

# Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5

Panji Bimo Nugroho Setio<sup>a,\*</sup>, Dewi Retno Sari Saputro<sup>b</sup>, Bowo Winarno<sup>c</sup>

<sup>a,b,c</sup>Program Studi Matematika FMIPA Universitas Sebelas Maret, Jl. Ir Sutami No.36 A, Surakarta 57126, Indonesia.

\* Alamat Surel: panjibimo24@gmail.com

## Abstrak

Data *mining* merupakan suatu proses menemukan pola dalam sejumlah data besar dengan tujuan untuk melakukan klasifikasi, estimasi, prediksi, asosiasi dan kluster. Salah satu metode klasifikasi yang populer adalah pohon keputusan (*decision tree*). Konsep dasar dari pohon keputusan adalah mengubah data menjadi sebuah model pohon keputusan, kemudian mengubah model pohon menjadi *rule* dan menyederhanakan *rule*. Data dalam pohon keputusan dinyatakan dalam bentuk tabel dengan atribut dan *record*. Terdapat beberapa cara untuk mengkonstruksikan pohon keputusan salah satunya menggunakan algoritme C4.5. Algoritme C4.5 merupakan pengembangan dari *Iterative Dichotomiser* (ID3). ID3 adalah algoritme pembelajaran pohon keputusan (*decision tree learning*) yang paling dasar. Algoritme ini melakukan pencarian secara menyeluruh (*greedy*) pada semua kemungkinan pohon keputusan. Algoritme ID3 membangun pohon keputusan secara *top-down* (dari atas ke bawah) yang diawali dengan suatu atribut. *Top-down* artinya pohon keputusan dibangun dari simpul akar ke daun. Algoritme ini tidak dapat mengatasi atribut dengan data kosong, oleh karena itu diperlukan algoritme C4.5 yang dapat mengatasi atribut dengan data kosong serta dapat melakukan pemangkasan cabang yang juga tidak terdapat pada algoritme ID3. Contoh dari algoritme C4.5 ditunjukkan pada Gambar 4.

## Kata kunci:

*Data mining*, klasifikasi, pohon keputusan, ID3, algoritme C4.5.

© 2020 Dipublikasikan oleh Jurusan Matematika, Universitas Negeri Semarang

## 1. Pendahuluan

Perkembangan data *mining* yang pesat tidak dapat lepas dari perkembangan teknologi. Data *mining* adalah suatu proses menemukan hubungan yang berarti, pola dan kecenderungan dengan memeriksa pada sekumpulan data dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Larose, 2005).

Salah satu teknik pengolahan data *mining* adalah klasifikasi. Klasifikasi adalah proses menemukan sekumpulan pola atau fungsi-fungsi yang mendeskripsikan dan memisahkan kelas data satu dengan yang lainnya, dan digunakan untuk memprediksi data yang belum memiliki kelas data tertentu (Han & Kamber, 2006). Pohon keputusan merupakan salah satu teknik yang dapat digunakan untuk melakukan klasifikasi terhadap sekumpulan objek. Menurut Tan *et al.* (2004), sebuah pohon keputusan mungkin dibangun dengan seksama secara manual, atau dapat tumbuh secara otomatis dengan menerapkan salah satu atau beberapa algoritme pohon keputusan untuk memodelkan himpunan data yang belum terklasifikasi.

Terdapat beberapa algoritme klasifikasi yang digunakan untuk mengkonstruksi pohon keputusan, yaitu algoritme *Classification and Regression Trees* (CART), algoritme *Iterative Dichotomiser* (ID3) dan algoritme C4.5 (Larose, 2005). Pada artikel ini penulis melakukan analisis beberapa pohon keputusan algoritme klasifikasi yang saat ini digunakan, termasuk ID3 dan C4.5.

To cite this article:

Setio, P.B.N., Saputro, D.R.S., & Winarno, B. (2020). Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5. *PRISMA, Prosiding Seminar Nasional Matematika* 3, 64-71

## 2. Pembahasan

### 2.1. Data mining

Data *mining* merupakan suatu proses menemukan pola dalam sejumlah data besar dengan tujuan untuk melakukan klasifikasi, estimasi, prediksi, asosiasi dan kluster (Han & Kamber, 2006). Data *mining* merupakan salah satu tahapan dalam keseluruhan proses *Knowledge Discovery in Database* (KDD). KDD adalah proses menentukan informasi yang berguna dalam data. Informasi ini terkandung dalam basis data yang berukuran besar (Han & Kamber, 2006). Menurut Larose (2005), data *mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu

- **Klasifikasi**  
 Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, sedang dan rendah.
- **Estimasi**  
 Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.
- **Prediksi**  
 Prediksi hampir sama dengan klasifikasi dan estimasi. Hanya saja dalam prediksi, nilai dari hasil akan ada di masa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat digunakan untuk prediksi.
- **Pengklasteran**  
 Pengklasteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Pengklasteran berbeda dengan klasifikasi karena tidak adanya variabel target. Pengklasteran tidak mengklasifikasi, mengestimasi, atau memprediksi nilai dari variabel target namun pengklasteran melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan.
- **Asosiasi**  
 Tugas asosiasi dalam data *mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. Atribut yang dimaksudkan dalam hal ini adalah faktor-faktor yang memengaruhi pengambilan keputusan.

### 2.2. Klasifikasi

Klasifikasi adalah salah satu teknik pada data *mining* yang memetakan data ke dalam kelompok atau kelas yang telah ditentukan. Klasifikasi merupakan metode *supervised learning* yang membutuhkan data *training* berlabel untuk menghasilkan sebuah aturan yang mengklasifikasikan data uji ke dalam kelompok atau kelas yang telah ditentukan (Dunham, 2003).

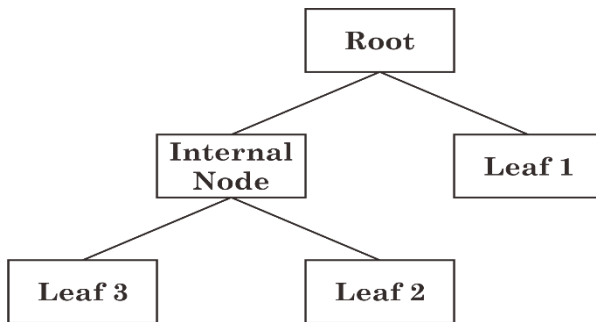
Beberapa teknik klasifikasi yang digunakan adalah *decision tree*, *rule-based classifier*, *neural-network*, *support machine* dan *naïve Bayes classifier*. Setiap teknik menggunakan algoritme pembelajaran untuk mengidentifikasi model yang memberikan hubungan yang paling sesuai antara himpunan atribut dan label kelas dari data input.

### 2.3. Pohon Keputusan (Decision Tree)

Pohon keputusan merupakan salah satu metode klasifikasi yang populer karena dapat dengan mudah diinterpretasi oleh manusia. 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 (Berry & Gordon, 2004).

Pohon keputusan memiliki *node* pohon yang merepresentasikan atribut yang telah diuji dan setiap cabangnya merupakan suatu pembagian hasil uji serta *node* daun (*leaf*) merepresentasikan kelompok kelas tertentu (Han & Kamber, 2006). Level *node* teratas dari sebuah pohon keputusan adalah *node* akar (*root*) yang biasanya berupa atribut yang memiliki pengaruh paling besar pada suatu kelas tertentu. Konsep dasar dari pohon keputusan adalah mengubah data menjadi model pohon keputusan, kemudian mengubah model pohon menjadi *rule* dan menyederhanakan *rule*. Data dalam pohon keputusan

dinyatakan dalam bentuk tabel dengan atribut dan *record*. Contoh pohon keputusan ditunjukkan pada Gambar 1.



**Gambar 1.** Contoh Pohon Keputusan

Cabang-cabang dari pohon keputusan merupakan pertanyaan klasifikasi sedangkan untuk daun-daunnya merupakan kelas-kelas atau kelompoknya. Tujuan dari algoritme C4.5 adalah untuk melakukan klasifikasi sehingga hasil dari pengolahan dataset berupa pengelompokan data ke dalam kelas-kelas tertentu. Pohon keputusan berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah variabel target. Dengan demikian, manfaat utama penggunaan pohon keputusan adalah kemampuannya untuk membuat proses pengambilan keputusan yang kompleks menjadi lebih sederhana sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan.

Berikut kelebihan metode pohon keputusan (Ratniasih, 2016).

- Daerah pengambilan keputusan yang sebelumnya kompleks dan sangat global, dapat diubah menjadi lebih simpel dan spesifik.
- Eliminasi perhitungan-perhitungan yang tidak diperlukan, karena ketika menggunakan metode pohon keputusan maka *sample* diuji hanya berdasarkan kriteria atau kelas tertentu.
- Fleksibel untuk memilih fitur dari *node* internal yang berbeda, fitur yang terpilih akan membedakan suatu kriteria dibandingkan kriteria yang lain dalam *node* yang sama. Kefleksibelan metode pohon keputusan ini meningkatkan kualitas keputusan yang dihasilkan jika dibandingkan ketika menggunakan metode penghitungan satu tahap yang lebih konvensional.
- Dalam analisis multivarian, dengan kriteria dan kelas yang jumlahnya sangat banyak, seorang penguji biasanya perlu mengestimasi baik itu distribusi dimensi tinggi ataupun parameter tertentu dari distribusi kelas tersebut. Metode pohon keputusan dapat menghindari munculnya permasalahan ini dengan menggunakan kriteria yang jumlahnya lebih sedikit pada setiap *node* internal tanpa banyak mengurangi kualitas keputusan yang dihasilkan.

Berikut adalah kekurangan pada pohon keputusan (Ratniasih, 2016).

- Terjadi *overlapping*, terutama ketika kelas-kelas dan kriteria yang digunakan jumlahnya sangat banyak. Hal tersebut juga dapat menyebabkan meningkatnya waktu pengambilan keputusan dan jumlah memori yang diperlukan.
- Pengakumulasian jumlah kesalahan dari setiap tingkat dalam sebuah pohon keputusan yang besar.
- Kesulitan dalam mendesain pohon keputusan yang optimal.
- Hasil kualitas keputusan yang didapatkan dari metode pohon keputusan sangat tergantung pada bagaimana pohon tersebut didesain.

#### 2.4. Iterative Dichotomiser (ID3)

*Iterative Dichotomiser* (ID3) adalah algoritme pembelajaran pohon keputusan (*decision tree learning*) yang paling dasar. ID3 pertama kali dikembangkan oleh J. Ross Quinlan. Algoritme ini melakukan pencarian secara menyeluruh (*greedy*) pada semua kemungkinan pohon keputusan. Algoritme ID3 membangun pohon keputusan secara *top-down* (dari atas ke bawah) yang diawali dengan suatu atribut. *Top-down* artinya pohon keputusan dibangun dari simpul akar ke daun (Wahyudin, 2009).

Algoritme ID3 menggunakan konsep dari *entropy* informasi dan pemilihan atribut menggunakan *information gain*. Secara ringkas, proses algoritme ID3 menurut Munawaroh (2015) diuraikan sebagai berikut.

- (1) Menentukan nilai dari *entropy* dengan rumus yang ditulis sebagai

$$Entropy(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

dengan  $S$  adalah data *sample* yang digunakan untuk *training*,  $P_+$  adalah jumlah yang bersolusi positif (mendukung) pada data *sample* untuk kriteria tertentu, dan  $P_-$  adalah jumlah yang bersolusi negatif (tidak mendukung) pada data *sample* untuk kriteria tertentu. *Entropy* merupakan jumlah bit yang dibutuhkan untuk menyatakan suatu kelas. Semakin kecil nilai *entropy*, semakin baik digunakan dalam mengekstraksi suatu kelas.

- (2) Setelah memperoleh nilai *entropy*, selanjutnya ditentukan nilai *information gain* dengan rumus yang ditulis sebagai

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{nilai}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

dengan  $A$  adalah atribut,  $v$  adalah suatu nilai yang mungkin untuk atribut  $A$ , *nilai* ( $A$ ) adalah himpunan yang mungkin untuk atribut  $A$ ,  $|S_v|$  adalah jumlah *sample* untuk nilai  $v$  dan  $|S|$  adalah jumlah seluruh *sample* data.

- (3) Memilih atribut yang memiliki nilai *information gain* terbesar.
- (4) Membentuk simpul yang berisikan atribut seperti yang dimaksudkan dalam (3).
- (5) Proses perhitungan *information gain* dilakukan secara berulang-ulang hingga semua data masuk ke dalam kelas yang sama. Atribut yang telah dipilih tidak diinputkan lagi pada perhitungan nilai *information gain*.

### 2.5. Algoritme C4.5

Algoritme C4.5 merupakan salah satu algoritme yang dapat digunakan untuk mengkonstruksi sebuah pohon keputusan. Algoritme C4.5 merupakan pengembangan algoritme ID3 (Quinlan, 1993), dimana kekurangan yang dimiliki algoritma ID3 ditutupi oleh algoritme C4.5. Empat hal yang membedakan algoritma C4.5 dengan ID3 antara lain: tahan (*robust*) terhadap data *noise*, mampu menangani variabel dengan tipe diskrit maupun kontinu, mampu menangani variabel yang memiliki *missing value*, dan dapat memangkas cabang dari pohon keputusan (Elisa, 2007).

Algoritme C4.5 mempunyai *input training sample* dan *samples*. *Training samples* merupakan contoh data yang digunakan untuk membangun pohon keputusan yang telah diuji kebenarannya. Sedangkan *samples* merupakan *field-field* data yang akan digunakan sebagai parameter dalam melakukan klasifikasi data.

Berikut ini adalah dasar algoritma C4.5 untuk proses pembentukan *decision tree* (Han & Kamber, 2006).

*Input* : *Training samples, Attribute*

*Output* : *Decision tree*

**Generate\_decision\_tree** (*Training samples, Attribute*) // *decision tree function*

*Method* :

- (1) **Create node**  $N$ ;
- (2) **If** *samples are all of the same class C then*
- (3) **Return**  $N$  as a leaf node labeled with the class  $C$ ;
- (4) **if** *atribute-list is empty then*
- (5) **Return**  $N$  as a leaf node labeled with the most common class in samples; // majority voting
- (6) **else**
- (7) *select test-attribute, attribute among attribute-list with the highest information gain;*
- (8) *label node N with test-attribute;*
- (9) **for** *each known value  $a_i$  of test-attribute // partition the samples*
- (10) *grow a branch from node N for the condition test-attribute =  $a_i$ ;*
- (11) *let  $s_i$  be the set of samples in samples for which test-attribute =  $a_i$ ; // a partition*
- (12) **if**  $s_i$  is empty **then**
- (13) *attach a leaf labeled with the most common class in samples;*  
**else** *attach the node returned by Generate\_decision\_tree( $s_i$ , attribute-list-test-attribute);*

Menurut Lakshmi *et al.* (2013), tahapan dari proses algoritme C4.5 diuraikan sebagai berikut.

- (1) Mempersiapkan data *training*.  
 (2) Menghitung nilai *entropy*. *Entropy* merupakan ukuran ketidakpastian, yakni perbedaan keputusan terhadap nilai atribut tertentu. Semakin tinggi nilai *entropy*, semakin tinggi perbedaan keputusan (ketidakpastian). Nilai *Entropy* dihitung dengan rumus yang ditulis sebagai

$$Entropy = -\sum_{i=1}^k p_i \times \log_2 p_i \quad (1)$$

dengan  $S$  adalah himpunan kasus,  $p_i$  adalah probabilitas yang diperoleh dari *sum* (ya) dibagi dengan total kasus.

- (3) Menghitung nilai *gain*. *Gain* merupakan salah satu langkah pemilihan atribut yang digunakan untuk memilih tes atribut setiap simpul pada pohon keputusan atau dengan kata lain *gain* merupakan tingkat pengaruh suatu atribut terhadap keputusan atau ukuran efektifitas suatu variabel dalam mengklasifikasikan data. *Gain* dihitung dengan rumus yang ditulis sebagai

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^k \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (2)$$

dengan  $S$  adalah himpunan kasus,  $A$  adalah atribut,  $|S_i|$  adalah jumlah kasus pada partisi ke  $i$ , dan  $|S|$  adalah jumlah kasus dalam  $S$ . Pada algoritma C4.5, nilai *gain* digunakan untuk menentukan variabel mana yang menjadi *node* dari suatu pohon keputusan. Suatu variabel yang memiliki *gain* tertinggi akan dijadikan *node* di pohon keputusan.

- (4) Menghitung nilai *split info* dengan rumus

$$SplitInfo(S, A) = -\sum_{j=1}^k \frac{S_j}{S} \times \log_2 \frac{S_j}{S} \quad (3)$$

dengan  $S$  adalah ruang *sample*,  $A$  adalah atribut, dan  $S_j$  adalah jumlah *sample* untuk atribut ke  $j$ .

- (5) Menentukan nilai *gain ratio* dengan rumus yang ditulis sebagai

$$GainRatio(S, A) = \frac{Gain(S, A)}{Split(S, A)} \quad (4)$$

dengan  $Gain(S, A)$  adalah *information gain* pada atribut  $(S, A)$ ,  $A$  adalah atribut, dan  $Split(S, A)$  adalah *split information* pada atribut  $(S, A)$ .

Nilai *gain ratio* tertinggi akan digunakan sebagai atribut akar. Dengan demikian akan terbentuk pohon keputusan sebagai *node* 1.

- (6) Mengulangi proses ke-2 hingga semua cabang memiliki kelas yang sama. Proses percabangan akan berhenti apabila
- semua kasus dalam simpul  $n$  mendapat kelas yang sama;
  - tidak ada variabel independen di dalam kasus yang dipartisi lagi;
  - tidak ada kasus di dalam cabang yang kosong.

Algoritme C4.5 memiliki kompleksitas waktu  $O(m \cdot n^2)$ , dengan  $m$  ukuran data pelatihan dan  $n$  adalah banyak atribut. Pembahasan tentang kompleksitas ini ditulis dalam artikel yang lain.

Berikut diberikan contoh dari algoritme C4.5 yaitu menentukan keputusan bermain bola dengan memerhatikan dua atribut yaitu suhu dan cuaca. Data atribut ditunjukkan pada Tabel 1.

**Tabel 1.** Atribut untuk menentukan keputusan bermain bola.

No.	Cuaca	Suhu	Bermain/Tidak
1	Cerah	Panas	Tidak
2	Cerah	Panas	Tidak
3	Berawan	Panas	Ya
4	Hujan	Sejuk	Ya
5	Hujan	Dingin	Ya
6	Hujan	Dingin	Ya
7	Berawan	Dingin	Tidak
8	Cerah	Sejuk	Ya
9	Cerah	Dingin	Ya
10	Hujan	Sejuk	Ya

Berdasarkan Tabel 1., dapat dikonstruksi pohon keputusan menggunakan algoritme C4.5 dengan terlebih dahulu menentukan simpul akar. Untuk menentukan akar, dapat menggunakan rumus seperti pada persamaan (1) kemudian menghitung *gain* menggunakan rumus pada persamaan (2) lalu

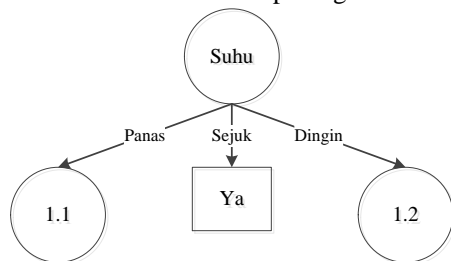
menghitung *split info* menggunakan rumus pada persamaan (3) dan terakhir menghitung *gain ratio* menggunakan rumus pada persamaan (4). Perhitungan ditunjukkan pada Tabel 2.

**Tabel 2.** Perhitungan *node* akar

Node		Jumlah Kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
1	Total	10	3	7	0.8812909	
	Cuaca					0.281290899
	Cerah	4	2	2	1	
	Berawan	2	1	1	1	
	Hujan	4	0	4	0	
	Suhu					0.281290899
	Panas	3	2	1	0.9182958	
	Sejuk	3	0	3	0	
	Dingin	4	1	3	0.8112781	

Berdasarkan perhitungan pada Tabel 2., dipilih atribut yang memiliki nilai *gain* tertinggi untuk dijadikan sebagai akar. Pada Tabel 2. terlihat kedua atribut memiliki nilai *gain* yang sama, oleh karena itu dipilih salah satu dari atribut tersebut. Pada artikel ini atribut yang dipilih yaitu suhu sebagai akar. Terdapat tiga nilai dari atribut suhu, yaitu panas, sejuk dan dingin. Dari ketiga nilai tersebut, nilai atribut sejuk sudah mengklasifikasikan kasus menjadi 1, yaitu keputusannya tidak, sehingga tidak perlu dilakukan perhitungan lagi. Untuk nilai atribut panas dan dingin masih perlu dilakukan perhitungan lagi.

Dari hasil tersebut dapat digambarkan pohon keputusan sementara seperti pada Gambar 2.



**Gambar 2.** Pohon keputusan hasil perhitungan *node* 1

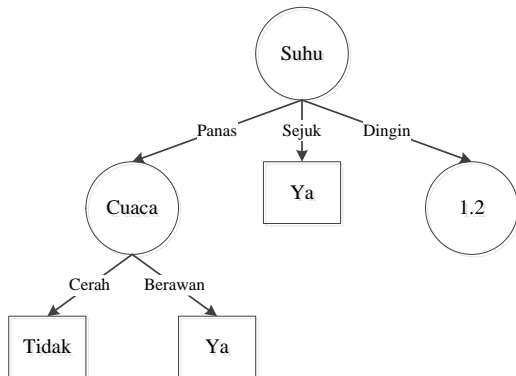
Selanjutnya menghitung nilai dari atribut cuaca untuk dijadikan node percabangan dari nilai atribut panas. Perhitungan dilakukan dengan cara yang sama seperti pada Tabel 1. dan hasil perhitungan dapat dilihat pada Tabel 3.

**Tabel 3.** Perhitungan *node* 1.1

Node		Jumlah Kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
1.1	Suhu (Panas)	3	2	1	0.9182958	
	Cuaca					0.9182958
	Cerah	2	2	0	0	
	Berawan	1	0	1	0	
	Hujan	0	0	0	0	

Berdasarkan perhitungan pada Tabel 3., dipilih atribut cuaca sebagai *node* percabangan dari nilai atribut panas. Pada Tabel 3. terlihat ketiga nilai tersebut, nilai atribut cerah sudah mengklasifikasikan kasus menjadi 1, yaitu keputusannya tidak. Kemudian nilai atribut berawan sudah mengklasifikasikan kasus menjadi 1, yaitu keputusannya ya sehingga tidak perlu dilakukan perhitungan lagi untuk atribut ini.

Dari hasil tersebut dapat digambarkan pohon keputusan sementara seperti pada Gambar 3.



**Gambar 3.** Pohon keputusan hasil perhitungan *node* 1.1.

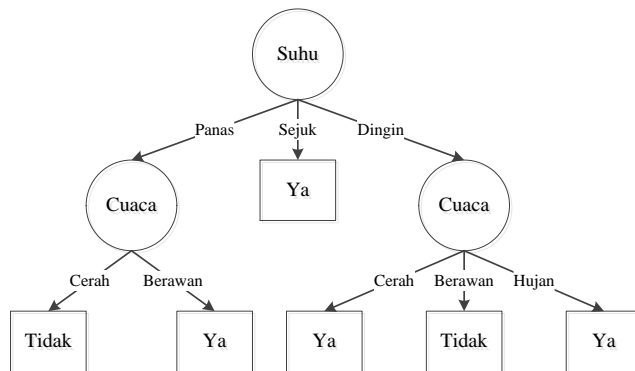
Selanjutnya menghitung nilai dari atribut cuaca untuk dijadikan *node* percabangan dari nilai atribut dingin. Perhitungan dilakukan dengan cara yang sama seperti pada Tabel 1. dan hasil perhitungan dapat dilihat pada Tabel 4.

**Tabel 4.** Perhitungan *node* 1.2

Node		Jumlah Kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
1.2	Suhu (Dingin)	4	1	3	0.8112781	
	Cuaca					0.8112781
	Cerah	1	0	1	0	
	Berawan	1	1	0	0	
	Hujan	2	0	2	0	

Berdasarkan perhitungan pada Tabel 4., dipilih atribut cuaca sebagai *node* percabangan dari nilai atribut dingin. Pada Tabel 4. terlihat ketiga nilai tersebut, nilai atribut cerah sudah mengklasifikasikan kasus menjadi 1, yaitu keputusannya ya. Kemudian nilai atribut berawan sudah mengklasifikasikan kasus menjadi 1, yaitu keputusannya tidak dan nilai atribut hujan sudah mengklasifikasikan kasus menjadi 1, yaitu keputusannya ya sehingga tidak perlu dilakukan perhitungan lagi untuk atribut ini.

Dari hasil tersebut dapat digambarkan pohon keputusan sementara seperti pada Gambar 4.



**Gambar 4.** Pohon keputusan hasil perhitungan *node* 1.2.

Dengan memperhatikan pohon keputusan pada Gambar 4., diketahui bahwa semua kasus sudah masuk dalam kelas. Dengan demikian, pohon keputusan pada Gambar 4. merupakan pohon keputusan terakhir yang terbentuk.

### 3. Simpulan

Berdasarkan pembahasan, disimpulkan bahwa algoritme C4.5 mengkonstruksi ulang pohon keputusan (*decision tree*) pada algoritme ID3 dengan memperbaiki skema pada ID3 yaitu atribut dengan data kosong dan pemangkasan cabang. Contoh dari algoritme C4.5 ditunjukkan pada Gambar 4.

### Daftar Pustaka

- Berry, Michael J.A. & Gordon S. Linoff. (2004). *Data Mining Techniques for Marketing, Sales and Customer Relationship Management, Second Edition*, Wiley Publishing, Inc.
- Dunham, M.H. (2003). *Data Mining: Introductory and Advanced Topics*, Pearson Education Inc.
- Elisa, Erlin. (2007). Analisa dan Penerapan Algoritma C4.5 Dalam *Data Mining* untuk Mengidentifikasi Faktor-Faktor Penyebab Kecelakaan Kerja Kontruksi PT. Arupadhatu Adisesantani, *Jurnal Sistem Informasi, Batam*.
- Han, J. & M. Kamber. (2006). *Data Mining Concept and Techniques*, Morgan Kaufmann Publishers, San Fransisco.
- Lakshmi, T.M., A. Martin, R.M. Begum, and Dr.V.P. Venkatesan. (2013). An Analysis on Performance of Decision Tree Algorithms using Student's Qualitative Data, *I.J.Modern Education and Computer Science*.
- Larose, Daniel T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*, John Willey and Sons, Inc.
- Munawaroh, Holisatul., Khusnul, Bain K., Kustiyahningsih, Yeni. (2015). Perbandingan Algoritma ID3 dan C5.0 Dalam Identifikasi Penjurusan Siswa SMA. *Program Studi Teknk Informatika, Universitas Tunojoyo, Bangkalan*.
- Quinlan, J. R., (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.
- Ratniasih, N. L. (2016). Konversi Data Training Tentang Pemilihan Kelas Menjadi Bentuk Pohon Keputusan dengan Teknik Klasifikasi. *Jurnal Eksplora Informatika*, 4(2), 145-154.
- Tan, Pang Ning., Michael Steinbach, Vipin Kumar (2004). *Introduction to Data Mining*.
- Wahyudin (2009). Metode Iterative Dichotomizer 3 (ID3) Untuk Penyeleksian Penerimaan Mahasiswa Baru, *Jurnal Pendidikan Teknologi Informasi dan Komunikasi (PTIK)*, Universitas Pendidikan Indonesia.