LINIER
LITERATUR INFORMATIKA & KOMPUTER
e-ISSN: 3063-2218
DOI : 10.33096/linier.v1i4.2539
Literatur Informatika & Komputer
Vol 1, No 4 (2024), pp 395-403

Rancang Bangun Penerapan Metode Elbow Pada K-Means Untuk Clustering Data Persediaan Barang

Andri Rajsya[a], Purnawansyah[b], Abdul Rachman Manga[c]
Universitas Muslim Indonesia, Makassar, Indonesia
[a]13020190381@umi.ac.id; [b]purnawansyah@umi.ac.id; [c]abdulrachman.manga@umi.ac.id

Received: 14-08-2024 | Revised: 01-10-2024 | Accepted: 28-11-2024 | Published: 29-12-2024

Abstrak

Penerapan metode *elbow* pada algoritma *K-Means* untuk *clustering* data persediaan barang bertujuan untuk mengoptimalkan penentuan jumlah *cluster* yang optimal, yang sangat penting dalam manajemen persediaan barang. Penelitian ini melibatkan analisis data persediaan barang menggunakan algoritma *K-Means* dan metode *elbow* untuk menentukan jumlah *cluster* yang tepat. Data yang digunakan dalam penelitian ini mencakup data persediaan barang dari sebuah toko retail. Metode penelitian meliputi pengumpulan data, penerapan algoritma *K-Means*, dan evaluasi hasil *clustering* dengan metode *elbow*. Hasil penelitian menunjukkan bahwa penerapan metode *elbow* dapat memberikan gambaran yang jelas tentang jumlah *cluster* optimal yang dapat digunakan untuk meningkatkan efisiensi manajemen persediaan barang. Kesimpulan penelitian ini menunjukkan bahwa metode *elbow* merupakan alat yang efektif dalam menentukan jumlah *cluster* optimal dalam *clustering* data persediaan barang.Penelitian ini juga menunjukkan bahwa dengan menentukan jumlah *cluster* yang optimal, toko retail dapat mengelompokkan produk secara lebih tepat, sehingga strategi pengelolaan stok dapat disesuaikan dengan karakteristik setiap *cluster*. Hal ini berkontribusi pada pengurangan biaya penyimpanan, peningkatan kecepatan rotasi barang, dan optimalisasi ruang penyimpanan.

Kata kunci: K-Means, metode elbow, clustering, data persediaan barang, manajemen persediaan.

Pendahuluan

Clustering merupakan teknik penting dalam analisis data yang digunakan untuk mengelompokkan objek-objek berdasarkan kesamaan karakteristik[1]. Dalam konteks manajemen persediaan barang, *clustering* dapat membantu dalam pengelompokan produk-produk berdasarkan pola penjualan atau karakteristik lainnya[2]. Algoritma *K-Means* adalah salah satu metode *clustering* yang paling populer karena kesederhanaan dan efisiensinya[3]. Namun, salah satu tantangan utama dalam K-Means adalah menentukan jumlah cluster yang optimal.

Metode *elbow* adalah salah satu pendekatan yang sering digunakan untuk menentukan jumlah *cluster* yang optimal dalam *K-Means*[4]. Metode ini bekerja dengan menghitung jumlah total dalam-*cluster variance* untuk berbagai nilai K dan memilih nilai K di mana penurunan dalam-*cluster variance* mulai melambat, membentuk bentuk siku (*elbow*) pada grafik[5].

Penelitian ini bertujuan untuk menerapkan metode *elbow* pada algoritma *K-Means* dalam *clustering* data persediaan barang. Dengan menentukan jumlah *cluster* yang optimal, diharapkan dapat meningkatkan efisiensi dan efektivitas manajemen persediaan barang.

Metode

Metode yang digunakan dalam penelitian ini meliputi beberapa langkah utama. Pertama, data persediaan barang dikumpulkan dari sebuah toko retail. Data ini mencakup informasi tentang jumlah barang, kategori barang, dan data penjualan historis. Setelah data terkumpul, *preprocessing* dilakukan untuk membersihkan dan memformat data agar siap untuk dianalisis.

Algoritma *K-Mea*ns kemudian diterapkan pada data untuk melakukan *clustering*[6]. *K-Means* bekerja dengan mengelompokkan data ke dalam *K cluster* berdasarkan jarak *Euclidean* antara titik data dan pusat *cluster*[7]. Nilai K yang digunakan dalam penelitian ini bervariasi dari 1 hingga 10 untuk menentukan jumlah *cluster* yang optimal[8].

Metode *elbow* digunakan untuk mengevaluasi hasil *clustering* dengan berbagai nilai K[9]. Grafik jumlah total dalam-*cluster variance* (*inertia*) terhadap jumlah *cluster* (K) dibuat, dan titik *elbow* pada grafik ini digunakan untuk menentukan jumlah *cluster* yang optimal[10].

Perancangan

Perancangan sistem melibatkan beberapa tahap utama. Analisis sistem dimulai dengan identifikasi permasalahan dalam manajemen persediaan barang[11]. Permasalahan utama yang diidentifikasi adalah ketidakmampuan untuk mengelompokkan barang-barang secara efisien berdasarkan pola penjualan dan karakteristik lainnya[12].

Rancangan sistem mencakup perancangan input/output, struktur sistem, dan flowchart sistem. Data persediaan barang yang dikumpulkan akan digunakan sebagai input untuk algoritma *K-Means*. Output dari sistem adalah pengelompokan barang-barang ke dalam *cluster-cluster* yang berbeda. Flowchart sistem menggambarkan alur proses mulai dari pengumpulan data hingga penerapan algoritma *K-Means* dan evaluasi hasil *clustering* menggunakan metode *elbow*[13], [14], [15].

1. Permasalahan utama yang dihadapi dalam manajemen persediaan barang adalah:
   a. Tidak adanya pengelompokan barang yang efisien berdasarkan pola penjualan dan karakteristik lainnya.
   b. Kesulitan dalam menentukan jumlah *cluster* yang optimal untuk pengelompokan barang.
   c. Ketidakmampuan untuk melakukan analisis data persediaan secara efektif yang dapat mendukung keputusan bisnis.
2. Perancangan Input/Output
   a. Input : Data persediaan barang
   b. Output: Pengelompokan barang ke dalam *cluster-cluster* yang berbeda berdasarkan pola penjualan dan karakteristik lainnya, Grafik *elbow* untuk menentukan jumlah *cluster* optimal.
3. Struktur Sistem
   a. Pengumpulan Data: Proses pengumpulan data persediaan barang dari *database* atau sumber data lainnya.
   b. *Preprocessing Data*: Pembersihan dan normalisasi data untuk memastikan kualitas dan konsistensi data sebelum analisis.
   c. *Clustering* dengan *K-Means*: Mengelompokkan data persediaan barang menggunakan algoritma *K-Means*.
   d. Evaluasi Hasil dengan Metode *Elbow*: Menggunakan metode *elbow* untuk menentukan jumlah *cluster* optimal.
   e. Visualisasi dan Analisis: Menyajikan hasil *clustering* dan grafik *elbow* untuk analisis lebih lanjut.

Berikut ini adalah Tabel 1 data persediaan barang yang digunakan sebagai input dalam sistem

Tabel 1. Data persediaan barang

No.	ID Barang	Nama Barang	Kategori Barang	Jumlah Barang	Data Penjualan Historis
1	101	Laptop	Elektonik	50	1000
2	102	Kaos	Pakaian	200	1500
3	103	Snack	Makanan	300	2000
4	104	Headphone	Elektronik	75	1200
5	105	Sepatu	Pakaian	180	1300
6	106	Sandal	Pakaian	80	1400
7	107	Sabun	Benda	100	1500
8	108	Sepeda	Benda	40	1600
9	109	Kursi	Benda	20	1700
10	110	Meja	Benda	20	1800

Algoritma *K-Means* bekerja dengan meminimalkan *within-cluster sum of squares* (WCSS). Rumus untuk menghitung jarak *Euclidean* antara dua titik data xxx dan yyy adalah sebagai berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Dimana:

d(x, y): Menyatakan jarak antara titik x dan y.

∑: Simbol *sigma*, yang menandakan penjumlahan.

xi, yi: Koordinat ke-i dari titik x dan y masing-masing.

n: Jumlah dimensi ruang.

Rumus ini menghitung jarak dengan cara mengkuadratkan selisih setiap koordinat, menjumlahkan semua kuadrat tersebut, lalu mengakarkan hasilnya.

Contoh Perhitungan

Mari kita ambil contoh sederhana di ruang 2 dimensi (bidang datar). Misalkan kita memiliki dua titik:
Titik A (x): Koordinatnya adalah (2, 3)
Titik B (y): Koordinatnya adalah (5, 1)
Untuk menghitung jarak antara titik A dan B, kita akan gunakan rumus di atas:
Hitung selisih setiap koordinat dan kuadratkan:
$(x1 - y1)^2 = (2 - 5)^2 = (-3)^2 = 9$
$(x2 - y2)^2 = (3 - 1)^2 = 2^2 = 4$
Jumlahkan hasil kuadrat:
9 + 4 = 13
Akarkan hasil penjumlahan:
√13
Jadi, jarak antara titik A dan B adalah √13 satuan.

Total WCSS dihitung dengan menjumlahkan jarak kuadrat dari setiap titik ke *centroid cluster* terdekatnya:

$$WCSS = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - \mu_k\|^2 \quad (2)$$

Dimana:

K adalah jumlah *cluster*

C*k* adalah *cluster* ke-k

μ*k* adalah *centroid cluster* ke-k

Cara Kerja:

a. Hitung jarak: Untuk setiap titik data dalam suatu *cluster*, hitung jaraknya ke *centroid* (pusat) *cluste*r tersebut[16].
b. Kuadratkan dan jumlahkan: Kuadratkan semua jarak yang telah dihitung, lalu jumlahkan semua kuadrat tersebut.
c. Jumlahkan seluruh *cluster*: Ulangi langkah di atas untuk semua *cluster*, kemudian jumlahkan semua hasil penjumlahan.

Langkah-langkah:

a. Hitung jarak setiap data ke *centroid cluster*nya dan kuadratkan:
   1) Untuk setiap data di *Cluster 1*, hitung jaraknya ke *centroid Cluster 1*, lalu kuadratkan.
   2) Ulangi langkah yang sama untuk semua data di *Cluster 2*.
b. Jumlahkan semua hasil kuadrat dalam setiap *cluster*:
   1) Jumlahkan semua hasil kuadrat di *Cluster 1*.
   2) Jumlahkan semua hasil kuadrat di *Cluster 2*.

c. Jumlahkan hasil dari langkah 2:
Jumlahkan hasil penjumlahan dari *Cluster 1* dan *Cluster 2*. Hasil akhirnya adalah nilai WCSS.

Setelah menghitung semua jarak kuadrat, kita jumlahkan semua nilai di kolom "Jarak Kuadrat". Hasil penjumlahan inilah yang menjadi nilai WCSS.
Metode *elbow* digunakan untuk menentukan jumlah *cluster* optimal dengan memplot WCSS terhadap jumlah *cluster* (K). Grafik ini akan menunjukkan titik *elbow* di mana penurunan WCSS mulai melambat, menunjukkan jumlah *cluster* yang optimal.

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

X, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)

wcss = []

K_range = range(1, 11)

for k in K_range:
    kmeans = KMeans(n_clusters=k, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(K_range, wcss, marker='o')
plt.title('Grafik Elbow untuk Menentukan Jumlah Cluster Optimal')
plt.xlabel('Jumlah Cluster (K)')
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')
plt.grid(True)
plt.show()
```

Gambar 1.Visualisasi grafik elbow

Pada Gambar 1 kode *python* tersebut dirancang untuk melakukan *clustering* data menggunakan algoritma *K-Means* dan kemudian memvisualisasikan hasilnya untuk menentukan jumlah *cluster* yang optimal.

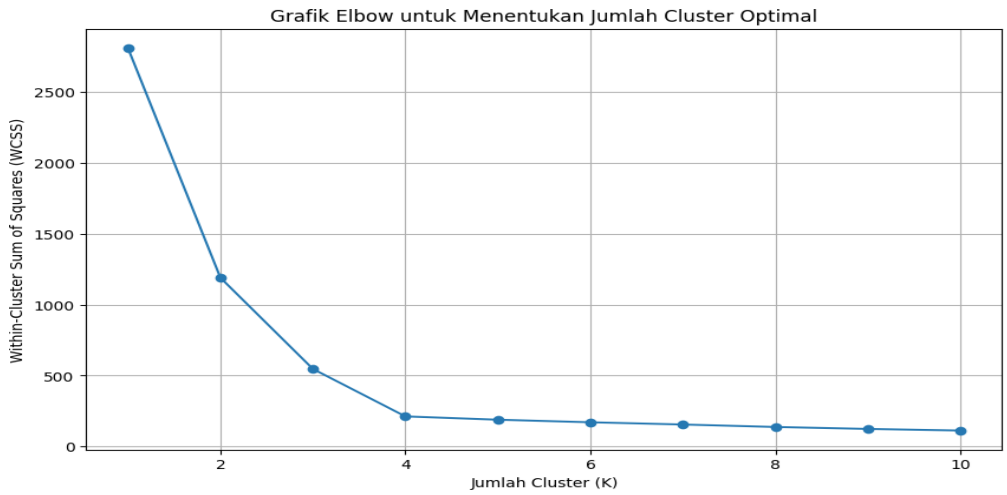
Dimana :

1. *Import Library:*
   a. *numpy* (sebagai np): Digunakan untuk operasi numerik pada *array*.
   b. *matplotlib.pyplot* (sebagai plt): Digunakan untuk visualisasi data.
   c. *sklearn.cluster*: Mengandung algoritma *clustering*, termasuk *K-Means*.
   d. *sklearn.datasets*: Menyediakan *dataset* contoh, seperti *make_blobs* yang digunakan di sini.
2. Membuat Data Sampel:
   a. *make_blobs*: Fungsi ini menghasilkan *dataset* dengan sejumlah titik data (sampel) yang terdistribusi dalam beberapa *cluster*. Parameter yang digunakan:
   b. *n_samples*: Jumlah sampel data.
   c. *centers*: Jumlah cluster yang diinginkan.
   d. *cluster_std*: Standar deviasi dari setiap *cluster*.
   e. *random_state*: Untuk memastikan hasil yang sama setiap kali kode dijalankan.
3. Inisialisasi *List* untuk WCSS:
   wcss = []: List kosong ini akan digunakan untuk menyimpan nilai *Within-Cluster Sum of Squares* (WCSS) untuk setiap nilai K (jumlah *cluster*).
4. Menentukan Rentang Nilai K:
   K_range = *range*(1, 11): Membuat list berisi nilai K dari 1 hingga 10. Kita akan mencoba nilai K dari 1 hingga 10 untuk melihat nilai K mana yang menghasilkan WCSS terendah.
5. *Looping* untuk Setiap Nilai K:
   a. *for k in K_range*: Melakukan iterasi untuk setiap nilai K dalam *K_range*.
   b. *kmeans* = *KMeans*(...): Membuat objek *K-Means* dengan parameter:

c. *n_clusters*: Jumlah *cluster* (sama dengan nilai K saat ini).
d. *init:* Metode inisialisasi *centroid*.
e. *max_iter*: Jumlah maksimum iterasi.
f. *n_init*: Jumlah kali algoritma dijalankan dengan *centroid* awal yang berbeda.
g. *random_state*: Untuk reproduksibilitas.
h. *kmeans.fit(X)*: Melatih model *K-Means* pada data X.
i. *wcss.append(kmeans.inertia_)*: Menambahkan nilai WCSS dari model *K-Means* ke dalam *list* WCSS.

6. Visualisasi Hasil:
   a. *plt.figure(figsize=(10, 6))*: Membuat *figure* dengan ukuran tertentu.
   b. *plt.plot(K_range, wcss, marker='o')*: Membuat *plot* garis antara nilai K dan nilai WCSS, dengan *marker* berupa lingkaran.
   c. *plt.title(...), plt.xlabel(...), plt.ylabel(...)*: Memberikan judul dan label pada sumbu x dan y.
   d. *plt.grid(True)*: Menampilkan *grid* pada *plot*.
   e. *plt.show()*: Menampilkan *plot*.

Plot yang dihasilkan adalah grafik *Elbow*. Titik belok (*elbow*) pada grafik menunjukkan jumlah *cluster* yang optimal. Pada titik ini, penurunan WCSS mulai melambat secara signifikan. Dengan memilih nilai K yang sesuai dengan titik belok, kita dapat menentukan jumlah *cluster* yang paling tepat untuk data kita[17].



*Gambar SEQ Gambar * ARABIC 2. Grafik Elbow*

Gambar 2. Grafik Elbow

Gambar 2 Sumbu X (Jumlah *Cluster* (K)): Menunjukkan jumlah *cluster* yang diuji. Semakin ke kanan, jumlah *cluster* semakin banyak.

Sumbu Y (*Within-Cluster Sum of Squares* (WCSS)): Menunjukkan total kuadrat jarak antara setiap data point dengan *centroid* (pusat) *cluster*nya. Semakin kecil nilai WCSS, semakin baik kualitas *clustering*-nya, karena data dalam setiap *cluster* semakin kompak.

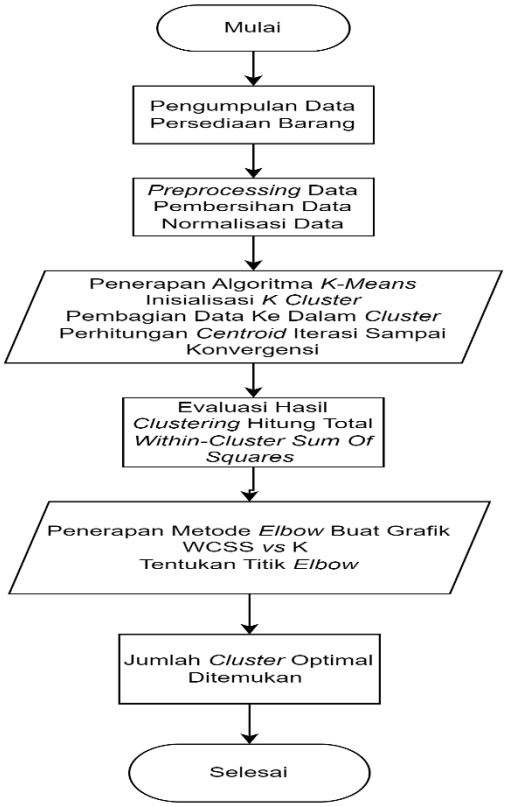
Grafik ini menunjukkan tren penurunan nilai WCSS seiring bertambahnya jumlah *cluster*. Ini masuk akal karena semakin banyak *cluster*, semakin dekat setiap data point dengan *centroid*nya.

Titik *Elbow*: Perhatikan bahwa penurunan nilai WCSS ini awalnya sangat drastis, lalu mulai melambat dan membentuk semacam "siku" atau "*elbow*". Titik inilah yang disebut sebagai titik *elbow*.

Menentukan K Optimal: Titik *elbow* dianggap sebagai indikasi jumlah *cluster* yang optimal. Pada titik ini, penambahan *cluster* baru tidak memberikan penurunan nilai WCSS yang signifikan lagi. Artinya, menambahkan lebih banyak *cluster* setelah titik *elbow* tidak memberikan peningkatan yang berarti dalam kualitas *clustering*.

Pemodelan

Pemodelan dalam penelitian ini mencakup penggunaan algoritma *K-Means* dan metode *elbow* untuk *clustering* data persediaan barang. Diagram alur proses (*flowchart*) digunakan untuk menggambarkan tahapan pemodelan, mulai dari pengumpulan data hingga evaluasi hasil *clustering*. Diagram UML juga digunakan untuk menggambarkan struktur dan interaksi dalam sistem yang dirancang.



*Gambar SEQ Gambar * ARABIC 3. Flowchart*

Gambar 3. Flowhchart Algoritma K-Means

Gambar 3 menggambarkan secara visual langkah-langkah yang dilakukan dalam sebuah proses *clustering* atau pengelompokan data persediaan barang menggunakan algoritma *K-Means*. Algoritma *K-Means* adalah salah satu metode *clustering* yang populer dan sering digunakan dalam *data mining*.

Metode *elbow* digunakan untuk menentukan jumlah *cluster* optimal dengan mem*plot Within-Cluster Sum of Squares* (WCSS) terhadap jumlah *cluster* (K). Grafik ini akan menunjukkan titik *elbow* di mana penurunan WCSS mulai melambat, menunjukkan jumlah *cluster* yang optimal.

Pada tahap pemodelan ini, dilakukan visualisasi data untuk memahami distribusi dan hubungan antara variabel dalam *dataset*. Salah satu cara yang digunakan adalah dengan membuat *plot scatter* antara jumlah barang dan data penjualan historis. Visualisasi ini berguna untuk mengidentifikasi pola-pola awal yang mungkin ada dalam data, yang kemudian dapat dijadikan dasar dalam penerapan algoritma *K-Means* untuk *clustering*. *Dataset* yang digunakan dalam penelitian ini terdiri dari beberapa fitur, di antaranya "Jumlah Barang" dan "Data Penjualan Historis". Fitur-fitur ini dipilih karena relevan untuk analisis persediaan barang, di mana "Jumlah Barang" mewakili kuantitas produk yang tersedia, sedangkan "Data Penjualan Historis" menggambarkan performa penjualan barang tersebut di masa lalu. Untuk memvisualisasikan hubungan antara kedua fitur ini, digunakan *plot scatter*, di mana setiap titik pada plot merepresentasikan satu item barang. Sumbu x pada plot menggambarkan jumlah barang yang tersedia, sedangkan sumbu y menggambarkan data penjualan historis. Dengan memplot kedua fitur ini, kita dapat melihat apakah ada hubungan *linier*, *Cluster*, atau *outlier* yang perlu diperhatikan.

```python
import pandas as pd
import matplotlib.pyplot as plt

data = pd.DataFrame({
    'ID Barang': [101, 102, 103, 104, 105, 106, 107, 108, 109, 110],
    'Nama Barang': ['Laptop', 'Kaos', 'Snack', 'Headphone', 'Sepatu', 'Sandal', 'Sabun', 'Sepeda', 'Kursi', 'Meja'],
    'Kategori Barang': ['Elektronik', 'Pakaian', 'Makanan', 'Elektronik', 'Pakaian', 'Pakaian', 'Benda', 'Benda', 'Benda', 'Benda'],
    'Jumlah Barang': [50, 200, 300, 75, 180, 80, 100, 40, 20, 20],
    'Data Penjualan Historis': [1000, 1500, 2000, 1200, 1300, 1400, 1500, 1600, 1700, 1800]
})

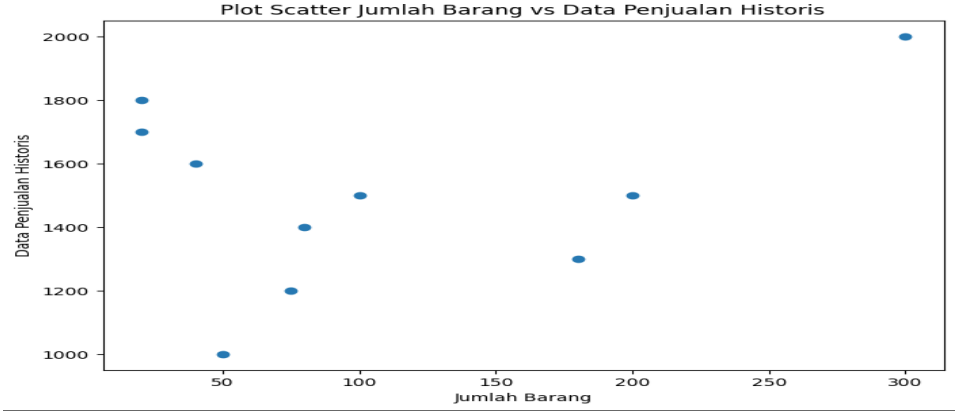
plt.figure(figsize=(8, 6))
plt.scatter(data['Jumlah Barang'], data['Data Penjualan Historis'])
plt.xlabel('Jumlah Barang')
plt.ylabel('Data Penjualan Historis')
plt.title('Plot Scatter Jumlah Barang vs Data Penjualan Historis')
plt.show()
```

Gambar 4. Kode Python

Code python tersebut bertujuan untuk :

1. Membuat *DataFrame* :
a. *import pandas as pd*: Mengimpor *library Pandas* untuk manipulasi data.
b. *data = pd.DataFrame(...)*: Membuat sebuah *DataFrame* (mirip tabel) yang berisi data tentang berbagai barang. Kolom-kolom dalam *DataFrame* ini mencakup ID barang, nama barang, kategori barang, jumlah barang, dan data penjualan historis.
2. Membuat *Plot Scatter*
a. *import matplotlib.pyplot as plt*: Mengimpor *library Matplotlib* untuk visualisasi data.
b. *plt.figure(figsize=(8, 6))*: Membuat *figure* (kanvas) untuk *plot* dengan ukuran 8x6 inci.
c. *plt.scatter(...)*: Membuat *scatter plot* dengan:
   1) x: Data pada kolom 'Jumlah Barang'.
   2) y: Data pada kolom 'Data Penjualan Historis'.
   3) Setiap titik pada plot merepresentasikan satu jenis barang. Posisi titik ditentukan oleh jumlah barang dan data penjualan historisnya.
d. *plt.xlabel(...), plt.ylabel(...), plt.title(...)*: Memberikan label pada sumbu x, sumbu y, dan judul pada *plot*.
e. *plt.show()*: Menampilkan *plot* yang telah dibuat.

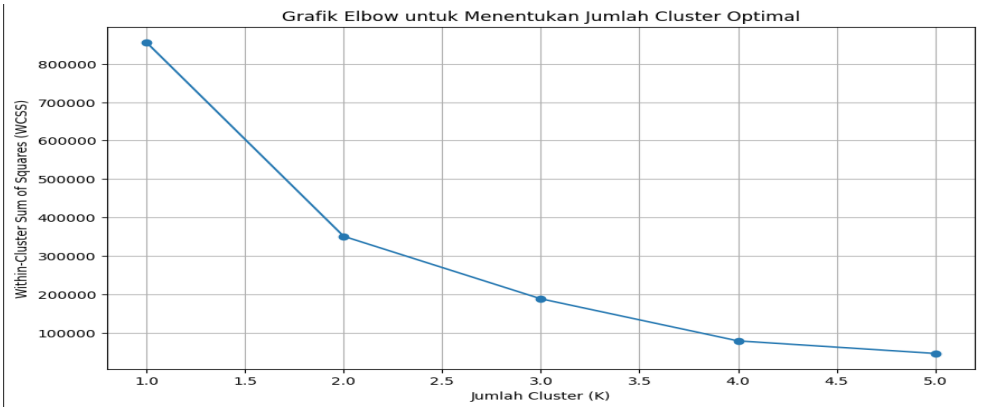
Code Python tersebut memberikan cara yang efektif untuk memvisualisasikan hubungan antara jumlah barang dan data penjualan historis. *Plot scatter* yang dihasilkan dapat memberikan wawasan awal tentang pola dan tren dalam data tersebut.



Gambar 5 Plot Scatter jumlah barang vs data penjualan historis

Gambar 5 menggambarkan hubungan antara jumlah barang yang tersedia dengan data penjualan historis. Setiap titik pada grafik mewakili satu pasangan data, di mana sumbu x menunjukkan jumlah barang dan sumbu y menunjukkan data penjualan historis.

Berdasarkan hasil pemodelan menggunakan algoritma *K-Means* dan metode *elbow*, grafik *elbow* berikut ini menunjukkan hubungan antara jumlah *cluster* (K) dan WCSS.



Gambar 6 Grafik Elbow Dataset

Pada Gambar 6 dapat diamati bahwa titik *elbow* berada pada K = 3, yang menunjukkan bahwa jumlah *cluster* optimal untuk data persediaan barang adalah 3. Pada titik ini, penurunan WCSS mulai melambat, yang mengindikasikan bahwa menambah lebih banyak *cluster* tidak memberikan peningkatan yang signifikan dalam variasi yang dijelaskan oleh model. Oleh karena itu, penggunaan 3 *cluster* dianggap paling efisien untuk mengelompokkan data persediaan barang berdasarkan pola penjualan dan karakteristik lainnya.

Kesimpulan

Penelitian ini menunjukkan bahwa penerapan metode *elbow* pada algoritma *K-Means* efektif dalam menentukan jumlah *cluster* yang optimal untuk *clustering* data persediaan barang. Dengan jumlah *cluster* yang optimal, manajemen persediaan barang dapat ditingkatkan efisiensinya, sehingga dapat mengurangi biaya dan meningkatkan kepuasan pelanggan. Metode *elbow* terbukti sebagai alat yang berguna dalam proses *clustering* dan dapat diterapkan pada berbagai jenis data lainnya

Daftar Pustaka

[1] N. A. Maori and E. Evanita, "Metode Elbow dalam Optimasi Jumlah Cluster pada K-Means Clustering," *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, vol. 14, no. 2, pp. 277–288, 2023, doi: 10.24176/simet.v14i2.9630.

[2] I. Wahyudi, M. B. Sulthan, and L. Suhartini, "Analisa Penentuan Cluster Terbaik Pada Metode K-Means Menggunakan Elbow Terhadap Sentra Industri Produksi Di Pamekasan," *Jurnal Aplikasi Teknologi Informasi dan Manajemen (JATIM)*, vol. 2, no. 2, pp. 72–81, 2021, doi: 10.31102/jatim.v2i2.1274.

[3] A. Winarta and W. J. Kurniawan, "Optimasi Cluster K-Means Menggunakan Metode Elbow Pada Data Pengguna Narkoba Dengan Pemrograman Python," *JTIK (Jurnal Teknik Informatika Kaputama)*, vol. 5, no. 1, pp. 113–119, Jan. 2021, doi: 10.59697/jtik.v5i1.593.

[4] Y. Jumaryadi, A. Wibowo, and M. Ilmu Komputer, "Analisis Pengelompokan Gangguan TIK Pada Sistem Pencatatan Layanan Menggunakan Algoritma K-Means dan Metode Elbow Analysis of Incidents Segmentation Based on Service Recording System Data Using K-Means Algorithm and Elbow Method," vol. 22, no. 2, pp. 348–357, 2023.

[5] M. Orisa, "Optimasi Cluster pada Algoritma K-Means," *Prosiding SENIATI*, vol. 6, no. 2, pp. 430–437, Jul. 2022, doi: 10.36040/seniati.v6i2.5034.

[6] M. Sholeh and K. Aeni, "Perbandingan Evaluasi Metode Davies Bouldin, Elbow dan Silhouette pada Model Clustering dengan Menggunakan Algoritma K-Means," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 8, no. 1, p. 56, 2023, doi: 10.30998/string.v8i1.16388.

[7] C. A. Sri Fastaf and Y. Yamasari, "Analisa Pemetaan Kriminalitas Kabupaten Bangkalan Menggunakan Metode K-Means dan K-Means++," *Journal of Informatics and Computer Science (JINACS)*, vol. 3, no. 04, pp. 534–546, 2022, doi: 10.26740/jinacs.v3n04.p534-546.

[8] F. M. Sarimole and L. Hakim, "Klasifikasi Barang Menggunakan Metode Clustering K-Means Dalam Penentuan Prediksi Stok Barang," *Jurnal Sains dan Teknologi*, vol. 5, no. 3, pp. 846–854, Feb. 2024, doi: 10.55338/saintek.v5i3.2709.

[9] R. Ishak and A. Bengnga, "Clustering Tingkat Pemahaman Mahasiswa Pada Perkuliahan Probabilitas Statistika Dengan Metode K-Means," *Jambura Journal of Electrical and Electronics Engineering*, vol. 4, no. 1, pp. 65–69, 2022, doi: 10.37905/jjeee.v4i1.11997.

[10] C. Darmawan, Y. Setiyawan, R. A. Prasetyo, and S. K. Qurrota'Ayyun, "Penerapan Algoritma K-means dan Metode Elbow Untuk Clustering Tingkat Pencemaran Sampah Plastik pada Kabupaten/Kota di Seluruh Indonesia," *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 1, pp. 349–358, 2024, doi: 10.33379/gtech.v8i1.3637.

[11] R. Yuliana Sari, H. Oktavianto, and H. Wahyu Sulistyo, "Algoritma K-Means Dengan Metode Elbow Untuk Mengelompokkan Kabupaten/Kota Di Jawa Tengah Berdasarkan Komponen Pembentuk Indeks Pembangunan Manusia K-Means Algorithm With Elbow Method To Grouping District/City in Central Java Based on Components of Human D," *Jurnal Smart Teknologi*, vol. 3, no. 2, pp. 2774–1702, 2022, [Online]. Available: http://jurnal.unmuhjember.ac.id/index.php/JST

[12] S. Nofita, H. H. Hanny, and M. S. Amril, "Implementasi Clustering Data Kasus Covid 19 Di Indonesia Menggunakan Algoritma K-Means," *Bianglala Informatika*, vol. 11, no. 1, pp. 7–12, 2023.

[13] F. Arifatul Ulya, A. N. Abdullah, T. Aisya Hanan, and I. Manfaati Nur, "Pengelompokkan Tingkat Pengangguran Terbuka Di Jawa Tengah Menggunakan Metode K – Means Clustering," *Journal Of Data Insights*, vol. 1, no. 2, pp. 71–80, 2023, doi: 10.26714/jodi.v1i2.148.

[14] V. A. Ekasetya and A. Jananto, "Klusterisasi Optimal Dengan Elbow Method Untuk Pengelompokan Data Kecelakaan Lalu Lintas Di Kota Semarang," *Jurnal Dinamika Informatika*, vol. 12, no. 1, pp. 20–28, 2020, doi: 10.35315/informatika.v12i1.8159.

[15] Taufik Hidayat, Mohamad Jajuli, and Susilawati, "Clustering Daerah Rawan Stunting Di Jawa Barat Menggunakan Algoritma K-Means," *INFOTECH : Jurnal Informatika & Teknologi*, vol. 4, no. 2, pp. 137–146, 2023, doi: 10.37373/infotech.v4i2.642.

[16] D. Indra, R. Satra, H. Azis, A. R. Manga, and H. L, "Detection System of Strawberry Ripeness Using K-Means," *ILKOM Jurnal Ilmiah*, vol. 14, no. 1, pp. 25–31, Apr. 2022, doi: 10.33096/ilkom.v14i1.1054.25-31.

[17] Purnawansyah, *Comparison Between K-Means and Fuzzy C-Means Clustering in Network Traffic Activities*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-59280-0.