BAB II

LANDASAN TEORI

2.1 Nilai Raport Siswa

Fungsi pokok evaluasi hasil belajar siswa secara umum adalah untuk mengukur tingkat kemajuan siswa dalam belajar, untuk menyusun rencana bealajar selanjutnya dan untuk memperbaiki proses pembelajaran (Muhammad Irham dan Novan A.W., 2013 : 217). Laporan evaluasi hasil belajar siswa dituliskan pada sebuah dokumen yaitu rapor. Nilai rapor ditulis berdasarkan hasil belajar siswa dalam satu semester dan ditulis pada akhir semester.

Nilai rapor merupakan hasil kumpulan nilai mata pelajaran dimiliki setiap siswa yang berisi laporan nilai selama satu semester. Rapor diterimakan sebagai tolak ukur dan untuk mengetahui perkembangan terhadap prestasi siswa setelah mengikuti proses pembelajaran.

Melalui rapor wali kelas dapat mengetahui kekuatan dan kelemahan siswa dalam kelas yang diampunya wali kelas dapat menentukan strategi dalam pengelolaan kelas yang menjadi tanggung jawabnya misalnya dengan menata strategis belajar untuk membantu siswa meningkatkan kompetensi siswa atau membantu mengatasi kesulitan belajar siswa yang lemah.

2.2 Try Out

Try Out pada hakikatnya merupakan evaluasi hasil belajar yang dilaksanakan oleh lembaga pendidikan sebelum menghadapi ujian nasional (UN). *Try Out* digunakan untuk menguji kesiapan siswa dalam menghadapi UN. Hasil *Try Out* dapat digunakan siswa untuk mengetahui materi apa yang sudah dikuasai dan yang belum dikuasai. Dari hasil tersebut diharapkan siswa mampu belajar ketertinggalan terhadap materi yang belum dikuasai.

2.3 PPDB

PPDB (Penerimaan Peserta Didik Baru) adalah salah satu kegiatan tahapan yang harus dilewati oleh setiap siswa yang melanjutkan pendidikan yang lebih tinggi. Siswa, orang tua, dan masyarakat perlu mendapat informasi yang jelas dan

lengkap tentang PPDB. Pada tahun pelajaran 2017-2018 PPDB SMPN di kabupaten Gresik menggunakan sistem skoring terpadu (SST) sesuai dengan Lampiran Keputusan Kepala Dinas Pendidikan Kabupaten Gresik tentan Penerimaan Peserta Didik Baru SMP Negeri di kabupaten Gresik.

2.3.1 Kriteria Penilaian

Kriteria penilaian untuk masuk dalam jenjang pendidikan negeri menggunakan sistem skorsing terpadu disebut juga (SST). Sistem *skorsing* terpadu (SST) merupakan sistem yang menggabungkan beberapa nilai komponen atau aspek sesuai dengan pembobotan masing-masing.

Pembobotan setiap aspek atau atribut adalah sebagai berikut :

a. Nilai Ujian Nasiona (3 Mata Pelajaran) = 30%
b. TPA (Tes Potensi Akademik) = 60%
c. Prestasi Akademik = 5%
d. Prestasi Non Akademik = 5%

2.4 Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti tentang kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013 : 8).

Pengertian prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan meruapakan input bagi proses perencanaan dan pengambilan keputusan.

Prediksi bisa beradasarkan metode ilmiah ataupun subjektif belaka. Ambil contoh, prediksi cuaca selalu berdasarkan data dan informasi terbaru yang didasarkan pengamatan termasuk oleh satelit. Begitupun prediksi gempa, gunung meletus ataupun bencana secara umum. Namun, prediksi seperti pertandingan

sepak bola, olahraga, dll umumnya berdasarkan pandangan subjektif dengan sudut pandang sendiri yang memprediksinya.

Secara Eksplisit, pembahasan mengenai teori peramalan kebijakan sangatlah sedikit. Namun, secara implisit, peramalan kebijakan terkait menjadi satu dengan proses analisa kebijakan. Karena didalam menganalisa kebijakan, untuk menformulasikan sebuah rekomendasi kebijakan baru, maka diperlukan adanya peramalan-peramalan atau prediksi mengenai kebijakan yang akan diberlakukan di masa yang akan datang. Namun, satu dari sekian banyak prosedur yang ditawarkan oleh para pakar Dunn, masih memberikan pembahasan tersendiri mengenai peramalan kebijakan. Menurut Dunn, peramalan kebijakan (*Policy Forecasting*) merupakan suatu prosedur untuk membuat informasi *factual* tentang situasi socal masa depan atas dasar informasi yang telah ada tentang masalah kebijakan.

Peramalan (*forecasting*) adalah suatu prosedur untuk membuat informasi factual tentang situasi sosial masa depan atas dasar informasi yang telah ada tentang masalah kebijakan. Ramalan mempunyai tiga bentuk utama: proyeksi, prediksi, dan perkiraan.

1. Suatu proyeksi adalah ramalan yang didasarkan pada ekstrapolasi atas kecenderungan masa lalu maupun masa kini ke masa depan. proyeksi membuat pertanyaan yang tegas berdasarkan argument yang diperoleh dari metode tertentu dan kasus yang pararel.
2. Sebuah prediksi adalah ramalan yang didasarkan pada asumsi teoritik yang tegas. Asumsi ini dapat berbentuk hukum teoretis (misalnya hukum berkurangnya nilai uang), proposisi teoritis (misalnya preposisi bahwa pecahnya masyarakat sipil diakibatkan oleh kesenjangan antara harapan dan kemampuan, atau analogi (misalnya analogi antara pertumbuhan organisasi pemerintah dengan pertumbuhan organisme biologis).
3. Suatu perkiraan (*conjecture*) adalah ramalan yang didasarkan pada penilaian yang *informative* atau penilaian pakar tentang situasi masyarakat masa depan.

Tujuan dari pada diadakannya peramalan kebijakan adalah untuk memperoleh informasi mengenai perubahan dimasa yang akan datang yang akan mempengaruhi terhadap implementasi kebijakan serta konsekuensinya. Oleh karenanya, sebelum rekomendasi diformulasikan perlu adanya peramalan kebijakan sehingga akan diperoleh hasil rekomendasi yang benar-benar akurat untuk diberlakukan pada masa yang akan datang. Didalam memprediksi kebutuhan yang akan datang dengan berpijak pada masa lalu, dibutuhkan seseorang yang memiliki daya sensitifitas tinggi dan mampu membaca kemungkinan-kemungkinan dimasa yang akan datang. Peramalan kebijakan juga diperlukan untuk mengontrol, dalam antrian, berusaha merencanakan dan menetapkan kebijakan sehingga dapat memberikan alternatif-alternatif yang terbaik yang dapat dipilih diantara berbagai kemungkinan yang ditawarkan oleh masa depan. dengan mengacu pada masa depan analisis kebijakan harus mampu manaksir nilai apa yang bisa atau harus membimbing tindakan di masa depan.

2.5 Data Mining

Data mining adalah salah satu cabang ilmu komputer yang banyak menarik perhatian masyarakat. Data mining adalah tumpukan data yang sudah tersimpan selama bertahun- tahun yang dimasukkan ke dalam *database* tetapi tidak digunakan kembali atau disebut “data sampah”. Data mining digunakan untuk menggali dan mendapatkan informasi dari data dengan jumlah besar (Gorunescu, 2011).

Menurut Larose (2005), data mining didefinisikan sebagai sebuah proses untuk menemukan hubungan, pola dan trend baru yang bermakna dengan menyaring data yang sangat besar dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika.

Berdasarkan beberapa pengertian tersebut dapat ditarik kesimpulan bahwa Data Mining adalah suatu teknik menggali informasi berharga yang tersembunyi pada suatu *database* yang sangat besar sehingga ditemukan pola yang menarik yang sebelumnya tidak diketahui.

Kemajuan luar biasa yang terus berlanjut dalam bidang data mining didorong oleh beberapa faktor, antara lain : (Larose, 2005)

1. Pertumbuhan yang cepat dalam kumpulan data
2. Penyimpanan data dalam data warehouse, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang baik.
3. Adanya peningkatan akses data melalui navigasi web dan internet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk data mining (ketersediaan teknologi).
6. Perkembangan yang hebdatat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

2.6 Metode Data Mining

Data mining mempunyai beberapa metode yang dilakukan pengguna untuk meningkatkan proses *mining* supaya lebih efektif. Oleh karena itu, data mining dibagi menjadi beberapa kelompok berdasarkan metodenya, yaitu (Larose, 2005: 11-17).

1. Estimasi

   Variabel target pada proses estimasi lebih condong ke arah numerik daripada ke arah kategorikal (nominal). Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target dibuat berdasarkan pada nilai prediksi.

2. Prediksi

   Prediksi hampir sama dengan klasifikasi dan estimasi, namun pada prediksi data yang digunakan adalah data rentet waktu (*data time series*) dan hasil nilai akhirnya digunakan untuk beberapa waktu mendatang.

3. Klasifikasi

   Pada klasifikasi terdapat variabel target yang berupa nilai kategorikal (nominal). Contoh dari klasifikasi adalah pendapatan masyarakat digolongkan ke dalam tiga kelompok, yaitu pendapatan tinggi,

pendapatan sedang, dan pendapatan rendah. Algoritma klasifikasi yang biasa digunakan adalah *Naive Bayes*, *K- Nearest Neighbor*, dan C4.5.

4. Klastering

   Klastering merupakan pengelompokkan data atau pembentukan data ke dalam jenis yang sama. *Klastering* tidak untuk mengklasifikasi, mengestimasi, atau memprediksi nilai, tetapi membagi seluruh data menjadi kelompok- kelompok yang relatif sama (homogen). Perbedaan algoritma *klastering* dengan algoritma klasifikasi adalah *klastering* tidak memiliki target/class/label, jadi *klastering* termasuk *unsupervised learning*. Contoh algoritma *klastering K-Means* dan *Fuzzy C-Means*.

5. Aturan Asosiasi

   Aturan asosiasi digunakan untuk menemukan atribut yang muncul dalam waktu yang bersamaan dan untuk mencari hubungan antara dua atu lebih data dalam sekumpulan data.

2.7 Decision Tree

Pohon keputusan merupakan salah satu metode klasifikasi yang kuat dan terkenal. Metode pohon keputusan mengubah fakta yang besar menjadi pohon keputusan yang mempresentasikan aturan, aturan tersebut dapat dengan mudah untuk diinterpretasi oleh manusia. Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah variabel input dengan sebuah variabel target (Berry & Linoff, 2004).

Model pohon keputusan terdiri dari sekumpulan aturan untuk membagi sejumlah populasi yang heterogen menjadi lebih kecil (homogen) dengan memperhatikan variabel tujuannya. Variabel tujuannya biasanya dikelompokkan dengan pasti dan model pohon keputusan lebih mengarah pada perhitungan probabilitas dari tiap-tiap *record* terhadap kategori tersebut atau untuk mengklasifikasi *record* dengan mengelompokkannya dalam satu kelas. Sebuah pohon keputusan dapat dibangun dengan menerapkan salah satu algoritma pohon

keputusan untuk memodelkan himpunan data yang belum terklasifikasi kelasnya (Kusrini, 2009).

Pada *decision tree* terdapat 3 jenis *node*, yaitu :

a. Root Node

Merupakan *node* paling atas, pada *node* ini tidak ada *input* dan biasa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.

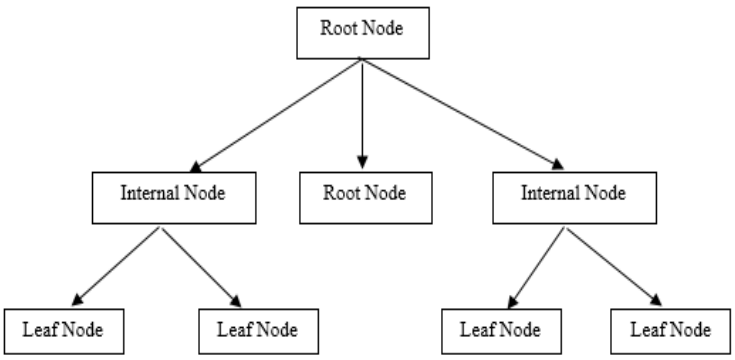
b. Intenal Node

Merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.

c. *Leaf Node* atau *Terminal Node*

d. Merupakan *node* akhir, pada *node* ini hanya terdapat satu *input* dan tidak mempunyai *output*.

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



Gambar 2.1 Model *Decision Tree*

Decision tree merupakan salah satu teknik klasifikasi terhadap objek atau record. Teknik ini terdiri dari kumpulan *decision node*, dan dihubungkan oleh cabang, bergerak ke bawah dari *root node* sampai berakhir di *leaf node* (Yusuf W, 2007).

2.8 Decision Tree C4.5

C4.5 merupakan salah satu metode yang digunakan untuk menginduksi pohon keputusan yang ditemukan oleh J. Ross Quinlan. Algoritma ini diturunkan dari algoritma ID3 yang populer digunakan dalam membuat pohon keputusan. C4.5 merupakan algoritma yang cocok digunakan untuk mengklasifikasi data dalam jumlah besar ke dalam kelas-kelas tertentu berdasarkan pola yang ada. Di dalam data mining dan *machine learning* C4.5 digunakan untuk mempelajari data dalam jumlah besar, membuat model pembelajaran berupa pohon keputusan yang dapat diterapkan untuk memprediksi data yang belum muncul. Beberapa kelebihan dari pohon keputusan yaitu :

1. Hasil analisa berupa diagram pohon yang sangat mudah dimengerti.
2. Mudah untuk dibangun, serta membutuhkan data percobaan yang lebih sedikit dibandingkan algoritma klasifikasi lainnya.
3. Mampu mengolah data nominal dan kontinyu.
4. Model yang dihasilkan dapat dengan mudah dimengerti, beberapa teknik klasifikasi yang berbeda seperti *neural network* menyajikan model dengan informasi logis yang tersirat, sehingga perlu dipelajari.
5. Menggunakan teknik statistik sehingga dapat divalidasikan.
6. Cepat dan handal dalam mengolah dataset besar.
7. Akurasi yang dihasilkan mampu menandingi teknik klasifikasi yang lainnya.

Pada tahap pembelajaran algpritma C4.5 memiliki 2 prinsip kerja :

1. Pembuatan pohon keputusan

   Tujuan dari algoritma penginduksi pohon keputusan adalah mengkontruksi struktur data pohon yang dapat digunakan untuk memprediksi kelas dari sebuah kasus atau record baru yang belum memiliki kelas. C4.5 melakukan konstruksi pohon keputusan dengan metode *divide* and *conquer*. Pada awalnya hanya dibuat node akar dengan menerapkan algoritma *divide* and *conquer*. Algpritma ini memilih pemecahan kasus-kasus yang terbaik dengan menghitung dan membnadingkan gain ratio, kemudian *node-node* yang terbentuk di level

berikutnya, algoritma *divide* and *conquer* akan diterapkan lagi sampai terbentuk daun-daun.

2. Pembuatan aturan- aturan (*rule set*)

   Aturan-aturan yang terbentuk dari pohon keptusan akan membentuk suatu kondisi dalam bentuk *if-then*. Aturan-aturan ini didapat dengan cara menelusuri pohon keputusan dari akar sampai daun. Setiap *node* dan syarat percabangan akan membentuk suatu kondisi atau suatu *if*. Sedangkan untuk nilai-nilai yang terdapat pada daun akan membentuk suatu hasil atau suatu *then*.

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 kaetika fitur ini menajdi node (akar maupun interval) hanya punya dua pilihan cabang.

2. Fitur kategorikal

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

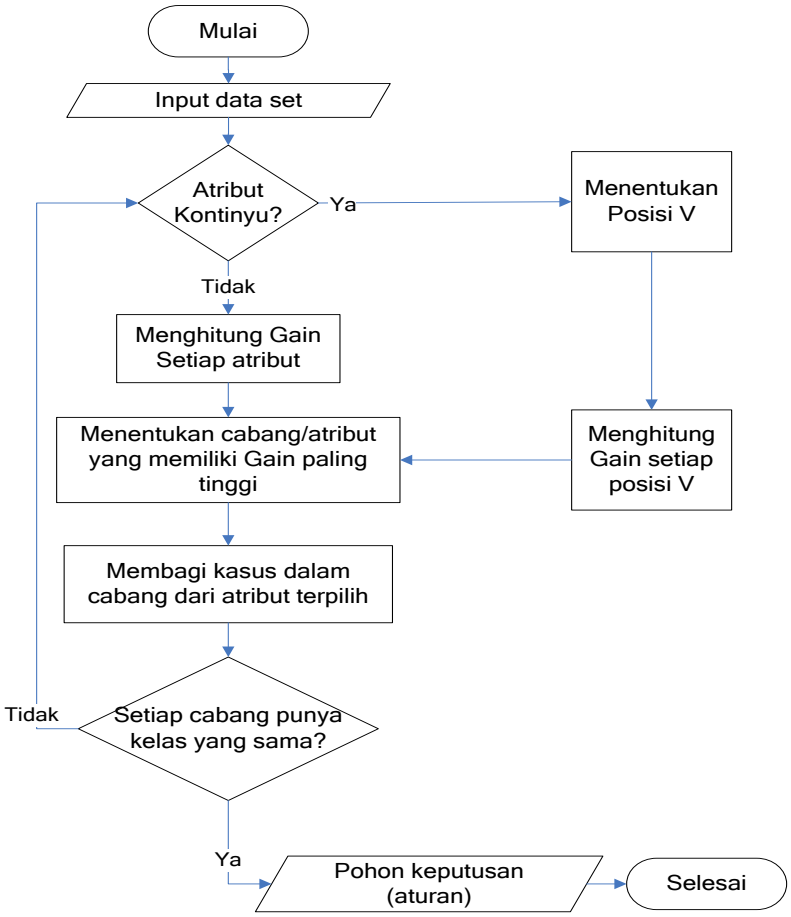
3. Fitur Numerik

   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 *flowcart* 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 *C*4.5 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.

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

Keterangan :

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|Si| : jumlah kasus pada partisi ke-i

|S| :jumlah kasus dalam S

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

Keterangan :

S : himpunan kasus

n : jumlah partisi S

pi : proporsi dari Si terhadap S

Selain *Information Gain* kriteria yang lain untuk memilih atribut sebagai pemecah adalah *Rasio Gain*. Perhitungan *rasio gain* adalah sebagai berikut :

$$\text{GainRasio (S, A)} = \frac{Gain\ (S,A)}{SplitInformation\ (S,A)}$$

Sedangkan untuk perhitungan *split information* menggunakan rumus sebagai berikut :

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

Dimana S_1 sampai S_0 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 sebagai berikut :

$$Akurasi = \frac{Jumlah\ data\ yang\ diprediksi\ secara\ benar}{Jumlah\ prediksi\ yang\ dilakukan}$$

Sedangkan untuk mengukur tingkat kesalahannya digunakan rumus sebagai berikut :

$$Laju\ Error = \frac{Jumlah\ data\ yang\ diprediksi\ secara\ salah}{Jumlah\ Prediksi\ yang\ dilakukan}$$

Sensistivitas akan mengukur proporsi positif asli yang dikenali (diprediksi) secara benar sebagai positif asli. Untuk rumus perhitungannya menggunakan rumus :

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

Keterangan:

TP : Kelas acc yang diprediksi secara benar sebagai kelas acc

FN : Kelas acc yang diprediksi secara salah sebagai kelas tolak

Sedangkan spesifisitas akan mengukur proporsi negatif asli yang dikenali (diprediksi) secara benar sebagai negatid asli. Rumus perhitungannya menggunakan rumus sebagai berikut :

$$Spesifisitas = \frac{TN}{FP + TN}$$

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 memprediksi untuk menentukan main *baseball* atau tidak dengan menjawab Ya atau Tidak. Atribut yang akan 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

Cuaca	Suhu	Kelembaban	Angin	Bermain
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

Cuaca	Suhu	Kelembaban	Angin	Bermain
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 adalh perhitungan *entropy* untuk semua data :

$$Entropy(S) = -\frac{9}{14} * log_2\left(\frac{9}{14}\right) - \frac{5}{14} * log_2\left(\frac{5}{14}\right)$$

$$= 0.9043$$

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 *Gain*nya disajikan pada tabel 2.2.

nilai *Gain* tertinggi didapatkan pada posisi *v* = 70. Maka untuk vitur 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**	
	<=	>	<=	>	<=	>
Ya	4	5	7	2	7	2
Tidak	1	4	3	2	4	1
Gain	**0.0453**		**0.0521**		**0.0005**	

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**	
	<=	>	<=	>	<=	>	<=	>
Ya	2	7	3	6	7	2	7	2
Tidak	1	4	1	4	2	3	3	2
Gain	**0.0005**		**0.0150**		**0.1022**		**0.0251**	

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	
	Cuaca					0,2467

		Cerah	5	2	3	0.9710	
		Mendung	4	4	0	0	
		Hujan	5	3	2	0.9710	
	Suhu						0.0453
		<=70	5	4	1	0.7219	
		>70	9	5	4	0.9911	
	Kelembaban						0.1022
		<=80	9	7	2	0.7642	
		>80	5	3	2	0.9710	
	Angin						0.0481
		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 dijadikan pada tabel 2.5.

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

$$SplitInfo(Semua, cuaca) = \left(-\frac{5}{14} * log_2\left(\frac{5}{14}\right)\right) + \left(-\frac{4}{14} * log_2\left(\frac{4}{14}\right)\right) + \left(-\frac{5}{14} * log_2\left(\frac{5}{14}\right)\right)$$

$$= 1.5774$$

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

$$= 0.16$$

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

Hasil pada tabel 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	
Opsi 1	Cuaca	Cerah Mendung Hujan	5 4 5	1.5774		0.16
Opsi 2	Cuaca	Cerah Mendung dan Hujan	5 9	0.9403		0.26
Opsi 3	Cuaca	Cerah, Mendung Hujan	9 5	0.9403		0.26
Opsi 4	Cuaca	Cerah, Hujan Mendung	10 4	0.8631		0.29

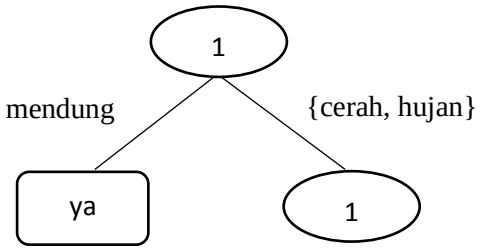
Hasil pemisahan data menurut *node* akar disajikan pada tabel 2.6.

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 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**

	<=	>	<=	>	<=	>
Ya	3	2	5	0	5	0
Tidak	1	4	3	2	4	1
Gain	**0.1245**		**0.2365**		**0.1080**	

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**	
	<=	>	<=	>	<=	>	<=	>
Ya	2	3	2	3	4	3	4	1
Tidak	1	4	1	4	2	1	3	2
Gain	**0.0349**		**0.0349**		**0.1245**		**0.0349**	

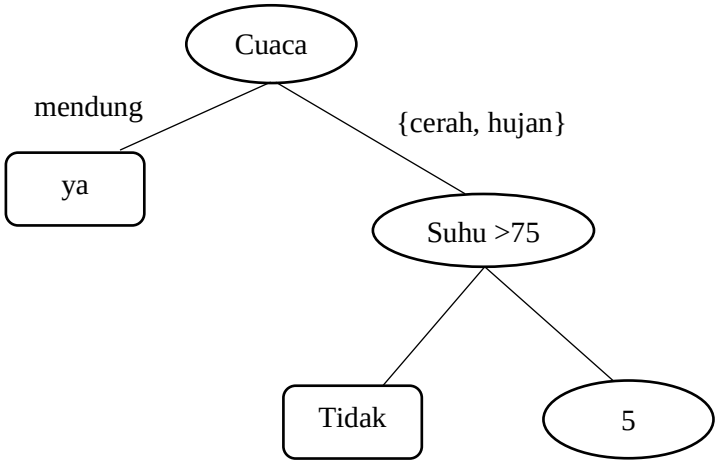
Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.9.

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

Node			**Jumlah**	**Ya**	**Tidak**	***Entropy***	***Gain***
3	Total		10	5	5	1.0000	
	Cuaca	Cerah	5	2	3	0.9710	0.0290
		Hujan	5	3	2	0.9710	
	Suhu	<=75	8	5	3	0.9544	0.2365
		>75	2	0	2	0	

	Kelembaban	<=80	6	4	2	0.9183	0.1245
		>80	4	1	3	0.8113	
	Angin	Pelan	6	4	2	0.9183	0.1245
		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

Cuaca	**Suhu**	**Kelembaban**	**Angin**	**Bermain**
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
Cuaca	**Suhu**	**Kelembaban**	**Angin**	**Bermain**
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**	
	<=	>	<=	>
Ya	3	2	5	0
Tidak	1	2	3	0
Gain	**0.0488**		**0**	

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.

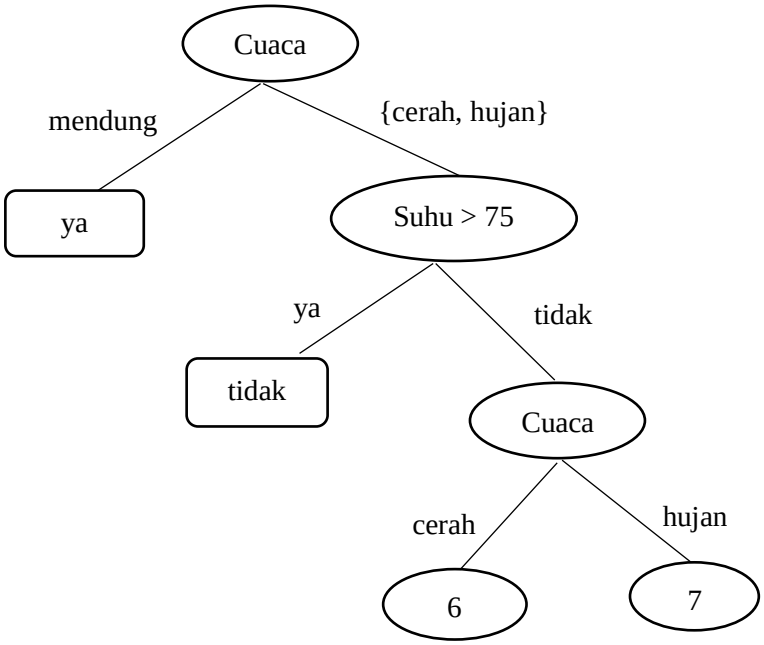
Tabel 2.12 Posisi *v* untuk pemecahan fitur kelembaban di *node* 5

Suhu	**70**		**75**		**80**	
	<=	>	<=	>	<=	>
Ya	2	3	2	3	4	1
Tidak	1	2	1	2	2	1
Gain	**0.0032**		**0.0032**		**0.0157**	

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* dari *gain* untuk *node* 5

Node			**Jumlah**	**Ya**	**Tidak**	***Entropy***	***Gain***
5	Total		8	5	3	0.9544	
	Cuaca						0.2013
		Cerah	3	2	3	0.3900	
		Hujan	5	3	2	0.9710	
	Suhu						0.0488
		<=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	
	Angin						0.1589
		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 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Tabel 2.15 Posisi *v* untuk pemecahan fitur suhu di *node* 6.

Suhu	**70**		**75**	
	<=	>	<=	>
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

Suhu	**70**	
	<=	>

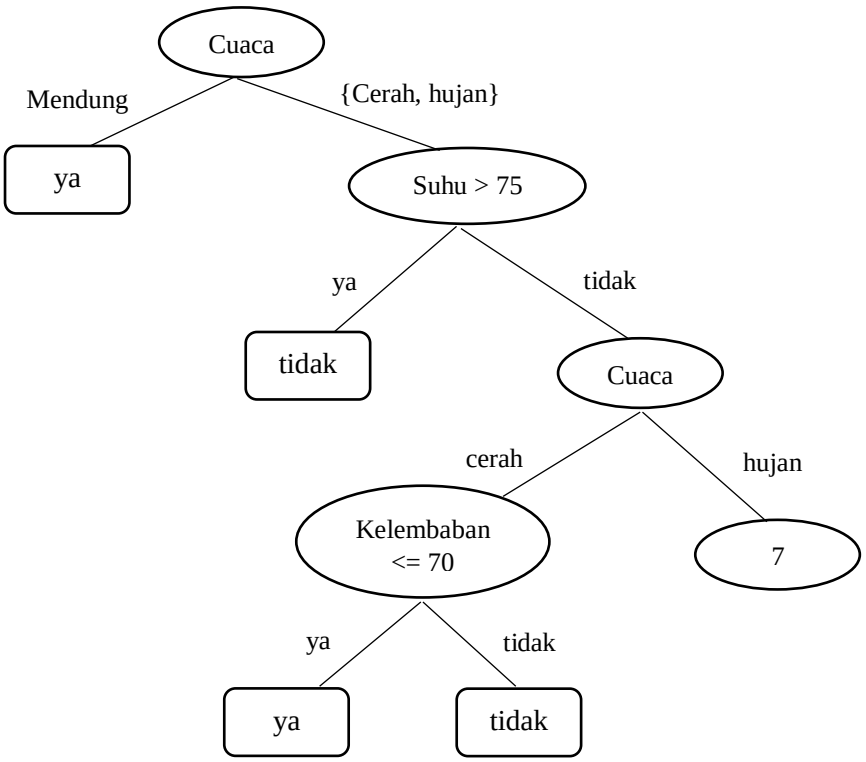
Ya	2	0
Tidak	0	1
Gain	**0.9183**	

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	
	Suhu						0.2516
		<=70	1	1	0	0	
		>70	2	1	1	1.0000	
	Kelembaban						0.9183
		<=70	2	2	0	0	
		>70	1	0	1	0	
	Angin						0.2516
		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	**0.0200**	

Hasil uji coba 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

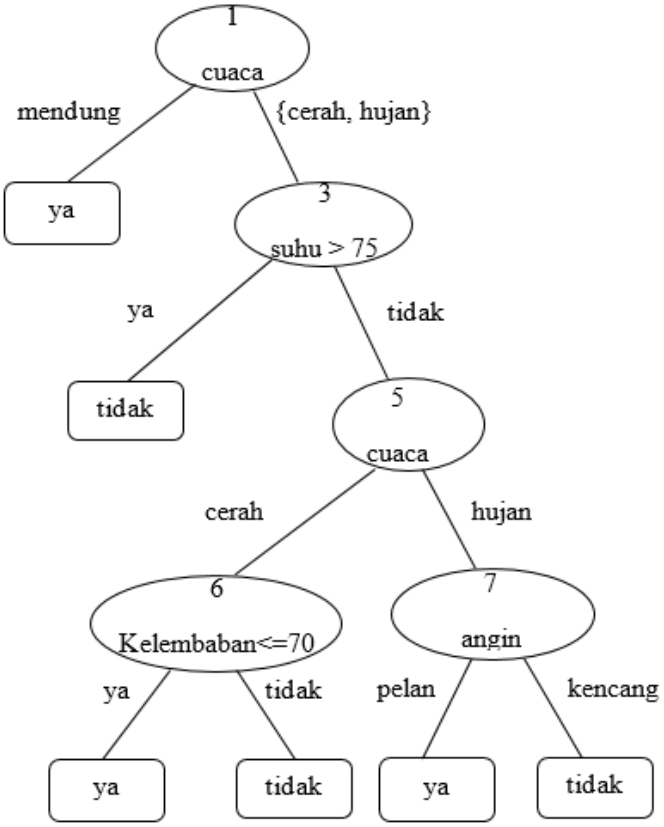
Kelembaban	**70**	
	<=	>
Ya	2	1
Tidak	2	0
Gain	**0.1710**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung nilai *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	
	Suhu						0.0200
		<=70	3	2	1	0.9183	
		>70	2	1	1	1.0000	
	Kelembaban						0.1710
		<=80	4	2	2	1.0000	
		>80	1	1	0	0	
	Angin						0.9710
		Pelan	3	3		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 khusus "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 di proses, 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 sebelumnya dilakukan oleh Nuris Wahidatun Nisak, Mahasiswa Teknik Informatika Universitas Muhammadiyah Gresik yang dilakukan tahun 2016. Penelitian ini berjudul " Prediksi Siswa SD Negeri 2 Sukomulyo Untuk Diterima Di SMP Negeri Menggunakan Metode *Naive Bayes*. Penelitian ini bertujuan untuk menentukan siswa SD 2 Sukomulyo untuk diterima di SMP Negeri atau Tidak. Atribut yang digunakan dalam penelitian ini yaitu nilai rapor siswa, prestasi akademik dan prestasi non akademik. Dengan menggunakan metode Naive Bayes diperoleh hasil presentasi sebesar 96,1% dengan jumlah 120 data latih dan 77 data uji.

Sedangkan penelitian lain yang digunakan sebagai rujukan dalam tugas akhir ini adalah penelitian yang dilakukan oleh lailatul Qomariyah, Mahasiswa Teknik Informatika Universitas Muhammadiyah Gresik yang dilakukan tahun 2016. Penelitian ini berjudul "Klasifikasi Calon Pendonor 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%.