

GENERATOR POHON KEPUTUSAN DENGAN MENERAPKAN ALGORITMA C4.5 UNTUK PROGRAM KONSULTASI

Irma Kharis
irma.kharis@gmail.com

Rosa Delima, S.Kom., M.Kom
rosadelima@ukdw.ac.id

Joko Purwadi, M.Kom
jokop@ukdw.ac.id

Abstract

C4.5 algorithm is used to simplify decision tree from decision table by generating decision tree from an existing decision tree. With this algorithm, knowledge base in the decision table can be simplified. This research will build a consultation program using C4.5 algorithm which is called decision tree generator.

Decision tree generator provides inference facility and a user interface for consultation. The user is required to build a knowledge base first, and the application will generate user interface automatically. There are two steps in decision tree generator: firstly, the application will build the decision tree, and after that the application will build the user interface for consultation session. The results of this research show that the decision tree generator can get goal and advice from tree exploration in consultation session.

Keywords : C4.5 algorithm, decision tree, consultation program.

1. Pendahuluan

Algoritma C4.5 merupakan salah satu cara yang dapat dilakukan untuk membangun aturan yang direpresentasikan dalam bentuk pohon keputusan. Fakta yang digunakan berasal dari tabel keputusan yang kemudian diproses dengan Algoritma C4.5, sehingga menghasilkan pohon keputusan. Algoritma ini juga dapat menyederhanakan pengetahuan yang dimiliki sistem sehingga proses inferensi dapat menjadi lebih cepat. Aplikasi ini dapat digunakan untuk membangun sebuah program konsultasi berbasis sistem cerdas secara otomatis. Pengguna hanya perlu memasukkan pengetahuan. Dengan algoritma C4.5, sistem akan langsung membentuk pohon keputusan dan basis pengetahuan berbentuk aturan. Pengetahuan yang dihasilkan merupakan pengetahuan yang telah disederhanakan.

Generator pohon keputusan ini dibuat secara visual menggunakan tabel keputusan agar memudahkan pengguna memasukkan pengetahuan bagi sistem. Dengan aplikasi ini pengguna dapat dengan mudah mengembangkan sistem konsultasi dengan waktu dan biaya yang murah.

2. Landasan Teori

2.1. Representasi Pengetahuan Berbasis Aturan

Menurut Martin dan Oxman, (Martin & Oxman, 1988), *knowledge base* adalah bagian dari sebuah sistem pakar yang berisi pengetahuan tentang spesifik domain masalah. Domain pengetahuan dapat direpresentasikan dalam berbagai cara, salah satunya adalah aturan. Skema representasi pengetahuan berbasis aturan merupakan skema yang paling populer. Sistem terkadang menyebutnya dengan *production systems*. Aturan atau produksi digunakan untuk merepresentasikan hubungan pasangan kondisi-aksi :

IF (kondisi) THEN (aksi)

Bagian kondisi dalam aturan disebut juga sebagai premis, *antecedent*, atau *left-hand side* (LHS). Bagian aksi dalam aturan disebut juga sebagai konklusi, *consequent*, atau *right-hand side* (RHS). Terdapat contoh aturan mengenai jenis kendaraan yang digunakan berdasarkan kondisi jarak dan cuaca untuk menggambarkan penggunaan aturan, yaitu :

- Aturan 1 : IF distance is 1 mile
AND weather is sunny
THEN transportation is bicycle.
- Aturan 2 : IF distance is 3 miles
AND weather is sunny
THEN transportation is bicycle.
- Aturan 3 : IF distance is 20 miles
AND weather is sunny
OR weather is rainy
THEN transportation is automobile.

2.2 Algoritma C4.5

Menurut Kusriani dan Luthfi, (Kusriani &Luthfi, 2009), Algoritma C4.5 merupakan algoritma yang digunakan untuk membentuk 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. Aturan juga dapat diekspresikan dalam bentuk bahasa basis data seperti *Structured Query Language* (SQL) untuk mencari record pada kategori tertentu.

Pohon keputusan juga berguna untuk mengeksplorasi data dan pemodelan. Sebuah model pohon keputusan terdiri dari sekumpulan aturan untuk membagi sejumlah populasi yang heterogen menjadi lebih kecil, lebih homogen dengan memperhatikan pada variabel tujuannya.

Sebuah pohon keputusan mungkin dibangun dengan saksama secara manual atau dapat tumbuh secara otomatis dengan menerapkan salah satu atau beberapa algoritma pohon keputusan untuk memodelkan himpunan data yang belum terklasifikasi.

Variabel tujuan biasanya dikelompokkan dengan pasti dan model pohon keputusan lebih mengarah pada perhitungan probabilitas dari tiap-tiap record terhadap kategori-kategori tersebut atau untuk mengklasifikasi record dengan mengelompokkan dalam satu kelas. Pohon keputusan juga dapat digunakan untuk mengestimasi nilai dari variabel *continue* meskipun ada beberapa teknik yang lebih sesuai untuk kasus tersebut. Salah satu algoritma yang dapat dipakai dalam pembentukan pohon keputusan adalah Algoritma C4.5.

Secara umum, Algoritma C4.5 terdiri dari 2 tahapan utama yaitu pemilihan atribut sebagai akar dari pohon keputusan, selanjutnya dibuat cabang-cabang dari pohon keputusan berdasarkan pengetahuan yang dimasukkan pengguna.

Dasar untuk memilih atribut sebagai akar dari pohon adalah dengan menghitung nilai gain tertinggi dari atribut-tribut yang ada. Rumus yang digunakan untuk menghitung gain tertera dalam Persamaan 1.

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

Keterangan :

- S : himpunan kasus
- A : atribut
- n : jumlah partisi atribut A
- |S_i| : jumlah kasus pada partisi ke-i
- |S| : jumlah kasus dalam S

Perhitungan dari Persamaan 2.1 akan memperoleh nilai gain dari atribut yang paling tertinggi. Gain adalah salah satu *attribute selection measure* yang digunakan untuk memilih test atribut tiap *node* pada pohon. Atribut dengan *information gain* tertinggi dipilih sebagai test atribut dari suatu *node*. Sementara itu, penghitungan nilai entropi dapat dilihat pada Persamaan 2.

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

[2]

Keterangan :

- S : himpunan kasus
- A : Atribut
- n : jumlah partisi S
- P_i : proporsi dari S_i terhadap S

3. Hasil dan Pembahasan

3.1. Pembangunan Pohon Keputusan

Pembangunan pohon keputusan pada generator pohon keputusan ini menggunakan algoritma C4.5 dengan tujuan agar *rule* yang terbentuk dapat lebih sederhana, sehingga saat konsultasi sistem dapat mengambil kesimpulan dengan lebih cepat. Untuk membangun pohon keputusan dan basis pengetahuan, pengguna harus memasukan pengetahuan awal bagi sistem. Contoh masukan pengguna dapat dilihat pada Gambar 1. Program konsultasi pada Gambar 1 adalah program konsultasi untuk Sistem Pengambilan Keputusan Makan di Restaurant. Tabel keputusan pada awalnya telah mengandung tujuh *rule*. Ketujuh *rule* tersebut adalah :

- 1) Rule 1
IF Alt = T AND Sat = F AND Hun = T AND Pat = Some AND Price = \$\$\$ AND Rain = F AND Res = T AND Est = 0-10 THEN Goal = T
- 2) Rule 2
IF Alt = T AND Sat = F AND Hun = T AND Pat = Full AND Price = \$ AND Rain = F AND Res = F AND Est = 30-60 THEN Goal = F
- 3) Rule 3
IF Alt = F AND Sat = F AND Hun = F AND Pat = Some AND Price = \$ AND Rain = F AND Res = F AND Est = 0-10 THEN Goal = T
- 4) Rule 4
IF Alt = T AND Sat = T AND Hun = T AND Pat = Full AND Price = \$ AND Rain = F AND Res = F AND Est = 10-30 THEN Goal = T

- 5) Rule 5
IF Alt = T AND Sat = T AND Hun = F AND Pat = Full AND Price = \$\$\$ AND Rain = T AND Res = T AND Est = >60 THEN Goal = F
- 6) Rule 6
IF Alt = F AND Sat = F AND Hun = T AND Pat = Some AND Price = \$\$ AND Rain = T AND Res = T AND Est = 0-10 THEN Goal = T
- 7) Rule 7
IF Alt = F AND Sat = F AND Hun = F AND Pat = None AND Price = \$\$ AND Rain = T AND Res = F AND Est = 0-10 THEN Goal = F



Gambar 1
Implementasi Halaman Tabel Keputusan

Tabel keputusan tersebut kemudian dimasukkan ke dalam generator pohon keputusan dan diproses dengan Algoritma C4.5 untuk mendapatkan pohon keputusan dan pengetahuan yang lebih sederhana. Sistem akan melakukan perhitungan untuk menentukan akar, cabang dan anak dari pohon keputusan sampai semua *leaf* pada pohon merupakan *goal* dari sistem. Berikut ini adalah tahapan yang dilakukan oleh sistem :

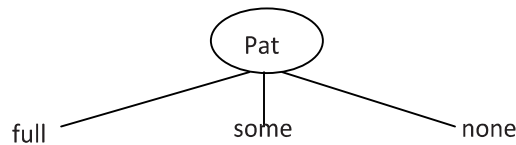
Langkah 1 :

Menghitung jumlah kasus, yaitu jumlah kasus untuk goal true, jumlah kasus untuk keputusan false, dan entropy dari semua kasus. Kasus terbagi berdasarkan atribut alt, sat, hun, pat, price, rain, res, dan est. Setelah itu, sistem menghitung gain untuk setiap atribut. Hasil perhitungan ditunjukkan oleh Tabel 1.

Tabel 1
Perhitungan Node 1

Entropy (s) = 0.98522813603425	
Atribut	Gain
Gain(s,Alt)	0.020244207153756
Gain(s,Est)	0.52164063634332
Gain(s,Hun)	0.12808527889139
Gain(s,Pat)	0.59167277858233
Gain(s,Price)	0.020244207153756
Gain(s,Rain)	0.12808527889139
Gain(s,Res)	0.020244207153756
Gain(s,Sat)	0.0059777114237741

Hasil pada Tabel 1 menunjukkan bahwa atribut dengan gain tertinggi adalah pat. Dengan demikian, pat dapat menjadi node akar. Ada tiga nilai atribut dari pat, yaitu full, some, dan none. Dari ketiga nilai tersebut, nilai atribut none sudah mengklasifikasikan kasus menjadi 1, yaitu goal false sehingga tidak perlu dilakukan perhitungan lebih lanjut. Dari ketiga nilai tersebut, nilai atribut some sudah mengklasifikasikan kasus menjadi 1, yaitu goal true sehingga tidak perlu dilakukan perhitungan lebih lanjut. Tetapi, untuk nilai atribut full masih perlu dilakukan perhitungan lagi. Pohon keputusan yang dihasilkan dari perhitungan pada Tabel 1 ditunjukkan oleh Gambar 2.



Gambar 2
Pohon Keputusan Hasil Perhitungan Node 1

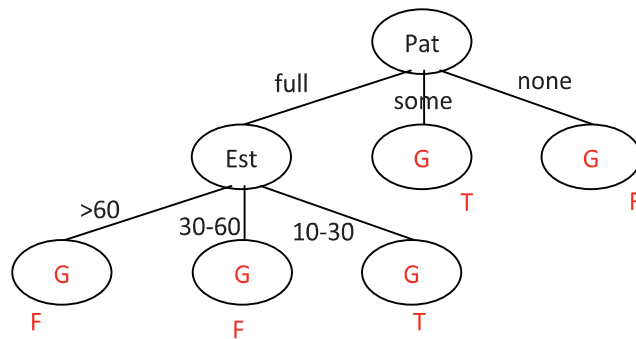
Langkah 2 :

Menghitung jumlah kasus, yaitu jumlah kasus untuk goal true, jumlah kasus untuk keputusan false, dan entropy dari semua kasus. Kasus terbagi berdasarkan atribut alt, sat, hun, price, rain, res, dan est yang dapat menjadi node anak dari nilai atribut pat. Setelah itu, sistem menghitung gain untuk setiap atribut. Hasil perhitungan ditunjukkan oleh Tabel 2.

Tabel 2.
Perhitungan Node 1.1

Entropy (s) = 0.91829583405449	
Atribut	Gain
Gain(s,Alt)	0
Gain(s,Est)	0.91829583405449
Gain(s,Hun)	0.25162916738782
Gain(s,Price)	0.25162916738782
Gain(s,Rain)	0.25162916738782
Gain(s,Res)	0.25162916738782
Gain(s,Sat)	0.25162916738782

Hasil pada Tabel 2 menunjukkan bahwa atribut dengan gain tertinggi adalah est. Dengan demikian, est dapat menjadi node akar. Ada tiga nilai atribut dari est yang memenuhi kasus, yaitu 10-30, 30-60, dan >60. Dari ketiga nilai tersebut, nilai atribut 10-30 sudah mengklasifikasikan kasus menjadi 1, yaitu goal true sehingga tidak perlu dilakukan perhitungan lebih lanjut. Dari ketiga nilai tersebut, nilai atribut 30-60 sudah mengklasifikasikan kasus menjadi 1, yaitu goal false sehingga tidak perlu dilakukan perhitungan lebih lanjut. Dari ketiga nilai tersebut, nilai atribut >60 sudah mengklasifikasikan kasus menjadi 1, yaitu goal false sehingga tidak perlu dilakukan perhitungan lebih lanjut. Pohon keputusan yang dihasilkan dari perhitungan pada Tabel 2 ditunjukkan oleh Gambar 3.



Gambar 3
Pohon Keputusan Hasil Perhitungan Node 1.1

Gambar 3 menunjukkan pohon keputusan yang sudah mengkategorikan semua kasus dalam kelas. Dengan demikian, pohon keputusan pada Gambar 3 merupakan pohon keputusan terakhir yang terbentuk. Berdasarkan pohon keputusan yang terbentuk, dibangun basis pengetahuan berbentuk aturan. Sistem ini menghasilkan 5 aturan utama yang dapat dilihat pada tabel 3.

Tabel 3
Urutan Penelusuran Pohon Keputusan

Rule
IF Pat = None THEN goal = F
IF Pat = Some THEN goal = T
IF Est = 10-30 AND Pat = Full THEN goal = T
IF Est = 30-60 AND Pat = Full THEN goal = F
IF Est = >60 AND Pat = Full THEN goal = F

3.2. Konsultasi

Bagian konsultasi berfungsi untuk sesi konsultasi pada sistem berdasarkan pohon keputusan yang telah dibangun. Proses penalaran akan dimulai dari *node* akar menuju *node* daun yang merupakan *goal* dari aturan yang direpresentasikan melalui pohon keputusan. Proses pengumpulan fakta dilakukan melalui serangkaian pertanyaan yang akan diajukan kepada pengguna ketika sesi konsultasi berlangsung, sehingga sistem perlu berinteraksi dengan pengguna. Proses penalaran juga dilakukan seiring dengan setiap pertanyaan yang dijawab oleh pengguna, sehingga penelusuran pohon dilakukan bersamaan dengan pertanyaan yang ditampilkan kepada pengguna. Keluaran dari generator pohon keputusan adalah program konsultasi yang akan dapat memberikan kesimpulan dan penjelasan berdasarkan fakta-fakta yang diberikan selama proses konsultasi. Proses penelusuran (inferensi) dilakukan mulai noda akar sampai noda daun terdalam yang merupakan keputusan sistem. Halaman konsultasi ditunjukkan pada Gambar 4.



Implementasi Halaman Konsultasi

3.3. Analisis Sistem

Analisa dilakukan untuk mengetahui seberapa cepat kinerja sistem dalam melakukan proses Algoritma C4.5 dan proses inferensi. Ada dua faktor utama yang mempengaruhi waktu pemrosesan, yaitu jumlah rule dan jumlah kondisi. Kedua faktor tersebut memiliki pengaruh yang sama dalam proses Algoritma C4.5 dan proses inferensi. Oleh karena itu, dilakukan suatu uji coba untuk mengetahui faktor mana yang lebih berpengaruh terhadap waktu proses sistem.

Setiap faktor yang telah diuji waktu pemrosesannya, kemudian dibandingkan satu sama lain untuk memperoleh pengaruh yang lebih besar. Perbandingan dilakukan untuk setiap uji coba dengan membuat tabel perbandingan seperti pada Tabel 4. Perbandingan antara kedua faktor dilakukan berbanding terbalik. Penjelasannya, uji coba 1 membandingkan antara faktor *rule* (42 kondisi dan 10 *rule*) dengan faktor kondisi (10 kondisi dengan 40 *rule*). Uji coba 2 membandingkan antara faktor *rule* (42 kondisi dan 20 *rule*) dengan faktor kondisi (20 kondisi dan 40 *rule*). Uji coba 3 membandingkan antara faktor *rule* (42 kondisi dan 30 *rule*) dengan faktor kondisi (30 kondisi dan 40 *rule*). Uji coba 4 membandingkan antara faktor *rule* (42 kondisi dan 40 *rule*) dengan faktor kondisi (42 kondisi dan 40 *rule*).

Tabel 4

Perbandingan Antara Faktor Jumlah Rule dan Faktor Jumlah Kondisi

UjiCoba	Generate Tabel (detik)		Generate Pohon (detik)		Inferensi (detik)	
	Faktor Rule	FaktorKondisi	Faktor Rule	FaktorKondisi	Faktor Rule	FaktorKondisi
1	7,77	24,35	0,44	0,71	0,21	0,27
2	19,27	32,89	0,65	0,92	0,44	0,63
3	34,60	37,93	0,88	1,01	0,68	0,71
4	49,35	49,35	1,31	1,31	0,90	0,90

Tabel 4 menunjukkan hasil perbandingan waktu antara faktor jumlah *rule* dan faktor jumlah kondisi. Diperoleh hasil bahwa kedua faktor memiliki pengaruh terhadap waktu pemrosesan sistem pakar, namun pengaruh yang lebih signifikan dimiliki oleh faktor jumlah rule. Hal ini terbukti dari jarak/*range* antara satu level uji coba dengan uji coba level selanjutnya. Jarak waktu yang dibutuhkan lebih besar pada faktor jumlah *rule* dibandingkan faktor jumlah kondisi. Semakin sedikit jumlah *rule*, maka waktu pemrosesan akan semakin cepat.

Penggunaan dan pengujian generator pohon keputusan juga dilakukan dengan menjalankan system pada 30 responden. Responden juga mengamati tampilan aplikasi serta beberapa hal mengenai materi yang ditampilkan tersebut. Setelah itu, responden diminta untuk mengisi kuesioner mengenai hasil pengujian dan penilaian mereka terhadap generator pohon keputusan, dimana ada beberapa pertanyaan menurut klasifikasinya yaitu dalam hal tampilan dan desain sistem, kemampuan sistem, serta pertanyaan uraian mengenai generator pohon keputusan.

Setiap responden diharuskan menguji generator pohon keputusan dengan membuat sebuah program konsultasi sebelum mengisi kuesioner yang ada. Hal tersebut dilakukan supaya kuesioner yang diisi oleh responden sesuai dengan penilaian responden yang sebenarnya, sehingga hasil pengujian yang diperoleh sesuai dengan harapan.

Aplikasi Generator Pohon Keputusan dinilai memuaskan dengan hasil lebih dari 85% responden menyatakan aplikasi memiliki antarmuka yang baik.

4. Kesimpulan

Penelitian mengenai generator pohon keputusan dengan menerapkan Algoritma C4.5 untuk program konsultasi menghasilkan beberapa kesimpulan, yaitu :

- a. Generator pohon keputusan yang dibangun mampu membangun basis pengetahuan dan melakukan inferensi untuk berbagai sistem konsultasi.
- b. Semakin banyak pengetahuan/rule yang diberikan ke dalam tabel keputusan, maka pohon yang dibangun akan semakin kompleks dan hasil kesimpulan yang diperoleh akan semakin akurat.
- c. Analisis waktu pemrosesan untuk pembangunan pohon keputusan dipengaruhi oleh factor jumlah *rule* dan jumlah kondisi. Semakin banyak *rule* dan kondisi dalam table keputusan, maka proses pembangunan pohon keputusan akan semakin lama. Hasil analisis dan pengujian system menunjukkan bahwa factor jumlah *rule* memiliki pengaruh waktu proses yang lebih besar dibandingkan factor jumlah kondisi.
- d. Hasil pengujian 30 responden mengenai tampilan dan desain sistem adalah 36.7% menyatakan sangat baik, 50% menyatakan baik, 11.4% menyatakan cukup, dan 1.9% menyatakan kurang, sehingga dapat disimpulkan bahwa generator pohon keputusan yang dibangun memiliki tampilan dan desain sistem yang baik.
- e. Hasil pengujian 30 responden mengenai kemampuan sistem dalam melakukan konsultasi dengan berbagai kasus sangat baik dan mereka menyatakan bahwa generator pohon keputusan dapat digunakan untuk membangun program konsultasi serta menghasilkan kesimpulan dan saran yang tepat.

5. Saran

Generator pohon keputusan secara visual untuk membuat program konsultasi masih memungkinkan untuk dikembangkan menjadi lebih baik. Berikut beberapa saran untuk pengembangan sistem:

- a. Sistem dapat menambahkan kemampuan *learning* terhadap kasus-kasus yang pernah ada, sehingga sistem mampu mengembangkan basis pengetahuan tanpa perlu melakukan proses perubahan basis pengetahuan oleh pengguna.
- b. Pertanyaan pada sesi konsultasi dapat disajikan dalam bentuk gambar, supaya dapat memenuhi mengakomodasi program konsultasi berbasis gambar.
- c. Bantuan penggunaan bisa disajikan dalam bentuk demo gerak/video.
- d. Mengembangkan generator pohon keputusan agar dapat digunakan pada perangkat *mobile*, seperti *smartphone* dan *tablet* sehingga Generator Pohon Keputusan dapat diakses dimanapun dan kapanpun.

Daftar Pustaka

- Durkin, J. (1994). *Expert Systems Design and Development*. London: Prentice Hall International, Inc.
- Giarratano, J. C. & Riley, G. D. (2005). *Expert Systems Principles and Programming* (4th ed.). Boston, Massachusetts: Course Technology.
- Kusrini & Luthfi, E.T. (2009). *Algoritma Data Mining*. Yogyakarta: Penerbit Andi Offset.

Martin, J. & Oxman, S. (1988). *Building Expert Systems a Tutorial*. New Jersey: Prentice Hall Englwood Cliffs.

Turban, E. (1992). *Expert System and Applied Artificial Intelligence*. New York: Macmillan Publishing Company Inc.

Turban, E., Aronson, J.A., & Liang.T.P. (2005). *Decision Support Systems and Intelligent Systems* (Trans.). Yogyakarta: Penerbit Andi Offset. (Original work published 1998).