

**LAPORAN TUGAS BESAR PEMBELAJARAN MESIN**  
***CLUSTERING***



**DISUSUN OLEH KELOMPOK 3**

- |                                |                     |
|--------------------------------|---------------------|
| <b>1. AURA AULIA A.H.P</b>     | <b>(1206230003)</b> |
| <b>2. JONATHAN MANGIRING P</b> | <b>(1206230051)</b> |
| <b>3. NOVITA CHELSEA LODAR</b> | <b>(1206230052)</b> |

## DAFTAR ISI

DAFTAR ISI.....	2
1. FORMULASI MASALAH.....	3
2. EKSPLORASI DAN PERSIAPAN DATA.....	2
2.1 Sumber Data.....	2
2.2. Deskripsi Dataset.....	2
2.3. Import Library.....	4
2.4. Load Dataset.....	4
2.5. Ukuran dan Struktur Data.....	4
2.6. Pengecekan Missing Values.....	5
2.7. Penanganan Missing Values.....	5
2.8. Pengecekan Duplikasi.....	6
2.9. Visualisasi Korelasi.....	6
2.10. Feature Engineering.....	6
2.11. Seleksi Fitur untuk Clustering.....	7
3. PEMODELAN (CLUSTERING).....	8
4. EVALUASI.....	9
5. EKSPERIMEN.....	10
6. KESIMPULAN.....	15
LAMPIRAN.....	16

## 1. FORMULASI MASALAH

Di tengah persaingan bisnis yang semakin ketat, perusahaan bukan hanya melakukan pengembangan terkait produk, namun juga memahami pelanggan. Salah satu kegunaan dalam memahami pelanggan, supaya perusahaan dapat menetapkan strategi pemasaran yang tepat sasaran, yang bukan hanya menarik pelanggan, namun juga mempertahankan kuantitas dan kualitas pelanggan.

Namun, data pelanggan cukup bervariasi dan tidak langsung dapat diaplikasikan untuk penetapan strategi yang tepat sasaran. Hal ini menjadi tantangan tersendiri bagi perusahaan dalam menyesuaikan produk atau layanan dengan kebutuhan masing-masing pelanggan. Oleh karena itu, diperlukan segmentasi pelanggan guna mencocokkan pendekatan yang tepat bagi setiap kelompok pelanggan. Salah satu metode yang dapat digunakan untuk tujuan ini adalah *clustering*.

Segmentasi pelanggan penting dilakukan mengingat perbedaan pada setiap pelanggan, hal ini juga didukung dengan data pelanggan yang cukup bervariasi sehingga menyulitkan dalam hal interpretasi secara langsung. Dengan mengelompokkan pelanggan berdasarkan kesamaan karakteristik, perusahaan dapat lebih mudah menyusun strategi pemasaran, pelayanan, dan penawaran produk yang sesuai untuk tiap kelompok. Ini tidak hanya membuat proses pemasaran menjadi lebih efisien, tetapi juga dapat meningkatkan loyalitas dan kepuasan pelanggan.

Melalui penerapan *clustering* menggunakan dataset *Customer Personality Analysis*, tujuan yang ingin dicapai adalah mengelompokkan pelanggan ke dalam klaster berdasarkan kemiripan perilaku dan karakteristik. Dengan informasi ini, perusahaan dapat lebih memahami siapa pelanggan mereka dan menyesuaikan produk atau layanan sesuai dengan kebutuhan tiap klaster. Hal ini diharapkan dapat menunjang pengambilan keputusan bisnis yang lebih tepat dan berfokus pada pelanggan.

## 2. EKSPLORASI DAN PERSIAPAN DATA

### 2.1 Sumber Data

Dataset yang digunakan dalam proyek ini adalah *Marketing Campaign Dataset* yang tersedia di platform Kaggle. Dataset ini berisi informasi tentang pelanggan dari sebuah perusahaan ritel, termasuk data demografis, kebiasaan pembelian, serta tanggapan terhadap kampanye pemasaran.

Link dataset:

<https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis>

### 2.2. Deskripsi Dataset

Dataset ini terdiri dari 2240 baris (pelanggan) dan 29 kolom (fitur). Berikut adalah deskripsi masing-masing fitur dalam dataset:

Fitur	Deskripsi
ID	ID unik untuk setiap pelanggan
Year_Birth	Tahun kelahiran pelanggan
Education	Tingkat pendidikan pelanggan
Marital_Status	Status pernikahan pelanggan
Income	Pendapatan tahunan rumah tangga pelanggan
Kidhome	Jumlah anak kecil di rumah pelanggan
Teenhome	Jumlah remaja di rumah pelanggan
Dt_Customer	Tanggal pertama kali pelanggan bergabung
Recency	Jumlah hari sejak pembelian terakhir
Complain	1 jika pelanggan pernah komplain dalam 2 tahun terakhir, 0 jika tidak
MntWines	Total pengeluaran untuk anggur dalam 2 tahun terakhir

MntFruits	Total pengeluaran untuk buah dalam 2 tahun terakhir
MntMeatProducts	Total pengeluaran untuk daging dalam 2 tahun terakhir
MntFishProducts	Total pengeluaran untuk ikan dalam 2 tahun terakhir
MntSweetProducts	Total pengeluaran untuk makanan manis dalam 2 tahun terakhir
MntGoldProds	Total pengeluaran untuk produk emas dalam 2 tahun terakhir
NumDealsPurchases	Jumlah pembelian dengan diskon
AcceptedCmp1	1 jika pelanggan menerima tawaran pada kampanye ke-1, 0 jika tidak
AcceptedCmp2	1 jika pelanggan menerima tawaran pada kampanye ke-2, 0 jika tidak
AcceptedCmp3	1 jika pelanggan menerima tawaran pada kampanye ke-3, 0 jika tidak
AcceptedCmp4	1 jika pelanggan menerima tawaran pada kampanye ke-4, 0 jika tidak
AcceptedCmp5	1 jika pelanggan menerima tawaran pada kampanye ke-5, 0 jika tidak
Response	1 jika pelanggan menerima tawaran pada kampanye terakhir, 0 jika tidak
NumWebPurchases	Jumlah pembelian yang dilakukan melalui situs web perusahaan
NumCatalogPurchases	Jumlah pembelian yang dilakukan menggunakan katalog
NumStorePurchases	Jumlah pembelian yang dilakukan secara langsung di toko fisik
NumWebVisitsMonth	Jumlah kunjungan ke situs web perusahaan dalam satu bulan terakhir
Z_CostContact	-

Z_Revenue	-
-----------	---

### 2.3. Import Library

```
[1] import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Gambar 1. *Import Library*

- numpy untuk perhitungan numerik dasar.
- pandas untuk membaca dan memproses data tabular.
- seaborn dan matplotlib.pyplot untuk membuat visualisasi eksploratif.

### 2.4. Load Dataset

```
df = pd.read_csv('/content/marketing_campaign.csv', sep=';')
df
```

Gambar 2. *Load Dataset*

Dataset dibaca menggunakan fungsi `read_csv()` dari pustaka pandas dengan parameter `sep=';'`. Hal ini dilakukan karena file menggunakan titik koma sebagai pemisah kolom.

### 2.5. Ukuran dan Struktur Data

Dataset memiliki 2240 baris dan 29 kolom. Beberapa kolom bertipe numerik (`int64`, `float64`), dan lainnya bertipe kategorikal (`object`). Terdapat *missing value*, khususnya pada kolom *Income*.

```
[3] df.shape
```

```
(2240, 29)
```

Gambar 3. Ukuran Data

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 2240 entries, 0 to 2239
```

```
Data columns (total 29 columns):
```

#	Column	Non-Null Count	Dtype
0	ID	2240 non-null	int64
1	Year_Birth	2240 non-null	int64
2	Education	2240 non-null	object
3	Marital_Status	2240 non-null	object
4	Income	2216 non-null	float64
5	Kidhome	2240 non-null	int64
6	Teenhome	2240 non-null	int64
7	Dt_Customer	2240 non-null	object
8	Recency	2240 non-null	int64
9	MntWines	2240 non-null	int64
10	MntFruits	2240 non-null	int64
11	MntMeatProducts	2240 non-null	int64
12	MntFishProducts	2240 non-null	int64
13	MntSweetProducts	2240 non-null	int64
14	MntGoldProds	2240 non-null	int64
15	NumDealsPurchases	2240 non-null	int64
16	NumWebPurchases	2240 non-null	int64
17	NumCatalogPurchases	2240 non-null	int64
18	NumStorePurchases	2240 non-null	int64
19	NumWebVisitsMonth	2240 non-null	int64
20	AcceptedCmp3	2240 non-null	int64
21	AcceptedCmp4	2240 non-null	int64
22	AcceptedCmp5	2240 non-null	int64
23	AcceptedCmp1	2240 non-null	int64
24	AcceptedCmp2	2240 non-null	int64
25	Complain	2240 non-null	int64
26	Z_CostContact	2240 non-null	int64
27	Z_Revenue	2240 non-null	int64
28	Response	2240 non-null	int64

```
dtypes: float64(1), int64(25), object(3)
```

```
memory usage: 507.6+ KB
```

Gambar 4. Tipe Data

Dari 29 Fitur, terdapat 3 fitur dengan tipe data object, 1 tipe data float, dan 25 int64

## 2.6. Pengecekan *Missing Values*

Untuk mengetahui jumlah nilai yang hilang (missing values), dilakukan pengecekan sebagai berikut:

```
# Mengecek Jumlah nilai Kosong
```

```
df.isnull().sum()
```

Gambar 5. *Missing Values*

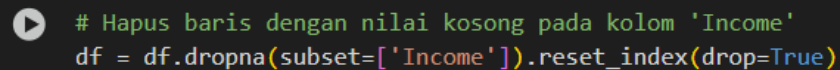
## 2.7. Penanganan *Missing Values*

Setelah dilakukan pengecekan *Missing Values*, ditemukan bahwa hanya ada 24 baris yang memiliki missing values, semuanya berada pada kolom Income. Kolom ini penting karena mencerminkan kondisi ekonomi pelanggan, sehingga tetap perlu disertakan dalam analisis. Namun karena jumlah missing values yang ada hanya 24 dari total 2240 baris, maka secara proporsional bisa dikatakan kecil.

Perhitungan persentase *missing values*:

$$\frac{24}{2240} \times 100 = 1.07\%$$

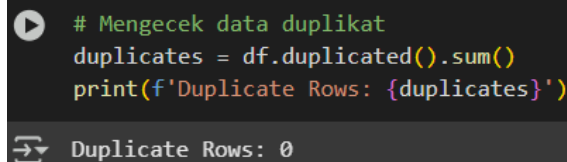
Nilai 1.07% ini berada toleransi umum 5% sehingga data tersebut masih dapat dihapus tanpa risiko kehilangan informasi yang signifikan.



```
# Hapus baris dengan nilai kosong pada kolom 'Income'
df = df.dropna(subset=['Income']).reset_index(drop=True)
```

Gambar 6. Penanganan *Missing Values*

## 2.8. Pengecekan Duplikasi



```
# Mengecek data duplikat
duplicates = df.duplicated().sum()
print(f'Duplicate Rows: {duplicates}')
```

Duplicate Rows: 0

Gambar 7. Pengecekan Duplikasi Data

Hasil menunjukkan bahwa tidak terdapat baris duplikat dalam dataset.

## 2.9. Visualisasi Korelasi

Visualisasi korelasi antar fitur numerik dilakukan menggunakan heatmap yang membantu memahami hubungan antar variabel untuk keperluan seleksi fitur dan pemodelan.



```

num_cols = df.select_dtypes(include='number') # ambil semua kolom numerik
corr_matrix = num_cols.corr()
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Matrix of Numerical Features')
plt.show()

```

Gambar 8. Korelasi

## 2.10. Feature Engineering

Untuk memudahkan analisis segmentasi pelanggan, dibuat fitur baru bernama Age, yaitu usia pelanggan saat tahun data dikumpulkan (diasumsikan 2021):

```

[9] df['Age'] = 2021 - df['Year_Birth']

```

Gambar 9. Feature Engineering

## 2.11. Seleksi Fitur untuk Clustering

Fitur-fitur numerik berikut dipilih sebagai input untuk model clustering karena menggambarkan karakteristik pelanggan yang relevan:

```

[10] features = [
    'Age', 'Income', 'Kidhome', 'Teenhome', 'Recency',
    'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts',
    'MntSweetProducts', 'MntGoldProds',
    'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases',
    'NumStorePurchases', 'NumWebVisitsMonth',
    'AcceptedCmp1', 'AcceptedCmp2', 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5',
    'Complain', 'Response']

```

Gambar 10. Feature Selection

### 3. PEMODELAN (*CLUSTERING*)

#### 3.1 Metode Clustering yang Dibangun: K-Means dari Nol

Metode clustering yang digunakan dalam proyek ini adalah K-Means, dan diimplementasikan dari nol tanpa menggunakan library pemodelan seperti scikit-learn. Tujuan dari penggunaan metode ini adalah untuk membagi data pelanggan ke dalam beberapa kelompok berdasarkan kemiripan karakteristik numerik mereka.

K-Means dipilih karena merupakan salah satu metode clustering yang paling sederhana dan efektif, terutama untuk data yang sudah terstandarisasi dan memiliki fitur numerik seperti dalam dataset ini.

#### 3.2 Logika Algoritma dan Parameter

Algoritma K-Means terdiri dari beberapa langkah utama:

- Inisialisasi centroid secara acak dari data.
- Assign cluster: setiap titik data dihubungkan ke centroid terdekat berdasarkan jarak Euclidean.
- Update centroid: setiap centroid diperbarui dengan mengambil rata-rata titik dalam cluster-nya.
- Iterasi berlanjut hingga centroid stabil (konvergen) atau mencapai iterasi maksimum.
- 

Parameter yang digunakan dalam implementasi:

- `k`: jumlah cluster yang diinginkan (dalam kasus ini: 4).
- `max_iter`: batas maksimum iterasi (default: 100).
- `np.random.seed(42)`: untuk replikasi hasil inisialisasi acak.

#### 3.3 Proses Training dan Pengujian Model

Training dilakukan dengan fungsi `kmeans()` yang menerima data yang sudah dinormalisasi dan menghasilkan label cluster serta posisi centroid. Proses ini diulang hingga konvergen, yaitu ketika centroid tidak berubah signifikan dari iterasi sebelumnya.

Pengujian dalam clustering tidak seperti supervised learning. Karena tidak ada "label benar", maka evaluasi dilakukan berdasarkan seberapa baik data terkelompok dan separasi antar cluster menggunakan metrik internal seperti Silhouette Score.

### 3.4 Penjelasan Kode (Tanpa Library Pemodelan)

#### 1. Standardisasi Fitur

```
def standardize(X):  
    mean = np.mean(X, axis=0)  
    std = np.std(X, axis=0)  
    return (X - mean) / std  
  
X_std = standardize(X)
```

Gambar 3.4.1 Standardisasi Fitur

#### 2. Jarak euclidean

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

Gambar 3.4.2 Jarak euclidean

#### 3. Kmean clustering

```
def kmeans(data, k, max_iter=100):  
    # Step 1: Inisialisasi centroid secara acak  
    np.random.seed(42)  
    centroids = data[np.random.choice(len(data), k, replace=False)]  
  
    for _ in range(max_iter):  
        # Step 2: Assign cluster  
        clusters = [[] for _ in range(k)]  
        for point in data:  
            distances = [euclidean(point, centroid) for centroid in centroids]  
            cluster_idx = np.argmin(distances)  
            clusters[cluster_idx].append(point)  
  
        # Step 3: Update centroid  
        new_centroids = []  
        for cluster in clusters:  
            if cluster: # hindari cluster kosong  
                new_centroid = np.mean(cluster, axis=0)  
            else:  
                # kalau cluster kosong, random lagi centroid  
                new_centroid = data[np.random.choice(len(data))]  
            new_centroids.append(new_centroid)  
  
        new_centroids = np.array(new_centroids)  
  
        # Cek konvergen  
        if np.allclose(centroids, new_centroids):  
            break  
        centroids = new_centroids  
  
    # Assign label akhir  
    final_labels = []  
    for point in data:  
        distances = [euclidean(point, centroid) for centroid in centroids]  
        final_labels.append(np.argmin(distances))  
  
    return final_labels, centroids
```

Gambar 3.4.2 Jarak euclidean

#### 4. Labeling dan interpretasi

```
k = 4
labels, centroids = kmeans(X_std, k)
df['Cluster'] = labels
```

✓ 7.4s

Gambar 4.1. Inisialisasi Centroid

Untuk interpretasi, centroid dikonversi kembali ke nilai asli (skala sebelum standarisasi) :

```
# 5. Dapatkan kembali nilai asli dari centroid
mean = np.mean(X, axis=0)
std = np.std(X, axis=0)
centroids_original = centroids * std + mean
```

Gambar 4.2. Labeling dan interpretasi

## 4. EVALUASI

### 4.1 Metode Evaluasi yang Digunakan

- Evaluasi dilakukan menggunakan dua metrik utama:  
Sum of Squared Errors (SSE): Mengukur kompaknya cluster. Semakin kecil SSE, semakin baik hasil clustering.
- Silhouette Score: Mengukur seberapa mirip objek dengan klasternya sendiri dibandingkan dengan klaster lain (nilai -1 hingga 1). Semakin tinggi nilainya, semakin baik separasi klaster.

### 4.2 Justifikasi Pemilihan Metode Evaluasi

Kedua metrik ini umum digunakan dalam evaluasi model clustering:

- SSE mendukung pemilihan jumlah klaster optimal dengan metode elbow.
- Silhouette Score mengukur kualitas cluster tanpa memerlukan label ground truth, cocok untuk data unsupervised.

### 4.3 Hasil Evaluasi dan Interpretasi

Eksperimen dilakukan dengan variasi nilai **K** dari 2 hingga 6, menghasilkan:

Nilai K	SSE	Silhouette score
2	39670.17	0.2668
3	36105.84	0.1906
4	33561.31	0.1950
5	32163.60	0.1213
6	31112.06	0.1261

Dari tabel di atas, dipilih **K = 4** karena memberikan **kompromi terbaik** antara nilai SSE yang relatif rendah dan Silhouette Score yang cukup stabil.

## 5. EKSPERIMEN

Sebagai bagian dari proses pemodelan, dilakukan beberapa eksperimen untuk menentukan parameter terbaik dalam metode clustering, serta membandingkan hasil dari pendekatan yang berbeda. Eksperimen ini mencakup pemilihan nilai  $K$  terbaik, pemilihan subset fitur, serta reduksi dimensi menggunakan PCA.

### a. Pemilihan nilai $K$ terbaik

Pemilihan nilai  $K$  dilakukan dengan menghitung Sum of Square Error (SSE) dan Silhouette Score. Dengan interpretasi semakin kecil nilai SSE, semakin baik pemodelan kluster karena menunjukkan data dalam kluster lebih baik, sedangkan Silhouette Score dilakukan untuk mengukur seberapa mirip objek dengan kluster mereka sendiri dibandingkan dengan kluster lain. Semakin tinggi nilainya, semakin baik pemisahan antar kluster. Perhitungan SSE dan Silhouette Score direpresentasikan secara manual pada code dengan menggunakan rumus sebagai berikut :

$$SSE = \sum (\hat{y}_i - y)^2 \quad (1)$$
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (2)$$

Sehingga jika diimplementasikan pada code sebagai berikut :

```
def compute_sse(X, labels, centroids):
    labels = np.array(labels)
    return sum(np.sum((X[labels == i] - centroids[i])**2) for i in range(len(centroids)))

# Silhouette Score manual
def silhouette_score(X, labels):
    labels = np.array(labels)
    n = len(X)
    scores = []
    for i in range(n):
        same_cluster = X[labels == labels[i]]
        other_clusters = [X[labels == j] for j in set(labels) if j != labels[i]]
        a = np.mean(np.linalg.norm(X[i] - same_cluster, axis=1))
        b = np.min([np.mean(np.linalg.norm(X[i] - cluster, axis=1)) for cluster in other_clusters])
        scores.append((b - a) / max(a, b))
    return np.mean(scores)
```

Gambar 5.1 Eksperimen Nilai  $K$  Terbaik

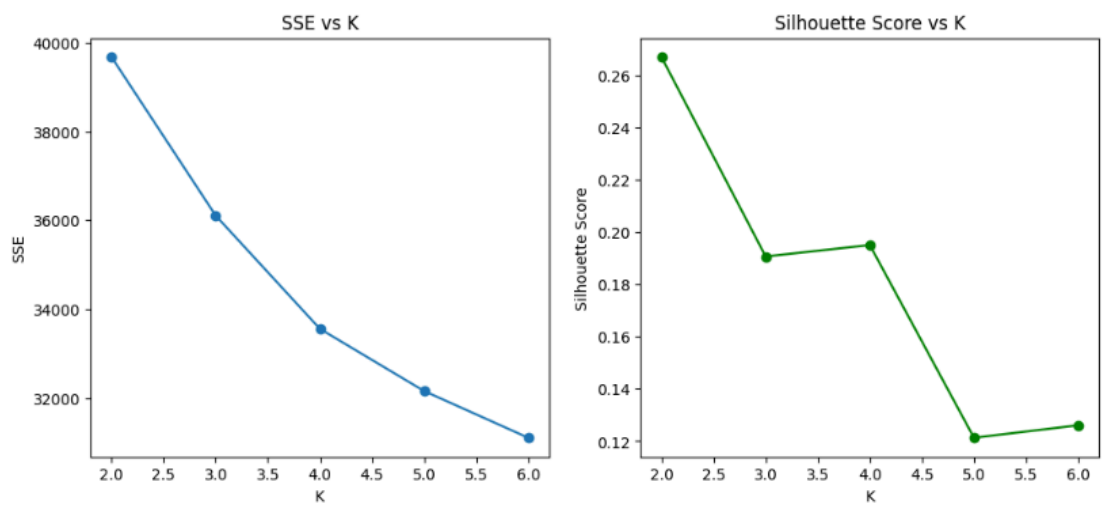
Hasil Eksperimen dilakukan dengan menggunakan nilai  $K$  dari 2 hingga 6 dan diperoleh hasil sebagai berikut :

Nilai $K$	SSE	Silhouette Score
2	39670.17	0.2668
3	36105.84	0.1906

4	33561.31	0.1950
5	32163.60	0.1213
6	31112.06	0.1261

Tabel 5.1 Hasil SSE dan Silhouette Score

Selain melalui perhitungan, kami melakukan visualisasi perbandingan nilai antara K dengan SSE sebagai berikut :



Gambar 5.2 Hasil Perbandingan SSE dan Silhouette Score dengan Nilai K

Berdasarkan grafik dan nilai evaluasi di atas, nilai  $K=4$  dipilih sebagai jumlah kluster yang optimal karena memberikan keseimbangan antara nilai SSE yang relatif rendah dan Silhouette Score yang cukup stabil dibanding nilai lainnya.

#### b. Eksperimen Fitur untuk Clustering

Kami melakukan eksperimen terhadap penggunaan fitur untuk clustering. Eksperimen pertama menggunakan dua fitur yaitu Age yang diperoleh dari hasil fitur engineering dari tahun 2021 - kolom Year\_Birth dan Income. Kedua kolom Age dan Income dipilih karena dianggap cukup mewakili perbedaan karakter pelanggan dari sisi usia dan tingkat pendapatan. Selain itu, keduanya mudah divisualisasikan dan dapat memberikan gambaran awal yang jelas terkait pengelompokan pelanggan tanpa perlu melibatkan semua fitur yang ada. Sedangkan eksperimen kedua menggunakan menggunakan semua fitur dan direduksi dimensi menggunakan Principal Component Analysis (PCA).

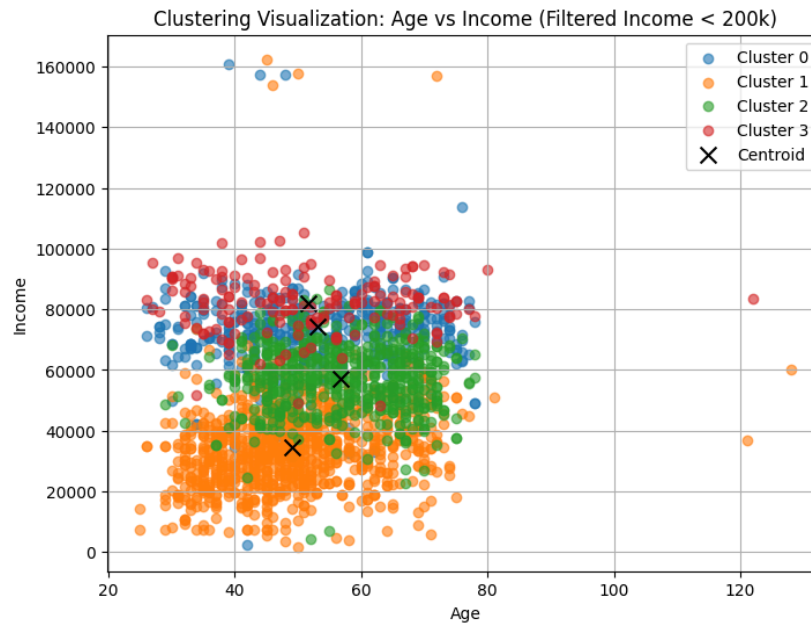
Pada eksperimen pertama dilakukan dua jenis clustering. Diperoleh insight yaitu Setiap warna merepresentasikan satu klaster dari hasil algoritma clustering dengan jumlah klaster (K) sebanyak empat. Titik-titik hitam besar menandai posisi centroid masing-masing klaster. Dari visualisasi tersebut, dapat terlihat bahwa sebagian besar pelanggan berada dalam rentang usia 30 hingga 70 tahun, serta memiliki pendapatan di bawah 100.000. Hal ini mencerminkan dominasi kelompok usia produktif dengan tingkat pendapatan menengah ke bawah dalam dataset. Namun, visualisasi cluster cenderung bertumpuk sehingga sulit untuk diinterpretasikan dan diperlukan fitur yang lebih banyak untuk membentuk segmentasi pelanggan yang benar-benar terpisah. Selain itu, terdapat sejumlah outlier ekstrem, baik dalam usia maupun pendapatan, yang berpotensi mempengaruhi akurasi pembentukan centroid dan dapat dipertimbangkan untuk ditangani lebih lanjut dalam proses preprocessing. Pertama menggunakan kolom Age dan Income sehingga diperoleh hasil sebagai berikut :



Gambar 5.3 Hasil Clustering Age vs Income

Kedua, masih menggunakan kolom Age dan Income, namun dilakukan filter dengan membatasi Income kurang dari 200000. Diperoleh interpretasi bahwa visualisasi mulai terlihat dengan membentuk tiga cluster dengan simbol silang sebagai centroid. Cluster 1 merepresentasikan pelanggan dengan pendapatan yang rendah, sedangkan cluster merah merepresentasikan pelanggan dengan pendapatan yang tinggi, sehingga hasil visualisasi sebagai berikut :





Gambar 5.4 Hasil Clustering Age vs Income dengan Filter Income < 200000

Eksperimen kedua dilakukan dengan melakukan clustering terhadap semua fitur yang tersedia pada dataset yang juga mengaplikasikan PCA dan dengan nilai K terbaik yaitu 4. Setelah dilakukan clustering dengan jumlah kluster terbaik ( $K=4$ ), distribusi data pada ruang PCA menunjukkan pemisahan antar kluster yang cukup jelas. Setiap warna merepresentasikan satu kluster, sementara tanda silang hitam menunjukkan posisi centroid masing-masing kluster setelah diproyeksikan. Dari visualisasi ini, terlihat bahwa PCA membantu menyederhanakan dimensi data sekaligus mempertahankan struktur kluster yang baik. Selain menggunakan Clustering secara manual menggunakan implementasi rumus, kami juga melakukan pengecekan menggunakan library untuk memastikan hasil clustering sama dan serupa.

c. Hasil dan Perbandingan Eksperimen

Eksperimen	Fitur yang Digunakan	Nilai K	Visualisasi/Keterangan
Pemilihan Nilai K Terbaik	Semua fitur	Dilakukan percobaan nilai K dari 2-6	Dihasilkan bahwa nilai K dengan 4 memiliki hasil SSE dan Silhouette terbaik.

Clustering Age dan Income	Kolom Age dan Income	4	Visualisasi cluster kurang terlihat dan masih terdapat outlier.
Clustering Age dan Income dengan filter income < 200000	Kolom Age dan Income	4	Visualisasi cluster lebih terlihat namun masih terdapat outlier dan tidak bisa diinterpretasikan untuk segmentasi pelanggan, sehingga diperlukan fitur lain.
Clustering Semua Fitur dan PCA	Semua Fitur	4	Visualisasi cluster terlihat jelas dan dapat digunakan untuk segmentasi pelanggan.

Tabel 5.2 Tabel Perbandingan Hasil Eksperimen

## 6. KESIMPULAN

Berdasarkan seluruh proses yang dilakukan dalam tugas besar ini, mulai dari eksplorasi data, persiapan fitur, implementasi algoritma K-Means secara manual, hingga evaluasi dan eksperimen, dapat disimpulkan beberapa hal penting sebagai berikut:

### 1. Eksplorasi dan Pembersihan Data

Dataset yang digunakan memiliki 2240 baris dan 29 kolom. Setelah dilakukan pengecekan, ditemukan 24 baris yang memiliki missing values pada kolom Income. Karena jumlah ini hanya sebesar 1.07% dari total data, maka baris-baris tersebut dihapus tanpa mempengaruhi kualitas dataset secara signifikan. Proses pembersihan data juga mencakup pengecekan duplikasi dan pembuatan fitur baru Age yang penting untuk segmentasi pelanggan.

### 2. Pemilihan dan Transformasi Fitur

Untuk proses clustering, dipilih fitur-fitur numerik yang relevan dan representatif seperti Age, Income, Recency, pengeluaran berbagai produk (Mnt...), serta respons terhadap kampanye (AcceptedCmp...). Data kemudian distandarisasi dan diproses lebih lanjut sebelum dimasukkan ke dalam algoritma clustering.

### 3. Implementasi Algoritma Clustering

Tugas besar ini berhasil mengimplementasikan algoritma K-Means dari nol (tanpa *library*) menggunakan Python. Logika dasar K-Means diterapkan mulai dari inisialisasi centroid acak, perhitungan jarak Euclidean, hingga proses iteratif update centroid.

### 4. Evaluasi dan Eksperimen

Model dievaluasi menggunakan *Sum of Squared Errors* (SSE) dan Silhouette Score untuk menentukan kualitas pemodelan klaster. Hasil evaluasi menunjukkan bahwa nilai  $K=4$  memberikan hasil terbaik dengan kompromi antara SSE rendah dan Silhouette Score yang stabil.

### 5. Eksperimen dan Visualisasi

Eksperimen dilakukan dengan menggunakan fitur Age dan Income saja, serta eksperimen lanjutan dengan semua fitur yang direduksi menggunakan PCA. Hasil visualisasi menunjukkan bahwa menggunakan seluruh fitur dengan PCA menghasilkan segmentasi pelanggan yang paling terpisah dan bermakna secara bisnis.

## LAMPIRAN

1. Link video presentasi (YouTube)

<https://youtu.be/vfCgZmcntyU?si=dvdAsgwybFcMJd4y>

2. Link repositori GitHub

<https://github.com/auraauliaan/Customer-Segmentation-Using-K-Means>

3. Sumber Dataset

<https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis>

4. Link Colab

[https://colab.research.google.com/drive/1fH8XSXyUS\\_8t71VUiP7RHPAcswP1QC6Z#scrollTo=Q0WtbK3pm8KU](https://colab.research.google.com/drive/1fH8XSXyUS_8t71VUiP7RHPAcswP1QC6Z#scrollTo=Q0WtbK3pm8KU)