

**Pengelompokan Daerah Rawan Kecelakaan di Bandar Lampung dengan
Metode K-Means Clustering**



Disusun oleh:

Indra Jaya Putra	120140059
Abi Luthfi Ramdan Fadhillah	120140151
Dean Andhika Ramadhan	120140155

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI, PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN**

2023

DAFTAR ISI

DAFTAR ISI	2
ABSTRAK	3
BAB I	
PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	4
1.3 Tujuan	4
BAB II	
LANDASAN TEORI	5
2.1 Metode K-Means	5
2.2 Clustering	6
2.3 Kecelakaan	6
2.4 Daerah Rawan Kecelakaan	6
2.5 Python	6
BAB III	
METODOLOGI DAN PEMBAHASAN	7
3.1 Alur Penelitian	7
3.2 K-Means	7
BAB IV	
HASIL DAN ANALISIS	10
4.1 Hasil	10
4.1.1 Struktur Data dan Implementasi	10
BAB V	
KESIMPULAN	18
DAFTAR PUSTAKA	19
BIODATA PENULIS	20

ABSTRAK

Bandar Lampung merupakan daerah yang memiliki tingkat kerawanan kecelakaan lalu lintas yang cukup tinggi, hal ini dikarenakan kota Bandar Lampung terdapat banyak jalan dengan kondisi yang rusak dan berlubang sehingga hal ini dapat menimbulkan berbagai permasalahan lalu lintas seperti terjadinya kemacetan serta kecelakaan. Penelitian ini menyajikan informasi berupa pengelompokan daerah-daerah rawan kecelakaan yang ada di Bandar Lampung. Untuk menganalisis daerah rawan kecelakaan menggunakan *K-Means clustering* dan kemudian dilakukan pengujian inertia untuk mengetahui kualitas clustering menggunakan algoritma *K-Means*. Hasil penelitian ini adalah informasi kelompok atau cluster daerah yang dikategorikan rawan kecelakaan dan sangat rawan kecelakaan.

Kata kunci : K-means clustering, daerah, kecelakaan.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kecelakaan lalu lintas merupakan kejadian tak terduga dan tidak disengaja yang melibatkan kendaraan, baik dengan atau tanpa melibatkan pengguna jalan lainnya, dan dapat mengakibatkan korban jiwa atau kerugian harta benda. Di Indonesia, kecelakaan lalu lintas menjadi penyebab utama kematian, yang dapat berdampak signifikan pada aspek ekonomi dan sosial. Peningkatan penggunaan kendaraan ini dapat memicu terjadinya kecelakaan lalu lintas yang dapat mengakibatkan cedera atau bahkan kematian. Kejadian ini tidak mudah untuk dikurangi karena akan terus bertambah seiring dengan pertambahan panjang jalan serta pergerakan dari suatu kendaraan [1]. Faktor-faktor yang dapat menjadi penyebab terjadinya kecelakaan meliputi kesalahan pengguna jalan, kondisi kendaraan, maupun kondisi jalan dan alam [2]. Oleh karena itu, diperlukan penelitian yang lebih rinci untuk mengidentifikasi daerah rawan kecelakaan yang ada di Bandar Lampung.

Bandar Lampung merupakan daerah yang memiliki tingkat kerawanan kecelakaan lalu lintas yang cukup tinggi, hal ini dikarenakan kota Bandar Lampung terdapat banyak sekali bangunan-bangunan besar, perkantoran, serta banyak infrastruktur dengan kondisi jalan yang rusak dan berlubang menyebabkan arus lalu lintas menjadi padat dan hal ini dapat menimbulkan berbagai permasalahan lalu lintas seperti terjadinya kemacetan serta kecelakaan. Kasatlantas Polresta Bandar Lampung, AKP M Rohmawan memaparkan beberapa daerah-daerah yang rawan akan terjadinya kecelakaan di Bandar Lampung contohnya pada Jl. Imba Kusuma (Lembah Hijau), Jl. Sisingamangaraja, Jl. Emir M Noer, Jl. Soekarno Hatta (PT Andatu Panjang), Jl. Ir Sutami, Jl. Soekarno Hatta (PT Hanjung).

Penelitian ini menyajikan informasi berupa pengelompokan daerah-daerah rawan kecelakaan yang ada di Bandar Lampung. Untuk menganalisis daerah rawan kecelakaan menggunakan *K-Means clustering*. Hasil dari penelitian ini adalah grafik sebaran data dari daerah-daerah rawan kecelakaan yang ada di Bandar Lampung.

1.2 Rumusan Masalah

Mengacu pada latar belakang, maka rumusan masalah dari penelitian ini adalah:

1. Bagaimana mengimplementasikan metode *K-Means clustering* untuk pengelompokan daerah rawan kecelakaan di Bandar Lampung?
2. Bagaimana hasil pengelompokan daerah rawan kecelakaan di Bandar Lampung menggunakan metode K-Means Clustering?
3. Bagaimana hasil pengujian kualitas clustering menggunakan metode inertia?

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah:

1. Mengetahui pengelompokan daerah rawan kecelakaan di Bandar Lampung dengan metode *K-Means clustering*
2. Mengetahui hasil pengelompokan daerah rawan kecelakaan di Bandar Lampung menggunakan metode *K-Means Clustering*
3. Mengetahui hasil pengujian kualitas clustering menggunakan metode inertia

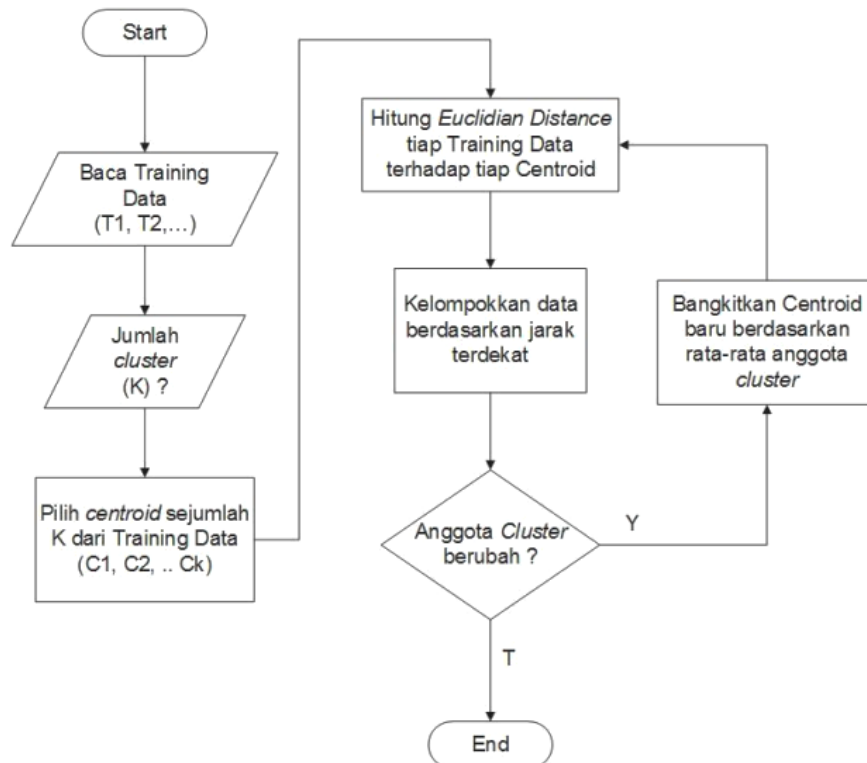
BAB II

LANDASAN TEORI

2.1 Metode K-Means

Metode K-Means merupakan suatu metode pengelompokan data atau metode data mining yang melakukan proses pemodelan dengan sifat partitional dan pembelajaran yang unsupervised. Metode ini bekerja dengan memasukkan K sebagai konstanta jumlah cluster yang diinginkan. Sedangkan, Means memiliki arti nilai rata-rata dari suatu kelompok data yang didefinisikan sebagai cluster. Metode K-Means merupakan metode non-hirarki yang berusaha mempartisi data yang memiliki karakteristik sama ke dalam bentuk satu atau lebih kelompok atau cluster yang sama [3].

Metode K-Means akan secara acak memilih pola k sebagai titik awal centroid. Kandidat centroid cluster awal yang ditentukan secara acak akan mempengaruhi jumlah iterasi untuk mencapai centroid cluster jika posisi centroid baru tidak berubah. Nilai K yang terpilih menjadi centroid awal akan dilakukan perhitungan dengan menggunakan rumus perhitungan jarak untuk mencari jarak terpendek antara titik centroid dengan data seperti Cosine Similarity, Manhattan, Euclidean Distance, atau lainnya. Data yang memiliki jarak terdekat dengan centroid akan membentuk kelompok [4]. Secara terstruktur langkah-langkah metode K-Means dijabarkan pada gambar 1.



Gambar 1. Flowchart metode K-Means

2.2 Clustering

Clustering adalah suatu proses pengelompokan data ke dalam beberapa kelompok atau cluster. Tingkat kemiripan data dalam satu cluster yang sama adalah lebih tinggi daripada antar cluster. Clustering merupakan metode segmentasi data dan saat ini sudah diimplementasikan kedalam beberapa bidang seperti prediksi dan analisa masalah bisnis segmentasi pasar, zona wilayah, hingga pengolahan media [5].

Clustering saat ini telah digunakan untuk mengelompokkan data dengan tujuan tertentu tergantung dengan kebutuhan. Prinsip dalam clustering adalah membagi data yang memiliki karakteristik sama ke dalam suatu kelompok. Data dimasukkan ke dalam suatu kelompok/cluster dengan memiliki kemiripan yang tinggi kemudian menjauhkan dengan data dengan kemiripan rendah [6].

2.3 Kecelakaan

Kecelakaan adalah sebuah insiden yang tak dapat diprediksi dan bisa terjadi di mana saja, termasuk kecelakaan lalu lintas di jalan raya yang merupakan penyebab kematian terbesar di dunia. Setiap tahunnya, menurut World Health Organization (WHO), terdapat 1,35 juta orang yang meninggal dunia karena kecelakaan lalu lintas, dengan pengendara motor dan mobil menjadi korban utamanya. Menurut POLRI, kecelakaan lalu lintas pada tahun 2020 mengakibatkan 80 orang meninggal dunia setiap harinya atau 3 orang setiap jam di jalan raya. Jumlah ini mengalami kenaikan sebanyak 260 peristiwa atau sekitar 26,7% dibandingkan dengan tahun sebelumnya pada minggu ke-33 dan minggu ke-32. Korlantas Polri melaporkan bahwa pada minggu ke-32 tahun 2020 terjadi 1.234 kecelakaan lalu lintas. Hal ini menunjukkan bahwa jalan raya di Indonesia masih rawan terjadinya kecelakaan [7].

2.4 Daerah Rawan Kecelakaan

Daerah rawan kecelakaan lalu lintas merujuk pada area yang memiliki tingkat kecelakaan lalu lintas yang tinggi, dengan kejadian kecelakaan yang terjadi secara berulang dalam kurun waktu yang relatif sama. Penyebab kecelakaan tersebut dapat bervariasi tergantung pada setiap daerahnya. Daerah rawan kecelakaan lalu lintas dibagi menjadi 3 kategori yaitu daerah tampak rawan kecelakaan, rute rawan kecelakaan, dan wilayah rawan kecelakaan [8].

2.5 Python

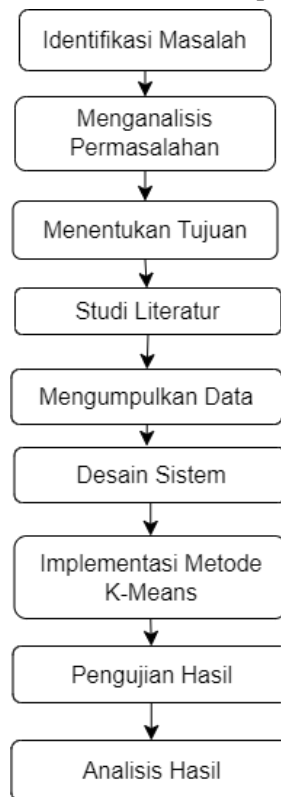
Python merupakan bahasa pemrograman tingkat tinggi (high-level programming language) yang beroperasi dengan sistem interpreted, dan dapat digunakan untuk berbagai tujuan (general-purpose). Python menerapkan metode pemrosesan interpreted, di mana kode program akan diproses baris per baris secara langsung dari kode program (tanpa memerlukan proses kompilasi). Cara ini serupa dengan bahasa skrip seperti PHP dan JavaScript [9].

Di Python, terdapat suatu konsep yang disebut sebagai "library". Library Python mengelola berbagai fungsi yang ada, sehingga ketika kita ingin menggunakan suatu fungsi, kita harus memanggil library yang mengelolanya terlebih dahulu. Python sebagai bahasa pemrograman dapat dimanfaatkan untuk melakukan prediksi berdasarkan data historis, mengotomatisasi tugas, menyederhanakan proses, dan mengolah data untuk mendapatkan wawasan dalam bidang Business Intelligence. Python menyediakan banyak paket, modul, dan library dalam bidang Data Science [10].

BAB III METODOLOGI DAN PEMBAHASAN

3.1 Alur Penelitian

Dalam melakukan penelitian, peneliti menggunakan tahapan atau alur untuk mencapai tujuan yang telah direncanakan. Dalam menunjang itu peneliti memerlukan tahapan alur penelitian seperti berikut :



Gambar 2. Alur Penelitian

3.2 K-Means

A. Algoritma K-Means Clustering

K-Means adalah salah satu metode pengelompokan data atau clustering yang bekerja dengan mempartisi data ke dalam satu atau beberapa kelompok atau cluster. Pengelompokan dilakukan berdasarkan karakteristik data yang serupa sehingga data dengan karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama. [6].

Clustering adalah teknik pengelompokan data berdasarkan tingkat kemiripannya. Data yang memiliki tingkat kemiripan yang tinggi akan dikelompokkan ke dalam sebuah cluster. Setiap data akan dimasukkan ke dalam cluster yang memiliki karakteristik yang paling mirip, dan berusaha untuk menjauhkannya dari cluster yang lain. Clustering sering digunakan untuk mengelompokkan data dengan tujuan dan manfaat yang berbeda-beda, tergantung pada kebutuhan yang ada.

Keuntungan menggunakan algoritma K-Means Clustering adalah:

1. Algoritma ini mudah dipahami dan di implementasikan

2. Dapat mengolah data dengan jumlah yang besar secara cepat
3. Efektif digunakan untuk pengelompokan data dengan jumlah kelompok yang kecil dan jelas
4. Algoritma ini dapat digunakan untuk berbagai jenis data seperti data kategorikal dan numerik

Namun, terdapat kelemahan dari algoritma K-Means Clustering:

1. K-means hanya menentukan hasil terbaik dengan bergantung pada nilai centroid yang acak, hasil clustering akan bervariasi jika dilakukan beberapa kali pada data yang sama.
2. Sensitif terhadap nilai yang ekstrim dalam data
3. Algoritma ini tidak efektif untuk pengelompokan data dengan data yang banyak dan tidak jelas

K-Means Clustering adalah salah satu algoritma clustering yang populer digunakan karena mudah diimplementasikan dan efektif untuk mengelompokkan data dengan jumlah kelompok yang kecil dan jelas. Namun, perlu diingat bahwa hasil clustering dapat bervariasi tergantung pada posisi centroid awal yang diacak dan sensitif terhadap nilai-nilai ekstrim dalam data. Oleh karena itu, pemilihan jumlah kelompok dan evaluasi hasil clustering sangat penting dalam menggunakan algoritma K-Means Clustering.

B. Dataset

Pada implementasi dan fungsionalitas algoritma K-Means Clustering, data yang akan dipakai dan diolah didapatkan dari berita yang ada di internet dan sudah dipastikan kebenarannya dalam 4 tahun terakhir. Data yang dipakai dalam perhitungan atau implementasi K-Means Clustering berupa seberapa banyak kecelakaan terjadi di kecamatan tersebut, jumlah korban, dan jenis kecelakaannya, data tersebut akan digunakan sebagai parameter penentuan apakah daerah tersebut rawan terjadi kecelakaan menggunakan algoritma ini.

C. Ilustrasi Perhitungan Metode

K-Means merupakan salah satu metode dalam analisis data, dimana data akan dikategorikan berkelompok sesuai karakteristik yang sama lalu akan dikelompokkan menjadi satu cluster.

Proses dari metode K-Means Clustering adalah sebagai berikut :

1. Menentukan banyak cluster yang dibutuhkan dalam penelitian.
2. Menentukan pusat cluster atau centroid awal, dalam penentuannya biasanya dipilih secara acak
3. Menggunakan rumus jarak euclidean untuk menghitung jarak terdekat antara setiap objek pengamatan dan centroid awal yang telah ditentukan sebelumnya.

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^n (x_{il} - x_{jl})^2}$$

$d(x_i, x_j)$ = Jarak antara objek ke i dengan objek ke j

X_{il} = Nilai objek ke – I pada variabel k

X_{jl} = Nilai objek ke – j pada variabel k

n = Banyaknya variabel

4. Menentukan jarak terdekat, dengan objek dengan centroid
5. Menentukan centroid baru dengan menghitung rata-rata masing-masing cluster menggunakan rumus.

$$c_{kl} = \frac{x_{1l} + x_{2l} + \dots + x_{pl}}{p}$$

Keterangan :

C_{kl} : Nilai centroid ke-k pada variabel ke-l

p : Banyaknya data

6. Langkah-langkah untuk melakukan clustering adalah memisahkan objek menjadi beberapa cluster awal, menghitung pusat cluster untuk setiap cluster, menghitung jarak setiap objek dengan pusat cluster dan menempatkannya ke dalam cluster dengan pusat terdekat, menghitung kembali pusat cluster baru dengan mengambil rata-rata objek di setiap cluster, dan mengulangi langkah 3-4 sampai tidak ada objek yang berpindah cluster atau pusat cluster tidak berubah.

D. Algoritma K-Means Clustering

Dalam Penerapan Metode K-Means Clustering dalam Pengelompokan Daerah Rawan Kecelakaan di Bandar Lampung terdapat algoritma perhitungan menggunakan program python yang menggunakan Library *pandas*, *numpy*, *sklearn*.

BAB IV HASIL DAN ANALISIS

4.1 Hasil

4.1.1 Struktur Data dan Implementasi

Dalam penelitian ini penulis menggunakan bahasa pemrograman Python untuk implementasi metode K-Means Clustering pada daerah rawan Kecelakaan di Bandar Lampung dan menggunakan library pandas, numpy, dan sklearn atau singkatan dari scikit-learn. Penggunaan library pada program python yang dibuat dapat dilihat pada tabel 1.

Library yang digunakan dalam program ini adalah sebagai berikut:

- ❖ *Pandas*

Library ini digunakan untuk memanipulasi data dalam bentuk tabel, membersihkan data

- ❖ *Numpy*

Library ini digunakan untuk perhitungan numerik

- ❖ *Scikit-learn*

Scikit-learn atau sklearn adalah library bahasa python yang banyak digunakan dalam analisis machine learning

- ❖ *Sklearn.cluster*

Modul sklearn.cluster merupakan bagian dari library Scikit-learn yang menyediakan berbagai algoritma clustering, seperti k-Means, DBSCAN, Spectral Clustering, dan Hierarchical Clustering. Modul ini digunakan untuk melakukan clustering pada data sebagai teknik unsupervised learning dalam machine learning.

Tabel 1. Memanggil Library

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans

print("    Analisis Pengelompokan Daerah Rawan Kecelakaan di Bandar
Lampung    ")
```

- ❖ *Membaca data dari file external*

Selanjutnya data akan dibaca menggunakan fungsi `np.loadtxt('data.txt', delimiter=',')`. Fungsi ini digunakan untuk membaca file external dengan bentuk format txt dengan delimiter koma, data ini kemudian dimasukkan ke dalam variabel data.

Tabel 2. Data Awal

kecamatan ▼	berapa kali kejadian ▼	jumlah korban ▼	mobil/motor ▼
Enggal	1	5	mobil
Panjang	6	18 (7 tewas)	mobil
pahoman	1	1 (tewas)	motor
sukabumi	4	18 (4 tewas)	mobil & motor
sukarame	3	9	mobil & motor
rajabasa	2	4 (2 tewas)	mobil & motor
way halim	1	16	mobil
kedaton	2	4	mobil

Tabel 2 merupakan data awal yang berisi nama tempat, kejadian, korban, dan jenis kendaraan terhadap kejadian kecelakaan. Data tersebut kemudian dibuat dalam bentuk file txt dengan hanya menampilkan kolom jumlah kejadian (kolom 2), kolom jumlah korban (kolom 3), dan kolom jenis kendaraan (kolom 4).

Tabel 3. File txt

```
1,5,1
6,18,1
1,1,1
4,18,2
3,9,2
2,4,2
1,16,1
2,4,1
```

Pada tabel 3 menampilkan data dari kolom 2, kolom 3, dan kolom 4 sesuai urutannya. Kolom-kolom yang akan digunakan sebagai fitur dalam clustering dipilih dari data menggunakan slicing. Pada tabel 4 dibuat inisialisasi variabel X yang merupakan kolom-kolom dari data yang sudah dibuat pada file txt. Kolom ke-0 adalah jumlah kecelakaan, kolom ke-1 adalah jumlah korban, dan kolom ke-2 adalah jenis kendaraan. Inisialisasi variabel dengan kolom dapat dilihat pada tabel 4..

Tabel 4. Membaca Data

```
# Membaca data dari file eksternal
data = np.loadtxt('data.txt', delimiter=',')

# Mengambil kolom jumlah korban jiwa, jumlah kecelakaan, dan jumlah
kendaraan
X = data[:, [0, 1, 2]]
```

- ❖ Data yang akan digunakan untuk clustering kemudian ditampilkan ke layar (dapat dilihat pada tabel 5).

Tabel 5. Menampilkan Data

```
print("\n\nDATA")
print("\n[Banyak kecelakaan, Jumlah Korban, Jenis Kendaraan]")
print(X)
```

Tabel 6. Output data yang sudah di tampilkan

Analisis Pengelompokan Daerah Rawan Kecelakaan di Bandar Lampung

DATA

```
[Banyak kecelakaan, Jumlah Korban, Jenis Kendaraan]
[[ 1.  5.  1.]
 [ 6. 18.  1.]
 [ 1.  1.  1.]
 [ 4. 18.  2.]
 [ 3.  9.  2.]
 [ 2.  4.  2.]
 [ 1. 16.  1.]
 [ 2.  4.  1.]]
```

- ❖ Kemudian, nilai K dan centroid awal secara acak diinisialisasi. Dalam contoh ini, K diinisialisasi dengan nilai 2 dan centroid awal diambil dari 2 baris acak dalam data X. Nilai K menunjukkan jumlah cluster yang diinginkan. Centroid awal kemudian ditampilkan ke layar.

Tabel 7. Inisialisasi nilai K dan centroid secara acak

```
# Inisialisasi nilai K dan centroid secara acak
k = 2
centroids = X[np.random.choice(X.shape[0], size=k, replace=False)]

# Print centroid
print("\ncentroids:")
print(centroids)
print("")
```

Tabel 8. Nilai centroid secara acak

```
centroids:  
[[ 4. 18.  2.]  
 [ 2.  4.  1.]]
```

- ❖ Selanjutnya, jarak Euclidean antara setiap data dan masing-masing centroid dihitung. Jarak ini disimpan dalam matriks distances.

Tabel 9. Menghitung jarak Euclidean distance

```
# hitung jarak Euclidean distance untuk setiap data terhadap  
masing-masing centroid  
distances = np.zeros((data.shape[0], centroids.shape[0]))  
for i in range(data.shape[0]):  
    for j in range(centroids.shape[0]):  
        distances[i,j] = np.sqrt(np.sum((data[i,:]-centroids[j,:])**2))
```

- ❖ Setelah itu, untuk setiap data, centroid terdekat dicari dan diambil indeksinya. Hasil ini disimpan dalam variabel closest_centroid.

Tabel 10. Mengambil index centroid terdekat

```
# cari centroid terdekat untuk setiap data  
closest_centroid = np.argmin(distances, axis=1)
```

- ❖ Setelah itu jarak data terhadap centroid akan di print, dan centroid terdekat untuk setiap data juga akan ditampilkan

Tabel 11. Menampilkan hasil dari perhitungan sebelumnya

```
# tampilkan hasil  
print("\nJarak data terhadap centroid:")  
print(distances)  
print("\nCentroid terdekat untuk setiap data:")  
print(closest_centroid)  
print("")  
print("=====")
```

Tabel 12. Output nilai

Jarak data terhadap centroid:

```
[[13.37908816  1.41421356]
 [ 2.23606798 14.56021978]
 [17.29161647  3.16227766]
 [ 0.          14.17744688]
 [ 9.05538514  5.19615242]
 [14.14213562  1.         ]
 [ 3.74165739 12.04159458]
 [14.17744688  0.         ]]
```

Centroid terdekat untuk setiap data:

```
[1 0 1 0 1 1 0 1]
```

- ❖ Jarak antara setiap centroid dan titik origin (0, 0, 0) dihitung, kemudian hasilnya diurutkan dari yang terkecil ke terbesar. Indeks dari centroid-centroid yang terurut kemudian disimpan dalam variabel `sorted_centroids_idx`.

Tabel 13. Hitung jarak dan pengurutan Centroid

```
# Hitung jarak antara setiap centroid ke titik origin (0, 0, 0)
centroid_distances = np.sqrt(np.sum(centroids ** 2, axis=1))

# Urutkan centroid berdasarkan jarak dari yang terkecil ke terbesar
sorted_centroids_idx = np.argsort(centroid_distances)
```

- ❖ Terakhir, data yang telah dikelompokkan ke dalam cluster ditampilkan ke layar. Data ini dikelompokkan berdasarkan urutan centroid terurut. Untuk setiap cluster, data yang termasuk dalam cluster tersebut ditampilkan.

Tabel 14. Menampilkan hasil akhir

```
# Tampilkan hasil pengelompokkan data ke dalam cluster berdasarkan
urutan centroid terurut
print('Pengelompokkan data ke dalam cluster:')
for i in range(k):
    cluster_idx = sorted_centroids_idx[i]
    print('Cluster', i+1, ':')
    print(X[closest_centroid == cluster_idx])
    print()
```

Tabel 15. Hasil akhir

Pengelompokan data ke dalam cluster:

Cluster 1 :

```
[[1. 5. 1.]
 [1. 1. 1.]
 [3. 9. 2.]
 [2. 4. 2.]
 [2. 4. 1.]]
```

Cluster 2 :

```
[[ 6. 18.  1.]
 [ 4. 18.  2.]
 [ 1. 16.  1.]]
```

- ❖ Berdasarkan hasil clustering menggunakan metode K-Means Clustering dapat disimpulkan bahwa C1 adalah cluster dengan daerah yang rawan kecelakaan, sedangkan C2 adalah Cluster dengan daerah yang sangat rawan kecelakaan.

Tabel 16. Hasil Cluster

kecamatan ▼	berapa kali kejadian▼	jumlah korban▼	mobil/motor ▼	Hasil Cluster▼
Enggal	1	5	mobil	C1
Panjang	6	18 (7 tewas)	mobil	C2
pahoman	1	1 (tewas)	motor	C1
sukabumi	4	18 (4 tewas)	mobil & motor	C2
sukarame	3	9	mobil & motor	C1
rajabasa	2	4 (2 tewas)	mobil & motor	C1
way halim	1	16	mobil	C2
kedaton	2	4	mobil	C1

4.2 Pengujian

Pada pengujian perhitungan clustering menggunakan algoritma K-Means Clustering digunakan kode program inertia untuk menentukan meminimalkan nilai inertia agar tiap titik data dapat dikelompokkan ke dalam klaster yang sesuai. Nilai inertia merupakan jumlah jarak kuadrat antara setiap titik data dan centroid dari klaster tempat titik data tersebut berada. Misalnya, Jika kita menggunakan algoritma K-Means untuk mengelompokkan data menjadi 3 kelompok. Setelah perhitungan, perbedaan nilai antar kelompok atau cluster yang dihasilkan dapat diukur. Nilai inertia tersebut mengindikasikan seberapa baik hasil clustering yang diperoleh dari model tersebut. Semakin rendah nilai inertia, semakin baik hasil clustering yang diperoleh. Karena itu, Nilai inersia dihitung dari agregat akhir yang dihasilkan oleh metode K-Means. Dengan menggunakan perintah `print("Inertia:", kmeans.inertia_)`, kita dapat menampilkan nilai inertia pada layar monitor dan mengetahui seberapa baik hasil clustering yang diperoleh oleh model K-Means.

Tabel 17. Model pengujian inertia

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans

# Membaca data dari file eksternal
data = np.loadtxt('data.txt', delimiter=',')

# Mengambil kolom jumlah korban jiwa, jumlah kecelakaan, dan jumlah
kendaraan sebagai fitur
X = data[:, [0, 1, 2]]

print("====DATA====")
print("")
print("[Banyak kecelakaan, Jumlah Korban, Jenis Kendaraan]")
print(X)
print("")

# Inisialisasi nilai K
k_values = [2, 3, 4, 5, 6]

# Menggunakan loop untuk menguji setiap nilai K
for k in k_values:
    # Inisialisasi centroid awal secara acak
    centroids = X[np.random.choice(X.shape[0], size=k, replace=False)]

    # Membuat model KMeans dengan jumlah klaster k
    kmeans = KMeans(n_clusters=k, init=centroids, n_init=1)

    # Melatih model dengan fit() pada data X
    kmeans.fit(X)

    # Hitung inertia untuk model KMeans yang telah dilatih
    inertia = kmeans.inertia_

    # Print hasil untuk setiap nilai K
    print("Nilai inertia untuk k =", k, "adalah:", inertia)
```


Tabel 18. Hasil metode inertia

```
=====DATA=====
[Banyak kecelakaan, Jumlah Korban, Jenis Kendaraan]
[[ 1.  5.  1.]
 [ 6. 18.  1.]
 [ 1.  1.  1.]
 [ 4. 18.  2.]
 [ 3.  9.  2.]
 [ 2.  4.  2.]
 [ 1. 16.  1.]
 [ 2.  4.  1.]]

Nilai inertia untuk k = 2 adalah: 53.2
Nilai inertia untuk k = 3 adalah: 36.0
Nilai inertia untuk k = 4 adalah: 18.0
Nilai inertia untuk k = 5 adalah: 20.0
Nilai inertia untuk k = 6 adalah: 3.5
```

Tabel 18 merupakan output dari pengujian inertia terhadap hasil clustering. Nilai inertia tersebut mengindikasikan seberapa baik hasil clustering yang diperoleh dari model tersebut. Semakin rendah nilai inertia, semakin baik hasil clustering yang diperoleh. Karena penelitian ini hanya menggunakan dua klaster, klaster daerah rawan kecelakaan (C1) dan klaster daerah sangat rawan kecelakaan (C2) sehingga nilai inertia yang didapatkan adalah 53.2. Nilai 53.2 didapatkan dari data yang ada pada penelitian ini. Hal ini membuktikan bahwa hasil clustering dengan dua klaster bernilai cukup namun tidak begitu baik karena dua klaster pada penelitian ini tidak memberikan klasifikasi dan kriteria yang lebih banyak sehingga tidak cukup akurat.

BAB V

KESIMPULAN

Kesimpulan yang dapat disimpulkan dari program ini adalah melakukan clustering data menggunakan algoritma perhitungan K- Means Clustering. K-means clustering merupakan salah satu teknik analisis data yang berfungsi untuk mengklasifikasikan data ke dalam beberapa kelompok yang saling dekat berdasarkan kesamaan centroid. Algoritma K-means clustering melakukan inisialisasi centroid secara acak, menghitung jarak dari setiap data terhadap setiap centroid, menemukan centroid terdekat untuk setiap data, dan mengelompokkan data sesuai urutan centroid.

Program ini menerima data file eksternal dalam bentuk array 2D, dari data kecelakaan yang diambil dari berita online 3 tahun terakhir. Parameter yang digunakan dalam perhitungan K-Means Clustering Pada penelitian ini banyaknya terjadi kecelakaan di wilayah tersebut dalam 3 tahun terakhir, jumlah korban jiwa secara kumulatif, dan jenis kendaraan yang digunakan pada kecelakaan tersebut.

DAFTAR PUSTAKA

- [1] C. Surbakti, B. Sudarsono, and Y. Wahyuddin, "Implementasi Metode Cluster Analysis Dalam Analisis Daerah Rawan kecelakaan Di Kecamatan Semarang Utara," *J. Geod. Undip Juli*, vol. 10, no. 2, pp. 1–10, 2021.
- [2] H. F. Efendi, R. Septima, and H. Syahputra, "SISTEM PENENTUAN TITIK RAWAN KECELAKAAN LALU LINTAS DI BENER," vol. 5, no. 1, pp. 61–69, 2023.
- [3] N. Puspitasari and Haviluddin, "Penerapan Metode K-Means Dalam Pengelompokkan Curah Hujan," *Semin. Nas. Ris. Ilmu Komput. (SNRIK)*, vol. 1, no. March 2017, pp. 2–7, 2016.
- [4] A. Agrawal and H. Gupta, "Global K-Means (GKM) Clustering Algorithm: A Survey," *Int. J. Comput. Appl.*, vol. 79, no. 2, pp. 20–24, 2013, doi: 10.5120/13713-1472.
- [5] E. Irwansyah and M. Faisal, *Advanced Clustering Teori dan Aplikasinya*, vol. 1. 2015.
- [6] E. Harpendi Bara, Y. Agus Pranoto, and F. . Ariwibisono, "Pengelompokan Data Obat Menggunakan Metode K-Means Clustering Pada Upt Puskesmas Kondoran Kec. Sangalla'," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 4, no. 2, pp. 92–97, 2020, doi: 10.36040/jati.v4i2.2720.
- [7] T. K. Titus and M. Jajuli, "Clustering Data Kecelakaan Lalu Lintas di Kecamatan Cileungsi Menggunakan Metode K-Means," *Gener. J.*, vol. 6, no. 1, pp. 1–12, 2022, doi: 10.29407/gj.v6i1.16103.
- [8] A. A. Vernanda, A. Faisol, and N. Vendyansyah, "Penerapan Metode K-Means Clustering Untuk Pemetaan Daerah Rawan Kecelakaan Lalu Lintas Di Kota Malang Berbasis Website," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 5, no. 2, pp. 836–844, 2021, doi: 10.36040/jati.v5i2.3791.
- [9] Andre, "Tutorial Belajar Python Part 1: Pengertian Bahasa Pemrograman Python | Duniaikom," 3 August 2018, 2018. .
- [10] DQLab, "Library Python Kenali Perbedaan Module, Package, dan Library Pada Python," *dqlab.id*, 2021. .

BIODATA PENULIS

Nama : Indra Jaya Putra
NIM : 120140059
Email : indra.120140059@student.itera.ac.id

Nama : Abi Luthfi Ramdan Fadhilah
NIM : 120140151
Email : abi.120140151@student.itera.ac.id

Nama : Dean Andhika Ramadhan
NIM : 120140155
Email : dean.120140155@student.itera.ac.id

