Nama/NIM:Rafli Limandijaya/1103210243

Bagian 1

Berdasarkan analisis clustering menggunakan K-Means (k=5), pelanggan dapat dibagi menjadi 5 segmen berbeda, masing-masing dengan karakteristik yang spesifik.

Cluster	Label	Recency (days)	Frequency	Monetary ($)	Jumlah Pelanggan	Strategi Bisnis
0	Average Customers	30.92	6.02	2325.65	867 (20.57%)	Tingkatkan engagement melalui konten personalisasi dan berikan insentif untuk mendorong pembelian berulang.
1	Lost Customers	251.99	1.49	443.47	988 (23.44%)	Lakukan kampanye reaktivasi dengan diskon besar dan analisis alasan churn.
2	Champions	17.30	15.04	11137.64	69 (1.64%)	Pertahankan loyalitas dengan program eksklusif dan rewards, serta berikan penghargaan khusus.
3	New Customers	52.25	2.09	621.22	2046 (48.54%)	Fokus pada onboarding dan edukasi produk, tawarkan diskon

						first-purchase untuk meningkatkan retensi.
4	Average Customers	19.27	13.39	4791.95	245 (5.81%)	Tingkatkan hubungan melalui komunikasi relevan dan loyalty rewards.

Penjelasan tentang Model Clustering yang digunakan:

1. K-Means Clustering:

- Algoritma partisi yang membagi data menjadi k kelompok berbeda

- Kelebihan: Sederhana, cepat, dan mudah diinterpretasi

- Kekurangan: Sensitif terhadap outlier, memerlukan jumlah cluster (k) ditetapkan sebelumnya

- Cocok untuk: Data dengan cluster berbentuk bulat dan ukuran hampir sama

2. Agglomerative Clustering:

- Algoritma hierarchical yang menggabungkan cluster secara iteratif

- Kelebihan: Dapat divisualisasikan dengan dendrogram, tidak perlu menentukan jumlah cluster sebelumnya

- Kekurangan: Komputasi lebih berat untuk dataset besar

- Cocok untuk: Data dengan hierarki alami atau ketika urutan penggabungan penting

3. DBSCAN:

- Algoritma berbasis densitas yang mengelompokkan area padat

- Kelebihan: Dapat menemukan cluster berbentuk tidak teratur, mendeteksi noise/outlier, tidak memerlukan jumlah cluster

- Kekurangan: Sensitif terhadap parameter eps dan min_samples

- Cocok untuk: Data dengan noise dan cluster berbentuk tidak teratur

4. Gaussian Mixture Model (GMM):

- Model probabilistik yang mengasumsikan data berasal dari campuran distribusi Gaussian

- Kelebihan: Memberikan probabilitas keanggotaan, lebih fleksibel daripada K-Means

- Kekurangan: Dapat overfitting jika komponen terlalu banyak

- Cocok untuk: Data dengan overlap antar cluster dan berbentuk elips

5. Spectral Clustering:

- Menggunakan eigenvector dari matriks kemiripan untuk clustering

- Kelebihan: Dapat menemukan cluster kompleks yang tidak dapat dipisahkan secara linear

- Kekurangan: Komputasi berat untuk dataset besar

- Cocok untuk: Data dengan struktur kompleks dan non-konveks

Perbandingan Metrik Evaluasi:

1. Silhouette Score:

- Mengukur seberapa baik objek cocok dengan clusternya dibandingkan cluster lain

- Rentang -1 hingga 1, nilai lebih tinggi lebih baik

- Silhouette Score tinggi: cluster terpisah dengan baik dan kompak

2. Davies-Bouldin Index:

- Mengukur rasio jarak intra-cluster dan inter-cluster

- Nilai lebih rendah lebih baik
- DB Index rendah: cluster padat dan terpisah dengan baik

3. Calinski-Harabasz Score:

- Dikenal juga sebagai Variance Ratio Criterion
- Mengevaluasi rasio dispersi antar-cluster dengan dispersi intra-cluster
- Nilai lebih tinggi lebih baik
- CH Score tinggi: cluster terpisah dengan baik

Berdasarkan evaluasi keseluruhan, model terbaik adalah: K-Means (k=5)

Analisis Model Terbaik:

K-Means berhasil membagi pelanggan menjadi segmen yang bermakna untuk analisis RFM.

Model ini memiliki keseimbangan yang baik antara interpretabilitas dan performa metrik.

Cluster yang dihasilkan memiliki karakteristik yang jelas dan dapat digunakan untuk strategi pemasaran yang ditargetkan.

Bagian 2(analisis 1-5)

1. Inkonsistensi antara Elbow Method dan Silhouette Score (Silhouette rendah 0.3)

Penyebab:

- Elbow method hanya melihat penurunan *within-cluster sum of squares* (WCSS), tidak mempertimbangkan seberapa "rapi" pembentukan cluster.
- Data mungkin tidak spherical: distribusi pelanggan tidak membentuk cluster bulat — misal ada cluster panjang, tipis, atau overlap.
- Ada outlier atau cluster-size imbalance (jumlah pelanggan antar cluster sangat tidak seimbang).

Strategi Validasi Alternatif:

- Gap Statistic: Membandingkan WCSS aktual dengan WCSS dari data acak, lebih robust.
- Bootstrapping Stability: Melihat apakah cluster stabil jika data sedikit diubah (sampling ulang).
- Kenapa penting? Karena metode ini tidak mengasumsikan bentuk bulat seperti K-Means.

Distribusi Non-Spherical:
Cluster yang bentuknya lonjong, berliku, atau ber-overlap akan membuat Silhouette Score jatuh meskipun WCSS rendah.
Model berbasis jarak Euclidean (seperti K-Means) gagal menangkap bentuk ini.

2. Preprocessing Data Numerik + Kategorikal High-Cardinality (Description)

Metode Efektif:

- StandardScaler untuk numerik (Quantity, UnitPrice).
- TF-IDF atau dimensionality reduction (UMAP/PCA) untuk fitur teks "Description".

Risiko One-Hot Encoding:

- "Description" bisa punya ribuan nilai unik → One-Hot Encoding menghasilkan matriks super sparse.
- Cluster akan lebih berdasarkan teks yang dominan, bukan pola numerik → Bias.

Mengapa TF-IDF / Embedding Lebih Baik:

- TF-IDF merepresentasikan pentingnya kata relatif terhadap semua data → menjaga struktur semantik.
- UMAP/PCA setelah TF-IDF dapat mengurangi dimensi menjadi fitur-fitur ringkas, menjaga jarak semantik antar produk.

3. Menentukan Epsilon Optimal pada DBSCAN untuk Data Imbalanced

Solusi:

- Buat k-distance graph: Untuk tiap titik, hitung jarak ke k tetangga terdekat (misal k=MinPts).
- Cari *elbow point* atau gunakan kuartil ke-3 dari jarak — ini membantu menghindari outlier berjarak jauh.
- MinPts: Disesuaikan, misal MinPts ≈ dimensi data + 1 atau lebih tinggi untuk data super padat.

Alasan pentingnya hal tersebut

- Kalau MinPts terlalu kecil → noise berlebihan.
- Kalau terlalu besar → cluster kecil yang valid bisa diabaikan.

4. Memperbaiki Overlap Cluster "High-Value" dan "Bulk Buyers"

Solusi:

- Constrained Clustering (semi-supervised):
    - Pakai pairwise constraint (must-link/cannot-link).
    - Misal: "Pelanggan A dan B TIDAK boleh di-cluster bareng karena behaviour pembelian berbeda."
- Metric Learning (Mahalanobis Distance):
    - Menyesuaikan jarak antar titik agar fitur penting lebih berpengaruh.
    - Misal: Pembeda utama bisa jadi *frequency* bukan *monetary*.

Tantangan:

- Mahalanobis Distance membuat cluster menjadi *elliptical*, lebih kompleks → Interpretasi bisnis menjadi sulit karena jarak antar pelanggan tidak lagi *intuitif* seperti Euclidean.
- Harus hati-hati menjelaskan hasilnya ke tim bisnis (misal dengan contoh konkret).

5. Merancang Temporal Features dari InvoiceDate

Feature Engineering:

- Buat fitur seperti:
    - Hari dalam minggu (Senin, Selasa, dll.)
    - Jam pembelian (pagi/sore/malam)
    - Hari libur vs hari kerja

Risiko Data Leakage:

- Kalau kamu hitung rata-rata pembelian *menggunakan semua data* → informasi masa depan bocor.
- Solusinya: Gunakan time-based split (misal cross-validation berdasarkan bulan).

Risiko Lag Features:

- Misal: "Jumlah pembelian 7 hari lalu" → noise jika pelanggan tidak rutin belanja mingguan.
- Harus thresholding, hanya gunakan lag jika benar-benar pola periodik terdeteksi.