BAB II
LANDASAN TEORI

2.1 Keluarga Mampu dan Tidak Mampu

Keluarga adalah unit terkecil dari masyarakat yang terdiri atas kepala keluarga dan beberapa orang yang terkumpul dan tinggal di suatu tempat di bawah suatu atap dalam keadaan saling ketergantungan (Wikipedia, 2014). Keluarga bisa di golongkan menjadi dua golongan yaitu keluarga mampu dan tidak mampu, keluarga mampu yaitu bisa dikatan keluarga yang bisa memenuhi kebutuhan dan keinginan bahkan lebih dan keluarga tidak mampu yaitu bisa dikatakan keluarga yang belum bisa memenuhi kebutuhan.

Ketentuan sekolah dalam menentukan keluarga mampu dan tidak mampu ialah dari hasil survei yang telah di lakukan pihak sekolah dari data keluarga, dilihat dari fisik rumah dan keluarga tersebut bisa memenuhi kebutuhan keluarga atau tidak, sedangkan pada penelitian ini untuk menentukan keluarga siswa mampu dan tidak mampu dilihat dari beberapa atribut yaitu jumlah saudara kandung, jumlah saudara tiri, jumlah saudara yang bekerja, rata-rata penghasilan saudara perbulan, pekerjaan ayah, pekerjaan ibu dan rata-rata penghasilan orang tua perbulan dan akan dilakukan perhitungan dengan sebuah metode mining agar mengetahui yang mana keluarga siswa mampu dan tidak mampu.

2.2 Definisi Sistem

Sistem merupakan istilah dari bahasa yunani “system” yang artinya adalah himpunan bagian unsu yang saling berhubungan secara teratur untuk mencapai tujuan bersama.

Beberapa pendapat menurut ahli yang mendukung tentang pengertian sistem antara lain adalah:

1. Sistem secara fisik adalah kumpulan dari elemen-elemen yang beroperasi bersama-sama untuk menyelesaikan suatu sasaran. [1]
2. Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. [2]

3. sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. [4]
4. sistem adalah suatu kerangka kerja terpadu yang mempunyai satu sasaran atau lebih. Sistem ini mengkoordinasikan sumber daya yang dibutuhkan untuk mengubah masukan-masukan menjadi keluaran. Sumber daya dapat berupa manusia, bahan, mesin, maupun tenaga surya tergantung pada jenis sistem yang dibicarakan . [9]

2.3 Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database. Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machinelearning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar. [10]

Data mining merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, database, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari database yang besar. [12]

2.3.1 Pengelompokan Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, [10]:

1. Deskripsi

   Deskripsi adalah menggambaran pola dan kecenderungan yang terdapat dalam data secara sederhana. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.
2. Klasifikasi

   Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang telah diklasifikasi dan dengan menggunakan

hasilnya untuk memberikan sejumlah aturan. Klasifikasi menggunakan *supervisedlearning*.

3. Estimasi

   Estimasi hampir sama dengan klasifikasi, perbedaanya adalah variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun dengan menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi.

4. Prediksi

   Prediksi memiliki kesamaan dengan klasifikasi dan estimasi, perbedaanya adalah hasil dari prediksi akan ada dimasa mendatang. Beberapa teknik yang digunakan dalam klasifikasi dan estimasi dapat juga digunakan (untuk keadaan yang tepat) untuk prediksi.

5. Klastering

   Klastering merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-obek yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain. Klastering menggunakan *unsupervised learning*.

6. Asosiasi

   Tugas asosiasi atau sering disebut juga sebagai *market basket analysis* dalam data mining adalah menemukan relasi atau korelasi diantara himpunan item-item dan menemukan atribut yang muncul dalam satu waktu. Asosiasi menggunakan *unsupervised learning*. Penting tidaknya suatu aturan assosiatif dapat diketahui dengan dua parameter, *support* dan *confidence*.

Metode yang akan digunakan pada penelitian ini termasuk kedalam kelompok prediksi, karena menggunakan teknik klasifikasi yang hasilnya akan ada dimasa mendatang.

2.3.2 Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Databases (KDD) adalah keseluruhan proses untuk mengkonversi data mentah menjadi suatu pengetahuan yang bermanfaat. Istilah data mining dan *Knowledge Discovery in Databases* (KDD) sering kali

digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Salah satu tahapan dalam keseluruhan proses KDD adalah data mining.

Proses KDD secara garis besar dapat dijelaskan sebagai berikut. [13]

1. Data Selection

   Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. Pre-processing/Cleaning

   Sebelum proses data mining, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak.

3. Transformation

   Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. Data mining

   Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik tertentu. Teknik, metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. Interpretation Evaluation

   Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.4 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. [11]

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.

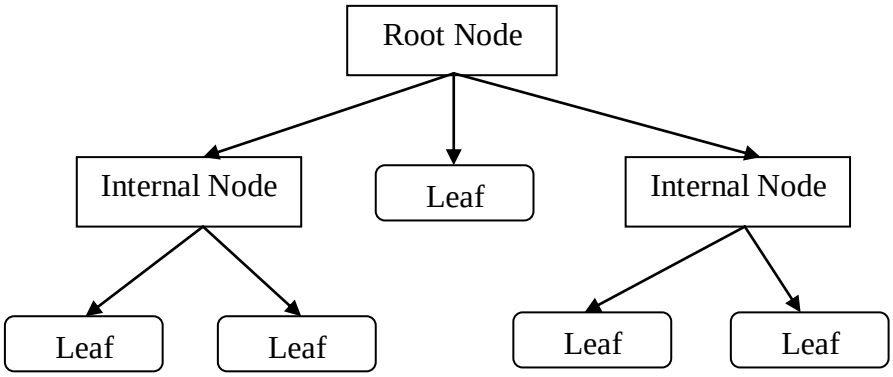
2.4.1 Model Decision Tree

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:

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*

2.4.2 Algoritma C4.5

Algoritma C4.5 diperkenalkan oleh Quinlan (1996) sebagai versi perbaikan dari ID3. Dala ID3, induksi decision tree hanya bisa dilakukan pada fitur bertipe kategorikal (nominal atau ordinal), sedangan tipe numerik (interval atau rasio) tidak apat digunakan. [3]

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 ketik fitur ini menjadi node (akar maupun interval) hanya punya dua pilihan cabang.

2. Fitur kategorikal

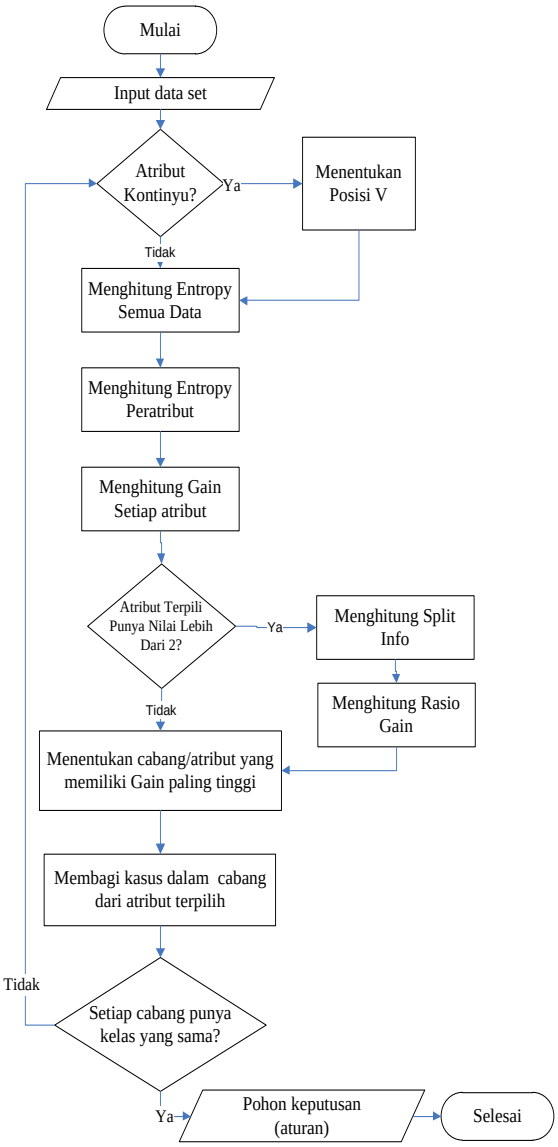
   Untuk fitur yang nilainya bertipe kategorikal (nominal atau ordinal) bisa mempunyai beberapa nilai berbeda. Secara umum ada 2 pemecahan yaitu pemecahan biner (*binary splitting*) dan (*multi splitting*).

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.



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

Flowcart pada **Gambar 2.2** diatas merupakan penjelaskan secara lebih detail tentang algoritma 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 darisebuah 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.

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

$$Laju\ error = \frac{Jumla\ h\ data\ \ yang\ \ diprediksi\ \ \ secara\ \ sala\ h}{Jumla\ h\ prediksi\ \ yang\ \ dilakukan} \quad (2.6)$$

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 : Label mampu yang diprediksi secara benar sebagai Label mampu

FN : Label mampu yang diprediksi secara salah sebagai Label tidak mampu

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

Keterangan:

TN : Label tidak mampu yang diprediksi secara benar sebagai Label tidak mampu

FP : Label tidak mampu yang diprediksi secara salah sebagai Label mampu

2.4.3 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 kontinyu sedangkan *outlook* dan *windy* bertipe kategorikal. Sedangkan kolom *Class* adalah kelas tujuannya atau label kelas-nya.

Tabel 2.1 Contoh data set

Outlook	Temperature	Humidity	Windy	Class
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	∑ Play	∑ 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:

$$Gain(outlook) = 0.940 - \left(\frac{5}{14} * \left(-\frac{2}{5} * log_2\left(\frac{2}{5}\right) - \frac{3}{5} * log_2\left(\frac{3}{5}\right)\right) + \frac{4}{14} * \left(-\frac{4}{4} * log_2\left(\frac{4}{4}\right) - \frac{0}{4} * log_2\left(\frac{0}{4}\right)\right) + \frac{5}{14} * \left(-\frac{3}{5} * log_2\left(\frac{3}{5}\right) - \frac{2}{5} * log_2\left(\frac{2}{5}\right)\right)\right)$$

$$= 0.940 - 0.694$$

$$= 0.246$$

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:

$$Gain(temp) = 0.940 - \left(\frac{2}{14} * \left(-\frac{1}{2} * log_2\left(\frac{1}{2}\right) - \frac{1}{2} * log_2\left(\frac{1}{2}\right)\right) + \frac{12}{14} * \left(-\frac{8}{12} * log_2\left(\frac{8}{12}\right) - \frac{4}{12} * log_2\left(\frac{4}{12}\right)\right)\right)$$

$$= 0.940 - 0.930$$

$$= 0.010$$

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

Tabel 2.3 Hasilperhitungan posisi V untuk atribut *temperature*

Temperature	**65**		**70**		**75**		**80**	
	≤	>	≤	>	≤	>	≤	>
Play	1	8	4	5	7	2	7	2
Don't Play	1	4	1	4	3	2	4	1
Jumlah	**2**	**12**	**5**	**9**	**10**	**4**	**11**	**3**

Entropy	**1.000**	**0.918**	**0.722**	**0.991**	**0.881**	**1.000**	**0.946**	**0.918**
Gain	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 Hasilperhitungan *Information gain* untuk setiap atribut

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

$$Split\ info(Semua, overcast) = \left(-\frac{5}{14} * log_2\left(\frac{5}{14}\right)\right) + \left(-\frac{4}{14} * log_2\left(\frac{4}{14}\right)\right) + \left(-\frac{5}{14} * log_2\left(\frac{5}{14}\right)\right)$$

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

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

$$= 0.156$$

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

Tabel 2.5 Hasilperhitungan *Rasio gain* untuk setiap pilihan cabang

		Jumlah	Split Inf	Gain	Rasio Gain
Total		14		0.247	
Pilihan 1	**sunny**	5	1.577		0.156
	overcast	4			
	rain	5			
Pilihan 2	**sunny**	5	0.940		0.262
	overcast Rain	9			
Pilihan 3	**sunny overcast**	9	0.940		0.262
	rain	5			
Pilihan 4	**sunny Rain**	10	0.863		0.286
	overcast	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*)

Outlook	Temperature	Humidity	Windy	Class
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*)

Outlook	Temperature	Humidity	Windy	Class
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*. Proses tersebut akan berulang sampai semua kasus pada cabang memiliki kelas yang sama atau menjadi daun(*leaf*).

2.5 Penelitian Sebelumnya

Penelitian sebelumnya yang menggunakan metode naive bayes di lakukan oleh Meinggan Vilian Sari, lulusan Universitas Muhammadiyah Gresik Tahun 2014. penelitian untuk "*memprediksi prestasi (IPK) mahasiswa berdasarkan latar belakang sekolah asal dan atribut mahasiswa ketika awal masuk kuliah mengunakan Naïve Baye"*. Adapun data yang diambil dalam penilitian ini adalah sampel dari 103 *record* dengan kelas "Tinggi" dan "Rendah" masing-masing berjumlah 69 dan 34 yang akan dibagi menjadi datalatih data uji. Dan mengunakan 6 variable, adapun variabel yang dipakai : Instansi Sekolah, Setatus Sekolah, Jurusan Sekolah, Motivasi Pilihan Kuliah, Status Kerja, Nilai Danem. Dan keakurasian hasil penelitian mengunakan metode Naïve Bayes ini menunjukan akurasi tertinggi pada penggujian pertama adalah 84.62%. dan pada pengujian keempat, ketiga percobaan menghasilkan nilai sensitivitas 100%, yang artinnya semua data uji kelas tinggi diprediksi secara benar mempunyai kelas tinggi.

Penelitian selanjutnya adalah metode pohon keputusan C4.5 adalah penelitian yang berjudul "*System prediksi prestasi akademik mahasiswa menggunakan metode decision tree C4.5 (Studi kasus:Jurusan Teknik informatika UNMUH GRESIK)"*, dibuat oleh Aunur Rasyid (Universitas Muhammadiyah Gresik, 2014). Tujuan dari penelitian tersebut adalah untuk menghasilkan informasi perkiraan kategori prestasi mahasiswa menggunakan metode *Decision Tree C4.5* sebagai peringatan dini dan motivasi mahasiswa dalam mendapatkan prestasi yang maksimal. Atribut-atribut yang digunakan adalah instansi sekolah asal (SMK,SMA atau MA), satatus sekolah asal (Negri atau Swasta), jurusan sekolah asal (IPA,IPS,Bahasa,Teknik,Administrasi), nilai rata-rata UN, status

kerja (Sudah atau Belum), dan pihak yang mempengaruhi mahasiswa dalam memilih kuliah (Sendiri,Orang tua atau Orang lain). Hasil dari penelitian tersebut, Sistim Prediksi mahasiswa yang dirancang menggunakan algoritma C4.5 dapat memprediksi prestasi mahasiswa agar mampu mempertahankan kondisinya atau melakukan perbaikan utuk mencapai prestasi yang maksimal. Hasil akurasi dari penelitian tersebut adalah 90%

Penelitian menggunakan algoritma C4.5 juga dilakukan oleh Angga Raditya (Universitas Gunadarma, 2011), yaitu penelitian tentang pencarian pola prediksi hujan menggunakan algoritma C4.5. Data yang digunakan adalah data cuaca yang tersimpan di *World Meteorologi Organization* (organisasi pengawas cuaca dunia). Dari data tersebut akan diolah untuk pola prediksi hujan dengan menggunakan algoritma C4.5. akurasi pola prediksi yang didapat mampu mencapai 79%. Kelebihan algoritma C4.5 dalam membangun pohon keputusan prediksi cuaca adalah kemampuannya menangani data kontinyu maupun data nominal, karena hampir seluruh atribut cuaca yang digunakan bertipe data kontinyu. Selain itu dalam membangun keputusan tingkat *error*-nya lebih sedikit.

Penelitian menggunakan algoritma C4.5 juga dilakukan oleh Liliana swistina (Universitas STMIK Indonesia, 2013), yaitu penelitian tentang penerapan algoritma c.4.5 untuk menentukan jurusan mahasiswa. Evaluasi pengukurn rapidminer yaitu membandingkan nilai akurasi antara algoritma c4.5 dan naive bayes. Data yang digunakan adalah data mahasiswa baru STMIK Indonesia Banjarmaasin tahun 2008/2009. Data sempel terdiri dari atribut nama, jenis kelamin, umur, asal sekolah, nilai uan, ipk semester 1 dan semester 2. Hasil akurasi yang didapatkan adalah 93,31% untuk metode c4.5 dan 89,02% untuk naive bayes.

Muhammad baharrudin rabbani (Universitas Muhammadiyah Gresik, 2014) melakukan penelitian sistem klasifikasi keluarga siswa mampu dan tidak mampu untuk mendapatkan beasiswa. Algoritma yang digunakan adalah *naive bayes.* Data yang digunakan terdiri dari 105 keluarga yang diperoleh dari data keluarga siswa kelas X dan XI MA Muhammadiyah 1 Sumbberrejo Kab.

Bojonegoro tahun ajaran 2013/2014. Penelitian ini diuji sebanyak 3 kali hasil uji pertama menghasilkan akurasi 81,13%, kedua 69,23% dan yang terakhir 86,67%.