TUGAS BESAR PEMBELAJARAN MESIN CLUSTERING SEGMENTASI PELANGGAN PADA DATA PENJUALAN RETAIL MENGGUNAKAN ALGORITMA K-MEANS

Disusun Dalam Rangka Memenuhi Sebagian Tugas Prodi Sains Data

Oleh

Auliya Ardhini Putri 1206230020 [1)]

Nur Nisrina Salsabilla 1206230030 [2)]

Ayunda Dewi Agustin 1206230040 [3)]

Universitas Telkom Surabaya

Program Studi Sains Data

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY SURABAYA

JUNI 2025

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan Tugas Besar mata kuliah Pembelajaran Mesin ini dengan judul "Segmentasi Pelanggan Pada Data Penjualan Retail menggunakan Algoritma K-Means" dengan lancar dan tepat waktu.

Tugas ini disusun sebagai bagian dari evaluasi akhir semester pada program studi Sains Data semester 4, dengan bimbingan dan arahan dari dosen pengampu, Bapak Hamim Zajuli Alfaroby. Melalui tugas besar ini, kami tidak hanya belajar secara teoritis mengenai konsep clustering dalam machine learning, tetapi juga mendapat pengalaman langsung dalam menerapkan algoritma clustering dari awal (coding from scratch), mulai dari eksplorasi data, pemodelan, evaluasi, hingga analisis eksperimen.

Kami menyadari bahwa tanpa dukungan dan kerja sama dari seluruh anggota kelompok, serta bantuan dari berbagai referensi dan sumber data terbuka, laporan ini tidak akan dapat diselesaikan dengan baik. Oleh karena itu, kami mengucapkan terima kasih kepada semua pihak yang telah membantu secara langsung maupun tidak langsung dalam proses penyusunan tugas besar ini.

Kami juga menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, kami sangat terbuka terhadap kritik dan saran yang membangun demi penyempurnaan tugas di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat dan menjadi bahan pembelajaran, baik bagi kami maupun bagi pembaca lainnya.

Surabaya, 10 Juni 2025

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Deskripsi Dataset

1.1.1 Retail-Sales-Analyzer.csv

Dataset ini berisi informasi transaksi pelanggan di berbagai kategori produk. Kolom-kolomnya mencakup ID pelanggan, kategori produk, jumlah yang dibeli, usia, gender, dan wilayah pelanggan.

1.1.2 Retail-Sales-Performance.csv

Dataset ini berisikan informasi mengenai performa penjualan yang berdasarkan wilayah dan waktu. Kolom-kolom dalam dataset ini mencakup tahun, bulan, total revenue, jumlah produk terjual, dan kode wilayahnya. Data ini memungkinkan analisis lebih lanjut terhadap tren penjualan dan kinerja di wilayah dalam periode waktu tertentu.

1.2 Permasalahan yang ingin diselesaikan

Dalam industry ritel, memahami perilaku konsumen dan performa penjualan secara mendalam merupakan kunci untuk meningkatkan efektivitas strategi pemasaran dan penjualan. Namun, dengan banyaknya transaksi pelanggan, beragam latar belakang demografis, serta variasi performa penjualan antar wilayah dan waktu, Perusahaan menghadapi tantangan dalam melakukan segmentasi pasar secara tepat dan efisien.

Permasalahan utama yang ingin diselesaikan dalam proyek ini adalah bagaimana melakukan segmentasi atau pengelompokan pelanggan dan wilayah penjualan berdasarkan data yang tersedia,seperti usia, gender, wilayah, kategori produk, jumlah pembelian, hingga performa penjualan. Secara khusus, proyek ini bertujuan untuk mengelompokkan pelanggan berdasarkan kombinasi atribut demografis, perilaku belanja, dan Tingkat kepuasan terhadap transaksi mereka, menggunakan algoritma clustering K-Means.

Dengan melakukan clustering menggunakan pendekatan unsupervised learning, Perusahaan diharapkan dapat menemukan pola tersembunyi dalam data, mengidentifikasi segmen pelanggan dengan karakteristik dan perilaku yang mirip, serta mengenali wilayah dengan performa serupa. Informasi ini dapat digunakan sebagai dasar untuk strategi pemasaran yang lebih terarah dan personalisasi layanan yang lebih tepat sasaran.

1.3 Tujuan clustering

Tujuan dari proyek ini untuk mengidentifikasi segmen-segmen pelanggan yang memiliki karakteristik dan yang berperilaku belanja yang serupa berdasarkan atribut yang tersedia, contohnya seperti usia, jenis kelamin, wilayah, jumlah pembelian, dan kategori produk. Dengan menerapkan algoritma K-Means, proses clustering dilakukan

untuk menemukan kelompok pelanggan yang secara alami terbentuk berdasarkan data demografis dan perilaku transaksi mereka.

Selain itu, hasil segmentasi ini diharapkan dapat memberikan insight yang bernilai bagi tim pemasaran atau manajemen Perusahaan dalam merancang strategi yang berbeda dan yang lebih spesifik untuk setiap kelompok pelanggan. Seperti strategi promosi atau penawaran produk, dan bisa saja personalisasi layanan yang dapat disesuaikan dengan karakteristik masing-masing segmen, sehingga dapat meningkatkan efektivitas kampanye pemasaran, kepuasan pelanggan, dan potensi peningkatan penjualan.

1.4 Manfaat

Penerapan metode clustering terhadap data pelanggan dan penjualan diharapkan dapat memberikn berbagai manfaat yang signifikan dalam konteks bisnis ritel. Manfaat – manfaat tersebut antara lain :

- Mempermudah dalam menentukan strategi pemasaran yang lebih personalisasi, dengan menyasar masing-masing segmen pelanggan sesuai karakteristik dan kebutuhannya.
- Menemukan segmen pelanggan potensial yang cocok untuk program loyalitas, pemberian reward, atau promosi khusus yang lebih tepat sasaran.
- Menyediakan dasar analisis yang kuat untuk pengambilan Keputusan bisnis berbasis data, sehingga Perusahaan dapat lebih responsive terhadap perilaku konsumen dan perubahan tren pasar.
- Meningkatkan efisiensi operasional dalam hal perencanaan penjualan dan distribusi, karena perusahaan dapat memetakan wilayah atau kelompok pelanggan berdasarkan potensi kontribusi terhadap pendapatan.

1.5 Relevansi topik

Topik clustering dalam analisis data penjualan dan perilaku konsumen sangat relevan dengan kebutuhan industry saat ini, terutama dalam era digital di mana jumlah data pelanggan semakin besar dan kompleks. Segmentasi pelanggan berbasis machine learning, khususnya dengan metode unsupervised seperti K-Means, memungkinkan Perusahaan untuk memahami struktur tersembunyi dalam data tanpa perlu label atau kategori yang sudah ditentukan sebelumnya.

Dalam konteks program studi Saind Data, topik ini juga sejalan dengan kompetensi yang harus dimiliki oleh mahasiswa, yaitu kemampuan untuk mrngolah, menganalisis, dan menafsirkan data untuk menghasilkan Solusi berdasarkan data. Penerapan clustering dari awal tanpa library juga memperkuat pemahaman konseptual dan tekknis mahasiswa terhadap dasar-dasar machine learning.

BAB II
EKSPLORASI & PERSIAPAN DATA

2.1 Dataset

2.1.1 Sumber Dataset

- infinite-dataset-Retail Sales Analyzer : https://huggingface.co/datasets/infinite-dataset-hub/RetailSalesAnalyzer/tree/main
- infinite-dataset-Retail Sales Performance : https://huggingface.co/datasets/infinite-dataset-hub/RetailSalesPerformance

2.1.2 Observasi Dataset

Kategori	Pengertian
Customer ID	ID pelanggan digunakan sebagai identik unik
Purchase Amount	Jumlah pembelian ini sangat penting untuk segmentasi
Customer Age Group	Kelompok umur pelanggan, kategori harus diubah ke numerik
Customer Gender	Kategori ubah ke angka
Customer Region	Lokasi pelanggan yang bisa dipertimbangkan atau di sederhanakan
Customer Statisfaction	Skor kepuasan pelanggan range 1 – 5
Retail Channel	Offline atau online diubah ke angka
Customer Income Level	Pendapatan pelanggan dengan kategori low medium high diubah ke angka

2.2 Deskripsi fitur

Kategori	Fungsi
Fitur Purchase Amount	Untuk mengukur perilaku belanja
Customer Satisfaction	Mencerminkan pengalaman pelanggan
Customer Age Group	Umur sebagai factor demografis
Customer Gender	Membedakan segmen pria dan Wanita
Retail Channel	Penjualan yang digunakan
Customer Income Level	Tingkat daya dari pelanggan

2.3 Praproses

- Buang Kolom Yang Tidak Relevan
- Gabungkan Nilai Duplikat
- Ubah Data Kategorik Menjadi Numerik :
  - Gender = Male : 0 | Female : 1
  - Retail Channel = Online 0 | Offline : 1
  - Income = Low : 0 | Medium : 1 | High : 2
  - Age : < 25 : 0 | 25 – 34 : 1 | 30 – 45 : 2 | 35 – 55 : 4 | 45 – 60 : 5 | >60 : 6

2.3.1 Handling missing values

Berdasarkan hasil pemeriksaan yang telah di proses oleh kami, diketahui bahwa dataset Retail-Sales_Analyzer.csv dan Retail-Sales-Performance.csv tidak ada nilai hilang pada kolom manapun. Oleh karena itu, tidak diperlukan hal tambahan seperti penghapusan data atau pengisian data yang hilang.

Langkah ini adalah Langkah yang penting dilakukan sebagai bentuk validasi awal terhadap kualitas datanya, sehingga dapat dipastikan bahwa proses praproses ini dapat dilakukan tanpa gangguang dari data yang tidak lengkap maupun data yang hilang.

2.3.2 Normalisasi

Pada tugas ini, kami melakukan proses normalisasi untuk fitur-fitur numerik yang akan digunakan untuk keperluan clustering, seperti PurchaseAmount, Satisfaction, CustomerAgeGroup, Gender, Channel, dan Income. Normalisasi ini kami lakukan tujuannya agar di setiap fitur berada pada skala yang seragam, sehingga algoritma K-Means dapat menghitung jarak antar data secara adil tanpa bias terhadap fitur yang memiliki skala lebih besar.

Metode yang kami gunakan untuk normalisasi adalah Min-Max Scaling, yang mengubah setiap nilai dalam fitur ke dalam rentang [0,1]. Proses ini dilakukan menggunakan fungsi MinMaxScaler dari library Scikit-learn. Rumus yang kami gunakan dalam metode ini adalah :

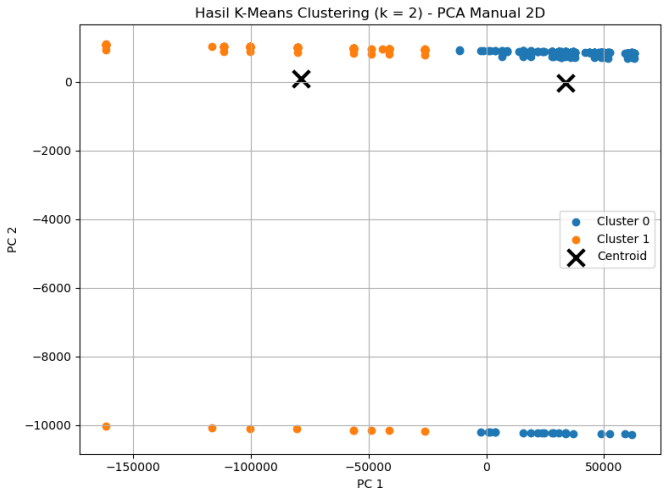
$$x' = (x - \min)/(\max - \min)$$

Setelah proses normalisasi, data akan disimpan ke dalam variabel X_preprocessed dan digunakan sebagai input untuk proses clustering. Dengan normalisasi ini, semua fitur memiliki kontribusi yang seimbang dalam penentuan klaster oleh algoritma K-Means.

2.3.3 Visualisasi awal

Visualisasi awal dilakukan untuk memahami struktur data setelah dilakukan normalisasi dan sebelum masuk ke tahap evaluasi hasil clustering. Salah satu metode visualisasi yang digunakan oleh kami adalah Principal Component Analysis (PCA) untuk mereduksi dimensi data menjadi dua komponen utama (PC1 dan PC2), sehingga dapat divisualisasikan secara dua dimensi.

Gambar di bawah ini menunjukkan hasil clustering awal yang menggunakan algoritma K-Means dengan nilai k = 2 terhadap data asli yang telah kami proses. Warna yang berbeda ini menunjukkan klaster yang terbentuk, sementara titik centroid masing-masing klister ditandai dengan symbol “X” berwarna hitam.



Berdasarkan visualisasi diatas ini terlihat bahwa data terbagi menjadi dua klaster yang memiliki penyebaran yang cukup terpisah secara horizontal (PC1), walaupun ada sekelompok titik dengan nilai yang ekstrim di sumbu yang vertical (PC2). Hal ini menunjukkan bahwa K-Means mampu mengetahui perbedaan yang dominan di antara data, tapi juga mengetahui adanya sebaran nilai ekstriim yang dapat mempengaruhi interpretasi lebih lanjut. Visualisasi ini menjadi landasan awal untuk mengevaluasi jumlah klister optimal dan kualitas segmentasi data.

2.3.4 Justifikasi Teknik

Pemilihan teknik praproses dalam tugas ini dilakukan secara terstruktur untuk memastikan kualitas data uang optimal sebelum dimasukkan ke dalam algoritma clustering. Fitur-fitur yang sudah kami gunakan kami pilih berdasarkan relevansi terhadap karakteristik pelanggan, seperti jumlah pembelian, tingkat kepuasan, dll. Di semua fitur kateegorikal dikonversi ke bentuk numerik melalui teknik label encoding agar dapat diproses di algoritma K-Means, yang hanya mendukung data numerik.

Normalisasi data dilakukan memakai metode Min-Max Scaling bertujuan untuk setiap fitur yang ada berada di skala yang beragam, yaitu dalam rentang [0,1]. Hal ini sangat penting karena K-Means ini menghitung jarak antar data menggunakan Euclidean distance, dan tanpa normalisasi, fitur dengan skala yang lebih besar ini mendominasi proses pembentukan klaster.

Selain itu, adapun teknik principal Component Analysis (PCA) yang digunakan untuk mereduksi dimensi data menjadi dua komponen utama yang memungkinkan visualisasi klaster secara dua dimensi. Teknik ini menurut kami sangat berguna karena dalam tahap awal yang untuk memahami pola pemisahan antar klaster dan memvalidasi hasil clustering secara visual.

Dengan kombinasi teknik praproses yang sudah dijelaskkan, proses clustering menjadi lebih terarah, jelas, efisien, dan sudah menghasilkan segmentasi data yang representatif.

BAB III
PEMODELAN

3.1 Penjelasan Algoritma

Algoritma yang digunakan dalam tugas besar ini adalah K – Means Clustering, yaitu salah satu metode unsupervised learning yang digunakan untuk mengelompokkan data berdasarkan kemiripan karakteristiknya, Algoritma ini bekerja dengan cara membagi data ke dalam sejumlah kelompok (cluster) yang telah di tentukan sebelumnya melalui parameter K. Tujuan utama dari algoritma ini adalah untuk meminimalkan nilai Within Cluster Sum of Squares (WCSS), yakni total jarak kuadrat antara setiap titik data dan pusat klusternya (centroid). Semakim kecil nilai WCSS, semakin baik kualitas pengelompokan karena menunjukkan bahwa data dalam kluster tersebut salig berdekatan.

Proses kerja algoritma k – means di mulai dengan pemilihan centroid awal secara acak sebanyak K buah. Setiap data kemudian dihitung jaraknya terhadap semua centroid menggunakan rumus jarak Euclidean. Data akan diklasifikasikan ke dalam kluster dengan jarak terdekat. Setelah semua data terkelompokkan, posisi centroid diperbarui berdasarkan rata-rata koordinat semua data dalam kluster tersebut. Langkah ini di ulang secara iteratif hingga posisi centroid tidak berubah signifikan atau jumlah iterasi maksimum tercapai.

Evaluasi kualitas hasil klasterisasi di lakukan menggunakan dua metrix, yaitu Within-Cluster Sum of Squares (WCSS) dan Silhoutte Score. WCSS digunakan untuk mengukur kekompakan data dalam satu kluster. Nilai WCSS yang lebih kecil menunjukkan bhawa titik – titik data lebih dekat ke pusat klusternya. Namun, WCSS tidak cukup untuk menentukan jumlah kluster terbaik. Oleh karena itu, digunakan juga Silhouette Score yang memberikan penilaian terhadap kualitas pengelompokkan berdasarkan nilai K tertentu.

3.2 Penjelasan Cara Kerja Model Dari Awal

Model K-Means pada projek ini dibangun sepenuhnya dari awal (form scartch) menggunakan pemrograman Python, tanpa menggunakan library clustering bawaah seperti Scikit-learn. Seluruh komponen utama, mulai dari perhitungan jarak antar data, inisialisasi centroid, hingga iterasi pembaruan klaster, ditulisa secara manual untuk memahami lebih dalam prinsip kerja olgoritma**.**

Langkah – Langkah implementasi model dilakukan seperti berikut :

a) Data yang telah di normalisasi (X_preprocessed) dikonversi ke dalam bentuk array numerik (numpy array) untuk mempermudah perhitungan.
b) Fungsi jarak Euclidean didefinisikan untuk mengukur jarak antara dua titik data. Rumus yang digunakan adalah:

$$d(x-y) = \sqrt{(xi-yi)^2}$$

c) Fungsi kmeans_from_scratch untuk melakukan proses clustering, yang terdiri dari :
   - Inisialisasi centroid secara acak dari data sebanyak K buah
   - Perhitungan jarak setiap titik data ke seluruh centroid
   - Penentuan klister berdasarkan jarak terdekat
   - Pembaruan posisi centroid berdasarkan rata – rata titik dalam klaster
   - Pengulangan proses hingga posisi centroid konvergen atau iterasi maksimum tercapai

d) Setelah proses clustering dilakukan, maka kami melakukan evalusasi model menggunakan dua metrik:
   - Within-Cluster Sum of Squares (WCSS) untuk mengukur kekompakan klister
   - Silhouette Score manuall untuk menilai pemisahan antar klister

e) Untuk keperluan visualisasi, kami mereduksi data ke dalam dua dimensi menggunakan Principal Component Analysis (PCA), dan hasil clustering divisualisasikan dalam bentuk scatter plot 2D.

Seluruh proses ini ditulis dan dijalankan oleh kami dalam satu notebook (Jupyter Notebook), tak lupa juga kami mencoba di berbagai device agar hasilnya bisa kami evaluasi apakah error atau tidak, sehingga alur kerja dari preprocessing hingga visualisasi dapat dilakukan secara terintegrasi.

3.3 Source Code Snippet Penting

Untuk mendukung pemahaman proses kerja model K-Means yang dibangun dari awal, berikut adalah beberapa potongan kode (snippet) penting yang digunakan dalam implementasi algoritma :

a) Fungsi Perhitungan Jarak Euclidean
Digunakan untuk mengukur kedekatan anatara titik data dengan centroid:

```python
def euclidean_distance(a, b):
    return np.sqrt(np.sum((a - b) ** 2))
```

b) Fungsi K-Means from Scratch
Fungsi utama untuk melakukan clustering:

```python
def kmeans_from_scratch(X, k=3, max_iter=100, tol=1e-4,
seed=42):
    np.random.seed(seed)
    n_samples, n_features = X.shape
    centroids = X[np.random.choice(n_samples, size=k,
replace=False)]

    for iteration in range(max_iter):
        labels = []
        for x in X:
            distances = [euclidean_distance(x, centroid) for
centroid in centroids]
            labels.append(np.argmin(distances))

        labels = np.array(labels)
        new_centroids = np.zeros_like(centroids)

        for i in range(k):
            cluster_points = X[labels == i]
            if len(cluster_points) > 0:
                new_centroids[i] = np.mean(cluster_points,
axis=0)

        if np.linalg.norm(new_centroids - centroids) < tol:
            break
        centroids = new_centroids

    return labels, centroids, iteration + 1
```

c) Evaluasi dengan WCSS dan Silhouette Score Manual

```python
def compute_wcss(X, labels, centroids):
    wcss = 0
    for i in range(len(centroids)):
        points = X[labels == i]
        for point in points:
            wcss += np.sum((point - centroids[i]) ** 2)
    return wcss
```

```python
def silhouette_score_manual(X, labels):
    n = len(X)
    k = np.max(labels) + 1
    scores = []

    for i in range(n):
        xi = X[i]
        same_cluster = X[labels == labels[i]]
        a = np.mean([euclidean_distance(xi, xj) for xj in same_cluster if not np.allclose(xi, xj)])
        b = np.min([
            np.mean([euclidean_distance(xi, xj) for xj in X[labels == j]])
            for j in range(k) if j != labels[i]
        ])
        scores.append((b - a) / max(a, b))

    return np.mean(scores)
```

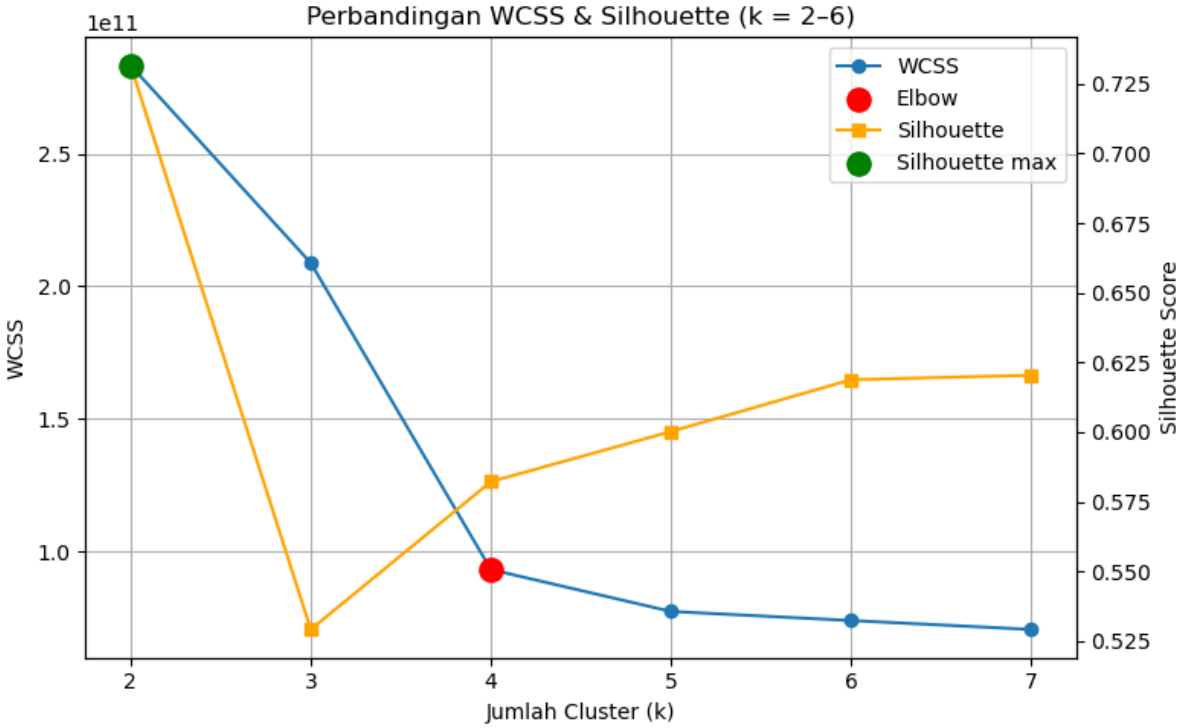
d) Visualisasi PCA Manual

```python
def manual_pca_2d(X):
    X_centered = X - np.mean(X, axis=0)
    cov_matrix = np.cov(X_centered, rowvar=False)
    eig_vals, eig_vecs = np.linalg.eigh(cov_matrix)
    top2_vecs = eig_vecs[:, np.argsort(eig_vals)[-2:]]
    return X_centered @ top2_vecs
```

3.4 Visualisasi Terakhir

Setelah proses clustering dilakukan pada data yang sudah kami proses, kami melakukan evaluasi terhadap beberapa variasi jumlah klaster (nilai k) untuk menentukan konfigurasi terbaik. Evaluasi yang kami lakukan ini menggunakan dua metrik utaa, yaitu WCSS dan Silhouette Score. WCSS ini kami gunakan untuk mengukur severapa kompak titik-titik dalam suatu klister, sementara untuk Silhouette Score ini untuk mengukur seberapa baik pemisahan antar klaster terhadap data lainnya.

Gambar dibawah ini menunjukkan bahwa visualisasi perbandingan nilai WCSS dan Silhaouette Score untuk variasi jumlah klaster dari k – 2 hingga k = 7:



Dari visualisasi yang sudah kami jabarkan, bisa dilihat bahwa nilai WCSS ini terus menurun seiring bertambahnya jumlah klaster, yang hal ini itu merupakan

karakteristik umum K-means. Tetapi, untuk menentukan titik optimum kami menggunakan dua pendekatan:

a) Titik Elbow (k=4): Terjadi penurunan WCSS yang signifikan dari k = 3 ke k = 4, tetapi penurunan yang di berikan mulai mendatar setelahnya. Hal ini menunjukkan bawha k = 4 dapat digunakan sebagai titik elbow, di mana penambahan klaster setelahnya memberikan manfaat tambahan.
b) Nilai Silhouette Score maksimum (k = 2): Skor tertinggi diperoleh saat k = 2 yaitu 0.7316, yaang menandakan bahwa dengan dua klaster saja, pemisahan antar kelompok cukup jelas dan data dalam tiap klaster relatif seragam.

Berdasarkan kedua indikator tersebut, baik k = 2 maupun k = 4 dapat menjadi kandidat konfigurasi klaster yang optinal. Tetapu, pemilihan akhir dapat mempertimbangkan konteks bisnis dan iterpretasi segmenrasi yang di inginkan

BAB IV
EVALUASI

4.1 Metode Evaluasi (Silhoutte, WCSS)

Dalam evaluasi performa model clustering, digunakan dua metode evaluasi utama yaitu:

1. Silhouette Score
   Silhouette Score adalah metrik yang mengukur seberapa baik suatu data berada dalam cluster nya sendiri dibandingkan dengan cluster lain. Nilai Silhouette Score berkisar dari -1 hingga 1:
   - Nilai mendekati 1 berarti data sangat cocok dengan cluster nya dan tidak cocok dengan cluster lain.
   - Nilai mendekati 0 berarti data berada di antara dua cluster.
   - Nilai mendekati -1 berarti data mungkin salah pengelompokan.

   Dengan kata lain, semakin tinggi nilai Silhouette Score, semakin baik pemisahan antar cluster dan semakin kompak data dalam masing-masing cluster. Metrik ini sangat berguna dalam mengevaluasi kualitas spasial dari hasil clustering, terutama dalam konteks K-Means.

2. WCSS (Within-Cluster Sum of Squares)
   WCSS atau SSE adalah total dari kuadrat jarak antara setiap titik data ke centroid dari cluster tempat ia tergabung. Metrik ini mengukur kompaksi dari sebuah cluster: semakin kecil WCSS, semakin dekat data-data dalam satu cluster terhadap centroid-nya, artinya semakin bagus hasil clustering.

   Dalam praktiknya, WCSS digunakan untuk metode Elbow sebuah pendekatan visual untuk menentukan jumlah cluster optimal. Metode ini bekerja dengan mencari "titik siku" (elbow point) dari grafik WCSS terhadap jumlah cluster, di mana setelah titik tersebut penurunan WCSS menjadi tidak signifikan. Titik ini dianggap sebagai jumlah cluster optimal karena menyeimbangkan antara kompleksitas model dan kualitas hasil clustering.

4.2 Penjelasan alasan pemilihan

Alasan pemilihan kedua metode evaluasi ini didasarkan pada pertimbangan sebagai berikut :

1. Silhouette Score dipilih karena mampu memberikan gambaran yang intuitif dan matematis terhadap kualitas cluster dari sisi *separability* dan *compactness*. Dalam konteks data penjualan (retail sales), penting untuk mengetahui apakah segmentasi pelanggan atau produk yang dilakukan sudah cukup berbeda secara signifikan, dan apakah data dalam setiap kelompok cukup homogen. Nilai Silhouette yang tinggi memberikan keyakinan bahwa klasterisasi yang dihasilkan benar-benar memisahkan kelompok yang berbeda secara bermakna.

2. WCSS (Elbow Method) dipilih karena merupakan salah satu metode paling umum dan sederhana untuk menentukan jumlah cluster optimal secara visual. Dengan menampilkan grafik WCSS terhadap nilai *k*, kita bisa menemukan jumlah cluster yang cukup representatif tanpa membuat model terlalu kompleks (overfitting). Elbow method juga membantu dalam mengevaluasi trade-off antara jumlah cluster dan hasil segmentasi.

   Kombinasi dari dua metode ini memberikan evaluasi yang lebih menyeluruh, karena satu menilai kualitas cluster (Silhouette), sedangkan yang lain menilai efisiensi penyebaran data terhadap pusat cluster-nya (WCSS). Dengan menggunakan keduanya, kita bisa menyeimbangkan antara kualitas segmentasi dan kompleksitas model.

4.3 Nilai hasil evaluasi

Berdasarkan eksperimen yang telah dilakukan pada data, didapatkan hasil sebagai berikut :

Jumlah cluster (k)	WCSS	Silhoutte Score
K = 2	283188495033.65	0.7316
K = 3	208799390255.83	0.5292
K = 4	93295374810.04	0.5882
K = 5	77520811767.11	0.6002
K = 6	74105123958.65	0.6188
K = 7	70734957498.48	0.6204

- Nilai Silhouette Score tertinggi terjadi pada k = 2, yang menunjukkan bahwa dengan 2 cluster, data memiliki pemisahan yang paling jelas dan konsisten.
- Namun, jika dilihat dari sisi WCSS, terjadi penurunan yang sangat signifikan dari k = 3 ke k = 4, menunjukkan bahwa k = 4 merupakan titik elbow, yaitu jumlah cluster optimal sebelum penurunan WCSS menjadi tidak signifikan.
- Meskipun Silhouette Score meningkat sedikit setelah k = 4, peningkatan tersebut tidak sebanding dengan kenaikan kompleksitas jumlah cluster. Sehingga, k = 4 menjadi pilihan yang seimbang antara efisiensi dan kualitas.

4.4 Interpretasi hasil

Dari hasil evaluasi di atas, terdapat dua rekomendasi utama:

1. Rekomendasi Pertama: **k = 2**
   - Memberikan segmentasi paling sederhana dan memiliki Silhouette Score tertinggi.
   - Dapat diartikan bahwa seluruh data bisa dikategorikan menjadi dua kelompok besar yang sangat berbeda.
   - Cocok jika perusahaan hanya ingin membedakan dua tipe pelanggan utama. misalnya pelanggan besar vs pelanggan kecil, atau pelanggan aktif vs pasif.
   - Cocok untuk strategi umum atau kebijakan luas, seperti promosi berbasis dua segmen pasar.

2. Rekomendasi Kedua**: k = 4**
    - Diidentifikasi sebagai titik elbow, di mana penurunan WCSS setelahnya tidak lagi signifikan.
    - Memberikan segmentasi yang lebih granular namun tetap efisien, misalnya menjadi 4 kelompok pelanggan berdasarkan pola pembelian, volume transaksi, atau loyalitas.
    - Cocok untuk strategi pemasaran yang lebih terpersonalisasi, seperti segmentasi pelanggan VIP, Diskon berdasarkan perilaku pembelian, Rekomendasi produk spesifik, dan Alokasi stok berdasarkan area dominan pembelian.

Penggunaan **k = 2** akan cocok bila tujuan segmentasi adalah untuk membedakan kategori besar secara garis besar. Ini memberikan stabilitas dan kejelasan yang tinggi pada struktur klaster.

Penggunaan **k = 4** akan lebih informatif jika diperlukan segmentasi yang lebih bernuansa dan kontekstual sesuai kebutuhan bisnis yang kompleks, seperti peningkatan loyalitas pelanggan, optimalisasi pengiriman, atau pemetaan wilayah penjualan.

BAB V
EKSPERIMEN

5.1 Tabel eksperimen

No	Preprocessing	Jumlah Klaster (k)	Inisialisasi Centroid	Evaluasi	Hasil Evaluasi	Catatan
1	Min-Max Scaling	2	Random	WCSS, Silhouette Score	WCSS: 283.1 M Silhouette: **0.7316**	Skor Silhouette tertinggi, cocok untuk dua kelompok besar pelanggan
2	Min-Max Scaling	3	Random	WCSS, Silhouette Score	WCSS: 208.8 M Silhouette: 0.5292	Segmentasi mulai terlihat tapi kurang optimal
3	Min-Max Scaling	4	Random	WCSS, Silhouette Score	WCSS: **93.2 M** Silhouette: 0.5882	Titik elbow, keseimbangan antara kompleksitas dan performa
4	Min-Max Scaling	5	Random	WCSS, Silhouette Score	WCSS: 77.5 M Silhouette: 0.6002	Kualitas klaster meningkat, namun kompleksitas juga meningkat
5	Min-Max Scaling	6	Random	WCSS, Silhouette Score	WCSS: 74.1 M Silhouette: 0.6188	Segmentasi lebih detail, perbedaan antar klaster cukup signifikan
6	Min-Max Scaling	7	Random	WCSS, Silhouette Score	WCSS: 70.7 M Silhouette: 0.6204	Peningkatan tidak signifikan, kompleksitas mulai berlebihan

5.1.1 Parameter/teknik yang diubah

Dalam proses eksperimen yang dilakukan, terdapat beberapa parameter yang diubah untuk melihat pengaruhnya terhadap hasil segmentasi menggunakan algoritma K-Means. Parameter utama yang divariasikan adalah jumlah klaster (k), yaitu dari nilai 2 hingga 7. Tujuan dari variasi ini adalah untuk mengetahui konfigurasi jumlah klaster yang paling optimal dalam membagi data pelanggan ke dalam kelompok-kelompok yang homogen. Selain jumlah klaster, metode inisialisasi centroid juga menjadi parameter yang diuji, di mana setiap eksperimen dilakukan dengan pemilihan centroid awal secara acak dari data yang telah diproses.

Seluruh eksperimen menggunakan data yang sama, yaitu data pelanggan retail yang telah melalui proses praproses dan dinormalisasi dengan metode *Min-Max Scaling*, sehingga setiap fitur numerik berada pada rentang nilai yang seragam. Normalisasi ini penting untuk memastikan bahwa perhitungan jarak antar data dalam algoritma K-Means tidak dipengaruhi oleh perbedaan skala antar fitur. Parameter lainnya, seperti jumlah iterasi maksimum dan toleransi konvergensi, sengaja tidak diubah dan tetap dijaga konsisten pada setiap eksperimen. Hal ini dilakukan untuk menjaga keadilan dalam evaluasi, sehingga hasil dari setiap konfigurasi jumlah klaster dapat dibandingkan secara objektif tanpa adanya pengaruh dari faktor eksternal lain.

5.1.2 Evaluasi Hasil

Evaluasi hasil dari setiap konfigurasi eksperimen dilakukan dengan menggunakan dua metrik utama yang umum digunakan dalam metode clustering, yaitu *Silhouette Score* dan *Within-Cluster Sum of Squares* (WCSS). Kedua metrik ini digunakan secara bersamaan untuk memberikan gambaran yang menyeluruh mengenai performa pengelompokan data yang dilakukan oleh algoritma K-Means. Silhouette Score berperan dalam mengukur seberapa baik suatu data ditempatkan di dalam klasternya masing-masing, serta seberapa jauh jaraknya dari klaster lain. Skor Silhouette bernilai antara -1 hingga 1, di mana nilai yang semakin mendekati 1 menunjukkan bahwa data berada dalam klaster yang tepat dan terpisah secara signifikan dari klaster lainnya. Dengan demikian, semakin tinggi nilai Silhouette Score, semakin baik kualitas pemisahan antar klaster.

Sementara itu, WCSS digunakan untuk mengukur kekompakan atau homogenitas klaster, yaitu dengan menghitung jumlah kuadrat jarak antara setiap titik dalam klaster dengan centroid-nya. Nilai WCSS yang kecil mengindikasikan bahwa data dalam satu klaster berada dekat satu sama lain dan tidak tersebar terlalu jauh dari pusat klaster, yang merupakan ciri dari klaster yang kompak dan baik. Salah satu pendekatan populer dalam mengevaluasi WCSS adalah metode elbow, yaitu dengan mencari titik di mana penurunan nilai WCSS mulai melambat secara signifikan meskipun jumlah klaster terus bertambah.

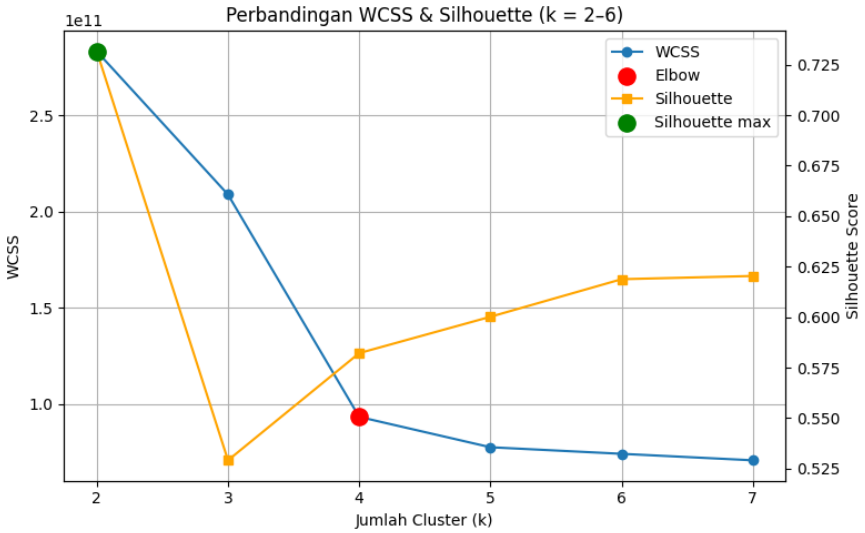
Berdasarkan hasil eksperimen yang telah dilakukan, ditemukan bahwa konfigurasi dengan nilai $k = 2$ menghasilkan nilai Silhouette Score tertinggi, yaitu sebesar 0.7316. Hal ini menunjukkan

bahwa dengan membagi data ke dalam dua klaster saja, pemisahan antar kelompok pelanggan sudah sangat baik dan masing-masing klaster terbentuk dengan struktur internal yang cukup kuat dan konsisten. Klaster yang terbentuk pada konfigurasi ini memiliki perbedaan karakteristik yang jelas, dan secara visual pun dapat dilihat bahwa pemisahan antar kelompok cukup tajam. Konfigurasi ini sangat cocok digunakan apabila perusahaan ingin menerapkan segmentasi sederhana, misalnya membedakan antara pelanggan potensial dan pelanggan reguler, atau antara pelanggan baru dan pelanggan loyal.

Di sisi lain, analisis terhadap nilai WCSS menunjukkan bahwa penurunan paling tajam terjadi saat jumlah klaster ditingkatkan dari tiga menjadi empat. Pada titik ini, nilai WCSS mengalami penurunan yang cukup besar, dan setelah $k = 4$, penurunan nilai WCSS menjadi relatif landai. Hal ini menjadi indikasi kuat bahwa titik elbow berada pada $k = 4$. Dengan kata lain, jumlah klaster empat merupakan titik optimal di mana efisiensi model dalam mengelompokkan data mencapai puncaknya tanpa menambah kompleksitas yang tidak perlu. Klasterisasi dengan empat kelompok memberikan segmentasi yang lebih kaya dan dapat dimanfaatkan untuk strategi yang lebih mendalam, seperti personalisasi layanan, penawaran produk yang disesuaikan dengan kebutuhan pelanggan, atau program loyalitas berdasarkan perilaku konsumen yang lebih spesifik.

Dengan menggabungkan analisis dari kedua metrik tersebut, dapat disimpulkan bahwa terdapat dua konfigurasi yang paling menonjol dalam eksperimen ini, yaitu $k = 2$ dan $k = 4$. Konfigurasi pertama menawarkan pemisahan antar klaster yang paling bersih dan sederhana, sementara konfigurasi kedua memberikan keseimbangan terbaik antara kedalaman segmentasi dan efisiensi model. Pemilihan konfigurasi akhir sangat bergantung pada tujuan strategis dari perusahaan dalam memanfaatkan hasil segmentasi untuk kepentingan bisnis yang lebih luas.

5.2 Grafik perbandingan hasil



Untuk mendukung dan memvisualisasikan hasil evaluasi dari berbagai konfigurasi jumlah klaster, dibuat dua grafik utama yang menampilkan hubungan antara nilai *k* (jumlah klaster) dengan masing-masing metrik evaluasi, yaitu *Within-Cluster Sum of Squares* (WCSS) dan *Silhouette Score*. Kedua grafik ini bertujuan untuk membantu mengidentifikasi jumlah klaster yang optimal secara visual, sekaligus memberikan pemahaman intuitif terhadap performa model pada masing-masing konfigurasi.

Grafik pertama merupakan grafik WCSS terhadap jumlah klaster (*k*). Grafik ini menunjukkan bahwa nilai WCSS mengalami penurunan yang sangat signifikan dari $k = 2$ hingga $k = 4$. Penurunan yang tajam ini menunjukkan bahwa penambahan jumlah klaster hingga empat kelompok memberikan kontribusi yang cukup besar dalam meningkatkan kekompakan klaster. Namun setelah mencapai $k = 4$, laju penurunan WCSS menjadi relatif lambat dan mendatar. Hal ini mengindikasikan bahwa mulai dari $k = 5$ ke atas, penambahan jumlah klaster tidak lagi memberikan peningkatan efisiensi yang signifikan. Dengan demikian, $k = 4$ dapat ditetapkan sebagai titik elbow, yaitu titik di mana nilai WCSS cukup rendah namun tanpa menimbulkan kompleksitas model yang berlebihan. Titik elbow ini menjadi referensi penting dalam memilih jumlah klaster optimal berdasarkan keseimbangan antara kualitas pengelompokan dan kesederhanaan model.

Grafik kedua adalah grafik Silhouette Score terhadap nilai *k*. Pada grafik ini terlihat bahwa nilai Silhouette Score tertinggi dicapai pada $k = 2$, yaitu sebesar 0.7316. Nilai tersebut menandakan bahwa dengan dua klaster saja, pemisahan antar kelompok pelanggan sudah sangat jelas dan kualitas internal masing-masing klaster juga sangat baik. Setelah $k = 2$, nilai Silhouette Score mengalami penurunan yang cukup signifikan pada $k = 3$, dan kemudian perlahan-lahan mengalami kenaikan kembali pada $k = 5$ hingga $k = 7$. Namun, kenaikan tersebut tidak pernah kembali mencapai skor awal yang tinggi seperti pada $k = 2$. Hal ini menunjukkan bahwa konfigurasi $k = 2$ memberikan hasil pemisahan klaster terbaik dari sisi kedekatan internal klaster dan jauhnya antar klaster. Meskipun konfigurasi dengan lebih banyak klaster memungkinkan segmentasi yang lebih rinci, dari segi kualitas pemisahan, $k = 2$ tetap menjadi pilihan terbaik.

Berdasarkan hasil visualisasi dari kedua grafik tersebut, dapat disimpulkan bahwa terdapat dua konfigurasi yang paling menonjol. Pertama, $k = 2$ cocok digunakan untuk keperluan segmentasi sederhana dan langsung diterapkan. Konfigurasi ini sangat ideal untuk skenario bisnis yang membutuhkan klasifikasi pelanggan secara cepat dan jelas, seperti membedakan pelanggan loyal dengan pelanggan umum, atau pelanggan baru dengan pelanggan lama. Segmentasi ini juga cocok untuk strategi komunikasi massal yang tidak memerlukan penyesuaian khusus per kelompok.

Sementara itu, konfigurasi kedua adalah $k = 4$, yang menjadi titik elbow pada grafik WCSS. Meskipun nilai Silhouette Score pada konfigurasi ini tidak setinggi $k = 2$, namun $k = 4$ memberikan keseimbangan terbaik antara kedalaman segmentasi dan efisiensi model. Dengan empat klaster, perusahaan dapat merancang strategi yang lebih spesifik untuk setiap segmen

pelanggan, seperti membuat program loyalitas yang lebih terarah, menyusun rekomendasi produk berbasis pola pembelian, atau merancang promosi khusus berdasarkan daya beli dan tingkat kepuasan pelanggan di tiap kelompok. Oleh karena itu, pilihan jumlah klaster ideal sangat bergantung pada konteks bisnis dan tujuan akhir dari analisis segmentasi ini.

5.3 Analisis performa antar eksperimen

Berdasarkan hasil eksperimen yang telah dilakukan dengan variasi jumlah klaster dari $k = 2$ hingga $k = 7$, dapat dilihat adanya perbedaan signifikan dalam performa model dan karakter segmentasi yang terbentuk. Salah satu konfigurasi yang paling menonjol adalah $k = 2$, yang menghasilkan dua klaster besar dengan perbedaan karakteristik yang sangat jelas dan mudah dikenali secara visual. Nilai Silhouette Score tertinggi juga diperoleh pada konfigurasi ini, yang mengindikasikan bahwa pemisahan antar klaster sangat baik dan masing-masing klaster memiliki struktur internal yang solid. Segmentasi dengan dua klaster ini sangat cocok diterapkan dalam strategi makro-segmentasi, misalnya untuk membedakan pelanggan aktif dan pasif, atau pelanggan loyal dan pelanggan sesekali. Dengan dua kelompok besar, perusahaan dapat merancang strategi promosi yang langsung menyasar kebutuhan umum masing-masing kelompok tanpa memerlukan analisis lanjutan.

Sementara itu, pada konfigurasi $k = 4$, hasil segmentasi menjadi lebih mendalam dan kompleks. Klaster yang terbentuk menunjukkan keberagaman perilaku dan karakteristik pelanggan yang lebih variatif, seperti perbedaan dalam tingkat kepuasan, volume pembelian, atau preferensi saluran belanja (online/offline). Meskipun nilai Silhouette Score pada konfigurasi ini tidak setinggi pada $k = 2$, hasilnya tetap tergolong baik dan dianggap sebagai titik optimal berdasarkan metode elbow pada grafik WCSS. Segmentasi ini lebih cocok digunakan untuk strategi personalisasi yang spesifik, seperti penyusunan rekomendasi produk berdasarkan histori pembelian, pemberian promosi yang disesuaikan dengan perilaku pelanggan di setiap klaster, atau pengelolaan stok barang berdasarkan wilayah yang memiliki karakteristik pembeli berbeda-beda. Keunggulan dari konfigurasi ini adalah kemampuannya dalam menangkap nuansa perbedaan antar pelanggan, sehingga memungkinkan pengambilan keputusan bisnis yang lebih tepat sasaran.

Adapun pada konfigurasi k lebih dari 4, memang terjadi peningkatan detail dalam segmentasi, namun perbedaan nilai evaluasi, baik WCSS maupun Silhouette Score, tidak lagi signifikan. Hal ini menunjukkan bahwa penambahan jumlah klaster hanya memperumit model tanpa menghasilkan nilai tambah yang sebanding. Kompleksitas yang meningkat ini bisa menjadi hambatan dalam implementasi, karena memerlukan sistem analitik dan manajemen data yang lebih canggih untuk mengelola banyak segmen pelanggan. Dalam konteks bisnis yang tidak terlalu kompleks atau sumber daya yang terbatas, konfigurasi ini mungkin kurang efisien untuk diterapkan.

Secara keseluruhan, hasil eksperimen ini menunjukkan adanya trade-off yang jelas antara kesederhanaan model dan kedalaman segmentasi. Segmentasi dengan $k = 2$ menawarkan kemudahan implementasi dan kejelasan struktur klaster, sehingga cocok untuk strategi cepat dan umum. Sebaliknya, konfigurasi $k = 4$ atau lebih memberikan pemahaman yang lebih tajam

terhadap pelanggan, namun membutuhkan sistem yang lebih terintegrasi serta pendekatan bisnis yang lebih matang. Oleh karena itu, pemilihan nilai *k* yang digunakan dalam penerapan akhir harus mempertimbangkan tujuan bisnis, kapasitas sistem, serta kebutuhan akan detail segmentasi yang ingin dicapai oleh perusahaan.

BAB VI
KESIMPULAN

6.1 Kesimpulan

Berdasarkan seluruh rangkaian proses yang telah dilakukan dalam proyek tugas besar ini, dapat disimpulkan bahwa penerapan algoritma K-Means Clustering terbukti efektif dalam membentuk segmentasi pelanggan berdasarkan data penjualan retail. Proyek ini mencakup tahap eksplorasi, praproses data, pembangunan model dari awal tanpa library machine learning bawaan, hingga evaluasi dan eksperimen model dengan variasi jumlah klaster. Proses praproses seperti konversi data kategorikal menjadi numerik dan normalisasi Min-Max Scaling berhasil menyamakan skala antar fitur, sehingga mendukung akurasi penghitungan jarak pada algoritma K-Means. Evaluasi model menggunakan metrik Silhouette Score dan WCSS menunjukkan bahwa konfigurasi klaster dengan nilai k = 2 menghasilkan pemisahan kelompok yang paling bersih dan kompak, dengan nilai Silhouette tertinggi sebesar 0.7316. Sementara itu, konfigurasi k = 4 diidentifikasi sebagai titik elbow, yang memberikan keseimbangan optimal antara kedalaman segmentasi dan kompleksitas model. Hasil eksperimen juga menunjukkan bahwa peningkatan jumlah klaster di atas 4 tidak memberikan peningkatan signifikan terhadap performa model, namun justru dapat menambah kompleksitas dalam pengambilan keputusan bisnis.

Implementasi model secara manual (coding from scratch) memberikan pemahaman yang lebih dalam terhadap prinsip kerja K-Means, mulai dari inisialisasi centroid, penghitungan jarak Euclidean, hingga proses iteratif pembaruan klaster hingga konvergen. Dari sisi bisnis, hasil segmentasi ini memiliki manfaat nyata, antara lain dalam mendukung perancangan strategi pemasaran yang lebih personal, mengenali pelanggan bernilai tinggi, dan mengoptimalkan alokasi sumber daya berdasarkan perilaku pelanggan dan wilayah penjualan. Segmentasi dua klaster cocok untuk strategi sederhana dan luas, sementara empat klaster lebih relevan untuk strategi personalisasi yang lebih tajam. Dari sisi akademik, proyek ini berhasil memperkuat kompetensi mahasiswa dalam menerapkan konsep unsupervised learning, eksplorasi data, pemrograman algoritma, serta mengevaluasi model berbasis data. Dengan demikian, keseluruhan proses dan hasil dari tugas besar ini tidak hanya memberikan solusi berbasis data untuk kasus bisnis, tetapi juga memperkaya pemahaman konseptual dan praktikal dalam bidang Pembelajaran Mesin dan Sains Data.

LAMPIRAN

1. LINK VIDEO YOUTUBE

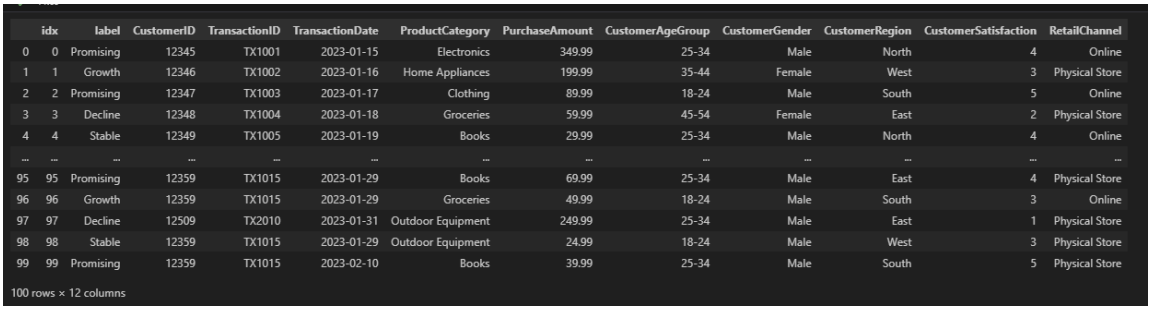
[LINK VIDEO PRESENTASI (YOUTUBE)]

2. LINK GITHUB REPOSITORY

[LINK GITHUB REPOSITORY KELOMPOK AQUAVERA]

3. SCREENSHOOT DATASET

- Data Retail-Sales-Analyzer.csv



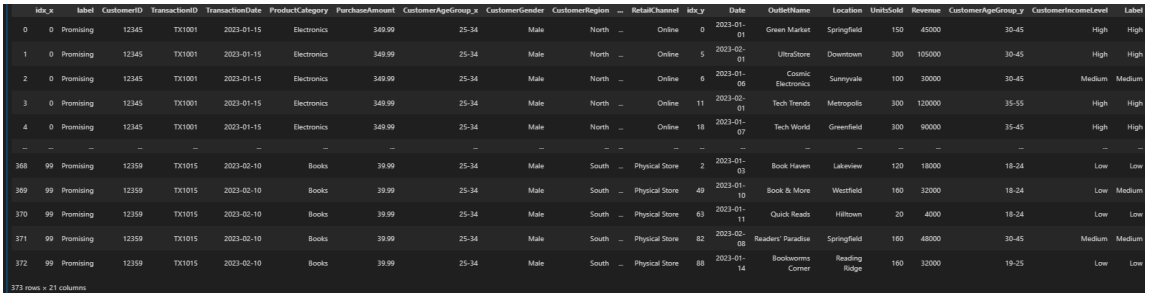
	idx	label	CustomerID	TransactionID	TransactionDate	ProductCategory	PurchaseAmount	CustomerAgeGroup	CustomerGender	CustomerRegion	CustomerSatisfaction	RetailChannel
0	0	Promising	12345	TX1001	2023-01-15	Electronics	349.99	25-34	Male	North	4	Online
1	1	Growth	12346	TX1002	2023-01-16	Home Appliances	199.99	35-44	Female	West	3	Physical Store
2	2	Promising	12347	TX1003	2023-01-17	Clothing	89.99	18-24	Male	South	5	Online
3	3	Decline	12348	TX1004	2023-01-18	Groceries	59.99	45-54	Female	East	2	Physical Store
4	4	Stable	12349	TX1005	2023-01-19	Books	29.99	25-34	Male	North	4	Online
...	...	...	...	...	...	...	...	...	...	...	...	...
95	95	Promising	12359	TX1015	2023-01-29	Books	69.99	25-34	Male	East	4	Physical Store
96	96	Growth	12359	TX1015	2023-01-29	Groceries	49.99	18-24	Male	South	3	Online
97	97	Decline	12509	TX2010	2023-01-31	Outdoor Equipment	249.99	25-34	Male	East	1	Physical Store
98	98	Stable	12359	TX1015	2023-01-29	Outdoor Equipment	24.99	18-24	Male	West	3	Physical Store
99	99	Promising	12359	TX1015	2023-02-10	Books	39.99	25-34	Male	South	5	Physical Store

100 rows × 12 columns

- Data Retail-Sales-Performance.csv

	idx	Date	OutletName	Location	ProductCategory	UnitsSold	Revenue	CustomerAgeGroup	CustomerIncomeLevel	Label
0	0	2023-01-01	Green Market	Springfield	Electronics	150	45000	30-45	High	High
1	1	2023-01-02	City Mall	Springfield	Home & Garden	200	50000	25-35	Medium	Medium
2	2	2023-01-03	Book Haven	Lakeview	Books	120	18000	18-24	Low	Low
3	3	2023-01-04	FashionFiesta	River City	Apparel	250	75000	40-50	High	High
4	4	2023-01-05	Food World	Springfield	Food & Beverages	300	60000	20-30	Medium	Medium
...	...	...	...	...	...	...	...	...	...	...
95	95	2023-02-10	Gastronomy Gallery	Seaside	Food & Beverages	600	180000	20-30	High	High
96	96	2023-01-15	Bakery Bites	Springfield	Food & Beverages	130	39000	20-30	Medium	Medium
97	97	2023-02-23	Junior Joy	Playland	Toys	5	1000	5-10	Low	Low
98	98	2023-01-15	Soda Pop Emporium	Soda Town	Food & Beverages	310	62000	22-32	High	High
99	99	2023-01-15	Tech Trends	Capital City	Electronics	110	33000	20-30	High	Medium

- Data Retail-Sales-Merged.xlsx



	idx_x	label	CustomerID	TransactionID	TransactionDate	ProductCategory	PurchaseAmount	CustomerAgeGroup_x	CustomerGender	CustomerRegion	...	RetailChannel	idx_y	Date	OutletName	Location	UnitsSold	Revenue	CustomerAgeGroup_y	CustomerIncomeLevel	Label
0	0	Promising	12345	TX1001	2023-01-15	Electronics	349.99	25-34	Male	North	...	Online	0	2023-01-01	Green Market	Springfield	150	45000	30-45	High	High
1	0	Promising	12345	TX1001	2023-01-15	Electronics	349.99	25-34	Male	North	...	Online	5	2023-02-01	UltraStore	Downtown	300	105000	30-45	High	High
2	0	Promising	12345	TX1001	2023-01-15	Electronics	349.99	25-34	Male	North	...	Online	6	2023-01-06	Cosmic Electronics	Sunnyvale	100	30000	30-45	Medium	Medium
3	0	Promising	12345	TX1001	2023-01-15	Electronics	349.99	25-34	Male	North	...	Online	11	2023-02-01	Tech Trends	Metropolis	300	120000	35-55	High	High
4	0	Promising	12345	TX1001	2023-01-15	Electronics	349.99	25-34	Male	North	...	Online	18	2023-01-07	Tech World	Greenfield	300	90000	35-45	High	High
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
368	99	Promising	12359	TX1015	2023-02-10	Books	39.99	25-34	Male	South	...	Physical Store	2	2023-01-03	Book Haven	Lakeview	120	18000	18-24	Low	Low
369	99	Promising	12359	TX1015	2023-02-10	Books	39.99	25-34	Male	South	...	Physical Store	49	2023-01-10	Book & More	Westfield	160	32000	18-24	Low	Medium
370	99	Promising	12359	TX1015	2023-02-10	Books	39.99	25-34	Male	South	...	Physical Store	63	2023-01-11	Quick Reads	Hilltown	20	4000	18-24	Low	Low
371	99	Promising	12359	TX1015	2023-02-10	Books	39.99	25-34	Male	South	...	Physical Store	82	2023-02-08	Readers' Paradise	Springfield	160	48000	30-45	Medium	Medium
372	99	Promising	12359	TX1015	2023-02-10	Books	39.99	25-34	Male	South	...	Physical Store	88	2023-01-14	Bookworms Corner	Reading Ridge	160	32000	19-25	Low	Low

373 rows × 21 columns

4. SOURCE CODE

- Import Dataset dan Penggabungan Data

```
import pandas as pd
import urllib.request

file_id = '1B_Z_WlLdvxR-NRTH3piCzVVTOLOwY5Lp'
url = f'https://drive.google.com/uc?export=download&id=
{file_id}'
output_path1 = 'Retail-Sales-Analyzer.csv'
urllib.request.urlretrieve(url, output_path1)
df1 = pd.read_csv(output_path1)
df1
```

```
import pandas as pd
import urllib.request

file_id = '1WmzaQMmUiVcxeYgM9ekSQi-insHnrHHz'
url = f'https://drive.google.com/uc?export=download&id=
{file_id}'
output_path2 = 'Retail-Sales-Performance.csv'

urllib.request.urlretrieve(url, output_path2)
df2 = pd.read_csv(output_path2)
df2
```

```
import pandas as pd
import urllib.request

ID = '1R8tZtts5_XVXpL3rmtPhQrRCNhOl1itS'
url =
f'https://docs.google.com/spreadsheets/d/{ID}/export?
format=xlsx'
output_path = 'Retail-Sales-Merged.xlsx'
urllib.request.urlretrieve(url, output_path)
df = pd.read_excel(output_path, engine='openpyxl')
df
```

- Praproses Data

```python
import pandas as pd

# Baca data
df = pd.read_excel("Retail-Sales-Merged.xlsx")

# Drop kolom yang tidak relevan untuk segmentasi pelanggan
df = df.drop(columns=[
    'idx_x', 'idx_y', 'label', 'Label',
    'TransactionID', 'TransactionDate',
    'Date', 'OutletName', 'Location',
    'UnitsSold', 'Revenue'
])

# Hilangkan duplikat baris
df = df.drop_duplicates()

# Rename kolom agar seragam
df = df.rename(columns={
    'CustomerAgeGroup_x': 'CustomerAgeGroup',
    'CustomerGender': 'Gender',
    'CustomerSatisfaction': 'Satisfaction',
    'RetailChannel': 'Channel',
    'CustomerIncomeLevel': 'Income'
})

# Map nilai kategorikal ke numerik
df['Gender'] = df['Gender'].map({'Male': 0, 'Female': 1})

# Retail Channel: Online = 0, Offline = 1
df['Channel'] = df['Channel'].map({'Online': 0, 'Offline':
1})

# Income Level: Low = 0, Medium = 1, High = 2
df['Income'] = df['Income'].map({'Low': 0, 'Medium': 1,
'High': 2})

# Map CustomerAgeGroup (urutan tergantung rentang
sebenarnya)
age_mapping = {
    '<25': 0,
    '25-34': 1,
    '30-45': 2,
    '35-45': 3,
    '35-55': 4,
    '45-60': 5,
    '>60': 6
}
df['CustomerAgeGroup'] =
df['CustomerAgeGroup'].map(age_mapping)

# Buang baris yang masih ada nilai kosong
df = df.dropna()

# Reset index
df = df.reset_index(drop=True)

# Ambil hanya fitur yang akan digunakan untuk clustering
X = df[['PurchaseAmount', 'Satisfaction',
'CustomerAgeGroup', 'Gender', 'Channel', 'Income']]

# Normalisasi data (opsional, tergantung nanti pada
perhitungan K-Means manual)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)

# Konversi ke DataFrame lagi (jika mau)
X_preprocessed = pd.DataFrame(X_scaled, columns=X.columns)

# Tampilkan hasil akhir siap pakai untuk clustering
print(X_preprocessed.head())
```

- Normalisasi

```
def manual_min_max_normalize(df):
    df_norm = df.copy()
    for column in df.columns:
        min_val = df[column].min()
        max_val = df[column].max()
        df_norm[column] = (df[column] - min_val) /
(max_val - min_val)
    return df_norm
```

- K-means Clustering From Scratch

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Fungsi menghitung jarak Euclidean
def euclidean_distance(a, b):
    return np.sqrt(np.sum((a - b) ** 2))

# Fungsi K-Means tanpa sklearn
def kmeans_from_scratch(X, k=3, max_iter=100, tol=1e-4,
seed=42):
    np.random.seed(seed)
    n_samples, n_features = X.shape

    # Inisialisasi centroid secara acak dari data
    initial_idx = np.random.choice(n_samples, size=k,
replace=False)
    centroids = X[initial_idx]

    for iteration in range(max_iter):
        labels = []

        # Assignment step
        for x in X:
            distances = [euclidean_distance(x, centroid)
for centroid in centroids]
            labels.append(np.argmin(distances))
        labels = np.array(labels)

        # Update centroids
        new_centroids = np.zeros_like(centroids)
        for i in range(k):
            points = X[labels == i]
            if len(points) > 0:
                new_centroids[i] = np.mean(points, axis=0)
            else:
                new_centroids[i] =
X[np.random.choice(n_samples)]

        # Cek konvergensi
        shift = np.linalg.norm(new_centroids - centroids)
        if shift < tol:
            break
        centroids = new_centroids

    return labels, centroids, iteration + 1
```

- Evaluasi WCSS & Silhouette Score

```
def compute_wcss(X, labels, centroids):
    wcss = 0.0
    for i in range(len(centroids)):
        cluster_points = X[labels == i]
        for point in cluster_points:
            wcss += np.sum((point - centroids[i]) ** 2)
    return wcss

def silhouette_score_manual(X, labels):
    n = len(X)
    k = np.max(labels) + 1
    scores = []

    for i in range(n):
        xi = X[i]
        label_i = labels[i]

        same_cluster = X[labels == label_i]
        a = np.mean([euclidean_distance(xi, xj) for xj in
same_cluster if not np.allclose(xi, xj)])

        b = np.inf
        for j in range(k):
            if j != label_i:
                other_cluster = X[labels == j]
                dist = np.mean([euclidean_distance(xi, xj)
for xj in other_cluster])
                b = min(b, dist)

        s = (b - a) / max(a, b)
        scores.append(s)

    return np.mean(scores)
```

- Clustering (PCA 2D)

```
# PCA manual
def manual_pca_2d(X):
    X_centered = X - np.mean(X, axis=0)
    cov = np.cov(X_centered, rowvar=False)
    eig_vals, eig_vecs = np.linalg.eigh(cov)
    idx = np.argsort(eig_vals)[::-1]
    top2 = eig_vecs[:, idx[:2]]
    return X_centered @ top2, top2
```

- Evaluasi

```
k = 3
labels, centroids, n_iter = kmeans_from_scratch(X, k=k)
wcss_value = compute_wcss(X, labels, centroids)
silhouette_value = silhouette_score_manual(X, labels)

print(f"📌 Evaluasi Clustering untuk k = {k}")
print(f"- Iterasi: {n_iter}")
print(f"- WCSS: {wcss_value:.4f}")
print(f"- Silhouette Score: {silhouette_value:.4f}")

X_2d, pca_basis = manual_pca_2d(X)
centroids_2d = (centroids - np.mean(X, axis=0)) @
pca_basis

plt.figure(figsize=(8, 6))
for i in range(k):
    plt.scatter(X_2d[labels == i, 0], X_2d[labels == i,
1], label=f"Cluster {i}")
plt.scatter(centroids_2d[:, 0], centroids_2d[:, 1],
marker='x', c='black', s=200, linewidths=3,
label='Centroid')
plt.title(f"Hasil Clustering K-Means (k = {k}) - PCA 2D")
plt.xlabel("PC 1")
plt.ylabel("PC 2")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

- Hasil Perbandingan

```
k_values = [2, 3, 4, 5, 6, 7]

wcss_list       = []
silhouette_list = []

for k in k_values:
    labels, centroids, _ = kmeans_from_scratch(X, k=k)
    wcss       = compute_wcss(X, labels, centroids)
    silhouette = silhouette_score_manual(X, labels)

    wcss_list.append(wcss)
    silhouette_list.append(silhouette)

    print(f"k={k} | WCSS={wcss:.2f} | Silhouette=
{silhouette:.4f}")

if len(wcss_list) >= 2:
    diff = np.diff(wcss_list)
    if len(diff) >= 2:
        elbow_idx = np.argmax(np.abs(diff[:-1] -
diff[1:])) + 1
        k_elbow = k_values[elbow_idx]
        print(f"\nRekomendasi terbaik (Elbow): {k_elbow}")
    else:
        print("\nTidak cukup data untuk menghitung Elbow
heuristik.")
else:
    print("\nTidak cukup data untuk menghitung Elbow
heuristik.")

if silhouette_list:
    k_sil = k_values[np.argmax(silhouette_list)]
    print(f"Rekomendasi terbaik (Silhouette max):
{k_sil}")
else:
    print("Tidak ada data Silhouette Score untuk
menentukan rekomendasi.")

if k_values and wcss_list and silhouette_list:
    fig, ax1 = plt.subplots(figsize=(8,5))

    ax1.plot(k_values, wcss_list, marker='o',
label='WCSS')
    ax1.set_xlabel('Jumlah Cluster (k)')
    ax1.set_ylabel('WCSS')
    ax1.grid(True)

    ax2 = ax1.twinx()
    ax2.plot(k_values, silhouette_list, marker='s',
color='orange', label='Silhouette')
    ax2.set_ylabel('Silhouette Score')

    if 'k_elbow' in locals():
        ax1.scatter(k_elbow,
wcss_list[k_values.index(k_elbow)],
                    color='red', s=120, zorder=5,
label='Elbow')
    if 'k_sil' in locals():
        ax2.scatter(k_sil,
silhouette_list[k_values.index(k_sil)],
                    color='green', s=120, zorder=5,
label='Silhouette max')

    lines, labels = [], []
    for ax in [ax1, ax2]:
        L, lab = ax.get_legend_handles_labels()
        lines.extend(L); labels.extend(lab)
    ax1.legend(lines, labels, loc='best')

    plt.title('Perbandingan WCSS & Silhouette (k = 2-6)')
    plt.tight_layout()
    plt.show()
else:
    print("Tidak cukup data untuk membuat plot.")
```

DAFTAR PUSTAKA

Alamsyah, A., Prasetyo, P. E., Sunyoto, S., & Bintari, S. H. (2022). Customer Segmentation Using the Integration of the Recency Frequency Monetary Model and the K-Means Cluster Algorithm. Scientific Journal of Informatics, 189-196.

Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques. Boston: Elsevier.

Harish, A. S., & Malathy, C. (2023). Customer segment prediction on retail transactional data using K-Means and Markov model. Intelligent Automation & Soft Computing, 589-600.

Kaufman, L., & Rousseeuw, P. J. (2009). Finding Groups in Data: An Introduction to Cluster Analysis. Hoboken: Wiley.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, p. 281 - 297 .

Raschka, S., & Mirjalili, V. (2019). Python Machine Learning . Birmingham : Packt Publishing .

Siagian, R., Sirait, P., & Halima, A. (2021). E-Commerce Customer Segmentation Using K-Means Algorithm and LRFM Model. Jurnal Teknologi Informasi dan Ilmu Komputer, 247-252.

Syakur, M. A., Khotimah, B. K., Rochman, E. M., & Satoto, B. D. (2018). Integration K-Means clustering method and elbow method for identification of the best customer profile cluster. IOP Conference Series: Materials Science and Engineering, 012017.

Tan, P.-N., Steinbach, M., & Kumar, V. (2019). Introduction to Data Mining . Upple Saddle River : Pearson.