

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Listrik**

Listrik adalah rangkaian fenomena fisika yang berhubungan dengan kehadiran dan aliran muatan listrik. Muatan listrik adalah muatan dasar yang dimiliki suatu benda, yang membuatnya mengalami gaya pada benda lain yang berdekatan dan juga memiliki muatan listrik. Listrik menimbulkan berbagai macam efek yang telah umum diketahui, seperti petir, listrik statis, induksi elektromagnetik dan arus listrik. Arus listrik adalah banyaknya muatan listrik yang disebabkan dari pergerakan elektron-elektron, mengalir melalui suatu titik dalam sirkuit listrik tiap satuan waktu.

Energi listrik merupakan salah satu bentuk energi yang paling banyak digunakan manusia selain minyak dan gas bumi (migas) untuk aktivitas sehari-hari. Listrik tidak dapat dipisahkan dalam kehidupan manusia. Tanpa listrik dunia akan gelap gulita. Dengan energi listrik kita dapat beraktivitas di malam hari. Listrik juga membantu kita untuk mempermudah pekerjaan-pekerjaan berat yang tidak dapat dikerjakan sendiri.

#### **2.2 Perusahaan Listrik Negara (PLN)**

Perusahaan Listrik Negara (disingkat PLN) atau nama resminya adalah PT. PLN (Persero) adalah sebuah BUMN yang mengurus semua aspek kelistrikan yang ada di Indonesia. Ketenagalistrikan di Indonesia dimulai pada akhir abad ke-19, ketika beberapa perusahaan Belanda mendirikan pembangkitan tenaga listrik untuk keperluan sendiri. Pengusahaan tenaga listrik untuk kepentingan umum dimulai sejak perusahaan swasta Belanda N.V. NIGM memperluas usahanya di bidang tenaga listrik, yang semula hanya bergerak di bidang gas. Kemudian meluas dengan berdirinya perusahaan swasta lainnya.

Unit PT. PLN (Persero) dibagi dalam beberapa Wilayah untuk mengurus Pembangkitan, Penyaluran (Transmisi) dan Pengaturan Beban, dan Distribusi kepada pelanggan. Namun khusus untuk kawasan dengan listrik terinterkoneksi Jawa - Bali bagian unit-unit dibagi tersendiri, untuk Pembangkitan tersendiri, Penyaluran (Transmisi) tersendiri, Pengaturan Beban tersendiri dan Distribusi tersendiri. Khusus untuk pembangkitan listrik kebanyakan pembangkitan listrik di Indonesia dipasok oleh Perusahaan Swasta walaupun ada beberapa milik PLN.

### **2.3 Proses Penyampaian Listrik ke Pelanggan**

Proses penyampaian listrik ke pelanggan dimulai dengan pembangkitan listrik pada pusat-pusat listrik seperti PLTA, PLTU, PLTG, PLTP, dan PLTD. Kemudian disalurkan melalui saluran transmisi setelah terlebih dahulu dinaikkan tegangannya oleh transformator penaik tegangan yang ada di pusat listrik. Saluran transmisi ada yang berupa saluran udara dan ada pula yang berupa kabel tanah (Marsudi, D, 2006).

Setelah tenaga listrik disalurkan melalui saluran transmisi maka tenaga listrik sampai ke Gardu Induk untuk diturunkan tegangannya melalui transformator penurun tegangan menjadi tegangan menengah atau yang disebut tegangan distributor premier. Setelah tenaga listrik disalurkan melalui jaringan primer maka kemudian listrik diturunkan tegangannya dalam gardu-gardu distribusi menjadi tegangan rendah dengan tegangan 380/220 Volt atau 220/127 Volt, kemudian disalurkan melalui Jaringan Tegangan Rendah untuk selanjutnya disalurkan ke rumah-rumah pelanggan PLN melalui Sambungan Rumah (Marsudi, D, 2006).

Setelah tenaga listrik melalui Jaringan Tegangan Menengah, Jaringan Tegangan Rendah dan Sambungan Rendah maka tenaga listrik selanjutnya melalui alat pembatas daya dan KWH meter (Marsudi, D, 2006).

## 2.4 Data Mining

Nama data mining sebenarnya mulai dikenal sejak tahun 1990, ketika pekerjaan pemanfaatan data menjadi sesuatu yang penting dalam berbagai bidang, mulai dari bidang akademik, bisnis, hingga medis (Gorunescu, 2011). Data mining dapat diterapkan pada berbagai bidang yang mempunyai sejumlah data, tetapi karena wilayah penelitian dengan sejarah yang belumlah, dan belum melewati masa ‘remaja’, maka data mining masih diperdebatkan posisi bidang pengetahuan yang memilikinya. Maka, Daryl Pregibon menyatakan bahwa “data mining adalah campuran dari statistik, kecerdasan buatan, dan riset basis data” yang masih berkembang (Gonunescu, 2011).

Data mining yang disebut juga dengan *Knowledge-Discovery in Database* (KDD) adalah sebuah proses secara 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*). Proses KDD ini terdiri dari langkah-langkah sebagai berikut (Han, J. dan M. Kamber. 2006):

1. *Data Cleaning*, proses menghapus data yang tidak konsisten dan kotor.
2. *Data Integration*, penggabungan beberapa sumber data.
3. *Data Selection*, pengambilan data yang akan dipakai dari sumber data.
4. *Data Transformation*, proses dimana data ditransformasikan menjadi bentuk yang sesuai untuk diproses dalam data mining.
5. *Data Mining*, suatu proses yang penting dengan melibatkan metode untuk menghasilkan suatu pola data.
6. *Pattern Evaluation*, proses untuk menguji kebenaran dari pola data yang mewakili *knowledge* yang ada didalam data itu sendiri.
7. *Knowledge Presentation*, proses visualisasi dan teknik menyajikan *knowledge* digunakan untuk menampilkan *knowledge* hasil *mining* kepada *user*.

## 2.5 Metode Data Mining

Pada umumnya metode *data mining* dapat dikelompokkan kedalam dua kategori yaitu *deskriptif* dan *prediktif*. Metode *deskriptif* bertujuan untuk mencari pola yang dapat dimengeti oleh manusia yang menjelaskan karakteristik dari data. Metode *prediktif* menggunakan ciri-ciri tertentu dari data untuk melakukan prediksi. Metode-metode yang ada dalam data mining adalah sebagai berikut:

### 1. *Classification*

Klasifikasi (*Classification*) merupakan proses untuk menemukan sekumpulan model yang menjelaskan dan membedakan kelas-kelas data, sehingga model tersebut dapat digunakan untuk memprediksi nilai suatu kelas yang belum diketahui pada sebuah objek. Untuk mendapatkan model, kita harus melakukan analisis terhadap data latih (*training set*). Sedangkan data uji (*test set*) digunakan untuk mengetahui tingkat akurasi dari model yang telah dihasilkan. Dalam proses klasifikasi pohon keputusan tradisional, fitur (atribut) dari tupel adalah kategorikal atau numerikal. Biasanya definisi ketepatan nilai (*point value*) sudah didefinisikan di awal. Pada banyak aplikasi nyata, terkadang muncul suatu nilai yang tidak pasti. (Tsang, Smith., 2009). Klasifikasi dapat digunakan untuk memprediksi nama atau nilai kelas dari suatu objek data. Metode inilah yang digunakan dalam tugas akhir ini.

### 2. *Clustering*

Pengelompokkan (*Clustering*) merupakan proses untuk melakukan segmentasi. Digunakan untuk melakukan pengelompokkan secara alami terhadap atribut suatu set data. Termasuk kedalam *unsupervised task*. Contoh *clustering* seperti mengelompokkan dokumen berdasarkan topiknya.

### 3. *Association*

Tujuan dari metode ini yaitu untuk menghasilkan sejumlah rule yang menjelaskan sejumlah data yang terhubung kuat satu dengan yang lainnya. Sebagai contoh *association analysis* dapat digunakan untuk menentukan produk yang datang dibeli secara bersamaan oleh banyak pelanggan atau bisa juga disebut dengan *market basket analysis*.

#### 4. *Regression*

*Regression* mirip dengan klasifikasi. Perbedaan utamanya adalah terletak pada atribut yang diprediksi berupa nilai yang kontinyu.

#### 5. *Forecasting*

Prediksi (*Forecasting*) berfungsi untuk melakukan prediksi kejadian yang akan datang berdasarkan data sejarah yang ada.

#### 6. *Sequence Analysis*

Tujuan dari metode ini adalah untuk mengenali pola dari data diskrit. Sebagai contoh adalah menemukan kelompok *gen* dengan tingkat ekspresi yang mirip.

#### 7. *Deviation Analysis*

Tujuan dari metode ini adalah untuk menemukan penyebab perbedaan antara data yang satu dengan data yang lain dan biasa disebut sebagai *oulier detection*. Sebagai contoh adalah apakah sudah terjadi penipuan terhadap pengguna kredit dengan melihat catatan transaksi yang tersimpan dalam basis data perusahaan kartu kredit. (Santosa, Budi. 2007).

## 2.6 **Klasifikasi**

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu (1) pembangunan model sebagai prototipe untuk disimpan sebagai memori dan (2) penggunaan model tersebut untuk melakukan pengenalan/ klasifikasi /prediksi pada suatu objek data lain agar diketahui dikelas mana objek data tersebut dalam model yang sudah disimpannya. Model dalam klasifikasi mempunyai arti yang sama dengan kotak hitam, dimana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban sebagai keluaran dari hasil pemikirannya. (Prasetyo, E. 2012).

Tahapan dari klasifikasi dalam data mining terdiri dari (Han, J. dan M. Kamber. 2006) :

## 1. Pembangunan Model

Pada tahapan ini dibuat sebuah model untuk menyelesaikan masalah klasifikasi class atau atribut dalam data. Tahap ini merupakan fase pelatihan, dimana data latih dianalisis menggunakan algoritma klasifikasi, sehingga model pembelajaran direpresentasikan dalam bentuk aturan klasifikasi.

## 2. Penerapan Model

Pada tahapan ini model yang sudah dibangun sebelumnya digunakan untuk menentukan atribut/kelas dari sebuah data baru yang atribut/kelasnya belum diketahui sebelumnya. Tahap ini digunakan untuk memperkirakan keakuratan aturan klasifikasi terhadap data uji. Jika model dapat diterima, maka aturan dapat diterapkan terhadap klasifikasi data baru.

### 2.7 *Decision Tree*

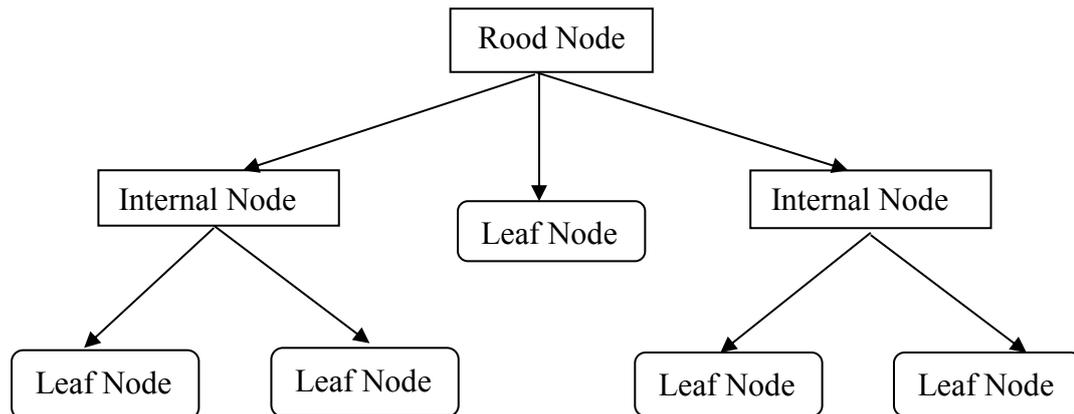
*Decision tree* atau pohon keputusan adalah pohon yang digunakan sebagai prosedur penalaran untuk mendapatkan jawaban dari masalah yang dimasukkan (Prasetyo, E. 2014). *Decision tree* banyak digunakan untuk menyelesaikan kasus penentuan keputusan seperti di bidang kedokteran (diagnosis penyakit pasien), ilmu komputer (struktur data), psikologi (teori pengambilan keputusan), dan sebagainya.

Karakteristik dari *Decision tree* dapat dibentuk sejumlah elemen sebagai berikut (Tan, 2006) :

- a. Node akar, tidak mempunyai lengan masukan dan mempunyai nol atau lebih lengan keluaran.
- b. Node internal, setiap node yang bukan daun (nonterminal) yang mempunyai tepat satu lengan masukan dan dua atau lebih lengan keluaran. Node ini menyatakan pengujian yang didasarkan pada nilai fitur.
- c. Lengan, setiap cabang menyatakan nilai hasil pengujian di node bukan daun.

- d. Node daun (terminal), node ini yang mempunyai tepat satu lengan masukan dan tidak mempunyai lengan keluaran. Node ini menyatakan label kelas (keputusan).

Contoh dari model pohon keputusan yaitu seperti pada **gambar 2.1** berikut:



**Gambar 2.1** Model *Decision Tree*

## 2.8 *Decision Tree C4.5*

Algoritma C4.5 diperkenalkan oleh Quinlan (1996) sebagai versi perbaikan dari ID3. Dalam ID3, induksi decision tree hanya bisa dilakukan pada fitur bertipe kategorikal (nominal atau ordinal), sedangkan tipe numerik (interval atau rasio) tidak dapat digunakan (Eko Prasetyo, 2014).

Yang menjadi hal penting dalam induksi decision tree adalah bagaimana menyatakan syarat pengujian pada node. Ada 3 kelompok penting dalam syarat pengujian node :

### 1. Fitur biner

Adalah Fitur yang hanya mempunyai dua nilai berbeda. Syarat pengujian ketika fitur ini menjadi node (akar maupun interval) hanya punya dua pilihan cabang.

### 2. Fitur kategorikal

Untuk fitur yang nilainya bertipe kategorikal (nominal atau ordinal) bisa mempunyai beberapa nilai berbeda. Secara umum ada 2 pemecahan yaitu pemecahan biner (*binary splitting*) dan (*multi splitting*).

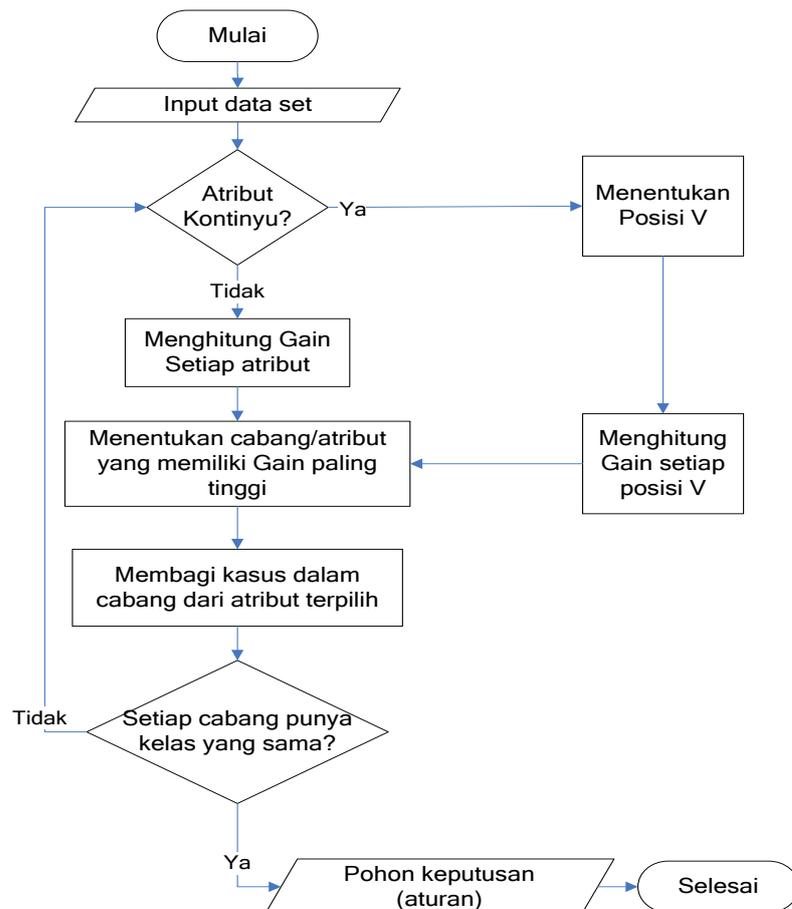
### 3. Fitur numerik

Untuk fitur bertipe numerik, Syarat pengujian dalam node (akar maupun internal) dinyatakan dengan pengujian perbandingan ( $A \leq V$ ) atau ( $A > V$ ) dengan hasil biner.

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.

Berikut ini akan dijelaskan secara lebih detail algoritma C4.5 menggunakan *flowchart* yang disajikan pada gambar 2.2.



**Gambar 2.2** Flowchart algoritma Decision Tree C4.5

Untuk memilih atribut sebagai simpul akar (*root node*) atau simpul dalam (*internal node*), didasarkan pada nilai *information gain* tertinggi dari atribut-atribut yang ada. Sebelum perhitungan *information gain*, akan dilakukan perhitungan *entropy*. *Entropy* merupakan distribusi probabilitas dalam teori informasi dan diadopsi kedalam algoritma C4.5 untuk mengukur tingkat homogenitas distribusi kelas dari sebuah himpunan data (*data set*). Semakin tinggi tingkat *entropy* dari sebuah data maka semakin homogen distribusi kelas pada data tersebut. Perhitungan *information gain* menggunakan rumus 2.1, sedangkan *entropy* menggunakan rumus 2.2.

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

dimana,

- S : Himpunan kasus
- A : Atribut
- n : Jumlah partisi atribut A
- |S<sub>i</sub>| : Jumlah kasus pada partisi ke i
- |S| : Jumlah kasus dalam S

$$Entropy(S) = - \sum_{i=1}^n p_i * \log_2 p_i \quad \dots\dots\dots (2.2)$$

dimana,

- S : Himpunan kasus
- A : Fitur
- n : Jumlah partisi S
- p<sub>i</sub> : Proporsi dari S<sub>i</sub> terhadap S

Selain *Information Gain* kriteria yang lain untuk memilih atribut sebagai pemecah adalah *Rasio Gain*. Perhitungan *rasio gain* menggunakan rumus 2.3, sedangkan *split information* menggunakan rumus 2.4.

$$GainRasio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)} \quad \dots\dots\dots (2.3)$$

$$SplitInformation(S,A) = - \sum_{i=1}^c \frac{S_i}{S} \log_2 \frac{S_i}{S} \quad \dots\dots\dots (2.4)$$

dimana  $S_1$  sampai  $S_c$  adalah  $c$  subset yang dihasilkan dari pemecahan  $S$  dengan menggunakan atribut  $A$  yang mempunyai sebanyak  $c$  nilai.

Untuk mengukur nilai akurasi yang didapat dari hasil pengujian, menggunakan rumus 2.5. Sedangkan untuk mengukur tingkat kesalahannya menggunakan rumus 2.6.

$$Akurasi = \frac{\text{Jumlah data yang diprediksi secara benar}}{\text{Jumlah prediksi yang dilakukan}} \dots\dots\dots (2.5)$$

$$Laju\ error = \frac{\text{Jumlah data yang diprediksi secara salah}}{\text{Jumlah prediksi yang dilakukan}} \dots\dots\dots (2.6)$$

Sensitivitas akan mengukur proporsi positif asli yang dikenali (diprediksi) secara benar sebagai positif asli. Rumus perhitungannya menggunakan rumus 2.7. Sedangkan spesifisitas akan mengukur proporsi negatif asli yang dikenali (diprediksi) secara benar sebagai negatif asli. Rumus perhitungannya menggunakan rumus 2.8.

$$Sensitivitas = \frac{TP}{TP + FN} \dots\dots\dots (2.7)$$

Keterangan :

TP : Kelas acc yang diprediksi secara benar sebagai Kelas acc

FN : Kelas acc yang diprediksi secara salah sebagai Kelas tolak

$$Spesifisitas = \frac{TN}{FP + TN} \dots\dots\dots (2.8)$$

Keterangan :

TN : Kelas tolak yang diprediksi secara benar sebagai Kelas tolak

FP : Kelas tolak yang diprediksi secara salah sebagai Kelas acc

## 2.9 Contoh Perhitungan

Berikut ini akan dijelaskan ilustrasi dari alur proses perhitungan algoritma *Decision Tree C4.5*. Data set yang digunakan pada contoh ini adalah data untuk melakukan prediksi “apakah harus bermain *baseball* ?” dengan menjawab Ya atau Tidak. Atribut yang digunakan ada 4 yaitu Cuaca, Suhu, Kelembaban, dan Angin. Dimana atribut Suhu dan Kelembaban bertipe

numerik sedangkan Cuaca dan Angin bertipe kategorikal. Sedangkan kolom bermain adalah kelas tujuannya atau label kelasnya.

**Tabel 2.1** Contoh data set

<b>Cuaca</b>	<b>Suhu</b>	<b>Kelembaban</b>	<b>Angin</b>	<b>Bermain</b>
Cerah	85	85	Biasa	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Hujan	70	96	Biasa	Ya
Hujan	68	80	Biasa	Ya
Hujan	65	70	Kencang	Tidak
Mendung	64	65	Kencang	Ya
Cerah	72	95	Biasa	Tidak
Cerah	69	70	Biasa	Ya
Hujan	75	80	Biasa	Ya

<b>Cuaca</b>	<b>Suhu</b>	<b>Kelembaban</b>	<b>Angin</b>	<b>Bermain</b>
Cerah	75	70	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya
Hujan	71	80	Kencang	Tidak

Proses pertama adalah menghitung *entropy* untuk *node* akar (semua data) terhadap komposisi kelas.

Berikut adalah perhitungan *entropy* untuk semua data:

$$\begin{aligned}
 Entropy(S) &= -\frac{9}{14} * \log_2\left(\frac{9}{14}\right) - \frac{5}{14} * \log_2\left(\frac{5}{14}\right) \\
 &= 0.9403
 \end{aligned}$$

Selanjutnya, untuk fitur yang bertipe *numeric*, harus ditentukan posisi  $v$  yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai Gainnya

disajikan pada tabel 2.2. Nilai Gain tertinggi didapatkan pada posisi  $v = 70$ . Maka untuk fitur Suhu dilakukan diskretisasi pada  $v = 70$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.2** Posisi  $v$  untuk pemecahan fitur Suhu di *node* akar

Suhu	70		75		80	
	<=	>	<=	>	<=	>
<b>Ya</b>	4	5	7	2	7	2
<b>Tidak</b>	1	4	3	2	4	1
<b>Gain</b>	<b>0.0453</b>		<b>0.0251</b>		<b>0.0005</b>	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada tabel 2.3. Nilai *Gain* tertinggi didapatkan pada posisi  $v = 80$ . Maka untuk fitur Kelembaban dilakukan diskretisasi pada  $v = 80$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.3** Posisi  $v$  untuk pemecahan fitur Kelembaban di *node* akar

Kelembaban	70		75		80		85	
	<=	>	<=	>	<=	>	<=	>
<b>Ya</b>	2	7	3	6	7	2	7	2
<b>Tidak</b>	1	4	1	4	2	3	3	2
<b>Gain</b>	<b>0.0005</b>		<b>0.0150</b>		<b>0.1022</b>		<b>0.0251</b>	

Selanjutnya dihitung entropy untuk setiap nilai fitur terhadap kelas, kemudian dihitung gain untuk setiap fitur. Hasilnya disajikan pada tabel 2.4.

**Tabel 2.4** Hasil perhitungan *entropy* dan *gain* untuk *node* akar

Node		Jumlah	Ya	Tidak	Entropy	Gain
1	Total	14	9	5	0.9403	

		Cerah	5	2	3	0.9710
		Mendung	4	4	0	0
		Hujan	5	3	2	0.9710
		<=70	5	4	1	0.7219
		>70	9	5	4	0.9911
		<=80	9	7	2	0.7642
		>80	5	3	2	0.9710
		Pelan	8	6	2	0.8113
		Kencang	6	3	3	0.8113

Hasil yang didapat di tabel 2.4 menunjukkan bahwa *Gain* tertinggi ada di fitur Cuaca, maka Cuaca dijadikan sebagai *node* akar. Selanjutnya, dihitung posisi split untuk fitur Cuaca dengan menghitung *Rasio Gain*, selengkapnya disajikan pada tabel 2.5.

Hasil perhitungan *rasio gain* posisi *split* untuk *opsi* satu sebagai berikut:

$$\begin{aligned}
 SplitInfo(Semua, cuaca) &= \left( -\frac{5}{14} * \log_2 \left( \frac{5}{14} \right) \right) + \left( -\frac{4}{14} * \log_2 \left( \frac{4}{14} \right) \right) \\
 &\quad + \left( -\frac{5}{14} * \log_2 \left( \frac{5}{14} \right) \right) \\
 &= 1.5774
 \end{aligned}$$

$$RasioGain(Semua, cuaca) = \frac{0.2467}{1.5774}$$

$$= 0.16$$

Dengan cara yang sama, akan didapatkan nilai *rasio gain* untuk *opsi* yang lain.

Hasil ditabel 2.5 menunjukkan bahwa *rasio gain* tertinggi ada di *opsi* 4 yaitu *split* {cerah, hujan} dengan {mendung}. Itu artinya, cabang untuk akar ada 2, yaitu: {cerah, hujan} dan {mendung}, seperti ditunjukkan pada gambar 2.3.

**Tabel 2.5** Perhitungan *Rasio Gain* untuk fitur Cuaca

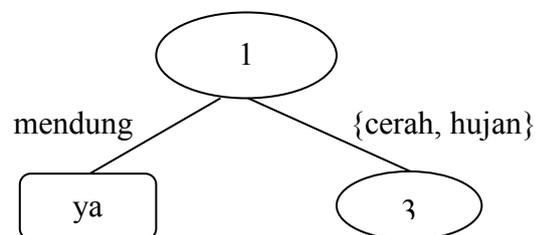
Node			Jumlah	Entropy	Gain	Rasio Gain
1	Total		14		0.2467	
		Cerah	5			
		Mendung	4			
		Hujan	5			
		Cerah	5			
		Mendung dan Hujan	9			
		Cerah, Mendung	9			
		Hujan	5			
		Cerah, Hujan	10			
		Mendung	4			

Hasil pemisahan data menurut *node* akar disajikan pada tabel 2.6.

**Tabel 2.6** Pemisahan data menurut *node* akar

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	85	85	Biasa	Tidak
Cerah	80	90	Kencang	Tidak
Hujan	70	96	Biasa	Ya
Hujan	68	80	Biasa	Ya
Hujan	65	70	Kencang	Tidak
Cerah	72	95	Biasa	Tidak
Cerah	69	70	Biasa	Ya
Hujan	75	80	Biasa	Ya
Cerah	75	70	Kencang	Ya
Hujan	71	80	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Untuk *node 2*, nilai *Entropy* yang didapat adalah 0 (karena semua baris memiliki kelas yang sama) maka dipastikan bahwa *node 2* menjadi daun, seperti ditunjukkan pada gambar 2.3.



**Gambar 2.3** Hasil pembentukan cabang di akar untuk kasus “apakah harus bermain *baseball* ?”

Selanjutnya, di *node 3*, harus dihitung dulu *entropy* untuk sisa data terhadap komposisi kelas yang tidak masuk dalam *node 2*.

Untuk fitur yang bertipe numerik, harus ditentukan lagi posisi  $v$  yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai *Gain* yang disajikan pada tabel 2.7. Nilai *Gain* tertinggi didapatkan pada posisi  $v = 75$ . Maka untuk fitur Suhu dilakukan diskretisasi pada  $v = 75$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.7** Posisi  $v$  untuk pemecahan fitur Suhu di *node 3*

Suhu	70		75		80	
	<=	>	<=	>	<=	>
<b>Ya</b>	3	2	5	0	5	0
<b>Tidak</b>	1	4	3	2	4	1
<b>Gain</b>	<b>0.1245</b>		<b>0.2365</b>		<b>0.1080</b>	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada tabel 2.8. Nilai *Gain* tertinggi didapatkan pada posisi  $v = 80$ . Maka untuk fitur Kelembaban dilakukan diskretisasi pada  $v = 80$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.8** Posisi  $v$  untuk pemecahan fitur Kelembaban di *node 3*

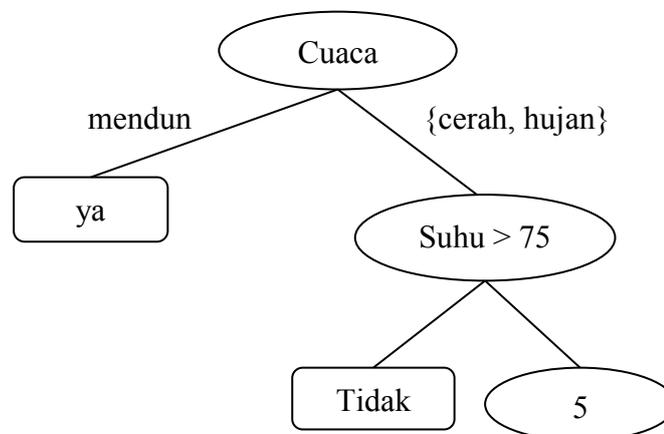
Kelembaban	70		75		80		85	
	<=	>	<=	>	<=	>	<=	>
<b>Ya</b>	2	3	2	3	4	3	4	1
<b>Tidak</b>	1	4	1	4	2	1	3	2
<b>Gain</b>	<b>0.0349</b>		<b>0.0349</b>		<b>0.1245</b>		<b>0.0349</b>	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.9.

**Tabel 2.9** Hasil perhitungan *entropy* dan *gain* untuk *node 3*

Node			Jumlah	Ya	Tidak	Entropy	Gain
3	Total		10	5	5	1.0000	
		Cerah	5	2	3	0.9710	
		Hujan	5	3	2	0.9710	
		$\leq 75$	8	5	3	0.9544	
		$> 75$	2	0	2	0	
		$\leq 80$	6	4	2	0.9183	
		$> 80$	4	1	3	0.8113	
		Pelan	6	4	2	0.9183	
		Kencang	4	1	3	0.8113	

Hasil yang ditunjukkan pada tabel 2.9 menunjukkan bahwa *gain* tertinggi ada di fitur Suhu, berarti fitur Suhu dijadikan syarat kondisi di *node 3*, seperti ditunjukkan pada gambar 2.4. Pemisahan datanya ditunjukkan pada tabel 2.10.



**Gambar 2.4** Hasil pembentukan cabang di *node 3* untuk kasus “apakah harus bermain *baseball*”

**Tabel 2.10** Pemisahan data menurut *node 3*

<b>Cuaca</b>	<b>Suhu</b>	<b>Kelembaban</b>	<b>Angin</b>	<b>Bermain</b>
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak
Cerah	72	95	Pelan	Tidak
Cerah	69	70	Pelan	Ya
<b>Cuaca</b>	<b>Suhu</b>	<b>Kelembaban</b>	<b>Angin</b>	<b>Bermain</b>
Hujan	75	80	Pelan	Ya
Cerah	75	70	Kencang	Ya
Hujan	71	80	Kencang	Tidak
Cerah	85	85	Pelan	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

*Node 4*, untuk cabang suhu  $> 75$  dimana label kelas bernilai tidak, dipastikan mempunyai entropy 0, maka *node 4* (yang dituju) dijadikan daun. Seperti ditunjukkan pada gambar 2.4.

Selanjutnya pada *node 5*, untuk fitur numerik kembali dilakukan perhitungan posisi  $v$  yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai *Gain* yang disajikan pada tabel 2.11. Nilai *Gain* tertinggi didapatkan pada posisi  $v = 70$ . Maka untuk fitur Suhu dilakukan diskretisasi pada  $v = 70$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.11** Posisi  $v$  untuk pemecahan fitur Suhu di *node 5*

Suhu	70		75	
	$\leq$	$>$	$\leq$	$>$
<b>Ya</b>	3	2	5	0
<b>Tidak</b>	1	2	3	0
<b>Gain</b>	<b>0.0488</b>		<b>0</b>	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada tabel 2.12. Nilai *Gain* tertinggi didapatkan pada posisi  $v = 80$ . Maka untuk fitur Kelembaban dilakukan diskretisasi pada  $v = 80$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.12** Posisi  $v$  untuk pemecahan fitur Kelembaban di *node 5*

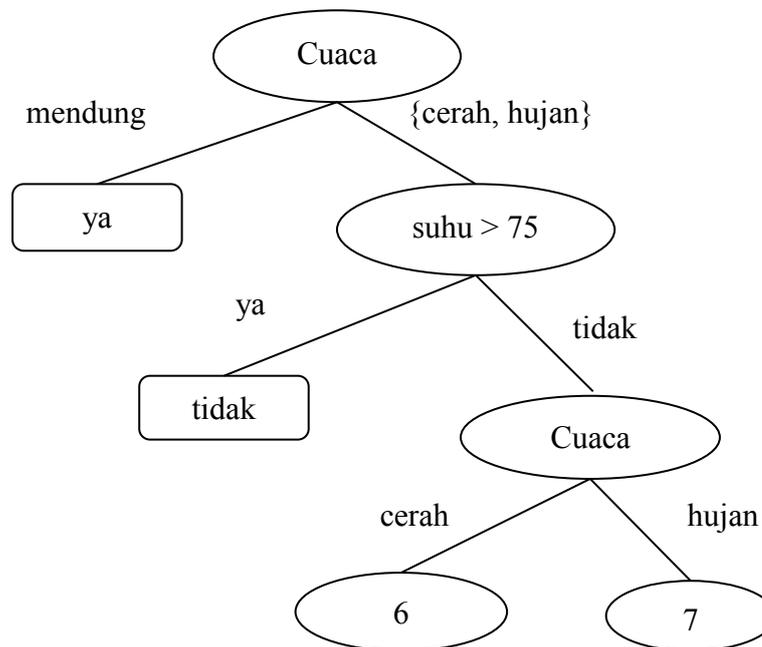
Kelembaban	70		75		80	
	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$
<b>Ya</b>	2	3	2	3	4	1
<b>Tidak</b>	1	2	1	2	2	1
<b>Gain</b>	<b>0.0032</b>		<b>0.0032</b>		<b>0.0157</b>	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.13.

**Tabel 2.13** Hasil perhitungan *entropy* dan *gain* untuk *node 5*

Node			Jumlah	Ya	Tidak	Entropy	Gain
5	Total		8	5	3	0.9544	
		Cerah	3	2	3	0.3900	
		Hujan	5	3	2	0.9710	
		$\leq 70$	4	3	1	0.8113	
		$> 70$	4	2	2	1.0000	

	Kelembaban					0.0157
		<=80	6	4	2	0.9183
		>80	2	1	1	1.0000
		Pelan	5	4	1	0.7219
		Kencang	3	1	2	0.9183



**Gambar 2.5** Hasil pembentukan cabang di *node 5* untuk kasus apakah harus bermain *baseball*

**Tabel 2.14** Pemisahan data menurut *node 5*

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	72	95	Pelan	Tidak
Cerah	69	70	Pelan	Ya
Cerah	75	70	Kencang	Ya
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak

Hujan	75	80	Pelan	Ya
Hujan	71	80	Kencang	Tidak
Cerah	85	85	Pelan	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Pada perhitungan berikutnya, fitur Cuaca tidak digunakan lagi karena kedua nilai berbeda yang tersisa sudah digunakan untuk syarat pengujian di *node 5*. Selanjutnya pada *node 6*, untuk fitur numerik kembali dilakukan perhitungan posisi  $v$  yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai *Gain* yang disajikan pada tabel 2.15. Nilai *Gain* tertinggi didapatkan pada posisi  $v = 70$ . Maka untuk fitur Suhu dilakukan diskretisasi pada  $v = 70$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.15** Posisi  $v$  untuk pemecahan fitur Suhu di *node 6*

Suhu	70		75	
	$\leq$	$>$	$\leq$	$>$
Ya	1	1	2	0
Tidak	0	1	1	0
Gain	0.2516		0	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada tabel 2.16. Nilai *Gain* didapatkan pada posisi  $v = 70$ . Maka untuk fitur Kelembaban dilakukan diskretisasi pada  $v = 70$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.16** Posisi  $v$  untuk pemecahan fitur Kelembaban di *node 6*

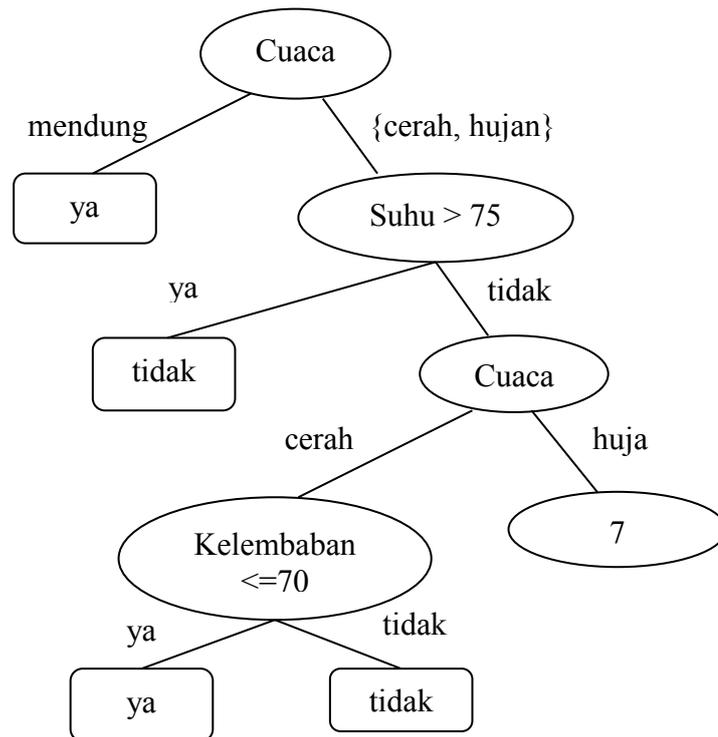
Kelembaban	70	
	$\leq$	$>$
<b>Ya</b>	2	0
<b>Tidak</b>	0	1
<b>Gain</b>	<b>0.9183</b>	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.17.

**Tabel 2.17** Hasil perhitungan *entropy* dan *gain* untuk *node 6*

Node			Jumlah	Ya	Tidak	Entropy	Gain
6	Total		3	2	1	0.9183	
		$\leq 70$	1	1	0	0	
		$> 70$	2	1	1	1.0000	
		$\leq 70$	2	2	0	0	
		$> 70$	1	0	1	0	
		Pelan	2	1	1	1.0000	
		Kencang	1	1	0	0	

Hasil yang ditunjukkan pada tabel 2.17 menunjukkan bahwa *gain* tertinggi ada di fitur Kelembaban, berarti fitur Kelembaban dijadikan syarat kondisi di *node 6*, seperti ditunjukkan pada gambar 2.6. Pemisahan datanya ditunjukkan pada tabel 2.18.



**Gambar 2.6** Hasil pembentukan cabang di *node 6* untuk kasus “apakah harus bermain *baseball*”

**Tabel 2.18** Pemisahan data menurut *node 6*

Cuaca	Suhu	Kelembaban	Angin	Bermain
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak
Hujan	75	80	Pelan	Ya
Hujan	71	80	Kencang	Tidak
Cerah	69	70	Pelan	Ya
Cerah	75	70	Kencang	Ya
Cerah	72	95	Pelan	Tidak
Cerah	85	85	Pelan	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya

Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Jika diamati tabel 2.18, untuk *node* 8 dan 9 (cabang dari *node* 6) dipastikan menjadi daun karena nilai *entropy* 0, dimana masing-masing cabang jatuh pada label kelas yang sama. Proses berikutnya dilanjutkan untuk *node* 7.

Selanjutnya pada *node* 7, untuk fitur numerik kembali dilakukan perhitungan posisi  $v$  yang terbaik untuk pemecahan. Dalam contoh ini, digunakan pemecahan biner. Hasil uji coba pada fitur Suhu dengan menghitung nilai *Gain* yang disajikan pada tabel 2.19. Nilai *Gain* tertinggi didapatkan hanya pada posisi  $v = 70$ . Maka untuk fitur Suhu dilakukan diskretisasi pada  $v = 70$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.19** Posisi  $v$  untuk pemecahan fitur Suhu di *node* 7

Suhu	70	
	<=	>
Ya	2	1
Tidak	0	1
Gain	<b>0.0200</b>	

Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada tabel 2.20. Nilai *Gain* didapatkan hanya pada posisi  $v = 80$ . Maka untuk fitur Kelembaban dilakukan diskretisasi pada  $v = 80$  ketika menghitung *Entropy* dan *Gain* pada semua fitur.

**Tabel 2.20** Posisi  $v$  untuk pemecahan fitur Kelembaban di *node 7*

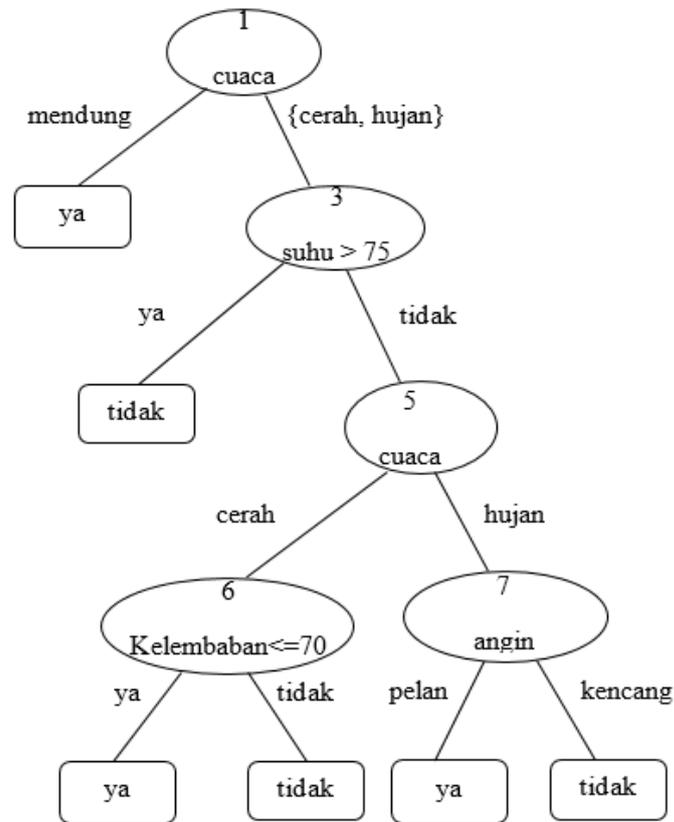
Kelembaban	70	
	$\leq$	$>$
<b>Ya</b>	2	1
<b>Tidak</b>	2	0
<b>Gain</b>	<b>0.1710</b>	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.21.

**Tabel 2.21** Hasil perhitungan *entropy* dan *gain* untuk *node 7*

Node			Jumlah	Ya	Tidak	Entropy	Gain
7	Total		5	3	2	0.9710	
		$\leq 70$	3	2	1	0.9183	
		$> 70$	2	1	1	1.0000	
		$\leq 80$	4	2	2	1.0000	
		$> 80$	1	1	0	0	
		Pelan	3	3	0	0	
		Kencang	2	0	2	0	

Hasil yang ditunjukkan pada tabel 2.21 menunjukkan bahwa *gain* tertinggi ada di fitur Angin, berarti fitur Angin dijadikan syarat kondisi di *node 7*, seperti ditunjukkan pada gambar 2.8. Pemisahan datanya ditunjukkan pada tabel 2.22.



**Gambar 2.7** Hasil pembentukan cabang di *node 7* untuk kasus “apakah harus bermain *baseball*”

**Tabel 2.22** Pemisahan data menurut *node 7*

Cuaca	Suhu	Kelembaban	Angin	Bermain
Hujan	70	96	Pelan	Ya
Hujan	68	80	Pelan	Ya
Hujan	75	80	Pelan	Ya
Hujan	65	70	Kencang	Tidak
Hujan	71	80	Kencang	Tidak
Cerah	69	70	Pelan	Ya
Cerah	75	70	Kencang	Ya
Cerah	72	95	Pelan	Tidak
Cerah	85	85	Pelan	Tidak

Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Mendung	64	65	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya

Jika diamati tabel 2.21, untuk *node* 10 dan 11 (cabang dari *node* 7) dipastikan menjadi daun karena nilai *entropy* 0, dimana masing-masing cabang jatuh pada label kelas yang sama.

Karena tidak ada lagi *node* yang harus diproses, maka induksi *decision tree* dinyatakan selesai. Hasil akhir *decision tree* seperti disajikan pada gambar 2.8.

Bentuk aturan *IF THEN* untuk *decision tree* sebagai berikut:

*IF* cuaca= mendung *THEN* playball = ya

*IF* cuaca= {cerah, hujan} *AND* suhu > 75 *THEN* playball = tidak

*IF* cuaca= cerah *AND* suhu <=75 *AND* kelembaban<=70 *THEN* playball = ya

*IF* cuaca=cerah *AND* suhu<=75 *AND* kelembaban >70 *THEN* playball= tidak

*IF* cuaca= hujan *AND* suhu <=75 *AND* angin = pelan *THEN* playball = ya

*IF* cuaca= hujan *AND* suhu<=75 *AND* angin= kencang *THEN* playball = tidak

## 2.10 Penelitian Sebelumnya

Penelitian selanjutnya dilakukan oleh Lailatul Qomariyah, Mahasiswa Teknik Informatika Universitas Muhammadiyah Gresik yang dilakukan tahun 2016. Penelitian ini berjudul "Klasifikasi Calon Pendorong Darah Dengan Metode Decision Tree C4.5 Di Kabupaten Gresik (Studi Kasus: PMI Kabupaten Gresik). Penelitian ini bertujuan untuk menentukan apakah calon pendonor darah dapat melakukan donor darah atau tidak. Atribut yang

digunakan dalam penelitian ini yaitu usia, kadar hemoglobin, berat badan dan tekanan darah. Data yang digunakan sebanyak 60 data yang diperoleh dari PMI Kabupaten Gresik. Untuk melakukan pengujian data dilakukan penghitungan nilai akurasi, laju error, sensitivitas dan spesifitas, pengujian dilakukan sebanyak 2 kali dalam 3 variasi data. Variasi data pertama yaitu 30 data latih dan 30 data uji, variasi kedua yaitu 36 data latih dan 24 data uji, dan variasi ketiga sebanyak 48 data latih dan 12 data uji. Dari ketiga variasi percobaan tersebut diperoleh rata-rata nilai akurasi sebesar 86,80%, rata-rata laju error sebesar 13,19%, rata-rata sensitivitas sebesar 91,39%, dan rata-rata spesifitas sebesar 82,22%. Dari ketiga variasi percobaan tersebut diketahui nilai akurasi tertinggi terjadi pada percobaan dengan 48 data latih dan 12 data uji dengan nilai akurasi mencapai 91,67%.

Sedangkan penelitian lain yang digunakan sebagai rujukan dalam tugas akhir ini adalah penelitian yang dibuat oleh Alfa Saleh, Mahasiswa Universitas Potensi Utama yang dilakukan pada tahun 2015. Penelitian yang dibuat berjudul "Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga". Dengan menggunakan metode tersebut diperoleh hasil prosentase sebesar 78,3333 %, dimana dari 60 data uji terdapat 47 data yang diklasifikasikan dengan benar.