PREDIKSI KELULUSAN MAHASISWA MENGGUNAKAN ALGORITMA *DECISION TREE* C4.5 DENGAN TEKNIK *PRUNING*

Isa Iskandar [1)] Lely Hiryanto [2)] Janson Hendryli [3)]

[1)]Teknik Informatika Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta 11440 Indonesia
email : isaiskandar@ymail.com
[2)]Teknik Informatika Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta 11440 Indonesia
email : lely@fti.untar.ac.id
[3)]Teknik Informatika Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta 11440 Indonesia
email : Jansonh@fti.untar.ac.id

ABSTRACT

The system created are used to predict the length of study period required for students to complete their studies based on their grades. The system created also have online consultation features that students use with their academic lecturer for academic consultations. To find the model tree with good accuracy, the system will use k-fold cross validation in the process of making model tree. Testing prediction system using students data from 2008 to 2012 who have completed their studies. The value data used is all mandatory courses in the Faculty of Information Technology except for thesis courses. Based on the tests performed, the system can already run and used in accordance with the design made. The test is to compare the accuracy of the selected tree model from different k values in the k-fold cross validation process. The results obtained show that if the value of k the greater, then the accuracy obtained better.

Key words

Decision Tree, C4.5 Algorithm, Classification, Prediction

1. Pendahuluan

Informasi akademik mahasiswa sangatlah penting bagi dosen pembimbing akademiknya, dimana dosen pembimbing akademik diharapkan mengetahui perkembangan akademik mahasiswa bimbingannya. Untuk mengetahui hal itu, dibuatlah sebuah sistem prediksi lama masa studi mahasiswa, untuk memudahkan dosen pembimbing akademik melihat masa studi mahasiswanya, agar dosen pembimbing akademik lebih memperhatikan mahasiswa yang memiliki masa studi lebih lama dari lama masa studi yang tepat waktu (4 tahun).

Sistem prediksi yang dibuat menggunakan metode *decision tree* dengan algoritma C4.5, dimana masa studi mahasiswa akan dibagi menjadi 8 kelas, yaitu 7 Semester, 8 Semester, 9 Semster, 10 Semester, 11 Semester, 12 Semester, 12 Semester, 13 Semester, dan 14 Semester. Terdapat 48 mata kuliah yang akan digunakan untuk melakukan pembelajaran. Mata kuliah yang digunakan adalah seluruh mata kuliah wajib yang terdapat pada Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, tetapi mata kuliah skripsi tidak digunakan sebagai data untuk pembelajaran.

2. Klasifikasi

2.1 *K-Fold Cross Validation*

K-fold cross validation adalah salah satu teknik yang digunakan untuk mengestimasi akurasi dari hasil proses *data mining*, tetapi *k-fold cross validation* juga dapat digunakan sebagai *model selector* seperti yang akan digunakan pada perancangan ini. Pada *k-fold cross validation*, data dipartisi secara *random* menjadi subset sebanyak *k* bagian. Setiap subset (*folds)* memiliki ukuran yang sama dan pembagian kelas pada masing – masing subset juga diharuskan sama rata. Proses training dan testing dilakukan sebanyak *k* kali. Pada setiap iterasinya, partisi D_i akan menjadi *data testing* sedangkan partisi lainnya digunakan menjadi *data training*. Setiap iterasinya akan menghasilkan satu model *decision tree*, model yang akan digunakan adalah iterasi yang memiliki tingkat akurasi tertinggi.[1]

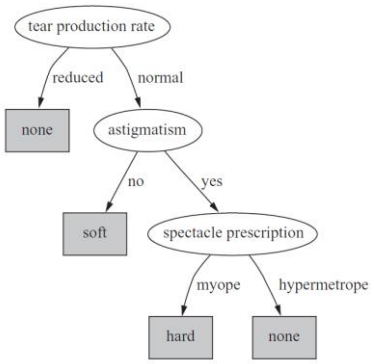
2.2 *Decision Tree*

Decision tree (pohon keputusan) merupakan salah satu metode yang populer digunakan dalam pengklasifikasian. Disebut sebagai pohon keputusan karena model yang dihasilkan untuk memprediksi data berupa pohon. Untuk mengklasifikasi sebuah data, setiap *attribute* dari data tersebut akan diuji melalui serangkaian

node yang terdapat pada pohon keputusan dan setelah data sampai pada *leaf node*, data tersebut akan terklasifikasi sesuai dengan kelas yang terdapat pada *leaf node.*

Pada pohon keputusan terdapat 3 jenis *node, node – node* tersebut adalah:

1. *Root Node*
   Root Node merupakan *node* yang letaknya berada diawal tree (diatas). *Root Node* tidak memiliki *input* yang berarti tidak ada cabang yang masuk ke *node* ini. *Root Node* dapat memiliki *output* lebih dari satu atau tidak memiliki *output* sama sekali.
2. *Internal Node*
   Internal Node merupakan node percabangan, pada *node* ini hanya terdapat satu *input* (satu cabang masuk) dan dapat memiliki *output* satu atau lebih.
3. *Leaf Node*
   Leaf Node atau *terminal node*, merupakan *node* akhir pada *decision tree*. *Node* ini hanya memiliki satu *input* dan tidak memiliki *output*. *Node* berperan untuk menunjukan kelas akhir dari pengklasifikasian.[2]



Gambar 1 Contoh *Decision Tree*

2.3 Algoritma C4.5

Algoritma C4.5 merupakan algoritma *supervised learning* yang berarti dalam pembuatan modelnya dibutuhkan data pembelajaran dan untuk pengujiannya dibutuhkan data pengujian. Terdapat 2 prinsip kerja dalam proses algoritma C4.5, yaitu Membangun pohon keputusan dan *pruning.*

2.3.1 Membangun Pohon Keputusan

Secara umum, langkah untuk membangun decision tree pada algoritma C4.5 adalah sebagai berikut:

1. Memilih *attribute* untuk menjadi akar (*Root Node*)
2. Membuat cabang untuk masing – masing nilai sebagai hasil dari attibute yang diuji
3. Membagi *attribute* sebagai *internal node* pada setiap cabang
4. Ulangi proses 2 dan 3 hingga setiap cabang berakhir pada *leaf node*

Untuk membangun *tree* dibutuhkan nilai *entropy*, *information gain*, *split info* dan *gain ratio*.

1. *Entropy*
   Entropy digunakan untuk menghitung *impurity* (kemiripan data) pada *dataset training.*

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

   Keterangan:
   S = *dataset training*.
   n = jumlah kelas dalam S.
   p_i = perbandingan jumlah data pada masing – masing kelas dengan total data yang terdapat dalam S.

2. *Information Gain*
   Information Gain digunakan untuk menentukan berapa banyak informasi yang dapat diberikan oleh *attribute* terhadap kelas yang ada.

$$\text{Gain(A)} = \text{Entropy(S)} - \sum_{i=1}^{n} \frac{|S_i|}{|S|} \text{ x Entropy}(S_i)$$

   Keterangan:
   A = *attribute*
   S = *dataset training*.
   n = jumlah partisi pada *attribute.*
   S_i = Partisi ke-i pada *attibute.*
   |S| = Jumlah data pada *attribute.*
   $|S_i|$ = Jumlah data pada partisi ke-i *attribute.*

   Untuk menentukan *root node*, nilai $Entropy(S)$ yang digunakan adalah *entropy* dari keseluruhan data. Pada pengulangan selanjutnya pada proses untuk menentukan *node hasil* dari *root node*, nilai $Entropy(S)$ yang digunakan adalah nilai *entropy* dari *attribute* yang menjadi *root node*. Sehingga dapat dikatakan nilai $Entropy(S)$ yang digunakan untuk mencari *Information Gain* sebuah attribute adalah *entropy* dari *node* sebelumnya (*Parent node)*.

3. *Split Info*
   Split Info digunakan untuk menghitung kemungkinan informasi yang dihasilkan dari pembagian. Semakin seragam pembagian nilai dari sebuah *attribute* nilai *split info* semakin besar.

$$Split(A) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \times \log_2\left(\frac{|S_i|}{|S|}\right)$$

   Keterangan:
   A = *attribute*.
   n = jumlah partisi pada *attribute.*
   S_i = Partisi ke-i pada *attibute.*
   |S| = Jumlah data pada *attribute.*
   $|S_i|$ = Jumlah data pada partisi ke-i *attribute.*

4. *Gain Ratio*
   Gain ratio digunakan untuk menggurangi bias dari information gain.

$$GainRatio(A) = \frac{Gain(A)}{Split(A)}$$

Keterangan:
A = *attribute*.
Gain(A) = nilai *information gain* pada *attribute* S.
Split(A) = nilai *Split information* pada *attribute* S.

2.3.2 *Pruning*

Saat pohon keputusan sudah berhasil dibangun, banyak dari cabang yang terbentuk menghasilkan klasifikasi yang tidak akurat dikarenakan terdapat *noise* atau *outlier* pada *data training*. Hal tersebut dinamakan *overfitting*. Untuk mengatasi masalah ini harus dilakukan *pruning* terhadap pohon keputusan yang telah dibuat. Teknik *pruning* umumnya menggunakan pendekatan statistikal untuk menghapus cabang yang bermasalah. Setelah di-*pruning*, pohon keputusan akan memiliki ukuran yang lebih kecil dan lebih tidak kompleks, sehingga lebih mudah untuk dimengerti.

Terdapat dua cara pruning yang biasa digunakan, yaitu *prepruning* dan *postpruning*. Pada *prepruning*, proses *pruning* akan dilakukan pada saat proses pembuatan *decision tree*. Pada sistem prediksi ini proses *pruning* yang akan digunakan adalah *postpruning* dengan menghitung estimasi error *decision tree*, cara ini dikenal dengan *reduced error pruning*.

Pada *reduced error pruning* akan dihitung rata – rata error dari setiap *leaf node* pada *subtree*, lalu subtree tersebut akan diubah *menjadi leaf node* dengan kelas yang yang terbanyak pada *subtree* tersebut. Lalu akan dihitung error baru dari *node* yang baru, jika error baru lebih kecil daripada error lama, *tree* berhasil di-*pruning*. Proses akan terus dilakukan hingga ditemukan error baru yang lebih baru dari error lama.[3]

Berikut ini adalah rumus untuk menghitung estimasi error:

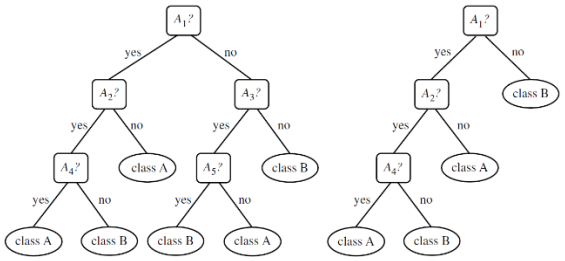
$$e = \frac{f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} + \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

Keterangan:
e = Estimasi error.
f = Hasil bagi dari jumlah data yang salah terklasifikasi dengan jumlah *data sample.*
N = Jumlah *data sample.*
z = nilai konstan yang secara *default* pada algoritma C4.5, z bernilai 0,69.

Untuk menghitung rata – rata error pada setiap kelas yang terdapat pada sebuah *subtree* menggunakan rumus:

$$TotalError = \sum_{i=1}^{n} \frac{E_i}{N} * e_i$$

Keterangan:
E_i = Jumlah data yang salah terklasifikasi pada *leaf node* ke-i.
N = Jumlah data sample.
e_i = estimasi error *leaf node* ke-i.



Gambar 2 Perbandingan *Tree Pruning*

3. Hasil Pengujian

Pengujian terhadap data dilakukan terhadap hasil prediksi lama masa studi mahasiswa. Pengujian ini dilakukan untuk mengetahui apakah hasil dari sistem untuk memprediksi lama masa studi yang dibuat sudah dapat dikatakan cukup akurat atau tidak. Data yang digunakan untuk pengujian adalah data mahasiswa angkatan 2008 sampai dengan 2012 yang telah menyelesaikan studinya. Data akan dibagi dengan ketentuan pembagian adalah 80% sebagai data *training* dan 20% sebagai data *testing*. Pada tahap ini akan dilakukan 2 pengujian, yaitu pengujian terhadap model *tree* yang terbentuk dan pengujian terhadap hasil prediksi.

Pada pengujian terhadap model *tree* ini dilakukan dengan cara membandingkan akurasi dari setiap model *tree* yang terpilih dari proses *k-fold cross validation* pada nilai *fold* yang berbeda. Pada pengujian ini juga akan diperhatikan rata – rata akurasi dari setiap model *tree* yang terbentuk dalam proses *k-fold cross validation*. Akurasi yang diperhatikan adalah akurasi dari *tree* sebelum dilakukan proses *pruning* dan akurasi setelah dilakukan proses *pruning* pada setiap *fold*. Hasil penujian ini dapat dilihat pada **Tabel 1** dan **Tabel 2.**

Tabel 1 Akurasi model tree yang terpilih

Percobaan	k	Akurasi model tree	
		Sebelum Prune	Setelah Prune
1	2	33,9449%	49,5412%
2	3	47,9452%	50,6849%
3	4	46,2962%	48,1481%
4	5	51,1628%	53,4883%
5	10	56,6667%	60%
6	15	56,5217%	65,2173%

Tabel 2 Rata – rata Akurasi model tree

Percobaan	k	Akurasi rata - rata	
		Sebelum Prune	Setelah Prune
1	2	35,6088%	43,4070%
2	3	39,7260%	46,1187%
3	4	41,2037%	44,3469%
4	5	43,6714%	45,3735%
5	10	49,4761%	47,4285%
6	15	46,6252%	48,6335%

Lalu akan dilakukan pengujian untuk membuktikan apakah proporsi jumlah kelas dapat mempengaruhi akurasi yang dihasilkan, akan dilakukan 2 kali percobaan. Untuk membuat model tree pada kedua percobaan ini, tidak menggunakan proses *k-fold cross validation*. Percobaan pertama menggunakan data training dengan jumlah proporsi kelas yang sama percobaan kedua mengguanakan data dengan jumlah proporsi kelas yang tidak sama. Hasil pengujian ini dapat dilihat pada **Tabel 3.**

Tabel 3 Hasil percobaan proporsi data

Proporsi Data	Rentang (Semester)			Akurasi
	0	1	2	
Sama	12	6	0	66,667%
Tidak Sama	6	12	0	33,333%

Lalu untuk menguji ketepatan akurasi yang dihasilkan oleh sistem, akurasi dari sistem prediksi yang dibuat akan dibandingkan dengan akurasi yang dihasilkan oleh aplikasi Weka. Aplikasi Weka adalah sebuah aplikasi yang digunakan untuk memproses data. Aplikasi Weka dapat memproses data dengan berbagai algoritma klasifikasi maupun *cluster* termasuk *decision tree* C4.5, yang pada aplikasi dinamakan *decision tree* J48. Pada percobaan ini tidak digunakan proses *k-fold cross validation* dalam membuat model *tree* yang digunakan untuk prediksi. Dari hasil yang didapatkan pada percobaan ini, akurasi pada sistem prediksi yang dibuat menunjukan angka yang lebih tinggi dengan akurasi sebesar 53,448%, sedangkan hasil akurasi yang didapatkan dari aplikasi Weka menunjukan angka yang lebih kecil dengan akurasi sebesar 46,551%. Hasil perngujian ini dapat dilihat pada **Tabel 4.**

Tabel 4 Perbandingan akurasi dengan WEKA

	Benar	Salah	Akurasi
Weka	27	31	46,551%
Sistem Prediksi	31	27	53,448%

Pada pengujian terhadap hasil prediksi ini dilakukan dengan cara membandingkan hasil prediksi yang didapatkan seorang mahasiswa dengan lama masa studi yang diperlukan olehnya untuk menyelesaikan studi. Pada pengujian ini, hasil yang diperhatikan adalah akurasi dari kebenaran prediksi sistem, jumlah data yang benar terprediksi, jumlah data yang salah terprediksi, dan rentang antara kelas yang sebenarnya dengan hasil prediksi yang salah. Pada pengujian ini, data testing akan dibandingkan dengan 2 model tree, yaitu model tree sebelum pruning dan model tree sesudah pruning. Hasil pengujian ini dapat dilihat pada **Tabel 5**, **Tabel 6**, **Tabel 7**, dan **Tabel 8**.

Tabel 5 Rentang kesalahan prediksi sebelum pruning

Percobaan	k	Rentang (Semester)							
		0	1	2	3	4	5	6	7
1	2	15	24	8	9	1	0	1	0
2	3	28	21	8	0	0	1	0	0
3	4	30	20	4	3	0	1	0	0
4	5	30	17	7	1	2	1	0	0
5	10	24	21	6	4	2	1	0	0
6	15	27	18	7	4	2	0	0	0

Tabel 6 Akurasi prediksi sebelum pruning

Percobaan	k	Benar	Salah	Akurasi
1	2	15	43	25,8620%
2	3	28	30	48,2758%
3	4	30	28	51,7241%
4	5	30	28	51,7241%
5	10	24	34	41,3793%
6	15	27	31	46,5517%

Tabel 7 Rentang kesalahan setelah pruning

Percobaan	k	Rentang							
		0	1	2	3	4	5	6	7
1	2	29	16	6	5	0	0	2	0
2	3	27	23	4	3	1	0	0	0
3	4	32	18	6	1	0	1	0	0
4	5	31	16	7	1	2	1	0	0
5	10	27	18	6	4	2	1	0	0
6	15	29	16	7	4	2	0	0	0

Tabel 8 Akurasi Prediksi setelah pruning

Percobaan	k	Benar	Salah	Akurasi
1	2	29	29	50%
2	3	27	31	46,5517%
3	4	32	26	55,1724%
4	5	31	27	53,4482%
5	10	27	31	46,5517%
6	15	29	29	50%

4. Kesimpulan

Kesimpulan yang didapatkan dari hasil pengujian yang dilakukan terhadap sistem prediksi kelulusan mahasiswa menggunakan algoritma *decision tree* C4.5 dengan teknik *pruning* adalah sebagai berikut:

1. Berdasarkan dari percobaan yang membandingkan akurasi dari model *tree* yang terpilih berdasarkan jumlah *fold* pada proses *k-fold cross validation* menunjukan bahwa semakin besar nilai *fold* maka akurasi dari model *tree* yang terpilih juga akan semakin besar.
2. Hasil yang didapatkan dari percobaan yang membandingkan akurasi proporsi jumlah kelas yang sama dengan proporsi yang berbeda menunjukan bahwa proporsi jumlah kelas dapat mempengaruhi akurasi dari model *tree* yang dihasilkan.
3. Melalui hasil yang didapatkan dari pengujian terhadap hasil prediksi, didapatkan kesimpulan bahwa model *tree* dengan akurasi yang tinggi pada proses membuat *tree* tidak *menjadikan* model *tree* tersebut menjadi yang terbaik jika digunakan untuk memprediksi data baru.
4. Berdasarkan hasil yang didapatkan dari pengujian terhadap hasil prediksi, dapat dilihat bahwa meskipun terdapat kesalahan prediksi, kebanyakan dari hasil yang salah terprediksi hanya memiliki rentang 1 semester dari hasil yang sebenarnya.

REFERENSI

[1] Han, J., & Kamber, M., 2006. Data mining Concepts and Techniques. San Fransisco: Morgan Kaufmann. H 291.
[2] Ruano, Antonio Eduardo de Barros. Artificial Neural Network. Portugal: University of Algrave, 2010.
[3] Bishop, Christoper M. Pattern Recognition and Machine Learning. Singapore: Science+Business Media, 2006.

Isa Iskandar, merupakan mahasiswa tingkat akhir Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta

Lely Hiryanto, memperoleh gelar ST dari Universitas Tarumanagara. Kemudian memperoleh M.Sc dari *Curtin University of Technology*. Saat ini aktif sebagai dosen tetap Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta.

Janson Hendryli, memperoleh gelar S.Kom dari Universitas Tarumanagara. Kemudian memperoleh M.Kom dari Universitas Indonesia. Saat ini aktif sebagai dosen tetap Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta.