Clustering of Mall Visitors Based on Shopping Behavior Using Hierarchical Clustering for Marketing Strategies

Bintang Muhammad Ramdhan 1, Christopher Kenneth David 2, Ian Pangeswara[3], Valen Claudia Chuardi[4]
[1-4] Multimedia Nusantara University 15810, Faculty of Engineering and Informatics, Indonesia
Email: Bintang.muhammad@student.umn.ac.id [1], christopher.kenneth1@student.umn.ac.id [2], ian.pangeswara@student.umn.ac.id [3], valen.claudia@student.umn.ac.id [4]

ABSTRACT

This research aims to understand the behaviour of mall visitors using the hierarchical clustering method to develop effective business strategies. The data used comes from Kaggle and consists of attributes such as annual income and expense scores. The analysis process begins with pre-processing data and plotting a dendrogram to determine the optimal number of clusters, which in this study was determined to be 11 clusters. Clustering evaluation using silhouette scores shows very good results (0.8465). Next, the data was tested using the K-Nearest Neighbours (KNN) algorithm for classification with an accuracy of 97.25%. Each cluster is then categorized based on income and expenditure scores, providing deeper insight into the characteristics of mall visitors.
The results of this research show that hierarchical clustering can be used as an effective tool in understanding mall customer behaviour and developing more focused business strategies. These strategies can help increase visitor satisfaction and loyalty, as well as increase profits for companies and vendors. Thus, this research makes an important contribution in the context of marketing and business management.

Keywords: Hierarchical Clustering, Mall Visitors, Annual Income, Expense Scores, Business Strategy, Data Analysis, K-Nearest Neighbors (KNN).

CHAPTER 1. Introduction

In today's competitive business landscape, it is crucial for malls to attract and retain potential customers from diverse backgrounds. Understanding the behavior of mall visitors is essential for developing effective marketing strategies that can drive profits and enhance customer satisfaction. Traditional methods of analyzing customer behavior, such as surveys and focus groups, often fall short due to their time-consuming nature and limited scope. This is where Artificial Intelligence (AI) and advanced statistical methods come into play, offering more efficient and comprehensive insights into customer behavior [1].

One such method is hierarchical clustering, a statistical technique used to group data based on similar characteristics. This approach is particularly useful in segmenting mall visitors by their lifestyle attributes, which can include age, gender, shopping frequency, occupation, income, and marital status [2, 3]. These variables are easily measurable and provide valuable insights into the different behaviors and preferences of mall visitors. By understanding these variables, businesses can tailor their marketing strategies to better meet the needs of their diverse customer base [4].

While previous research has explored various methods for analyzing mall visitor behavior, many studies have focused on simpler segmentation techniques or have relied on outdated data collection methods [2, 3, 4]. These studies often fail to capture the dynamic and multifaceted nature of consumer behavior in modern shopping environments. This research aims to bridge this gap by leveraging hierarchical clustering to provide a more nuanced and comprehensive analysis of mall visitor behavior. This approach provides a way to identify distinct customer segments based on lifestyle characteristics and to understand their unique shopping interests and preferences.

The primary aim of this research is to classify and analyze the behavior of mall visitors using hierarchical clustering based on their lifestyle characteristics. Specifically, this research implements agglomerative clustering, a type of hierarchical clustering, to analyze various data features. Through this method, this research aims to uncover patterns and insights that can inform more targeted and effective marketing strategies. By emphasizing the use of advanced clustering techniques, this research seeks to demonstrate how the approach used in this research can achieve a deeper understanding of customer behavior and help malls optimize their marketing efforts.

CHAPTER 2. Literature Review

2.1 Classification

Machine learning has three methods: supervised, unsupervised, and reinforcement learning. Classification is a supervised learning that functions by predicting the correct label of any given data[5]. Classification assigns a label value to a specific class and then identifies that particular type to be of one kind or another. In a Classification machine learning model, the data is split into a training dataset that includes all possible scenarios of the problem that in turn, the model will train itself.

Classification accuracy could help to evaluate the performance of any model based on the predicted class labels and it could serve as a foundation of classification tasks. The ROC curve, a graph that shows the performance of a classification model at all classification thresholds, can be a helpful indicator of how accurate a model is.

Classification has 4 types of classification tasks, as follows:

1. **Binary Classification**
   Binary classification refers to tasks that can give either of any two class labels as the output. In binary classification, generally, one is considered as the normal state and the other is the abnormal state. The simplest example of binary classification is in disease prediction, the "No Disease" is the normal state and "Disease Detected" is the abnormal state. Binary Classification algorithms that are commonly used are K-Nearest Neighbours, Logistic Regression and Support Vector Machine.
2. **Multi-Class Classification**
   In a multi-class classification, there are no two fixed labels; instead, it could have any number of labels. In this classification tasks, there are no normal and abnormal outcomes. Instead, the result will be one of many variables of known classes. An example of multi-class classification is "plant species classification" where the results of the classification could be one of many plant species that is specified in the input data.
3. **Imbalanced Classification**
   In this classification task, the number of examples in each class are unequally distributed, hence the name "imbalanced". Imbalanced classification typically is binary classification that has unequally distributed normal class type and abnormal class type.

2.2 Hierarchical Clustering

Hierarchical Clustering is an algorithm designed to organize data into clusters by constructing a hierarchy through successive stages [1]. Initially, each data point is treated as a separate cluster. The algorithm then repeatedly merges the most similar clusters until all data points are combined into a single cluster. One primary approach in Hierarchical Clustering is the Agglomerative (Bottom-Up) method. This method begins with each data point as its own cluster and iteratively merges the two most similar clusters until a single comprehensive cluster is formed. The merging process is governed by the distance between clusters, which can be quantified using various metrics such as Euclidean distance, Manhattan distance, or the correlation coefficient. Common merging methods include Complete Linkage, Average Linkage, Ward's Method, and Single Linkage. The specific formula used in Hierarchical Clustering is contingent on the chosen merging method. Below are the formulas for several of these methods:

1. Complete Linkage:

$$d(C_1, C_2) = \max\{d(x, y) \mid x \in C_1, y \in C_2\}$$

Where:

1. $d(C_1, C_2)$ is the distance between clusters C_1 and C_2
2. $d(x, y)$ is the distance between data x and y

2. Average Linkage:

$$d(C_1, C_2) = 1/|C_1||C_2| \sum_{x \in C_1} \sum_{y \in C_2} d(x, y)$$

Where:

1. $d(C_1, C_2)$ is the distance between clusters C_1 and C_2
2. $|C_1|$ and $|C_2|$ is the amount of data in clusters C_1 and C_2
3. $d(x, y)$ is the distance between data x and y

3. Ward's Method:

$$d(C_1, C_2) = \Delta SSE(C_1, C_2)$$

$$\Delta SSE(C_1, C_2) = SSE(C_1) + SSE(C_2) - SSE(C_1 \cup C_2)$$

Where:

1. $d(C_1, C_2)$ is the distance between clusters C_1 and C_2

2. $\Delta SSE(C_1, C_2)$ is the change in total squared error (SSE) resulting from merging clusters C_1 and C_2

3. $SSE(C_1)$ and $SSE(C_2)$ are SSE clusters C_1 and C_2

4. $SSE(C_1 \cup C_2)$ is the SSE of cluster $C_1 \cup C_2$

4. Single Linkage:

$$d(C_1, C_2) = \min\{d(x, y) \mid x \in C_1, y \in C_2\}$$

Where:

1. $d(C_1, C_2)$ is the distance between clusters C_1 and C_2

2. d(x, y) is the distance between data x and y

There are also advantages and disadvantages to implementing Hierarchical Clustering algorithm:

A. Advantages Hierarchical Clustering:

1. Ability to Handle Data Without Labels: Does not require previous class labels in the data.
2. Hierarchical Structure Formation: Forms a hierarchical structure in the data, facilitating a better understanding of the relationships between clusters.
3. Flexibility in Handling Various Cluster Structures: Able to handle various forms of cluster structures.

B. Disadvantages Hierarchical Clustering:

1. Scalability Limitations: Inefficient for very large data due to time-consuming computational complexity.
2. Sensitivity to Noise: Susceptible to noise in the data, which can affect the formation of stable clusters.

CHAPTER 3. Dataset Exploration and Research Methodology

3.1. Dataset Exploration

This research carries out Exploratory Data Analysis (EDA) on the dataset used. The dataset used in this research is https://www.kaggle.com/datasets/datascientistanna/customers-dataset , this dataset was obtained from Kaggle.com, a platform that provides various datasets for data analysis. Initially, the dataset consists of 2000 rows and 8 columns, covering a variety of attributes about customer behavior and characteristics.

[2]:

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
0	1	Male	19	15000	39	Healthcare	1	4
1	2	Male	21	35000	81	Engineer	3	3
2	3	Female	20	86000	6	Engineer	1	1
3	4	Female	23	59000	77	Lawyer	0	2
4	5	Female	31	38000	40	Entertainment	2	6
...	...	...	...	...	...	...	...	...
1995	1996	Female	71	184387	40	Artist	8	7
1996	1997	Female	91	73158	32	Doctor	7	7
1997	1998	Male	87	90961	14	Healthcare	9	2
1998	1999	Male	77	182109	4	Executive	7	2
1999	2000	Male	90	110610	52	Entertainment	5	2

2000 rows × 8 columns

Fig.1: *Initial customer dataset*

The dataset was modified in order to focus the analysis, all columns except 'Annual Income ($)' and 'Spending Score (1-100)' were removed. The dataset is then normalized to prepare it for further analysis. Thus, the dataset used in this research consists of 2000 rows and 2 columns, which will be the main focus of the exploration and analysis that will be carried out.

[4]:

	Annual Income ($)	Spending Score (1-100)
0	0.078958	0.39
1	0.184236	0.81
2	0.452694	0.06
3	0.310569	0.77
4	0.200027	0.40
...	...	...
1995	0.970591	0.40
1996	0.385095	0.32
1997	0.478808	0.14
1998	0.958600	0.04
1999	0.582238	0.52

2000 rows × 2 columns

Fig.2: *Dataset after drop and normalization*

3.2. Flow Algorithm

3.2.1 Flowchart (General)

1. **Stage 1: Traditional Marketing Strategy (Traditional Marketing Strategy)**

   At this stage, it refers to conventional marketing methods, such as print advertisements, television advertisements, or radio advertisements.

2. **Stage 2: Data Collection**

This stage involves gathering information about the target audience and market. This data may be collected through surveys, market research, or customer relationship management (CRM) systems.

3. **Stage 3: Analysis of Customer Characteristics**

This state analyzes the data collected from previous stage to identify the demographics, interests and needs of the target audience.

4. **Stage 4: Determining a New Marketing/Sales Strategy**

Based on the analysis of customer characteristics, this stage involves developing new marketing or sales strategies to reach the target audience and achieve the desired marketing objectives.

5. **Stage 5: Implementation of an Artificial Intelligence System (AI System Implementation)**

This stage can refer to the implementation of artificial intelligence (AI) tools or marketing automation software to support a new marketing strategy.

6. **Stage 6: System Prediction Output and Customer Grouping**

This stage refers to the use of AI or marketing automation software to generate customer insights and segment the customer base into groups with similar characteristics. This prediction can be used to personalize marketing campaigns and target the specific customers with the right messages and action.

3.2.2 Flowchart (System)

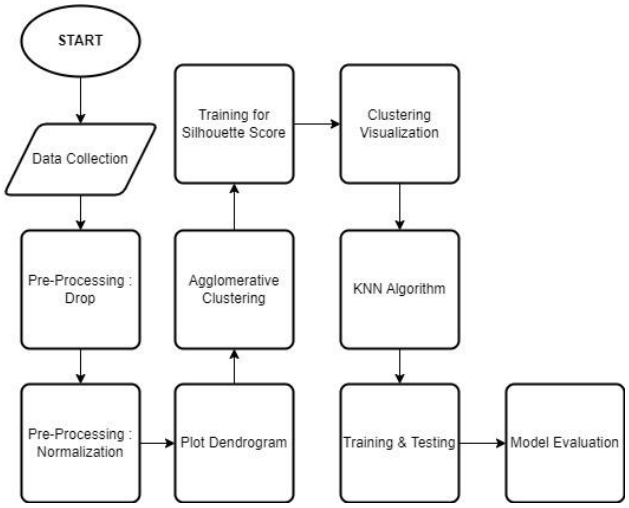

Fig.4: *System Flowchart*

1. **Stage 1: Data Collection**

This stage is a process of collecting data. In this research, customer data is taken from the Kaggle dataset.

2. **Stage 2: Data Pre-Processing**

This stage includes cleaning and preparing the collected data for analysis. These steps involve:

A. Data Pre-Processing: Drop

This stage involves cleaning and preparing the collected data for analysis. This refers to the identification and removal of irrelevant or unusable data from a dataset. This is done for various reasons, such as improving data quality or focusing analysis on certain aspects.

B. Data Pre-Processing: Normalization

This stage is the process of scaling or transforming data to a common range. This is important for machine learning algorithms that are sensitive to data scale.

3. **Stage 3: Plot the Dendrogram**

In this stage, a dendrogram is constructed to ascertain the optimal number of clusters for clustering analysis. By utilizing metrics like 'euclidean' and 'n_neighbors', the dendrogram is graphically depicted, aiding in the identification of the ideal number of clusters.

4. **Stage 4: Agglomerative Clustering**

The clustering process using the Agglomerative Clustering algorithm is carried out to group data into clusters that have been determined based on the dendrogram.

5. **Stage 5: Training for Silhouette Score**

The model that has been formed is evaluated using a silhouette score to determine how well the clustering was carried out. Silhouette scores are calculated to ensure that the resulting clusters are of good quality.

6. **Stage 6: Clustering Visualization**

Clustering visualization is to provide an overview of the division of clusters in the data. This visualization helps understand the data structure and validity of the clusters formed.

7. **Stage 7: KNN Algorithm**

After clustering, the K-Nearest Neighbors (KNN) algorithm is used for data classification. K-Nearest Neighbors (KNN) is a supervised algorithm that classifies data points based on the labels of their nearest neighbors in the training data.

8. **Stage 8: Training & Testing**

Data gets split into a training group and a test group. The KNN model undergoes training with the training group and then faces testing with the test group to assess how well it performs.

9. **Stage 9: Model Evaluation**

The last step involves assessing the performance of the trained model. Evaluation metrics such as accuracy, precision, recall, and F1-score are employed to gauge how effectively the model can classify new data.

CHAPTER 4. Algorithm Implementation and Results

4.1 Algorithm Flow

In this chapter, we explain in detail the stages of implementing the algorithm as well as analyzing the results obtained from applying clustering and classification methods to the mall store customer dataset. There are several stages carried out to obtain results of the effectiveness of the method used.

4.2 Data Pre-Processing

The algorithm starts by opening the data and doing pre-processing. This stage includes deleting all columns except 'Annual Income ($)' and 'Spending Score (1-100)', as well as normalizing the data. The previous ***Fig.1*** and ***Fig.2*** show this process.

4.3 Dendrogram Plot

```
[5]: #Plot Dendrogram untuk cari number of clusters
from scipy.cluster.hierarchy import dendrogram, linkage

#Distance Metric & Linkage Method dengan euclidean & ward
Z = linkage(normalized_df, method='ward', metric='euclidean')

plt.figure(figsize=(12, 6))
dendrogram(Z)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Customer Index')
plt.ylabel('Distance')
plt.show()
```

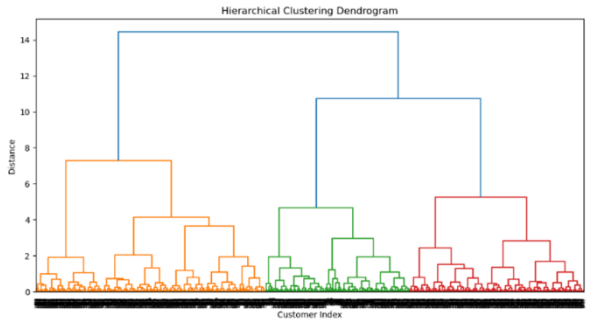

Fig.5: Dendrogram Algorithm & Visualization.

The following stage involves generating a dendrogram to establish the number of clusters utilized. Employing the 'euclidean' metric and the 'ward' method, a dendrogram visualization akin to Fig. 4: Dendrogram Algorithm & Visualization is obtained.

4.4 Determination of the Number of Clusters

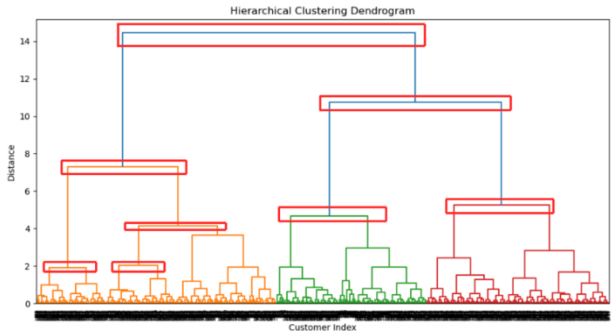

Fig.6: Horizontal lines are called clusters

From the dendrogram visualization results, the clusters values used for implementing Agglomerative Clustering (n_clusters) can be determined. Each horizontal line in the dendrogram represents a cluster. The horizontal line formed shows the similarity between the data groups below it, the higher it shows the clearer the similarity until one large line is formed which represents the entire cluster. Because this is Agglomerative Clustering, the data starts from the bottom individually and is grouped based on similarities.

4.5 Implementation of Agglomerative Clustering

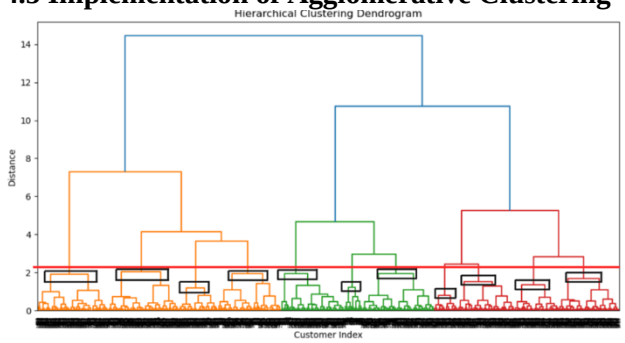

Fig.7: Cutting lines to define clusters.

To determine the number of n_clusters to be used, a horizontal line is drawn that intersects the vertical line in the dendrogram (***Fig.7***). From this image, it is found that there are 11 clusters marked with black boxes. This means there are 11 groups of data.

4.6 Implementation of Agglomerative Clustering

```
[6]: #Hierarchical Clustering
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering

scaler = StandardScaler()
scaled_data = scaler.fit_transform(normalized_df)

#Pilih n_clusters
n_clusters = 11
clusterer = AgglomerativeClustering(n_clusters=n_clusters, linkage='complete')
cluster_labels = clusterer.fit_predict(scaled_data)

normalized_df['Cluster'] = cluster_labels
normalized_df
```

Fig.8: Implementation of Agglomerative Clustering.

[6]:

	Annual Income ($)	Spending Score (1-100)	Cluster
0	0.078958	0.39	2
1	0.184236	0.81	1
2	0.452694	0.06	0
3	0.310569	0.77	9
4	0.200027	0.40	2
...	...	...	...
1995	0.970591	0.40	6
1996	0.385095	0.32	3
1997	0.478808	0.14	0
1998	0.958600	0.04	6
1999	0.582238	0.52	7

2000 rows × 3 columns

Fig.9: Table with new column clusters.

Agglomerative Clustering is then applied with an n_clusters value of 11 and StandardScaler is used in this algorithm. Once clustering is implemented, a new column appears in the table containing the cluster number for each customer, as shown in ***Fig.9.***

4.7 Evaluation with Silhouette Score

```
[7]: #Training datanya
customer_train= AgglomerativeClustering(n_clusters=11, metric='euclidean', linkage='ward')
y_pred= customer_train.fit_predict(normalized_df)
customer_train

[7]: AgglomerativeClustering
AgglomerativeClustering(metric='euclidean', n_clusters=11)

[8]: #Silhouette score biar bisa lihat gimana penyebaran dan pembagian kelompok Clusternya
from sklearn.metrics import silhouette_score

silhouette_avg = silhouette_score(normalized_df, y_pred)

print("The average silhouette score is:", silhouette_avg)

The average silhouette score is: 0.8465650884565473
```

Fig.10: Training and Silhouette Score.

The data that has been grouped is then trained using affinity 'euclidean', linkage 'ward', and n_clusters fixed at 11. To see the training results, the silhouette score function is used. ***Fig.10*** shows the training data and silhouette score with a result of 0.8465, which shows very good clustering.

4.8 Categories of Each Cluster

After training, each cluster is labeled and categorized. The following are the categories for each cluster:

- Cluster 0: Low income (0.2-0.6) & Low spending scores (0-0.4)
- Cluster 1: Moderate to high income (0.6-1.0) & High spending scores (0.6-1.0)
- Cluster 2: High income & Moderate to high spending scores (0.4-0.6)
- Cluster 3: Moderate income (0.4-0.7) & High spending scores (0.6-1.0)
- Cluster 4: Low to moderate income (0.2-0.6) & Low to moderate spending scores (0.2-0.6)
- Cluster 5: Low to moderate income (0.0-0.4) & very low spending scores (0-0.2)
- Cluster 6: High income (0.7-1.0) & Low spending scores (0.0-0.4)
- Cluster 7: Low income (0.0-0.3) & High spending scores (0.6-1.0)
- Cluster 8: Moderate to high income (0.5-0.8) & Moderate to high spending scores (0.5-0.8)
- Cluster 9: Moderate income (0.3-0.6) & Moderate to high spending scores (0.5-0.8)
- Cluster 10: Low income (0.0-0.4) & Low spending scores (0.2-0.4)

```
[9]: def categorize_cluster(row):
        if row['Cluster'] == 0:
            return 'Low income & Low spending scores'
        elif row['Cluster'] == 1:
            return 'Moderate to high income & High spending scores'
        elif row['Cluster'] == 2:
            return 'High income & Moderate to high spending scores'
        elif row['Cluster'] == 3:
            return 'Moderate income & High spending scores'
        elif row['Cluster'] == 4:
            return 'Low to moderate income & Low to moderate spending scores'
        elif row['Cluster'] == 5:
            return 'Low to moderate income & Very low spending scores'
        elif row['Cluster'] == 6:
            return 'High income & Low spending scores'
        elif row['Cluster'] == 7:
            return 'Low income & High spending scores'
        elif row['Cluster'] == 8:
            return 'Moderate to high income & Moderate to high spending scores'
        elif row['Cluster'] == 9:
            return 'Moderate income & Moderate to high spending scores'
        elif row['Cluster'] == 10:
            return 'Low income & Low spending scores'
        else:
            return 'Unknown'

    normalized_df['Customer Category'] = normalized_df.apply(categorize_cluster, axis=1)

    normalized_df
```

Fig.11: If else to assign categories to clusters.

[9]:

	Annual Income ($)	Spending Score (1-100)	Cluster	Customer Category
0	0.078958	0.39	2	High income & Moderate to high spending scores
1	0.184236	0.81	1	Moderate to high income & High spending scores
2	0.452694	0.06	0	Low income & Low spending scores
3	0.310569	0.77	9	Moderate income & Moderate to high spending sc...
4	0.200027	0.40	2	High income & Moderate to high spending scores
...	...	...	...	...
1995	0.970591	0.40	6	High income & Low spending scores
1996	0.385095	0.32	3	Moderate income & High spending scores
1997	0.478808	0.14	0	Low income & Low spending scores
1998	0.958600	0.04	6	High income & Low spending scores
1999	0.582238	0.52	7	Low income & High spending scores

2000 rows × 4 columns

Fig.12: Dataset with 'Customer Category' column.

Fig.11 shows the algorithm used to assign categories to each cluster and the results are shown in ***Fig.12.***

4.9 Clustering Visualization

```
[10]: import matplotlib.pyplot as plt
      import numpy as np

      # Define 11 warna cluster
      color_list = ['#e6194B', '#3cb44b', '#ffe119', '#4363d8', '#f58231',
                    '#911eb4', '#46f0f0', '#f032e6', '#bcf60c', '#fabebe', '#008080']

      plt.figure(figsize=(18, 15))

      for cluster_label in set(y_pred):
          plt.scatter(normalized_df.loc[y_pred == cluster_label, 'Annual Income ($)'],
                      normalized_df.loc[y_pred == cluster_label, 'Spending Score (1-100)'],
                      s=100, label=f'Cluster {cluster_label}', color=color_list[cluster_label])

          # Add labels
          for index, row in normalized_df.loc[y_pred == cluster_label].iterrows():
              plt.text(row['Annual Income ($)'], row['Spending Score (1-100)'], str(index), fontsize=8)

      plt.title('Clusters of customers')
      plt.xlabel('Annual Income ($)')
      plt.ylabel('Spending Score (1-100)')
      plt.legend()
      plt.grid(True)
      plt.show()
```

Fig.13: Algorithm for cluster visualization.

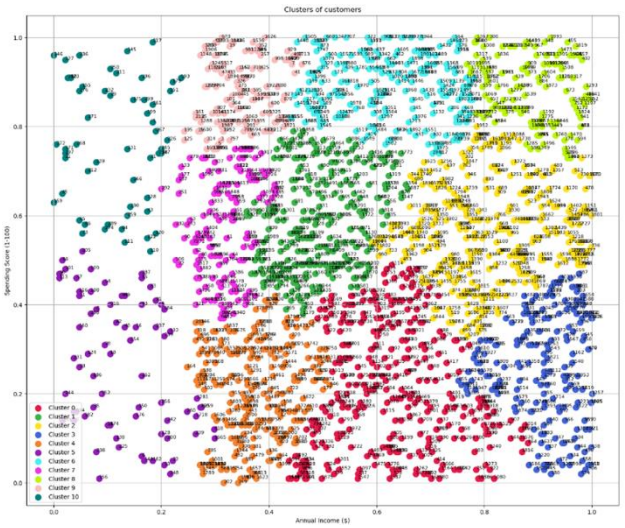

Fig.14: Cluster Visualization Results.

The clusters formed are then visualized using the scatter function, with the parameters "Annual Income ($)" and "Spending Score (1-100)". ***Fig.13*** shows the algorithm used to create the visualization, and ***Fig.14*** shows the visualization results. The cluster distribution looks orderly, indicating successful clustering.

4.10 Model Evaluation with KNN

```
[11]: import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# 11 warna cluster
color_list = ['#e61948', '#3cb44b', '#ffe119', '#4363d8', '#f58231',
              '#911eb4', '#46f0f0', '#f032e6', '#bcf60c', '#fabebe', '#008080']

# Splitting data menjadi training & testing
X = normalized_df[['Annual Income ($)', 'Spending Score (1-100)']]
y = normalized_df['Cluster']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# KNN Classification
knn = KNeighborsClassifier(n_neighbors=4, metric='manhattan')
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)

# Performa model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", classification_report(y_test, y_pred))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# Cross-Validation
cv_scores = cross_val_score(knn, X, y, cv=8, scoring='accuracy')
print("Cross-validation scores:", cv_scores)
print("Mean cross-validation score:", cv_scores.mean())

# Hyperparameter Tuning with Grid Search
param_grid = {'n_neighbors': np.arange(1, 10), 'metric': ['euclidean', 'manhattan']}
grid = GridSearchCV(KNeighborsClassifier(), param_grid, cv=8)
grid.fit(X_train, y_train)

print("Best parameters:", grid.best_params_)
print("Best cross-validation score:", grid.best_score_)

# Visualisasi Setelah KNN
plt.figure(figsize=(18, 15))

for cluster_label in set(y):
    plt.scatter(normalized_df.loc[y == cluster_label, 'Annual Income ($)'],
                normalized_df.loc[y == cluster_label, 'Spending Score (1-100)'],
                s=100, label=f'Cluster {cluster_label}', color=color_list[cluster_label])

    for index, row in normalized_df.loc[y == cluster_label].iterrows():
        plt.text(row['Annual Income ($)'], row['Spending Score (1-100)'], str(index), fontsize=8)

plt.title('Clusters of customers')
plt.xlabel('Annual Income ($)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.grid(True)
plt.show()
```

Fig.15: KNN, *Confusion Matrix* , *Cross-Validation* , and *Hyperparameter Tuning Grid Search algorithms.*

To prove the effectiveness of hierarchical clustering, the clustered data was tested using the KNN algorithm. This process includes split data, training, testing, and model evaluation using Confusion Matrix, Cross-Validation, and Hyperparameter Tuning Grid Search. The data is split with a proportion of 80:20 for training and testing, and uses random_state of 42. The KNN algorithm uses the 'manhattan' metric and n_neighbors of 4.

4.11 Model Results and Evaluation

```
Accuracy: 0.9725
              precision    recall  f1-score   support

           0       0.88      1.00      0.93        64
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        15
           3       1.00      0.89      0.94        46
           4       1.00      1.00      1.00        37
           5       0.98      0.95      0.97        61
           6       1.00      0.97      0.99        38
           7       1.00      0.96      0.98        54
           8       1.00      1.00      1.00        24
           9       1.00      1.00      1.00        25
          10       0.96      1.00      0.98        23

    accuracy                           0.97       400
   macro avg       0.98      0.98      0.98       400
weighted avg       0.98      0.97      0.97       400
```

Fig.16: Accuracy Results.

Fig.16 shows the accuracy of KNN after training and testing with a value of 0.9725 or 97.25%. Each 0–10 cluster has precision, recall, and f1-score above 0.88, indicating excellent performance.

4.12 Checking for Overfitting and Underfitting

```
Cross-validation scores: [0.992 0.968 0.984 0.964 0.972 0.976 0.984 0.964]
Mean cross-validation score: 0.9755
Best parameters: {'metric': 'euclidean', 'n_neighbors': 1}
Best cross-validation score: 0.98875
```

Fig.17: *Cross-Validation* Results and *Hyperparameter Tuning Grid Search* .

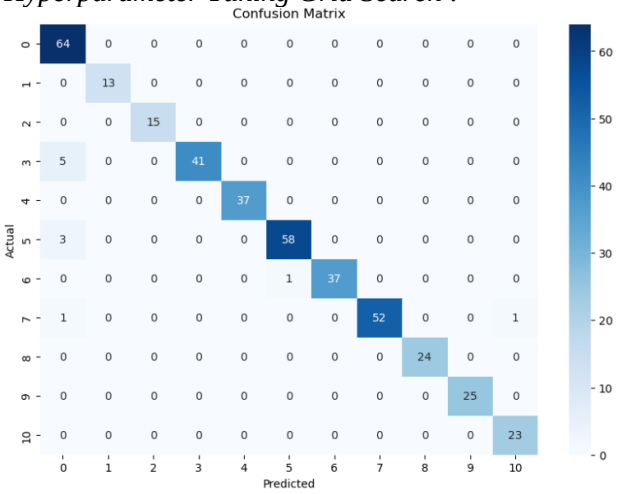

Fig.18: *Confusion Matrix* .

Checking for overfitting or underfitting was carried out using Cross-Validation and Hyperparameter Tuning Grid Search with a CV of 8. Based on the Cross-Validation Scores and Best Parameters, there were no indications of overfitting or underfitting. Cross-Validation Scores ranged from 0.964 to 0.992 with a mean of 0.9755. The best parameters ({'metric': 'euclidean', 'n_neighbors': 1}) indicate a simple and effective model. ***Fig.17*** shows the results of Cross-Validation and Hyperparameter Tuning

Grid Search, while ***Fig.18*** shows the Confusion Matrix.

CHAPTER 5. Conclusion

This research aims to understand the behavior of mall visitors in order to develop effective business strategies through the hierarchical clustering method. The method used in this research is the agglomerative clustering algorithm, which groups mall visitors based on their life characteristics. The analyzed dataset is taken from Kaggle, consisting of attributes such as annual income and expense scores.

The analysis process begins with data pre-processing, which includes removing irrelevant columns and normalizing the data in preparation for further analysis. Dendrogram plotting was used to decide the optimal number of clusters, which in this study was determined to be 11 clusters. Clustering evaluation using the silhouette score shows very good results with a value of 0.8465, indicating high cluster quality.

After clustering, the data was tested using the K-Nearest Neighbors (KNN) algorithm for classification. The test results show very good accuracy, namely 97.25%, with precision, recall and f1-score above 0.88 for each cluster. Each cluster is then categorized based on income and expenditure scores, providing deeper insight into the characteristics of mall visitors.

This research answers the following key questions:

1. **How to build a business strategy to attract customers from various economic groups?**

By using hierarchical clustering, businesses can identify and understand different customer groups based on their economic characteristics. Business strategies can be built by targeting each group with appropriate approaches, such as special offers for low-income groups or loyalty programs for high-spending customers.

2. **How to compare each economic group of customers?**

By grouping customers into 11 clusters based on income and expense scores, the research allows for comparisons between groups. This analysis shows how each group behaves differently in the context of shopping at the mall.

3. **How to identify customer groups' shopping interests?**

Through clustering analysis, the shopping interests of each group can be identified based on spending patterns. Clusters with high spending scores indicate groups of customers with greater shopping interest, while clusters with low spending scores indicate groups that tend to be more thrifty.

Overall, this research makes a significant contribution to understanding mall customer behavior and developing more structured and effective marketing strategies. Thus, the results of this research can be used to increase visitor satisfaction and loyalty, as well as increase profits for companies and vendors.

REFERENCE

[1] Alfian, G., Octava, MQH, Hilmy, FM, Nurhaliza, RA, Saputra, YM, Putri, DGP, Syahrian, F., Fitriyani, NL, Atmaji, FTD, Farooq, U., Nguyen, DT, & Syafrudin, M. (2023). Customer shopping behavior analysis using RFID and machine learning models. Information, 14(10), 551. https://doi.org/10.3390/info14100551

[2] Jiang, H., He, M., Xi, Y., & Zeng, J. (2021). Machine-Learning-Based user position prediction and behavior analysis for location services. Information, 12(5), 180. https://doi.org/10.3390/info12050180

[3] Pradana, M. (2021). Maximizing Strategy Improvement in Mall Customer Segmentation using K-means Clustering. Journal of Applied Data Sciences, 2(1). https://doi.org/10.47738/jads.v2i1.18

[4] Du H, Yu Z, Guo B, Han Q and Chen C. (2020). GroupShop: monitoring group shopping behavior in the real world using mobile devices. Journal of Ambient Intelligence and Humanized Computing. 10.1007/s12652-019-01673-9. 14:5. (6367-6378). Online publication date: 1-May-2023. https://link.springer.com/10.1007/s12652-019-01673-9

[5] Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. Informatica, 31(3), 249-268. https://doi.org/10.31449/inf.v31i3.148

[6] Machine learning-based intelligent recommendation in virtual malls. (nd). IEEE Conference Publications | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/138224

[7] Werner, E., Clark, J.N., Hepburn, A., Bhamber, R.S., Ambler, M., Bourdeaux, C.P., McWilliams, C.J., & Santos-Rodriguez, R. (2023). Explainable hierarchical clustering for patient subtyping and

risk prediction. Experimental Biology and Medicine. https://doi.org/10.1177/15353702231214253

[8] Li, J., Pan, S., Huang, L., & Zhu*, X. (2019). A machine learning based method for customer behavior prediction. Tehnički Vjesnik/TehničKi Vjesnik, 26(6). https://doi.org/10.17559/tv-20190603165825

[9] Setiyani, R. (2014). PDF scientific articles. Jenderalsoedirman University. https://www.academia.edu/5512773/Article_ilmiah_PDF

[10] Torrens P. (2022). Agent models of customer journeys on retail high streets. Journal of Economic Interaction and Coordination. 10.1007/s11403-022-00350-z. 18:1. (87-128). Online publication date: 1-Jan-2023. https://link.springer.com/10.1007/s11403-022-00350-z

[11] Wang P, Guo B, Wang Z and Yu Z. ShopSense:Customer Localization in Multi-Person Scenario With Passive RFID Tags. IEEE Transactions on Mobile Computing. 10.1109/TMC.2020.3029833. 21:5. (1812-1828). https://ieeexplore.ieee.org/document/9219250/

[12] Bermejo C, Chatzopoulos D and Hui P. EyeShopper. Proceedings of the 28th ACM International Conference on Multimedia. (2765-2774). https://doi.org/10.1145/3394171.3413683

[13] Guo B, Liu Y, Ouyang Y, Zheng V, Zhang D and Yu Z. Harnessing the Power of the General Public for Crowdsourced Business Intelligence: A Survey. IEEE Access. 10.1109/ACCESS.2019.2901027. 7. (26606-26630). https://ieeexplore.ieee.org/document/8649614 /

[14] Guo B, Wang Z, Wang P, Xin T, Zhang D and Yu Z. DeepStore: Understanding Customer Behaviors in Unmanned Stores. IT Professional. 10.1109/MITP.2019.2928272. 22:3. (55-63). https://ieeexplore.ieee.org/document/9098002/

[15] Zhou Z, Shangguan L, Zheng X, Yang L and Liu Y. (2017). Design and Implementation of an RFID-Based Customer Shopping Behavior Mining System. IEEE/ACM Transactions on Networking. 25:4. (2405-2418). Online publication date: 1-Aug-2017. https://doi.org/10.1109/TNET.2017.2689063

[16] Jiao M, Chen X, Su Z and Chen X. (2016). Research on personalized recommendation optimization of E-commerce system based on customer trade behavior data 2016 Chinese Control and Decision Conference (CCDC). 10.1109/CCDC.2016.7532169. 978-1-4673-9714-8. (6506-6511). http://ieeexplore.ieee.org/document/7532169/