MODUL PRAKTIKUM

MATA KULIAH DATA MINING

PERTEMUAN 7

SEMESTER GENAP

TAHUN AJARAN 2024/2025

Disusun oleh:

Dwi Welly Sukma Nirad S.Kom, M.T

Aina Hubby Aziira M.Eng

Miftahul Khaira

Dhiya Gustita Aqila

DEPARTEMEN SISTEM INFORMASI

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

TAHUN 2025

IDENTITAS PRAKTIKUM

IDENTITAS MATA KULIAH

Kode mata kuliah	JSI62122
Nama mata kuliah	Data Mining
CPMK yang dibebankan pada praktikum	CPMK-3, CPMK-4 Mahasiswa mampu memahami teknik klasterisasi dalam data mining (CP-2)
Materi Praktikum Pertemuan 7	Konsep Dasar K-Means
	Menghitung Nilai K Optimal
	Cara Kerja Algoritma K-Means
	Implementasi Algoritma K-Means

IDENTITAS DOSEN DAN ASISTEN MAHASISWA

Nama Dosen Pengampu	1. Dwi Welly Sukma Nirad S.Kom, M.T 2. Aina Hubby Aziira M.Eng
Nama Asisten Mahasiswa (Kelas A)	1. 2211523034 - Muhammad Fariz 2. 2211521012 - Rizka Kurnia Illahi 3. 2211521010 - Dhiya Gustita Aqila 4. 2211522013 - Benni Putra Chaniago 5. 2211521017 - Ghina Anfasha Nurhadi 6. 2211523022 - Daffa Agustian Saadi 7. 2211521007 - Annisa Nurul Hakim

	8. 2211522021 - Rifqi Asverian Putra 9. 2211521009 - Miftahul Khaira 10. 2211521015- Nurul Afani 11. 2211523028 - M.Faiz Al-Dzikro
Nama Asisten Mahasiswa (Kelas B)	1. 2211523034 - Muhammad Fariz 2. 2211521012 - Rizka Kurnia Illahi 3. 2211521010 - Dhiya Gustita Aqila 4. 2211522013 - Benni Putra Chaniago 5. 2211521017 - Ghina Anfasha Nurhadi 6. 2211523022 - Daffa Agustian Saadi 7. 2211521007 - Annisa Nurul Hakim 8. 2211522021 - Rifqi Asverian Putra 9. 2211521009 - Miftahul Khaira 10. 2211521015- Nurul Afani 11. 2211523028 - M.Faiz Al-Dzikro

DAFTAR ISI

K-MEANS

A. KONSEP DASAR K-MEANS

K-Means adalah algoritma unsupervised learning yang digunakan untuk mengelompokkan data yang tidak berlabel ke dalam kelompok atau kluster. Bentuk pengelompokkan ini menetapkan bahwa hanya terdapat satu titik data (centroid) dalam satu kluster. Jenis analisis klaster ini umumnya digunakan untuk menentukan segmentasi pasar, pengelompokkan dokumen, segmentasi gambar, identifikasi daerah rawan kejahatan, deteksi penipuan, dan mengidentifikasi kanker. Algoritma K-Means merupakan metode yang banyak digunakan dalam analisis klaster karena efisien, efektif, dan sederhana.

K-Means adalah algoritma pengelompokkan berbasis centroid berulang yang mempartisi kumpulan data menjadi beberapa klaster berdasarkan jarak ke centroid. K-Means meminimalkan jumlah jarak antara titik-titik data dan centroid. Setelah itu, titik-titik data tersebut dikategorikan ke dalam klaster dengan menggunakan ukuran jarak matematis dari pusat klaster

Kelebihan dari algoritma K-Means, yaitu:

1. Algoritma yang sederhana dan mudah diterapkan
2. Pemrosesan yang cepat
3. Mudah beradaptasi dengan contoh baru
4. Umum diimplementasikan ke klaster dengan bentuk dan ukuran yang berbeda

Kekurangan dari algoritma K-Means, yaitu:

1. Hasilnya sensitif terhadap jumlah klaster (K)
2. Sensitif terhadap inisialisasi “seed”
3. Sensitif terhadap outlier
4. Sensitif terhadap data dengan variabel yang memiliki skala berbeda

B. MENGHITUNG NILAI K OPTIMAL

Kinerja algoritma K-Means yang efisien sangat bergantung pada ukuran klaster yang dibentuk. Terdapat beberapa metode yang dapat digunakan dalam menemukan ukuran klaster atau nilai K pada algoritma K-Means.

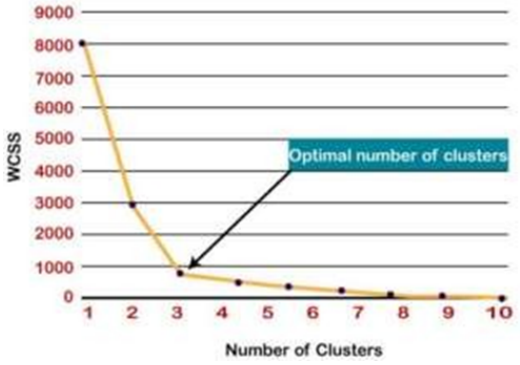
a. Metode Elbow

Metode ini digunakan untuk menghasilkan informasi dalam menentukan jumlah klaster terbaik dengan cara melihat persentase perbandingan antara jumlah klaster yang akan membentuk siku pada suatu titik. Metode ini memberikan ide/gagasan dengan cara memilih nilai klaster dan kemudian menambah nilai klaster tersebut untuk dijadikan model data dalam penentuan klaster terbaik. Selain itu, persentase perhitungan yang dihasilkan menjadi pembanding antara jumlah klaster yang ditambah. Hasil persentase yang berbeda dari setiap nilai klaster dapat ditunjukan dengan menggunakan grafik sebagai sumber informasinya. Dengan menghitung Within-Cluster Sum of Squares. (WCSS) untuk nilai K yang berbeda, dan pilih K yang WCSS mengalami penurunan drastis sehingga mendekati bentuk siku dibandingkan nilai K yang lain. Semaking berar jumlah klaster maka nilai WCSS akan semakin kecil. Berikut merupakan perhitungan rumus WCSS.

$$\mathrm{WCSS}(k) = \sum_{j=1}^{K} \sum_{\mathbf{x}_i \in \text{cluster } j} \|\mathbf{x}_i - \bar{\mathbf{x}}_j\|^2,$$

where $\bar{\mathbf{x}}_j$ is the sample mean in cluster j

Gambar berikut menunjukkan bagaimana grafik hasil metode elbow serta penentuan nilai K. Setelah dilihat akan ada beberapa nilai K yang mengalami penurunan paling besar dan selanjutnya hasil dari nilai K akan turun secara perlahan- lahan sampai hasil dari nilai K tersebut stabil. Misalnya, saat nilai K=2 ke K=3, terjadi penurunan, selanjutnya K=3 ke K=4 terjadi lagi penurunan drastis yang membentuk siku pada titik K=3. Oleh karena itu, nilai kluster K yang dianggap optimal adalah K=3.



b. Metode Silhoutette

Silhouette Coefficient digunakan untuk melihat kualitas dan kekuatan cluster, seberapa baik suatu objek ditempatkan dalam suatu kluster. Metode ini merupakan gabungan dari metode cohesion dan separation. Cohesion diukur dengan menghitung seluruh objek yang terdapat dalam sebuah kluster dan separation diukur dengan menghitung jarak rata-rata setiap objek dalam sebuah cluster dengan cluster terdekatnya.

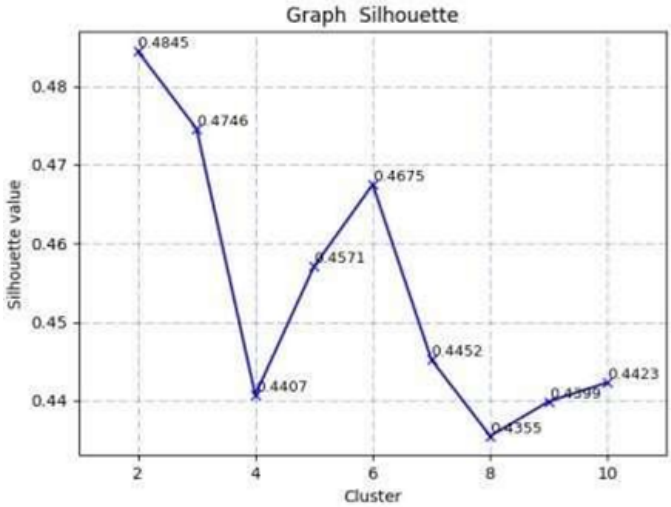
Nilai silhouette untuk keseluruhan data dengan jumlah klaster k, dapat didefinisikan sebagai sil (k) yang dihitung dengan persamaan berikut yakni rata-rata silhouette value untuk semua kluster.

$$sil\,(c) = sil\,(k)\frac{1}{|k|}\sum_{i=1}^{k} sil(c_i)$$

Keterangan:

sil (k) : nilai silhouette semua cluster
|k| : banyaknya *cluster k*
sil (c_i) : rata-rata nilai *silhouette*

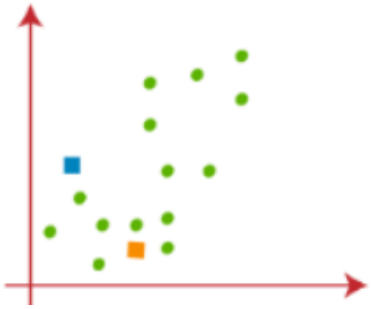
Gambar berikut menunjukkan bagaimana grafik hasil metode silhouette dalam penentuan nilai K. Dilakukan pengujian dengan jumlah K=2 sampai K=10. Berdasarkan hasil perhitungan metode sillhoutter coefficient, terlihat bahwa nilai silhouette maksimum ada pada K=2.



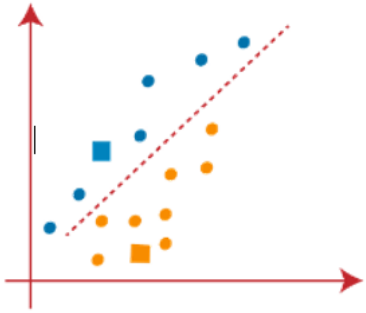
C. CARA KERJA ALGORITMA K-MEANS

Berikut merupakan cara kerja algoritma K-Means

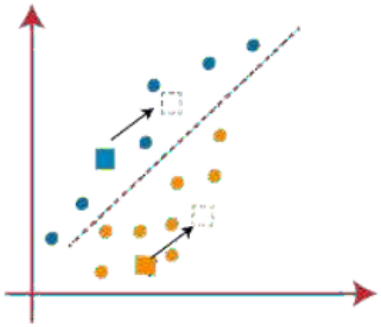
1. Tentukan jumlah klaster
2. Inisialisasi titik K atau centroid secara acak



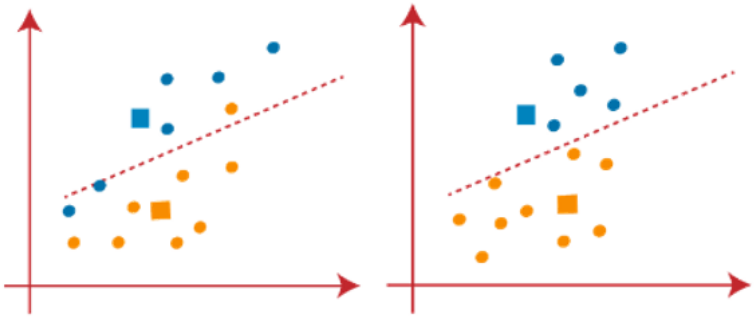
3. Berikan label pada semua data berdasarkan centroid terdekat



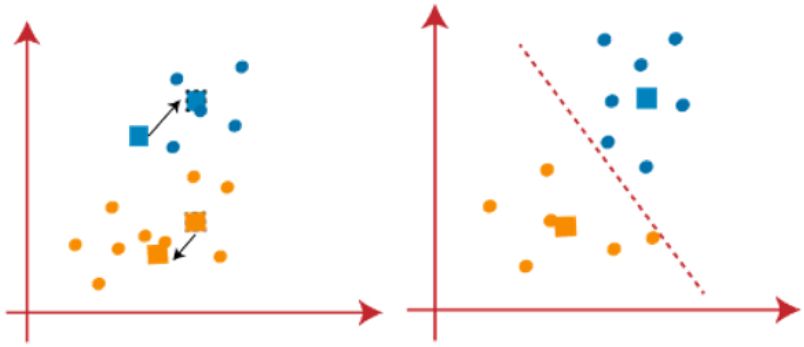
4. Tentukan letak centroid baru berdasarkan rata-rata jarak setiap titik data dalam klaster dan posisikan setiap titip data ke centroid terdekat. Metode yang dapat digunakan seperti euclidean distance.



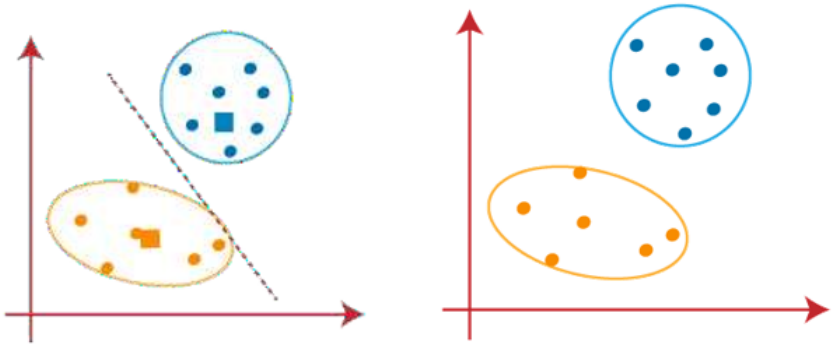
5. Lakukan pelabelan ulang data berdasarkan jarak terhadap centroid baru



6. Ulangi langkah keempat dan kelima hingga nilai centroid tetap/tidak berubah lagi



Berikut hasil akhir setelah tidak ditemukan perpindahan pada centroid di setiap klaster



D. IMPLEMENTASI ALGORITMA K-MEANS

Camellia Mall ingin menyesuaikan strategi promosi agar lebih tepat sasaran, terutama untuk produk dengan harga dan nilai berbeda. Oleh karena itu, Camellia Mall melakukan segmentasi pelanggan berdasarkan pendapatan per tahun dan daya beli pelanggan. Dalam melakukan segmentasi, Mall menggunakan data pelanggan yang dimiliki lalu kemudian menggunakan model algoritma K-Means untuk mengelompokkan pelanggan. Langkah-langkahnya sebagai berikut.

1. Load dataset

```
[123]: df = pd.read_csv('Mall_Customers.csv')
       df.head(10)
```

[123]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

2. Melakukan preprocessing data
    a. Memilih atribut yang relevan dan melakukan indexing

```
[124]: df = df.iloc[:,[3,4]]
       df
```

[124]:

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

b. Melakukan pengecekan nilai null

```
[126]: df.isnull().sum()

[126]: Annual_Income     0
       Spending_Score    0
       dtype: int64
```

c. Melakukan pengecekan duplikat

```
[127]: df.duplicated().sum()

[127]: 4
```

d. Drop baris yang terdapat null value atau duplikat

```
[137]: df = df.drop_duplicates()
       df.duplicated().sum()

[137]: 0
```

3. Konversi data frame menjadi array yang akan digunakan untuk proses scaling

```
[128]: x_array = nm.array(df)
       x_array

[128]: array([[ 15,  39],
              [ 15,  81],
              [ 16,   6],
              [ 16,  77],
              [ 17,  40],
              [ 17,  76],
              [ 18,   6],
              [ 18,  94],
              [ 19,   3],
              [ 19,  72],
              [ 19,  14],
              [ 19,  99],
              [ 20,  15],
              [ 20,  77],
              [ 20,  13],
              [ 20,  79],
              [ 21,  35],
              [ 21,  66],
```

4. Lakukan proses scaling agar skala semua atribut sama

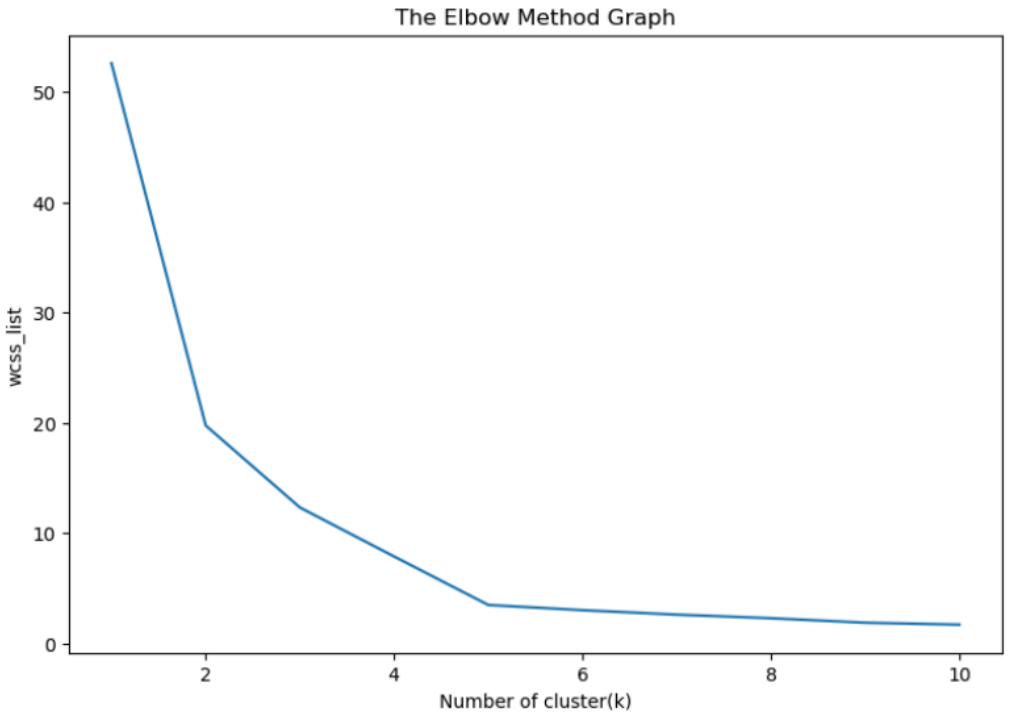
```
[139]: from sklearn.preprocessing import MinMaxScaler

       scaler = MinMaxScaler()
       x_scaled = scaler.fit_transform(x_array)
       x_scaled
```

```
[139]: array([[0.        , 0.3877551 , 0.5       ],
              [0.        , 0.81632653, 0.75      ],
              [0.00819672, 0.05102041, 0.5       ],
              [0.00819672, 0.7755102 , 0.75      ],
              [0.01639344, 0.39795918, 0.5       ],
              [0.01639344, 0.76530612, 0.75      ],
              [0.02459016, 0.05102041, 0.5       ],
              [0.02459016, 0.94897959, 0.75      ],
              [0.03278689, 0.02040816, 0.5       ],
              [0.03278689, 0.7244898 , 0.75      ],
              [0.03278689, 0.13265306, 0.5       ],
              [0.03278689, 1.        , 0.75      ],
              [0.04098361, 0.14285714, 0.5       ],
              [0.04098361, 0.7755102 , 0.75      ],
              [0.04098361, 0.12244898, 0.5       ],
              [0.04098361, 0.79591837, 0.75      ],
              [0.04918033, 0.34693878, 0.5       ],
              [0.04918033, 0.66326531, 0.75      ],
```

5. Menentukan jumlah klaster optimal menggunakan metode Elbow

```
[146]: from sklearn.cluster import KMeans
       wcss_list=[]
       for i in range (1,11):
           kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
           kmeans.fit(x_scaled)
           wcss_list.append(kmeans.inertia_)
       fig = mtp.figure(figsize=(9,6))
       mtp.plot(range(1,11),wcss_list)
       mtp.title('The Elbow Method Graph')
       mtp.xlabel('Number of cluster(k)')
       mtp.ylabel('wcss_list')
       mtp.show()
```



6. Inisialisasi dan training model K-Means

```
[141]: kmeans = KMeans(n_clusters=5,init='k-means++', random_state=42,n_init=10)
       kmeans.fit(x_scaled)
```

```
[141]: KMeans
       KMeans(n_clusters=5, n_init=10, random_state=42)
```

7. Menambahkan label klaster pada dataframe

[142]:
```
df["klaster"] = kmeans.labels_
df
```

[142]:

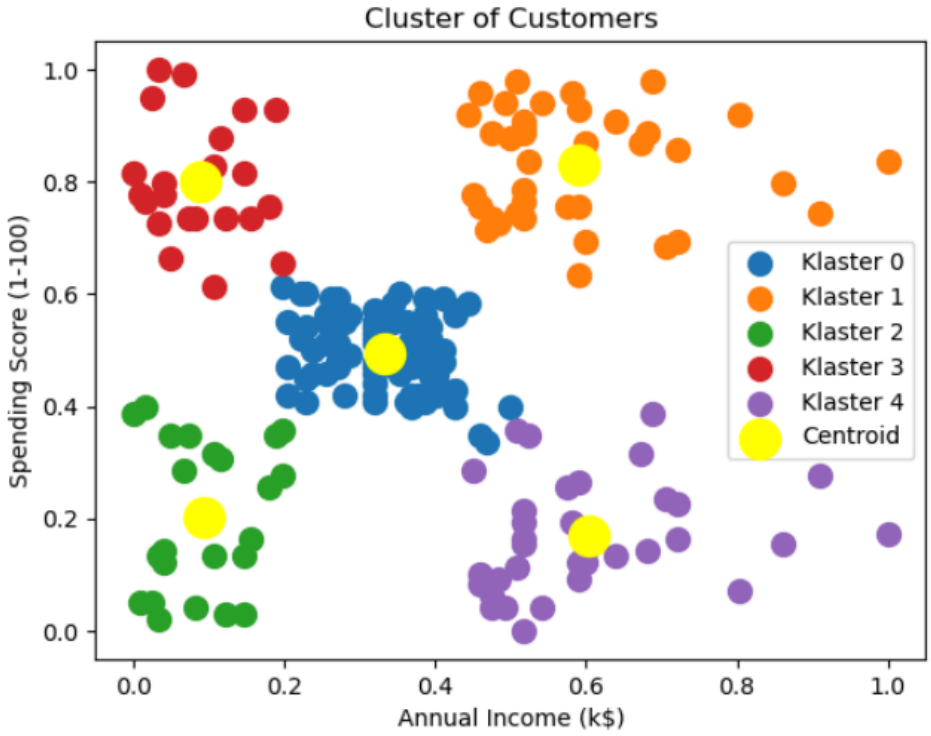
	Annual_Income	Spending_Score	klaster
0	15	39	2
1	15	81	3
2	16	6	2
3	16	77	3
4	17	40	2
...	...	...	...
195	120	79	1
196	126	28	4
197	126	74	1
198	137	18	4
199	137	83	1

196 rows × 3 columns

8. Memvisualisasikan hasil klastering menggunakan scatter plot dan boxplot

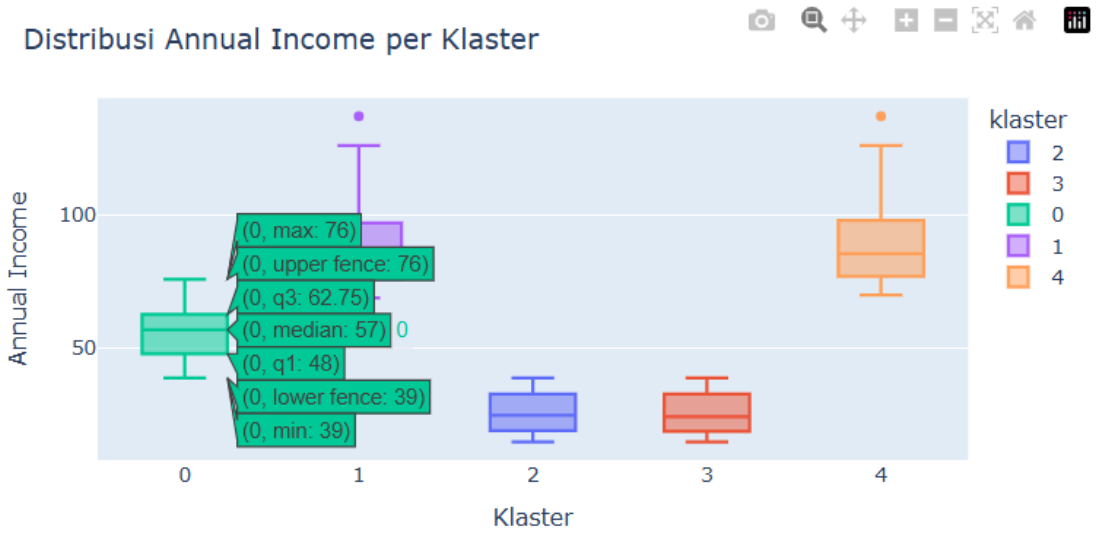
[143]:
```
for i in range(kmeans.n_clusters):
    mtp.scatter(
        x_scaled[df.klaster == i, 0],
        x_scaled[df.klaster == i, 1],
        s=100,
        label=f"Klaster {i}",
        alpha=1,
        marker="o"
    )

centers = kmeans.cluster_centers_
mtp.scatter(centers[:,0], centers[:,1], c="yellow", s=300, alpha=1,marker="o", label="Centroid")
mtp.title("Cluster of Customers")
mtp.xlabel('Annual Income (k$)')
mtp.ylabel('Spending Score (1-100)')
mtp.legend()
mtp.show()
```



[147]:
```
import plotly.express as px

fig = px.box(df, x='klaster', y='Annual_Income', hover_data=['Annual_Income'], color='klaster')
fig.update_layout(
    title="Distribusi Annual Income per Klaster",
    xaxis_title="Klaster",
    yaxis_title="Annual Income"
)
fig.show()
```



9. Analisis dan interpretasi hasil

Berdasarkan klastering yang telah dilakukan menggunakan algoritma K-Means didapatkan lima klaster yang memiliki karakteristik yang berbeda ,yaitu:

a. **Klaster 0** : Pelanggan Pendapatan Sedang & Spending Score Sedang

   Belanja cukup rutin dengan mempertimbangkan harga dan kebutuhan, tawarkan program loyalitas, paket bundling atau program cashback, tekankan pada kualitas dan manfaat produk, serta ciptakan pengalaman belanja yang nyaman

b. **Klaster 1** : Pelanggan Pendapatan Tinggi & Spending Score Tinggi

   Sering belanja cocok untuk produk premium dan program loyalty eksklusif

c. **Klaster 2** : Pelanggan Pendapatan Rendah & Spending Score Rendah

   Jarang belanja, perlu pendekatan khusus seperti kupon atau diskon besar.

d. **Klaster 3** : Pelanggan Pendapatan Rendah & Spending Score Tinggi

   Belanja impulsif, cocok untuk penawaran menarik, bundling murah, dan promo cicilan.

e. **Klaster 4** : Pelanggan Pendapatan Tinggi & Spending Score Rendah

   Daya beli tinggi tapi jarang belanja, cocok untuk promosi personal & produk edisi terbatas.

REFERENSI

IBM. 2025. “Bagaimana Cara Kerja dengan Clustering?”. Apa itu k-means clustering? | IBM. Diakses pada 26 April 2025.

Trivusi. 2022. “K-Means Clustering : Pengertian, Cara kerja, Kelebihan, dan Kekurangannya”. K-Means Clustering: Pengertian, Cara Kerja, Kelebihan, dan Kekurangannya - Trivusi. Diakses pada 26 April 2025

Scikit-Learn. 2025.”K-Means”.KMeans — scikit-learn 1.6.1 documentation. Diakses pada 26 April 2025.