

# **TUGAS DATA MINING**

## **PERTEMUAN 7**



**NAMA : THORIQ HAFIDZUZUMAR**

**NIM : A11.2019.11729**

**KELAS : A11.4611**

**UNIVERSITAS DIAN NUSWANTORO**

**FAKULTAS ILMU KOMPUTER**

**2022**

**Latihan Soal (Kuis)**

1. Hitung Entropy dan Gain serta tentukan pohon keputusan yang terbentuk dari contoh kasus keputusan bermain tenis dibawah ini :

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
Sunny	Hot	High	No	Don't Play
Sunny	Hot	High	Yes	Don't Play
Cloudy	Hot	High	No	Play
Rainy	Mild	High	No	Play
Rainy	Cool	Normal	No	Play
Rainy	Cool	Normal	Yes	Play
Cloudy	Cool	Normal	Yes	Play
Sunny	Mild	High	No	Don't Play
Sunny	Cool	Normal	No	Play
Rainy	Mild	Normal	No	Play
Sunny	Mild	Normal	Yes	Play
Cloudy	Mild	High	Yes	Play
Cloudy	Hot	Normal	No	Play
Rainy	Mild	High	Yes	Don't Play

36

Terdapat kriteria yang diperlukan meliputi:

1. Cuaca
2. Angin
3. Kelembaban
4. Temperatur udara

Salah satu atribut merupakan data solusi per item data yang disebut target atribut -> misalnya atribut “play” dengan nilai “main” atau “tidak main”

Atribut memiliki nilai-nilai yang dinamakan “instance”

Misalkan atribut “Cuaca” memiliki instance -> cerah, berawan, dan hujan.

Berikut Tabel :

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

Berdasarkan tabel diatas akan dibuat tabel keputusan untuk menentukan main tenis atau tidak dengan melihat keadaan Outlook (cuaca), Temperatur, Humidity (kelembaban), dan windy (keadaan angin).

Algoritma secara umum:

- Pilih atribut sebagai akar
- Buat cabang untuk tiap2 nilai
- Bagi kasus dalam cabang
- Ulangi proses utk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama
- Memilih atribut berdasarkan nilai “gain” tertinggi dari atribut-atribut yang ada.

Perhitungan Gain :

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

Keterangan:

- S : himpunan
- A : atribut

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

Menghitung Nilai Entropy :

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

Keterangan:

- S : himpunan kasus
- A : fitur
- n : jumlah partisi S
- $p_i$  : proporsi dari  $S_i$  terhadap S

Perincian algoritma ( langkah 1)

- Menghitung jumlah kasus seluruhnya, jumlah berkeputusan “Yes” maupun “No”.
- Menghitung Entropy dari semua kasus yg terbagi berdasarkan atribut “Outlook”, “Temperature”, “Humidity”, “Windy”.
- Lakukan penghitungan Gain utk setiap atributnya

Perhitungan :

Node			Jumlah Kasus (S)	Tidak (S <sub>1</sub> )	Ya (S <sub>2</sub> )	Entropy	Gain
1	TOTAL		14	4	10	0,863120569	
	OUTLOOK						0,258521037
		CLOUDY	4	0	4	0	
		RAINY	5	1	4	0,721928095	
		SUNNY	5	3	2	0,970950594	
	TEMPERATURE						0,183850925
		COOL	4	0	4	0	
		HOT	4	2	2	1	
		MILD	6	2	4	0,918295834	
	HUMIDITY						0,370506501
		HIGH	7	4	3	0,985228136	
		NORMAL	7	0	7	0	
	WINDY						0,005977711
		FALSE	8	2	6	0,811278124	
		TRUE	6	2	4	0,918295834	

Perhitungan Total Entropy :

Node			Jumlah Kasus (S)	Tidak (S <sub>1</sub> )	Ya (S <sub>2</sub> )	Entropy	Gain
1	TOTAL		14	4	10	0,863120569	
...	...	...	...	...	...	...	...

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

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

Menghitung gain pada baris Outlook

$$\begin{aligned}
 &\text{Gain (Total, Outlook)} \\
 &= \text{Entropy(Total)} - \sum_{i=1}^n \frac{|\text{Outlook}_i|}{|\text{Total}|} * \text{Entropy(Outlook}_i) \\
 &= 0,863120569 - ((4/14 * 0) + (5/14 * 0,72) + (5/14 * 0,97)) \\
 &= 0,258521037
 \end{aligned}$$

Lakukan Hitung Gain untuk temperature, humidity dan windy

Sepert yg terlihat pd tabel, diperoleh bhw atribut dgn Gain tertinggi adalah Humidity -> 0,37

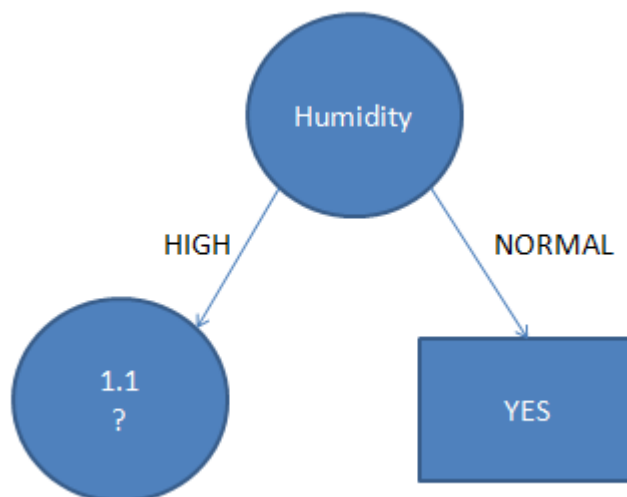
Maka Humidity menjadi node akar

Humidity memiliki dua nilai yaitu “High” dan “Normal”

Humidity -> “Normal” sdh mengklasifikasikan kasus menjadi 1 yaitu keputusannya “yes”

Untuk humidity -> “High” msh perlu dilakukan perhitungan lagi (karena masih terdapat “yes” dan “no”)

Pohon Keputusan Node 1



## Perincian Algoritma (Langkah 2)

- Pilih node akar “High” dan hitung:
  - Jumlah kasus
  - Jumlah kasus keputusan “Yes”
  - Jumlah kasus keputusan “No”
  - Entropy



Atribut:

- Outlook
- Temperature
- Windy

## Hasil perhitungan (Langkah 2)

Node			Jumlah Kasus (S)	Tidak (S <sub>i</sub> )	Ya (S <sub>y</sub> )	Entropy	Gain
1.1	HUMIDITY HIGH		7	4	3	0,985228136	
	OUTLOOK						0,69951385
		CLOUDY	2	0	2	0	
		RAINY	2	1	1	1	
		SUNNY	3	3	0	0	
	TEMPERATURE						0,020244207
		COOL	0	0	0	0	
		HOT	3	2	1	1	
		MILD	4	2	2	2	
	WINDY						0,020244207
		FALSE	4	2	2	1	
		TRUE	3	2	1	0,918295834	

## Hasil perhitungan (Langkah 2)

Didapat Gain tertinggi -> outlook -> 0,69

Maka “Outlook” menjadi node cabang dari atribut humidity yg bernilai “High”

Berdasarkan atribut “Outlook” terdpt 3 nilai

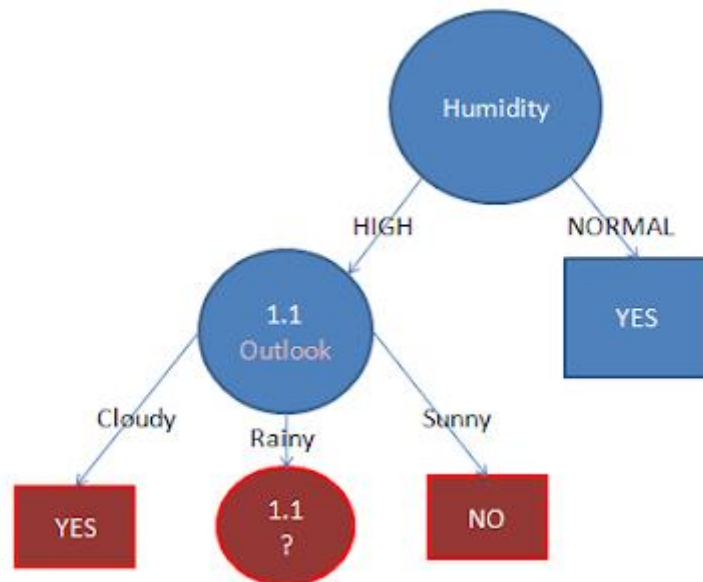
Cloudy

Rainy

Sunny

Krn “Cloudy” pasti bernilai “Yes” dan “Sunny” pasti bernilai “No”, maka tdk perlu dilakukan perhitungan lagi  
Sedangkan “Rainy” bernilai “yes” dan “No”, maka masih perlu dilakukan perhitungan lagi

Pohon keputusan node 1.1



Perincian algoritma (Langkah 3)



Hasil perhitungan (Langkah 3)



Node			Jumlah Kasus (S)	Tidak (S <sub>+</sub> )	Ya (S <sub>-</sub> )	Entropy	Gain
1.1	HUMIDITY "HIGH" dan OUTLOOK "RAINY"		2	1	1	1	
	TEMPERATURE						0
		COOL	0	0	0	0	
		HOT	0	0	0	0	
		MILD	2	1	1	1	
	WINDY						1
		FALSE	1	0	1	0	
		TRUE	1	1	0	0	

Hasil perhitungan (Langkah 3)

Didapat Gain tertinggi -> Windy -> 1

Maka "Windy" menjadi node cabang dari atribut humidity yg bernilai "High" dan outlook yg bernilai "Rainy"

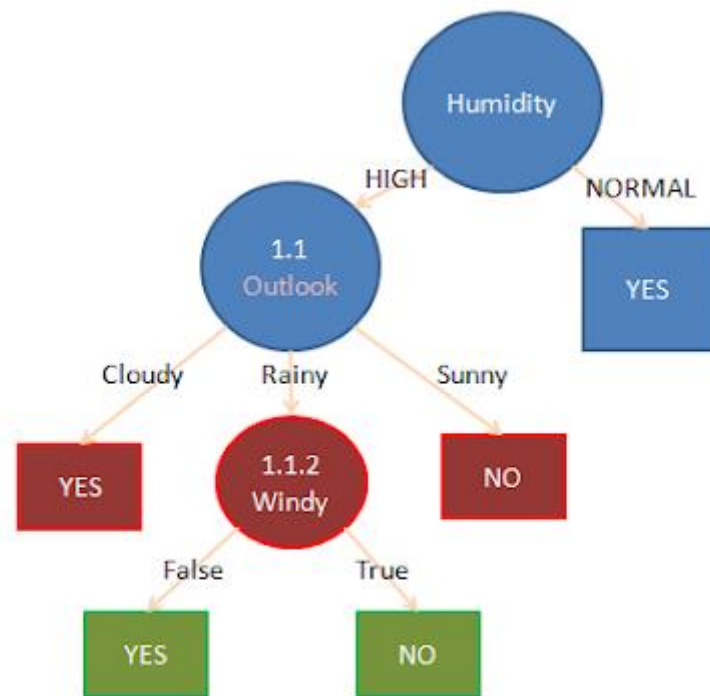
Berdasarkan atribut "Windy" terdpt 2 nilai

True

False

Karena "True" sdh terklasifikasi pasti bernilai "No" dan "False" pasti bernilai "Yes", maka tidak perlu dilakukan perhitungan lagi

Pohon keputusan node 1.1.2



Hasil perhitungan (Langkah 3)

Berdasarkan node 1.1.2, maka:

“Semua kasus sudah masuk dapat kelas”

Sehingga pohon keputusan diatas merupakan pohon keputusan terakhir yang terbentuk