

LAPORAN TUGAS PEMROGRAMAN 3
PENGANTAR KECERDASAN BUATAN



Disusun oleh :

1. Muhammad Rifq Arrahim (1301190425)
2. Ikhlasul Arifin (1301194187)
3. Bayu Muhammad Iqbal (1301184241)

Program Studi S1 Informatika

Fakultas Informatika

2021

ANALISIS MASALAH

Diberikan 17 data mobil dengan 6 atribut yaitu nama mobil, ukuran, kenyamanan, irit, kecepatan, dan harga. Dalam tugas ini kami akan memberikan 3 rekomendasi mobil berdasarkan 4 formula penghitungan jarak yang mendekati input file test.xls yang telah kami tentukan. File test.xls ini berisi baris header dan 1 baris data.

Training data (mobil.xls)

Nama Mobil	Ukuran	Kenyamanan	Irit	Kecepatan	Harga (Ratus Juta)
Toyota Agya	4	4	9	6	1
Daihatsu Alya	4	3	9	6	1.1
Toyota Avanza	6	5	6	6	2
Daihatsu Xenia	6	4	6	6	1.75
Xpander	7	7	6	7	2.25
Livina	7	7	6	7	2.1
Karimun	3	4	10	5	1.2
Toyota Innova	8	8	5	7	4
Alphard	9	10	4	8	10
Toyota Vios	5	7	9	8	2.5
Honda City	5	8	7	8	2.7
Toyota Hiace	10	5	8	6	5
Toyota Fortuner	9	8	5	8	5
Toyota Foxy	9	9	5	7	5.5
Toyota Corolla Altis	5	9	7	9	6
Suzuki Ertiga	7	7	7	7	2.3
Suzuki Carry	7	3	9	5	0.8

test file (test.xls)

nama mobil	ukuran	kenyamanan	irit	kecepatan	harga
Tayo	9	4	4	7	1

STRATEGI PENYELESAIAN MASALAH

Dalam menyelesaikan permasalahan ini, strategi yang kami gunakan adalah kNN dengan menerapkan 4 metode perhitungan jarak yang berbeda. Berikut ini adalah perhitungan jarak dan implementasi pada fungsi program :

1. Euclidean Distance

Jarak Atribut Numerik: *Euclidean Distance*

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2}$$

FUNGSI MENGHITUNG EUCLIDEAN

```
[ ] def euclidean(x, y):  
    #x=list 1 mobil  
    #y=list data test  
    n1 = x[1]-y[0][1]  
    n2 = x[2]-y[0][2]  
    n3 = x[3]-y[0][3]  
    n4 = x[4]-y[0][4]  
    n5 = x[5]-y[0][5]  
    return ((n1)**2 + (n2)**2 + (n3)**2 + (n4)**2 + (n5)**2)**(1/2)
```

2. Manhattan Distance

Jarak Atribut Numerik: *Manhattan Distance*

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

FUNGSI MENGHITUNG MANHATTAN

```
[▶] def manhattan(x, y):  
    #x=list 1 mobil  
    #y=list data test  
    n1 = x[1]-y[0][1]  
    n2 = x[2]-y[0][2]  
    n3 = x[3]-y[0][3]  
    n4 = x[4]-y[0][4]  
    n5 = x[5]-y[0][5]  
    return (abs(n1) + abs(n2) + abs(n3) + abs(n4) + abs(n5))
```

3. Minkowski Distance

Jarak Atribut Numerik: *Minkowski Distance*

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \dots + |x_{ip} - x_{jp}|^h}$$

FUNGSI MENGHITUNG MINKOWSKI

```
[▶] def minkowski(x, y):  
    #x=list 1 mobil  
    #y=list data test  
    #h=1.5=3/2  
    n1 = (abs(x[1]-y[0][1]))**(3/2)  
    n2 = (abs(x[2]-y[0][2]))**(3/2)  
    n3 = (abs(x[3]-y[0][3]))**(3/2)  
    n4 = (abs(x[4]-y[0][4]))**(3/2)  
    n5 = (abs(x[5]-y[0][5]))**(3/2)  
    return (n1 + n2 + n3 + n4 + n5)**(2/3)
```

4. Supremum Distance

Jarak Atribut Numerik: *Supremum Distance*

$$d(i, j) = \lim_{h \rightarrow \infty} \left(\sum_{f=1}^p |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_f^p |x_{if} - x_{jf}|$$

FUNGSI MENGHITUNG SUPREMUM

```
[▶] def supremum(x, y):  
    #x=list 1 mobil  
    #y=list data test  
    n1 = (abs(x[1]-y[0][1]))  
    n2 = (abs(x[2]-y[0][2]))  
    n3 = (abs(x[3]-y[0][3]))  
    n4 = (abs(x[4]-y[0][4]))  
    n5 = (abs(x[5]-y[0][5]))  
    return max(n1, n2, n3, n4, n5)
```

HASIL DAN KESIMPULAN

Program

IMPORT LIB

```
[ ] import pandas as pd
import numpy as np
import xlwt
```

DONWLOAD FILE TEST

```
[ ] !gdown --id 15zW6ehWDbjWCpPA4Cmi99nFeY1-wjbG1

Downloading...
From: https://drive.google.com/uc?id=15zW6ehWDbjWCpPA4Cmi99nFeY1-wjbG1
To: /content/test.xls
100% 25.6k/25.6k [00:00<00:00, 21.6MB/s]
```

DONWLOAD TRAINING DATA

```
[ ] !gdown --id 1rFd_1NiSsxLCAzvdFz-91h9NJEfBRg02


Downloading...
From: https://drive.google.com/uc?id=1rFd\_1NiSsxLCAzvdFz-91h9NJEfBRg02
To: /content/mobil.xls
100% 29.2k/29.2k [00:00<00:00, 47.0MB/s]
```

Import library yang akan digunakan, lalu download data training dan data test.

MEMBACA FILE TEST

```
[ ] tes = pd.read_excel('test.xls')
```

MEMBACA FILE TRAINING DATA

```
 data = pd.read_excel('mobil.xls')
```

Gunakan pandas untuk membaca data.


```

#define beberapa array
jarak = []
euc_list = []
man_list = []
min_list = []
supre_list = []
#ubah dataframe menjadi array
jarak = data.to_numpy()
test = tes.to_numpy()
#lakukan proses penghitungan setiap data
for i in range(17):
    euc_list.append(euclidean(jarak[i], test))
    man_list.append(manhattan(jarak[i], test))
    min_list.append(minkowski(jarak[i], test))
    supre_list.append(supremum(jarak[i], test))

```

Define beberapa array untuk menampung data. jarak untuk menampung data training, test untuk menampung data test, euc_list untuk menampung hasil perhitungan jarak dengan formula euclidean, man_list untuk menampung hasil perhitungan jarak dengan formula manhattan, min_list untuk menampung hasil perhitungan jarak dengan formula minkowski, dan supre_list untuk menampung hasil perhitungan jarak dengan formula supremum. lakukan looping sebanyak jumlah data training (17), di setiap loopingnya hitung jarak antara data test dan data training menggunakan 4 formula tersebut, lalu masukan kedalam array.

```

#masukan hasil perhitungan kedalam dataframe
data.insert(6,"Euclidean Distance",euc_list,True)
data.insert(7,"Manhattan Distance",man_list,True)
data.insert(8,"Minkowski Distance",min_list,True)
data.insert(9,"Supremum Distance",supre_list,True)

```

Masukan array hasil perhitungan jarak ke dalam data frame.

```

data.sort_values(by=['Manhattan Distance'], inplace=True) # jika ingin me-run program gunakan run all, jika tidak maka kolom dataframe akan bertambah terus
print("-----Berikut mobil yang mendekati input berdasarkan Manhattan Distance-----")
print(data)
result = data.to_numpy()
#masukan hasil ke dalam excel
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('Hasil')
for i in range(3):
    worksheet.write(i,0,result[i][0])
workbook.save('Rekomendasi_Manhattan.xls')

```

Urutkan data frame berdasarkan kolom Manhattan Distance secara Ascending, print seluruh data frame, tulis 3 index teratas ke dalam excel “Rekomendasi_Manhattan.xls”

```
data.sort_values(by=['Euclidean Distance'], inplace=True) # jika ingin me-run program gunakan run all, jika tidak maka kolom dataframe akan bertambah terus
print("-----Berikut mobil yang mendekati input berdasarkan Euclidean Distance-----")
print(data)
result = data.to_numpy()
#masukan hasil ke dalam excel
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('Hasil')
for i in range(3):
    worksheet.write(i,0,result[i][0])
workbook.save('Rekomendasi_Euclidean.xls')
```

Urutkan data frame berdasarkan kolom Euclidean Distance secara Ascending, print seluruh data frame, tulis 3 index teratas ke dalam excel “Rekomendasi_Euclidean.xls”.

```
data.sort_values(by=['Minkowski Distance'], inplace=True) # jika ingin me-run program gunakan run all, jika tidak maka kolom dataframe akan bertambah terus
print("-----Berikut mobil yang mendekati input berdasarkan Minkowski Distance-----")
print(data)
result = data.to_numpy()
#masukan hasil ke dalam excel
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('Hasil')
for i in range(3):
    worksheet.write(i,0,result[i][0])
workbook.save('Rekomendasi_Minkowski.xls')
```

Urutkan data frame berdasarkan kolom Minkowski Distance secara Ascending, print seluruh data frame, tulis 3 index teratas ke dalam excel “Rekomendasi_Minkowski.xls”.

```
data.sort_values(by=['Supremum Distance'], inplace=True) # jika ingin me-run program gunakan run all, jika tidak maka kolom dataframe akan bertambah terus
print("-----Berikut mobil yang mendekati input berdasarkan Supremum Distance-----")
print(data)
result = data.to_numpy()
#masukan hasil ke dalam excel
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('Hasil')
for i in range(3):
    worksheet.write(i,0,result[i][0])
workbook.save('Rekomendasi_Supremum.xls')
```

Urutkan data frame berdasarkan kolom Supremum Distance secara Ascending, print seluruh data frame, tulis 3 index teratas ke dalam excel “Rekomendasi_Supremum.xls”.

Hasil

Setelah program dijalankan, program akan memberikan keluaran berupa 4 file xls yang masing masing merupakan hasil rekomendasi 3 mobil yang mendekati input file test.xls dengan formula perhitungan jarak yang berbeda.

1. Euclidean Distance

	A	B
1	Daihatsu Xenia	
2	Toyota Avanza	
3	Livina	
4		
5		

2. Manhattan Distance

	A	B
1	Daihatsu Xenia	
2	Toyota Avanza	
3	Livina	
4		
5		

3. Minkowski Distance

	A	B
1	Daihatsu Xenia	
2	Toyota Avanza	
3	Livina	
4		
5		

4. Supremum Distance

	A	B
1	Daihatsu Xenia	
2	Toyota Avanza	
3	Livina	
4		

Kesimpulan

Dari ke-4 formula, semuanya memberikan output yang sama yaitu Daihatsu Xenia, Toyota Avanza, Livina terhadap input test file yang kami berikan. Namun output ke-4 formula bisa berbeda jika banyaknya mobil yang ingin di rekomendasikan > 3 atau input test file yang di berikan berbeda dengan sebelumnya.

Contoh dengan menggunakan input test file yang sama, namun memberikan 6 rekomendasi mobil setiap formulanya.

Berikut mobil yang mendekati input berdasarkan Manhattan Distance										
Nama Mobil	Ukuran	Kenyamanan	Irit	Kecepatan	Harga (Ratus Juta)	Euclidean Distance	Manhattan Distance	Minkowski Distance	Supremum Distance	
Daihatsu Xenia	6	4	6	6	1.75	3.816084	6.75	4.540186	3.0	
Toyota Avanza	6	5	6	6	2.00	4.000000	8.00	4.953453	3.0	
Livina	7	7	6	7	2.10	4.267318	8.10	5.243433	3.0	
Xpander	7	7	6	7	2.25	4.308422	8.25	5.314190	3.0	
Toyota Innova	8	8	5	7	4.00	5.196152	9.00	6.135111	4.0	
Suzuki Ertiga	7	7	7	7	2.30	4.867237	9.30	6.001639	3.0	
Berikut mobil yang mendekati input berdasarkan Euclidean Distance										
Nama Mobil	Ukuran	Kenyamanan	Irit	Kecepatan	Harga (Ratus Juta)	Euclidean Distance	Manhattan Distance	Minkowski Distance	Supremum Distance	
Daihatsu Xenia	6	4	6	6	1.75	3.816084	6.75	4.540186	3.0	
Toyota Avanza	6	5	6	6	2.00	4.000000	8.00	4.953453	3.0	
Livina	7	7	6	7	2.10	4.267318	8.10	5.243433	3.0	
Xpander	7	7	6	7	2.25	4.308422	8.25	5.314190	3.0	
Suzuki Ertiga	7	7	7	7	2.30	4.867237	9.30	6.001639	3.0	
Toyota Innova	8	8	5	7	4.00	5.196152	9.00	6.135111	4.0	
Berikut mobil yang mendekati input berdasarkan Minkowski Distance										
Nama Mobil	Ukuran	Kenyamanan	Irit	Kecepatan	Harga (Ratus Juta)	Euclidean Distance	Manhattan Distance	Minkowski Distance	Supremum Distance	
Daihatsu Xenia	6	4	6	6	1.75	3.816084	6.75	4.540186	3.0	
Toyota Avanza	6	5	6	6	2.00	4.000000	8.00	4.953453	3.0	
Livina	7	7	6	7	2.10	4.267318	8.10	5.243433	3.0	
Xpander	7	7	6	7	2.25	4.308422	8.25	5.314190	3.0	
Suzuki Ertiga	7	7	7	7	2.30	4.867237	9.30	6.001639	3.0	
Toyota Innova	8	8	5	7	4.00	5.196152	9.00	6.135111	4.0	
Berikut mobil yang mendekati input berdasarkan Supremum Distance										
Nama Mobil	Ukuran	Kenyamanan	Irit	Kecepatan	Harga (Ratus Juta)	Euclidean Distance	Manhattan Distance	Minkowski Distance	Supremum Distance	
Daihatsu Xenia	6	4	6	6	1.75	3.816084	6.75	4.540186	3.0	
Toyota Avanza	6	5	6	6	2.00	4.000000	8.00	4.953453	3.0	
Livina	7	7	6	7	2.10	4.267318	8.10	5.243433	3.0	
Xpander	7	7	6	7	2.25	4.308422	8.25	5.314190	3.0	
Suzuki Ertiga	7	7	7	7	2.30	4.867237	9.30	6.001639	3.0	
Honda City	5	8	7	8	2.70	6.700000	13.70	8.415443	4.0	

terjadi perbedaan antara mobil rekomendasi ke-5 dan ke-6.

link source code :

https://colab.research.google.com/drive/1XXxQg5b25FqsFp_A1DAcXB7HErxOwSSw?usp=sharing

link video presentasi :

<https://drive.google.com/file/d/1gnaD7Ssz1Ccl5iBxBoW73lqQvy5LNyBJ/view>