk data yang akan diprediksi dengan derajat kepastian tertentu, berguna, dan baru. *Data mining* adalah bagian dari *Knowledge Discovery in Database. Knowledge Discovery in Database* adalah sebuah proses otomatis atas pencarian data di dalam sebuah memori yang amat besar dari data untuk mengetahui pola dengan menggunakan alat seperti klasifikasi, hubungan (*association*) atau pengelompokan (*clustering*).

Secara umum, fungsi proses data *mining* dapat diklasifikasikan dalam dua kategori, yaitu deskriptif dan prediktif. Fungsi prediktif menyediakan aturan-aturan global yang dapat diaplikasikan terhadap basis data. Aplikasi yang dimaksudkan menggunakan beberapa variabel untuk memprediksi sesuatu atau suatu nilai yang akan datang. Fungsi–fungsi prediktif adalah klasifikasi, *decision tree*, analisis *time series*, regresi, prediksi, jaringan syaraf tiruan. Sedangkan fungsi deskriptif bertujuan untuk menyediakan deskripsi dari data sumber yang

tersedia, untuk mendapatkan pola penafsiran (*human interpretable patterns*) untuk menjelaskan data. Fungsi–fungsi deskriptif adalah *clustering*, *summarization*, aturan asosiasi, *sequence discovery* (Firdi, 2009).

Knowledge Discovery in Database (KDD) adalah proses menentukan informasi yang berguna serta pola–pola yang ada dalam data (Goharian & Grossman, 2003). Informasi ini terkandung dalam basis data yang berukuran besar yang sebelumnya tidak diketahui dan bermanfaat (Han & Kamber, 2006). *Data mining* merupakan salah satu langkah dari serangkaian proses *iterative* KDD.

Tahapan proses KDD menurut Han & Kamber (2006) adalah:

1. *Data Cleaning*: Pembersihan data dilakukan untuk menghilangkan data yang tidak konsisten dan mengandung *noise.*
2. *Data Integration*: Proses integrasi data dilakukan untuk menggabungkan data dari berbagai sumber menjadi bentuk sebuah penyimpanan data yang saling berhubungan, seperti dalam *data warehousing*.
3. *Data Selection*: Proses seleksi data mengambil data yang relevan digunakan untuk proses analisis.
4. *Data Transformation*: Proses ini mentransformasikan atau menggabungkan data ke dalam bentuk yang tepat untuk dilakukan proses *mine* dengan cara melakukan peringkasan atau operasi agregasi. Dalam beberapa kasus, proses transformasi dilakukan sebelum proses seleksi, misalnya dalam kasus *data warehouse*.
5. *Data Mining*: *Data mining* merupakan proses yang penting, dimana metode-metode cerdas diaplikasikan untuk mengekstrak pola-pola dalam data.

6. *Pattern Evaluation*: Evaluasi pola diperlukan untuk mengidentifikasi beberapa pola yang menarik dalam merepresentasikan pengetahuan.

7. *Knowledge Presentation*: Penggunaan visualisasi dan teknik representasi untuk menunjukkan pengetahuan hasil penggalian dari tumpukan data kepada pengguna.

Penjelasan lebih rinci mengenai langkah-langkah *Knowledge Discovery in Database* (KDD) akan diuraikan pada masing-masing subbab berikut ini.

2.1 Data Cleaning

Data Cleaning adalah proses yang digunakan untuk membuang data yang tidak konsisten dan bersifat *noise* dari data yang terdapat di berbagai basis data yang mungkin berbeda format maupun *platform*, yang kemudian diintegrasikan ke dalam suatu *database data warehourse* (Noer, 2013). Pada tahap *cleaning*, hal yang harus diperhatikan adalah menganalisis adanya data *outlier*. Data *outlier* adalah data yang menyimpang terlalu jauh dari data lainnya dalam suatu rangkaian data. Adanya data *outlier* ini akan membuat analisis terhadap serangkaian data menjadi bias, atau tidak mencerminkan fenomena yang sebenarnya (Konsultan statistic, 2010).

Ada empat penyebab timbulnya data *outlier*, yaitu kesalahan dalam memasukkan data, kegagalan dalam spesifikasi *missing value* ke dalam progam komputer, *outlier* bukan merupakan anggota populasi yang diambil sebagai sampel, dan *outliers* berasal dari populasi yang berasal dari sampel tetapi distribusi dari variabel dalam populasi tersebut memiliki nilai ekstrim dan tidak terdistribusi secara normal (Han, 2006). Adapun *outliers* dapat dianalisis dengan

beberapa pendekatan, yaitu pendekatan grafis, *model based*, *distance based*, *devitation based,* dan lain-lain.

Menurut Imelda (2012), Pendekatan grafis dilakukan dengan menggunakan *blox pot* (1D), *scatter plot* (2D), dan *spin plot* (3D) pada grafik. Pendekatan grafis tidak terlalu bagus dalam menentukan data *outlier*, dikarenakan pendekatan grafis bersifat sangat subjektif dalam penentuan *outlier* dan memerlukan waktu yang sangat banyak untuk pendekatan ini.

Salah satu pendekatan *based* yang dibahas adalah pendekatan *statistic*. Dalam pendekatan *statistic* ada beberapa fungsi distribusi yang bisa digunakan, misalnya distribusi normal, distribusi *poisson*, distribusi *gamma* dan sebagainya. Pendekatan statistik bergantung pada distribusi data, parameter distribusi (*mean*, *median*, *variance*), jumlah *outlier* yang dapat diterima. Kelebihan pendekatan statistik adalah jika pengetahuan data cukup (jenis distribusi data dan jenis uji yang diperlukan), maka pendekatan statistik akan sangat efektif. Kekurangan dari pendekatan statistik sendiri adalah sulit untuk menemukan fungsi distribusi dan jenis uji yang tepat untuk data berdimensi tinggi.

Distance based memiliki banyak teknik yang dikelompokkan ke dalam teknik pohon keputusan, *Bayesian* (*Naïve Bayesian*, *Bayesian belief network*), jaringan saraf tiruan (*backpropagation*), teknik yang berbasis konsep dari penambangan aturan–aturan asosiasi, dan teknik lain (*k-Nearest Neighboor*, algoritma genetik, teknik dengan pendekatan himpunan *rough* dan *fuzzy*). Setiap teknik memiliki kelebihan dan kekurangan tersendiri. Teknik–teknik tertentu

akan optimal dengan data yang memiliki kriteria yang sesuai, sehingga data juga mempengaruhi kinerja dari teknik yang ada.

Devitation based mengidentifikasi *outliers* dengan menentukan karakteristik utama dari objek-objek dalam sebuah grup. Objek yang memiliki "deviasi" dari deskripsi ini akan dianggap sebagai *outlier*. Dalam *deviation based* terdapat beberapa teknik seperti teknik *sequential exception,* yaitu mensimulasikan cara manusia membedakan objek yang "berbeda" dari sederetan objek yang "normal" dan teknik OLAP data *cube,* yang menggunakan data *cubes* untuk mengidentifikasi daerah–daerah *anomaly* pada data *multidimensional* yang besar.

2.2 Data Integration

Data integration adalah penggabungan beberapa sumber data ke dalam data *warehouse*. Sumber data ini adalah *database*. *Data warehouse* adalah pusat repositori informasi yang mampu memberikan *database* berorientasi subjek untuk informasi yang bersifat historis yang mendukung DSS (*Decision Support System*). Pada tahap ini, redudansi sangatlah penting, untuk menghilangkan atribut–atribut yang tidak konsisten yang dapat berpengaruh pada data. Redudansi tidak dapat dihilangkan sama sekali karena berguna untuk integritas referensial yang menghubungkan satu *field* pada suatu tabel dengan *field* lain pada tabel yang berbeda (global).

Sebagian redudansi data dapat dideteksi dengan analisis korelasi. Menurut Riskawati (2013), kolerasi merupakan teknik analisis yang termasuk dalam salah satu teknik pengukuran asosiasi atau hubungan (*measures of association*).

Pengukuran asosiasi merupakan istilah umum yang mengacu pada sekelompok teknik dalam *statistic bivariate* yang digunakan untuk mengukur kekuatan hubungan antara dua variabel. Pengukuran asosiasi menggunakan nilai numerik untuk mengetahui tingkatan asosiasi atau kekuatan hubungan antara variabel. Dua variabel dikatakan berasosiasi jika perilaku variabel yang satu mempengaruhi variabel yang lain. Jika tidak terjadi pengaruh, maka kedua variabel tersebut disebut independen. Korelasi bermanfaat untuk mengukur kekuatan hubungan antara dua variabel (kadang lebih dari dua variabel) dengan skala-skala tertentu.

2.3 Data Selection

Data selection adalah pemilihan data yang ada dalam *database data warehouse*, kemudian direduksi untuk mendapatkan hasil yang akurat. Tujuannya adalah untuk penentuan jenis data, sumber dan instrumen yang sesuai. Ada tiga metode seleksi pada *data mining* menurut Nurhada (2010), yaitu *sampling, denoising, feature*. *Sampling* adalah seleksi subset representatif dari populasi data yang besar, *denoising* adalah proses menghilangkan *noise* dari data yang akan ditransformasikan, sedangkan *feature extraction* adalah proses membuka spesifikasi data yang signifikan dalam konteks tertentu.

2.4 Data Transformation

Data transformation adalah proses dimana data ditransformasikan menjadi bentuk yang sesuai untuk diproses dalam *data mining*. Menurut Nurhada (2010) ada tiga metode transformasi pada *data mining*, yaitu *centering*, *normalization*, dan *scaling*. *Centering* adalah mengurangi setiap data dengan rata-rata dari setiap

atribut yang ada, sedangkan *normalization* untuk membagi setiap data dengan standar deviasi dari atribut bersangkutan, dan *scaling* adalah mengubah data sehingga berada dalam skala tertentu.

2.5 Data Mining

Data mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Pramudiono, 2006). Beberapa teknik dalam *data mining* adalah *clustering*, klasifikasi, *association*, dan lain-lain.

Menurut Kusnawi (2007), klasifikasi adalah suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi. Teknik ini menggunakan *supervised induction*, yang memanfaatkan kumpulan pengujian dari *record* yang terklasifikasi untuk menentukan kelas-kelas tambahan. Salah satu contoh yang mudah dan populer adalah dengan *decision tree*, yaitu salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi. *Decision tree* adalah struktur *flowchart* yang menyerupai *tree* (pohon), dimana setiap simpul internal menandakan suatu tes pada atribut, setiap cabang merepresentasikan hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas. Alur pada *decision tree* ditelusuri dari simpul akar ke simpul daun yang memegang prediksi.

Association digunakan untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana *link* asosiasi muncul pada setiap kejadian. Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, *support* yaitu presentasi kombinasi atribut tersebut dalam basis data dan *confidence* yaitu kuatnya hubungan antar atribut dalam aturan asosiatif (Kusnawi, 2007).

Clustering digunakan untuk menganalisis pengelompokkan berbeda terhadap data. Mirip dengan klasifikasi, namun pengelompokkan belum didefinisikan, sebelum dijalankannya *tool data mining*. Biasanya menggunakan metode *neural network* atau statistik. *Clustering* membagi *item* menjadi kelompok-kelompok berdasarkan yang ditemukan *tool data mining*. Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar *cluster*. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruang *multidimensi* (Kusnawi, 2007).

2.6 Pattern Evaluation

Pattern evaluation adalah proses untuk menguji kebenaran dari pola data yang mewakili *knowledge* yang ada di dalam data itu sendiri. Menurut Nuqson (2010) dalam tahap ini hasil dari teknik *data mining* berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Dengan begitu pengembangan teknik untuk menilai ketertarikan pola sangat diperlukan, agar mendapatkan pola terbaik untuk hasil dari penelitian.

2.7 Knowledge Presentation

Menurut Kusnawi (2007), *knowledge presentation* adalah teknik *representative* dan visualisasi data yang digunakan untuk mempresentasikan pengetahuan yang didapat kepada *user*. Pada *knowledge presentation*, konsep dasar manajemen pengetahuan diterapkan.

Merepresentasikan suatu pengetahuan seharusnya dilakukan dengan cara menggali data-data dan informasi secara mendalam. Setelah menggali data adalah membagikan pengetahuan tersebut ke orang lain dan orang lain akan memberikan respon berupa pertanyaan kritis untuk menilai pemahaman mengenai pengetahuan tersebut.

Hasil dari *knowledge presentation* harus disajikan dalam bahasa yang mudah dipahami, menggunakan *representative visual*, atau bentuk lain yang mudah dimengerti pengguna. Tujuannya agar orang dapat memahami pengetahuan yang dibagikan, pengetahuan yang diberikan dapat melalui seminar, diskusi, pelatihan, dan lain-lain.

2.8 Algoritma C4.5

Menurut Kusnawi (2007), algoritma C4.5 merupakan algoritma yang digunakan untuk membentuk pohon keputusan. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami.

Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah

variabel target. 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.

Algoritma C4.5 adalah pengembangan dari algoritma ID3. Algoritma C4.5 sendiri sebenarnya sudah ada pengembangannya yaitu C5.0, akan tetapi algoritma C5.0 masih bersifat komersil.

Untuk memudahkan penjelasan mengenai algoritma C4.5, berikut ini adalah contoh kasus mengenai keputusan bermain basket yang dapat dilihat dalam tabel 2.1. Kasus ini diambil dari bahan mata kuliah *data mining* di STMIK AMIKOM Yogyakarta. Luthfi (2013).

Tabel 2.1. Keputusan Bermain Basket

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

Dalam kasus yang tertera pada tabel 2.1, akan dibuat pohon keputusan untuk menentukan dapat bermain basket atau tidak berdasarkan keadaan cuaca, *temperature*, kelembaban dan keadaan angin.

Secara umum algoritma C4.5 untuk membangun pohon keputusan dengan:

a. Pilih atribut sebagai akar.
b. Buat cabang untuk masing-masing nilai.
c. Bagi kasus dalam cabang.
d. Ulangi proses untuk masing-masing cabang yang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *gain* tertinggi dari atribut–atribut yang lain. Untuk menghitung *gain* digunakan rumus :

$$\text{Gain}(S,A) = \text{Entrophy}(S) - \sum_{i-1}^{n} \frac{|S_i|}{S} * \text{Entropy}(Si) \quad \text{....(2.1)}$$

dimana :

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|Si| : Jumlah kasus pada partisi ke-i

|S| : Jumlah kasus dalam S

Untuk menghitung nilai dari *entropy* dengan rumus :

$$\textit{Entropy}(S) = \sum_{i-1}^{n} - pi * \log_2 pi \quad \text{....(2.2)}$$

dimana :

S : Himpunan kasus

A : Fitur

n : Jumlah partisi S

P_i : Proporsi dari Si terhadap S

Berikut adalah langkah-langkah untuk mengerjakan permasalahan pada contoh tabel di atas dengan menggunakan algoritma 4.5.

a. Menghitung jumlah kasus untuk semua keputusan, jumlah kasus *Yes*, jumlah kasus *No*, *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *outlook*, *temperature*, *humidity* dan *windy*. Setelah itu, lakukan perhitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh tabel 2.2.

Tabel 2.2. Perhitungan *Node* 1

Node			**Jumlah kasus (S)**	**Tidak (S1)**	**Ya (S2)**	**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						0.000597

Tabel 2.2. Perhitungan *Node* 1 (lanjutan)

Node			**Jumlah kasus (S)**	**Tidak (S1)**	**Ya (S2)**	**Entropy**	**Gain**
		FALSE	8	2	6	0.811278124	
		TRUE	6	4	2	0.918295834	

Baris TOTAL kolom *Entropy* pada tabel 2.2 perhitungan *Node* 1 dihitung dengan rumus *entropy*:

$$\text{Entropy(Total)} = (-\tfrac{4}{14} * \log_2(\tfrac{4}{14})\ ) + (-\tfrac{10}{14} * \log_2(\tfrac{10}{14}))$$

$$\text{Entropy(Total)}=0.863120569$$

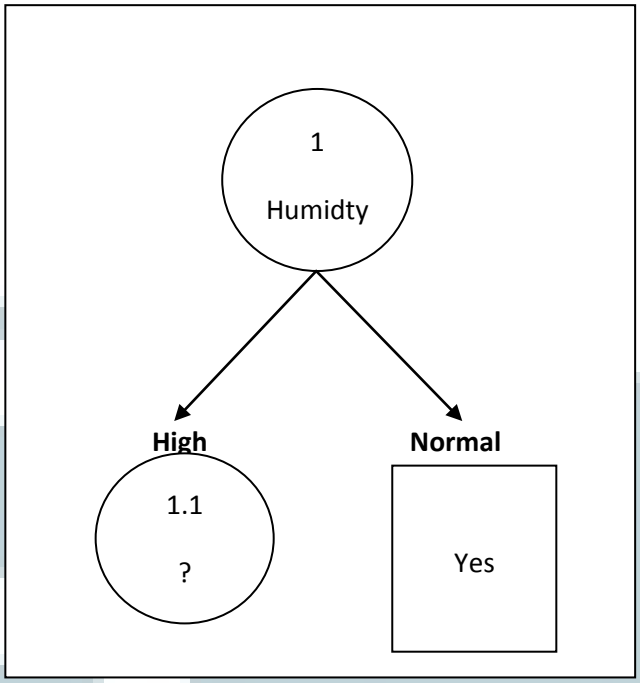
Baris OUTLOOK dihitung dengan menggunakan rumus *gain*:

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

$$\text{Gain(Total, Outlook)} = 0.863120569 - ((\tfrac{4}{14}*0) + (\tfrac{5}{14}*0.723) + (\tfrac{5}{14}*0.97))$$

$$\text{Gain(Total, Outlook)}= 0.23$$

Untuk mencari *node* akar, syaratnya bahwa atribut tersebut memiliki *gain* tertinggi, dan sesuai tabel di atas atribut *HUMIDITY* memiliki *gain* tertinggi dengan nilai 0.37. Dengan demikian, *HUMIDITY* dapat menjadi *node* akar. Atribut *HUMIDITY* memiliki dua atribut, yaitu *HIGH* dan *NORMAL*. Dari kedua nilai atribut tersebut, nilai atribut *NORMAL* sudah mengklasifikasikan kasus menjadi satu yaitu keputusan-nya *Yes*, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut *HIGH* masih perlu dilakukan perhitungan lagi karena memiliki hasil *Yes* dan *No*. Dari hasil tersebut dapat digambarkan pohon keputusan sementara seperti gambar 2.1.



Gambar 2.1 Pohon Keputusan Hasil Perhitungan *Node* 1

b. Menghitung jumlah kasus untuk semua keputusan, jumlah kasus *Yes*, jumlah kasus *No*, *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *outlook*, *temperature* dan *windy* yang dapat menjadi *node* cabang dari nilai atribut *high*. Setelah itu, lakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukan oleh tabel 2.3.

Tabel 2.3 Perhitungan *Node* 1.1

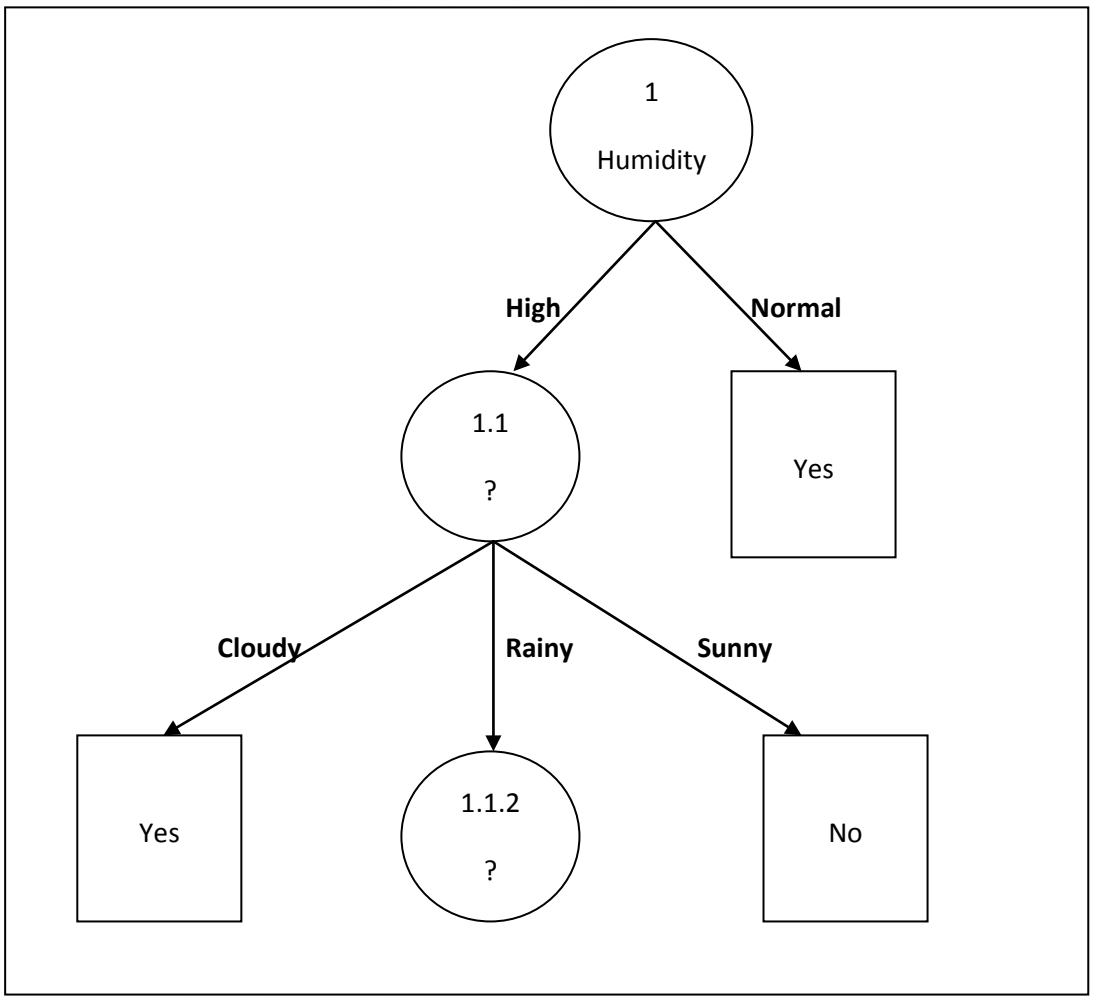
Node			**Jumlah kasus (S)**	**Tidak (S1)**	**Ya (S2)**	**Entropy**	**Gain**
1.1	HUMIDITY-HIGH		7	4	3	0.985228136	
	OUTLOOK						0.69951385
		CLOUDY	2	0	2	0	
		RAINY	2	1	1	1	
		SUNNY	3	3	0	0	
	TEMPERATURE						0.020244207
		COOL	0	0	0	0	

Tabel 2.3 Perhitungan *Node* 1.1 (lanjutan)

Node			**Jumlah kasus (S)**	**Tidak (S1)**	**Ya (S2)**	**Entropy**	**Gain**
		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	

Pada tabel 2.3 atribut *OUTLOOK* memiliki *gain* tertinggi dengan nilai 0.67. Dengan demikian, *OUTLOOK* dapat menjadi *node* cabang dari nilai atribut *HIGH*. Atribut *OUTLOOK* memiliki tiga atribut, yaitu *CLOUDY*, *RAINY* dan *SUNNY*. Dari ketiga nilai atribut tersebut, nilai atribut *CLOUDY* sudah mengklasifikasikan kasus menjadi satu, yaitu keputusan-nya *Yes*, dan nilai atribut *SUNNY* sudah mengklasifikasikan kasus menjadi satu, yaitu keputusan-nya *No*, sehingga tidak perlu dilakukan perhitungan lebih lanjut, tetapi untuk nilai atribut *RAINY* masih perlu dilakukan perhitungan lagi karena memiliki hasil *Yes* dan *No.*

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



Gambar 2.2 Pohon Keputusan Hasil Perhitungan *Node* 1.1

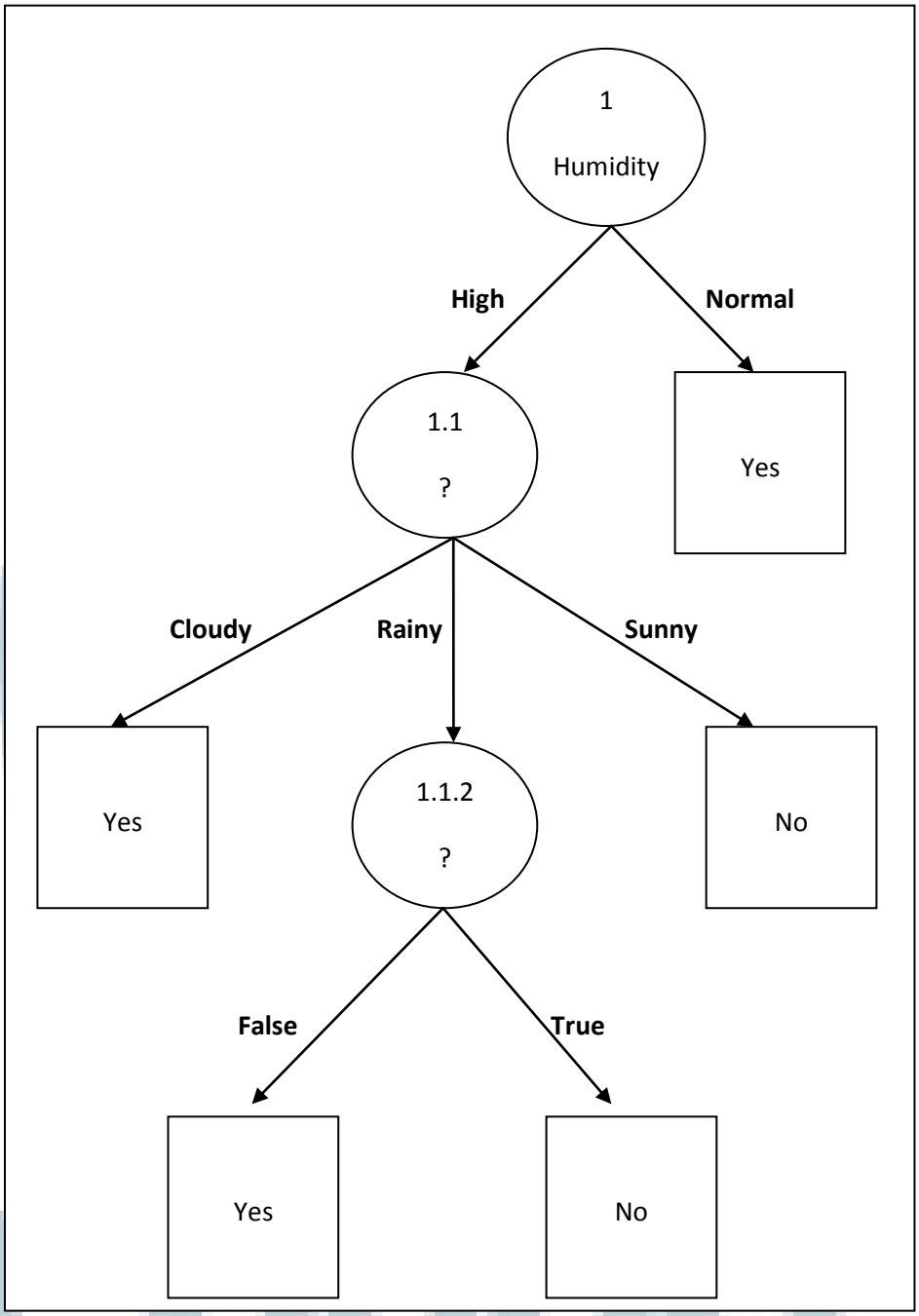
c. Menghitung jumlah kasus untuk semua keputusan, jumlah kasus *Yes*, jumlah kasus *No*, *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut *temperature* dan *windy* yang dapat menjadi *node* cabang dari nilai atribut *rainy*. Setelah itu, lakukan penghitungan *Gain* untuk masing-masing atribut. Hasil perhitungan ditunjukkan oleh tabel 2.4.

Tabel 2.4. Perhitungan *Node* 1.1.2

Node			**Jumlah kasus (S)**	**Tidak (S1)**	**Ya (S2)**	**Entropy**	**Gain**
1.1.2	HUMIDITY-HIGH dan OUTLOOK-RAINY		2	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	

Pada tabel 2.4 atribut *WINDY* memiliki *gain* tertinggi dengan nilai 1. Dengan demikian *WINDY* dapat menjadi *node* cabang dari nilai atribut *RAINY*. Atribut *WINDY* memiliki dua atribut, yaitu *FALSE* dan *TRUE*. Dari kedua nilai atribut tersebut, nilai atribut *FALSE* sudah mengklasifikasikan kasus menjadi satu, yaitu keputusan-nya *Yes,* dan nilai atribut *TRUE* sudah mengklasifikasikan kasus menjadi satu, yaitu keputusan-nya *No,* sehingga tidak perlu dilakukan perhitungan lebih lanjut.

Pohon keputusan pada tahap akhir ditunjukkan pada gambar 2.3 berikut.



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.