

TUGAS PEMROGRAMAN 3

Laporan

Disusun untuk memenuhi tugas pada mata kuliah Pengantar Kecerdasan Buatan

Oleh:

Otniel Abiezer (1301180469)

Muhammad Haidir Ali (1301180205)



KELOMPOK 18

FAKULTAS INFORMATIKA

S1 INFORMATIKA

IF-43-05

PEMBAHASAN

Klasifikasi merupakan permasalahan yang fundamental di banyak bidang penelitian khususnya pada bidang computer vision, salah satunya adalah untuk mengklasifikasikan jenis kendaraan berbasis pengolahan citra digital pada sistem transportasi cerdas. Tujuan dari klasifikasi adalah untuk mengklasifikasi sebuah objek yang tidak diketahui ke dalam kelas yang dimodelkan dengan sebuah algoritme pembelajaran mesin berdasarkan data latih.

KNN merupakan salah satu algoritme yang banyak digunakan untuk melakukan klasifikasi. KNN sangat sederhana dan mudah untuk diimplementasikan dengan memiliki tingkat akurasi yang baik, dengan mengklasifikasi terhadap suatu objek yang berdasarkan k buah data.

Dengan pembahasan seperti ini dalam proses pembangunan Source Code, kami membentuk prosedur atau fungsi yang berbasis system KNN untuk memilih 3 mobil terbaik sesuai dengan inputan user, yang dibaca melalui file *mobil.xls* dan dapat mengeluarkan output berupa sebuah file *rekomendasi.xls*

1. Dalam perhitungan jarak rumus yang digunakan yaitu Euclidean Distance yaitu sebuah teori yang perhitungan jarak dari 2 buah titik dalam Euclidean Space, yang dalam proses nya membandingkan kedekatan nilai dari dua buah variable antara uji dan acuan dengan mencari nilai terdekat.

```
#Rumus menghitung Euclidean Distance
```

```
def EuclideanDistance(train, test):  
    jumlah = 0  
    for i in range (0, len(test)):  
        hasil = (train[i+1] - test[i]) ** 2  
        jumlah += hasil  
    return math.sqrt(jumlah)
```

```
#Tabel Klasifikasi menampung hasil Euclidean Distance
```

```
def tabelEuclidean(data_mobil, mobil):  
    tbl_euclidean = []  
    for i in range (0, len(data_mobil)):  
        tampung = []  
        nama_mobil = data_mobil[i][0]  
        nilai = EuclideanDistance(data_mobil[i], mobil)  
        tampung.append(nama_mobil)  
        tampung.append(nilai)  
        tbl_euclidean.append(tampung)  
    return tbl_euclidean
```

2. Pada Teknik pemrosesan data kami menggunakan Validasi Range Input yaitu membatasi input data yang bisa dimasukan, sehingga data yang di inputkan dapat memenuhi kriteria tertentu yang ditetapkan agar dapat meminimalisir kesalahan entry data

```
#Pre Processing data dengan validasi inputan agar sesuai range
```

```
def validasiInput1sd10(inputan):
```

```
    if (0 <= inputan <= 10):
```

```
        print("Inputan berhasil")
```

```
        return True
```

```
    else:
```

```
        print("Tolong masukkan Angka yang valid dari 1-10")
```

```
        return False
```

```
def validasiInputHarga(harga):
```

```
    if (harga >= 0):
```

```
        print("Inputan berhasil")
```

```
        return True
```

```
    else:
```

```
        print("Tolong masukkan Angka positif yang valid >0 ")
```

```
        return False
```

3. Strategi algoritma yang digunakan pada KNN yaitu dengan parameter K=3

```
#Memilih 3 nearest neighbour
```

```
def choose_3_nearest_neighbour(arr):
```

```
    three_neighbour = []
```

```
    for i in range(0,3):
```

```
        three_neighbour.append(arr[i][0])
```

```
    return three_neighbour
```



4. Progres-progres yang dibentuk dan dibangun pada Source Code yaitu seperti

- Pengimporan data dari file *mobil.xls*

```
#Melakukan import data dari mobil.xls
def importData():
    data = pd.read_excel("mobil.xls")
    nama_mobil = data['Nama Mobil'].to_list()
    ukuran = data['Ukuran'].to_list()
    kenyamanan = data['Kenyamanan'].to_list()
    irit = data['Irit'].to_list()
    kecepatan = data['Kecepatan'].to_list()
    harga = data['Harga (Ratus Juta)'].to_list()
    data_mobil = []
    for i in range (0,len(nama_mobil)):
        tampung = []
        tampung.append(nama_mobil[i])
        tampung.append(ukuran[i])
        tampung.append(kenyamanan[i])
        tampung.append(irit[i])
        tampung.append(kecepatan[i])
        tampung.append(harga[i])
        data_mobil.append(tampung)
    return data_mobil
```

- Fungsi Proses data dengan Validasi Range Input yang ada pada penjelasan sebelumnya.
- Inputan dari user yang menerima dari 5 kriteria seperti *Ukuran, Kenyamanan, Irit, Kecepatan*, dan juga *Harga*.

```
#Menampung inputan dari User dari 5 kriteria
def inputDariUser():
    mobil = []
    print("Masukkan kriteria angka yang ingin Anda cari rekomendasi mobilnya : ")
    while True:
        ukuran = float(input("Masukkan angka untuk ukuran (dari 1 sampai 10): "))
        if validasiInput1sd10(ukuran):
            mobil.append(ukuran)
            break
    while True:
        kenyamanan = float(input("Masukkan angka untuk kenyamanan (dari 1 sampai 10): "))
        if validasiInput1sd10(kenyamanan):
            mobil.append(kenyamanan)
            break
    while True:
        irit = float(input("Masukkan angka untuk irit (dari 1 sampai 10): "))
        if validasiInput1sd10(irit):
            mobil.append(irit)
            break
    while True:
        kecepatan = float(input("Masukkan angka untuk kecepatan (dari 1 sampai 10): "))
        if validasiInput1sd10(kecepatan):
            mobil.append(kecepatan)
            break
    while True:
        harga = float(input("Masukkan angka untuk harga (lebih besar dari 0): "))
        if validasiInputHarga(harga):
            mobil.append(harga)
            break
    return mobil
```

- Penghitungan rumus dengan Euclidean Distance yang sudah dijelaskan sebelumnya.



- Algoritma untuk mensorting data

```
#Algoritma Sorting
def SelectionSort(arr):
    for i in range(0, len(arr)):
        minimal = i
        for j in range(i+1, len(arr)):
            if (arr[minimal][1] > arr[j][1]):
                minimal = j
        arr[minimal], arr[i] = arr[i], arr[minimal]
```

- Hasil data yang menghasilkan file *rekomendasi.xlsx*

```
#Export Data ke Excel
def exportData(tbl_euclidean):
    df = pd.DataFrame(tbl_euclidean, columns = ["Nama Mobil"])
    df.to_excel(r'rekomendasi.xlsx', index=False, header=True)
```

- Main Program

```
=====MAIN PROGRAM=====

#Melakukan import data mobil.xls
data_mobil = importData()

#Menampung inputan dari User
mobil = inputDariUser()

#Tabel Klasifikasi menampung hasil menghitung jarak Euclidean
tbl_euclidean = tabelEuclidean(data_mobil, mobil)

#Mengurutkan tabel
SelectionSort(tbl_euclidean)

#Memilih 3 neighbour terdekat
three_neighbour = choose_3_nearest_neighbour(tbl_euclidean)

#Melakukan export data ke excel
exportData(three_neighbour)
```



KESIMPULAN

Dari percobaan tersebut, dapat disimpulkan bahwa algoritma K-Nearest-Neighbour (KNN) cukup efektif dan mudah diimplementasikan untuk kasus dengan data latih berupa mobil dengan 5 karakteristik dan juga data latih berupa *inputan* user dengan menggunakan jarak Euclidean dan parameter $k=3$, karena menghasilkan rekomendasi berupa 3 dan memiliki nama kelas yang berbeda semua.

Output:

```
Masukkan kriteria angka yang ingin Anda cari rekomandasi mobilnya :  
Masukkan angka untuk ukuran (dari 1 sampai 10): 4  
Inputan berhasil  
Masukkan angka untuk kenyamanan (dari 1 sampai 10): 3  
Inputan berhasil  
Masukkan angka untuk irit (dari 1 sampai 10): 2  
Inputan berhasil  
Masukkan angka untuk kecepatan (dari 1 sampai 10): 5  
Inputan berhasil  
Masukkan angka untuk harga (lebih besar dari 0): 1.2  
Inputan berhasil
```

Dan juga file “rekomendasi.xlsx”

	A	B
1	Nama Mobil	
2	Daihatsu Xenia	
3	Toyota Avanza	
4	Livina	

Link Video Presentasi :

<https://drive.google.com/file/d/11VoYh9KsN9zO7SHcSZZiX4hCEwYlOZtf/view?usp=sharing>