

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **1.1. Tinjauan Pustaka**

Dalam penelitian ini, penulis menggunakan beberapa sumber pustaka. Sumber pustaka yang dimaksudkan untuk digunakan sebagai pedoman dan pembandingan dalam penelitian yang akan penulis lakukan. Pustaka yang digunakan ditinjau dari masalah yang ada, objek penelitian dan metode yang digunakan dalam penelitian.

Beberapa penelitian tentang Data Mining yang menggunakan metode *Decision tree* yaitu

Penelitian tentang evaluasi kinerja motor servo dengan melakukan percobaan pada beban dan tegangan *input* yang berbeda dalam dua keadaan *loop* (*open loop* dan *close loop*) serta diklasifikasikan dalam kategori perputaran motor yang termasuk cepat, cukup cepat dan lambat. Data percobaan motor servo yang telah ada dianalisis menggunakan metode *decision tree* (Pohon Keputusan) dengan algoritma ID3. Metode *decision tree* merupakan metode yang merubah fakta yang sangat besar menjadi sebuah pohon keputusan yang merepresentasikan aturan-aturan. Pohon keputusan ini juga berguna untuk mengeksplorasi data, serta menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah *variabel* target. Berdasarkan hasil pengujian yang telah dilakukan dengan memasukkan data motor servo yang baru kemudian dianalisis dengan menggunakan metode pohon keputusan algoritma ID3 pada sistem aplikasi, hasil yang diperoleh sama dengan hasil kategori data sebenarnya dengan rincian

kecepatan 0-320 RPM masuk ke dalam kategori lambat, 321 – 710 RPM masuk ke dalam kategori cukup cepat dan 711 – 1070 RPM masuk ke dalam kategori cepat. Penelitian ini disusun oleh Aryansyah dan Wijaya (2014).

Penelitian tentang analisis data mining pada data penjualan batik mahkota untuk menentukan target pasar berdasarkan jenis batik. Batik Mahkota Laweyan memiliki jenis dan model batik yang bermacam macam. Banyak kombinasi jenis dan model yang telah dibuat baik untuk memenuhi pemesanan maupun untuk dijual di *showroom*. Ketersediaan data penjualan yang melimpah akan dimanfaatkan dalam bentuk analisis data mining sehingga dapat digunakan sebagai pendukung keputusan untuk membuat solusi bisnis yaitu menentukan target pasar. Penggunaan metode klasifikasi decision tree ID3 yang telah diimplementasikan pada aplikasi weka yaitu klasifikasi *decision tree*. Penerapan konsep pohon keputusan pada data di batik mahkota laweyan digunakan untuk menentukan target pasar berdasarkan jenis batik. Dengan pengimplementasian data mining menggunakan *decision tree* diketahui bahwa pada bulan januari, maret, april, juni, agustus, dan november batik print adalah jenis batik yang paling digemari di enam provinsi yang diteliti. Pada bulan September dan desember keenam provinsi memilih jenis batik lain seperti tulis dan cap. Penelitian ini disusun oleh Dodik Setyawan Handoko (2013).

Penelitian tentang pembelian cat yang kurang efektif, menyebabkan produk khususnya cat pada perusahaan ini kurang diminati oleh *customer*. Untuk memudahkan dalam pengambilan keputusan dalam proses pembelian cat di *departement* penjualan *home smart* medan yang selama ini pengambilan

keputusan seorang *manager* dalam mengambil keputusan dengan melihat seberapa dekat hubungan *supplier* dan seberapa banyak dana sponsor yang diberikan kepada perusahaan. Selain harga, *type*, merek juga masih kalah saing dengan perusahaan lain. Pembelian cat dengan menggunakan metode data mining khususnya algoritma ID3 akan bermanfaat sekali dalam proses pengambilan keputusan dalam pembelian cat pada *home smart* medan. Dari permasalahan tersebut yang menjadi faktor tertinggi yang mempengaruhi pembelian cat pada *home smart* adalah faktor kompetisi *supplier* dalam memasarkan produknya. Faktor kedua yang mempengaruhi pembelian cat *home smart* adalah kualitas cat dan animo masyarakat untuk mengetahui dan membeli produk cat yang dipasarkan dengan berbagai cara yang dilakukan pihak produsen cat tersebut. Faktor harga tidak mempengaruhi pembelian pada *departement penjualan home smart medan*, karena cat dengan harga mahal ternyata masih diminati oleh pelanggan *home smart medan*. Penelitian ini disusun oleh Fitriana Harahap (2015).

Penelitian tentang pendistribusian keterkaitan antara suatu perusahaan dengan atribut tipe motor dan kode warna yang ada pada data distribusi penjualan sepeda motor, seiring terus berlalunya proses transaksi, data akan terus terbentuk hingga bertumpuk-tumpuk sehingga perusahaan kesulitan untuk menganalisis data guna meningkatkan strategi pendistribusian sepeda motor. Dari data tersebut peneliti membuat sebuah aplikasi yang dapat menguji nilai dari atribut untuk mengetahui pola aturan dari pendistribusian yang ada sehingga perusahaan mempunyai bekal pengetahuan dalam mendistribusikan sepeda motor ke

perusahaan target pendistribusiannya. Penelitian ini disusun oleh Jaka Adi Swara (2011).

Penelitian tentang masalah pemahaman tujuan bisnis dan data yang akan digunakan pada proses data mining menggunakan algoritma id3 untuk menentukan tata letak strategis dan tata ruang pasar tradisional dalam rangka menaikkan laba penjualan. Pedagang di pasar tradisional mulai tergeser keberadaannya dikarenakan makin banyaknya pasar moderen yang dibangun di sekitarnya. Animo masyarakat lebih besar untuk memilih berbelanja di pasar moderen dibandingkan berbelanja di pasar tradisional. Hal ini menimbulkan banyak kerugian bagi para pedagang yang berjualan di pasar tradisional. Permasalahan tersebut diselesaikan dengan cara merelokasi letak pasar ke tempat yang lebih strategis. Data yang digunakan untuk langkah pemodelan diambil dari hasil kuesioner data letak strategis pasar dan kuesioner barang. Hal ini dikarenakan tidak ada aturan atau regulasi tertulis pada Dinas Pengelola Pasar yang menjelaskan tentang tata letak strategis pasar. Selain itu data pendukung yang digunakan adalah data spasial (peta) letak dari pasar yang ada di bawah Dinas Pengelolaan Pasar. Target penelitian pada tahun pertama terpenuhi dengan hasil berupa pendefinisian atribut yang akan digunakan pada tahap pemodelan (tahun kedua penelitian). Jumlah atribut yang didefinisikan adalah data letak strategis pasar terdiri dari 23 atribut dan data barang yang dibeli terdiri dari 27 atribut. Data ini bisa digunakan untuk penemuan pola tata letak strategis pasar dan tata ruang pada suatu pasar. Penelitian ini disusun oleh Mira Kania Sabariah, dkk (2012).

**Tabel 2.1 Perbandingan Hasil Penelitian**

Penulis	Objek	Metode	Hasil
Aryansyah dan Wijaya (2014)	Kinerja Motor Servo	<i>Decision tree ID3</i>	Hasil yang diperoleh sama dengan hasil kategori data sebenarnya dengan rincian kecepatan motor sebagai berikut : 0 – 320 RPM = Lambat, 321 – 710 RPM = Cukup Cepat, 711 – 1070 RPM = Cepat
Dodik Setyawan Handoko (2013)	Batik Mahkota Laweyan Surakarta	<i>Decision tree ID3</i>	Terwujudnya suatu analisis data mining untuk menentukan target pasar berdasarkan jenis batik di Batik Mahkota Kampung Batik Laweyan.
Fitriana Harahap (2015)	<i>Departement</i> Penjualan <i>Home Smart</i> Medan	<i>Decision tree ID3</i>	Pengklasifikasian pembelian cat berdasarkan penjualan pada <i>Departement Penjualan Home Smart</i> dalam menentukan pembelian cat dari <i>Supplier</i> .
Jaka Adi Swara (2011)	Sepeda Motor	<i>Decision tree ID3</i>	Aplikasi ini dapat menguji nilai dari atribut untuk mengetahui pola aturan dari pendistribusian yang ada sehingga perusahaan mempunyai bekal pengetahuan dalam mendistribusikan sepeda motor ke perusahaan target pendistribusiannya
Mira Kania Sabariah dkk (2012)	Tata Letak Dan Tata Ruang Pasar Tradisional Kota Bandung	<i>Association rule</i> dan <i>Decision tree</i>	Memenuhi target penelitian tahun pertama dengan hasil berupa pendefinisian atribut yang akan digunakan pada tahap pemodelan (tahun kedua penelitian).
Usulan Penulis (2017)	Toko Sepatu Gita Shoes	<i>Decision tree</i>	Diharapkan menghasilkan nilai penjualan sepatu yang tinggi berdasarkan merk dan jenis sepatu yang dijual dan menemukan informasi/pengetahuan baru yang berguna sebagai pembantu dalam mengambil keputusan.

Pada penelitian yang penulis lakukan, diharapkan dapat memberi rekomendasi kepada penjual berdasarkan nilai penjualan sepatu terbanyak menggunakan merk, jenis sepatu dan kuantitas yang terjual dan menemukan informasi yang berguna sebagai pembantu dalam mengambil keputusan, dalam hal ini yaitu sepatu apa saja yang banyak terjual. Informasi yang dihasilkan adalah

analisis keterkaitan suatu antara merk sepatu yang sama dengan jenis yang berbeda yang tersimpan dalam suatu *database*.

## **1.2. Dasar Teori**

### **1.2.1. Penjualan**

Penjualan merupakan kegiatan yang terintegrasi untuk mengembangkan rencana-rencana penting yang ditujukan kepada usaha pemuas kebutuhan dan keinginan pembeli/konsumen, gunanya adalah untuk memperoleh penjualan yang bisa menghasilkan keuntungan atau laba.

Definisi penjualan ialah suatu kegiatan transaksi yang dilakukan oleh 2 (dua) orang atau lebih dengan memakai alat pembayaran yang sah. Penjualan juga merupakan salah satu sumber dari pendapatan seseorang atau perusahaan yang melangsungkan transaksi jual beli. Dalam suatu perusahaan ataupun perorangan jika penjualannya besar maka pendapatan atau laba yang diperoleh perusahaan atau seseorang tersebut sangat besar pula. (Sora N, 2014).

### **1.2.2. 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. Data mining dapat diterapkan pada berbagai bidang yang mempunyai sejumlah data, tetapi karena wilayah penelitian dengan sejarah yang belum lama, dan belum melewati masa remaja, maka data mining masih diperdebatkan posisi bidang pengetahuan yang memilikinya. Maka Daryl Pregibon menyatakan data mining adalah campuran

dari statistik, kecerdasan buatan, dan riset basis data yang masih berkembang. (Eko Prasetyo, 2014)

### **1.2.3. *Decision tree***

*Decision tree* atau pohon keputusan adalah pohon yang digunakan sebagai prosedur penalaran untuk mendapatkan jawaban dari masalah yang dimasukkan. Pohon yang dibentuk tidak selalu berupa pohon biner. Jika semua fitur dalam dataset menggunakan 2 macam nilai kategorikal maka bentuk pohon yang didapatkan berupa pohon biner. Jika dalam fitur berisi dari 2 macam nilai kategorikal atau menggunakan tipe numerik maka bentuk pohon yang didapatkan biasanya tidak berupa pohon biner.

Bagian awal dari pohon keputusan ini adalah titik akar (*root*), sedangkan setiap cabang dari pohon keputusan merupakan pembagian berdasarkan hasil uji, dan titik akhir (daun) merupakan pembagian kelas yang dihasilkan. Karakteristik dari *decision tree* dibentuk kedalam sejumlah elemen sebagai berikut:

- 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 yang mempunyai tepat satu lengan masukan dan tidak mempunyai lengan keluaran. Node ini menyatakan label kelas (keputusan).

Ada banyak pilihan algoritma untuk *decision tree* seperti Hunt, CART(C&RT), ID3, C4.5, SLIQ, SPRINT, QUEST, DTREG, THAID, CHAID, dan sebagainya. Dalam perkembangannya, ID3 banyak mengalami perbaikan pada versi berikutnya seperti C4.5 atau C5.0. Algoritma ID3 dapat bekerja dengan baik pada semua fitur yang mempunyai tipe data kategorikal (nominal atau ordinal). Induksi *Decision tree* ID3 menurut (Eko Prasetyo, 2014).

1. Dimulai dari node akar
2. Untuk semua fitur, hitung nilai *entropy* untuk semua sampel (data latih) pada node.
3. Pilih fitur dengan informasi *gain* yang maksimal
4. Gunakan fitur tersebut sebagai node sebagai pemecahan menjadi cabang
5. Lakukan secara rekursif pada setiap cabang yang dibuat dengan mengulangi langkah 2 sampai 4 hingga semua data dalam setiap node hanya memberikan satu label kelas. Node yang tidak dapat dipecah lagi merupakan daun yang berisi keputusan.

Secara umum algoritma ID3 untuk membangun pohon keputusan adalah sebagai berikut:

1. Hitung *entropy* dan *gain* dari setiap atribut

Perhitungan pertama yaitu *entropy*. Fitur ini dihitung pada persamaan (2.1)

dengan menggunakan rumus:

$$E(s) = - \sum_{i=1}^m p(\omega_i | s) \log_2 p(\omega_i | s) \quad (i) \quad (2.1)$$

Dengan :

$E = Entropy$  yang digunakan untuk menentukan yang manakah node yang akan menjadi pemecah data latih berikutnya

$p(\omega_i | s)$  = Proporsi kelas  $i$  dalam semua data latih yang diproses di node  $s$ .  
Jumlah semua baris data dengan label kelas  $i$  dibagi jumlah baris semua data.

$m$  = Jumlah nilai berbeda dalam data.

Perhitungan kedua yaitu *gain*. Fitur ini dihitung pada persamaan (2.2)

dengan menggunakan rumus:

$$G(s, j) = E(s) - \sum_{i=1}^n p(v_i | s) \times E(s_i) \quad (i) \quad (2.2)$$

Dimana:

$G = Gain$  untuk memperkirakan pemilihan fitur yang tepat untuk menjadi pemecah pada node tersebut.

$p(v_i | s)$  = Proporsi nilai  $v$  muncul pada kelas dalam node

$E(s_i) = Entropy$  komposisi nilai  $v$  dari kelas ke- $j$  dalam data ke- $i$  node tersebut  
 $n$  = Jumlah nilai berbeda dalam node

Dari contoh kasus mengenai kemungkinan bermain *baseball* yang akan dihitung menggunakan algoritma ID3 dengan mencari node yang tidak akan bisa dipecah lagi yang merupakan simpul daun berisi keputusan (label kelas). Dari contoh kasus tersebut terdapat data latih yang digunakan untuk melakukan prediksi apakah harus bermain *baseball*? dengan menjawab ya atau tidak. Fitur yang digunakan ada 4 yaitu cuaca, suhu, kelembaban, dan angin. Data klasifikasi bermain *baseball* dapat dilihat pada tabel 2.2. Dari data tersebut dilakukan perhitungan dengan mencari nilai *entropy* dan *gain* sampai menemukan simpul daun (label kelas).

**Tabel 2.2 Data klasifikasi bermain *baseball***

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	Panas	Tinggi	Pelan	Tidak
Cerah	Panas	Tinggi	Kencang	Tidak
Mendung	Panas	Tinggi	Pelan	Ya
Hujan	Lembut	Tinggi	Pelan	Ya
Hujan	Dingin	Normal	Pelan	Ya
Hujan	Dingin	Normal	Kencang	Tidak
Mendung	Dingin	Normal	Kencang	Ya
Cerah	Lembut	Tinggi	Pelan	Tidak
Cerah	Dingin	Normal	Pelan	Ya
Hujan	Lembut	Normal	Pelan	Ya
Cerah	Lembut	Normal	Kencang	Ya
Mendung	Lembut	Tinggi	Kencang	Ya
Mendung	Panas	Normal	Pelan	Ya
hujan	Lembut	Tinggi	Kencang	Tidak

(Sumber: Eko Prasetyo, 2014:61)

- a. Hitung *entropy* untuk semua node akar (semua data)

$$\begin{aligned}
 E(\text{semua}) &= - \left[ \frac{(p(\text{ya}|\text{semua}) \times \log_2 p(\text{ya}|\text{semua})) + (p(\text{tidak}|\text{semua}) \times \log_2 p(\text{tidak}|\text{semua}))}{(p(\text{ya}|\text{semua}) + p(\text{tidak}|\text{semua}))} \right] \\
 &= - \left( \left( \frac{9}{14} \times \log_2 \frac{9}{14} \right) + \left( \frac{5}{14} \times \log_2 \frac{5}{14} \right) \right) = 0.9403
 \end{aligned}$$

b. Hitung *entropy* setiap fitur terhadap kelas

$$\begin{aligned} E(\text{semua}_{\text{cerah}}) &= - \left[ \begin{array}{l} (p(\text{ya}|\text{cerah}) \times \log_2 p(\text{ya}|\text{cerah})) + \\ (p(\text{tidak}|\text{cerah}) \times \log_2 p(\text{tidak}|\text{cerah})) \end{array} \right] \\ &= - \left( \left( \left( \frac{2}{5} \times \log_2 \frac{2}{5} \right) + \left( \frac{3}{5} \times \log_2 \frac{3}{5} \right) \right) \right) = 0.9710 \end{aligned}$$

$$\begin{aligned} E(\text{semua}_{\text{mendung}}) &= - \left[ \begin{array}{l} (p(\text{ya}|\text{mendung}) \times \log_2 p(\text{ya}|\text{mendung})) + \\ (p(\text{tidak}|\text{mendung}) \times \log_2 p(\text{tidak}|\text{mendung})) \end{array} \right] \\ &= - \left( \left( \left( \frac{4}{4} \times \log_2 \frac{4}{4} \right) + \left( \frac{0}{4} \times \log_2 \frac{0}{4} \right) \right) \right) = 0 \end{aligned}$$

$$\begin{aligned} E(\text{semua}_{\text{hujan}}) &= - \left[ \begin{array}{l} (p(\text{ya}|\text{hujan}) \times \log_2 p(\text{ya}|\text{hujan})) + \\ (p(\text{tidak}|\text{hujan}) \times \log_2 p(\text{tidak}|\text{hujan})) \end{array} \right] \\ &= - \left( \left( \left( \frac{3}{5} \times \log_2 \frac{3}{5} \right) + \left( \frac{2}{5} \times \log_2 \frac{2}{5} \right) \right) \right) = 0.9710 \end{aligned}$$

c. Hitung informasi *gain* dari setiap atribut

$$\begin{aligned} G(\text{semua}, \text{cuaca}) &= E(\text{semua}) - \sum_{i=1}^n p(v_i|\text{semua}) \times E(s_{\text{cuaca}}) \\ &= E(\text{semua}) - \left[ \begin{array}{l} (p(\text{cerah}|\text{semua}) \times E(\text{semua}_{\text{cerah}})) + \\ (p(\text{mendung}|\text{semua}) \times E(\text{semua}_{\text{mendung}})) \\ (p(\text{hujan}|\text{semua}) \times E(\text{semua}_{\text{hujan}})) \end{array} \right] \\ &= 0.9403 - \left( \left( \left( \frac{5}{14} \times 0.9710 \right) + \left( \frac{4}{14} \times 0 \right) + \left( \frac{5}{14} \times 0.9710 \right) \right) \right) \end{aligned}$$

$$= 0.2467$$

$$G(\text{semua}, \text{suhu}) = 0.0292$$

$$G(\text{semua}, \text{kelembaban}) = 0.1518$$

$$G(\text{semua}, \text{angin}) = 0.0481$$

2. Tabel perhitungan *entropy***Tabel 2.3 Hasil perhitungan *entropy* dan *gain* untuk node akar**

Node			Jumlah	Ya	Tidak	<i>Entropy</i>	<i>Gain</i>
<b>1</b>	<b>Total</b>		<b>14</b>	<b>9</b>	<b>5</b>	<b>0.9403</b>	
	Cuaca	Cerah	5	2	3	0.9710	0.2467
		Mendung	4	4	0	0	
		Hujan	5	3	2	0.9710	
	Suhu	Panas	4	2	2	1.0000	0.0292
		Lembut	6	4	2	0.9183	
		Dingin	4	3	1	0.8113	
	Kelembaban	Tinggi	7	3	4	0.9852	0.1518
		Normal	7	6	1	0.5917	
	Angin	Pelan	8	6	2	0.8113	0.0481
		Kencang	6	3	3	1.0000	

(Sumber: Eko Prasetyo, 2014:62)

*Entropy* tersebut dihitung pada setiap fitur. Dari perhitungan *entropy* dan *gain* ditampilkan pada tabel 2.3 hasil perhitungan *entropy* dan *gain* untuk node akar

## 3. Pemisahan Data

Hasil yang didapat di Tabel 2.3 menunjukkan bahwa *gain* tertinggi ada di fitur cuaca sehingga cuaca dijadikan sebagai node akar. Untuk cabangnya, digunakan 3 nilai di dalam node akar. Selanjutnya data akan terpecah menjadi 3 kelompok, yaitu cerah, mendung, dan hujan. Pemisahan data yang didapatkan dari fitur cuaca disajikan pada tabel 2.4.

**Tabel 2.4 Pemisahan Data oleh fitur cuaca.**

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	Panas	Tinggi	Pelan	Tidak
Cerah	Panas	Tinggi	Kencang	Tidak
Cerah	Lembut	Tinggi	Pelan	Tidak
Cerah	Dingin	Dingin	Pelan	Ya
Cerah	Lembut	Normal	Kencang	Ya
Mendung	Panas	Tinggi	Kencang	Ya
Mendung	Dingin	Normal	Pelan	Ya
Mendung	Lembut	Tinggi	Kencang	Ya
Mendung	Panas	Normal	Pelan	Ya
Hujan	Lembut	Tinggi	Pelan	Ya
Hujan	Dingin	Normal	Pelan	Ya
Hujan	Dingin	Normal	Kencang	Tidak
Hujan	Lembut	Normal	Pelan	Ya
Hujan	Lembut	Tinggi	Kencang	Tidak

(Sumber: Eko Prasetyo, 2014:62)

4. Perhitungan *entropy* dan *gain* berikutnya.

**Tabel 2.5 Hasil perhitungan *entropy* dan *gain* untuk node 2**

Node			Jumlah	Ya	Tidak	Entropy	Gain
2	Total		5	2	3	0.9710	
	Suhu	Panas	2	0	2	0	0.8281
		Lembut	2	1	1	1.0000	
		Dingin	1	1	0	0	
	Kelembaban	Tinggi	3	0	3	0	0.9710
		Normal	2	2	0	0	
	Angin	Pelan	3	1	2	0.9183	0.6313
		Kencang	2	1	1	1.0000	

(sumber: Eko Prasetyo, 2014:63)

Hasil node internal dicabang cerah ditampilkan pada tabel 2.5. Dalam perhitungan *entropy* dan *gain* berikutnya, fitur cuaca tidak dilibatkan.

#### 5. Menghitung note di cabang mendung

Untuk node di cabang mendung (node 3), *entropy* node bernilai nol sehingga dapat dipastikan semua data di node cabang mendung masuk di kelas yang sama dan node yang dituju oleh cabang mendung menjadi daun (node terminal). Untuk hasil node di cabang hujan ditampilkan pada tabel 2.6

**Tabel 2.6 Hasil perhitungan *entropy* dan *gain* untuk node 4**

Node			Jumlah	Ya	Tidak	<i>Entropy</i>	<i>Gain</i>
2	Total		5	2	3	0.9710	
	Suhu	Panas	0	0	2	0	0.6313
		Lembut	3	2	1	0.9183	
		Dingin	2	1	1	1.0000	
	Kelembaban	Tinggi	3	2	1	1.0000	0.9710
		Normal	2	3	2	0.9183	
	Angin	Pelan	3	3	9	0	0.9710
		Kencang	2	0	2	0	

(sumber: Eko Prasetyo, 2014:63)

Hasil yang didapat di Tabel 2.6 menunjukkan bahwa *gain* tertinggi ada di fitur kelembaban sehingga kelembaban dijadikan sebagai node intenal (node 2). Untuk cabangnya, digunakan 2 nilai di dalani node kelembaban. Selanjutnya data di dalamnya akan terpecah lagi menjadi 2 kelompok, yaitu tinggi dan normal. Dua cabang yang didapat untuk kelembaban dan normal mempunyai *entropy* nol, berarti dua node tersebut menjadi daun.

#### 6. Pemisahan data oleh fitur kelembaban dan angin

Hasil yang didapat di Tabel 2.6 menunjukkan bahwa *gain* tertinggi ada di fitur angin, maka angin dijadikan sebagai node intenal (node 4). Untuk cabangnya, digunakan 2 nilai di dalam node angin. Selanjutnya, data di dalamnya

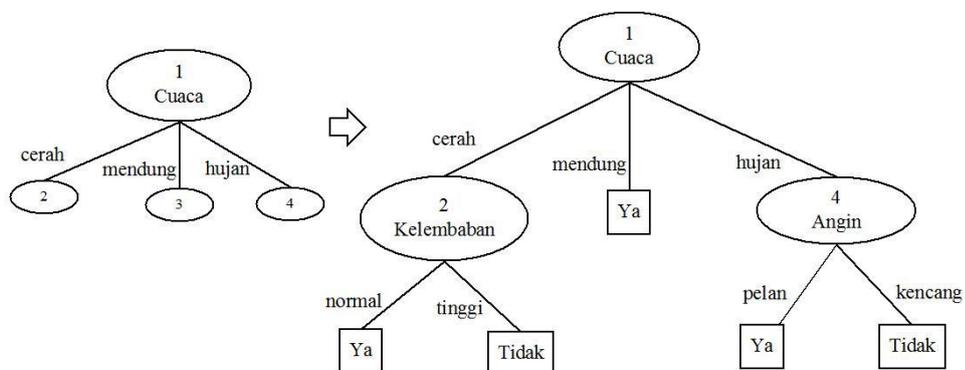
akan terpecah menjadi kelompok, yaitu kencang dan pelan. Dua cabang yang didapat untuk angin kencang dan pelan mempunyai *entropy* nol, berarti dua node tersebut menjadi daun. Hasil pemisahan data yang didapatkan dari fitur kelembaban ditampilkan pada tabel 2.7.

**Tabel 2.7** Pemisahan data oleh fitur kelembaban dan angin

Cuaca	Suhu	Kelembaban	Angin	Bermain
Cerah	Panas	Tinggi	Pelan	Tidak
Cerah	Panas	Tinggi	Kencang	Tidak
Cerah	Lembut	Tinggi	Pelan	Tidak
Cerah	Dingin	Dingin	Pelan	Ya
Cerah	Lembut	Normal	Kencang	Ya
Mendung	Panas	Tinggi	Kencang	Ya
Mendung	Dingin	Normal	Pelan	Ya
Mendung	Lembut	Tinggi	Kencang	Ya
Mendung	Panas	Normal	Pelan	Ya
Hujan	Lembut	Tinggi	Pelan	Ya
Hujan	Dingin	Normal	Pelan	Ya
Hujan	Dingin	Normal	Kencang	Tidak
Hujan	Lembut	Normal	Pelan	Ya
Hujan	Lembut	Tinggi	Kencang	Tidak

(sumber: Eko Prasetyo, 2014:64)

#### 7. Hasil akhir *decision tree*



**Gambar 2.1** Hasil Akhir Perhitungan Data Latih *Decision tree*

Hasil akhir induksi *decision tree* ditampilkan pada Gambar 2.1 hasil akhir perhitungan data latih *decision tree*. Pohon yang dihasilkan juga dapat dinyatakan dalam bentuk aturan *IF THEN* sebagai berikut:

*IF* cuaca = cerah *AND* kelembaban = normal *THEN* *playball* = ya

*IF* cuaca = cerah *AND* kelembaban = tinggi *THEN* *playball* = tidak

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

*IF* cuaca = hujan *AND* angin pelan *THEN* *playball* = ya

*IF* cuaca = hujan *AND* angin = kencang *THEN* *playball* = tidak

#### **2.2.4 Pemrograman Desktop**

Pemrograman Visual Berbasis Desktop adalah pemrograman yang dilakukan dengan memanipulasi elemen-elemen visual yang dilakukan pada sebuah PC tunggal yang pengoperasiannya tidak bergantung pada PC lain dalam jaringan maupun web. *Java* desktop merupakan IDE merupakan kependekan dari

*Integrated Development Environment* atau *Integrated Design Environment* atau *Integrated Debugging Environment*. IDE adalah bagian aplikasi yang menyediakan fasilitas baru bagi programmer untuk melakukan RPL. (Asmaul, 2013)

#### **2.2.5 Java Development Kit (JDK)**

*Java* adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program *java* tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram *Java* banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas *Java*, yang disebut dengan *Java Application*

*Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*). *Java* API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam *Java*, yaitu mempelajari bahasa *Java* dan bagaimana mempergunakan kelas pada *Java* API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada *Java* program *javac* untuk mengkompilasi file kode sumber *Java* menjadi kelas-kelas *bytecode*. File kode sumber mempunyai ekstensi *x.java*. Kompilator *javac* menghasilkan file *bytecode* kelas dengan ekstensi *x.class*. Interpreter merupakan modul utama sistem *Java* yang digunakan aplikasi *Java* dan menjalankan program *byte code Java*.

Beberapa keunggulan *java* yaitu *java* merupakan bahasa yang sederhana. *Java* dirancang agar mudah dipelajari dan digunakan secara efektif. *Java* tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan *Java* secara otomatis seperti dealokasi memori. Bagi pemrogram yang sudah mengenal bahasa C++ akan cepat belajar susunan bahasa *Java* namun harus waspada karena mungkin *Java* mengambil arah (semantiks) yang berbeda dibanding C++. (Didi Indra, 2009)

*Java Development Kit* (JDK) merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi *Java*. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan

compiler dan debugger yang diperlukan untuk mengembangkan applet dan aplikasi. (Oppung Doli, 2012)