International Journal of Education and Life Sciences (IJELS)
Vol. 3, No. 1, 2025 : 1643-1654

Application of the K-Means Clustering Algorithm for Sales Analysis in a Padang Restaurant Business

Arrafi Jehansyah Sobari Al'Ayubi[1], Aswan Supriyadi Sunge[2], Suherman[3]
Universitas Pelita Bangsa

Corresponding Author : Arrafi Jehansyah Sobari Al'Ayubi: arrafijehansyah@gmail.com

A R T I C L E I N F O

Keywords : K-Means Clustering, Padang Restaurant

Received : 25, November
Revised : 20, December
Accepted: 12, Januari

A B S T R A C T

This study aims to apply the K-Means Clustering algorithm to analyze sales data and categorize products based on their sales performance, namely best-selling, moderately-selling, and least-selling products. The data used in this research comes from a Padang restaurant in Deltamas in 2022 to 2023. The research process began with data collection, followed by data cleaning and normalization using the Min-Max method, and then the application of the K-Means algorithm for clustering. The results show that the K-Means algorithm successfully grouped the products into three categories: Cluster 0 (best-selling products), Cluster 1 (moderately-selling products), and Cluster 2 (least-selling products). Thus, this study demonstrates that the K-Means algorithm can be used to cluster sales data and assist business owners in managing product inventory and making more informed decisions.

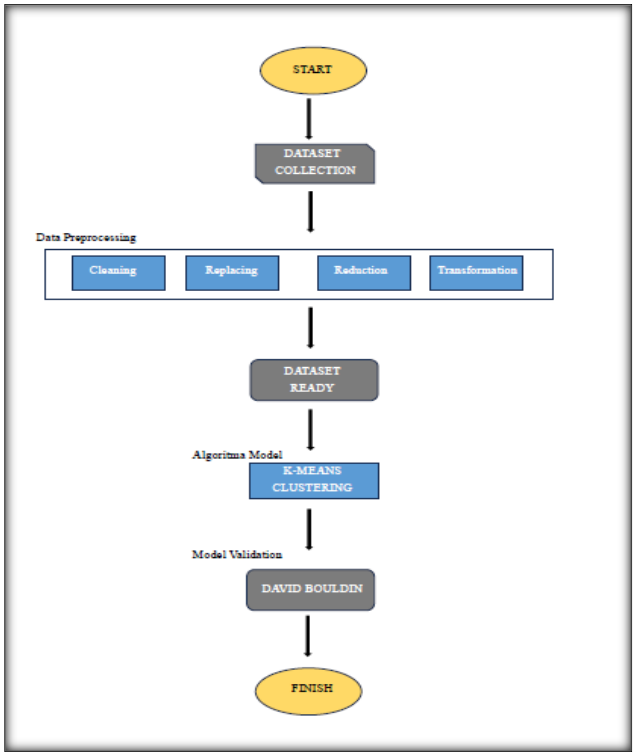
INTRODUCTION

In the era of rapidly advancing information technology, information systems have become one of the key factors supporting organizations or companies in facilitating their operational activities, from data input and processing to providing reports and other essential information. For business or trade practitioners, sales information systems play a crucial role in simplifying transaction record-keeping, inventory management, customer service, and generating sales reports. These reports can then be analyzed and used to improve marketing strategies and sales performance. For large and medium-sized companies with extensive customer databases and a wide range of products, techniques or methods are required to extract information and knowledge that can enhance competitive advantage. With data mining and its various algorithms, businesses can identify consumer purchasing trends, recognize sales patterns, and analyze the market (through market basket analysis). This data becomes valuable information or knowledge that can support informed decision-making to ensure the continued success of a business (Lestari, 2017).

The culinary business is considered one of the economic ventures with promising prospects, even during times of crisis. This is because food is a basic human need essential to daily life. The culinary industry has many categories, from snacks and beverages to staple foods, which have great potential depending on how well they are marketed. However, many businesses fail due to their inability to properly manage data and track business developments. In the restaurant business, managing stock is critical, yet many restaurant owners still rely on estimates for stock procurement. They want to avoid running out of a popular dish if customer demand spikes. As a result, business owners often opt to stock up on certain dishes more than necessary. While this can prevent stock shortages, it can also lead to excess inventory of less popular items. This can result in waste if the unsold dishes spoil, as the ingredients used may have limited shelf life. Observations have shown that many restaurants face issues with a shortage of popular dishes and an overstock of less popular ones, leading to potential food waste or loss of sales (Yang, S., Pan, L., & Liu, S, 2019).

METHODOLOGY

The method used in this study begins with a literature review, followed by data collection, data preprocessing, algorithm implementation, evaluation, and testing. The data used in this research comes from the sales records of a Padang restaurant owned by Mr. Hamdan, located at Jl. Deltamas Boulevard No. 18 Blok. H, Kota Deltamas, RT005/RW003, Desa Sukamahi, Kecamatan Cikarang Pusat, Kabupaten Bekasi, West Java, 17530.



Data Collection

The first step in this research begins with selecting the dataset to be used. The dataset chosen for this study is the sales data from Mr. Hamdan's Padang restaurant, located at the address mentioned above. The data collection process was carried out from 2022 to 2023, covering all sales data within that period. The data collection was conducted based on the research requirements, aiming to obtain relevant, accurate, and reliable data.

Preprocessing/Cleaning

In this study, the data was first collected and analyzed, followed by the preprocessing stage, which includes data cleaning and replacing missing values. After the preprocessing stage, the cleaned and corrected data is ready to be processed further in the next stages.

Algorithm Implementation

In this research, the K-Means algorithm was used. In this stage, the K-Means algorithm was applied to cluster the data into three distinct clusters.

Evaluation

The final section describes the evaluation and validation of the results obtained from applying the research methods.

Testing

The testing method was conducted to evaluate the accuracy level of each algorithm. This testing process was carried out using Python programming

language and operated on Google Colab, which was used to implement the K-Means algorithm and obtain the automatic clustering results for each algorithm.

RESEARCH RESULT AND DISCUSSION

This study employs the K-Means Clustering algorithm, implemented using the Python programming language in the Google Colab environment, to classify sales data into distinct categories based on product sales performance. The goal of the analysis is to group products into three primary categories: fast-selling products, moderately-selling products, and slow-selling products. To achieve this, the algorithm identifies patterns and clusters in the data, which are then used to make informed decisions about inventory management and sales strategies. The dataset utilized in this research consists of 20 different products, with sales data gathered over the course of one year. By applying unsupervised machine learning techniques, the study aims to provide actionable insights into sales trends and assist businesses in optimizing their product offerings and marketing strategies.

Picture 1. Import Libray command

```
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import metrics
from pandas import read_csv
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_samples, silhouette_score
import pandas as pd
from sklearn.model_selection import cross_val_score
from sklearn.exceptions import ConvergenceWarning
import warnings
import numpy as np
```

After inputting the required data, the next step involved the researcher creating several commands, such as performing calculations, loading libraries using the K-Means algorithm, displaying visualizations, conducting data analysis, calculating metrics, and other commands to support this research.

The next step involved the researcher creating commands to ensure that the Excel-formatted data could be read in Google Colab and then displayed within the platform.

Picture 2. Dataset

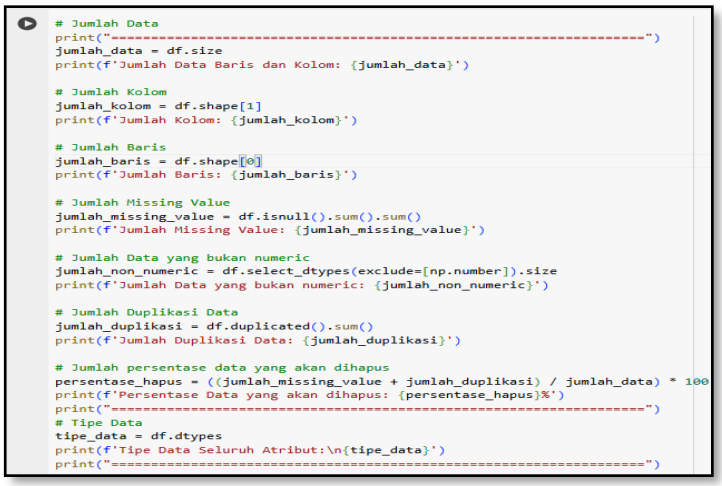
At this stage, the researcher created commands to analyze the imported data in order to understand its details and gain insights from it.

```
[ ] # 1. Baca dataset
    df = pd.read_excel('/content/drive/MyDrive/Clustering Penjualan/Data Transaksi Penjualan.xlsx')
    df
```

	Menu	Minggu 1	Minggu 2	Minggu 3	Minggu 4	Minggu 5	Minggu 6	Minggu 7	Minggu 8	Minggu 9	...	Minggu 41	Minggu 42	Minggu 43	Minggu 44	Minggu 45	Minggu 46	Minggu 47	Minggu 48	Minggu 49	Minggu 50
0	Dendeng Balado	4	6	9	14	10	7	11	16	6	...	16	19	20	17	4	7	22	6	7	21
1	Ayam Goreng	15	8	8	21	28	19	17	11	9	...	10	11	6	9	9	17	20	14	5	12
2	Ayam Gulai	25	7	9	13	15	15	17	8	19	...	7	7	13	12	17	10	11	16	9	8
3	Daging Cincang	3	5	6	11	9	9	18	2	3	...	7	13	13	5	17	10	12	8	7	6
4	Rendang	15	10	10	23	21	19	17	17	9	...	23	29	24	19	11	7	4	5	8	12
5	Ayam Pop	11	6	7	7	9	10	4	17	7	...	5	10	11	8	13	9	15	6	9	7
6	Ayam Bakar	22	18	9	9	11	26	7	27	16	...	7	9	8	18	11	11	14	37	23	25
7	Telur Dadar	12	9	10	25	23	20	17	6	8	...	24	17	13	15	9	8	8	7	10	23
8	Ikan Asam Pedas	4	5	1	0	1	3	8	11	9	...	6	4	3	9	2	0	4	3	3	1
9	Ikan Balado	9	8	2	3	11	8	8	12	2	...	4	3	5	5	7	8	6	2	3	2
10	Perkedel	4	8	7	6	21	5	20	13	22	...	9	4	10	7	7	16	17	26	6	8
11	Paru Kering	2	2	5	3	0	1	2	5	4	...	0	3	5	11	9	2	4	5	5	10
12	Ikan Bakar	4	4	3	0	7	1	11	2	6	...	0	2	0	4	5	5	7	7	2	3
13	Gulai Kakap	2	3	3	4	7	10	1	1	0	...	4	4	0	1	8	0	5	2	1	4
14	Kikil Tunjang	2	5	8	8	11	15	12	2	3	...	9	8	4	7	5	12	0	5	7	9
15	Udang Saus Padang	3	5	8	10	10	8	9	5	6	...	9	10	4	8	3	7	6	9	5	8
16	Hati Sapi	2	5	4	6	2	4	7	3	5	...	2	6	8	6	6	5	6	5	5	8
17	Gulai Ikan	7	9	8	6	10	7	7	9	5	...	8	5	6	7	5	3	3	6	2	4
18	Peyek Udang	5	7	5	6	4	8	6	1	5	...	4	5	2	6	3	7	4	5	2	4
19	Gulai Cumi	8	6	8	7	5	9	6	7	2	...	6	7	6	5	7	7	4	3	4	2

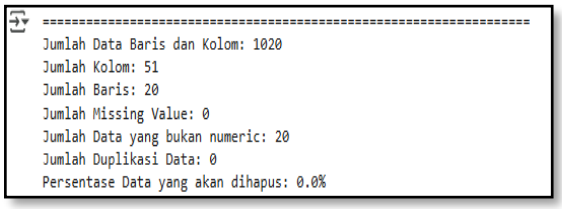
20 rows × 51 columns

Picture 3. Data Analysis Commands



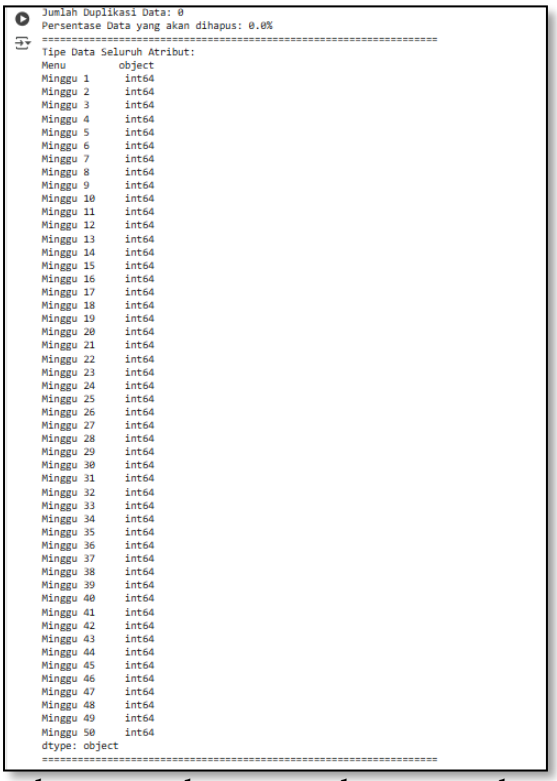
```
# Jumlah Data
print("=====================================================================")
jumlah_data = df.size
print(f'Jumlah Data Baris dan Kolom: {jumlah_data}')

# Jumlah Kolom
jumlah_kolom = df.shape[1]
print(f'Jumlah Kolom: {jumlah_kolom}')

# Jumlah Baris
jumlah_baris = df.shape[0]
print(f'Jumlah Baris: {jumlah_baris}')

# Jumlah Missing Value
jumlah_missing_value = df.isnull().sum().sum()
print(f'Jumlah Missing Value: {jumlah_missing_value}')

# Jumlah Data yang bukan numeric
jumlah_non_numeric = df.select_dtypes(exclude=[np.number]).size
print(f'Jumlah Data yang bukan numeric: {jumlah_non_numeric}')

# Jumlah Duplikasi Data
jumlah_duplikasi = df.duplicated().sum()
print(f'Jumlah Duplikasi Data: {jumlah_duplikasi}')

# Jumlah persentase data yang akan dihapus
persentase_hapus = ((jumlah_missing_value + jumlah_duplikasi) / jumlah_data) * 100
print(f'Persentase Data yang akan dihapus: {persentase_hapus}%')
print("=====================================================================")
# Tipe Data
tipe_data = df.dtypes
print(f'Tipe Data Seluruh Atribut:\n{tipe_data}')
print("=====================================================================")
```

Picture 4. Data analysis results



```
=====================================================================
Jumlah Data Baris dan Kolom: 1020
Jumlah Kolom: 51
Jumlah Baris: 20
Jumlah Missing Value: 0
Jumlah Data yang bukan numeric: 20
Jumlah Duplikasi Data: 0
Persentase Data yang akan dihapus: 0.0%
```

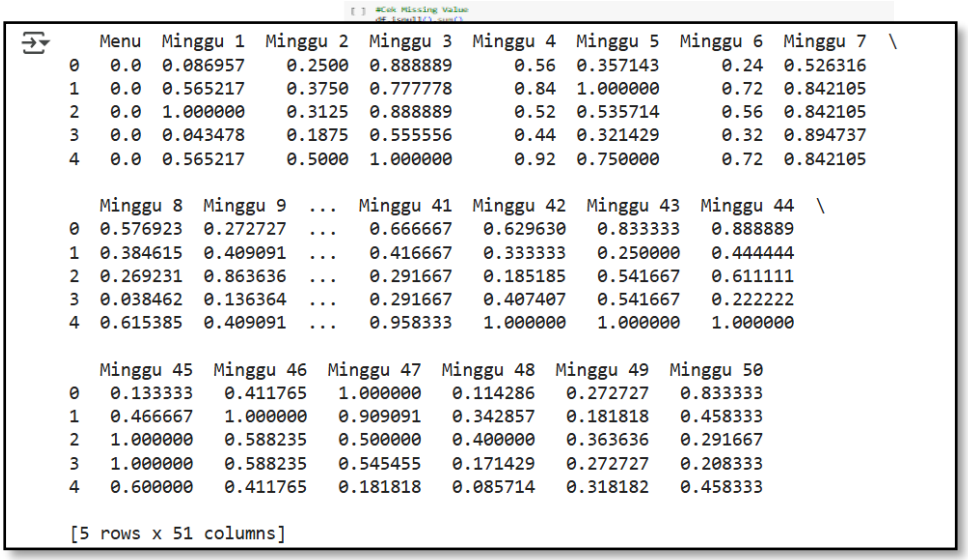
Picture 5. Data analysis results



```
Jumlah Duplikasi Data: 0
Persentase Data yang akan dihapus: 0.0%
=====================================================================
Tipe Data Seluruh Atribut:
Menu          object
Minggu 1       int64
Minggu 2       int64
Minggu 3       int64
Minggu 4       int64
Minggu 5       int64
Minggu 6       int64
Minggu 7       int64
Minggu 8       int64
Minggu 9       int64
Minggu 10      int64
Minggu 11      int64
Minggu 12      int64
Minggu 13      int64
Minggu 14      int64
Minggu 15      int64
Minggu 16      int64
Minggu 17      int64
Minggu 18      int64
Minggu 19      int64
Minggu 20      int64
Minggu 21      int64
Minggu 22      int64
Minggu 23      int64
Minggu 24      int64
Minggu 25      int64
Minggu 26      int64
Minggu 27      int64
Minggu 28      int64
Minggu 29      int64
Minggu 30      int64
Minggu 31      int64
Minggu 32      int64
Minggu 33      int64
Minggu 34      int64
Minggu 35      int64
Minggu 36      int64
Minggu 37      int64
Minggu 38      int64
Minggu 39      int64
Minggu 40      int64
Minggu 41      int64
Minggu 42      int64
Minggu 43      int64
Minggu 44      int64
Minggu 45      int64
Minggu 46      int64
Minggu 47      int64
Minggu 48      int64
Minggu 49      int64
Minggu 50      int64
dtype: object
=====================================================================
```

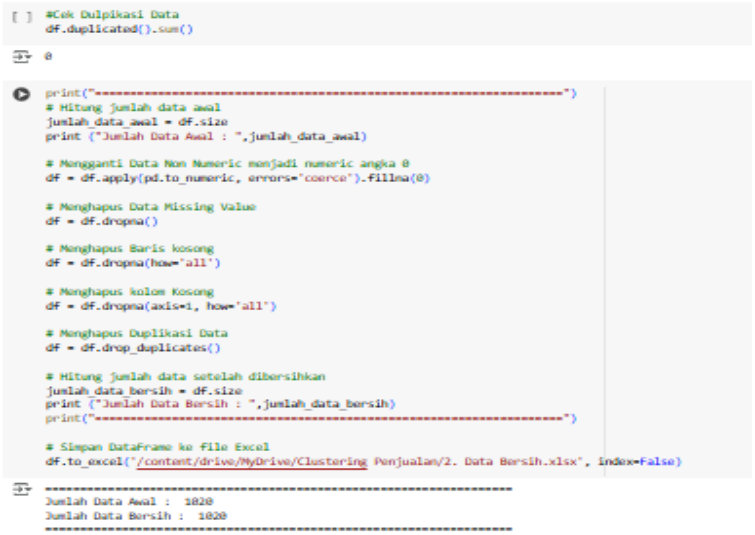
At this stage, the researcher created commands to correct or remove incorrect, corrupted, misformatted, duplicate, or incomplete data within the dataset.

Picture 6. *Cleaning Data*



```
   Menu  Minggu 1  Minggu 2  Minggu 3  Minggu 4  Minggu 5  Minggu 6  Minggu 7  \
0   0.0  0.086957    0.2500  0.888889      0.56  0.357143      0.24  0.526316
1   0.0  0.565217    0.3750  0.777778      0.84  1.000000      0.72  0.842105
2   0.0  1.000000    0.3125  0.888889      0.52  0.535714      0.56  0.842105
3   0.0  0.043478    0.1875  0.555556      0.44  0.321429      0.32  0.894737
4   0.0  0.565217    0.5000  1.000000      0.92  0.750000      0.72  0.842105

   Minggu 8  Minggu 9  ...  Minggu 41  Minggu 42  Minggu 43  Minggu 44  \
0  0.576923  0.272727  ...   0.666667   0.629630   0.833333   0.888889
1  0.384615  0.409091  ...   0.416667   0.333333   0.250000   0.444444
2  0.269231  0.863636  ...   0.291667   0.185185   0.541667   0.611111
3  0.038462  0.136364  ...   0.291667   0.407407   0.541667   0.222222
4  0.615385  0.409091  ...   0.958333   1.000000   1.000000   1.000000

   Minggu 45  Minggu 46  Minggu 47  Minggu 48  Minggu 49  Minggu 50
0   0.133333   0.411765   1.000000   0.114286   0.272727   0.833333
1   0.466667   1.000000   0.909091   0.342857   0.181818   0.458333
2   1.000000   0.588235   0.500000   0.400000   0.363636   0.291667
3   1.000000   0.588235   0.545455   0.171429   0.272727   0.208333
4   0.600000   0.411765   0.181818   0.085714   0.318182   0.458333

[5 rows x 51 columns]
```

Picture 7. *Cleaning Data*



```
#Cek Dulpikasi Data
df.duplicated().sum()
```

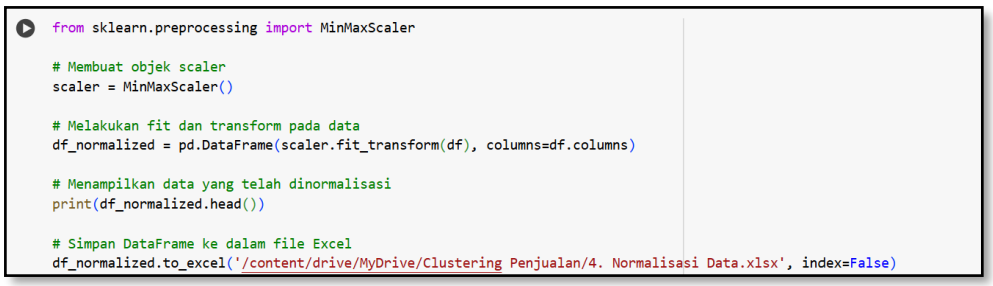
```
0
```

```
print("=============================================================")
# Hitung jumlah data awal
jumlah_data_awal = df.size
print ("Jumlah Data Awal : ",jumlah_data_awal)

# Mengganti Data Non Numeric menjadi numeric angka 0
df = df.apply(pd.to_numeric, errors='coerce').fillna(0)

# Menghapus Data Missing Value
df = df.dropna()

# Menghapus Baris kosong
df = df.dropna(how='all')

# Menghapus kolom Kosong
df = df.dropna(axis=1, how='all')

# Menghapus Duplikasi Data
df = df.drop_duplicates()

# Hitung jumlah data setelah dibersihkan
jumlah_data_bersih = df.size
print ("Jumlah Data Bersih : ",jumlah_data_bersih)
print("=============================================================")

# Simpan DataFrame ke File Excel
df.to_excel('/content/drive/MyDrive/Clustering Penjualan/2. Data Bersih.xlsx', index=False)
```

```
=============================================================
Jumlah Data Awal :  1020
Jumlah Data Bersih :  1020
=============================================================
```

The next step involved the researcher creating commands to perform data normalization using the Min-Max method.

Picture 8. Data normalization command



```
from sklearn.preprocessing import MinMaxScaler

# Membuat objek scaler
scaler = MinMaxScaler()

# Melakukan fit dan transform pada data
df_normalized = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)

# Menampilkan data yang telah dinormalisasi
print(df_normalized.head())

# Simpan DataFrame ke dalam file Excel
df_normalized.to_excel('/content/drive/MyDrive/Clustering Penjualan/4. Normalisasi Data.xlsx', index=False)
```

Picture 9. Data normalization command

At this stage, the researcher created commands for the K-Means algorithm to implement it in this study using Python programming language. The researcher also created commands to compute data from week 1 to week 50 based on the attributes provided.

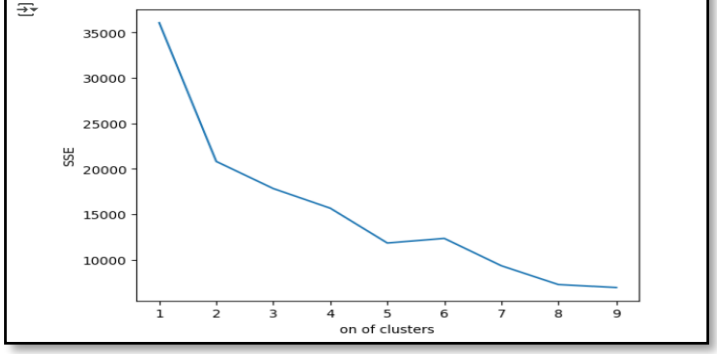
Picture 10. K-means algorithm

```
[ ]  #Membuat Fungsi K-Means
     import pandas as pd
     from sklearn.cluster import KMeans
     from sklearn.preprocessing import StandardScaler
     from sklearn.metrics import silhouette_score
     import matplotlib.pyplot as plt
     import seaborn as sns
     from scipy.spatial.distance import cdist
     import numpy as np
     import warnings
     warnings.simplefilter(action='ignore', category=FutureWarning)

[ ]  #Data yang di komputasi

     X = df.loc[:, ["Minggu 1", "Minggu 2", "Minggu 3", "Minggu 4", "Minggu 5", "Minggu 6", "Minggu 7", "Minggu 8", "Minggu 9", "Minggu 10",
```

At this stage, the researcher created commands to determine the optimal number of clusters using the Elbow method and displayed the corresponding visualization.

Picture 11. Elbow Methode

```
#Membuat Elbow Method
sse={}
for k in range(1,10):
    kmeans = KMeans(n_clusters= k, max_iter=1000).fit (X)
    sse[k] = kmeans.inertia_
plt.figure()
plt.plot(list(sse.keys()), list(sse.values()))
plt.xlabel('on of clusters')
plt.ylabel('SSE')
plt.show()
```



At this stage, the researcher created commands to calculate the centroid centers of the K-Means algorithm.

Picture 12. Centroid center

```
[ ]  #Pusat centoid K-Means
     kmeans = KMeans (n_clusters=3, random_state=0)
     kmeans.fit (X)
     pusat = kmeans.cluster_centers_
     print(pusat)

[[17.         11.          8.33333333  9.33333333 15.66666667 15.33333333
  14.66666667 16.         19.         18.66666667 15.33333333 15.66666667
  15.66666667 21.66666667 11.66666667 19.         19.66666667 18.33333333
  17.66666667 20.          7.          5.33333333  9.33333333 12.33333333
   6.         12.66666667 10.66666667 14.         16.33333333 21.33333333
  11.33333333 11.33333333 13.33333333 11.         13.33333333 14.
  17.33333333 17.33333333 21.         14.66666667  7.66666667  6.66666667
  10.33333333 12.33333333 11.66666667 12.33333333 14.         26.33333333
  12.66666667 13.66666667]
 [ 4.71428571  5.42857143  5.5         6.07142857  6.85714286  7.14285714
   7.85714286  6.64285714  4.5         6.42857143  6.64285714  6.07142857
   5.42857143  4.85714286  5.64285714  5.71428571  5.28571429  5.64285714
   6.78571429  6.21428571  5.57142857  6.07142857  5.14285714  5.35714286
   6.14285714  5.78571429  6.          6.28571429  6.          5.71428571
   8.07142857  5.28571429  5.28571429  8.28571429  6.28571429  5.85714286
   6.14285714  6.71428571  4.64285714  4.42857143  5.71428571  7.07142857
   6.21428571  7.07142857  6.71428571  5.85714286  7.          5.14285714
   4.42857143  6.35714286]
 [14.          9.          9.33333333 23.         24.         19.33333333
  17.         11.33333333  8.66666667  6.66666667 13.66666667 11.33333333
   7.66666667 18.33333333 16.33333333 13.33333333  9.33333333 11.
  18.66666667 13.33333333 12.33333333  9.66666667 11.66666667 15.66666667
  17.66666667 22.33333333 16.         12.33333333 11.66666667 25.66666667
  15.66666667 12.66666667 12.33333333  9.33333333 11.66666667 16.
   8.33333333 11.66666667 11.66666667 17.         19.         19.
  14.33333333 14.33333333  9.66666667 10.66666667 10.66666667  8.66666667
   7.66666667 15.66666667]]
```

At this stage, the researcher created commands to label all the existing clusters, which included Cluster 0, Cluster 1, and Cluster 2.

Picture 13. cluster labeling

```
#Menampilkan jumlah cluster
df['cluster'] = kmeans.labels_
df.head
print('Jumlah Anggota Setiap Cluster')
print('Cluster 0(laris) = ', df[df['cluster'] == 0] ['cluster'].count())
print('Cluster 1(cukup laris) = ', df[df['cluster'] == 1] ['cluster'].count())
print('Cluster 2(kurang laris) = ', df[df['cluster'] == 2] ['cluster'].count())
```

```
Jumlah Anggota Setiap Cluster
Cluster 0(laris) =  3
Cluster 1(cukup laris) =  14
Cluster 2(kurang laris) =  3
```

Picture 14. Cluster column

In the image above, the researcher created commands to assign a column to each cluster with the corresponding data.

Picture 15. Display of the number of clusters

```
#Membuat label cluster
label = kmeans.labels_
print(label)
```

```
[1 2 0 1 2 1 0 2 1 1 0 1 1 1 1 1 1 1 1 1]
```

In the image above, the researcher created commands to display the number of members in each existing cluster.

Picture 16. Display of mean value

```
#Membuat kolom untuk label cluster
df.insert (0, 'cluster', kmeans.labels_)
df
```

	cluster	Menu	Minggu 1	Minggu 2	Minggu 3	Minggu 4	Minggu 5	Minggu 6	Minggu 7	Minggu 8	...	Minggu 41	Minggu 42	Minggu 43	Minggu 44	Minggu 45	Minggu 46	Minggu 47	Minggu 48	Minggu 49	Minggu 50
0	1	0.0	4	6	9	14	10	7	11	16	...	16	19	20	17	4	7	22	6	7	21
1	2	0.0	15	8	8	21	28	19	17	11	...	10	11	6	9	9	17	20	14	5	12
2	0	0.0	25	7	9	13	15	15	17	8	...	7	7	13	12	17	10	11	16	9	8
3	1	0.0	3	5	6	11	9	9	18	2	...	7	13	13	5	17	10	12	8	7	6
4	2	0.0	15	10	10	23	21	19	17	17	...	23	29	24	19	11	7	4	5	8	12
5	1	0.0	11	6	7	7	9	10	4	17	...	5	10	11	8	13	9	15	6	9	7
6	0	0.0	22	18	9	9	11	26	7	27	...	7	9	8	18	11	11	14	37	23	25
7	2	0.0	12	9	10	25	23	20	17	6	...	24	17	13	15	9	8	8	7	10	23
8	1	0.0	4	5	1	0	1	3	8	11	...	6	4	3	9	2	0	4	3	3	1
9	1	0.0	9	8	2	3	11	8	8	12	...	4	3	5	5	7	8	6	2	3	2
10	0	0.0	4	8	7	6	21	5	20	13	...	9	4	10	7	7	16	17	26	6	8
11	1	0.0	2	2	5	3	0	1	2	5	...	0	3	5	11	9	2	4	5	5	10
12	1	0.0	4	4	3	0	7	1	11	2	...	0	2	0	4	5	5	7	7	2	3
13	1	0.0	2	3	3	4	7	10	1	1	...	4	4	0	1	8	0	5	2	1	4
14	1	0.0	2	5	8	8	11	15	12	2	...	9	8	4	7	5	12	0	5	7	9
15	1	0.0	3	5	8	10	10	8	9	5	...	9	10	4	8	3	7	6	9	5	8
16	1	0.0	2	5	4	6	2	4	7	3	...	2	6	8	6	6	5	6	5	5	8
17	1	0.0	7	9	8	6	10	7	7	9	...	8	5	6	7	5	3	3	6	2	4
18	1	0.0	5	7	5	6	4	8	6	1	...	4	5	2	6	3	7	4	5	2	4
19	1	0.0	8	6	8	7	5	9	6	7	...	6	7	6	5	7	7	4	3	4	2

20 rows × 52 columns

```
final_kmeans = KMeans(n_clusters=3, random_state=0)
final_kmeans.fit(df)
df_labeled = pd.concat([df,pd.Series(final_kmeans.labels_)],axis=1)
df_labeled
```

In the image above, the researcher created commands to display the mean values of each cluster member from week 1 to week 50.

At this stage, the researcher conducted an evaluation using the Silhouette Coefficient method. This method is used to assess the optimality of the clustering results in this study, based on the highest value, which is close to 1.

Picture 17. Silhouette coefficient evaluation

```
#Menampilkan nilai mean
df.groupby('cluster')[['Minggu 1', 'Minggu 2', 'Minggu 3', 'Minggu 4', 'Minggu 5', 'Minggu 6', 'Minggu 7', 'Minggu 8', 'Minggu 9', 'Minggu 10', 'Minggu 11', 'Minggu 12', 'Minggu 13', 'Minggu 14', 'Minggu 15', 'Minggu 16', 'Minggu 17',
```

cluster	Minggu 1	Minggu 2	Minggu 3	Minggu 4	Minggu 5	Minggu 6	Minggu 7	Minggu 8	Minggu 9	Minggu 10	...	Minggu 41	Minggu 42	Minggu 43	Minggu 44	Minggu 45	Minggu 46	Minggu 47	Minggu 48	Minggu 49	Minggu 50
0	17.000000	11.000000	8.333333	9.333333	15.666667	15.333333	14.666667	16.000000	19.000000	18.666667	...	7.666667	6.666667	10.333333	12.333333	11.666667	12.333333	14.000000	26.333333	12.666667	13.666667
1	4.714286	5.428571	5.500000	6.071429	6.857143	7.142857	7.857143	6.642857	4.500000	6.428571	...	5.714286	7.071429	6.214286	7.071429	6.714286	5.857143	7.000000	5.142857	4.428571	6.357143
2	14.000000	9.000000	9.333333	23.000000	24.000000	19.333333	17.000000	11.333333	8.666667	6.666667	...	19.000000	19.000000	14.333333	14.333333	9.666667	10.666667	10.666667	8.666667	7.666667	15.666667

At this stage, the researcher created commands to display the visualization of the results from the K-Means algorithm calculations.

Picture 18. Visualization Commands

```
#Evaluasi silhoutte coefficient
silhouette_avg = metrics.silhouette_score(X, df["cluster"])
print ('sillhoutte coefficient for the above clustering = ', silhouette_avg)
```

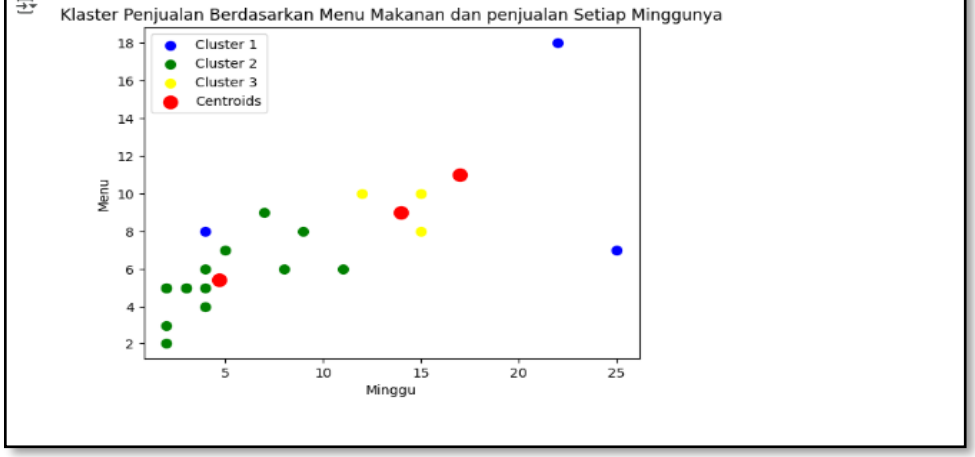
```
sillhoutte coefficient for the above clustering =  0.3741840935687887
```

Picture 19. Visualization Commands

```
[ ] centroids = final_kmeans.cluster_centers_
    centroids

array([[ 0.        ,  0.        , 17.        , 11.        ,  8.33333333,
         9.33333333, 15.66666667, 15.33333333, 14.66666667, 16.        ,
        19.        , 18.66666667, 15.33333333, 15.66666667, 15.66666667,
        21.66666667, 11.66666667, 19.        , 19.66666667, 18.33333333,
        17.66666667, 20.        ,  7.        ,  5.33333333,  9.33333333,
        12.33333333,  6.        , 12.66666667, 10.66666667, 14.        ,
        16.33333333, 21.33333333, 11.33333333, 11.33333333, 13.33333333,
        11.        , 13.33333333, 14.        , 17.33333333, 17.33333333,
        21.        , 14.66666667,  7.66666667,  6.66666667, 10.33333333,
        12.33333333, 11.66666667, 12.33333333, 14.        , 26.33333333,
        12.66666667, 13.66666667],
       [ 1.        ,  0.        ,  4.71428571,  5.42857143,  5.5       ,
         6.07142857,  6.85714286,  7.14285714,  7.85714286,  6.64285714,
         4.5       ,  6.42857143,  6.64285714,  6.07142857,  5.42857143,
         4.85714286,  5.64285714,  5.71428571,  5.28571429,  5.64285714,
         6.78571429,  6.21428571,  5.57142857,  6.07142857,  5.14285714,
         5.35714286,  6.14285714,  5.78571429,  6.        ,  6.28571429,
         6.        ,  5.71428571,  8.07142857,  5.28571429,  5.28571429,
         8.28571429,  6.28571429,  5.85714286,  6.14285714,  6.71428571,
         4.64285714,  4.42857143,  5.71428571,  7.07142857,  6.21428571,
         7.07142857,  6.71428571,  5.85714286,  7.        ,  5.14285714,
         4.42857143,  6.35714286],
       [ 2.        ,  0.        , 14.        ,  9.        ,  9.33333333,
        23.        , 24.        , 19.33333333, 17.        , 11.33333333,
         8.66666667,  6.66666667, 13.66666667, 11.33333333,  7.66666667,
        18.33333333, 16.33333333, 13.33333333,  9.33333333, 11.        ,
        18.66666667, 13.33333333, 12.33333333,  9.66666667, 11.66666667,
        15.66666667, 17.66666667, 22.33333333, 16.        , 12.33333333,
        11.66666667, 25.66666667, 15.66666667, 12.66666667, 12.33333333,
         9.33333333, 11.66666667, 16.        ,  8.33333333, 11.66666667,
        11.66666667, 17.        , 19.        , 19.        , 14.33333333,
        14.33333333,  9.66666667, 10.66666667, 10.66666667,  8.66666667,
         7.66666667, 15.66666667]])
```

Picture 20. Visualization Result

```
[ ] #Visualisasi Clustering

    plt.scatter(X[kmeans.labels_ == 0, 0], X[kmeans.labels_ == 0, 1], s = 50, c = 'blue', label = 'Cluster 1')
    plt.scatter(X[kmeans.labels_ == 1, 0], X[kmeans.labels_ == 1, 1], s = 50, c = 'green', label = 'Cluster 2')
    plt.scatter(X[kmeans.labels_ == 2, 0], X[kmeans.labels_ == 2, 2], s = 50, c = 'yellow', label = 'Cluster 3')
    plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'red', label = 'Centroids')
    plt.title('Klaster Penjualan Berdasarkan Menu Makanan dan penjualan Setiap Minggunya')
    plt.xlabel('Minggu')
    plt.ylabel('Menu')
    plt.legend()
    plt.show()
```



CONCLUSIONS

1. Based on the research conducted by the author using the K-Means clustering algorithm, calculated manually with Python programming and operated on Google Colab, supported by data collected using Microsoft Excel, it can be concluded that the K-Means clustering algorithm is highly suitable for grouping data and making more accurate decisions based on the clustered data.

2. The K-Means clustering algorithm can be used to apply data mining in the sales data information system. Based on the sales volume, the data was grouped into 3 clusters: Cluster 0 (best-selling), Cluster 1 (moderately-

selling), and Cluster 2 (least-selling), with a Silhouette Coefficient value of 0.3741840935687887.

3. This clustering method can be used to categorize sales data. Based on the manual testing results using Python programming and operated on Google Colab, the findings are as follows: Cluster 0 contains 3 items (best-selling), Cluster 1 contains 14 items (moderately-selling), and Cluster 2 contains 3 items (least-selling). This way, business owners can identify and categorize products from least to most popular.

RECOMMENDATIONS

The author acknowledges several limitations in this research. Therefore, to improve the quality of future research, the author provides the following suggestions:

1. Future researchers are advised to try different algorithms in order to make comparisons and observe the differences in the accuracy levels of the results.
2. It is recommended to increase the dataset size to improve the accuracy of the research results.
3. Future researchers may also consider adding additional supporting systems to enrich the research findings.

With these suggestions, it is hoped that future research will yield more optimal and beneficial results.

REFERENCES

Dewi, A. O. P. (2020). Big Data di Perpustakaan dengan Memanfaatkan Data Mining. *Anuva: Jurnal Kajian Budaya, Perpustakaan, Dan Informasi*, *4*(2), 223–230. https://doi.org/10.14710/anuva.4.2.223-230

Dodi Nofri Yoliad. (2023). Data mining Dalam Analisis Tingkat Penjualan Barang ElektronikMenggunakan Algoritma K-means. *Insearch (Information System Research) Journal*, *3*(1).

Haditsah Annur. (2019). Penerapan Data Mining Menentukan Strategi Penjualan Variasi Mobil Menggunakan Metode K-Means Clustering (Studi Kasus Toko Luxor Variasi Gorontalo). *Jurnal Informatika Upgris*, *5*(1), 40–45.

Hakim, Muhammad, L. (2021). Pengelompokan Data Transaksi Penjualan Menggunakan Algoritma Clustering K-Means Pada Toko an Nur. *Nuevos Sistemas de Comunicación e Información*, 2013–2015.

Kurniawati, I., Muhidin, A., & Eko, F. P. (2024). Klasifikasi Perpanjangan Kontrak Karyawan pada PTMilpo Menggunakan Algoritma Naive bayes. *Riset Dan E-Jurnal Manajemen Informatika Komputer*, *8*, 1–11.

Lubis, W., Hamid, A., Murroh, A., Syekh Ali Hasan Ahmmad Addary Padangsidimpuan, U., Rizal Nurdin, J. T., & -Kota Padangsidimpuan, S. (2022). Strategi Pemasaran Rumah Makan Dalam Meningkatkan Konsumen (Studi Rumah Makan Porang Goti). *Profjes*, *01*(01), 286–298.

Marisa, F. (2013). Educational Data Mining (Konsep Dan Penerapan). *Jurnal Teknologi Informasi*, *4*(35), 90–97. https://doi.org/10.36382/jti-tki.v4i2.108

Muhidin, A., Danny, M., & Rivali, E. (2021). Grouping the spread of the covid-19 virus based on the positive case number, population and area width using the k-means clustering method. *ILKOM Jurnal Ilmiah*, *13*(3), 314–321.

Salsabila, N. (2019). *Klasifikasi Barang Menggunakan Metode Clustering K-Means Dalam Penentuan Prediksi Stok Barang (Studi Kasus : Ukm Mar ' Ah Jilbab Kediri)*. http://etheses.uin-malang.ac.id/16985/

Yang, S., Pan, L., & Liu, S. (2019). An online algorithm for selling your reserved IaaS instances in Amazon EC2 marketplace. *Proceedings of the 2019 IEEE International Conference on Web Services (ICWS)*, 296-303. https://doi.org/10.1109/ICWS.2019.00057