Nama Dosen : Teguh Iman Hermanto, M.Kom
Mata Kuliah : Machine Learning 1
Pembahasan : K-Means Clustering
Pokok Pemb : - Membangun Model K-Means Clustering
- Simulasi Algoritma K-means

1. Membangun Model K-means

```
1 mport pandas as pd
2
3 from sklearn.cluster import KMeans
4 from sklearn.preprocessing import StandardScaler
5 import matplotlib.pyplot as plt
6
7 import seaborn as sns
8 from sklearn.decomposition import PCA
```

```
1 # Memuat dataset
2 df = pd.read_csv("pelanggan.csv")
```

```
1 # Pie chart distribusi gender pelanggan
2 gender_counts = df['Gender'].value_counts()
3 plt.pie(gender_counts,
4         labels=gender_counts.index,
5         autopct='%1.1f%%',
6         colors=['skyblue', 'pink'],
7         startangle=90)
8 plt.title("Distribusi Gender Pelanggan")
9 plt.show()
```

```python
# Boxplot pengeluaran rata-rata per transaksi berdasarkan kategori produk
sns.boxplot(data=df,
            x='Preferred_Product_Category',
            y='Average_Spending_Per_Transaction',
            palette='Set3')
plt.title("Pengeluaran Rata-Rata Per Transaksi Berdasarkan Kategori Produk")
plt.xlabel("Kategori Produk")
plt.ylabel("Rata-Rata Pengeluaran Per Transaksi (Rp)")
plt.xticks(rotation=45)
plt.show()
```

```python
# Frekuensi pembelian berdasarkan kota
sns.barplot(data=df,
            x='City',
            y='Purchase_Frequency',
            estimator=sum,
            palette='pastel')
plt.title("Frekuensi Pembelian Berdasarkan Kota")
plt.xlabel("Kota")
plt.ylabel("Total Frekuensi Pembelian")
plt.show()
```

```python
# Bar plot kategori produk yang disukai
sns.countplot(data=df,
              x='Preferred_Product_Category',
              order=df['Preferred_Product_Category'].value_counts().index,
              palette='coolwarm')
plt.title("Kategori Produk yang Disukai Pelanggan")
plt.xlabel("Kategori Produk")
plt.ylabel("Jumlah Pelanggan")
plt.show()
```

```python
# Distribusi frekuensi pembelian pelanggan
sns.histplot(df['Purchase_Frequency'],
             kde=True,
             color='orange',
             bins=15)
plt.title("Distribusi Frekuensi Pembelian Pelanggan")
plt.xlabel("Frekuensi Pembelian")
plt.ylabel("Jumlah Pelanggan")
plt.show()
```

2. Simulasi Algoritma K-Means

```python
# Memilih fitur untuk clustering
features = ["Annual_Income",
            "Purchase_Frequency",
            "Average_Spending_Per_Transaction",
            "Total_Spent"]
```

```python
# Standarisasi data
scaler = StandardScaler()
scaled_features = scaler.fit_transform(df[features])
```

```python
# Menentukan jumlah cluster menggunakan metode Elbow
inertia = []
k_values = range(1, 11)

for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_features)
    inertia.append(kmeans.inertia_)
```

```python
# Plot Elbow Method
plt.figure(figsize=(8, 5))
plt.plot(k_values, inertia, marker='o')
plt.title("Elbow Method for Optimal k")
plt.xlabel("Number of Clusters (k)")
plt.ylabel("Inertia")
plt.grid()
plt.show()
```

```python
# Menggunakan jumlah cluster optimal
optimal_k = 4  # Berdasarkan elbow method (misalnya)
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
df['Cluster'] = kmeans.fit_predict(scaled_features)
```

```python
# Mengurangi dimensi data untuk visualisasi
pca = PCA(n_components=2)
reduced_features = pca.fit_transform(scaled_features)
```

```python
# Warna untuk setiap cluster
colors = ['blue', 'green', 'red', 'purple']
```

```python
# Plot hasil clustering
plt.figure(figsize=(10, 7))
for cluster in range(optimal_k):
    cluster_points = reduced_features[df['Cluster'] == cluster]
    plt.scatter(cluster_points[:, 0],
                cluster_points[:, 1],
                s=50,
                label=f'Cluster {cluster}',
                color=colors[cluster])

# Menambahkan centroid pada plot
centroids_reduced = pca.transform(kmeans.cluster_centers_)
plt.scatter(centroids_reduced[:, 0],
            centroids_reduced[:, 1],
            s=200,
            c='black',
            marker='X',
            label='Centroids')

# Menambahkan detail plot
plt.title("Hasil Clustering K-Means (2D PCA Projection)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.legend()
plt.grid()
plt.show()
```