

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep Beasiswa**

##### **2.1.1 Pengertian Beasiswa**

Beasiswa dapat diartikan sebagai bentuk penghargaan yang diberikan kepada individu agar dapat melanjutkan pendidikan ke jenjang yang lebih tinggi, penghargaan itu dapat berupa akses tertentu pada suatu instansi atau penghargaan berupa bantuan keuangan. Murniasih (2009).

Menurut Murniarsih untuk Beasiswa Peningkatan Prestasi Akademik (PPA) adalah beasiswa yang diberikan untuk peningkatan atau pemerataan dan kesempatan belajar bagi mahasiswa yang mengalami kesulitan membayar biaya pendidikannya sebagai akibat krisis ekonomi, terutama bagi mahasiswa yang berprestasi akademik. Adapun tujuan beasiswa peningkatan prestasi akademik (PPA) secara umum yaitu:

1. Meningkatkan Pemerataan dan kesempatan belajar bagi mahasiswa yang mengalami kesulitan membayar biaya pendidikan.
2. Mendorong dan mempertahankan semangat belajar mahasiswa agar mereka dapat menyelesaikan studi atau pendidikan tepat waktu.
3. Mendorong untuk meningkatkan prestasi akademik sehingga memacu peningkatan kualitas pendidikan.

Proses Pemberian Beasiswa di Universitas Muhammadiyah Gresik mengacu pada Pedoman Pimpinan Pusat Muhammadiyah Nomor 02/PED/I.O/B/2012 tentang Perguruan Tinggi Muhammadiyah dan mengingat pada peraturan Menteri nomor 30 tahun 2010 tentang Pemberian

Bantuan Biaya Pendidikan kepada peserta didik yang orang tua atau walinya tidak mampu membiayai pendidikan.

Syarat yang harus di penuhi oleh mahasiswa Universitas Mhammadiyah Gresik untuk bisa mendapatkan beasiswa BPP-PPA bagi program studi S1 dibawah kordinasi Kopertis Wilayah VII JATIM adalah:

1. Mahasiswa aktif Universitas Muhammadiyah Gresik terdaftar sebagai mahasiswa semester IV dan VI yang dibuktikan dengan fotocopy KTM (Kartu Tanda Mahasiswa)
2. Surat keterangan penghasilan orang tua atau wali dari instansi tempat bekerja (Pegawai Negeri atau Swasta oleh bagian keuangan di instansi yang bersangkutan) untuk wiraswasta surat pernyataan penghasilan orang tua atau wali bermaterai 6000 yang disahkan oleh kepala desa atau lurah.
3. Memiliki IPK minimal 3.25 yang dibuktikan dengan fotocopy KRS atau Transkrip Nilai Sementara.
4. Forocopy atau scan piagam Prestasi yang diselenggarakan oleh KEMDIKBUD atau Instansi lain di tingkat Regional, Nasional, dan Internasional.
5. Fotocopy Kartu Keluarga (KK).
6. Fotocopy rekening buku tabungan pada bank yang sudah ditentukan.
7. Surat pernyataan tidak menerima beasiswa dari sumber APBD dan APBN bermaterai 6000 yang diketahui oleh Wakil Rektor III (disediakan oleh bagian kemahasiswaan).
8. Surat rekomendasi dari program studi masing masing.

## 2.2 Konsep Sistem Pendukung Keputusan

### 2.2.1 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau Decision Support System (DSS) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Turban, 2001).

SPK merupakan implementasi teori teori pengambilan keputusan yang telah diperkenalkan oleh ilmu-ilmu seperti operation research dan menegement science, hanya bedanya adalah bahwa jika dahulu untuk mencari penyelesaian masalah yang dihadapi harus dilakukan perhitungan literasi secara manual (biasanya untuk mencari nilai nilai minimum, maksimum, atau optimum), saat ini computer PC telah menawarkan kemampuannya untuk menyelesaikan persoalan yang sama dalam waktu relatif singkat.

Sprague dan Watson mendefinisikan Sistem Pendukung Keputusan (SPK) sebagai sistem yang memiliki lima karakteristik utama yaitu (Sprague et.al, 1993):

1. Sistem yang berbasis komputer.
2. Dipergunakan untuk membantu para pengambil keputusan .
3. Untuk memecahkan masalah-masalah rumit yang mustahil dilakukan dengan kalkulasi manual.
4. Melalui cara simulasi yang interaktif.
5. Dimana data dan model analisis sebagai komponen utama.

### 2.2.2 Konsep Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan (SPK) / Decision Support Sistem (DSS) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah Management Decision Sistem. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur.

Istilah SPK mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan. Untuk memberikan pengertian yang lebih mendalam, akan diuraikan beberapa definisi mengenai SPK yang dikembangkan oleh beberapa ahli, diantaranya oleh Man dan Watson yang memberikan definisi sebagai berikut, SPK merupakan suatu sistem yang interaktif, yang membantu pengambil keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah yang sifatnya semi terstruktur maupun yang tidak terstruktur.

Tahapan Dalam Pengambilan Keputusan :

1. Tahap Pemahaman.
2. Tahap Perancangan.
3. Tahap Pemilihan.
4. Tahap Penerapan.

Sistem pendukung Keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur dengan menambahkan kebijaksanaan manusia dan informasi komputerisasi.

Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari / interogasi informasi.

Dengan berbagai karakter khusus diatas, SPK dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah:

1. SPK memperluas kemampuan pengambil keputusan dalam memproses data atau informasi bagi pemakainya.
2. SPK membantu pengambilan keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.

Walaupun suatu SPK, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun ia dapat menjadi stimulant bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan.

Disamping berbagai keuntungan dan manfaat seperti dikemukakan diatas SPK juga memiliki beberapa keterbatasan, diantaranya adalah :

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
2. Kemampuan suatu SPK terbatas pada perbendaharaan pengetahuan yang dimilikinya (pengetahuan dasar dan model dasar).

3. Proses-proses yang dapat dilakukan SPK biasanya juga tergantung pada perangkat lunak yang digunakan.
4. SPK tidak memiliki kemampuan intuisi seperti yang dimiliki manusia.

Sistem ini dirancang hanyalah untuk membantu pengambil keputusan dalam melaksanakan tugasnya, jadi secara dapat dikatakan bahwa SPK dapat memberikan manfaat bagi pengambil keputusan dalam meningkatkan efektifitas dan efisiensi kerja terutama dalam proses pengambil keputusan.

Tahapan SPK:

1. Definisi masalah.
2. Pengumpulan data atau elemen informasi yang relevan.
3. Pengolahan data menjadi informasi baik dalam bentuk laporan grafik maupun tulisan.
4. Menentukan alternatif-alternatif solusi (bias dalam Persentasi)

### **2.3 Decision Tree (Pohon Keputusan)**

Pohon keputusan merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Selain itu dapat diekspresikan dalam bentuk bahasa basis data seperti *Structure Query Language* untuk mencari *record* pada kategori tertentu (Kusrini dan Emha, 2009). Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan variabel target.

Sebuah pohon keputusan adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan,

dengan masing-masing rangkaian pembagian, anggota himpunan hasil menjadi mirip satu dengan yang lain.

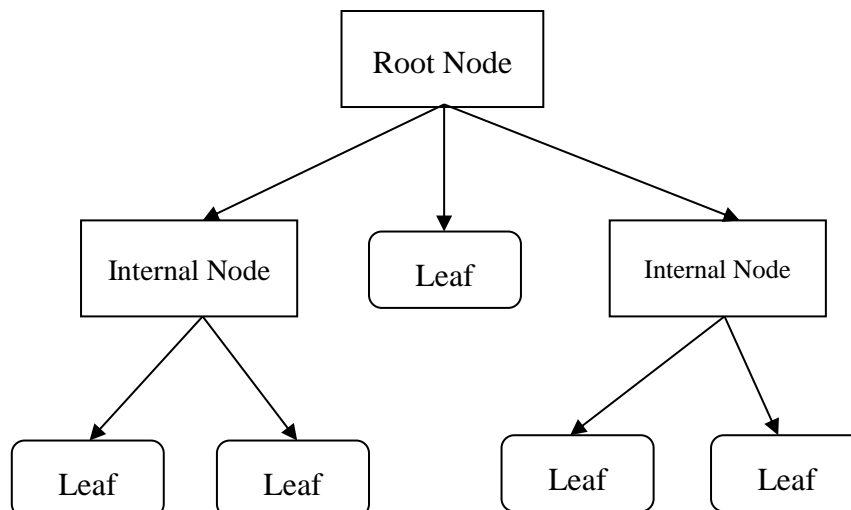
### 2.3.1 Model *Decision Tree*

*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:

1. *Root Node*, merupakan *node* paling atas, pada *node* ini tidak ada *input* dan bisa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.
2. *Internal Node*, merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.
3. *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* ( Pramudiono,2008 )

### 2.3.2 Algoritma C4.5

Algoritma C4.5 diperkenalkan oleh Quinlan (1996) sebagai versi perbaikan dari ID3. Dalam ID3, induksi decision tree hanya bisa dilakukan pada fitur bertipe kategorikal (nominal atau ordinal), sedangkan tipe numerik (interval atau rasio) tidak dapat digunakan. Hal terpenting dalam induksi decision tree adalah bagaimana menyatakan syarat pengujian pada node. Ada 3 kelompok penting dalam syarat pengujian node (Eko Prasetyo, 2014).:

1. Fitur biner

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

2. Fitur kategorikal

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

3. Fitur numerik

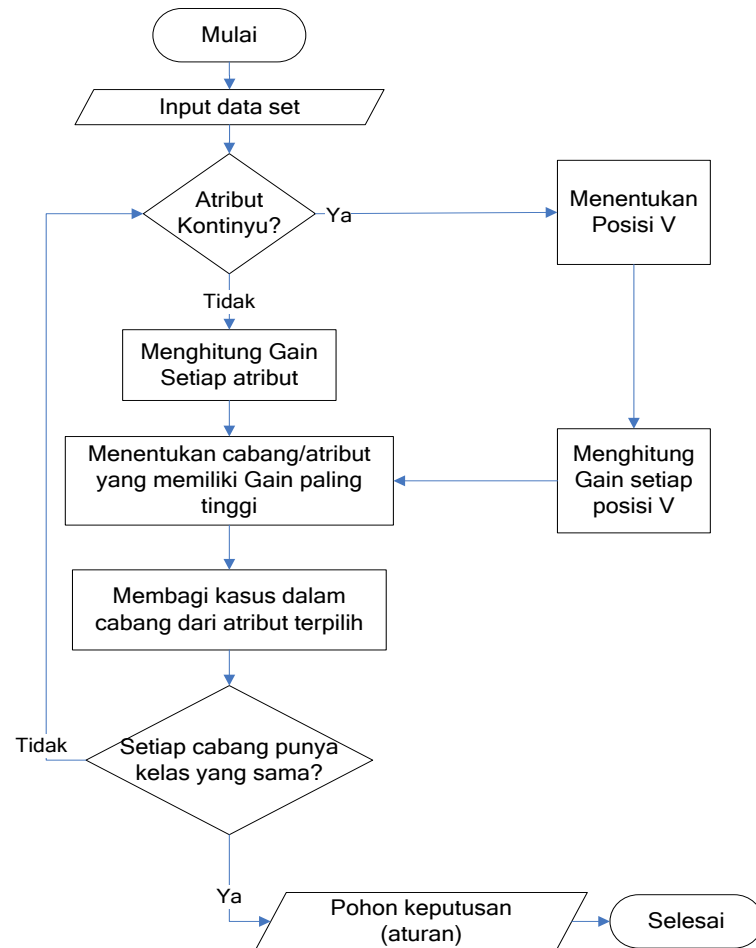
Untuk fitur bertipe numerik, Syarat pengujian dalam node (akar maupun internal) dinyatakan dengan pengujian perbandingan ( $A \leq V$ ) atau ( $A > V$ ) dengan hasil biner.

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

1. Pilih atribut sebagai akar.
2. Buat cabang untuk tiap-tiap nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.



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



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

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

homogen distribusi kelas pada data tersebut. Perhitungan *information gain* menggunakan rumus 2.1, sedangkan *entropy* menggunakan rumus 2.2.

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

dimana,

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|S<sub>i</sub>| : Jumlah kasus pada partisi ke i

|S| : Jumlah kasus dalam S

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

dimana,

S : Himpunan kasus

A : Fitur

n : Jumlah partisi S

p<sub>i</sub> : Proporsi dari S<sub>i</sub> terhadap S

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

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

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

dimana S<sub>1</sub> sampai S<sub>c</sub> 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 : Keterangan terima yang diprediksi secara benar sebagai Keterangan terima

FN : Keterangan terima yang diprediksi secara salah sebagai Keterangan tidak terima

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

Keterangan :

TN: Keterangan tidak terima yang diprediksi secara benar sebagai Keterangan tidak terima

FP : Keterangan tidak terima yang diprediksi secara salah sebagai Keterangan terima

## 2.4 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 menentukan *Play* atau *Don't Play* dengan beberapa atribut yaitu atribut *outlook*, *temperature*, *humidity*, dan *windy*. Dimana atribut

*temperature* dan *humidity* bertipe kontinu sedangkan *outlook* dan *windy* bertipe kategorikal. Sedangkan kolom *Class* adalah kelas tujuannya atau label kelas-nya.

**Tabel 2.1** Contoh data set

<b>Outlook</b>	<b>Temperature</b>	<b>Humidity</b>	<b>Windy</b>	<b>Class</b>
sunny	75	70	TRUE	Play
sunny	80	90	TRUE	Don't Play
sunny	85	85	FALSE	Don't Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
overcast	72	90	TRUE	Play
overcast	83	78	FALSE	Play
overcast	64	65	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play
rain	65	70	TRUE	Don't Play
rain	75	80	FALSE	Play
rain	68	80	FALSE	Play
Rain	70	96	FALSE	Play

Pada contoh ini rumus yang digunakan untuk memilih atribut sebagai *node* adalah rumus *information gain*. Proses pertama adalah menghitung *entropy* untuk semua data.

Jumlah class play = 9

Jumlah class don't play = 5

Berikut adalah perhitungan *entropy* untuk semua data:

$$Entropy(S) = -\frac{9}{14} * \log_2\left(\frac{9}{14}\right) - \frac{5}{14} * \log_2\left(\frac{5}{14}\right) = 0.940$$

Selanjutnya menghitung *gain* untuk setiap atribut. Berikut adalah contoh perhitungan gain untuk atribut *outlook*:

**Tabel 2.2** Distribusi jumlah atribut *outlook*

Nilai Outlook	$\Sigma$ Play	$\Sigma$ Don't Play	Total
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Berdasarkan tabel 2.2, maka nilai *information gain* untuk atribut *outlook* adalah sebagai berikut:

$$\begin{aligned}
 Gain(outlook) &= 0.940 - \left( \frac{5}{14} * \left( -\frac{2}{5} * \log_2 \left( \frac{2}{5} \right) - \frac{3}{5} * \log_2 \left( \frac{3}{5} \right) \right) \right. \\
 &\quad + \frac{4}{14} * \left( -\frac{4}{4} * \log_2 \left( \frac{4}{4} \right) - \frac{0}{4} * \log_2 \left( \frac{0}{4} \right) \right) \\
 &\quad \left. + \frac{5}{14} * \left( -\frac{3}{5} * \log_2 \left( \frac{3}{5} \right) - \frac{2}{5} * \log_2 \left( \frac{2}{5} \right) \right) \right) \\
 &= 0.940 - 0.694 \\
 &= 0.246
 \end{aligned}$$

Untuk perhitungan atribut yang bertipe kontinyu, harus menentukan *posisi V* terbaik yang dinyatakan dalam perbandingan ( $A \leq V$ ) atau ( $A > V$ ). Berikut akan dijelaskan contoh perhitungan dari atribut *temperature*.

Misal posisi *V* yang akan digunakan pada atribut *temperature* adalah 65,70,75,dan 80, kemudian dihitung nilai *information gain*-nya.

Contoh perhitungan *temperature* posisi  $v=65$ :

$$\begin{aligned}
 Gain(temp) &= 0.940 - \left( \frac{2}{14} * \left( -\frac{1}{2} * \log_2 \left( \frac{1}{2} \right) - \frac{1}{2} * \log_2 \left( \frac{1}{2} \right) \right) \right. \\
 &\quad \left. + \frac{12}{14} * \left( -\frac{8}{12} * \log_2 \left( \frac{8}{12} \right) - \frac{4}{12} * \log_2 \left( \frac{4}{12} \right) \right) \right) \\
 &= 0.940 - 0.930 \\
 &= 0.010
 \end{aligned}$$

Berikut hasil perhitungan atribut numerik untuk setiap posisi yang telah ditentukan :

**Tabel 2.3** Hasil perhitungan posisi V untuk atribut *temperature*

Temperature	65		70		75		80	
	≤	>	≤	>	≤	>	≤	>
<b>Play</b>	1	8	4	5	7	2	7	2
<b>Don't Play</b>	1	4	1	4	3	2	4	1
<b>Jumlah</b>	<b>2</b>	<b>12</b>	<b>5</b>	<b>9</b>	<b>10</b>	<b>4</b>	<b>11</b>	<b>3</b>
<b>Entropy</b>	<b>1.000</b>	<b>0.918</b>	<b>0.722</b>	<b>0.991</b>	<b>0.881</b>	<b>1.000</b>	<b>0.946</b>	<b>0.918</b>
<b>Gain</b>	0.010		0.045		0.025		0.0005	

Berdasarkan tabel 2.3, nilai gain tertinggi adalah 70, maka nilai *information gain* pada atribut *temperature* adalah 0.045. Hasil perhitungan pada setiap atribut disajikan pada tabel 2.4.

**Tabel 2.4** Hasil perhitungan *Information gain* untuk setiap atribut

		Jumlah	Play	Don't Play	Entropy	Gain
<b>Total</b>		14	9	5	0.940	
<b>Outlook</b>	<b>Sunny</b>	5	2	3	0.971	0.247
	<b>Overcast</b>	4	4	0	0.000	
	<b>Rain</b>	5	3	2	0.971	
<b>Temperature</b>	≤ 70	5	4	1	0.722	0.045
	> 70	9	5	4	0.991	
<b>Humidity</b>	≤ 80	9	7	2	0.764	0.102
	> 80	5	2	3	0.971	
<b>Windy</b>	<b>TRUE</b>	6	3	3	1.000	0.048
	<b>FALSE</b>	8	6	2	0.811	

Berdasarkan tabel 2.4 menunjukkan bahwa atribut *outlook* memiliki nilai gain tertinggi, maka atribut *outlook* akan menjadi *node*. Karena atribut *outlook* memiliki tiga nilai atribut atau lebih dari dua, maka dilakukan perhitungan rasio

gain untuk memilih pilihan percabangan terbaik. Berikut adalah contoh perhitungan rasio gain untuk pilihan percabangan {sunny, overcast, rain}.

$$\begin{aligned} \text{Split info}(\text{Semua}, \text{overcast}) &= \left( -\frac{5}{14} * \log_2 \left( \frac{5}{14} \right) \right) + \left( -\frac{4}{14} * \log_2 \left( \frac{4}{14} \right) \right) \\ &\quad + \left( -\frac{5}{14} * \log_2 \left( \frac{5}{14} \right) \right) \end{aligned}$$

$$= 0.531 + 0.516 + 0.531 = 1.577$$

$$\text{Rasio Gain}(\text{Semua}, \text{overcast}) = \frac{0.247}{1.577}$$

$$= 0.156$$

Hasil untuk perhitungan *rasio gain* lainnya ada pada tabel 2.5.

**Tabel 2.5** Hasil perhitungan *Rasio gain* untuk setiap pilihan cabang

		Jumlah	Split Inf	Gain	Rasio Gain
<b>Total</b>		14		0.247	
<b>Pilihan 1</b>	<b>sunny</b>	5	1.577		0.156
	<b>overcast</b>	4			
	<b>rain</b>	5			
<b>Pilihan 2</b>	<b>sunny</b>	5	0.940		0.262
	<b>overcast    rain</b>	9			
<b>Pilihan 3</b>	<b>sunny    overcast</b>	9	0.940		0.262
	<b>rain</b>	5			
<b>Pilihan 4</b>	<b>sunny    rain</b>	10	0.863		0.286
	<b>overcast</b>	4			

Dari tabel 2.5 pilihan 4 yaitu {*sunny*, *rain*} dan {*overcast*} memiliki nilai *rasio gain* tertinggi, maka atribut terpilih (*outlook*) akan dibagi menjadi dua cabang. Pembagian cabang disajikan pada tabel 2.6 dan tabel 2.7.

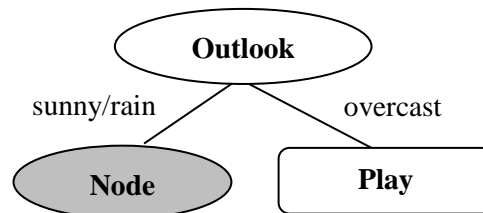
**Tabel 2.6** Pembagian cabang (*sunny, rain*)

<b>Outlook</b>	<b>Temperature</b>	<b>Humidity</b>	<b>Windy</b>	<b>Class</b>
Sunny	75	70	TRUE	Play
Sunny	80	90	TRUE	Don't Play
Sunny	85	85	FALSE	Don't Play
Sunny	72	95	FALSE	Don't Play
Sunny	69	70	FALSE	Play
Rain	71	80	TRUE	Don't Play
Rain	65	70	TRUE	Don't Play
Rain	75	80	FALSE	Play
Rain	68	80	FALSE	Play
Rain	70	96	FALSE	Play

**Tabel 2.7** Pembagian cabang (*overcast*)

<b>Outlook</b>	<b>Temperature</b>	<b>Humidity</b>	<b>Windy</b>	<b>Class</b>
Overcast	72	90	TRUE	Play
Overcast	83	78	FALSE	Play
Overcast	64	65	TRUE	Play
Overcast	81	75	FALSE	Play

Pada cabang *overcast* memiliki kelas yang sama yaitu *Play*, maka *node* ini akan menjadi daun dengan nilai *Play*. Sedangkan cabang *sunny* dan *rain* masih ada kelas yang berbeda, maka akan memilih atribut sebagai *node* seperti ditunjukkan pada gambar 2.3. Proses tersebut akan berulang sampai semua kasus pada cabang memiliki kelas yang sama atau menjadi daun(*leaf*).

**Gambar 2.3** Hasil pembentukan cabang pada node akar

Berikut aturan IF THEN untuk *decision tree* pada penjelasan pada gambar 2.3.

IF Outlook = overcast THEN Class = Play



## 2.5 Evaluasi Sistem

Sistem prediksi (klasifikasi) tidak bisa bekerja 100% benar, maka pada bagian ini akan mengevaluasi hasil perhitungan prediksi. Evaluasi ini menggunakan *Confusion Matrik* yaitu tabel yang digunakan untuk menentukan kinerja suatu model klasifikasi.

Untuk mengukur nilai akurasi yang didapat dari hasil pengujian, menggunakan rumus 3.1. Sedangkan untuk mengukur tingkat kesalahannya menggunakan rumus 3.2.

$$Akurasi = \frac{Jumlah\ h\ data\ yang\ diklasifikasi\ secara\ benar}{Jumlah\ h\ klasifikasi\ yang\ dilakukan} \times 100\% \dots\dots\dots(3.1)$$

$$Laju\ Error = \frac{Jumlah\ h\ data\ yang\ diklasifikasi\ secara\ salah}{Jumlah\ h\ klasifikasi\ yang\ dilakukan} \times 100\% \dots\dots\dots(3.2)$$

Sensitivitas akan mengukur proporsi positif asli yang dikenali (diprediksi) secara benar sebagai positif asli. Rumus perhitungannya menggunakan rumus 3.3. Sedangkan spesifisitas akan mengukur proporsi negatif asli yang dikenali (diprediksi) secara benar sebagai negatif asli. Rumus perhitungannya menggunakan rumus 3.4.

$$Sensitivitas = \frac{TP}{TP + FN} \times 100\% \dots\dots\dots(3.3)$$

Keterangan:

TP : Terima yang diprediksi secara benar sebagai Terima

FN : Terima yang diprediksi secara salah sebagai Tidak

$$Spesifisitas = \frac{TN}{FP + TN} \times 100\% \dots\dots\dots(3.4)$$

Keterangan:

TN : Tidak yang diprediksi secara benar sebagai Tidak

FP : Tidak yang diprediksi secara salah sebagai Terima

Tabel 3.78 merupakan tabel *Confusion Matrik* yang mengambil nilai dari hasil pengujian sistem.

**Tabel 3.78** *Confusion Matrik*

Evaluasi		Klasifikasi	
		Terima	Tidak
Status penerima Asli	Terima	TP	FN
	Tidak	TN	FP