ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

ANALISA TEKNIK PENENTUAN ATRIBUT DALAM MEMBUAT POHON KEPUTUSAN PADA PENAMBANGAN DATA

Dian Wirdasari

Program Studi Ilmu Komputer, Universitas Sumatera Utara
Jl. Alumni No. 9 Kampus USU Padang Bulan Medan
E-mail: dianws@gmail.yahoo.com

Abstrak

Dalam *data mining* (penambangan data), salah satu teknik yang sering digunakan adalah teknik *decision tree* (pohon keputusan). Dalam membuat pohon keputusan diperlukan 3 tahapan, yaitu: tahap pembentukan pohon, pemangkasan pohon, dan pembentukan aturan dan keputusan. Tahap pembentukan pohon yaitu tahap yang akan dibentuk suatu pohon yang terdiri dari akar yang merupakan node paling awal, daun sebagai distribusi kelas, dan batang yang menggambarkan hasil keluaran dari pengujian. Dalam pohon keputusan, data dinyatakan dalam bentuk tabel dengan atribut dan record. Atribut memiliki nilai-nilai yang disebut dengan instans (*instance*). Penentuan atribut sebagai node terpilih dalam pohon keputusan dapat dilakukan dengan menggunakan beberapa metode. Pada artikel ini akan diuraikan dua metode, yaitu dengan indeks *gini* dan *information gain*.

Kata Kunci: *Penambangan Data, Pohon Keputusan, Atribut, Indeks Gini, Information Gain*

Abstract

One of many technique on data mining is decision tree technique. Three phases are required to build a decision tree. There are, phase of forming tree, phase of barbering tree, and phase of establishing the *rules*. Phase of forming tree is a phase to build a tree which consisting of a root as a beginning node, leaves as class distribution, and branches as output of test. In a decision tree, data is clarified on a table with attributs and records. An attribut have one or some value which called its instances. To determine an attribut as a selected node in a decision tree, is carried out with use some methods. In this article, will described two methods, that is gini index and information gain.

Key Word: *Data Mining, Decision Tree, Attribut, Gini Index, Information Gain*

PENDAHULUAN

Data mining (penambangan data) merupakan proses untuk menemukan pengetahuan (*knowledge discovery*) yang ditambang dari sekumpulan data yang volumenya sangat besar. Aplikasi *data mining* pada pengelolaan bisnis, pengendalian produksi, dan analisa pasar misalnya, memungkinkan diperolehnya pola dan hubungan yang dapat dimanfaatkan untuk peningkatan penjualan, atau pengelolaan sumber daya dengan lebih baik.

Data mining mengacu pada proses untuk menambang (*mining*) pengetahuan dari sekumpulan data yang sangat besar [*Jiawei*, 2001]. Sebenarnya *data mining* merupakan suatu langkah dalam *knowledge discovery in databases* (KDD). *Knowledge discovery* sebagai suatu proses terdiri atas pembersihan data (*data cleaning*), integrasi data (*data integration*), pemilihan data (*data selection*), transformasi data (*data transformation*), *data mining*, evaluasi pola (*pattern evaluation*) dan penyajian pengetahuan (*knowledge presentation*).

Data mentah yang dihasilkan dari pengumpulan data, biasanya tersimpan dalam bentuk beberapa tabel basis data. Karena analisis data umumnya dilakukan terhadap suatu tabel tunggal, maka perlu dilakukan penggabungan (join) beberapa tabel yang relevan. Hasilnya adalah suatu struktur yang disebut dengan dataset, seperti tampak pada tabel 1 berikut. [*Nilakant*, 2004].

Tabel 1. Format Dataset

	Atribut -1	Atribut -2	…	Atribut-n
Instans-1	$X_{1,1}$	$X_{1,2}$		$X_{1,n}$
Instans-2	$X_{2,1}$	$X_{2,2}$		$X_{2,n}$
…				
Instans-m	$X_{m,1}$	$X_{m,2}$		$X_{m,n}$

Dataset dapat dikelompokkan secara vertikal sebagai kumpulan atribut dan secara horisontal sebagai kumpulan instans (*instance*). Setiap atribut mempunyai tipe data, yang dapat berupa numerik, teks, atau bentuk lainnya. Jika domain nilai suatu atribut berhingga, maka disebut atribut nominal. Suatu instans adalah data yang dihasilkan dari suatu kejadian di dunia nyata, yang dicatat dalam beberapa atribut.

Proses analisis data dengan menerapkan teknik *data mining* dapat dilakukan melalui analisis statistik atau dengan pendekatan *machine learning*. Salah satu teknik analisis data dengan pendekatan *machine learning* yangakan dijelaskan pada artikel ini yaitu teknik klasifikasi (*classification*).

KLASIFIKASI (*CLASSIFICATION*)

Classification adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri dapat berupa aturan "jika-maka", berupa pohon keputusan (*decision tree*), formula matematika atau *neural network*.

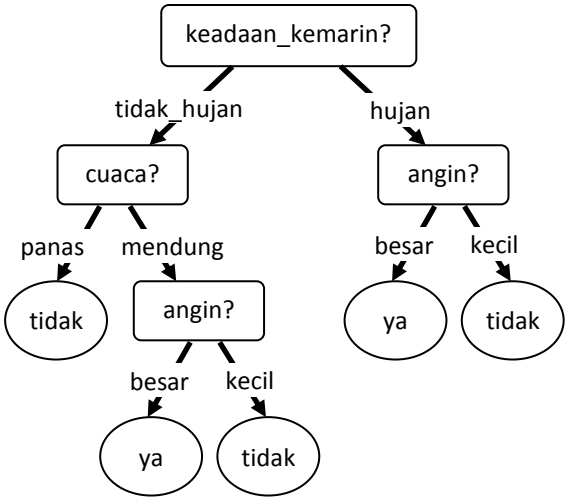
Teknik *classification* bekerja dengan mengelompokkan data berdasarkan data training dan nilai atribut klasifikasi. Aturan pengelompokan tersebut akan digunakan untuk klasifikasi data baru ke dalam kelompok yang ada.

Classification yang direpresentasikan dalam bentuk pohon keputusan (*decision tree*) adalah dengan cara, setiap node dalam pohon keputusan menyatakan suatu tes terhadap atribut dataset, sedangkan setiap cabang menyatakan hasil dari tes tersebut. Pohon keputusan yang terbentuk dapat diterjemahkan menjadi sekumpulan aturan dalam bentuk IF condition THEN outcome.

POHON KEPUTUSAN (*DECISION TREE*)

Dalam *decision tree*, untuk mengelompokkan objek digunakan atribut dan nilai atribut tersebut. Misalnya, untuk objek buah-buahan yang dapat dibedakan berdasarkan atribut bentuk, warna, ukuran dan rasa. Bentuk, warna, ukuran dan rasa adalah besaran nominal, yaitu bersifat kategoris dan tiap nilai tidak dapat dijumlahkan atau dikurangkan. Dalam atribut warna, ada beberapa nilai yang mungkin yaitu, hijau, kuning, dan merah. Dalam atribut ukuran, ada nilai besar, sedang dan kecil. Dengan nilai-nilai atribut ini kemudian akan dibuat *decision tree* untuk menetapkan suatu objek termasuk ke dalam jenis buah apa jika nilai tiap-tiap atribut diberikan.

Teknik *decision tree* dibagi menjadi 3 tahap, yaitu: tahap pembentukan pohon, pemangkasan pohon, dan pembentukan aturan dan keputusan. Tahap pembentukan pohon yaitu tahap yang akan dibentuk suatu pohon yang terdiri dari akar yang merupakan node paling awal, daun sebagai distribusi kelas, dan batang yang menggambarkan hasil keluaran dari pengujian. Contoh pohon keputusan diperlihatkan pada Gambar 1 berikut.



Gambar 1. Contoh *Decision Tree*

Dari Gambar 1 terlihat bahwa setiap percabangan menyatakan kondisi yang harus dipenuhi dan tiap ujung pohon menyatakan kelas data. Gambar 1 merupakan contoh pohon keputusan untuk memprediksi hujan atau tidak hari ini, dari pohon keputusan tersebut, diketahui bahwa prediksi hari ini akan terjadi hujan jika keadaan kemarin hujan dan hari ini terjadi angin besar.

Konsep dari pohon keputusan adalah bagaimana mengubah data menjadi pohon keputusan dan aturan-aturan keputusan (*rule*). Dari contoh pohon keputusan pada Gambar 1, dapat ditentukan aturan (*rule*) yang dapat digunakan untuk menentukan apakah hari ini hujan atau tidak berdasarkan data keadaan kemarin, cuaca, dan angin.

R1: IF keadaan_kemarin = hujan ^ angin = besar THEN hari_ini_hujan = ya

R2: IF keadaan_kemarin = hujan ^ angin = kecil THEN hari_ini_hujan = tidak

R3: IF keadaan_kemarin = tidak_hujan ^ cuaca = panas THEN hari_ini_hujan = tidak

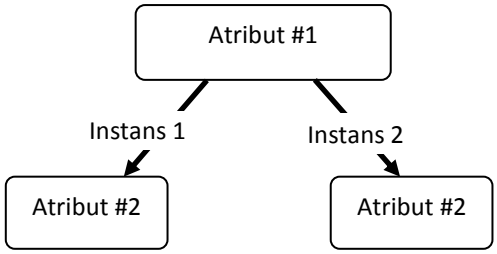
R4: IF keadaan_kemarin = tidak_hujan ^ cuaca = mendung ^ angin = besar THEN hari_ini_hujan = ya

R5: IF keadaan_kemarin = tidak_hujan ^ cuaca = mendung ^ angin = kecil THEN hari_ini_hujan = tidak

PROSES DATA MENJADI POHON

Dalam pohon keputusan, data dinyatakan dalam bentuk tabel dengan atribut dan record. Atribut menyatakan suatu *parameter* yang dibuat sebagai kriteria dalam pembentukan pohon. Misalkan pada contoh Gambar 1, untuk menentukan hari ini hujan atau tidak, kriteria yang diperhatikan adalah keadaan kemarin, cuaca, dan angin. Salah satu atribut merupakan atribut yang menyatakan data solusi per-item data yang disebut dengan target atribut.

Atribut memiliki nilai-nilai yang disebut dengan instans (*instance*). Misalkan atribut cuaca mempunyai instans berupa panas dan mendung. Atribut itulah yang nantinya akan menjadi node pada pohon keputusan yang akan dibuat. Dari tabel format dataset pada tabel 1, dapat dibuat pohon keputusan seperti gambar 2 berikut.



Gambar 2. Pohon keputusan dari format dataset pada tabel 1

Bagaimana memilih atribut? Jawabnya adalah: atribut yang memungkinkan untuk memperoleh pohon keputusan yang paling kecil ukurannya. Atau atribut yang bias memisahkan objek menurut kelasnya.

Secara *heuristic* dipilih atribut yang menghasilkan node yang paling *purest* (paling bersih). Kalau dalam satu cabang anggotanya berasal dari satu kelas maka cabang ini disebut *pure*. Semakin *pure* suatu cabang akan semakin baik. Ukuran *purity* dinyatakan dengan tingkat *impurity*. Kriteria-kriteria *impurity* yang sering digunakan adalah *information gain*, *gain ratio*, dan Indeks *gini*. Dalam artikel ini akan dijelaskan penggunaan *information gain* dan indeks *gini* dalam membuat suatu pohon keputusan.

1. *Information Gain*

Dalam memilih atribut dengan *information gain* adalah dengan memilih atribut yang memiliki *information gain* paling besar. Sebelum menghitung *information gain* maka perlu dihitung terlebih dahulu nilai informasi dalam satuan bits dari suatu kumpulan objek. Digunakan *entropi* untuk menghitungnya. Entropi menyatakan *impurity* suatu kumpulan objek.

Jika diberikan sekumpulan objek dengan output/label *y* yang terdiri dari objek berlabel 1, 2 sampai *n*, maka besarnya entropi dari objek dengan kelas *n* ini dihitung dengan rumus:

Entropi(y) = $-p_1\log_2 p_1 - p_2\log_2 p_2 \ldots - p_n\log_2 p_n$

Dimana p_1, p_2, … , p_n masing-masing menyatakan proporsi kelas 1, kelas 2, …, kelas *n* dalam output *y*.

Jika perbandingan dua kelas, rasionya sama maka nilai entropinya 1. Jika satu set hanya terdiri dari satu kelas, maka nilai entropinya 0.

Information gain dari output data *y* yang dikelompokkan berdasarkan atribut *A*, dinotasikan dengan $Gain_{info}(y,A)$ yang dihitung dengan rumus berikut:$Gain_{info}(y, A) = entropi(y) - \sum_{c \in nilai(A)} \frac{y_c}{y} entropi(y_c)$

Dimana nilai(*A*) adalah semua nilai yang mungkin dari atribut *A*, dan $y_c$adalah subset dari *y* dimana *A* mempunyai nilai *c*.

2. Indeks *Gini*

Jika kelas objek dinyatakan dengan *k, k = 1, 2, …, c,* dimana *c* adalah jumlah kelasuntuk output *y*, indeks *gini* (*IG*) untuk suatu cabang *A* dihitung dengan rumus:

$$IG(A) = 1 - \sum_{k=1}^{c} p_k^2$$

Dimana p_k adalah rasio observasi dalam cabang *A* yang masuk ke dalam kelas *k*. Jika *IG(A) = 0* berarti semua data dalam cabang *A* berasal dari kelas yang sama. Nilai *IG(A)* mencapai maksimum jika dalam cabang *A* proporsi data dari masing-masing kelas yang ada memiliki nilai yang sama. Setelah itu kemudian dihitung nilai *GiniSplit* untuk masing-masing atribut. Rumus *GiniSplit* adalah:

$$GiniSplit(A) = p_1 IG(p_1) + p_2 IG(p_2) + \cdots + p_k IG(p_k)$$

Dalam memilih atribut dengan *indeks gini* adalah dengan memilih atribut yang memiliki *GiniSplit* paling kecil.

Berikut ini adalah contoh pembuatan pohon keputusan untuk memprediksi hari ini hujan atau tidak berdasarkan data cuaca, angin, dan keadaan_kemarin yang disajikan pada tabel 2 berikut.

Tabel 2. Data cuaca, angin dan keadaan_kemarin untuk prediksi hujan atau tidak hari ini

Cuaca	Angin	Keadaan_Kemarin	Akan_ Hujan
Panas	Kecil	Tdk_Hujan	Tidak
Mendung	Kecil	Hujan	Ya
Cerah	Besar	Hujan	Ya
Cerah	Kecil	Tdk_Hujan	Tidak
Mendung	Besar	Tdk_Hujan	Ya
Panas	Besar	Tdk_Hujan	Tidak
Panas	Besar	Hujan	Ya
Mendung	Besar	Hujan	Ya

3. Penyelesaian dengan *InformationGain*

Pertama sekali akan dihitung entropi untuk masing-masing atribut. Kemudian akan dihitung nilai *information gain* nya masing-masing. Atribut dengan nilai *information gain* yang paling besar yang diambil menjadi atribut terpilih.

Ent(Ya/Tidak) = Ent(5,4)
= – (5/9)($\log_2$(5/9))
– (4/9) ($\log_2$(4/9))
= 0,472 + 0,521 = 0,993

Mencari entropi Cuaca:

Ent(Cuaca[Panas]) = Ent(1,2)
= – (1/3)($\log_2$(1/3))
– (2/3) ($\log_2$(2/3))
= 0,529 + 0,392 = 0,921

Ent(Cuaca[Mendung]) = Ent(3,1)
= – (3/4)($\log_2$(3/4))
– (1/4) ($\log_2$(1/4))
= 0,814

Ent(Cuaca[Cerah]) = Ent(1,1) = 1

Ent(Cuaca[Panas,Mendung,Cerah]) =
= Ent((1,2), (3,1), (1,1))
= (3/9)(0,921) +
(4/9)(0,814) + (2/9)(1)
= 0,891

Gain$_{info}$(Cuaca)= Entropi sebelum dipisah – Entropi sesudah dipisah
= 0,993 – 0,891 = **0,102**

Mencari entropi Angin:

Ent(Angin[Kecil]) = Ent(1,3) = 0,814

Ent(Angin[Besar]) = Ent(4,1)
= – (4/5)($\log_2$(4/5))
– (1/5) ($\log_2$(1/5))
= 0,724

Ent(Angin[Kecil,Besar]) =
= Ent((1,3), (4,1))
= (4/9)(0,814) +
(5/9)(0,724)
= 0,764

Gain$_{info}$(Angin)= 0,993 – 0,764 = **0,229**

Mencari entropi Keadaan_Kemarin:

Ent(Keadaan_Kemarin[Tdk_Hujan]) =
= Ent(1,4)
= – (1/5)($\log_2$(1/5))
– (4/5) ($\log_2$(4/5))
= 0,724

Ent(Keadaan_Kemarin[Hujan]) =
= Ent(4,0) = **0**

Ent(Keadaan_Kemarin[Tdk_Hujan, Hujan])
= Ent((1,4), (4,0))
= (5/9)(0,724) + 0
= 0,402

Gain$_{info}$(Keadaan_Kemarin)= 0,993 – 0,402
= **0,591 →paling besar**

Karena nilai *information gain* dari atribut Keadaan_Kemarin lebih besar yaitu **0,591**, maka atribut ini terpilih sebagai node akar (level 0). Selanjutnya, untuk mencari atribut pada level 1 (di bawah

"Keadaan_Kemarin") adalah dengan menghitung nilai *entropi* dari masing-masing atribut sisa di bawah atribut "Keadaan_Kemarin".

Mencari *entropi* Keadaan_Kemarin ([Tdk_Hujan]| Cuaca):

Ent(Cuaca[Panas]) = Ent(0,2) = 0
Ent(Cuaca[Mendung]) = Ent(1,1) = 1
Ent(Cuaca[Cerah]) = Ent(0,1) = 0
Ent(Cuaca[Panas,Mendung,Cerah]) =
= Ent((0,2), (1,1), (0,1))
= (2/5)(0) + (2/5)(1)
+ (1/5)(0)
= 0,4

$Gain_{info}$(Keadaan_Kemarin[Tdk_Hujan]|Cuaca)
= Entropi sebelum dipisah – Entropi sesudah dipisah
=Ent(Keadaan_Kemarin[Tdk_Hujan]) – Ent(Cuaca[Panas,Mendung,Cerah])
= 0,724 – 0,4 = **0,324**

Mencari *entropi* Keadaan_Kemarin ([Tdk_Hujan] | Angin):

Ent(Angin[Kecil]) = Ent(0,3) = 0
Ent(Angin[Besar]) = Ent(1,1) = 1
Ent(Angin[Kecil,Besar]) =
= Ent((0,3), (1,1))
= (3/5)(0) + (2/5)(1)
= 0,4

$Gain_{info}$(Keadaan_Kemarin[Tdk_Hujan]|Angin) = Ent(Keadaan_Kemarin[Tdk_Hujan])– Ent(Angin[Kecil,Besar])= 0,724 – 0,4 = **0,324**

Karena nilai *information gain* dari atribut Cuaca dan Angin nilainya sama, yaitu 0,324 , maka dipilih atribut Cuaca yang berada di bawah Keadaan_Kemarin untuk instans Tdk_Hujan.

Instans dari atribut Cuaca ada tiga buah yaitu "Panas", "Mendung" dan "Cerah", tetapi karena Ent(Keadaan_Kemarin [Tdk_Hujan] | Cuaca[Panas]) = 0 dan Ent(Keadaan_Kemarin [Tdk_Hujan] | Cuaca [Cerah]) juga = 0, maka level berikut (di bawah Cuaca) untuk instans Panas dan Cerah tidak perlu dilanjutkan lagi.

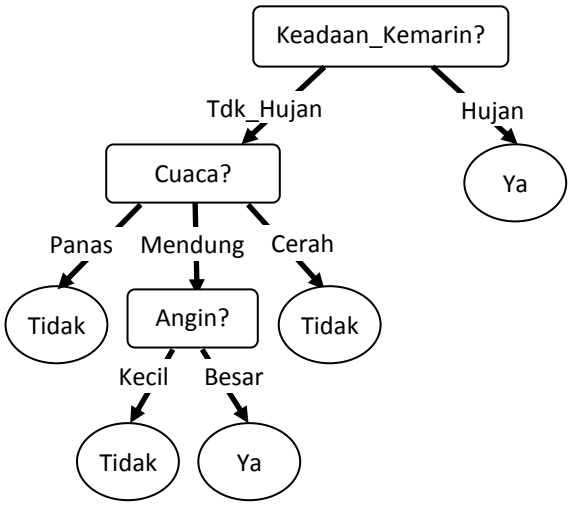
Maka yang dilakukan selanjutnya adalah mencari *entropi* untuk atribut lain di bawah atribut Cuaca untuk instans Mendung. Karena tinggal atribut Angin yang tersisa, maka atribut Angin ditempatkan di bawah instans Mendung.

Mencari *entropi* Keadaan_Kemarin ([Tdk_Hujan]| Cuaca[Mendung]|Angin):

Ent(Angin[Kecil]) = Ent(0,1) = 0
Ent(Angin[Besar]) = Ent(1,0) = 0
Ent(Angin[Kecil,Besar]) = 0

$Gain_{info}$(Keadaan_Kemarin[Tdk_Hujan]| Cuaca[Mendung]|Angin) = 1 – 0 = **1**

Dari hasil perhitungan ini, maka gambar pohon keputusan yang dihasilkan diberikan pada gambar 3 berikut.



Gambar 3. Pohon Keputusan Prediksi Hujan dari Data Cuaca Tabel 2 dengan *InformationGain*

Pertama sekali akan dihitung indeks *gini* (IG) untuk masing-masing atribut. Kemudian akan dihitung nilai *GiniSplit* nya masing-masing. Atribut yang memiliki nilai *GiniSplit* yang paling kecil yang diambil menjadi atribut terpilih.

Menentukan IG(Cuaca[Panas, Mendung, Cerah]):

	Panas	Mendung	Cerah
Ya	1	3	1
Tidak	2	1	1

IG(Cuaca[Panas]) $= 1 - (1/3)^2 - (2/3)^2$
$= 1 - 0{,}111 - 0{,}444$
$= 0{,}445$

IG(Cuaca[Mendung]) $= 1 - (3/4)^2 - (1/4)^2$
$= 1 - 0{,}5625 - 0{,}0625$
$= 0{,}375$

IG(Cuaca[Cerah]) $= 1 - (1/2)^2 - (1/2)^2$
$= 1 - 0{,}25 - 0{,}25$
$= 0{,}5$

GiniSplit(Cuaca) = (3/9) (0,445) + (4/9) (0,375) + (2/9)(0,5)
= **0,425**

Menentukan IG(Angin[Kecil, Besar]):

	Kecil	Besar
Ya	1	4
Tidak	3	1

IG(Angin[Kecil]) $= 1 - (1/4)^2 - (3/4)^2$
$= 1 - 0{,}0625 - 0{,}5625$
$= 0{,}375$

IG(Angin[Besar]) $= 1 - (4/5)^2 - (1/5)^2$
$= 1 - 0{,}64 - 0{,}04 = 0{,}32$

GiniSplit(Angin) = (4/9)(0,375) + (5/9)(0,32)
= **0,343**

Menentukan IG(Keadaan_Kemarin[Tdk_Hujan, Hujan]):

	Tdk_Hujan	Hujan
Ya	1	4
Tidak	4	0

IG(Keadaan_Kemarin[Tdk_Hujan]) =
$= 1 - (1/5)^2 - (4/5)^2$
$= 1 - 0{,}04 - 0{,}64$
$= 0{,}32$

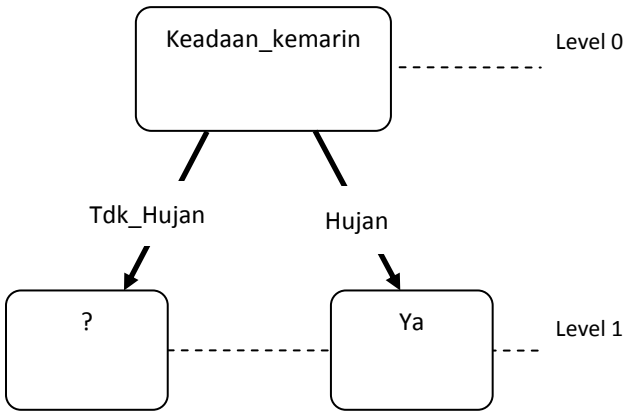
IG(Keadaan_Kemarin[Hujan]) =
$= 1 - (4/4)^2 - (0/4)^2$
= **0**

GiniSplit(Keadaan_Kemarin) = (5/9)(0,32) + 0
= **0,178 → paling kecil**

Karena nilai *GiniSplit* dari atribut "Keadaan_ Kemarin" lebih kecil dari nilai *GiniSplit* atribut-atribut yang lain, maka yang menjadi node akar (level 0) pada pohon keputusan adalah atribut Keadaan_Kemarin.

Langkah selanjutnya adalah menentukan atribut-atribut pada level 1, di bawah atribut "Keadaan_Kemarin". Untuk menentukannya, maka perlu kembali untuk mencari indeks *gini* dari tiap atribut dan nilai *GiniSplit*nya. Karena instans dari atribut "Keadaan_Kemarin" ada dua buah yaitu "Tdk_Hujan" dan "Hujan", maka perlu dicari untuk tiap-tiap nilai instansnya. Tetapi karena indeks *gini* (IG) untuk Keadaan_Kemarin [Hujan] nilainya = 0 maka level berikut (di bawah Keadaan_Kemarin) untuk instans(*instance*) Hujan tidak perlu dilanjutkan lagi.

Maka yang dilakukan selanjutnya adalah mencari indeks *gini* untuk atribut lain di bawah Keadaan_Kemarin untuk instans Tdk_Hujan.



Menentukan IG(Keadaan_Kemarin[Tdk_Hujan] | Cuaca[Panas, Mendung, Cerah]):

	Panas	Mendung	Cerah
Ya	0	1	0
Tidak	2	1	1

IG(Cuaca[Panas]) = 0
IG(Cuaca[Mendung]) = $1 - (1/2)^2 - (1/2)^2$
= 0,5
IG(Cuaca[Cerah]) = 0
GiniSplit(Keadaan_Kemarin[Tdk_Hujan] |Cuaca) = (2/5)(0) + (2/5)(0,5) + (1/5)(0)= **0,2**

Menentukan
IG(Keadaan_Kemarin[Tdk_Hujan] | Angin[Kecil, Besar]):

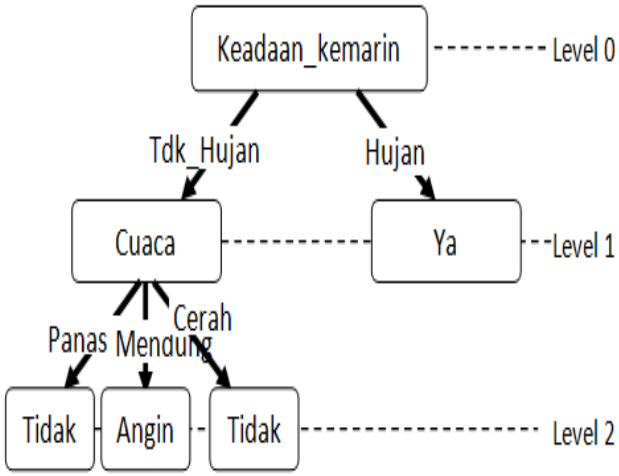
	Kecil	Besar
Ya	0	1
Tidak	3	1

IG(Angin[Kecil]) = 0
IG(Angin[Besar]) = $1 - (1/2)^2 - (1/2)^2$ = 0,5
GiniSplit(Keadaan_Kemarin[Tdk_Hujan] |Angin) = 0 + (2/5)(0,5) =**0,2'**

Karena nilai *GiniSplit* untuk atribut "Cuaca" dan "Angin" sama besarnya yaitu 0,2 maka dapat diambil saja salah satu sebagai atribut pada level 1 di bawah atribut "Keadaan_Kemarin" untuk instans "Tdk_Hujan". Untuk contoh ini dipilih atribut "Cuaca".

Karena instans(*instance*) dari atribut Cuaca ada tiga buah yaitu "Panas", "Mendung" dan "Cerah", maka perlu dicari untuk tiap-tiap nilai instansnya. Tetapi karena indeks *gini* (IG) untuk Keadaan_Kemarin [Tdk_Hujan] |Cuaca [Panas] nilainya = 0 maka level berikut (di bawah Cuaca) untuk instans Panas tidak perlu dilanjutkan lagi. Hal ini sama dengan nilai IG untuk Keadaan_Kemarin [Tdk_Hujan] | Cuaca [Cerah] yang nilainya juga = 0 , maka level berikutnya untuk instans Cerah tidak dilanjutkan lagi.

Maka yang dilakukan selanjutnya adalah mencari indeks *gini* untuk atribut lain di bawah atribut Cuaca untuk instans Mendung. Karena tinggal atribut Angin yang tersisa, maka secara otomatis atribut Angin ditempatkan di bawah instans Mendung. Hasilnya seperti pada gambar 4 berikut.



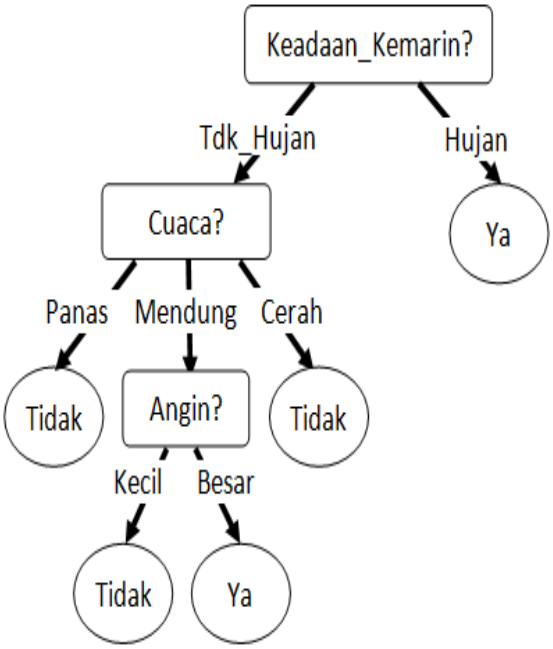
Gambar 4. Atribut Angin ditempatkan di bawah instans Mendung

Menentukan
IG(Keadaan_Kemarin[Tdk_Hujan] | Cuaca[Mendung] | Angin[Kecil, Besar]):

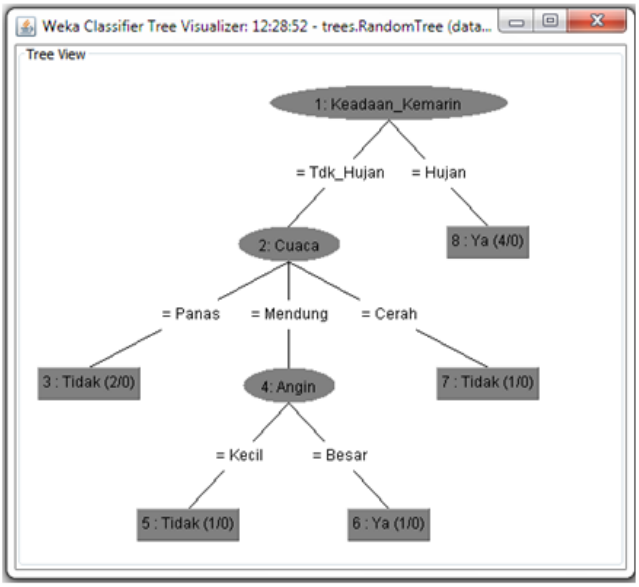
	Kecil	Besar
Ya	0	1
Tidak	1	0

IG(Angin[Kecil]) = 0
IG(Angin[Besar]) = 0
GiniSplit(Keadaan_Kemarin[Tdk_Hujan] |Cuaca[Mendung] |Angin) = **0**

Dari hasil perhitungan ini, maka gambar pohon keputusan yang dihasilkan diberikan pada gambar 5 berikut.



Hasil ini sama dengan pohon keputusan yang diselesaikan menggunakan *information gain*. Jika diuji dengan menggunakan software, dalam kasus ini digunakan *Weka* ver. 3.6.9, pohon keputusan yang dihasilkan seperti pada gambar 6 berikut.

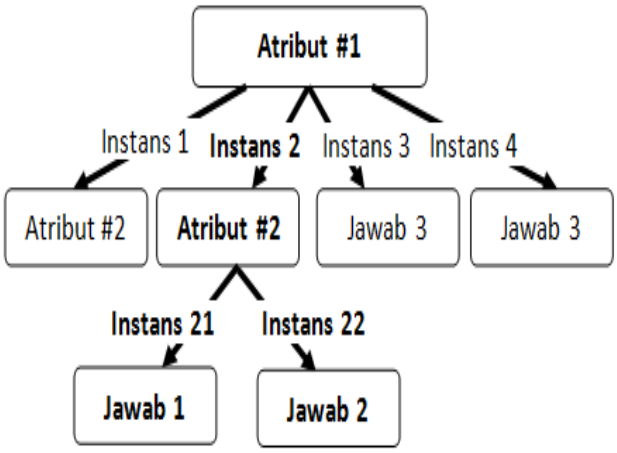


Gambar 6. Pohon Keputusan Prediksi Hujan dari Data Cuaca Tabel 2 dengan menggunakan Weka 3.6.9

MENGUBAH POHON MENJADI *RULES* (ATURAN-ATURAN)

Dalam mengubah sebuah pohon keputusan menjadi aturan-aturan (*rules*), maka yang diperlukan adalah memperhatikan hubungan antara tiap atribut dengan instansnya dan operator yang akan digunakan. Ada dua operator yang biasanya dipakai dalam *rules*, yaitu operator *conjunction* AND (^) dan operator *disjunction* OR (V).

Perhatikan pohon keputusan pada gambar 7 berikut.



Gambar 7. Contoh pohon keputusan yang akan dibuat *rules* nya

Dari pohon keputusan pada gambar 7, maka *rules* yang dapat dibuat untuk mendapatkan atribut "Jawab 1", "Jawab 2", dan "Jawab 3" adalah:

R1: IF Atribut#1 = Instans2 ^ Atribut#2 = Instans21 THEN Jawab = Jawab1

R2: IF Atribut#1 = Instans2 ^ Atribut#2 = Instans22 THEN Jawab = Jawab2

R3: IF Atribut#1 = Instans3 V Atribut#1 = Instans4 THEN Jawab = Jawab3

Ketiga *rules* di atas (R1, R2 dan R3), R1 dan R2 sama-sama menggunakan operator

conjunction. Sedangkan R3 menggunakan operator *disjunction*.

Dari kasus pada Prediksi Hujan dari Data Cuaca Tabel 2 yang pohon keputusannya disajikan pada Gambar 5, maka dapat dibuat *rule-rule*nya sebagai berikut:

R1: IF Keadaan_Kemarin = Tdk_Hujan ^ Cuaca = Panas THEN Hujan = Tidak

R2: IF Keadaan_Kemarin = Tdk_Hujan ^ Cuaca = Mendung ^ Angin = Kecil THEN Hujan = Tidak

R3: IF Keadaan_Kemarin = Tdk_Hujan ^ Cuaca = Mendung ^ Angin = Besar THEN Hujan = Ya

R4: IF Keadaan_Kemarin = Tdk_Hujan ^ Cuaca = Cerah THEN Hujan = Tidak

R5: IF Keadaan_Kemarin = Hujan THEN Hujan = Ya

Ada lima buah *rules* yang dapat dibuat dari pohon keputusan yang telah dihasilkan.

SIMPULAN

Teknik penambangan data (*data mining*) merupakan salahsatu teknik untuk mencari suatu *knowledge* (pengetahuan) berdasarkan banyaknya data. Salah satu teknik yang diperlukan dalam penambangan data adalah penentuan pohon keputusan (*decision tree*). Dalam membuat pohon keputusan, diperlukan teknik penentuan atribut, agar atribut yang terpilih adalah atribut yang benar. Ada dua metode dalam menentukan atribut, yakni menggunakan indeks *gini*dan *information gain*.

Diantara kedua metode ini, metode yang lebih cepat untuk digunakan adalah dengan menggunakan indeks *gini*, karena nilai entropi untuk tiap atribut tidak perlu dicari lagi. Tetapi, pada dasarnya, kedua metode ini menghasilkan hasil yang sama. Walaupun telah banyak *software data mining* yang beredar di pasaran, tetapi perhitungan untuk memperoleh sebuah pohon keputusan tetap harus dipahami.

DAFTAR PUSTAKA

Jiawei, H., Kamber, M. 2001. Data Mining Concepts and Techniques, Morgan Kaufmann Publishers.

Jing, L. 2004. *Data Mining Applications in Higher Education*, www.spss.com/events/e_id_1471/Data Mining in Higher Education.pdf

Merceron, A., Yacef, K. 2005. *Educational Data Mining: a Case Study*, http://www.it.usyd.edu.au/~kalina/publis/merceron_yacef_aied05.pdf

--------. 2005. *TADA-Ed for Educational Data Mining*, http://imej.wfu.edu/articles/2005/1/03/printver.asp

Nilakant, K. 2004. *Application of Data Mining in Constraint Based Intelligent Tutoring System,* www.cosc.canterbury.ac.nz/research/reports/HonsReps/2004/hons_0408.pdf

Pramudiono, Iko. 2003. *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. http://www.ilmukomputer.com

Santosa, Budi. 2007. Data Mining: *Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.