MODUL PRAKTIKUM

IS411 – DATA MODELING
PROGRAM SARJANA S1 SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA

Gedung B Lantai 5, Kampus UMN
Jl. Scientia Boulevard, Gading Serpong, Tangerang, Banten-15811 Indonesia
Telp: +62-21.5422.0808 (ext. 1803), email: ict.lab@umn.ac.id, web: umn.ac.id

MODUL 12
K-Prototype and K-Modes

DESKRIPSI TEMA

1. K-Prototype
2. K-Modes

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Students are able to illustrate data modeling problems using classification techniques (C4)

PENUNJANG PRAKTIKUM

1. Anaconda Navigator
2. Jupyter Notebook

(+ Perlengkapan Apd/Alat Pelindung Diri Yang Harus Digunakan, Jika Ada)

LANGKAH-LANGKAH PRAKTIKUM

K-Prototype

Import Library

1. Lakukan langkah yang sama seperti minggu sebelumnya untuk meng-import library Numpy dan Pandas, numpy, plotnine, module for k-protoype cluster.

```
# Import module for data manipulation
import pandas as pd
# Import module for linear algebra
import numpy as np
# Import module for data visualization
from plotnine import *
import plotnine

# Import module for k-protoype cluster
from kmodes.kprototypes import KPrototypes
```

```
# Ignore warnings
import warnings
warnings.filterwarnings('ignore', category = FutureWarning)
```

```
# Format scientific notation from Pandas
pd.set_option('display.float_format', lambda x: '%.3f' % x)
```

Import Data

2. Gunakan dataset **10000 Sales Records.csv'**dan masukkan ke dalam dataframe menggunakan Pandas. Berikan nama **df** (Petunjuk: gunakan delimiter=;).
3. Tampilkan informasi dataset dan isi data.

```
df = pd.read_csv('data/10000 Sales Records.csv')
```

```
print('Dimension data: {} rows and {} columns'.format(len(df), len(df.columns)))
df.head()
```

Data tersebut sebenarnya adalah Data Penjualan Negara. Data tersebut memiliki 10.000 baris dan 14 kolom dengan tipe data campuran (numerik dan kategorikal). Ini mencatat transaksi penjualan berdasarkan negara di seluruh dunia.

```
Dimension data: 10000 rows and 14 columns
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	Sub-Saharan Africa	Chad	Office Supplies	Online	L	1/27/2011	292494523	2/12/2011	4484	651.21	524.96	2920025.64	2353920.64	566105.00
1	Europe	Latvia	Beverages	Online	C	12/28/2015	361825549	1/23/2016	1075	47.45	31.79	51008.75	34174.25	16834.50
2	Middle East and North Africa	Pakistan	Vegetables	Offline	C	1/13/2011	141515767	2/1/2011	6515	154.06	90.93	1003700.90	592408.95	411291.95
3	Sub-Saharan Africa	Democratic Republic of the Congo	Household	Online	C	9/11/2012	500364005	10/6/2012	7683	668.27	502.54	5134318.41	3861014.82	1273303.59
4	Europe	Czech Republic	Beverages	Online	C	10/27/2015	127481591	12/5/2015	3491	47.45	31.79	165647.95	110978.89	54669.06

Untuk memastikan tipe data setiap kolom dipetakan dengan benar, kita harus memeriksa tipenya secara manual menggunakan perintah **df.info().** Jika menemukan ada pemetaan yang salah, harus memperbaikinya ke tipe data yang benar.

4. Inspect the data type

```
# Inspect the data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Region          10000 non-null  object
 1   Country         10000 non-null  object
 2   Item Type       10000 non-null  object
 3   Sales Channel   10000 non-null  object
 4   Order Priority  10000 non-null  object
 5   Order Date      10000 non-null  object
 6   Order ID        10000 non-null  int64
 7   Ship Date       10000 non-null  object
 8   Units Sold      10000 non-null  int64
 9   Unit Price      10000 non-null  float64
 10  Unit Cost       10000 non-null  float64
 11  Total Revenue   10000 non-null  float64
 12  Total Cost      10000 non-null  float64
 13  Total Profit    10000 non-null  float64
dtypes: float64(5), int64(2), object(7)
memory usage: 1.1+ MB
```

Untungnya, semua kolom memiliki tipe data yang tepat. Harap abaikan ID Pesanan karena kita tidak akan menggunakannya dan akan dihapus nanti.

Ada tujuh variabel kategori dalam dataset. Negara yang memiliki 185 nilai unik, Tanggal Pemesanan dengan 2691 nilai unik, dan Tanggal Pengiriman dengan 2719 nilai unik akan dihapus dari analisis klaster karena memiliki banyak nilai unik. Sisa kolom akan disimpan. Mereka adalah Wilayah dengan 7 nilai unik, Tipe Barang dengan 12 nilai unik, Saluran Penjualan yang memiliki 2 nilai unik, dan Prioritas Pesanan dengan 4 nilai unik.

5. Inspect the categorical variables

```
1 #Inspect the categorical variabels
2 df.select_dtypes('object').nunique()
```

```
Region              7
Country           185
Item Type          12
Sales Channel       2
Order Priority      4
Order Date       2691
Ship Date        2719
dtype: int64
```

Kolom lainnya adalah variabel numerik. Mereka adalah Order OF, Unit Terjual, Unit Price, Unit Cost, Total Revenue, Total Cost, dan Total Profit.

6. Inspect the numerical variables.

```
## Inspect the numerical variables
df.describe()
```

	Order ID	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	1.000000e+04	10000.000000	10000.000000	10000.000000	1.000000e+04	1.000000e+04	1.000000e+04
mean	5.498719e+08	5002.855900	268.143139	188.806639	1.333355e+06	9.382658e+05	3.950893e+05
std	2.607835e+08	2873.246454	217.944092	176.445907	1.465026e+06	1.145914e+06	3.775550e+05
min	1.000892e+08	2.000000	9.330000	6.920000	1.679400e+02	1.245600e+02	4.338000e+01
25%	3.218067e+08	2530.750000	109.280000	56.670000	2.885511e+05	1.647855e+05	9.832914e+04
50%	5.485663e+08	4962.000000	205.700000	117.110000	8.000512e+05	4.816058e+05	2.890990e+05
75%	7.759981e+08	7472.000000	437.200000	364.690000	1.819143e+06	1.183822e+06	5.664227e+05
max	9.999342e+08	10000.000000	668.270000	524.960000	6.680027e+06	5.241726e+06	1.738178e+06

Tugas terakhir sebelum melakukan eksplorasi dan analisis data adalah memastikan data tidak mengandung nilai yang hilang.

7. Check missing value.

```
# Check missing value
df.isna().sum()
```

```
Region            0
Country           0
Item Type         0
Sales Channel     0
Order Priority    0
Order Date        0
Order ID          0
Ship Date         0
Units Sold        0
Unit Price        0
Unit Cost         0
Total Revenue     0
Total Cost        0
Total Profit      0
dtype: int64
```

Explanatory Data Analysis

Sebelum masuk ke analisis klaster, kita harus melakukan eksplorasi data untuk analisis deskriptif. Hal ini bertujuan untuk mengetahui hal menarik yang dapat berguna untuk menghasilkan laporan dan menangkap fenomena dalam data.

Kita memiliki hipotesis bahwa jumlah pembelian di setiap wilayah memiliki korelasi linier yang kuat terhadap total keuntungan. Untuk menyimpulkan ini, kami memiliki dua pilihan, analisis deskriptif, dan analisis inferensi. Untuk bagian ini, kita akan memilih opsi pertama.

8. Distribution of region

```
# The distribution of sales each region
df_region = pd.DataFrame(df['Region'].value_counts()).reset_index()
df_region['Percentage'] = df_region['Region'] / df['Region'].value_counts().sum()
df_region.rename(columns = {'index':'Region', 'Region':'Total'}, inplace = True)
df_region = df_region.sort_values('Total', ascending = True).reset_index(drop = True)
df_region
```

```
# The dataframe
df_region = df.groupby('Region').agg({
    'Region': 'count',
    'Units Sold': 'mean',
    'Total Revenue': 'mean',
    'Total Cost': 'mean',
    'Total Profit': 'mean'
    }
).rename(columns = {'Region': 'Total'}).reset_index().sort_values('Total', ascending = True)
```

```
df_region
```

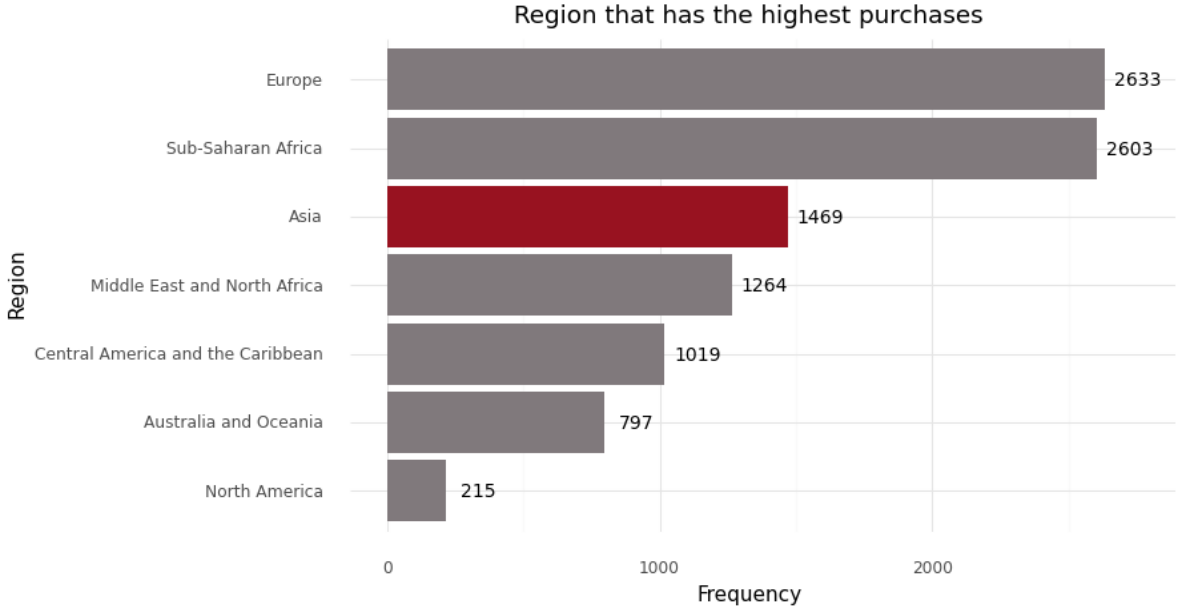
	Region	Total	Units Sold	Total Revenue	Total Cost	Total Profit
5	North America	215	5373.358140	1.559779e+06	1.097009e+06	462769.837767
1	Australia and Oceania	797	4986.769134	1.317192e+06	9.105785e+05	406613.816073
2	Central America and the Caribbean	1019	5081.062807	1.369509e+06	9.736721e+05	395836.947713
4	Middle East and North Africa	1264	5116.219146	1.357305e+06	9.538842e+05	403420.802634
0	Asia	1469	5015.488087	1.365082e+06	9.652160e+05	399866.097243
6	Sub-Saharan Africa	2603	4967.807530	1.287190e+06	9.031555e+05	384034.610642
3	Europe	2633	4920.384732	1.322207e+06	9.321582e+05	390049.226282

Dari hasil di atas, kita dapat menyimpulkan bahwa Amerika Utara adalah wilayah dengan jumlah penjualan terendah tetapi mereka mengungguli semua wilayah di kolom tertentu seperti Unit Terjual, Total Pendapatan, Total Biaya, dan Total Laba. Tidak seperti wilayah lain, Amerika Utara melakukan banyak pembelian pada saat yang bersamaan. Sedangkan Eropa yang memiliki jumlah pembelian terbanyak tidak memberikan kontribusi total keuntungan yang signifikan. Artinya jumlah pembelian tidak memiliki hubungan linier yang kuat terhadap total laba.

9. Data Visualization

```
#Data visualization

plotnine.options.figure_size = (8, 4.8)
(
    ggplot(data = df_region)+
    geom_bar(aes(x = 'Region',
                 y = 'Total'),
             fill = np.where(df_region['Region'] == 'Asia', '#981220', '#80797c'),
             stat = 'identity')+
    geom_text(aes(x = 'Region',
                  y = 'Total',
                  label = 'Total'),
              size = 10,
              nudge_y = 120)+
    labs(title = 'Region that has the highest purchases')+
    xlab('Region')+
    ylab('Frequency')+
    scale_x_discrete(limits = df_region['Region'].tolist())+
    theme_minimal()+
    coord_flip()
)
```



Untuk eksplorasi data, kita akan membuat tabulasi silang antara Region dan Item Type untuk melihat pola apa pun.

10. Distribution of item type

```
# Order the index of cross tabulation
order_region = df_region['Region'].to_list()
order_region.append('All')
order_region
```

```
df_item = pd.crosstab(df['Region'], df['Item Type'], margins = True).reindex(order_region, axis = 0).reset_index()
# Remove index name
df_item.columns.name = None
df_item
```

	Region	Baby Food	Beverages	Cereal	Clothes	Cosmetics	Fruits	Household	Meat	Office Supplies	Personal Care	Snacks	Vegetables	All
0	North America	21	20	16	21	20	15	20	17	20	17	16	12	215
1	Australia and Oceania	65	50	69	77	75	55	78	61	50	76	72	69	797
2	Central America and the Caribbean	74	92	77	84	77	81	104	75	94	82	89	90	1019
3	Middle East and North Africa	105	96	104	111	99	104	128	101	103	112	95	106	1264
4	Asia	132	108	121	116	125	111	116	114	132	137	120	137	1469
5	Sub-Saharan Africa	235	220	211	229	203	230	218	207	207	223	221	199	2603
6	Europe	210	196	227	234	235	199	211	223	231	241	203	223	2633
7	All	842	782	825	872	834	795	875	798	837	888	816	836	10000

Data Pre-processing

Pra-pemrosesan data bertujuan untuk menghapus kolom yang tidak digunakan yaitu Negara, Tanggal Pemesanan, ID Pesanan, dan Tanggal Pengiriman. Mereka tidak relevan mengenai algoritma pengelompokan K-Prototype. Ada dua alasan mengapa kita perlu menghapus kolom ini sebagai berikut:

• Negara — memiliki banyak nilai unik yang menambah beban komputasi. Informasinya terlalu banyak untuk diproses dan menjadi tidak berarti

• Tanggal Pemesanan dan Tanggal Pengiriman — algoritma pengelompokan membutuhkan asumsi bahwa baris dalam data mewakili pengamatan unik yang diamati dalam periode waktu tertentu

• ID Pesanan — memiliki informasi yang tidak berarti untuk analisis klister.

11. Data pre-processing

```
df.drop(['Country', 'Order Date', 'Order ID', 'Ship Date'], axis = 1, inplace = True)
```

```
print('Dimension data: {} rows and {} columns'.format(len(df), len(df.columns)))
df.head()
```

```
Dimension data: 10000 rows and 10 columns
```

	Region	Item Type	Sales Channel	Order Priority	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	Sub-Saharan Africa	Office Supplies	Online	L	4484	651.21	524.96	2920025.64	2353920.64	566105.00
1	Europe	Beverages	Online	C	1075	47.45	31.79	51008.75	34174.25	16834.50
2	Middle East and North Africa	Vegetables	Offline	C	6515	154.06	90.93	1003700.90	592408.95	411291.95
3	Sub-Saharan Africa	Household	Online	C	7683	668.27	502.54	5134318.41	3861014.82	1273303.59
4	Europe	Beverages	Online	C	3491	47.45	31.79	165647.95	110978.89	54669.06

Cluster Analysis

Algoritma clustering K-Prototype dalam modul K-Modes membutuhkan variabel kategori atau posisi kolom dalam data. Tugas ini bertujuan untuk menyimpannya dalam variabel tertentu **catColumnsPos**. Ini akan ditambahkan untuk tugas selanjutnya dalam analisis klaster. Posisi kolom kategoris ada di empat kolom pertama dalam data.

12. Get the position of categorical colums

```
# Get the position of categorical columns
catColumnsPos = [df.columns.get_loc(col) for col in list(df.select_dtypes('object').columns)]
print('Categorical columns           : {}'.format(list(df.select_dtypes('object').columns)))
print('Categorical columns position  : {}'.format(catColumnsPos))
```

```
Categorical columns           : ['Region', 'Item Type', 'Sales Channel', 'Order Priority']
Categorical columns position  : [0, 1, 2, 3]
```

Penting! Dalam kasus nyata, kolom numerik akan memiliki skala yang berbeda sehingga Anda harus menormalkannya menggunakan teknik yang sesuai, seperti normalisasi min-max, Normalisasi Z-Score, dll.

Selanjutnya, konversikan data dari data frame ke matriks. Ini membantu modul kmodes menjalankan algoritma pengelompokan K-Prototype. Simpan matriks data ke variabel **dfMatrix**.

13. Convert dataframe to matrix

```
dfMatrix = df.to_numpy()
```

```
dfMatrix
```

```
array([['Sub-Saharan Africa', 'Office Supplies', 'Online', ...,
        2920025.64, 2353920.64, 566105.0],
       ['Europe', 'Beverages', 'Online', ..., 51008.75, 34174.25,
        16834.5],
       ['Middle East and North Africa', 'Vegetables', 'Offline', ...,
        1003700.9, 592408.95, 411291.95],
       ...,
       ['Sub-Saharan Africa', 'Vegetables', 'Offline', ..., 388847.44,
        229507.32, 159340.12],
       ['Sub-Saharan Africa', 'Meat', 'Online', ..., 3672974.34,
        3174991.14, 497983.2],
       ['Asia', 'Snacks', 'Offline', ..., 55081.38, 35175.84, 19905.54]],
      dtype=object)
```

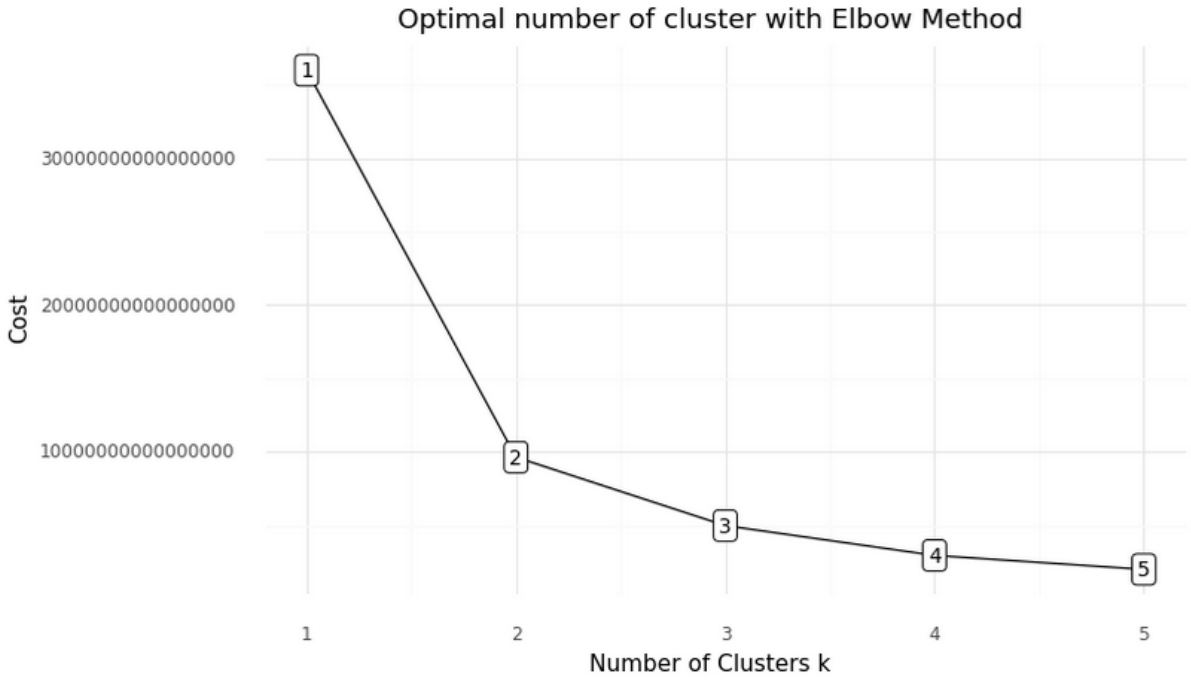
Kita menggunakan metode Elbow untuk menentukan jumlah cluster yang optimal untuk cluster K-Prototype. Alih-alih menghitung kesalahan dalam jumlah kuadrat (WSSE) dengan jarak Euclidian, K-Prototype menyediakan fungsi biaya yang menggabungkan perhitungan untuk variabel numerik dan kategoris. Kita dapat melihat ke dalam Elbow untuk menentukan jumlah cluster yang optimal.

14. Choose optima K using Elbow method

```
# Choosing optimal K
cost = []
for cluster in range(1, 10):
    try:
        kprototype = KPrototypes(n_jobs = -1, n_clusters = cluster, init = 'Huang', random_state = 0)
        kprototype.fit_predict(dfMatrix, categorical = catColumnsPos)
        cost.append(kprototype.cost_)
        print('Cluster initiation: {}'.format(cluster))
    except:
        break
```

```
# Converting the results into a dataframe and plotting them
df_cost = pd.DataFrame({'Cluster':range(1, 6), 'Cost':cost})
df_cost.head()
```

```
#Data Visualization
plotnine.options.figure_size = (8, 4.8)
(
    ggplot(data = df_cost)+
    geom_line(aes(x = 'Cluster',
                  y = 'Cost'))+
    geom_point(aes(x = 'Cluster',
                   y = 'Cost'))+
    geom_label(aes(x = 'Cluster',
                   y = 'Cost',
                   label = 'Cluster'),
               size = 10,
               nudge_y = 1000) +
    labs(title = 'Optimal number of cluster with Elbow Method')+
    xlab('Number of Clusters k')+
    ylab('Cost')+
    theme_minimal()
)
```



Penting! Berdasarkan scree plot dari fungsi biaya di atas, kita mempertimbangkan untuk memilih jumlah cluster k = 3. Ini akan menjadi jumlah cluster yang optimal untuk analisis cluster K-Prototype.

15. Fit the cluster

```
# Fit the cluster
kprototype = KPrototypes(n_jobs = -1, n_clusters = 3, init = 'Huang', random_state = 0)
kprototype.fit_predict(dfMatrix, categorical = catColumnsPos)
```

Algoritma ini memiliki 7 iterasi untuk dikonvergensi dan memiliki biaya 4.960.713.581.025.175.0 . Kita dapat mencetak centroid dari cluster menggunakan **kprototype.cluster_centroids_.** Untuk variabel numerik akan menggunakan rata-rata sedangkan kategorik menggunakan modus.

16. Cluster centroid

```
# Cluster centorid
kprototype.cluster_centroids_
```

```
[array([[7.90436555e+03, 5.93526513e+02, 4.57785496e+02, 4.62276055e+06,
         3.55912112e+06, 1.06363942e+06],
        [4.04666949e+03, 1.63259107e+02, 1.05796869e+02, 4.67709452e+05,
         2.81142955e+05, 1.86566497e+05],
        [6.09327542e+03, 3.84264504e+02, 2.75097501e+02, 1.99588812e+06,
         1.38053953e+06, 6.15348596e+05]]),
 array([['Europe', 'Household', 'Offline', 'L'],
        ['Sub-Saharan Africa', 'Personal Care', 'Online', 'C'],
        ['Europe', 'Cosmetics', 'Online', 'H']], dtype='<U18')]
```

17. Check the iteration of the clusters created

```
# Check the iteration of the clusters created
kprototype.n_iter_
```

18. Check the cost of the clusters created

```
# Check the cost of the clusters created
kprototype.cost_
```

Penafsiran cluster diperlukan. Interpretasinya menggunakan centroid pada setiap cluster. Untuk melakukannya, kita perlu menambahkan label cluster ke data mentah. Urutan label cluster akan sangat membantu untuk mengatur interpretasi berdasarkan label cluster.

19. Add the cluster to the dataframe

```
# Add the cluster to the dataframe
df['cluster_id'] = kprototype.labels_
```

```
df_region = pd.DataFrame(df['Region'].value_counts()).reset_index()
df_region['Percentage'] = df_region['Region'] / df['Region'].value_counts().sum()
df_region.rename(columns = {'index':'Region', 'Region':'Total'}, inplace = True)
df_region = df_region.sort_values('Total', ascending = True).reset_index(drop = True)
df_region
```

```
# Add the cluster to the dataframe
df['Cluster Labels'] = kprototype.labels_
df['Segment'] = df['Cluster Labels'].map({0:'First', 1:'Second', 2:'Third'})
# Order the cluster
df['Segment'] = df['Segment'].astype('category')
df['Segment'] = df['Segment'].cat.reorder_categories(['First','Second','Third'])
df.head()
```

	Region	Item Type	Sales Channel	Order Priority	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit	Cluster Labels	Segment
0	Sub-Saharan Africa	Office Supplies	Online	L	4484	651.21	524.96	2920025.64	2353920.64	566105.00	2	Third
1	Europe	Beverages	Online	C	1075	47.45	31.79	51008.75	34174.25	16834.50	1	Second
2	Middle East and North Africa	Vegetables	Offline	C	6515	154.06	90.93	1003700.90	592408.95	411291.95	1	Second
3	Sub-Saharan Africa	Household	Online	C	7683	668.27	502.54	5134318.41	3861014.82	1273303.59	0	First
4	Europe	Beverages	Online	C	3491	47.45	31.79	165647.95	110978.89	54669.06	1	Second

Untuk menginterpretasikan cluster, untuk variabel numerik akan menggunakan rata-rata sedangkan kategorik menggunakan mode. Tetapi ada metode lain yang dapat diterapkan seperti menggunakan median, persentil, atau komposisi nilai untuk variabel kategori.

20. Cluster interpretation

```
# Cluster interpretation
df.rename(columns = {'Cluster Labels':'Total'}, inplace = True)
df.groupby('Segment').agg(
    {
        'Total':'count',
        'Region': lambda x: x.value_counts().index[0],
        'Item Type': lambda x: x.value_counts().index[0],
        'Sales Channel': lambda x: x.value_counts().index[0],
        'Order Priority': lambda x: x.value_counts().index[0],
        'Units Sold': 'mean',
        'Unit Price': 'mean',
        'Total Revenue': 'mean',
        'Total Cost': 'mean',
        'Total Profit': 'mean'
    }
).reset_index()
```

	Segment	Total	Region	Item Type	Sales Channel	Order Priority	Units Sold	Unit Price	Total Revenue	Total Cost	Total Profit
0	First	1190	Europe	Household	Offline	L	7904.365546	593.526513	4.622761e+06	3.559121e+06	1.063639e+06
1	Second	6381	Sub-Saharan Africa	Personal Care	Online	C	4046.669488	163.259107	4.677095e+05	2.811430e+05	1.865665e+05
2	Third	2429	Europe	Cosmetics	Online	H	6093.275422	384.264504	1.995888e+06	1.380540e+06	6.153486e+05

Kesimpulan

K-Prototype adalah algoritma clustering yang merupakan kombinasi dari K-Means dan K-Mode yang dikembangkan oleh Huang. Untuk implementasi algoritmanya, peneliti perlu menyaring kolom dengan hati-hati terutama untuk variabel kategoris. Variabel kategoris harus relevan dengan analisis yang bukan informasi yang tidak berarti. Selain itu, kualitas input (data) mempengaruhi hasil pengelompokan (inisialisasi cluster) dan bagaimana algoritma memproses data untuk mendapatkan hasil yang konvergen. Sebagai peneliti, untuk tugas akhir, interpretasi, kita perlu mempertimbangkan metrik yang akan digunakan untuk variabel numerik dan kategorik.

K-Modes Clustering

Kita akan mengerjakan algoritma K-Modes Clustering, untuk mengelompokkan tugas pengelompokan variabel kategorikal tunggal.

1. Import library

   Seperti biasa, mari kita impor library terlebih dahulu:

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

2. Sekarang, mari kita gunakan dataset dummy dan lihat bagaimana jadinya:

```
data = pd.read_csv('data.csv')
data
```

	Individual	Skill
0	1	C
1	1	Python
2	1	Java
3	1	Haskell
4	2	Python
5	2	React
6	2	JS
7	2	PHP
8	3	C++
9	3	JS
10	3	Flutter

3. Kita akan memberikan tanda kepada individual yang memiliki skill dengan cara :

```
one_hot_df = data.copy()
for i,name in enumerate(data['Skill'].unique()):
    one_hot_df[name] = 0
def set_product(x):
    x[str(x['Skill'])] = 1
    return x

one_hot_df = one_hot_df.apply(set_product, axis=1)
one_hot_df = one_hot_df.groupby(['Individual']).sum()
one_hot_df
```

Individual	C	Python	Java	Haskell	React	JS	PHP	C++	Flutter	Android	iOS	Fortran	Pascal	.NET	C#	MATLAB
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

4. Lalu kita coba menggunakan 3 culster dengan initial Huang untuk memprediksi mendapat K-Modes :

```python
from kmodes.kmodes import KModes
# define the k-modes model
km = KModes(n_clusters=3, init='Huang', n_init=4, verbose=1)
# fit the clusters to the skills dataframe
clusters = km.fit_predict(one_hot_df)
# get an array of cluster modes
kmodes = km.cluster_centroids_
shape = kmodes.shape
# For each cluster mode (a vector of "1" and "0")
# find and print the column headings where "1" appears.
# If no "1" appears, assign to "no-styles" cluster.
for i in range(shape[0]):
    if sum(kmodes[i,:]) == 0:
        print("\ncluster " + str(i) + ": ")
        print("no-style cluster")
    else:
        print("\ncluster " + str(i) + ": ")
        cent = kmodes[i,:]
        for j in one_hot_df.columns[np.nonzero(cent)]:
            print(j)
```

Setelah dijalankan :

```
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 0, cost: 8.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 2, iteration: 1/100, moves: 0, cost: 8.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 3, iteration: 1/100, moves: 0, cost: 8.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 4, iteration: 1/100, moves: 0, cost: 7.0
Best run was number 4
```

```
cluster 0:
JS

cluster 1:
C
Python
Java
Haskell

cluster 2:
C
Fortran
Pascal
.NET
C#
MATLAB
```

PENGUMPULAN

1. File yang dikumpulkan terdiri dari:
    a. File project (.ipynb)
    b. File PDF berisi screenshot output dan jawaban (jika ada pertanyaan)
2. File di-compress (.zip) dan diberi nama **KODEMATAKULIAH_KELAS_NIM_NAMA_WEEK KE-XX.zip** (contoh: IS5411_A_13110310017_Monika Evelin Johan_Week-01.zip).

REFERENSI

Z. Huang. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values (1998). Data Mining and Knowledge Discovery. 2(3): 283–304.