Journal of Applied Informatics and Computing (JAIC)
Vol.6, No.1, Juli 2022, pp. 07~15
e-ISSN: 2548-6861

Application of Data Mining with the K-Means Clustering Method and Davies Bouldin Index for Grouping IMDB Movies

Ilham Firman Ashari [1]*, Romantika Banjarnahor [2]*, Dede Rodhatul Farida[3]*, Sicilia Putri Aisyah[4]*, Anastasia Puteri Dewi [5]*, Nuril Humaya [6]*
* Teknik Informatika, Institut Teknologi Sumatera
firman.ashari@if.itera.ac.id [1], romantika.118140045@student.itera.ac.id [2], dede.118140059@student.itera.ac.id [3], sicillia.118140091@student.itera.ac.id [4], anastasia.118140127@student.itera.ac.id [5], nuril.118140169@student.itera.ac.id [6]

Article Info

Article history:

Received 2021-10-13
Revised 2021-12-06
Accepted 2021-02-14

Keyword:

Data Mining,
IMDb,
Clustering,
K-means.

ABSTRACT

Along with the development of technology, the film industry continues to increase, this can be seen from the number of films that appear both in cinemas and tv shows. The Internet Movie Database (IMDb) is a website that provides information about films from around the world, including the people involved in the films. Information contained on IMDB such as actor/actress, director, writer, to the soundtrack used. In addition, IMDb is the most popular and trusted source of information for movies, TV, and other celebrity content. In this case, the researcher will conduct research on the film with what title is the most popular among the public by looking at some of the parameters contained in IMDB such as the number on the rating, score, certificate, and votes obtained from the audience. The data used comes from the Kaggle.com website. The data mining method used is the K-Means clustering method. To find out the optimal cluster value, the Davies Bouldin index is used. The K-Means algorithm will group the data based on the centroid. The parameters used for clustering are runtime, IMDB rating, meta score, number of votes, and gross. The results of the study obtained that the average calculation of the highest attributes was 48.74 and the number of clusters formed was 4 clusters. The results of the evaluation using the confusion matrix obtained an accuracy value of 100%.

I. INTRODUCTION

The number of film productions in various parts of the world has increased significantly [1]. With rapid technological advances in this day and age, all data or information about films is readily available on the internet [2]. The internet has also become one of the most widely used media for distributing films. The internet itself acts as a medium of communication between filmmakers and film lovers. Film information regarding themes, genres, actors, ratings, directors, year of publication, scores, votes and others can be found easily via the internet [3]. We can get several sources of information for movies from websites, such as IMDb, Kaggle, and Netflix [4]. IMDb is a web that provides information related to films from around the world, with complete information on the name of the actor/actress, duration, income, and rating also provided on the IMDb web.

The more information about films that are easily obtained from the internet, the more difficult it will be to determine which films are suitable for the needs of film lovers. One of the big providers of information or data is the kaggle.com website. The Kaggle.com website provides various types of datasets, including information related to movies. One of the datasets is IMDB Movies which accommodates various kinds of wide-screen film information with movie titles up to 1000 titles or also the top ranking of the top films from the IMDB ranking.

Sometimes it is not just an application that is needed by someone to get information but also a technique is needed to *determine the best parameters that can be used by someone*

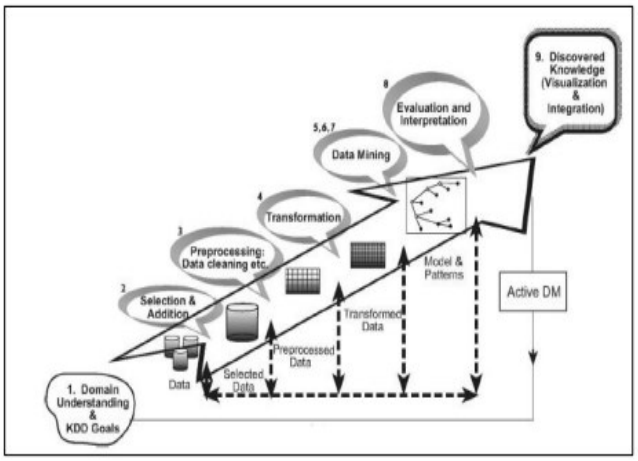
to get the best information from several large data sets [5]. Therefore, to process a lot of data, an algorithm is used. One of the algorithms that can be used to process data is the k-means clustering algorithm [6].

Related research that has been done by other researchers, namely the researchers Vlandari, et al., Alfianti, and Handoko, et al. The research conducted by Vulandari is condcting clustering of criminal acts in the province of Central Java using the k-means clustering algorithm [7]. The data used is from 2014 to 2016, where the data obtained consists of 4 clusters. The results obtained by clustering can be done by region and visualization is displayed in the form of a map. Another study was carried out by Alfianti, where Alfianti conducted surveillance of the Covid-19 distribution area in the Karawang district. Funds were taken on the covid19 karawang website. From the data obtained 16.7% with the highest distribution of data, 33.3% with a moderate level of distribution, and 50% with a low level of distribution [8]. Other research has also been carried out by Handoko, who conducted research to determine the level of sales of Telkomsel data packages using the k-means clustering algorithm. Where the data are grouped into 3, low sales data, moderate sales data, and high sales data. Where the percentage of 100% conformity is obtained compared to manual clustering [9].

In performing clustering there are no special provisions to determine the right K value, so this is a deficiency or error that often causes the resulting model to be low in accuracy, fortunately machine learning already has a technique to deal with this problem, namely by conducting validation tests on the clustering model. Therefore, the data grouping or data model generated from the K-Means clustering method needs to be evaluated on the clustering results using one of the clustering validation techniques, namely the Davies Bouldin Index technique, by using this technique the cohesion matrix (closeness of one group) and separation matrix will be known (differences between groups), the smaller the Davies Bouldin Index (DBI) value produced, the more optimal the clustering model [10] evaluation using the Davies Bouldin Index will produce the optimal number of clusters as in research [11] which performs clustering optimization on provincial data using Davie s Bouldin Index produces the smallest DBI value with the number of clusters 3, which means that the optimal cluster is generated from the number of clusters equal to 3, research [10] which evaluates clustering uses 2 evaluation techniques for the number of clusters, namely Sum of Square Error (SSE) and Davis Bouldin Index (DBI) obtained the result that the performance evaluation technique using DBI was better than SSE because DBI approached the intra-cluster distance so that it not only determined the optimal number of clusters but also provided more patterned and detailed information.In this case, the researcher wants to conduct research on the characteristics of the films in the IMDb Movie dataset. So that film connoisseurs can use these results as reference material to choose films according to current trends. Data mining processing using k-means clustering using the python programming language can be used as a tool for processing large data sets.

II. METHODOLOGY

The methodology used in this study uses Knowledge Discovery in Database (KDD). KDD is a method used to obtain new knowledge from some data that has been taken and processed, so that information is obtained for strategic decisions [12]. The stages of the KDD process consist of 6 stages as follows:



Gambar 1. KDD Stage [13]

1. Data Selection

The initial process carried out is to select the data so that the modeling can be appropriate, and the results can be accurate. The dataset that has been selected and collected will be processed using an algorithm. The dataset used in this study is the IMDB top 1000 dataset obtained from the Kaggle.com website. From this dataset there are 16 variables namely *poster_link, series_title, released_year, certificate, runtime, genre, imdb_rating, overview, meta_score, director, star1, star2, star3, star4, no_of_votes, gross*. The data used for processing are five parameters, including variable *runtime, imdb_rating, meta_score, no_of_votes,* and *gross.*

2. Pre-processing

After the data is obtained, the preprocessing process will then be carried out by cleaning the data. It aims to normalize data, eliminate data inconsistencies, and fix data duplication, or unwanted data. In this research there is still some data that is not needed so that it must be eliminated and cleaned so that the data processing process becomes more leverage, unused data are *poster_links, series_title, released_year, certificate, genre, director, star1, star2, star3,* dan *star4.* Parameters used as evaluation are movie time, imdb_rating, meta score, no of votes and gross. This parameter indicates the performance of the displayed film.

3. Transformation

The data transformation process is carried out so that the data becomes more appropriate and precise in the processing with

data mining. If the data used consists of several variables that do not match, it is also necessary to transform these variables. The data used are as in the following table. Where the data for the IMDB rating is adjusted for the rating from 1 to 10, the meta score is from 10 to 100. An example can be seen in the table 1 below.

Table 1. The example of transformation of dataset IMDB movies

Name	Run time	IMDB rating	Meta Score	No Of Votes	Gross
Movie 1	120 Min	9.4	90	20021	28281 922
Movie 2	130 Min	9.2	80	1019101	43421 893
Movie 3	122 Min	6.8	70	1818181	82728 812

4. Data Mining

After the data transformation process, the next step is the data mining process. The data mining process is carried out for patterns from previously selected data using certain algorithms or techniques. The method used in this research is to use k means. The k-means method is a method of grouping data by taking parameters from several k clusters and dividing the data into clusters based on the similarities between the data in a cluster and the differences between clusters [14]. In other words, this technique tries to minimize the variation between the data of one group and maximize the variation with the data of another group [15]. The stages of the K-Means algorithm are as shown in Figure 2 [16][17]:

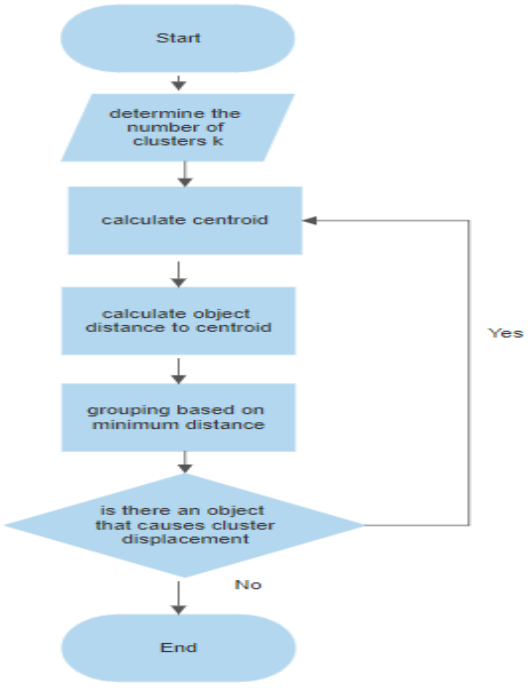


Figure 2. Data Mining Process

a. Set the value of k as the number of clusters to be generated

b. Sets the cluster centre or centroid. The value of this centre point is determined randomly.

c. Calculate the distance of each data to the centre of the cluster using the Euclidean Distance equation.

$$d_{ik} = \sqrt{\sum_j^m (C_{ij} - C_{kj})^2} \quad (1)$$

Note :

D_{ik} = Object distance between data values of cluster centre values

m = Number of data dimensions

C_{ij}= Data value from dimension to k

C_{kj} = Cluster centre value from dimension to k

d. Grouping data into clusters with short distances or minimal distances. The formula used is the following equation

$$min \sum_{k=1}^{k} d_{ik} = \sqrt{\sum_j^m (C_{ij} - C_{kj})^2} \quad (2)$$

e. Calculate the new cluster centre using the following equation.

$$C_{kj} = \frac{\sum_{i=1}^{p} X_{ij}}{P} \quad (3)$$

Where $X_{ij} \in C_k$ or cluster K and *P is the number of cluster members K*

An illustration of the k-means algorithm can be seen in the following figure [13].

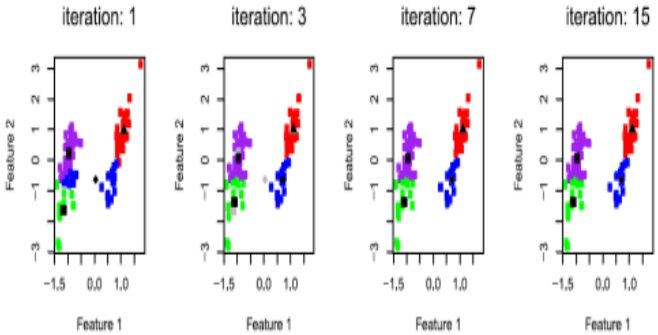


Figure 3. Illustration of cluster iteration on k-means

5. Evaluation/Interpretation

The last stage is the process of translating the results of the pattern from the data mining process. The resulting pattern of information will then be displayed in the form of data visualization. Presentation of data using visualization makes the data easier to understand.

III. RESULT AND DISCUSSION

A. *Data collection*

The initial dataset used is IMDB movies top 1000 data obtained from the Kaggle.com website. The snippet of this dataset can be seen in Figure 4.

Poster_	Series_	Released	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_	Director	Star1	Star2	Star3	Star4	No_of_Vo	Gross
https://m	The Sh	1994	A	142 min	Drama	9.3	Two impris	80	Frank Da	Tim Rc	Morga	Bob G	William	2343110	28,341,469
https://m	The Go	1972	A	175 min	Crime, D	9.2	An organiz	100	Francis f	Marlo	Al Pac	James	Diane K	1620367	134,966,411
https://m	The Da	2008	UA	152 min	Action, (	9	When the	84	Christop	Christi	Heath	Aaron	Michae	2303232	534,858,444
https://m	The Go	1974	A	202 min	Crime, D	9	The early l	90	Francis f	Al Pac	Rober	Rober	Diane K	1129952	57,300,000
https://m	12 Ang	1957	U	96 min	Crime, D	9	A jury holc	96	Sidney L	Henry	Lee J.	Martin	John Fi	689845	4,360,000

Figure 4. Initial Dataset Snippet

This dataset consists of 1000 rows with 16 columns, the 16 columns are *Poster_Link, Series_Title, Released_Year, Certificate, Runtime, Genre, IMDB_Rating, Overview, Meta_score, Director, Star1, Star2, Star3, Star4, No_of_Votes, Gross.*

B. *Pre-Processing*

Data preprocessing was carried out before being used for research; the results of the data preprocessing process determined the accuracy of the clustering process in this study. From the data processed as much as 1000 data, it turns out that there are still things that are not normal, so changes need to be made, between min data at runtime or characters that do not match gross. Preprocessing performed on this parameter can be seen in the image below. In Figure 5, a process is performed to remove the 'min' character at runtime.

```
#menghilangkan 'min' pada data runtime
data['Runtime'] = data['Runtime'].map(lambda x: x.rstrip(' min'))
```

```
data
```

	Poster_Link	Series_Title	Released_Year	Certificate	Runtime
0	https://m.media-amazon.com/images/M/MV5BMDFkYT...	The Shawshank Redemption	1994	A	142
1	https://m.media-amazon.com/images/M/MV5BM2MyNj...	The Godfather	1972	A	175

Figure 5. Pre-processing process

In addition, to facilitate the calculation process, the runtime data will be converted into numeric form by using the code as shown in Figure 6 below:

```
#membuat runtime menjadi numerik
data.Runtime = data.Runtime.astype(float)
```

Figure 6. Convert runtime format to numeric

Then, another process is removing the comma character in the gross parameter, this is used to simplify the calculation process. The result can be seen in figure 7.

```
#menghilangkan karakter koma pada Gross
data['Gross'] = data['Gross'].str.replace(',','')

data
```

	Poster_Link	Series_Title	Released_Year	Certificate	Runtime	Genre	IMDB_Rating	Overview	Meta_score	Director
0	https://m.media-amazon.com/images/M/MV5BMDFkYT...	The Shawshank Redemption	1994	A	142.0	Drama	9.3	Two imprisoned men bond over a number of years...	80.0	Frank Darabont
1	https://m.media-amazon.com/images/M/MV5BM2MyNj...	The Godfather	1972	A	175.0	Crime, Drama	9.2	An organized crime dynasty's aging patriarch t...	100.0	Francis Ford Coppola
2	https://m.media-amazon.com/images/M/MV5BMTMxNT...	The Dark Knight	2008	UA	152.0	Action, Crime, Drama	9.0	When the menace known as the Joker wreaks havo...	84.0	Christopher Nolan
3	https://m.media-amazon.com/images/M/MV5BMWMwMG...	The Godfather: Part II	1974	A	202.0	Crime, Drama	9.0	The early life and career of Vito Corleone in ...	90.0	Francis Ford Coppola
4	https://m.media-amazon.com/images/M/MV5BMWU4N2...	12 Angry Men	1957	U	96.0	Crime, Drama	9.0	A jury holdout attempts to prevent a miscarria...	96.0	Sidney Lumet

Figure 7. Preprocessing removes the comma character in the gross. parameter

The results of the preprocessing process can be seen in the image below. Where the parameters that are filtered to display are runtime, imdb_rating, meta score, number of votes, and gross. These parameters will be used for clustering.

```
data = data[[ 'Runtime', 'IMDB_Rating', 'Meta_score', 'No_of_Votes', 'Gross']]
data
```

	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross
0	142.0	9.3	80.0	2343110	28341469.0
1	175.0	9.2	100.0	1620367	134966411.0
2	152.0	9.0	84.0	2303232	534858444.0
3	202.0	9.0	90.0	1129952	57300000.0
4	96.0	9.0	96.0	689845	4360000.0
...	...	...	...	...	...
990	157.0	7.6	77.0	30144	696690.0
991	144.0	7.6	50.0	45338	1378435.0
992	78.0	7.6	65.0	166409	141843612.0
994	87.0	7.6	96.0	40351	13780024.0
997	118.0	7.6	85.0	43374	30500000.0

714 rows × 5 columns

Figure 8. The results from the preprocessing process

The pre-processed dataset consists of 714 rows with 5 columns. The results of calculations with statistics on 5 parameters can be seen in Figure 9 below.

```
data.describe()
```

	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross
count	714.000000	714.000000	714.000000	7.140000e+02	7.140000e+02
mean	123.715686	7.937115	77.158263	3.561348e+05	7.851359e+07
std	25.887535	0.293278	12.401144	3.539011e+05	1.149780e+08
min	72.000000	7.600000	28.000000	2.522900e+04	1.305000e+03
25%	104.250000	7.700000	70.000000	9.600975e+04	6.157408e+06
50%	120.000000	7.900000	78.000000	2.366025e+05	3.485015e+07
75%	136.000000	8.100000	86.000000	5.077922e+05	1.024641e+08
max	238.000000	9.300000	100.000000	2.343110e+06	9.366622e+08

Figure 9. Data statistics from the results of preprocessing of 1000 data

C. *Clustering*

In determining K or the number of clusters, a function in the library is used using the kmeans-sklearn. In this study, to determine the optimal number of clusters, the elbow method and Davies Bouldin index are used. The program code can be seen in Table 2.

Table 2. K-Means Cluster Code

```
X, y = make_blobs(n_samples=300, centers=4,
cluster_std=0.60, random_state=0)
plt.scatter(X[:,0], X[:,1])
wcss = []
results = {}
for i in range(2, 11):
    kmeans = KMeans(n_clusters=i, init='k-
means++', max_iter=300, n_init=10,
random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
    labels = kmeans.fit_predict(X)
    db_index = davies_bouldin_score(X, labels)
    results.update({i: db_index})
plt.plot(range(2, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

The results obtained from the code above are shown in Figure 10. The formula for finding the Davies Bouldin Index (DBI) value is by adding up the maximum value of the ratio between clusters and then dividing by the number of clusters or K. The following is the equation to calculate the DBI value [10].

$$DBI = \frac{1}{K}\sum_{i=1}^{k} max_{i=j} R(i,j) \quad (4)$$

Information:
K = number of clusters
maxi=jR(i,j) = the maximum value of the ratio of cluster i and cluster j

The clustering evaluation technique with the Davies Bouldin Index needs to compare the DBI value in each cluster result, then the smallest DBI value or which is close to 0 but not negative is the most optimal cluster result.

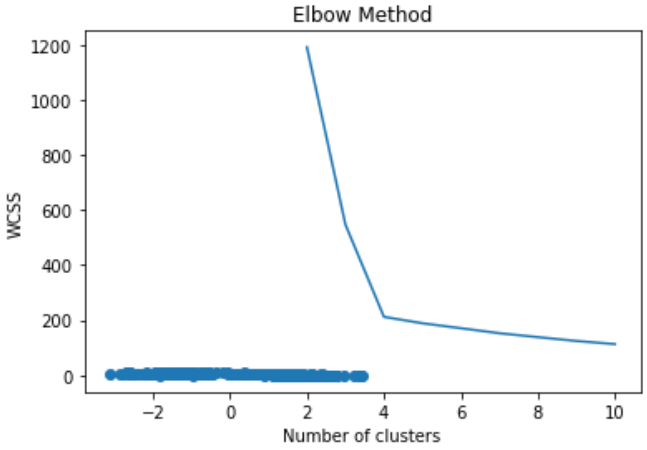


Figure 10. Elbow Method Visualization

The code to calculate the DBI value and display it in the form of a visualization can be seen in table 3 and figure 11.

Table 3. DBI Score and Visualization Code

```
print("the value of DBI : ", db_index)
plt.plot(list(results.keys()),
list(results.values()))
plt.xlabel("Number of clusters")
plt.ylabel("Davies-Boulding Index")
plt.show()
```

To determine the value of K, it is done by checking the value of K from 2 to 11, then look at the score obtained by using a graphical visualization by using the Davies Bouldin index. Located in the number of clusters 4. The results of the visualization can be seen in Figure 11.

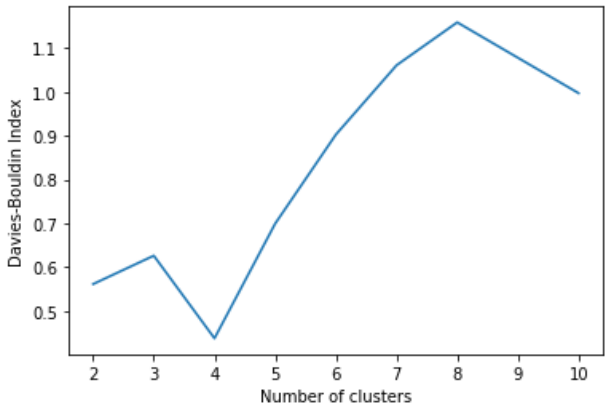


Figure 11. Visualization of Davies Boulden Index (DBI)

The graph shows that the right K value to be used in this study is K = 4. After getting the K value, clustering processing can be carried out using the k-means method with the k-means sklearn library, namely by adding a cluster attribute that

accommodates the cluster value in each row or row. The process of labeling 4 clusters into array values can be seen in Figure 12 below.

```
kmeans = KMeans(n_clusters=4)
kmeans.fit(data)
```

```
KMeans(n_clusters=4)
```

```
kmeans.labels_
```

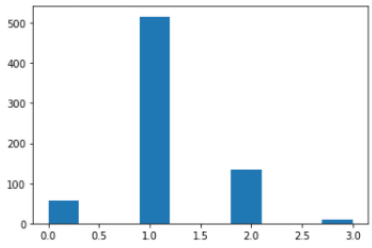
```
array([0, 1, 3, 0, 0, 2, 1, 1, 2, 0, 2, 2, 0, 2, 1, 0, 2, 1, 0, 1, 0, 0,
       1, 1, 0, 1, 1, 2, 0, 2, 0, 0, 0, 1, 0, 1, 0, 0, 0, 2, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 3, 3, 1, 1, 2, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 2,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0,
       0, 2, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 2, 0, 0, 1, 0, 0, 0, 1, 1, 2, 0, 1,
       0, 0, 1, 0, 0, 0, 0, 0, 2, 1, 0, 2, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 2, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 2, 0, 1, 0, 0, 0, 1, 2, 1, 0, 1, 1,
       0, 1, 1, 2, 0, 0, 1, 1, 0, 3, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0,
       1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 2, 0, 0, 1, 3, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 2, 0,
       0, 0, 1, 0, 0, 2, 0, 1, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 3, 2, 0, 1, 0, 0, 0, 1, 0, 0,
       2, 0, 0, 1, 0, 1, 1, 1, 0, 2, 0, 0, 1, 1, 0, 0, 2, 0, 0, 1, 3, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 2, 0,
       0, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 2, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 2, 0, 1, 0, 0, 2,
       2, 1, 1, 1, 2, 2, 1, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 1, 0,
       0, 0, 3, 2, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 2, 0, 0, 0, 0, 0,
       1, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 2, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0])
```

Figure 12. Labeling Array Cluster value against 1000 Data

Figure 12 is a cluster distribution diagram. It shows the distribution of 4 clusters from clusters 0-3, cluster 0 has the most values, and cluster 2 has the least values. The distribution of clustering values can also be seen with the distribution of values as shown in Figure 13 below.

```
plt.hist(data['cluster'])
```

```
(array([ 56.,   0.,   0., 516.,   0.,   0., 133.,   0.,   0.,   9.]),
 array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3. ]),
 <a list of 10 Patch objects>)
```




```
print(kmeans.cluster_centers_)
```

```
[[1.25892857e+02 8.01785714e+00 7.75714286e+01 7.81134446e+05
  3.12128918e+08]
 [1.21503876e+02 7.91627907e+00 7.72112403e+01 2.42957306e+05
  2.44825860e+07]
 [1.29451128e+02 7.97443609e+00 7.68721805e+01 5.72403910e+05
  1.48525453e+08]
 [1.52222222e+02 8.07777778e+00 7.57777778e+01 1.00456056e+06
  6.88064530e+08]]
```

Figure 13. Cluster distribution diagram with data center points

To display the centroid of the 4 clusters, namely clusters 0, 1, 2, and 3, it can be seen in table 4 below.

Table 4. Visualization of Centroids from Clusters

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans
== 0, 1], s = 60, c = 'red', label =
'Cluster0')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans
== 1, 1], s = 60, c = 'blue', label =
'Cluster1')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans
== 2, 1], s = 60, c = 'green', label =
'Cluster2')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans
== 3, 1], s = 60, c = 'violet', label =
'Cluster3')

plt.scatter(kmeans.cluster_centers_[:,    0],
kmeans.cluster_centers_[:, 1], s = 100, c =
'gray', label = 'Centroids')
plt.legend()
plt.show()
```

The results of the visualization can be seen in Figure 14.

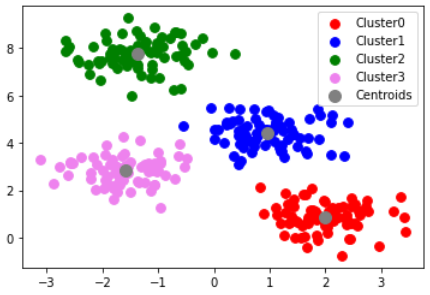


Figure 14. The Visualization of Centroids from Clusters

The results from the clusters that have been labeled for 997 movie titles can be seen in Figure 15.

```
data['cluster'] = kmeans.labels_
# data.head()
```

```
data
```

	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross	cluster
0	142.0	9.3	80.0	2343110	28341469.0	1
1	175.0	9.2	100.0	1620367	134966411.0	2
2	152.0	9.0	84.0	2303232	534858444.0	3
3	202.0	9.0	90.0	1129952	57300000.0	1
4	96.0	9.0	96.0	689845	4360000.0	1
...	...	...	...	...	...	...
990	157.0	7.6	77.0	30144	696690.0	1
991	144.0	7.6	50.0	45338	1378435.0	1
992	78.0	7.6	65.0	166409	141843612.0	2
994	87.0	7.6	96.0	40351	13780024.0	1
997	118.0	7.6	85.0	43374	30500000.0	1

714 rows × 6 columns

Figure 15. Cluster of each Parameters

Visualization results for clusters 0, 1, 2, 3, and 4 of the runtime parameters, imdb_rating, meta_score, no_of_votes, and gross can be seen in Figure 16.

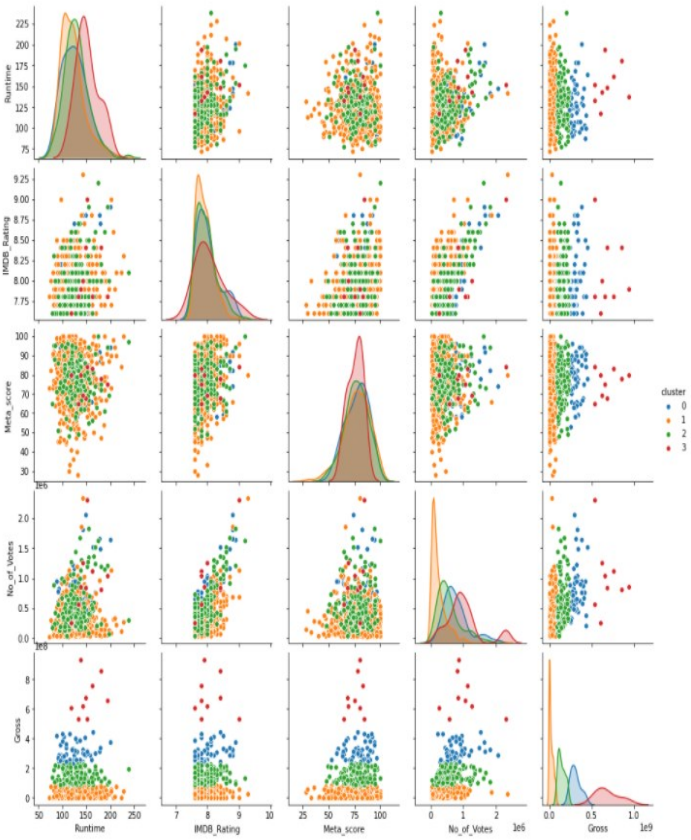


Figure 16. Visualization of cluster distrbution

Some of the data included in cluster 0 can be seen in Figure 17 below. Cluster 0 has 516 rows, a description of the information on cluster 0 data can be seen in Figure 18.

Runtime	IMDB_Rat	Meta_sco	No_of_Votes	Gross	cluster
142.0	9.3	80.0	2343110	28341469.0	0
202.0	9.0	90.0	1129952	57300000.0	0
96.0	9.0	96.0	689845	4360000.0	0
139.0	8.8	66.0	1854740	37030102.0	0
161.0	8.8	90.0	688390	6100000.0	0
146.0	8.7	90.0	1020727	46836394.0	0
132.0	8.6	96.0	552778	53367844.0	0
130.0	8.6	79.0	699256	7563397.0	0
125.0	8.6	96.0	651376	10055859.0	0

Figure 17. Cluster 0 snippet

	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross
count	516.000000	516.000000	516.000000	5.160000e+02	5.160000e+02
mean	121.503876	7.916279	77.211240	2.429573e+05	2.448259e+07
std	25.556828	0.269207	12.991556	2.536258e+05	2.408090e+07
min	72.000000	7.600000	28.000000	2.522900e+04	1.305000e+03
25%	102.750000	7.700000	69.000000	7.368300e+04	3.951812e+06
50%	118.000000	7.900000	78.000000	1.518975e+05	1.558499e+07
75%	133.000000	8.100000	87.000000	3.222990e+05	4.251998e+07
max	228.000000	9.300000	100.000000	2.343110e+06	8.630000e+07

Figure 18. The statistical information of cluster 3

Some of the data included in cluster 1 can be seen in Figure 19 below. Cluster 1 has 9 rows, a description of the information on cluster 1 data can be seen in Figure 20.

Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross	cluster
152.0	9.0	84.0	2303232	534858444.0	1
181.0	8.4	78.0	809955	858373000.0	1
149.0	8.4	68.0	834477	678815482.0	1
143.0	8.0	69.0	1260806	623279547.0	1
138.0	7.9	80.0	860823	936662225.0	1
133.0	7.8	65.0	556608	532177324.0	1
162.0	7.8	83.0	1118998	760507625.0	1
194.0	7.8	75.0	1046089	659325379.0	1
118.0	7.6	80.0	250057	608581744.0	1

Figure 19. Cluster 1 snippet

	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross
count	9.000000	9.000000	9.000000	9.000000e+00	9.000000e+00
mean	152.222222	8.077778	75.777778	1.004561e+06	6.880645e+08
std	23.736985	0.440959	6.924193	5.727595e+05	1.393076e+08
min	118.000000	7.600000	65.000000	2.500570e+05	5.321773e+08
25%	138.000000	7.800000	69.000000	8.099550e+05	6.085817e+08
50%	149.000000	7.900000	78.000000	8.608230e+05	6.593254e+08
75%	162.000000	8.400000	80.000000	1.118998e+06	7.605076e+08
max	194.000000	9.000000	84.000000	2.303232e+06	9.366622e+08

Figure 20. The statistical information of cluster 3

Some of the data included in cluster 2 can be seen in Figure 21 below. Cluster 2 has 133 rows, a description of the information on cluster 2 data can be seen in Figure 22.

Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross	cluster
175.0	9.2	100.0	1620367	134966411.0	2
154.0	8.9	94.0	1826188	107928762.0	2
195.0	8.9	94.0	1213505	96898818.0	2
136.0	8.7	73.0	1676426	171479930.0	2
133.0	8.7	83.0	918088	112000000.0	2
169.0	8.6	74.0	1512360	188020017.0	2
169.0	8.6	91.0	1235804	216540909.0	2
189.0	8.6	61.0	1147794	136801374.0	2
127.0	8.6	65.0	1445096	100125643.0	2

Figure 21. Cluster 2 snippet

	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross
count	133.000000	133.000000	133.000000	1.330000e+02	1.330000e+02
mean	129.451128	7.974436	76.872180	5.724039e+05	1.485255e+08
std	25.564081	0.327676	11.009922	3.709044e+05	4.122817e+07
min	78.000000	7.600000	48.000000	2.765000e+04	8.851350e+07
25%	113.000000	7.700000	70.000000	3.056110e+05	1.117220e+08
50%	128.000000	7.900000	77.000000	4.631880e+05	1.385306e+08
75%	141.000000	8.100000	85.000000	6.996730e+05	1.834172e+08
max	238.000000	9.200000	100.000000	1.826188e+06	2.287787e+08

Figure 22. The statistical information of cluster 3

Some of the data included in cluster 3 can be seen in Figure 23 below. Cluster 3 has 56 rows, a description of the information on cluster 3 data can be seen in Figure 24.

Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross	cluster
201.0	8.9	94.0	1642758	377845905.0	3
148.0	8.8	74.0	2067042	292576195.0	3
178.0	8.8	92.0	1661481	315544750.0	3
142.0	8.8	82.0	1809221	330252182.0	3
179.0	8.7	87.0	1485555	342551365.0	3
124.0	8.7	82.0	1159315	290475067.0	3
121.0	8.6	90.0	1231473	322740140.0	3
122.0	8.5	59.0	939252	335451311.0	3
88.0	8.5	88.0	942045	422783777.0	3

Figure 23. Cluster 3 snippet

	Runtime	IMDB_Rating	Meta_score	No_of_Votes	Gross
count	9.000000	9.000000	9.000000	9.000000e+00	9.000000e+00
mean	152.222222	8.077778	75.777778	1.004561e+06	6.880645e+08
std	23.736985	0.440959	6.924193	5.727595e+05	1.393076e+08
min	118.000000	7.600000	65.000000	2.500570e+05	5.321773e+08
25%	138.000000	7.800000	69.000000	8.099550e+05	6.085817e+08
50%	149.000000	7.900000	78.000000	8.608230e+05	6.593254e+08
75%	162.000000	8.400000	80.000000	1.118998e+06	7.605076e+08
max	194.000000	9.000000	84.000000	2.303232e+06	9.366622e+08

Figure 24. The statistical information of cluster 3

From the information obtained in each cluster, namely clusters 0-3, the mean data can be calculated to see the conclusions of the data in each cluster by calculating the average of the mean data in each cluster.

Cluster 0 : (121 + 7,91 + 77 + 2,4 + 2,44) / 5 = **42,15**

Cluster 1 : (152 + 8,07 + 75,77 + 1,0 +6,88) / 5 = **48,74**

Cluster 2 : (129 + 7,97 + 78,87 + 5,7 + 1,48) / 5 = **44,604**

Cluster 3 : (125 + 8,01 + 77,57 + 7,8 + 3,12) / 5 = **44,3**

From the above calculations, it can be concluded that the cluster that has the highest score or value obtained from runtime, IMDB_Rating, Meta_score, No_of_votes, and Gross is cluster 1 followed by cluster, cluster 3 and finally cluster 0. So, it can be said that cluster 1 contains a list the most popular film because it has the highest rating score.

The last step is to evaluate the cluster data obtained by using the confusion matrix. Obtained an accuracy of 100% by using the code in table 2.

Table 5. Confusion Matrix Code

```
from sklearn.metrics import confusion_matrix,classification_report

print("ConfusionMatrix",confusion_matrix(data['cluster'],kmeans.labels_))

print(classification_report(data['cluster'],kmeans.labels_))
```

The results of the evaluation can be seen in Figure 24.

```
ConfusionMatrix [[135   0   0   0   0   0   0   0   0   0]
 [  0  13   0   0   0   0   0   0   0   0]
 [  0   0  70   0   0   0   0   0   0   0]
 [  0   0   0   3   0   0   0   0   0   0]
 [  0   0   0   0  38   0   0   0   0   0]
 [  0   0   0   0   0 101   0   0   0   0]
 [  0   0   0   0   0   0  26   0   0   0]
 [  0   0   0   0   0   0   0   6   0   0]
 [  0   0   0   0   0   0   0   0 282   0]
 [  0   0   0   0   0   0   0   0   0  40]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       135
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        70
           3       1.00      1.00      1.00         3
           4       1.00      1.00      1.00        38
           5       1.00      1.00      1.00       101
           6       1.00      1.00      1.00        26
           7       1.00      1.00      1.00         6
           8       1.00      1.00      1.00       282
           9       1.00      1.00      1.00        40

    accuracy                           1.00       714
   macro avg       1.00      1.00      1.00       714
weighted avg       1.00      1.00      1.00       714
```

Figure 24. Accuracy using Confusion Matrix

V. CONCLUSIONS

From the results of clustering, there are four cluster groups using davies bouldin index with the first cluster character (Cluster 1) being the highest cluster score rating with a value of 48.74, then followed by Cluster 2 with a value of 44.60, and the highest after that there is Cluster 3 with a value of 44.3. For clusters the lowest score is in Cluster 0 which has a value of 42.15. The determination of the cluster name using Python is done randomly, so it can be seen in the discussion that the cluster that stores the lowest number is out of sync, this is because the program running process is carried out in stages or not all at once. at the same time, in the first experiment running a cluster program called cluster 2 and in the second experiment running a cluster program called cluster 0, but this actually doesn't have much effect because the data generated remains the same, it's just that the cluster names are different or randomly. Determination of values in each cluster is obtained from runtime calculations, IMDB_Rating, Meta_score, No_of_votes, and Gross on the data used. The results of the cluster obtained in the K-Means algorithm show the relationship between variables, namely runtime, IMDB_Rating, Meta_score, No_of_votes, and Gross. This study resulted that cluster 1 is the score with the highest rating, while cluster 0 is the score with the lowest

rating. From the results of the calculation of the cluster obtained an accuracy of 100%.

REFERENCES

[1] H. Ardiyanti, "Perfilman Indonesia: Perkembangan dan Kebijakan, Sebuah Telaah dari Perspektif Industri Budaya," *Kajian*, vol. 22, no. 2, pp. 163–179, 2017, [Online]. Available: http://jurnal.dpr.go.id/index.php/kajian/article/view/1521/789.

[2] G. N. H. Pratama, "Sistem Rekomendasi Film Menggunakan Metode Content Based Filtering," vol. 5, no. 6, 2019, [Online]. Available: http://e-journal.uajy.ac.id/20600/.

[3] J. Fang and W. Xiong, "Impact of digital technology and internet to film industry," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 768, no. 7, 2020, doi: 10.1088/1757-899X/768/7/072112.

[4] B. G. Sudarsono, M. I. Leo, A. Santoso, and F. Hendrawan, "Analisis Data Mining Data Netflix Menggunakan Aplikasi Rapid Miner," *JBASE - J. Bus. Audit Inf. Syst.*, vol. 4, no. 1, pp. 13–21, 2021, doi: 10.30813/jbase.v4i1.2729.

[5] I. F. Ashari, "Implementation of Cyber-Physical-Social System Based on Service Oriented Architecture in Smart Tourism," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 66–73, 2020, doi: 10.30871/jaic.v4i1.2077.

[6] N. Wakhidah, "Clustering Menggunakan K-Means Algorithm," *J. Transform.*, vol. 8, no. 1, p. 33, 2010, doi: 10.26623/transformatika.v8i1.45.

[7] R. T. Vulandari, W. L. Y. Saptomo, and D. W. Aditama, "Application of K-Means Clustering in Mapping of Central Java Crime Area," *Indones. J. Appl. Stat.*, vol. 3, no. 1, p. 38, 2020, doi: 10.13057/ijas.v3i1.40984.

[8] Z. I. Alfianti, "Pengelompokkan Wilayah penyebaran COVID-19 di Kabupaten Karawang Menggunakan Algoritma K-Means," *J. Ilm. Inform. Komput.*, vol. 26, no. 2, pp. 111–122, 2020.

[9] S. Handoko, F. Fauziah, and E. T. E. Handayani, "Implementasi Data Mining Untuk Menentukan Tingkat Penjualan Paket Data Telkomsel Menggunakan Metode K-Means Clustering," *J. Ilm. Teknol. dan Rekayasa*, vol. 25, no. 1, pp. 76–88, 2020, doi: 10.35760/tr.2020.v25i1.2677.

[10] D. Jollyta, S. Efendi, M. Zarlis, and H. Mawengkang, "Optimasi Cluster Pada Data Stunting: Teknik Evaluasi Cluster Sum of Square Error dan Davies Bouldin Index," *Pros. Semin. Nas. Ris. Inf. Sci.*, vol. 1, no. September, p. 918, 2019, doi: 10.30645/senaris.v1i0.100.

[11] E. Muningsih, I. Maryani, and V. R. Handayani, "Penerapan Metode K-Means dan Optimasi Jumlah Cluster dengan Index Davies Bouldin untuk Clustering Propinsi Berdasarkan Potensi Desa," *Evolusi J. Sains dan Manaj.*, vol. 9, no. 1, pp. 95–100, 2021.

[12] N. E. Saputra, K. D. Tania, and R. I. Heroza, "Penerapan Knowledge Management System (KMS) Menggunakan Teknik Knowledge Data Discovery (KDD) Pada PT PLN (Persero) WS2JB Rayon Kayu Agung," *J. Sist. Inf.*, vol. 8, no. 2, pp. 1038–1055, 2016.

[13] M. R. Muttaqin and M. Defriani, "Algoritma K-Means untuk Pengelompokan Topik Skripsi Mahasiswa," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 121–129, 2020, doi: 10.33096/ilkom.v12i2.542.121-129.

[14] Asroni and R. Adrian, "Penerapan Metode K-Means Untuk Clustering Mahasiswa Berdasarkan Nilai Akademik Dengan Weka Interface Studi Kasus Pada Jurusan Teknik Informatika UMM Magelang," *J. Ilm. Semesta Tek.*, vol. 18, no. 1, pp. 76–82, 2015.

[15] A. Almayda and S. Saepudin, "Penerapan Data Mining K-Means Clustering Untuk Menglelompokkan Berbagai Jenis Merek Smartphone," in *SISMATIK (Seminar Nasional Sistem Informasi dan Manajemen Informatika)*, 2021, pp. 241–249.

[16] N. Dwitri, J. A. Tampubolon, S. Prayoga, F. Ilmi Zer, and D. Hartama, "Penerapan Algoritma K-Means Dalam Menentukan Tingkat Penyebaran Pandemi COVID-19 di Indonesia," *Jti (Jurnal Teknol. Informasi)*, vol. 4, no. 1, pp. 101–105, 2020.

[17] I. F. Ashari, "The Evaluation of Image Messages in MP3 Audio Steganography Using Modified Low-Bit Encoding," *Telematika*, vol. 15, 2021.