

Metode *Iterative Dichotomizer 3 (ID3)*

Untuk Penerimaan Mahasiswa Baru

Wahyudin

Program Studi Ilmu Komputer

Universitas Pendidikan Indonesia

Abstrak

Konsep pohon merupakan salah satu konsep teori graf yang paling penting. Pemanfaatan pohon dalam kehidupan sehari-hari adalah untuk menggambarkan hierarki dan memodelkan persoalan, contohnya pohon keputusan (*decision tree*). *iterative dichotomiser 3 (ID3)* merupakan suatu metode dalam learning yang akan membangun sebuah pohon keputusan yang merupakan suatu pemodelan dalam mencari solusi dari persoalan. Dalam jurnal ini akan dibahas pemakaian pohon keputusan dalam bidang pendidikan, yaitu dalam hal penerimaan mahasiswa baru. Penerimaan mahasiswa baru yang baik mencakup langkah-langkah yang dapat memaksimalkan peluang keberhasilan. Salah satu tahap yang penting adalah tahap pengambilan keputusan. Kegiatan analisis kemahasiswaan diperlukan untuk mendapatkan keputusan yang bersifat menguntungkan demi maju dan berkembangnya suatu universitas dan analisis penerimaan mahasiswa baru tersebut dapat dilakukan melalui berbagai metode, salah satunya dengan *decision tree* menggunakan ID3 (*Iterative Dichotomiser 3*). Jurnal ini akan memperlihatkan pemakaian pohon keputusan untuk memudahkan pengambilan keputusan penerimaan mahasiswa baru dalam suatu universitas.

Kata kunci : *Decision tree, Iterative Dichotomiser 3*

PENDAHULUAN

LATAR BELAKANG

Mahasiswa merupakan input bagi Universitas yang sangat penting. Kualitas perguruan tinggi salah satunya dilihat dari prosentase mahasiswa yang berminat masuk ke perguruan tinggi dengan yang diterima di perguruan tinggi tersebut. Pengambilan keputusan penerimaan mahasiswa baru merupakan kebijakan yang sangat penting demi kemajuan dan berkembangnya suatu perguruan tinggi, Agar kegiatan pendidikan yang akan dilaksanakan maupun yang sedang berjalan tetap berada pada jalur yang benar dan akan memberikan dampak positif bagi universitas.

Dalam pengambilan keputusan penerimaan mahasiswa baru ini dibatasi dengan memperhatikan tiga atribut yaitu nilai SPMB, UAN, dan psikotest. Metode yang digunakan adalah metode *Decision Tree* menggunakan ID3 (*Iterative Dichotomiser 3*). Pengambilan keputusan ini dapat memberikan data yang lengkap sehingga akan lebih mudah bagi pihak universitas menentukan seseorang/peserta SPMB (Seleksi Penerimaan Mahasiswa

Baru) diterima atau tidak di universitas tersebut.

Permasalahan

Berdasarkan latar belakang diatas salah satu pokok permasalahan yang diangkat adalah memperoleh pengetahuan dan mengidentifikasi mahasiswa baru yang akan diterima disebuah universitas sehingga akan memberikan kemudahan bagi pihak universitas untuk menentukan siapa saja yang berhak masuk atau diterima berdasarkan kriteria tertentu dalam hal ini nilai SPMB, nilai UAN (Ujian Akhir Nasional) dan nilai psikotest. Metode atau cara yang digunakan adalah membentuk pohon keputusan dengan algoritma *Iterative Dichotomizer 3 (ID3)*.

Decision tree learning

Decision tree learning adalah salah satu metode belajar yang sangat populer dan banyak digunakan secara praktis. Metode ini merupakan metode yang berusaha menemukan fungsi-fungsi pendekatan yang bernilai diskrit dan tahan terhadap data-data yang terdapat kesalahan (*noisy data*) serta mampu mempelajari ekspresi-ekspresi *disjunctive* (ekspresi OR). *Iterative Dychotomizer version 3 (ID3)* adalah salah satu jenis *Decision tree* yang sangat populer.

Konsep Decision Tree

Mengubah data menjadi pohon keputusan (decision tree) dan aturan-aturan keputusan (rule)

DATA \longrightarrow DECISION TREE \longrightarrow RULE

decision tree adalah struktur *flowcart* yang mempunyai tree(pohon), dimana setiap simpul internal menandakan suatu tes atribut,setiap cabang merepresentasikan

hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas. alur pada *decision tree* ditelusuri dari simpul ke akar ke simpul daun yang memegang prediksi kelas untuk contoh tersebut. *decision tree* mudah untuk dikonversi ke aturan klasifikasi (*classification rule*).

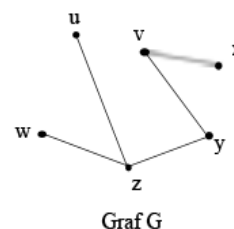
Konsep data dalam *decision tree* dinyatakan dalam bentuk tabel dengan atribut dan record.

Teori Graf

Graf merupakan sekumpulan simpul (*node*) dan lengkungan yang menghubungkan simpul-simpul tersebut. Graf berlabel memiliki satu atau lebih label yang terkait untuk setiap simpul yang membedakannya dari setiap simpul yang ada pada graf tersebut. Dalam sebuah graf ruang keadaan, label-label ini mengidentifikasi keadaan-keadaan dalam proses pemecahan problema. Secara matematis, graf dapat didefinisikan sebagai berikut :

Graf $G (V,E)$ adalah suatu sistem yang terdiri dari himpunan titik berhingga tak kosong. $V = V (G)$ dan himpunan sisi $E = E (G)$ yaitu himpunan pasangan tak terurut dari anggota anggota V .

Sebagai contoh, gambar 1 adalah graf dengan himpunan titik $V(G) = \{u, v, x, y, z, w\}$ dan himpunan sisi $E (G) = \{vx, vy, yz, zu, zw\}$.



Gambar 1: Graf G

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis :

1) Graf tak-berarah (*undirected graph*)

Graf yang sisinya tidak mempunyai orientasi arah. Pada graf tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(u,v) = (v,u)$ adalah sisi yang sama.

2) Graf berarah (*directed graph* atau *digraph*)

Graf yang setiap sisinya diberikan orientasi arah. Pada graf berarah sisi berarah disebut dengan busur(arc). Pada graf berarah, (u,v) dan (v,u) menyatakan dua buah busur yang berbeda, dengan kata lain $(u,v) \neq (v,u)$. Untuk busur (u,v) simpul u dinamakan simpul asal (initial vertex) dan simpul v dinamakan simpul terminal (terminal vertex).

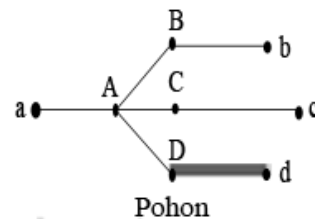
Pohon (tree)

Pohon merupakan sebuah graf terhubung yang tidak mengandung sirkuit. Konsep pohon (tree) dalam teori graf merupakan konsep yang sangat penting, karena terapanannya diberbagai bidang ilmu. Oleh karenanya antara pohon (tree) sangat erat hubungannya dengan teori graf.

Definisi pohon adalah graf tak berarah terhubung yang tidak mengandung sirkuit, menurut definisi tersebut, ada dua sifat penting pada pohon yaitu terhubung dan tidak mengandung sirkuit.

Pohon(tree) merupakan graf dimana dua simpul memiliki paling banyak satu lintasan

yang menghubungkannya. Pohon seringkali memiliki akar . karena setiap simpul pada pohon hanya memiliki satu lintasan akses dari setiap simpul lainnya, maka tidak mungkin bagi sebuah lintasan untuk membentuk simpul (*loop*) atau siklus (*cycle*) yang secara berkesinambungan melalui serangkaian simpul.



Gambar 2 : pohon

Algoritma ID3

Iterative Dichotomiser 3 (ID3) adalah algoritma *decision tree learning* (algoritma pembelajaran pohon keputusan) yang paling dasar. Algoritma ini melakukan pencarian secara rakus/menyeluruh (greedy) pada semua kemungkinan pohon keputusan.

Salah satu algoritma induksi pohon keputusan yaitu ID3 (Iterative Dichotomiser 3). ID3 dikembangkan oleh J. Ross Quinlan.

Algoritma ID3 dapat diimplementasikan menggunakan fungsi rekursif (fungsi yang memanggil dirinya sendiri). Algoritma ID3 berusaha membangun decision tree (pohon keputusan) secara top-down (dari atas ke bawah), mulai dengan pertanyaan : “atribut mana yang pertama kali harus dicek dan diletakkan pada root?” pertanyaan ini dijawab dengan mengevaluasi semua atribut yang ada dengan menggunakan suatu ukuran statistik (yang banyak digunakan adalah information gain) untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan kumpulan sampel data.

Karakteristik ID3 dalam membangun pohon keputusan adalah secara *top-down* dan *divide-and-conquer*. *Top-down* artinya pohon keputusan dibangun dari simpul akar ke daun, sementara *divide-and-conquer* artinya *training data* secara rekursif dipartisi ke dalam bagian-bagian yang lebih kecil saat pembangunan pohon.

Decision Tree adalah sebuah struktur pohon, dimana setiap node pohon merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan node daun (*leaf*) merepresentasikan kelompok kelas tertentu. Level node teratas dari sebuah *Decision Tree* adalah node akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu. Pada umumnya *Decision Tree* melakukan strategi pencarian secara *top-down* untuk solusinya. Pada proses mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari node akar (*root*) sampai node akhir (daun) dan kemudian akan diprediksi kelas yang dimiliki oleh suatu data baru tertentu.

Entropy dan Information Gain

Sebuah obyek yang diklasifikasikan dalam pohon harus dipesan nilai entropinya. Entropy adalah ukuran dari teori informasi yang dapat mengetahui karakteristik dari impurity, dan homogeneity dari kumpulan data. Dari nilai entropy tersebut kemudian dihitung nilai information gain (IG) masing-masing atribut.

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

dimana :

S adalah ruang (data) sample yang digunakan untuk training.

P_+ adalah jumlah yang bersolusi positif (mendukung) pada data sample untuk kriteria tertentu.

P_- adalah jumlah yang bersolusi negatif (tidak mendukung) pada data sample untuk kriteria tertentu.

Dari rumus entropy diatas dapat disimpulkan bahwa definisi entropy adalah, entropy (S) adalah jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas (+ atau -) dari sejumlah data acak pada suatu ruang sampel S. Entropy bisa dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Semakin kecil nilai entropy maka semakin baik digunakan dalam mengekstraksi suatu kelas. Panjang kode untuk menyatakan informasi secara optimal adalah $-\log_2 p$ bits untuk messages yang mempunyai probabilitas p. Sehingga jumlah bit yang diperkirakan untuk mengekstraksi S ke dalam kelas adalah : $-p_+ \log_2 p_+ - p_- \log_2 p_-$.

Information Gain

setelah mendapat nilai entropy untuk suatu kumpulan data, maka kita dapat mengukur efektivitas suatu atribut dalam mengklasifikasikan data. Ukuran efektifitas ini disebut information gain. Secara matematis, information gain dari suatu atribut A, dituliskan sebagai berikut :

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

dimana :

A : atribut

V : menyatakan suatu nilai yang mungkin untuk atribut A

Values(A) : himpunan yang mungkin untuk atribut A

|Sv| : jumlah sampel untuk nilai v

|S| : jumlah seluruh sampel data

Entropy(Sv): entropy untuk sampel-sampel yang memiliki nilai v

METODE

- Mencari sampel data yang akan digunakan untuk melaksanakan proses transformasi menjadi pengetahuan
- Menghitung nilai entropy dan information gain untuk menentukan the best classifier
- Melakukan konstruksi pohon keputusan
- Lakukan operasi conjunction (\wedge) pada setiap simpul yang dikunjungi sampai ditemukan leaf node.

PEMBAHASAN

DESKRIPSI PERMASALAHAN

Data penerimaan Mahasiswa

Terdapat 11 orang mahasiswa yang mengikuti seleksi penerimaan mahasiswa baru (SPMB) dengan memperhatikan 3 parameter / atribut penilaian

Tiga parameter / atribut :

- Nilai SPMB dikelompokkan dalam 3 kategori (Tinggi, sedang, Rendah)
- UAN dikelompokkan dalam 3 kategori (bagus, cukup, kurang)
- Psikotest dikelompokkan dalam 2 kategori (baik dan buruk)

Untuk data yang lengkap seharusnya terdapat $3 \times 3 \times 2 = 18$ kombinasi sampel data, tetapi dalam tabel kombinasi dibawah

hanya terdapat 11 sampel data artinya masih ada 7 sampel data lain yang belum diketahui. Contoh : [nilai SPMB 'Tinggi' UAN 'Kurang' Psikotest 'Buruk']. Kita tidak bisa menentukan peserta SPMB tersebut diterima atau tidak.

Dengan Decision Tree menggunakan algoritma ID3 aturan atau data yang belum diketahui bisa kita temukan sehingga kita bisa menentukan seorang peserta tes diterima atau tidak.

4.2 ANALISIS INFORMATION GAIN

pada data penerimaan mahasiswa, jumlah kelas adalah 2, yaitu : 'ya' dan 'tidak' (c = 2

Peserta SPMB	Nilai SPMB	UAN	Psikotes t	Diterima
P1	Tinggi	Bagus	Baik	Ya
P2	Tinggi	Cukup	Baik	Ya
P3	Tinggi	Cukup	Buruk	Ya
P4	Tinggi	Kurang	Buruk	tidak
P5	Sedang	Bagus	Baik	Ya
P6	Sedang	Cukup	Baik	Ya
P7	Sedang	Cukup	Buruk	Ya
P8	Sedang	Kurang	Buruk	tidak
P9	Rendah	Bagus	Baik	Ya
P10	Rendah	Cukup	Buruk	tidak
P11	Rendah	Kurang	Baik	Ya

). Jumlah sampel kelas 1 ('ya') adalah 8 dan jumlah sampel untuk kelas 2 ('tidak') adalah

3 . jadi $p_1 = 8$ dan $p_2 = 3$. Dengan demikian entropy untuk kumpulan sampel data S adalah :

$$\begin{aligned} \text{Entropy (S)} &= - (8/11) \log_2 (8/11) - (3/11) \\ &\log_2 (3/11) \\ &= 0,8454 \end{aligned}$$

Dari table missal atribut Diterima='ya' merupakan sampel (+), dan atribut Diterima='Tidak' merupakan sampel (-) , dari sampel data pada table didapat :

Values (nil.SPMB) = Tinggi, Sedang, Rendah

$$S = [8+ , 3-] , | S | = 11$$

$$S_{\text{Tinggi}} = [3+ , 1-] , | S_{\text{Tinggi}} | = 4$$

$$S_{\text{sedang}} = [3+ , 1-] , | S_{\text{sedang}} | = 4$$

$$S_{\text{Rendah}} = [2+ , 1-] , | S_{\text{Rendah}} | = 3$$

Hitung entropy S_{Tinggi} , S_{sedang} , S_{Rendah} dan Information Gain Untuk nilai SPMB adalah :

$$\text{Entropy (S)} = 0,8454$$

$$\text{Entropy (S}_{\text{Tinggi}}) = (-3/4) \log_2 (3/4) - (1/4) \log_2 (1/4) = 0,8113$$

$$\text{Entropy (S}_{\text{sedang}}) = (-3/4) \log_2 (3/4) - (1/4) \log_2 (1/4) = 0,8113$$

$$\text{Entropy (S}_{\text{Rendah}}) = (-2/3) \log_2 (2/3) - (1/3) \log_2 (1/3) = 0,9183$$

• Information Gain (S, Nil.SPMB)

$$= \text{Entropy(S)} -$$

$$\sum_{\text{v}} (\text{Tinggi,Sedang,Rendah}) \frac{|S_v|}{|S|} \text{Entropy (S}_v)$$

$$= 0,8454 - (4/11)0,8113 - (3/11) 0,9183$$

$$= 0,8454 - 0,2950 - 0,2950 - 0,25044$$

$$= 0,0049$$

Values(UAN) = Bagus, Cukup, Kurang

$$S_{\text{Bagus}} = [3+ , 0-] , | S_{\text{Bagus}} | = 3$$

$$S_{\text{Cukup}} = [4+ , 1-] , | S_{\text{Cukup}} | = 5$$

$$S_{\text{Kurang}} = [1+ , 2-] , | S_{\text{Kurang}} | = 3$$

$$\begin{aligned} \text{Entropy (S}_{\text{Bagus}}) &= (-3/3) \log_2 (3/3) - (0/3) \log_2 (0/3) \\ &= 0 \end{aligned}$$

menunjukkan entropy minimum karena jumlah sampel pada salah satu kelas adalah = 0 (keberagaman data minimum).

$$\text{Entropy (S}_{\text{Cukup}}) = (-4/5) \log_2 (4/5) - (1/5) \log_2 (1/5) = 0,72192$$

$$\text{Entropy (S}_{\text{Kurang}}) = (-1/3) \log_2 (1/3) - (2/3) \log_2 (2/3) = 0,91829$$

$$\text{IG (S, UAN)} = \text{Entropy (S)} - \sum_{\text{v}} (\text{Tinggi,Sedang,Rendah}) \frac{|S_v|}{|S|} \text{Entropy (S}_v)$$

$$= 0,8454 - 0 - (5/11) 0,7219 - (3/11) 0,9183$$

$$= 0,8454 - 0,32814 - 0,25044$$

$$= 0,26682$$

$$S_{\text{Baik}} = [6+ , 0-] , | S_{\text{Baik}} | = 6$$

$$S_{\text{Buruk}} = [2+ , 3-] , | S_{\text{Buruk}} | = 5$$

$$\text{Entropy (S}_{\text{Baik}}) = (-6/6) \log_2 (6/6) - (0/6) \log_2 (0/6)$$

$$= 0 (\text{ keberagaman data minimum })$$

$$\text{Entropy (S}_{\text{Buruk}}) = (-2/5) \log_2 (2/5) - (3/5) \log_2 (3/5)$$

$$= 0,97094$$

$$\text{IG (S, Psikotest)} = 0,8454 - 0 - (5/11) 0,97094$$

$$= 0,8454 - 0,44134$$

$$= 0,40406$$

Dari perhitungan diatas didapat nilai Information Gain dari ketiga atribut (Nil.SPMB, UAN, dan Psikotest)

$$IG (S, Nil.SPMB) = 0,0049$$

$$IG (S, UAN) = 0,26682$$

$$IG (S, Psikotest) = 0,40406$$

Dari ketiga nilai Information Gain diatas Gain (S, Psikotest) adalah yang terbesar sehingga atribut Psikotest merupakan the best classifier dan harus diletakkan sebagai root.

■ Rekursi Level 0 iterasi ke-1

Memanggil fungsi ID3 dengan kumpulan sampel berupa semua sampel data = [8+ , 3-];

Atribut target = 'Diterima' dan kumpulan atribut [nil.SPMB, UAN, Psikotest]

Hitung entropy dan Information Gain untuk menentukan the best classifier dan meletakkannya sebagai root.

Dari penjelasan sebelumnya didapat nilai Information Gain (S, Psikotest) sebagai the best classifier karena IG nya terbesar. Setelah mendapat the best classifier langkah selanjutnya adalah setiap nilai pada atribut Psikotest akan di cek apakah perlu dibuat subtree di level berikutnya atau tidak.. atribut Psikotest, ada 2 sampel (baik dan buruk). Untuk nilai 'Baik' terdapat 6 sampel, berarti sampel baik tidak kosong. Sehingga perlu memanggil fungsi ID3 dengan kumpulan sampel berupa sampel baik = [6+, 0-] , atribut target ='Diterima' dan kumpulan atribut ={ nil.SPMB, Psikotest }

■ Rekursi level 1 iterasi ke 1

Memanggil fungsi ID3 dengan kumpulan sampel berupa sampel baik [6+, 0-] atribut target = 'Diterima' dan kumpulan atribut (nil.SPMB, UAN). Semua sampel baik termasuk dalam kelas "ya" maka fungsi ini akan berhenti dan mengembalikan satu simpul tunggal Root dengan label 'ya' .

■ Rekursi level 0 , Iterasi ke 2

Pada proses rekursi level 0 , iterasi ke 1, sudah dilakukan pengecekan untuk atribut 'Psikotest' dengan nilai 'baik'. Untuk nilai 'buruk', terdapat 5 sampel, berarti Sampel_{buruk} tidak kosong. Sehingga, perlu memanggil fungsi ID3 dengan kumpulan sampel berupa Sampel_{buruk} = [2+, 3-], AtributTarget = 'Diterima', dan KumpulanAtribut = { nil.SPMB, UAN }.

■ Rekursi level 1 iterasi ke 2

Memanggil fungsi ID3 dengan kumpulan sampel berupa Sampel_{buruk} = [2+, 3-], AtributTarget = 'Diterima', dan KumpulanAtribut = { nil.SPMB, Psikotest }. Pada tahap ini dilakukan perhitungan information gain untuk atribut nil.SPMB dan UAN . tetapi kumpulan sampel yang diperhitungkan adalah Sampel_{buruk} dengan 5 sampel data, yaitu [2+ , 3-]. Dengan kata lain

$$S = \text{Sampel}_{\text{buruk}}$$

$$\text{Value (nil.SPMB)} = \text{Tinggi, Sedang, rendah}$$

$$S = \text{Sampel}_{\text{buruk}} = [2+, 3-] , | S | = 5$$

$$S_{\text{Tinggi}} = [1+, 1-] , | S_{\text{Tinggi}} | = 2$$

$$S_{\text{Sedang}} = [1+, 1-] , | S_{\text{Sedang}} | = 2$$

$$S_{\text{Rendah}} = [0+, 1-] , | S_{\text{Rendah}} | = 1$$

Hitung nilai entropy untuk S, S_{Tinggi} , S_{Sedang} , S_{Rendah} dan Information Gain untuk nil.SPMB adalah :

$$\begin{aligned} \text{Entropy (S)} &= - (2/5) \log_2 (2/5) - (3/5) \log_2 (3/5) \\ &= 0,9710 \end{aligned}$$

$$\begin{aligned} \text{Entropy (S}_{\text{Tinggi}}) &= - (1/2) \log_2 (1/2) - (1/2) \log_2 (1/2) \\ &= 1 \end{aligned}$$

$$\begin{aligned} \text{Entropy (S}_{\text{Sedang}}) &= - (1/2) \log_2 (1/2) - (1/2) \log_2 (1/2) \\ &= 1 \end{aligned}$$

$$\begin{aligned} \text{Entropy (S}_{\text{Rendah}}) &= - (0/1) \log_2 (0/1) - (1/1) \log_2 (1/1) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{Gain (S, nil.SPMB)} &= \text{Entropy (S)} - \sum_{ve \text{ (Tinggi,Sedang,Rendah)}} \frac{|S_v|}{|S|} \text{Entropy (S}_v) \\ &= \text{Entropy (S)} - (2/5) \text{Entropy (S}_{\text{Tinggi}}) - (2/5) \text{Entropy (S}_{\text{Sedang}}) - (1/5) \text{Entropy (S}_{\text{Rendah}}) \\ &= 0,9710 - (2/5) 1 - (2/5) 1 - 0 \\ &= 0,1710 \end{aligned}$$

Value (UAN) = Bagus, Cukup, Kurang

S = Sample_{Buruk} = [2+, 3-], |S| = 5, Entropy (S) = 0,9710

S_{Bagus} = [0+, 0-], |S_{Bagus}| = 0

S_{Cukup} = [2+, 1-], |S_{Cukup}| = 3

S_{Kurang} = [0+, 2-], |S_{Kurang}| = 2

Entropy (S_{Bagus}) = 0

$$\begin{aligned} \text{Entropy (S}_{\text{Cukup}}) &= - (2/3) \log_2 (2/3) - (1/3) \log_2 (1/3) \\ &= 0,9183 \end{aligned}$$

Entropy (S_{Kurang}) = 0

$$\begin{aligned} \text{Gain (S, UAN)} &= \text{Entropy (S)} - \sum_{ve \text{ (Tinggi,Sedang,Rendah)}} \frac{|S_v|}{|S|} \text{Entropy (S}_v) \\ &= \text{Entropy (S)} - (0/5) \text{Entropy (S}_{\text{Tinggi}}) - (3/5) \text{Entropy (S}_{\text{Sedang}}) - (2/5) \text{Entropy (S}_{\text{Rendah}}) \\ &= 0,9710 - 0 - (3/5) 0,9183 - 0 \\ &= 0,4200 \end{aligned}$$

Dari kedua nilai Information Gain diatas, Gain (S, UAN) adalah yang terbesar. Sehingga UAN adalah atribut yang merupakan the best classifier dan harus diletakkan sebagai simpul dibawah simpul 'Psikotest' pada cabang nilai 'buruk'. Selanjutnya, setiap nilai pada atribut UAN akan dicek apakah perlu dibuat subtree dilevel berikutnya atau tidak. Untuk nilai 'bagus' (pada kumpulan sample berupa Sample_{Buruk} = [2+, 3-]) terdapat 0 sample berarti Sample_{Bagus} kosong. Sehingga, perlu dibuat satu simpul daun (leaf node, simpul yang tidak punya anak dibawahnya).

Dengan label yang paling sering muncul pada Sample_{Buruk}, yaitu 'Tidak'. Kemudian dilakukan pengecekan untuk atribut 'UAN' bernilai 'Cukup'. Untuk nilai 'Cukup' (pada kumpulan sample berupa Sample_{Buruk} = [2+, 3-]). Terdapat 3 sample, berarti sample 'Cukup' tidak kosong. Sehingga perlu memanggil fungsi ID3 dengan kumpulan Sample berupa Sample_{Cukup} = [2+, 1-], AtributTarget = 'Diterima'. Dan kumpulan Atribut = { nil.SPMB}.

■ Rekursi level 2 iterasi ke-1

Memanggil fungsi ID3 dengan kumpulan Sample berupa Sample_{Cukup} = [2+, 1-] AtributTarget = 'Diterima', dan KumpulanAtribut = {nil.SPMB}. karena

kumpulan atribut hanya berisi satu atribut { yaitu nil.SPMB } , maka atribut yang menjadi the best classifier adalah nil.SPMB dan harus diletakkan sebagai simpul dibawah simpul 'UAN' pada cabang nilai 'Cukup' . selanjutnya setiap nilai pada atribut nilai.SPMB akan dicek apakah dibuat subtree dilevel berikutnya atau tidak. Untuk nilai 'Tinggi' (pada kumpulan berupa $Sample_{Cukup} = [2+, 1-]$), terdapat 1 sampel , berarti $Sample_{Tinggi}$ tidak kosong. Sehingga, perlu memanggil fungsi ID3 dengan kumpulan sample berupa $Sample_{Tinggi} = [1+, 0-]$, AtributTarget = 'Diterima' dan kumpulan atribut = {}.

■ Rekursi level 3 iterasi ke-1

Memanggil fungsi ID3 dengan kumpulan sample berupa $Sample_{Tinggi} = [1+, 0-]$, AtributTarget = 'Diterima' dan kumpulan atribut = {}. Karena semua sample pada $Sample_{Tinggi}$ termasuk dalam kelas 'ya' , maka fungsi ini akan berhenti dan mengembalikan satu simpul tunggal Root dengan label 'ya'.

Selanjutnya, proses akan kembali ke rekursi level 2 iterasi ke-2.

■ Rekursi level 2 iterasi ke-2

Pengecekan atribut nil.SPMB untuk nilai 'Tinggi' sudah dilakukan pada rekursi level 2 itersi ke-1. Selanjutnya, pengecekan dilakukan pada atribut nil.SPMB untuk nilai 'Sedang' . ternyata terdapat 1 sampel pada kumpulan sampel dimana psikotest bernilai 'Buruk' dan UAN bernilai 'Cukup' . karena $Sample_{Sedang}$ tidak kosong maka perlu memanggil fungsi ID3 dengan KumpulanSampel berupa $Sample_{Sedang} = [1+, 0-]$, AtributTarget = 'Diterima', dan KumpulanAtribut = {}.

■ Rekursi level 3 iterasi ke-2

Memanggil fungsi ID3 dengan KumpulanSampel berupa $Sample_{Sedang} = [1+, 0-]$, AtributTarget = 'Diterima', dan KumpulanAtribut = {}. Karena sample pada $Sample_{Sedang}$ termasuk kedalam kelas 'Ya' , maka fungsi ini akan berhenti dan mengembalikan satu simpul tunggal Root dengan label 'Ya'

Selanjutnya proses akan kembali pada rekursi level 2 iterasi ke-3.

■ Rekursi level 2 iterasi ke-3

Pada rekursi level 2 iterasi ke-1 dan ke-2, sudah dilakukan pengecekan atribut nil.SPMB untuk nilai 'Tinggi' dan 'Sedang'. Selanjutnya, pengecekan dilakukan pada Atribut nil.SPMB untuk nilai 'Rendah'. Ternyata terdapat 1 sample pada KumpulanSample dimana Psikotest bernilai 'Buruk' dan UAN bernilai 'Cukup' . karena $Sample_{Rendah}$ tidak kosong, maka perlu memanggil fungsi ID3 dengan KumpulanSample berupa $Sample_{Rendah} = [0+, 1-]$, AtributTarget = 'Diterima', dan KumpulanAtribut = {}.

■ Rekursi level 3 iterasi ke-3

Memanggil fungsi ID3 dengan kumpulanSample berupa $Sample_{Kurang} = [0+, 1-]$, AtributTarget = 'Diterima', dan kumpulanAtribut = {}. Karena semua sample pada $Sample_{Kurang}$ termasuk dalam kelas 'Tidak', maka fungsi ini akan berhenti dan mengembalikan satu simpul tunggal Root dengan label 'Tidak'

■ Rekursi level 1 iterasi ke-3

Pengecekan atribut UAN untuk nilai 'Bagus' dan 'Cukup' yaitu pada rekursi level 1 iterasi ke-2. Selanjutnya pengecekan dilakukan pada atribut UAN untuk nilai 'Kurang' . ternyata terdapat 2 sample pada KumpulanSample dimana Psikotest bernilai

'Buruk' dan UAN bernilai 'Rendah'. Karena SampleRendah tidak kosong sehingga perlu memanggil fungsi ID3 dengan KumpulanSample berupa $Sample_{Rendah} = [0+, 2-]$, AtributTarget = 'Diterima', dan KumpulanAtribut = { IPK }

Selanjutnya proses akan kembali ke rekursi level 2 iterasi ke-4

■ Rekursi level 2 iterasi ke-4

Memanggil fungsi ID3 dengan KumpulanSample berupa $Sample_{Rendah} = [0+, 2-]$, AtributTarget = 'Diterima', dan KumpulanAtribut = { IPK }. Karena semua sample pada $Sample_{Rendah}$ termasuk kedalam kelas 'Tidak', maka fungsi ini akan berhenti dan mengembalikan satu simpul tunggal Root dengan label 'Tidak'

Dilihat dari langkah-langkah diatas bahwa ID3 melakukan strategis pencarian yaitu dimulai dari pohon kosong, kemudian secara progresif berusaha menemukan sebuah pohon keputusan yang mampu mengklasifikasikan sampel-sampel data secara akurat tanpa kesalahan.

Dan dari akhir proses ID3 didapat pohon keputusan akhir di dapat 7 sampel data lain yang belum diketahui sebelumnya, ketujuh sampel itu adalah sebagai berikut :

Sebelumnya kita melakukan penelusuran mulai dari simpul Root menuju ketiga leaf node tersebut. Lakukan operasi conjunction (^) pada setiap simpul yang dikunjungi sampai ditemukan leaf node 'Ya'. Kemudian, lakukan operasi disjunction (v) pada penelusuran tersebut. Dengan demikian 7 sampel yang belum diketahui adalah sebagai berikut :

$(Psikotest = 'Buruk') \wedge (UAN='Bagus') \wedge (nil.SMB='Tinggi') \rightarrow Diterima='Tidak'$

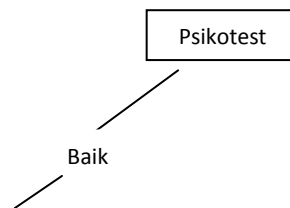
$(Psikotest = 'Baik') \wedge (UAN='Kurang') \wedge (nil.SMB='Tinggi') \rightarrow Diterima='Ya'$

$(Psikotest = 'Buruk') \wedge (UAN='Bagus') \wedge (nil.SMB='Sedang') \rightarrow Diterima='Tidak'$

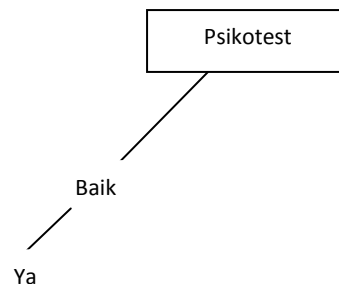
$(Psikotest = 'Baik') \wedge (UAN='Kurang') \wedge (nil.SMB='Sedang') \rightarrow Diterima='Ya'$

POHON KEPUTUSAN

■ Rekursi Level 0 iterasi ke-1



■ Rekursi level 1 iterasi ke 1



$(Psikotest = 'Buruk') \wedge (UAN='Bagus') \wedge (nil.SMB='Rendah') \rightarrow Diterima='Tidak'$

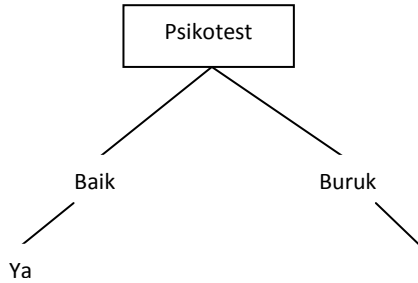
$(Psikotest = 'Baik') \wedge (UAN='Cukup') \wedge (nil.SMB='Rendah') \rightarrow Diterima='Yas'$

$(Psikotest = 'Buruk') \wedge (UAN='Kurang') \wedge (nil.SMB='Rendah') \rightarrow Diterima='Tidak'$

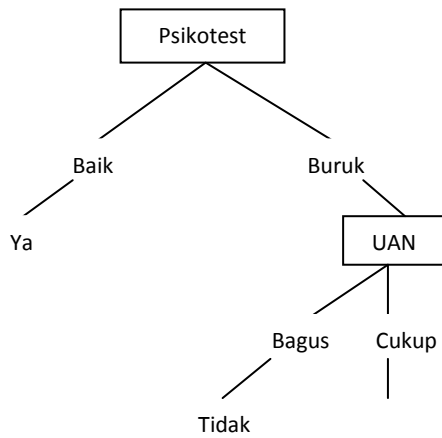
Peserta SPMB	Nil.SPM B	UAN	Psikot est	Diteri ma
P12	Tinggi	Bagus	Buruk	Tidak
P13	Tinggi	Kurang	Baik	Ya
P14	Sedang	Bagus	Buruk	Tidak

P15	Sedang	Kurang	Baik	Ya
P16	Rendah	Bagus	Buruk	Tidak
P17	Rendah	Cukup	Baik	Ya
P18	Rendah	Kurang	Buruk	Tidak

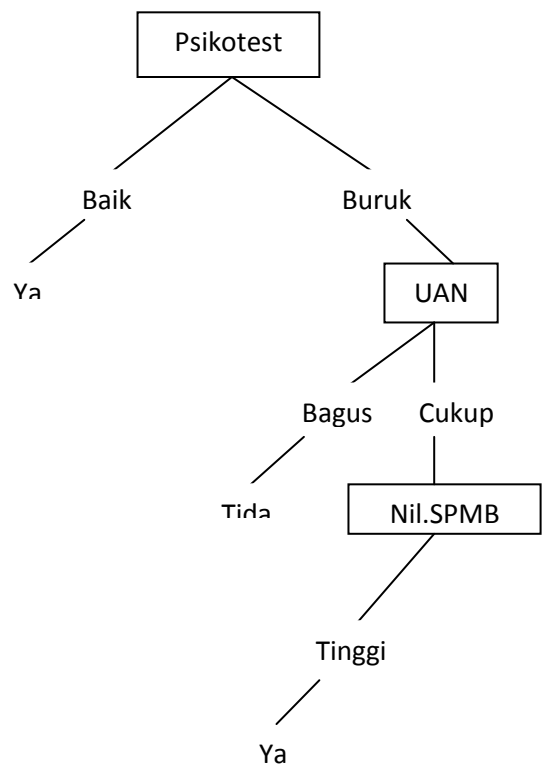
Rekursi level 0 , Iterasi ke 2



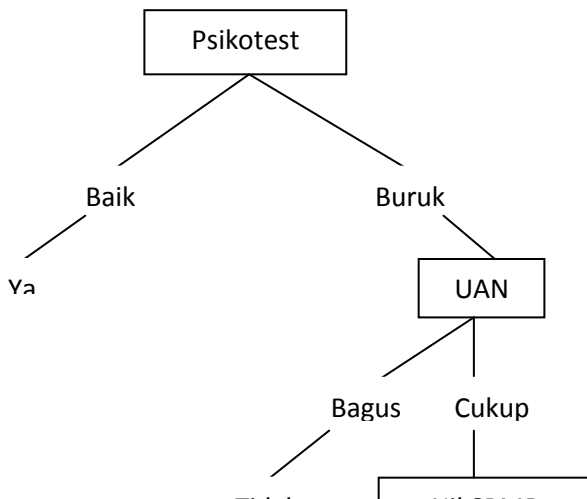
Rekursi level 1 iterasi ke 2



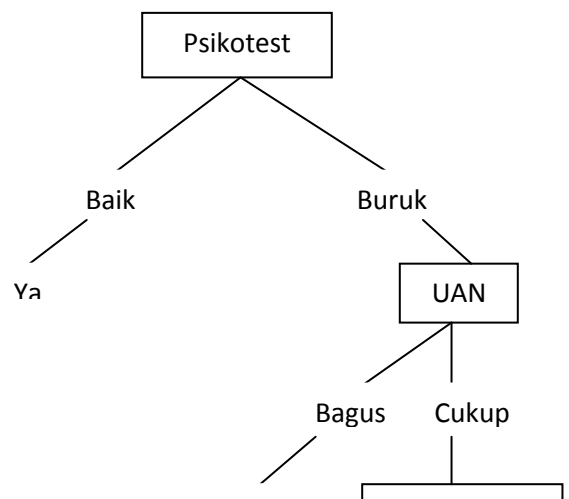
Rekursi level 3 iterasi ke-1



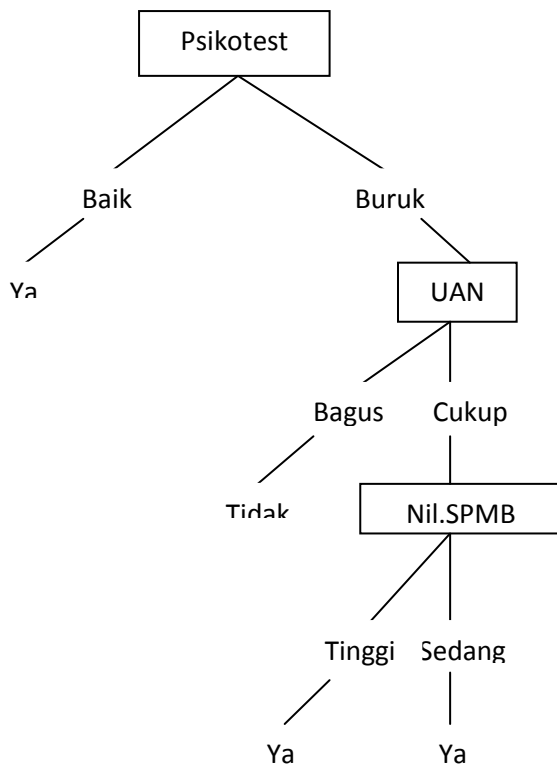
Rekursi level 2 iterasi ke-1



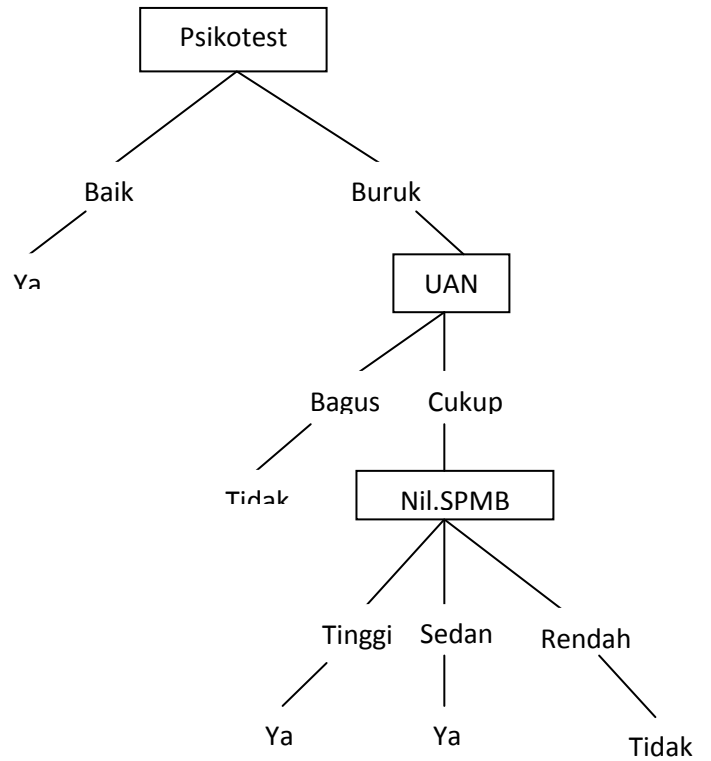
Rekursi level 2 iterasi ke-2



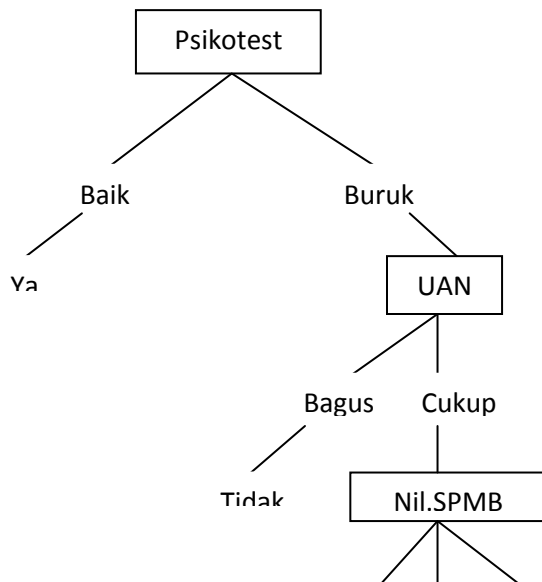
Rekursi level 3 iterasi ke-2



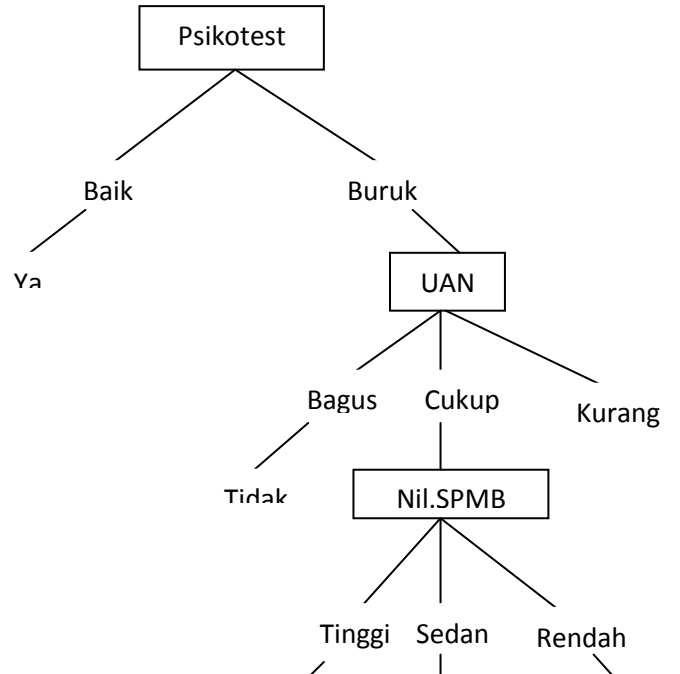
Rekursi level 3 iterasi ke-3



Rekursi level 2 iterasi ke-3



Rekursi level 1 iterasi ke-3

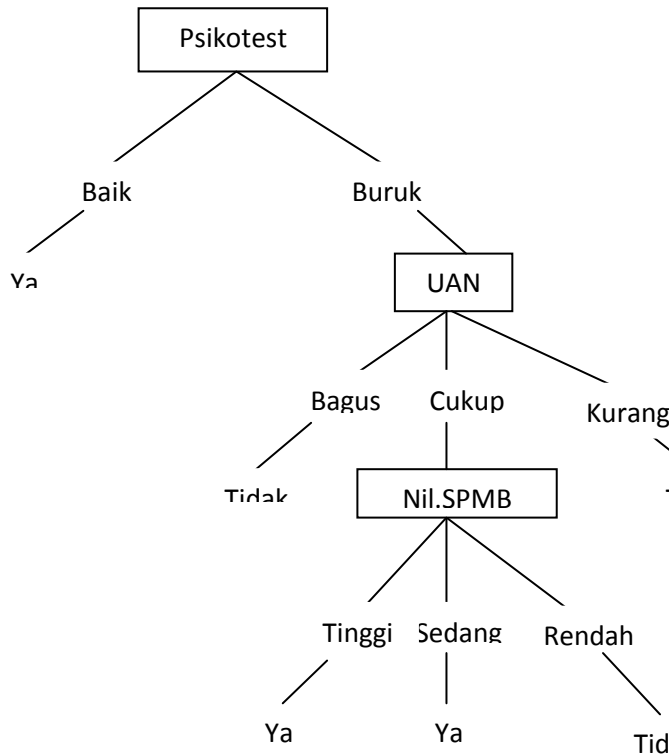


(Psikotest = 'Buruk') ^ (UAN='Bagus') ^
 (nil.SMB=' Rendah') → Diterima='Tidak'

(Psikotest = 'Baik') ^ (UAN=' Cukup') ^
 (nil.SMB=' Rendah') → Diterima=' Yas'

(Psikotest = 'Buruk') ^ (UAN=' Kurang') ^
 (nil.SMB=' Rendah') → Diterima='Tidak'

Rekursi level 2 iterasi ke-4



Pohon keputusan akhir yang dihasilkan oleh fungsi ID3.

Dengan demikian 7 sampel yang belum diketahui adalah sebagai berikut :

(Psikotest = 'Buruk') ^ (UAN='Bagus') ^
 (nil.SMB='Tinggi') → Diterima='Tidak'

(Psikotest = 'Baik') ^ (UAN='Kurang') ^
 (nil.SMB='Tinggi') → Diterima=' Ya'

(Psikotest = 'Buruk') ^ (UAN='Bagus') ^
 (nil.SMB='Sedang') → Diterima='Tidak'

(Psikotest = 'Baik') ^ (UAN='Kurang') ^
 (nil.SMB='Sedang') → Diterima=' Ya'

Peserta SPMB	Nil. SPMB	UAN	Psiko test	Di terima
P12	Tinggi	Bagus	Buruk	Tidak
P13	Tinggi	Kurang	Baik	Ya
P14	Sedang	Bagus	Buruk	Tidak
P15	Sedang	Kurang	Baik	Ya
P16	Rendah	Bagus	Buruk	Tidak
P17	Rendah	Cukup	Baik	Ya
P18	Rendah	Kurang	Buruk	Tidak

5. PENUTUP

SIMPULAN

Berdasarkan pembahasan bab-bab sebelumnya, dapat ditarik kesimpulan sebagai berikut :

1. pohon keputusan dengan algoritma ID3 dapat digunakan untuk memperoleh pengetahuan pada bidang pendidikan khususnya memberikan keputusan dalam hal penerimaan mahasiswa baru
2. seseorang peserta SPMB dinyatakan diterima atau tidak tergantung pada pihak universitas berdasarkan pertimbangan beberapa kriteria.

5.2 SARAN

adapun saran yang diberikan berdasarkan pembahasan sebelumnya adalah : perlu di implementasikan menggunakan bahasa pemrograman tertentu sehingga proses editing data dapat dimungkinkan

6. DAFTAR PUSTAKA

Suyanto, ST, MSc (2007). ARTIFICIAL INTELLIGENCE. Bandung : informatika

Sela, eny itje (2006). KNOWLEDGE DISCOVERY MENGGUNAKAN DECISION TREE. Depok seminar ilmiah nasional komputer dan intelejen [ONLINE]. Tersedia: <http://repository.gunadarma.ac.id/> (04 Oktober 2009).

Munir, rinaldi (2005). MATEMATIKA DISKRIT. Bandung : CV. Informatika Bandung

Setiawan, sandi (1993).ARTIFICIAL INTELEAGENT. Yogyakarta: andi.

Niwanputri, ginar santika (2007). PENGGUNAAAAN POHON DALAM *DECISION TREE*

ANALYSIS UNTUK PENGAMBILAN KEPUTUSAN. Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung [ONLINE]. Tersedia:

<http://www.cs.cmu.edu/afs/cs.cmu.edu/> (04 Oktober 2009).

Iwan syarif, ahmad basuki (2003). DECISION TREE. Politeknik Elektronika Negeri Surabaya [ONLINE]. Tersedia: http://www.pens.ac.id/~basuki/lecture/decisi_ontree.pdf (04 Oktober 2009).

