

Penerapan Pohon Keputusan dengan Algoritma KNN Dalam Mencari Teman Belajar Terbaik

Kevin Katsura Dani Sitanggung - 13519216

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13519216@std.stei.itb.ac.id

Abstract—Pohon Keputusan (Decision Tree) mencakup banyak hal dalam kehidupan manusia. Setiap pilihan yang tersedia selalu didampingi dengan keputusan yang harus kita tentukan. Dalam penerapan Pohon Keputusan, penentuan keputusan akhir didapat dengan perbandingan perbandingan yang telah terpenuhi. Dalam Algoritma KNN (K-Nearest Neighbors), penentuan hasil terdekat didasarkan pada pohon keputusan. Untuk mencari teman belajar terbaik, kita dapat mencari kecocokan cara belajar dengan orang lain yang terdaftar. Dengan Algoritma KNN ini, kita bisa menentukan orang mana yang cocok untuk menjadi teman belajar kita.

Keywords— pohon, keputusan, jarak, loss, k-nearest neighbors.

I. PENDAHULUAN

Penerapan Pohon Keputusan tidaklah jauh dari kehidupan kita. Berbagai keputusan yang kita ambil merupakan bentuk dari penerapan pohon keputusan. Penggunaan Pohon Keputusan dalam kehidupan nyata tidak lagi menjadi hal yang minor sehingga pada dasarnya, Pohon Keputusan tidak menjadi hal yang asing dalam penerapannya.

Dengan Menggunakan Pohon Keputusan, kita dimungkinkan untuk mendapatkan keputusan yang sesuai dengan komponen penentu yang kita tetapkan dalam berbagai persoalan. Salah satu problema yang dapat ditangani adalah penentuan teman belajar terbaik.

Dalam menjalani proses pembelajaran di kampus maupun sekolah, kita kerap mendapatkan kesulitan dalam menerima materi pembelajarannya. Hal itu menyebabkan kita perlu menambah waktu belajar kita di luar pembelajaran wajib sekolah. Untuk mendukung pembelajaran ekstra tersebut, kita membutuhkan teman yang dapat mendukung proses belajar itu. Oleh karena itu, kita perlu mencari teman belajar yang sesuai dengan kita.

Penentuan teman belajar yang tepat akan membantu kita dalam proses belajar sebab gaya belajar yang sejalan akan mempermudah kita dalam mencapai target-target pembelajaran kita. Teman belajar ini bisa kita dapat dengan melakukan perbandingan cara belajar kita dengan cara belajar siswa lain yang juga mencari teman belajar. Dengan begitu, penentuan teman belajar yang sesuai akan berimplikasi pada pembelajaran yang lebih efektif.

Algoritma KNN merupakan algoritma yang berguna dalam penentuan data yang memiliki nilai terdekat dengan data yang diinginkan pengguna. Algoritma ini berjalan dengan mengukur

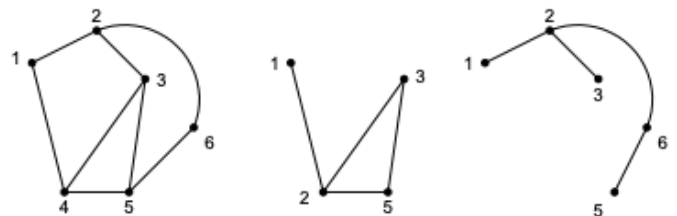
jarak antara data yang diinginkan pengguna dengan data target yang dihitung berdasarkan Euclidean Distance kemudian akan disimpulkan data dengan nilai terdekat. Pada makalah ini, penggunaan algoritma KNN dalam penentuan teman belajar terbaik akan menampilkan data mahasiswa dengan kecocokan yang paling sempurna dengan user sehingga teman belajar terbaik yang dicari akan ditemukan.

II. LANDASAN TEORI

A. Pohon

1. Definisi Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Secara garis besar, pohon merupakan graf terhubung yang tidak memiliki sisi ganda dan tidak memuat sirkuit. Suatu pohon dapat dibentuk dari upagraf suatu graf. Oleh karena itu, pohon merupakan suatu bentuk penerapan dari graf.



Gambar 2.1.pohon (gambar terkanan) dari graf 6 simpul (gambar ter kiri) (sumber : .

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm>)

Pohon merupakan graf yang memiliki syarat-syarat tertentu yang harus terpenuhi. Untuk membentuk pohon dari suatu graf dapat dilakukan dengan menetapkan upagraf yang memenuhi syarat-syarat sebagai berikut.

1. Setiap simpul di dalam graf terhubung dengan lintasan tunggal.
2. Jumlah sisi dalam graf berjumlah $n-1$ buah sisi dengan n merupakan banyak simpul pada graf.
3. Graf tidak mengandung sirkuit dan setiap penambahan satu sisi pada graf akan membentuk hanya satu sirkuit.

Sifat sifat ini juga dijelaskan dalam Teorema 1 graf pohon. Adapun isi dari Teorema 1 graf pohon adalah sebagai berikut.

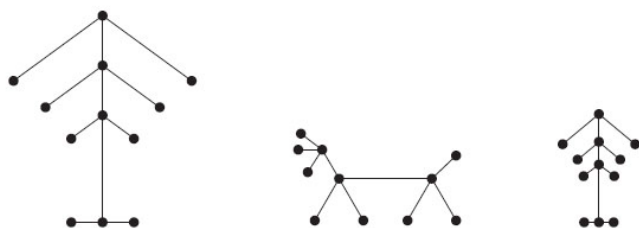
TEOREMA 1

Pernyataan di bawah ini adalah ekuivalen

- #1. Graf T_T adalah graf pohon
- #2. Terdapat lintasan tunggal untuk sebarang dua titik yang berbeda.
- #3. Graf T_T merupakan terhubung minimal, yaitu graf T_T terhubung akan tetapi $T - e_{T-e}$ bukan graf terhubung untuk setiap sisi e_e di T .
- #4. Graf T_T adalah graf tak bersikuit yang maksimal, yaitu T_T tak memuat sirkuit namun $T + xy_{T+xy}$ memiliki dengan $x, y \in T, x, y \in T$ adalah sebarang dua titik yang bertetangga.

2. Struktur Pohon

Struktur pada pohon menyerupai struktur pada graf. Pada pohon terdapat simpul sebagai titik temu sisi dan terdapat sisi sebagai pemisah antara simpul. Pada gambar di bawah ini, simpul direpresentasikan oleh setiap dot dan sisi direpresentasikan oleh setiap garis penghubung antara simpul.

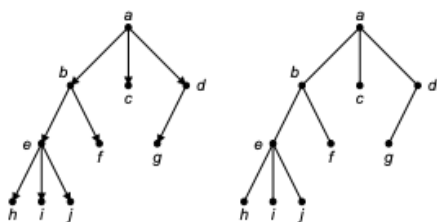


Gambar 2.2. pohon (sumber :

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm>)

3. Pohon Berarah (rooted tree)

Pohon berarah (rooted tree) merupakan pohon yang satu simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah.



(a) Pohon berakar (b) sebagai perjanjian, tanda panah pada sisi dapat dibuang

Gambar 2.3. pohon berakar (sumber :

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm>)

Setelah mengatur satu titik sebagai akar, kita dapat membuat sisi berarah dari graf pohon tersebut yang mengarah keluar dari akar tersebut.

4. Terminologi Pohon Berakar

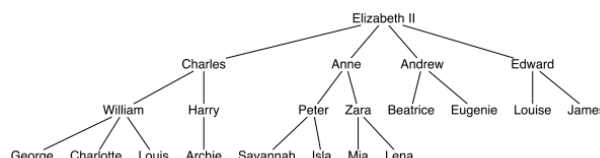
- i. Anak (child atau children), Saudara, dan Orang tua (parents)
Anak merupakan simpul yang berada pada 1 tingkat di bawah

orang tua. Hal ini juga berarti bahwa simpul yang tidak memiliki orang tua merupakan simpul yang bukan anak dari suatu simpul. Pada gambar 2.3, simpul b, c, dan d merupakan anak-anak dari simpul a dan a merupakan orang tua dari anak-anak tersebut.

Saudara merupakan anak-anak yang memiliki orang tua yang sama. Pada gambar 2.3, simpul b, c, dan d merupakan saudara karena memiliki simpul orang tua yang sama yaitu a.

Pada penerapan pohon berakar, akar pohon mewakili suatu nilai. Hal ini berarti masing-masing orang tua dan anak memiliki nilai yang berbeda-beda. Nilai dari suatu orang tua dengan anaknya akan saling berhubungan.

Contohnya : Pohon keluarga bangsawan Inggris



Gambar 2.4. Pohon yang memiliki nilai pada akhirnya (sumber : <https://stei19.kuliah.itb.ac.id/course/view.php?id=1070>)

Pada gambar 2.4, Jika Elizabeth II merupakan a dan Charles, Anne, Andrew dan Edward merupakan b, maka a adalah orang tua dari b dan b adalah anak-anak dari a.

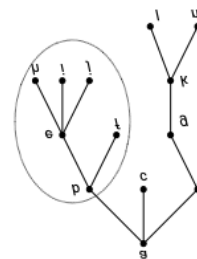
ii. Lintasan (path)

Lintasan merupakan jalur yang dilalui dari suatu simpul ke simpul lain. Dalam penentuan suatu lintasan, list dari simpul-simpul yang dilewati maupun sisi-sisi yang dilewati dalam pencapaian simpul tujuan merupakan representasi lintasan.

Pada gambar 2.3, lintasan dari a ke h adalah a, b, e, h sehingga panjang dari lintasan tersebut sama dengan jumlah sisi yang dilalui untuk mencapai simpul yaitu 3 satuan.

iii. Upapohon (subtree)

Upapohon merupakan sub pohon dari suatu pohon.



gambar 2.5. Upapohon (sumber :

<http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm>)

iv. Derajat (degree)

Derajat sebuah simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut. Hal ini berarti bahwa derajat dari pohon merupakan jumlah derajat keluar dari suatu simpul. Pada gambar 2.3.b, derajat dari simpul a, dan e adalah 3, simpul b adalah 2, simpul d adalah 1, dan simpul c, h, i, dan j adalah 0.

Dengan begitu, derajat dari suatu daun pada pohon selalu 0 karena tidak terdapat sisi berarah keluar dari setiap daun.

v. Daun (leaf)

Daun merupakan simpul yang memiliki jumlah derajat sama dengan 0 atau tidak memiliki anak. Pada gambar 2.3, simpul h,

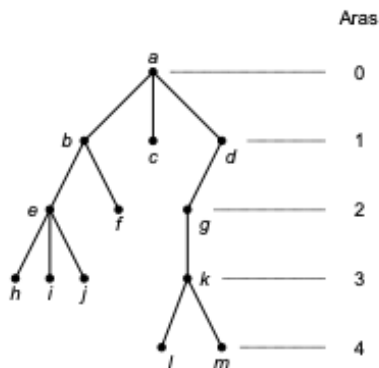
i, dan j merupakan daun dengan orang tua e. Begitu juga dengan simpul c dan g adalah daun karena nilai dari derajat simpul tersebut adalah 0.

vi. Simpul Dalam (*Internal Nodes*)

Simpul dalam merupakan simpul yang memiliki anak. Hal ini berarti bahwa daun tidak termasuk dalam simpul anak karena derajat dari daun adalah 0. Pada gambar 2.3, simpul a,b,d, merupakan simpul dalam dan sisanya bukan simpul dalam karena tidak memiliki anak.

vii. Aras (*level*) atau Tingkat

Tingkat merupakan panjang jalan dari akar sampai simpul tertentu.



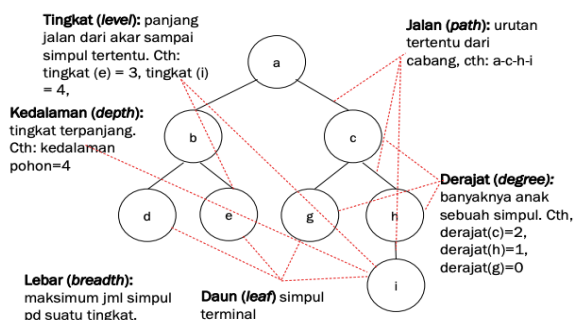
Gambar 2.6. Penggambaran tingkatan pada pohon (sumber : <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm>)

Pada gambar di atas, tingkat dari simpul a adalah 0 karena jika dihitung panjang jalannya, panjang jalan dari a ke a adalah 0. Selain itu, tingkat dari simpul b, c, dan d adalah 1 karena jika dihitung panjang jalannya, panjang jalan dari akar ke simpul tujuan yaitu dari a ke b adalah 1. Dengan begitu, tingkat tertinggi selalu dimiliki oleh simpul daun.

viii. Tinggi (*height*) atau Kedalaman (*depth*)

Tinggi atau kedalaman suatu pohon merupakan nilai dari aras maksimum dalam pohon tersebut. Nilai aras terbesar selalu dimiliki oleh simpul daun. Oleh karena itu, tinggi dari suatu pohon didapat dari aras terbesar dari aras masing-masing simpul daun pada pohon tersebut.

Pada Gambar 2.6, terdapat 7 simpul daun dengan beragam nilai aras masing-masing simpul. Oleh karena simpul m dan l memiliki aras paling tinggi maka aras tersebut akan menjadi tinggi dari pohon.

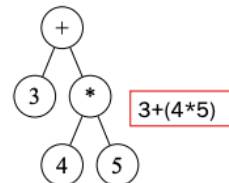


Gambar 2.7. Terminologi pohon berakar. (sumber : <https://stei19.kuliah.itb.ac.id/course/view.php?id=1070>)

5. Pohon Biner

Pohon Biner merupakan pohon dengan jumlah anak maksimum dari masing-masing simpul adalah 2. Oleh karena hanya terdapat dua anak pada masing-masing simpul, penyebutan anaknya dibedakan menjadi anak kiri (*left child*) dan anak kanan (*right child*). Akan tetapi, perbedaan penyebutan ini mengindikasikan keterurutan sehingga pohon biner adalah pohon terurut.

Penggunaan Pohon Biner sangat aplikatif dalam dunia nyata. Salah satu contoh dari penggunaan pohon biner adalah operasi aritmatika.



Gambar 2.8. Pohon Biner (sumber : <https://stei19.kuliah.itb.ac.id/course/view.php?id=1070>)

Penggunaan pohon biner dalam operasi aritmatika menetapkan nilai dari setiap simpul dalam menjadi operator sedangkan nilai dari simpul daun menjadi operan.

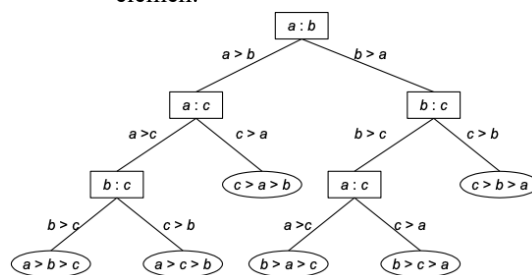
Contoh : Pada gambar 2.8, proses pembacaan yang dilakukan adalah inorder sehingga hasil dari pohon tersebut adalah sebagai berikut.

$$3+(4*5) = 23$$

6. Pohon Keputusan

Pohon Keputusan merupakan pohon dengan yang setiap simpulnya berisi perbandingan-perbandingan yang akan menimbulkan suatu simpulan yang dimana masing-masing simpulan yang didapat terlabel pada setiap sisi pada pohon. Hasil dari perbandingan-perbandingan ini akan menghasilkan suatu keputusan dimana hasil akhir dari keputusannya akan diletakkan pada setiap simpul daun.

Contoh : Pohon keputusan untuk mengurutkan 3 buah elemen.



Gambar 2.9. Pohon Keputusan (sumber : <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm>)

Pada pohon keputusan di atas, untuk mendapatkan 3 elemen yang terurut dilakukan perbandingan antara 3 elemen tersebut. Pada akhirnya, setiap perbandingan antara elemen tersebut akan menghasilkan elemen terurut dimana jumlah keputusan yang memungkinkan sebanyak sebagai berikut.

Pada pohon, akan ada 6 keputusan. Keputusan tersebut diwakili oleh 6 simpul daun pada pohon. Secara sederhana perhitungan jumlah keputusannya adalah sebagai berikut.

Perbandingan elemen tidak dapat dilakukan pada elemen yang sama sehingga perbandingannya adalah sebagai berikut.
 3 elemen > 2 elemen > 1 elemen = P(3,3)
 $P(3,3) = 3! = 3.2.1 = 6$ Keputusan

B. K-Nearest Neighbors Algorithm

1. Teori Keputusan

Pada penentuan keputusan, dalam beberapa kasus akan terdapat kasus dimana setiap keputusan yang diambil berdasar pada hasil pendekatan nilai karena keputusan yang diambil ditentukan berdasarkan nilai terdekat. Hal ini dipraktikkan dalam machine learning dimana Kinerja dari pembelajaran ini memiliki nilai *loss* dari fungsi errornya.

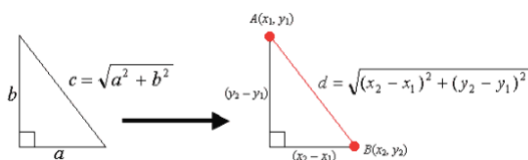
Loss adalah ukuran seberapa dekat atau berbeda model yang dihasilkan dengan konsep asli sedangkan *error* adalah salah satu fungsi untuk mengukur *loss*. Dalam pengukurannya, nilai *loss* ini dapat dihitung dengan fungsi cross entropy loss. Rumus dari fungsi adalah sebagai berikut.

$$H = - \sum_{i=1}^N q(x) \log(p(x))$$

q(x) : distribusi data asli
 p(x) : distribusi data sampel

2. Konsep Algoritma KNN

Algoritma K-Nearest Neighbor (KNN) adalah sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. Secara sederhana, algoritma ini mengelompokkan data ke kelompok yang memiliki sifat termirip dengannya. Algoritma ini termasuk dalam supervised learning, dimana data akan diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada. Dengan begitu, jika digambarkan dalam peraga dua dimensi, kedekatan dari kedua data ini dapat diukur berdasarkan Pythagoras.



Gambar 2.10. Konsep Pythagoras. (sumber : <https://medium.com/bee-solution-partners/cara-kerja-algoritma-k-nearest-neighbor-k-nn-389297de543e>)

Untuk menghitung jarak antara 1 komponen pada kedua data dapat dilakukan dengan rumus pythagoras (jika *independent variable*-nya bernilai 2) sehingga jika terdapat K tetangga (data sample) maka akan dicari K jumlah jarak dari data ke masing-masing data sample.

Dalam dunia nyata, bukan tidak mungkin bahwa nilai dari *independent variable*-nya tidak sama dengan 2. Hal ini menyatakan bahwa rumus pythagoras tidak mencukupi untuk menangani kasus tersebut. Oleh karena itu, kita dapat menggunakan rumus Euclidean Distance untuk dimensi lebih dari 2. Rumusnya adalah sebagai berikut.

$$= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Dalam pengklasifikasian data, data akan diklasifikasikan berdasarkan jarak terdekat ke data sample.

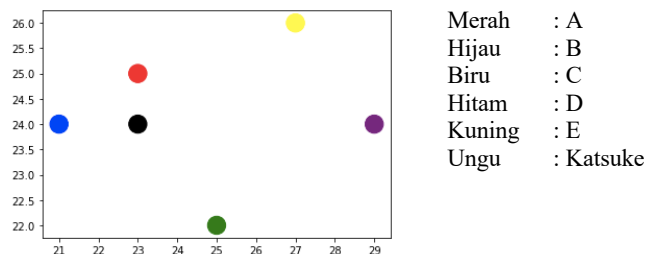
Contoh : Terdapat 5 rumah yaitu A,B, C,D, dan E. Rumah A berada di Lembang, rumah B berada di Cisitu, rumah C berada di Tamansari, rumah D berada di braga, dan rumah E berada di Dago. Katsuke ingin mencari tahu rumahnya termasuk dalam daerah Lembang, Cisitu, Tamansari, Braga, atau Dago dengan diketahui posisi dari rumahnya berada pada latitude 29 dan longitude 24. Diketahui posisi dari masing-masing rumah A,B,C,D, dan E adalah sebagai berikut.

Rumah	Latitute	Longitude
A	23	25
B	25	22
C	21	24
D	23	24
E	27	26

Jarak ke A
 $= \text{sqrt}((29-23)^2 + (24-25)^2) = 6.08$ satuan
 Jarak ke B
 $= \text{sqrt}((29-25)^2 + (24-22)^2) = 4.47$ satuan
 Jarak ke C
 $= \text{sqrt}((29-21)^2 + (24-24)^2) = 8$ satuan
 Jarak ke D
 $= \text{sqrt}((29-23)^2 + (24-24)^2) = 6$ satuan
 Jarak ke E
 $= \text{sqrt}((29-27)^2 + (24-26)^2) = 2.82$ satuan

Dari hasil perhitungan di atas, didapat bahwa rumah E memiliki jarak terdekat ke rumah Katsuke yaitu sebesar 2.82 satuan sehingga rumah E memiliki kemiripan terdekat terhadap rumah Katsuke dibandingkan rumah A,B,C,dan D. Oleh karena rumah E berada pada daerah Dago maka dapat disimpulkan bahwa rumah Katsuke berada pada daerah Dago juga.

Perhitungan diatas telah mewakili cara kerja dari algoritma KNN. Dengan melakukan perbandingan terhadap setiap data sample maka akan menyimpulkan suatu keputusan berdasarkan jarak data terdekat dengan data pengguna tsb.



Gambar 2.11. Scatter plot dari masing-masing rumah. (sumber: penulis)

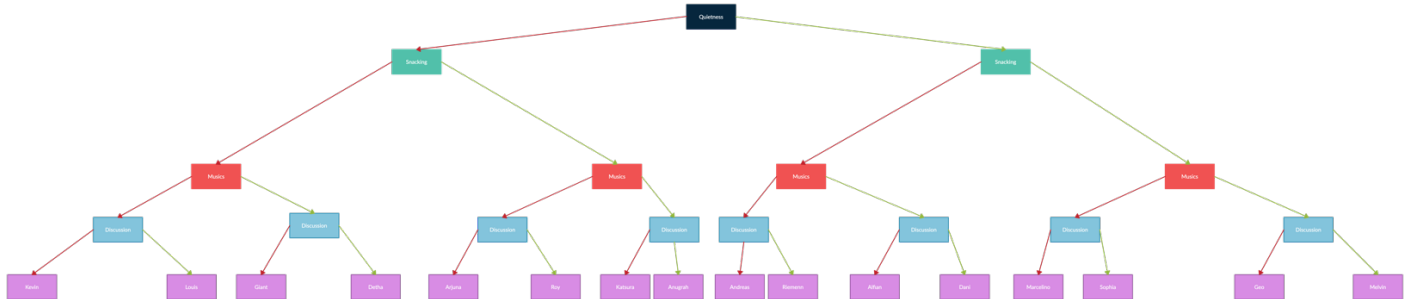
Dari gambar di atas terlihat bahwa titik terdekat dari titik berwarna ungu adalah titik berwarna kuning sehingga dari scatter plot di atas, dapat disimpulkan bahwa rumah Katsuke berada pada daerah yang sama dengan rumah E yaitu Dago.

III. HASIL DAN PEMBAHASAN

Pada penentuan teman belajar terbaik, penggunaan prinsip pohon keputusan akan membantu. Masing-masing simpul pada pada pohon keputusan ini akan menjadi kategori atau *independent variable* yang akan dibandingkan nilainya. Hal ini berarti bahwa jumlah simpul dalam pada pohon keputusan yang akan diterapkan untuk menentukan teman

belajar terbaik ini sama dengan jumlah *independent variable* yang akan digunakan dalam algoritma *K-Nearest Neighbors*.

Dalam penentuan teman belajar terbaik ini, pohon biner akan merepresentasikan pohon yang akan digunakan sehingga akan terdapat anak kanan (*right child*) dan anak kiri (*left child*). Masing-masing arah dari anak pohon ini akan membawa suatu nilai hasil perbandingan. Pohon penentuan teman belajar terbaik adalah sebagai berikut.



Gambar 3.1. Pohon Keputusan dalam penentuan teman belajar terbaik (*zoom* untuk melihat secara jelas) (sumber : penulis).

Pada Gambar 3.1, daun direpresentasikan dengan warna ungu dan warna lain merepresentasikan simpul dalam sehingga dapat disimpulkan terdapat 16 buah simpul daun dan 15 buah simpul dalam. Akan tetapi, hanya terdapat 4 kategori yang akan menjadi *independent variable* yang akan digunakan dalam algoritma KNN. Keempat kategori tersebut direpresentasikan oleh 4 warna berbeda di luar warna ungu (simpul daun) yaitu hitam, hijau, merah, dan biru.

Simpul daun yang direpresentasikan oleh warna ungu pada pohon tersebut akan menjadi hasil keputusan. Nilai dari simpul daun ini berisi nama dari masing-masing siswa dalam data sample yang memiliki cara belajar masing-masingnya sehingga untuk mencapai masing-masing siswa ini, akan ada kategori/*independent variable* yang akan dibandingkan. Adapun data dari siswa adalah sebagai berikut.

Name	Quietness	Snacking	Musics	Discussion
Kevin	Yes	Yes	Yes	Yes
Louis	Yes	Yes	Yes	No
Giant	Yes	Yes	No	Yes
Detha	Yes	Yes	No	No
Arjuna	Yes	No	Yes	Yes
Roy	Yes	No	Yes	No
Katsura	Yes	No	No	Yes
Anugrah	Yes	No	No	No
Andreas	No	Yes	Yes	Yes
Riemenn	No	Yes	Yes	No
Alfian	No	Yes	No	Yes
Dani	No	Yes	No	No
Marcelino	No	No	Yes	Yes
Sophia	No	No	Yes	No
Geo	No	No	No	Yes
Melvin	No	No	No	No

Tabel 3.1. Tabel penentuan teman belajar terbaik.

Berdasarkan tabel 3.1, terdapat 4 *independent variables* yaitu *Quietness*, *Snacking*, *Musics*, dan *Discussion* dan 1 *dependent variable* yaitu *Name*. Untuk mencapai masing masing nama, terdapat 4 *independent variables* yang harus terpenuhi.

Berdasarkan pohon pada gambar 3.1, akan ada sejumlah 4 *independent variable* yang direpresentasikan warna yang berbeda. Warna hitam merepresentasikan *Quietness*, warna hijau merepresentasikan *snacking*, warna merah merepresentasikan *musics*, dan warna biru merepresentasikan *Discussion*. Jika dihitung maka akan ada daun (nama) sejumlah sebagai berikut.

$$2^n = 2^4 = 16 \text{ daun}$$

n : jumlah simpul dalam

Dari hasil perhitungan tersebut maka dapat disimpulkan bahwa peluang untuk mendapatkan masing-masing nama adalah 1/16 atau 6.25%.

1. Penggunaan Algoritma K-Nearest Neighbors untuk mencari teman belajar yang sesuai.

Dalam penerapan program, akan digunakan bahasa python dalam implementasi library KNN. Pada python, untuk mengakses library KNN dapat digunakan perintah import sebagai berikut.

```
import pandas as pd
from sklearn.neighbors import NearestNeighbors
```

Kemudian, akan dibaca file yang memuat data yang memuat syarat-syarat (*independent variable*) untuk masing nama (*dependent variable*) seperti pada tabel 3.1. Pada makalah ini, akan digunakan file bernama data.xlsx sebagai data yang berisi tabel 3.1. File akan dimasukkan ke dalam variable df.

```
df = pd.read_excel("data.xlsx")
```

Untuk melakukan perbandingan, masing-masing nilai string data akan diubah menjadi tipe int. Pada program, nilai dari

masing masing independent variable untuk masing-masing nama akan diubah menjadi representasi bilangan. String ‘Yes’ menjadi angka 1 dan ‘No’ menjadi angka 0. Hasilnya akan seperti gambar di bawah ini.

```
df['Quietness'] = df['Quietness'].apply(lambda x:0 if x=='no' else 1)
df['Snacking'] = df['Snacking'].apply(lambda x:0 if x=='no' else 1)
df['Musics'] = df['Musics'].apply(lambda x:0 if x=='no' else 1)
df['Discussion'] = df['Discussion'].apply(lambda x:0 if x=='no' else 1)
print(df)
```

	Name	Quietness	Snacking	Musics	Discussion
0	Kevin	1	1	1	1
1	Louis	1	1	1	0
2	Giant	1	1	0	1
3	Dettha	1	1	0	0
4	Arjuna	1	0	1	1
5	Roy	1	0	1	0
6	Katsura	1	0	0	1
7	Anugrah	1	0	0	0
8	Andreas	0	1	1	1
9	Riemenn	0	1	1	0
10	Alfian	0	1	0	1
11	Dani	0	1	0	0
12	Marcelino	0	0	1	1
13	Sophia	0	0	1	0
14	Geo	0	0	0	1
15	Melvin	0	0	0	0

Selanjutnya, untuk mengimplementasikan program sesuai dengan keinginan pengguna, digunakan fungsi input() yang nilainya akan dimasukkan ke variable. Pengguna akan memasukkan input string dengan input terbatas pada kata ‘yes’/‘no’.

```
#input pengguna
Q = input("Apakah kamu senang belajar ditempat yang sunyi? (yes/no)")
S = input("Apakah kamu senang belajar sambil ngemil? (yes/no)")
M = input("Apakah kamu senang belajar sambil dengerin musics? (yes/no)")
D = input("Apakah kamu senang belajar dengan metode diskusi? (yes/no)")
```

Pada program di atas, Q merepresentasikan nilai Quietness, S merepresentasikan nilai Snacking, M merepresentasikan nilai Musics, dan D merepresentasikan nilai Discussion. Input yang merepresentasikan masing-masing independent variables ini akan dibentuk dalam sebuah dataframe yang bernama dfinput sehingga tipe nya dapat dibandingkan dengan variabel df yang menyimpan nilai seperti pada tabel 3.1. Kemudian, input pengguna tersebut akan diubah lagi dalam bentuk angka, ‘yes’ menjadi angka 1 dan ‘no’ menjadi angka 0.

```
#Membuat dataframe input pengguna
input_data = {}
input_data['Quietness'] = [Q]
input_data['Snacking'] = [S]
input_data['Musics'] = [M]
input_data['Discussion'] = [D]
dfinput = pd.DataFrame(data=input_data)

#mengubah yes menjadi 1 dan no menjadi 0
dfinput['Quietness'] = dfinput['Quietness'].apply(lambda x:0 if x=='no' else 1)
dfinput['Snacking'] = dfinput['Snacking'].apply(lambda x:0 if x=='no' else 1)
dfinput['Musics'] = dfinput['Musics'].apply(lambda x:0 if x=='no' else 1)
dfinput['Discussion'] = dfinput['Discussion'].apply(lambda x:0 if x=='no' else 1)
```

Setelah itu, sebelum library KNN digunakan, nilai dependent variable pada df harus di drop karena dependent variable dalam penerapan library KNN ini tidak dapat digunakan (dependent variable tidak mengandung nilai yang perbandingan karena variabel in hanya merupakan suatu entity). Akan tetapi, nilai dari dependent variable ini akan masih digunakan untuk menampilkan nama dari siswa sehingga data yang dependent variable nya didrop akan dimasukkan ke dalam variabel baru bernama X.

```
#drop dependent variable
X = df.drop(['Name'],axis=1)
```

Kemudian, fungsi NearestNeighbors dipanggil dengan penyetah jumlah K nya sebesar 1 untuk menampilkan 1 nilai terdekat saja nantinya. Dalam fungsi tersebut juga diatur file frame yang akan dibaca yaitu X. Fungsi ini akan diwakilkan dalam bentuk 1 variabel yang bernama nbrs yang kemudian akan dipakai dan dihubungkan dengan dataframe input pengguna yang telah dibentuk tadi yaitu dfinput.

```
#panggil fungsi NearestNeighbors
nbrs = NearestNeighbors(n_neighbors=1).fit(X)
_, idx = nbrs.kneighbors(dfinput.values)

#Menampilkan nama yang cocok dijadikan sebagai teman belajar
print(df.iloc[idx[0]]['Name'].values[0])
```

Hasilnya akan dimasukkan ke dalam variabel idx yang merupakan posisi/indeks keberapa data terdekat itu berada. Akhirnya, nama dari siswa yang berada pada indeks ke idx akan ditampilkan ke layar.

2. Pengujian Program

Pada saat program dijalankan, input pengguna akan diminta. Jumlah input pengguna yang akan ditampilkan berjumlah 4 sesuai dengan jumlah independent variable yang digunakan. Adapun tampilan input adalah sebagai berikut.

input pertama :

```
yes|
Apakah kamu senang belajar ditempat yang sunyi? (yes/no) (Press 'Enter' to confirm or 'Escape' to cancel)
```

input kedua :

```
no|
Apakah kamu senang belajar sambil ngemil? (yes/no) (Press 'Enter' to confirm or 'Escape' to cancel)
```

input ketiga :

```
yes|
Apakah kamu senang belajar sambil dengerin musics? (yes/no) (Press 'Enter' to confirm or 'Escape' to cancel)
```

input keempat :

```
no|
Apakah kamu senang belajar dengan metode diskusi? (yes/no) (Press 'Enter' to confirm or 'Escape' to cancel)
```

Sebagai contoh, jika input pertama adalah ‘yes’, input kedua adalah ‘no’, input ketiga adalah ‘yes’, dan input keempat adalah ‘no’ maka program akan menampilkan hasil sebagai berikut.

Roy

Hal ini berarti bahwa ketika pengguna belajar dalam keadaan sunyi, tidak suka mengemil ketika belajar, senang mendengarkan musik sambil belajar, dan tidak suka metode diskusi ketika belajar, maka siswa terbaik yang memiliki kesamaan cara belajar dengan pengguna tersebut adalah ‘Roy’.

Jika dilihat data Roy pada tabel 3.1, nilai quietnessnya adalah ‘yes’, nilai snacking adalah ‘no’, nilai musics adalah ‘yes’, dan nilai discussion adalah ‘no’. Ini berarti bahwa nilai masukan pengguna sesuai dengan nilai ‘Roy’.

Dalam pohon keputusan ini, nilai loss dari program ini adalah 0% karena dalam pohon keputusan, setiap perbandingan menghasilkan nilai yang jelas ada pada data (setiap keputusan

pasti berakhir pada satu daun yang mempunyai nilai nama dari suatu siswa). Oleh karena kesesuaian hasil dan keberjalanan program yang baik, pembentukan suatu program dengan algoritma KNN untuk mencari teman belajar terbaik adalah berhasil.

IV. KESIMPULAN

Pengaplikasian pohon keputusan dapat diterapkan dalam pencarian teman belajar terbaik. Untuk mencari teman belajar terbaik terdapat komponen komponen cara belajar yang harus disesuaikan. Dengan kinerja dari algoritma K-Nearest Neighbors, sebuah program pencarian teman belajar yang sesuai dengan cara belajar kita dapat dilakukan. Dengan sistem pencarian jarak terdekat pada algoritma KNN, algoritma ini dapat dimanfaatkan dengan nilai *loss* yang akan selalu bernilai 0% untuk suatu pohon keputusan. Hal ini disebabkan karena nilai suatu perbandingan pada pohon keputusan selalu di-cover sehingga akan selalu merujuk pada suatu keputusan yang direpresentasikan oleh daun pada pohon.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur yang sebesar-besarnya kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga makalah berjudul "Penerapan Pohon Keputusan dengan Algoritma KNN Dalam Mencari Teman Belajar Terbaik" dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada Dr. Nur Ulfa Maulidevi ST,M.Sc. selaku dosen IF2120 Matematika Diskrit yang telah memberikan bimbingan dan membagikan ilmunya terkait materi matematika, khususnya Pohon Keputusan yang merupakan materi dasar yang digunakan dalam makalah ini. Penulis juga mengucapkan terima kasih kepada teman-teman yang telah mendukung saya selama proses pembuatan makalah ini. Tak lupa juga penulis mengucapkan terima kasih kepada para penyedia referensi yang memberikan informasi yang sangat penulis butuhkan dalam menyelesaikan makalah ini. Akhir kata, penulis mengucapkan terima kasih kepada semua orang yang telah terlibat dalam pembuatan makalah ini.

REFERENCES

- [1] Wira Gotama Putra, Jan. 2020. *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*. Tokyo.
- [2] Munir, Rinaldi. 2020. <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm>. Diakses tanggal 8 Desember 2020.
- [3] Informatikalogi.com. <https://informatikalogi.com/algoritma-k-nn-k-nearest-neighbor>. Diakses tanggal 8 Desember 2020.
- [4] Hai Matematika. <https://www.haimatematika.com/2018/12/graf-pohon-teori-graf.html>. Diakses tanggal 8 Desember 2020.
- [5] ITB. https://stei19.kuliah.itb.ac.id/pluginfile.php/76455/mod_resource/content/1/W14_A1_PohonBiner_Bagian2.pdf. Diakses tanggal 8 Desember 2020.
- [6] Maulana Ismail, Asep. 2018. <https://medium.com/bee-solution-partners/cara-kerja-algoritma-k-nearest-neighbor-k-nn-389297de543e>. Diakses tanggal 8 Desember 2020.
- [7] Open Notebooks. 2019. <https://moonbooks.org/Articles/How-to-create-a-scatter-plot-with-several-colors-in-matplotlib/>. Diakses tanggal 8 Desember 2020

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2020



Kevin Katsura Dani Sitanggang
13519216