BAB II

LANDASAN TEORI

2.1 Pengertian Prediksi

Prediksi adalah suatu proses memperkiraan 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 kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto,2013: 8).

Pengertian prediksi sama dengan ramalan atau. Menurut kamus besar Bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengampilan keputusan.

Prediksi bisa berdasarkan 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 pertandiangan sepak bola, olahraga, dll umunya berdasarkan pandangan subjektif dengan sudut pandang sendiri yang memprediksinya.

Secara Eksplidit, pembahasan mengenai teori peramalan kebijakan sangatlah sedikit. Namun, secara implisit, peramalan kebijakan menjadi satu dengan proses analisa kebajikan. Karena dalam menganalisa kebajikan, untuk memformulasikannya sebuah rekomendasi kebijakan baru, maka diperlukan adanya peramalan-peramalan atau prediksi mengenai kebijakan yang akan diberlakukan dimasa yang akan datang. Namun, satu dari sekian banyak prosedur yang ditawarkan oleh pakar Dunn, masih memberikan pembahasan

tersendiri mengenai peramalan kebijakan. Menurut Dunn, peramalan kebijakan (policy forecasting) merupakan suatu prosedur untuk membuat informasi factual tentang situasi social masa depan atas dasar informasi yang telah ada tentang masalah kebijakan.

Peramalan (forecasting) adalah suatu prosedur untuk membuat informasi factual tentang situasi social masa depan atas dasar informasi yang telah ada tentang masalah kebijakan. Ramalan mempunyai tiga bentuk utama :

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 rmalan yang didasarkan pada asumsi teoritik yang tegas. Asumsi ini dapat berbentuk hukum teoretis (misalnya hokum berkurangnya nilai uang), proposisi teoritus (misalnya prpoposisi bahwa pecahnya masyarakat sipil diakibatkan oleh kesenjangan antara harapan dan kemampuan), atau analogi (misalnya analogi antara pertumbuhan organisasi permerintah dengan pertumbuhan oraganisme 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 berpijak pada masa lalu, dibutuhkan seseorang yang memiliki daya sensitifitas yang tinggi dan mampu membaca kemungkinan-kemungkinan dimasa yang akan datang. Peramalan kebijakan juga diperlukan untuk mengontrol, dalam artian, berusaha

merencanakan dan menetapkan kebijakan sehingga dapat memberikan alternative-alternatif tindakan yang terbaik yang dapat dipilih diantara berbagai kemungkinan yang ditawarkan olehg masa depan. Masa depan juga terkadang banyak dipengaruhi oleh masa lalu. Dengan mengacu pada masa depan analisis kebijakan harus mampu menaksir nilai apa yang bisa atau harus membimbing tindakan di masa depan.

2.2 Prestasi Siswa

Prestasi seorang siswa itu ditemukan dengan berbagai macam proses seperti dengan belajar. Dengan belajar akan membentuk suatu prestasi belajar yang sangat penting dalam menentukan sebuah prestasi seorang siswa. Dengan belajar siswa akan dapat memperoleh pengetahuan secara luas dan keberhasilan seorang siswa dalam belajar juga ditentukan dengan indikator yang dijadikan sebagai tolak ukur dalam menyatakan bahwa suatu proses belajar mengajar dapat dikatakan berhasil atau tidak.

2.3 Ujian Nasional

Ujian Nasional adalah sistem evaluasi standar pendidikan dasar dan menengah secara nasional dan persamaan mutu tingkat pendidikan antar daerah yang dilakukan oleh Pusat Penelitian Pendidikan. Depdiknas di Indonesiaerdasarkan Undang-undang Republik Indonesia nommor 20 tahun 2003 menyatakan bahwa dalam rangkan pengendalian mutu pendidikan secara nasional dilakukan evaluasi sebagai bentuk akuntabilitas penyelenggara pendidikan kepada pihak-pihak yang berkepentingan.

2.4 *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.

2.5 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 tentang Penerimaan Peserta Didik Baru SMP Negeri di kabupaten Gresik. .

2.5.1 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 Nasional (3 Mata Pelajaran) = 30%
b. TPA (Tes Potensi Akademik) = 60%
c. Prestasi Akademik = 5%
d. Prestasi Non Akademik = 5%

2.6 Data Mining

Secara sederhana, *data mining* merupakan ekstraksi informasi yang tersirat dalam sekumpulan data. Data mining merupakan sebuah proses untuk menggali kumpulan data dan menemukan informasi di dalamnya. (Turban, E., dkk. 2005). Data mining merupakan proses pengekstrakan informasi dari

jumlah kumpulan data yang besar dengan menggunakan algoritma dan tehnik gambar dari statistik, mesin pembelajaran dan sistem manajemen *database*. Penggalian data ini dilakukan pada sekumpulan data yang besar untuk menemukan pola atau hubungan yang ada dalam kumpulan data tersebut (Kusrini dan E.T. Lutfi. 2009). Hasil penemuan yang diperoleh setelah proses penggalian data ini, kemudian dapat digunakan untuk analisis yang lebih lanjut.

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*.

2.7 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 keputusantradisional, fitur (atribut) dari tupel adalah kategorikal atau numerikal. Biasanya definisi ketepan 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 lainyya. 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).

2.8 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 denga kotak hitam, dimana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban sebagai keluaran dari hasil pemikiannya. (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 masalahklasifikasi class atau attribut 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.

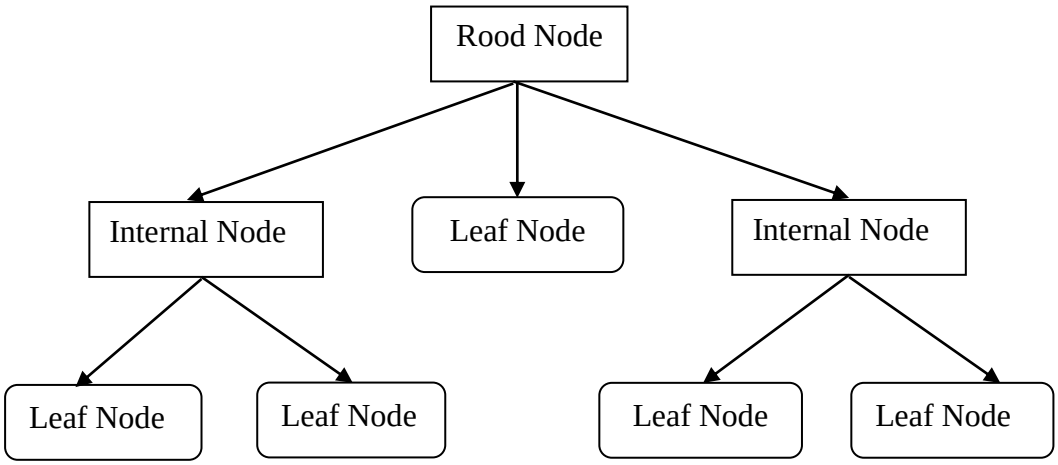
2.9 *Decision Tree*

Decision tree adalah *flow-chart* seperti *struktur tree*, dimana tiap *internal node* menunjukkan sebuah test pada sebuah atribut, tiap cabang menunjukkan hasil dari test, dan *leaf node* menunjukkan *class-class* atau *class distribution*.

Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami. Pada *decision tree* terdapat 3 jenis *node*, yaitu:

a. *Root Node*, merupakan *node* paling atas, pada *node* ini tidak ada *input* dan bisa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.
b. *Internal Node*, merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.
c. *Leaf node* atau *terminal node*, 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*

2.10 *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), sedangan 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*).

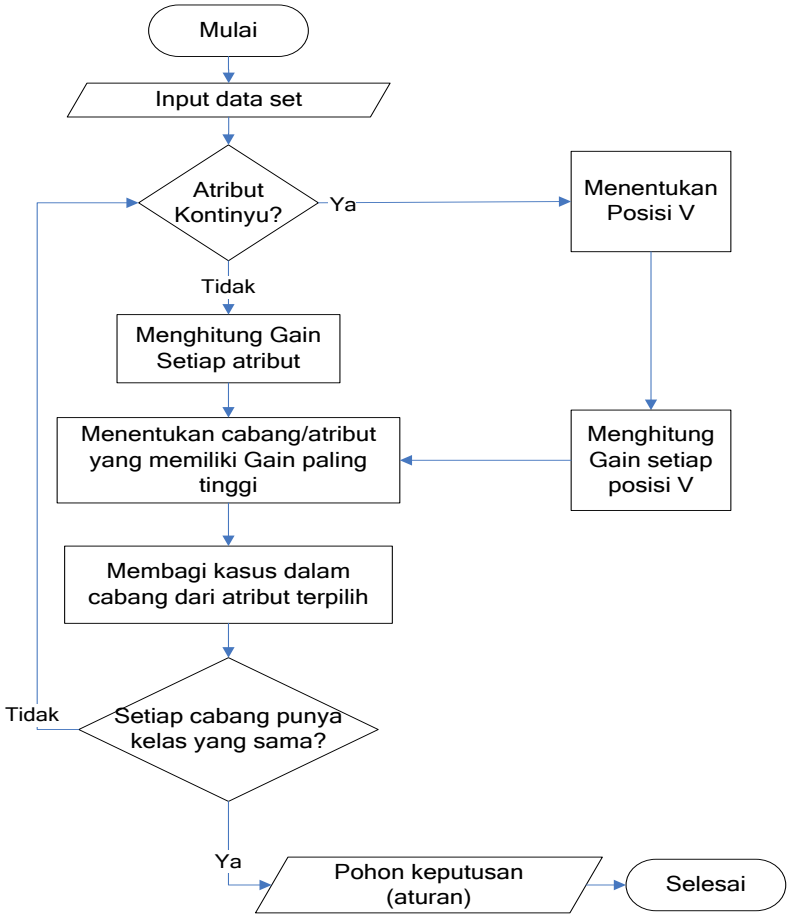
3. Fitur numerik

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 *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.

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

dimana,

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|Si| : Jumlah kasus pada partisi ke i

|S| : Jumlah kasus dalam S

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

dimana,

S : Himpunan kasus

A : Fitur

n : Jumlah partisi S

p_i : Proporsi dari S_i 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.

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

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

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.

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

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

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.

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

Keterangan :

TP : Kelas acc yang diprediksi secara benar sebagai Kelas acc

FN : Kelas acc yang diprediksi secara salah sebagai Kelas tolak

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

Keterangan :

TN : Kelas tolak yang diprediksi secara benar sebagai Kelas tolak

FP : Kelas tolak yang diprediksi secara salah sebagai Kelas acc

2.11 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

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 adalah perhitungan *entropy* untuk semua data:

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

$$= 0.9403$$

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**	
	<=	>	<=	>	<=	>
Ya	4	5	7	2	7	2
Tidak	1	4	3	2	4	1
Gain	**0.0453**		**0.0251**		**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 disajikan pada tabel 2.5.

Hasil peritungan *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 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

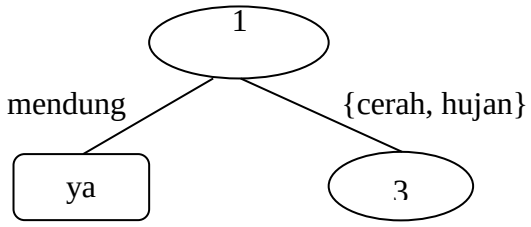
Node			**Jumlah**	**Entropy**	**Gain**	**Rasio Gain**
1	Total		14		0.2467	
Opsi 1	Cuaca	Cerah Mendung Hujan	5 4 5	1.5774		0.16
Opsi 2	Cuaca	Cerah Mendung dan Hujan	5 9	0.9403		0.26
Opsi 3	Cuaca	Cerah, Mendung Hujan	9 5	0.9403		0.26
Opsi 4	Cuaca	Cerah, Hujan Mendung	10 4	0.8631		0.29

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

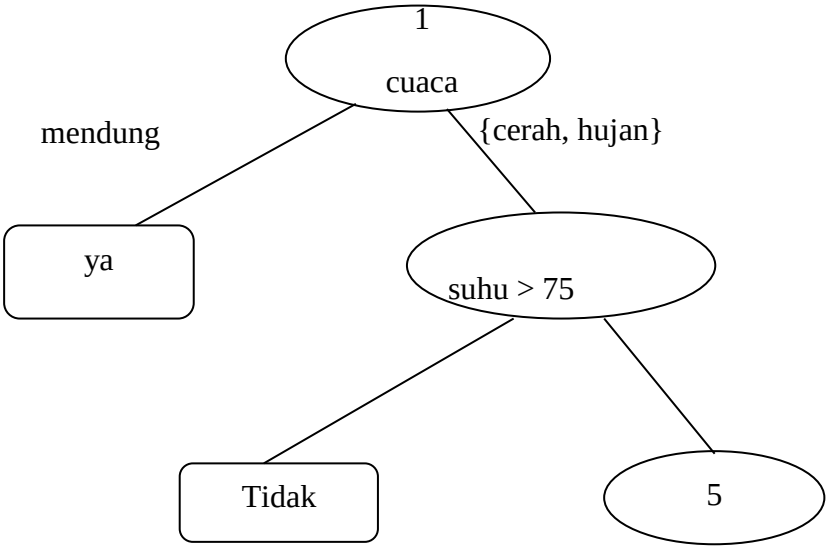
Hasil uji coba pada fitur Kelembaban dengan menghitung nilai *Gain* yang disajikan pada tabel 2.8. Nilai *Gain* tertinggi didapatkan pada posisi *v* = 80.

Tabel 2.8 Perhitungan *Entroppy* dan *Rasio Gain* untuk node 3

Node			**Jumlah**	**Ya**	**Tidak**	**Entropy**	**Gain**
3	Total		10	5	5	1.0000	
	Cuaca						0.0290
		Cerah	5	2	3	0.9710	
		Hujan	5	3	2	0.9710	
	Suhu						0.2365
		<=75	8	5	3	0.9544	
		>75	2	0	2	0	
	Kelembaban						0.1245
		<=80	6	4	2	0.9183	
		>80	4	1	3	0.8113	
	Angin						0.1245
		Pelan	6	4	2	0.9183	
		Kencang	4	1	3	0.8113	

Maka untuk fitur Kelembaban dilakukan diskretisasi pada v = 80 ketika menghitung *Entropy* dan *Gain* pada semua fitur.

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.8. 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 ditunjukan pada gambar 2.4. Pemisahan datanya ditunjukkan pada tabel 2.9.



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

Tabel 2.9 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 2.10. 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 2.10 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 2.11. 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 2.11 Posisi *v* untuk pemecahan fitur Kelembaban di *node* 5

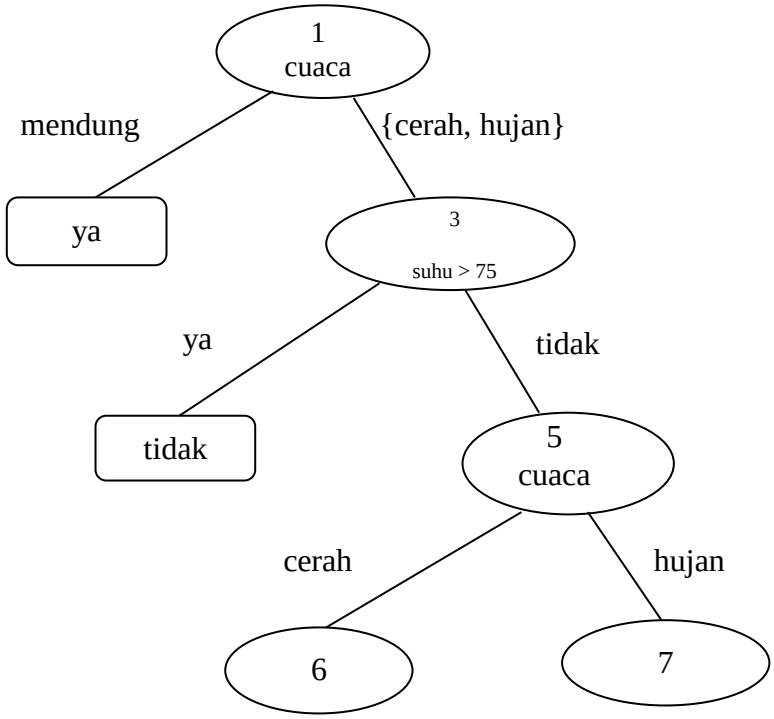
Kelembaban	**70**		**75**		**80**	
	<=	>	<=	>	<=	>
Ya	2	3	2	3	4	1
Tidak	1	2	1	2	2	1
Gain	**0.0032**		**0.0032**		**0.0157**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.12.

Tabel 2.12 Hasil perhitungan *entropy* dan *gain* untuk *node* 5

Node			**Jumlah**	**Ya**	**Tidak**	**Entropy**	**Gain**
5	Total		8	5	3	0.9544	
	Cuaca	Cerah	3	2	3	0.3900	0.2013

		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 2.13 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 2.14. 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 2.14 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 2.15. 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 2.15 Posisi *v* untuk pemecahan fitur Kelembaban di *node* 6

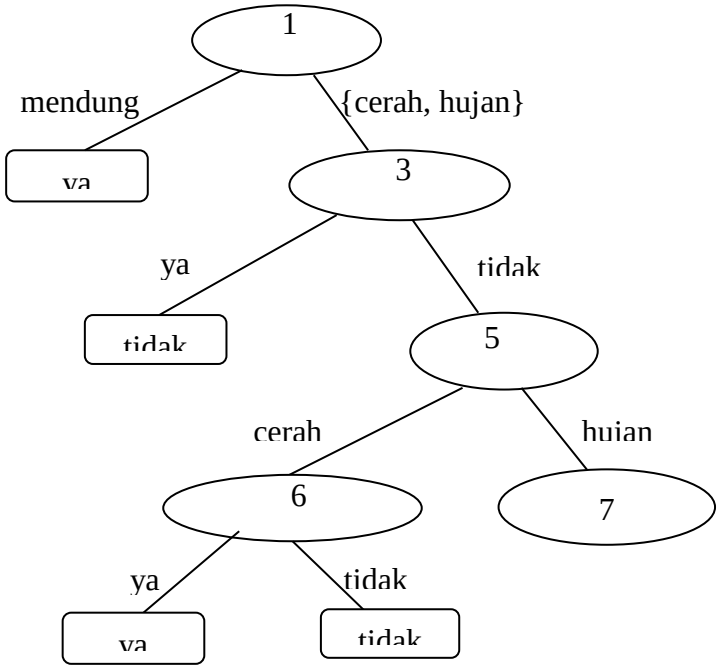
Kelembaban	**70**	
	<=	>
Ya	2	0
Tidak	0	1
Gain	**0.9183**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.16.

Tabel 2.16 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 2.16 menunjukkan bahwa *gain* tertinggi ada di fitur Kelembaban, berarti fitur Kelembaban dijadikan syarat kondisi di *node* 6, seperti ditunjukan pada gambar 2.6. Pemisahan datanya ditunjukkan pada tabel 2.17.



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

Tabel 2.17 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 2.18. 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 2.18 Posisi *v* untuk pemecahan fitur Suhu di *node* 7

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

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 2.19 Posisi *v* untuk pemecahan fitur Kelembaban di *node* 7

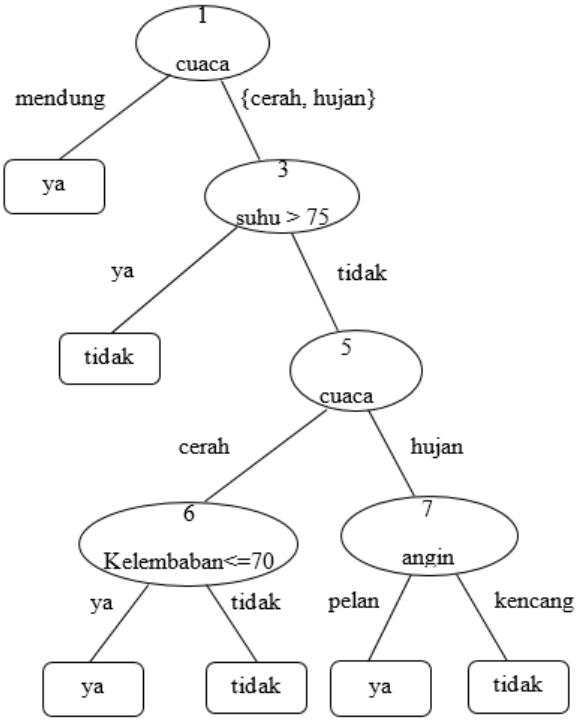
Kelembaban	**70**	
	<=	>
Ya	2	1
Tidak	2	0
Gain	**0.1710**	

Selanjutnya dihitung *entropy* untuk setiap nilai fitur terhadap kelas, kemudian dihitung *gain* untuk setiap fitur. Hasilnya disajikan pada tabel 2.20.

Tabel 2.20 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 2.20 menunjukkan bahwa *gain* tertinggi ada di fitur Angin, berarti fitur Angin dijadikan syarat kondisi di *node* 7, seperti ditunjukan pada gambar 2.7. Pemisahan datanya ditunjukkan pada tabel 2.21.



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

Tabel 2.21 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 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 disajikan pada gambar 2.7.

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

2.12 Penelitian Sebelumnya

Penelitian sebelumnya yang menggunakan metode *decision tree C4.5* adalah penelitian yang berjudul "Penerapan Algoritma Decision Tree C4.5 Untuk Diagnosa Penyakit Stroke Dengan Klasifikasi Data Mining Pada

Rumah Sakit Santa Maria Pemalang". Penelitian ini disusun oleh Sigit Abdullah yang berasal dari Program Studi Teknik Informatika Universitas Dian Nuswantoro Semarang. Dalam penelitian ini atribut yang digunakan adalah Jenis Kelamin, Umur, Hipertensi, serta Diabetes dengan variabel target klasifikasi keputusan berupa Stroke atau Non Stroke. Data yang digunakan dalam penelitian ini yaitu sebanyak 156 data pasien yang diperoleh dari rumah sakit, data tersebut dibagi menjadi dua bagian yaitu data training sebanyak 130 data dan sisanya pada data testing yang berjumlah 26 data. Dari metode klasifikasi data mining dengan algoritma C4.5 diperoleh akurasi sebesar 82,31% untuk data training sedangkan akurasi pada data testing sebesar 76,92%. Perhitungan tingkat akurasi keduanya menggunakan confusion matrix.

Penelitian selanjutanya 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 Kabupatem 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 sebayak 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 peecobaan tersebut diketahui nilai akurasi tertinggi terjadi pada percobaan dengan 48 data latih dan 12 data uji dengan nilai akurasi mencapai 91,67%.