ANALISIS REKOMENDASI PEMINATAN MENGGU NAKAN METODE DECISION TREE DENGAN ALGORITMA C4.5

Stefanny Claudia[1)] **Tri Sutrisno** [2)]

[1)]Teknik Informatika Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta 11440 Indonesia
email : stefannyclaudia96@yahoo.com
[2)]Teknik Informatika Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta 11440 Indonesia
email : tris@fti.untar.ac.id

ABSTRAK

The application created are used to analyze which thesis preference subject suits students academic performance based on their academic grades. The application also provide online academic consultations features which students can use for their academic consultations. To find their thesis preference, the application use decision tree method with C4.5 algorithm. Testing prediction system using students data from 2012 to 2015 who have found their thesis preference. The value data used is 32 mandatory courses in the Faculty of Information Technology before thesis preference. The application can run , use and perform well in accordance with the design made. Testing is to compare the accuracy of the selected tree model build from training data and the thesis preference students have selected. The average accuracy percentage of this a 72,6227%.

Key Words

Algorithm, Analyze, C4.5, Classification Decision Tree,

1. Pendahuluan

Pada Perguruan tinggi, mahasiswa diharapkan untuk memilih Fakultas, Program Studi, dan Jurusan demi menyelesaikan studi di jenjang perguruan tinggi. Di Universitas Tarumanagara, Fakultas Teknologi Informasi, Program Studi Teknik Informatika, ditawarkan 5 jenis peminatan yang berfungsi sebagai referensi untuk menyelesaikan skripsi.

Dalam pemilihan peminatan yang ditawarkan oleh Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, banyak mahasiswa yang tidak memilih peminatan yang sesuai dengan kemampuan akademik mereka sehingga mahasiswa mengalami kesulitan dalam pembelajaran hingga proses pembuatan skripsi sesuai dengan peminatan yang dipilih. Minimnya pengetahuan tentang peminatan, membuat peminatan tidak befungsi
dengan baik. Peminatan seharusnya dapat memudahkan mahasiswa dalam penyelesaian studi, karena penjurusan bertujuan agar mahasiswa dapat fokus pada pembelajaran konsentrasi spesifik dari peminatan.

Guna membantu memecahkan persoalan yang dihadapi dalam kegiatan konsultasi akademik, dibuat sebuah sistem yang dapat memberikan analisis rekomendasi peminatan mahasiswa menggunakan metode *decision tree* dengan algoritma C4.5 pada aplikasi konsultasi akademik online berbasis *website*.

2. Klasifikasi

Klasifikasi adalah proses menemukan model(fungsi) yang menjelaskan dan membedakan kelas-kelas atau konsep, dengan tujuan agar model yang diperoleh dapat digunakan untuk memprediksikan kelas atau objek yang memiliki label kelas tidak diketahui. Model yang diturunkan didasarkan pada analisis dari pemebelajaran data, yaitu objek yang memiliki label kelas yang diketahui. Model yang diturunkan dapat direpresentasikan dalam berbagai bentuk seperti aturan IF-THEN klasifikasi, pohon keputusan, formula matematika atau jaringan syaraf tiruan.

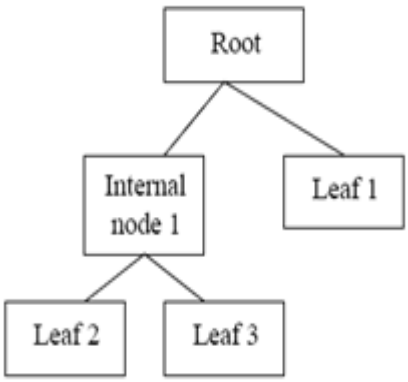
Input merupakan sekumpulan *record*(*training set)*, setiap *record* terdiri atas kumpulan atribut, salah satu atribut adalah kelas. Tujuannya adalah menyatakan kelas dengan akurat dari *record* yang sebelumnya belum memiliki kelas. Model klasifikasi digunakan untuk pemodelan deskriptif sebagai perangkat penggambaran untuk membedakan objek-objek dari kelas yang berbeda

dan melakukan prediksi label kelas untuk *record* yang belum diketahui.

2.1 Decision Tree

Decision tree atau pohon keputusan adalah salah satu metode klasifikasi yang dapat diinterpretasikan, yang menghasilkan model prediksi menggunakan struktur pohon. Konsep dari pohon keputusan adalah mengubah data menjadi pohon keputusan dan aturan-aturan keputusan. Secara visualisasi, pohon keputusan menyerupai *flowchart* seperti struktur *tree*, dimana tiap model *internal node* menunjukkan sebuah *test* pada sebuah atribut, tiap cabang menunjukkan hasil dari *test* dan *leaf node* menunjukkan *class-class* atau *clas distribution.* 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*

2.2 Algoritma C4.5

Algoritma C4.5 merupakan kelompok algoritma pohon keputusan. Algoritma ini mempunyai input berupa *training samples* dan *samples*. *Training samples* berupa data contoh yang akan digunakan untuk membangun sebuah *tree* yang telah diuji kebenarannya, sedangkan *samples* adalah *field* data yang nantinya akan digunakan sebagai parameter dalam melakukan klasifikasi data.

Pada tahap pembelajaran algoritma C4.5 memiliki 2 prinsip kerja, yaitu:

1. Pembuatan pohon keputusan atau *decision tree*. Tujuan dari algoritma penginduksi pohon keputusan adalah mengkonstruksi struktur data pohon yang dapat digunakan untuk memprediksi keslas dari sebuah kasus atau *record* baru yang belum memiliki kelas. Algoritma C4.5 melakukan konstruksi pohon keputusan dengan metode *divide and conquer*. Pada awalnya hanya dibuat *root node* dengan menerapkan algoritma *divide and conquer*. Algoritma ini memilih pemecahan kasus-kasus yang terbaik dengan menghitung dan membandingkan *gain ratio*, kemudian *node-node* yang terbentuk di tingkat berikutnya, algoritma ini akan diterapkan kembali untuk membentuk *leaf node*.
2. Pembuatan aturan atau *rule set*. Aturan-aturan yang terbentuk dari *decision tree* akan menghasilkan kondisi dalam bentuk *if-then*. Aturan ini didapatkan dengan cara melakukan penelusuran pohon keputusan dari *root node* hingga *leaf node*. Setiap *node* dan percabangan akan membentuk kondisi *if*, sedangkan untuk nilai-nilai yang berada pada *leaf node* akan membentuk kondisi *then* atau hasil..[2]

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.

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

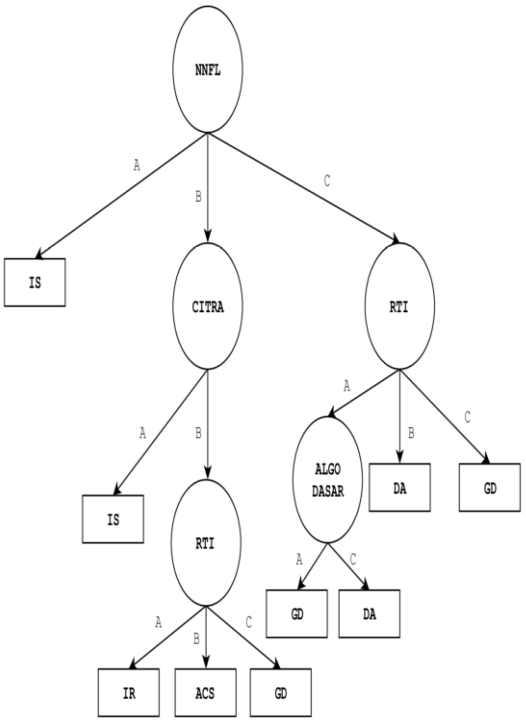
Keterangan:
A = *attribute*.
n = jumlah partisi pada *attribute.*
S_i = Partisi ke-i pada *attibute.*
|S| = Jumlah data pada *attribute.*
$|S_i|$ = Jumlah data pada partisi ke-i *attribute.*

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

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

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

Tree yang terbentuk :



Gambar 2 Tree yang terbentuk

3. Hasil Pengujian

Pengujian terhadap data dilakukan terhadap hasil analisis rekomendasi peminatan mahasiswa. Pengujian ini dilakukan untuk mengetahui apakah hasil dari aplikasi menganalisis rekomendasi peminatan mahasiswa yang dibuat sudah dapat dikatakan cukup akurat atau tidak. Data yang digunakan untuk pengujian adalah data nilai mahasiswa angkatan 2012 sampai dengan 2015, dari semester 1 hingga semester 5. Data akan dibagi dengan ketentuan pembagian adalah 70% sebagai *data training* dan 30% sebagai *data testing*. Data yang diambil adalah data nilai mahasiswa yang telah mengambil peminatan tanpa melihat nilai fitur yang

digunakan dan jumlah distribusi kelas. Pada tahap ini akan dilakukan pengujian akurasi dengan membandingkan hasil analisis rekomendasi peminatan dengan kondisi nyata peminatan yang diambil oleh mahasiswa.

Data Training	Data Testing	Benar	Salah	Akurasi
177 data 2012-2014	53 data 2012-2014	44	9	83,0189 %
114 data 2012-2014	34 data 2012-2014	25	9	73,5294 %
150 data 2012-2014	45 data 2012-2014	30	15	66,6667 %
195 data 2012-2014	58 data 2012-2014	46	12	69,3103 %
170 data 2012-2014	51 data 2012-2014	36	15	70,5882 %

Tabel 1 Percobaan Data

Pada percoban kedua hingga kelima dilakukan percobaan dengan proporsi jumlah kelas *data training* tidak seimbang, dan hasil analisis menunjukkan hasil akurasi yang berbeda-beda. Hal yang mempengaruhi hasil analisis adalah proporsi kelas *data training* dan model tree yang terbentuk dari *data training*. Presentase rata-rata akurasi dari kelima percobaan sebesar 72,6227%.

4. Kesimpulan

Kesimpulan yang dapat ditarik berdasarkan pembuatan dan pengujian dari aplikasi analisis rekomendasi peminatan menggunakan metode decision tree dengan algoritma C4.5 adalah sebagai berikut :

1. Pada percobaan 1 menghasilkan akurasi sebesar 83,0189% , pada percobaan 2 menghasilkan akurasi sebesar 73,5294%, pada percobaan 3 menghasilkan akurasi sebesar 66,6667%, pada percobaan 4 menghasilkan akurasi sebesar 39,3103% dan pada percobaan 5 menghasilkan akurasi sebesar 70,5882%.
2. Dari 5 percobaan yang telah dilakukan dapat disimpulkan bahwa besar jumlah *data training* tidak menentukan hasil akurasi yang baik.
3. Terdapat beberapa hasil pengujian analisis peminatan yang tidak sesuai dengan kenyataannya.
4. Dari hasil pengujian terhadap modul, dapat diketahui bahwa semua modul dalam sistem dapat berjalan dengan baik dan sesuai fungsinya.
5. Setiap modul telah dapat menjalankan fungsinya masing-masing dengan baik, semua tombol dan *textbox* dari antar muka program dapat berjalan dengan baik.
6. Hasil rata-rata akurasi dari seluruh percobaan1 hingga percobaan 5 memilki akurasi sebesar 72,6227%
7. Hasil yang didapatkan dari percobaan yang membandingakan akurasi proporsi jumlah kelas yang sama dengna proporsi jumlah kelas yang berbeda menunjukkan bahwa proporsi jumlah kelas dapat mempengaruhi akurasi dari model tree yang dihasilkan.
8. Model tree yang dihasilkan dengan akurasi tinggi tidak menjadikan model tree tersebut menjadi model yang terbaik jika digunakan untuk data yang baru.
9. Menu dan fitur yang dibuat sudah dapat berjalan sebagaimana mestinya sehingga hasil pengujian terhadap modul yang dibuat menggunakan metode blackbox testing sudah sesuai.

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] Sunjana, 2010, Aplikasi Mining Data Mahasiswa Dengan Metode Klasifikasi Decision Tree, (Yogyakarta :Seminar Nasional Aplikasi Teknologi Informasi, 2010) h.3.

Stefanny Claudia, merupakan mahasiswa tingkat akhir Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta.

Tri Sutrisno, memperoleh gelar S.Si dari Universitas Diponegoro. Kemudian memperoleh M.Sc dari Universitas Gadjah Mada. Saat ini aktif sebagai dosen tetap Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta.