Materi 5

DATA MINING

3 SKS | Semester 6 | S1 Sistem Informasi | UNIKOM | 2016

Nizar Rabbi Radliya | nizar.radliya@yahoo.com

Nama Mahasiswa	
NIM	
Kelas	

Kompetensi Dasar

Memahami teknik data mining klasifikasi dan mampu menerapkan teknik klasifikasi berbasis *decision tree* menggunakan algoritma C4.5.

Pokok Bahasan

Klasifikasi
1. Konsep klasifikasi
2. Klasifikasi berbasis *decision tree*
3. Pembahasan algoritma

I. Konsep Klasifikasi

Kasifikasi merupakan salah satu teknik dari model prediksi. Teknik ini digunakan untuk pembuatan model yang dapat melakukan pemetaan dari setiap himpunan variabel ke setiap targetnya, kemudian menggunakan model tersebut untuk memberikan nilai target pada himpunan variabel baru yang didapat (nilai targetnya belum tersedia).

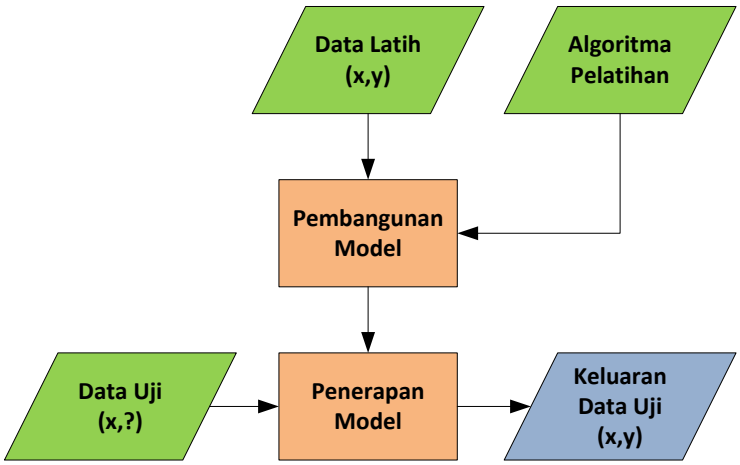
Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu:

1. Pembangunan model sebagai prototipe untuk digunakan pada pekerjaan ke dua.
2. Penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain yang target kelasnya belum diketahui.

Berikut beberapa contoh masalah yang dapat diselesaikan dengan klasifikasi:

1. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
2. Melakukan deteksi penyakit pasien berdasarkan sejumlah nilai parameter penyakit yang diderita.
3. Memprediksi pelanggan mana yang akan berpindah ke kompetitor atau tetap setia.

Framework (kerangka kerja) teknik klasifikasi dalam data mining dapat digambarkan seperti pada gambar 1 di bawah ini.



Gambar 1. Konsep Kerangka Kerja Teknik Klasifikasi

Pada gambar 1 di atas terdapat data latih (x,y) yang target kelasnya sudah diketahui (nilai y sudah ada), dimana data latih ini kita gunakan untuk pembangunan model. Pembangunan model juga akan melibatkan algoritma pelatihan. Pada matari ini model yang dibangun hasilnya dalam bentuk pohon keputusan. Ada beberapa algoritma pelatihan yang digunakan untuk membentuk pohon keputusan, diantaranya ID3, C4.5, CART, dll. Pada materi ini kita akan mempelajari algoritma C4.5 untuk pembentukan pohon keputusan (*decision tree*). Pohon keputusan nanti bisa kita transformasi menjadi sebuah *rule* dalam bentuk seleksi kondisi dan dapat diimplementasikan menggunakan bahasa pemrograman atau SQL untuk pembangunan sistem.

Model yang dihasilkan akan diterapkan untuk memprediksi target kelas (y) pada data uji (x,?) sehingga dihasilkan data uji (x,y). Umumnya model yang dihasilkan dari teknik klasifikasi dapat memprediksi seluruh data latih dengan benar. Tetapi belum tentu mampu memprediksi dengan benar terhadap seluruh data uji. Maka kita bisa menentukan kelayakan penggunaan model dari nilai *accuracy* dan *error rate* (tingkat kesalahan).

Suatu model layak digunakan apabila nilai akurasinya lebih tinggi dari nilai tingkat kesalahannya. Jadi semakin nilai akurasinya tinggi, maka model tersebut semakin layak (berfungsi) untuk diimplementasikan. Berikut persamaan/formula untuk menentukan nilai akurasi dan tingkat kesalahan.

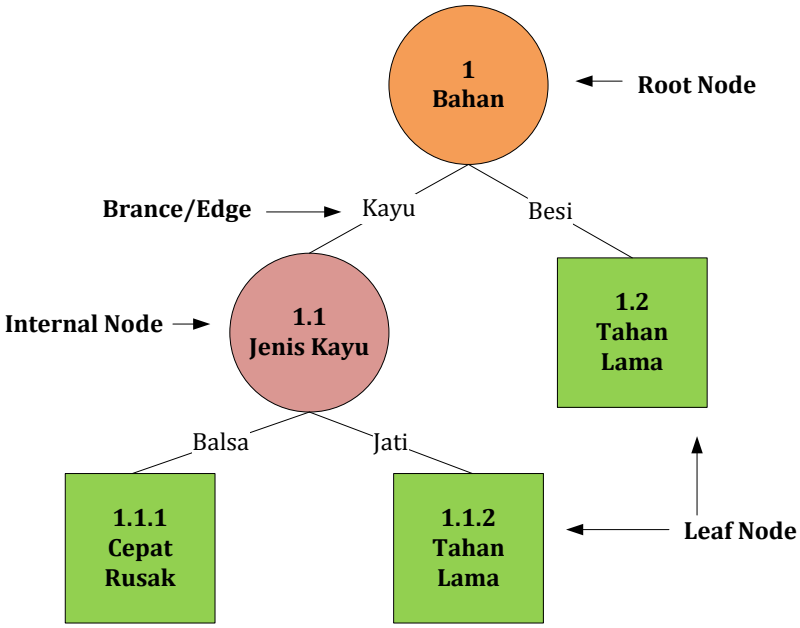
$$Akurasi = \frac{Jumlah\ data\ yang\ prediksinya\ benar}{Jumlah\ data\ yang\ diprediksi}$$

$$Tingkat\ Kesalahan = \frac{Jumlah\ data\ yang\ prediksinya\ salah}{Jumlah\ data\ yang\ diprediksi}$$

II. Klasifikasi Berbasis *Decision Tree*

Pohon keputusan adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. Pohon keputusan banyak dijadikan model dari hasil klasifikasi. Pohon keputusan digunakan untuk memecahkan masalah dengan cara memetakan setiap kondisi nilai pada setiap atribut, sehingga kita dapat melakukan pengambilan keputusan berdasarkan target kelas hasil prediksi.

Pohon (*tree*) merupakan struktur data yang terdiri dari simpul (*node*) dan cabang (*brance*). Simpul (*node*) pada struktur pohon dibedakan menjadi tiga yaitu simpul akar (*root node*), simpul percabangan/internal (*brance/internal node*) dan simpul daun (*leaf node*). Penggambaran pohon keputusan dapat dilihat pada gambar di bawah 2 ini.



Gambar 2. Struktur Pohon Keputusan

Model pohon keputusan dalam teknik klasifikasi dihasilkan dengan penggunaan algoritma pelatihan terhadap data latih. *Root node* dan *internal node* dalam model pohon keputusan dibentuk oleh atribut-atribut yang akan menentukan nilai kelas targetnya (*leaf node*). Nilai *brance/edge* untuk setiap *node* akan ditentukan oleh domain/*instance* dari setiap atribut.

III. Algoritma C4.5

Algoritma C4.5 digunakan untuk pembentukan pohon keputusan. Berikut adalah rangkuman alur algoritma C4.5:

1. Pilih atribut sebagai akar,
2. Buat cabang untuk tiap-tiap nilai,

3. Bagi kasus dalam cabang,
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *gain* digunakan rumus atau persamaan sebagai berikut:

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

Keterangan:

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

$|S_i|$: jumlah kasus pada partisi ke-i (S_i)

|S| : jumlah kasus dalam S

Sedangkan persamaan untuk menghitung *entropy* sebagai berikut:

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

Keterangan:

S : himpunan kasus

A : atribut

n : jumlah partisi S

p_i : proporsi dari S_i terhadap S

IV. Pembahasan Contoh Kasus

Disini kita akan coba menerapkan teknik klasifikasi untuk pembuatan model pohon keputusan dan *rule* dalam melakukan prediksi pelaksanaan bermain tenis. Pada kasus ini penentuan keputusan bermain tenis atau tidak, sementara ditentukan oleh 4 atribut, yaitu diantranya:

1. Atribut *outlook* (domain: *sunny*, *cloudy*, *rainy*);
2. Atribut *temperature* (domain: *hot*, *mild*, *cool*);
3. Atribut *humidity* (domain: *high*, *normal*);
4. Atribut *windy* (domain: *true*, *false*).

Pada kasus ini yang menjadi target kelas adalah atribut play, sehingga terdapat dua target kelas yaitu Tidak/No (S_1) dan Ya/Yes (S_2). Jumlah objek/kasus yang kita jadikan

sebagai data latih adalah sebanyak 14 kasus. Data latih yang akan kita gunakan dapat dilihat pada tabel 1 di bawah ini.

Tabel 1. Data Latih

No	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Cloudy	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	Yes
7	Cloudy	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Cloudy	Mild	High	True	Yes
13	Cloudy	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Pembentukan model pohon keputusan menggunakan algoritma C4.5. Dalam algoritma C4.5, pemilihan atribut sebagai akar didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk memudahkan perhitungan gain data latih dipetakan terlebih dahulu seperti pada tabel 2 di bawah ini.

Tabel 2. Hasil Perhitungan *Gain* Untuk Penentuan *Root Node* (*Node* 1)

Node			Jml Kasus (S)	Tidak (S_1)	Ya (S_2)	Entropy	Gain
1	Total		14	4	10	0.863	
	Outlook						0.259
		Cloudy	4	0	4	0.000	
		Rainy	5	1	4	0.722	
		Sunny	5	3	2	0.971	
	Temperature						0.184
		Cool	4	0	4	0.000	
		Hot	4	2	2	1.000	
		Mild	6	2	4	0.918	
	Humidity						0.371
		High	7	4	3	0.985	
		Normal	7	0	7	0.000	
	Windy						0.006
		False	8	2	6	0.811	
		True	6	4	2	0.918	

Sebelum mencari nilai *gain*, kita cari dulu nilai *entropy* total dan *entropy* semua domain pada setiap atribut. Berikut adalah perhitungan nilai *entropy* total.

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

Entropy(Total) $= (-\frac{S_1}{S} *\ log_2(\frac{S_1}{S}))+ (-\frac{S_2}{S} *\ log_2(\frac{S_2}{S}))$

Entropy(Total) $= (-\frac{4}{14} *\ log_2(\frac{4}{14}))+ (-\frac{10}{14} *\ log_2(\frac{10}{14}))$

Entropy(Total) = 0.863

Setelah nilai *entropy* kita temukan, maka langkah selanjutnya kita hitung nilai *gain* untuk setiap atribut. Berikut adalah perhitungan *gain* untuk atribut *outlook*.

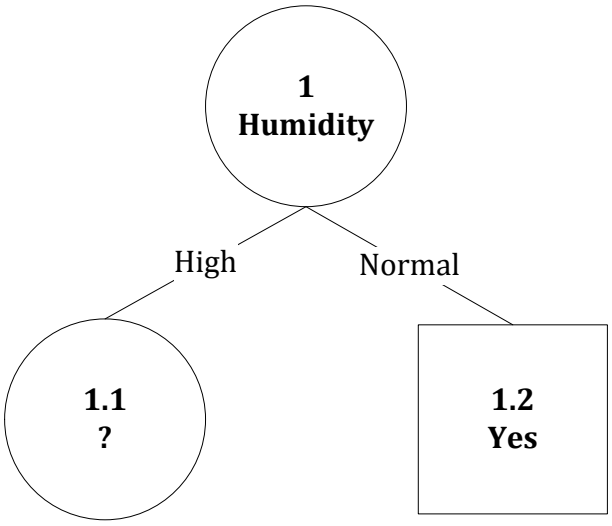
Gain(S,A) $= Entropy(S) - \sum_{i=1}^{n} \frac{|\ S_i\ |}{|\ S\ |} * Entropy(S_i)$

Gain(Total,Outlook) $= Entropy(Total) - \sum_{i=1}^{n} \frac{|\ Outlook_i\ |}{|\ Total\ |} * Entropy(Outlook_i)$

Gain(Total,Outlook) $= 0.863 - ((\frac{4}{14} * 0)+ (\frac{5}{14} * 0.722)+ (\frac{5}{14} * 0.971))$

Pada perhitungan *gain* untuk semua kasus, ditemukan nilai *gain* tertinggi adalah atribut *humidity*. Maka atribut *humidity* kita jadikan sebagai *root node* (*node* 1). Untuk *brance*/*edge* pada *root node* ditentukan oleh domain. Lalu *node* untuk setiap *brance*/*edge* kita jadikan sebagai *leaf node* apabila semua kasus untuk domain tersebut ada pada target kelas yang sama. Contohnya domain *normal* semua kasus masuk ke dalam target kelas Ya/Yes (S_2).

Pohon keputusan yang dihasilkan dari perhitungan *gain* pada semua kasus dapat dilihat pada gambar 3 di bawah ini.



Gambar 3. Pohon Keputusan Berdasarkan Semua Kasus Data Latih

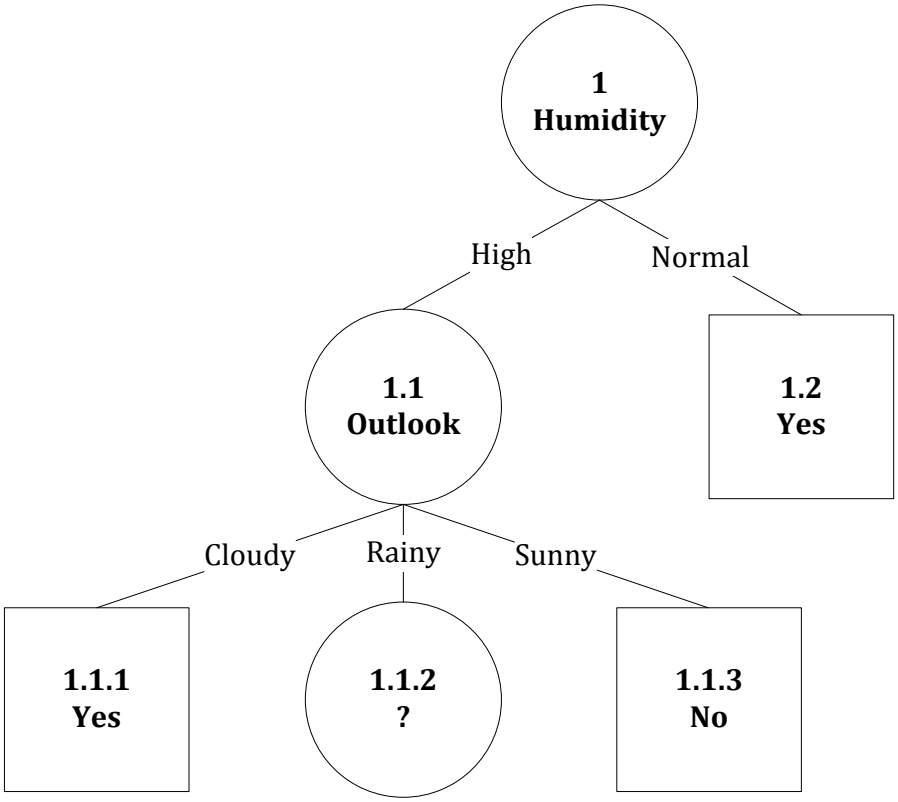
Apabila semua kasus pada domain tidak pada target yang sama maka kita jadikan sebagai *internal node* (contoh: *node* 1.1), dimana atribut yang mengisi *node* tersebut kita

tentukan sesuai hasil perhitungan nilai *gain* tertinggi dari beberapa kasus tertentu. Maksud dari beberapa kasus tertentu adalah kasus yang memenuhi/memiliki nilai domain yang target kelasnya tidak sama. Contohnya pada kasus yang atribut *humidity* bernilai *high*, tidak memiliki target kelas yang sama. Maka perhitungan *gain* berikutnya hanya melibatkan kasus yang atribut *humidity* bernilai *high*. Hasil dari perhitungan *gain* untuk penentuan *internal node* dapat dilihat pada tabel 3 di bawah ini.

Tabel 3. Hasil Perhitungan *Gain* Untuk Penentuan *Internal Node 1.1*

Node			**Jml Kasus (S)**	**Tidak (S_1)**	**Ya (S_2)**	**Entropy**	**Gain**
1.1	Humidity:High		7	4	3	0.985	
	Outlook						0.700
		Cloudy	2	0	2	0.000	
		Rainy	2	1	1	1.000	
		Sunny	3	3	0	0.000	
	Temperature						0.020
		Cool	0	0	0	0.000	
		Hot	3	2	1	0.918	
		Mild	4	2	2	1.000	
	Windy						0.020
		False	4	2	2	1.000	
		True	3	2	1	0.918	

Hasil dari perhitungan gain dari kasus data latih yang atribut *humidity* bernilai *high*, menyatakan nilai *gain* tertinggi adalah atribut *outlook*. Semua kasus data latih yang atribut *humidity* bernilai *high* dan atribut *outlook* bernilai *cloudy*, masuk ke dalam target kelas Ya/Yes (S_2) dan yang bernilai *sunny* semua kasusnya masuk ke dalam target kelas Tidak/No (S_1). Dengan begitu maka *brance cloudy* dan *sunny* akan menghasilkan *leaf node*. Sedangkan domain *rainy* akan menghasilkan *internal node*, karena semua kasusnya tidak masuk pada target kelas yang sama. Pohon keputusan yang dihasilkan dari perhitungan *gain* pada kasus yang atribut *humidity* bernilai *high* dapat dilihat pada gambar 4 di bawah ini.



Gambar 4. Pohon Keputusan Berdasarkan Kasus yang Atribut *Humidity* Bernilai *High*

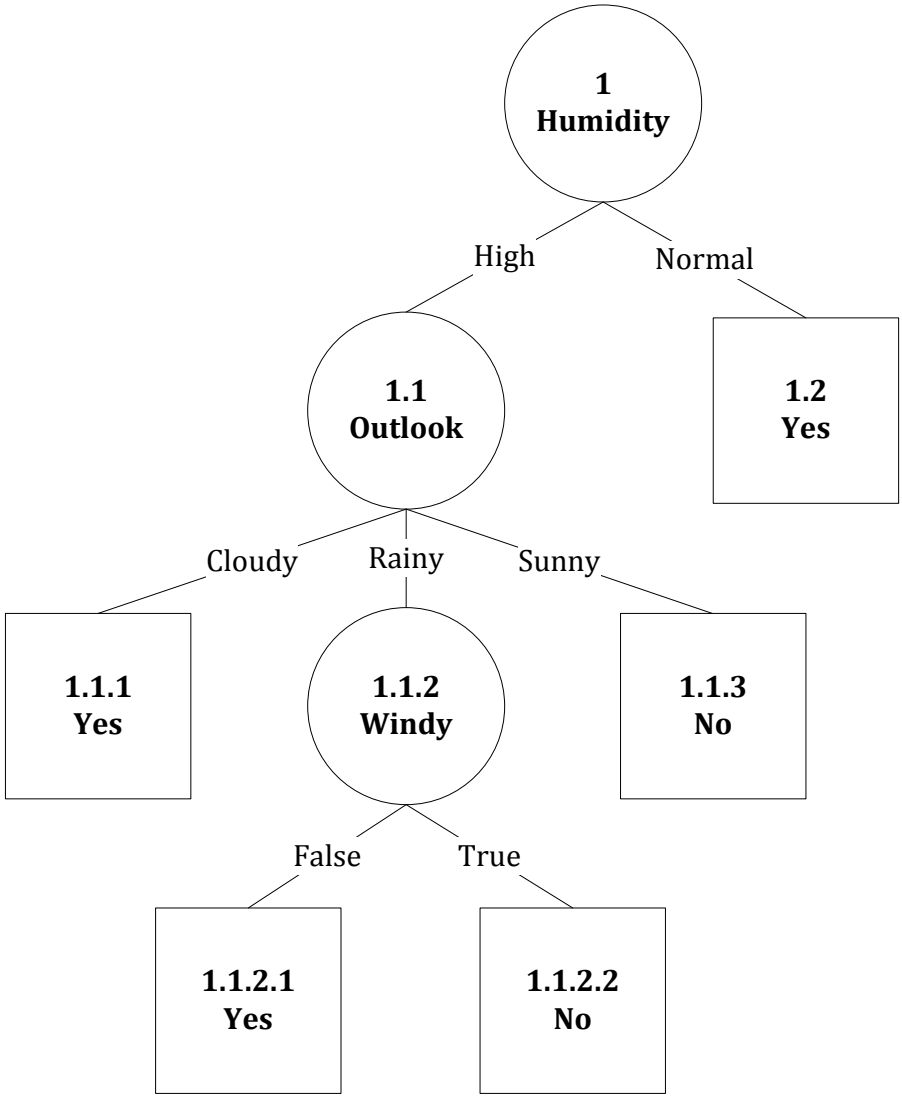
Penentuan *internal node* pada *brance rainy* akan dilakukan perhitungan *gain* pada kasus yang atribut *humidity* bernilai *high* dan atribut *outlook* bernilai *rainy*. Hasil dari perhitungan *gain* tersebut dapat dilihat pada tabel 4 di bawah ini.

Tabel 4. Hasil Perhitungan *Gain* Untuk Penentuan *Internal Node 1.1.2*

Node			Jml Kasus (S)	Tidak (S_1)	Ya (S_2)	Entropy	Gain
1.1.2	Humidity:High & Outlook:Rainy		2	1	1	1.000	
	Temperature						0.000
		Cool	0	0	0	0.000	
		Hot	0	0	0	0.000	
		Mild	2	1	1	1.000	
	Windy						1.000
		False	1	0	1	0.000	
		True	1	1	0	0.000	

Hasil dari perhitungan gain dari kasus data latih yang atribut *humidity* bernilai *high* dan atribut *outlook* bernilai *rainy*, menyatakan nilai *gain* tertinggi adalah atribut *windy*. Semua kasus data latih yang atribut *humidity* bernilai *high* dan atribut *outlook* bernilai

rainy serta atribut *windy* bernilai *false*, masuk ke dalam target kelas Ya/Yes (S_2) dan yang bernilai *true* semua kasusnya masuk ke dalam target kelas Tidak/No (S_1). Dengan begitu maka *brance false* dan *true* akan menghasilkan *leaf node*. Pohon keputusan yang dihasilkan dari perhitungan *gain* pada kasus yang atribut *humidity* bernilai *high* dan atribut *outlook* bernilai *rainy* dapat dilihat pada gambar 5 di bawah ini.



Gambar 5. Pohon Keputusan Berdasarkan Kasus yang Atribut *Humidity* Bernilai *High* dan Atribut *Outlook* Bernilai *Rainy*

Pada pohon keputusan di gambar 5, semua cabang sudah diakhiri dengan *leaf node*. Maka dari itu pohon keputusan ini sudah bisa kita gunakan. Sebaiknya pertama kita lakukan pemetaan dari semua kasus data latih pada pohon keputusan tersebut. Apabila semua kasus sudah sesuai dengan target kelasnya masing-masing, maka pohon keputusan tersebut bisa kita gunakan untuk data uji.

Pohon keputusan juga bisa kita transformasi ke dalam bentuk rule yang nantinya bisa kita jadikan acuan dalam pembangunan sistem/program. Berikut hasil transformasi dari pohon keputusan ke dalam bentuk rule dengan mengunakan algoritma seleksi kondisi bersarang/bertingkat.

```
If (humidity = "normal") {
      play = "yes";
} else if (humidity = "high") {
      if (outlook = "cloudy") {
           play = "yes";
      } else if (outlook = "sunny") {
           play = "no";
      } else if (outlook = "rainy") {
           if (windy = "false") {
                play = "yes";
           } else if (windy = "true") {
                play = "no";
           }
      }
}
```

V. Daftar Pustaka

[1] Astuti, F.A. 2013. Data Mining. Yogyakarta: Andi.
[2] Kusrini & Taufiz, E.L. 2009. Algoritma Data Mining. Yogyakarta: Andi.
[3] Prasetyo, E. 2012. Data Mining: Konsep dan Aplikasi Menggunakan MATLAB. Yogyakarta: Andi.
[4] Prasetyo, E. 2014. Data Mining: Mengolah Data Menjadi Informasi Menggunakan MATLAB. Yogyakarta: Andi.

VI. Materi Berikutnya

Pokok Bahasan	Klasifikasi
Sub Pokok Bahasan	1. Klasifikasi berbasis *nearest neighbor* 2. Pembahasan algoritma