JOURNAL LA MULTIAPP

VOL. 05 ISSUE 03 (270-285), 2024
DOI: 10.37899/journallamultiapp.v5i3.1479

Measurement of Centroid Distance in Determining Stunting Clusters

Muhammad Taufik Hakim Lubis[1], Muhammad Siddik Hasibuan[2]

[1]*Ilmu Komputer, Fakultas Sains dan Teknologi,*
Universitas Islam Negeri Sumatera Utara, Medan Indonesia

**Corresponding Author: Muhammad Taufik Hakim Lubis*
Email: taufikhakimlbs007@gmail.com

Article Info

Article history:
Received 27 June 2024
Received in revised form 24 July 2024
Accepted 4 August 2024

Keywords:
Stunting
K-Means
Clustering
Euclidean Distance
Manhattan Distance
Sum of Squared Errors
Mean Squared Error

Abstract

This study evaluates the effectiveness of distance measurement methods in the K-Means clustering algorithm for determining stunting clusters by comparing Euclidean and Manhattan distances. The goal is to obtain optimal cluster centroids and the closest distances within each cluster. The study uses a sample of 552 records with 3 attributes. The process begins with applying the K-Means algorithm, followed by distance measurement using Euclidean and Manhattan methods. Iterations are performed until optimal results are achieved. Evaluation is conducted using Sum of Squared Errors (SSE) to assess the total error within clusters and Mean Squared Error (MSE) to calculate the average nearest distance within clusters. The results indicate that both SSE and MSE methods are effective in identifying cluster quality and provide insights into the accuracy and effectiveness of Euclidean and Manhattan methods in clustering.

Introduction

As technology develops and develops very rapidly in processing data with the aim of knowing patterns so as to obtain information stored from that data. For this case, Clustering is the process of grouping data objects into several scattered clusters so that the data in each cluster is combined into a group of data where the similarity of the data is identical. The K-Means algorithm is a partition clustering method that is capable of grouping data and partitioning data into one or more clusters that have the same characteristics (Fadilah et al., 2022).

A cluster is a collection of data objects that have the same characteristics as another but are in the same cluster and if the data from these characteristics is different from the data object then the data is in another cluster . The cluster center point (centroid) is the starting point that begins with grouping clusters using the K-Means algorithm . The stages in grouping data are carried out by calculating the distance from the initial cluster center point (centroid) as the midpoint of cluster formation . The output produced from the clustering process using the K-Means method plays a very important role in selecting the initial cluster center point (centroid).

In selecting the cluster center point (centroid) The initial process will be carried out randomly and an iterative process will be carried out to determine the distance from several data points to the nearest centroid and before that the number of clusters (k) has been determined before analysis (Retno, 2019). In the stage of calculating the difference in data distance from each cluster center point (centroid), the K-Means algorithm repeats this stage until the resulting data does not experience cluster movement or until the end of the specified iteration limit. The application of the K-Means algorithm produces a midpoint or centroid value from the data obtained in accordance with clustering provisions.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

The initial Clustering process provides an initial value for the Centroid in carrying out the analysis (Nasution & Hasibuan, 2020) . To obtain the same data with the aim of developing strong algorithms and data mining classification and grouping functions , distance calculations such as Euclidean and Manhattan are very helpful in this regard, as well as measuring similarity and regularity between data items (Widodo et al., 2021) .

K-Means algorithm is one of the Clustering Algorithms that is widely used in various fields such as education, health, social, biology and computer science. In Indonesia, which is currently experiencing health problems, especially stunting, which is a crucial public health issue and is experiencing a very high prevalence rate that exceeds WHO standards . The problem of stunting in Indonesia is considered a problem that cannot be ignored and must be addressed because it can cause a decline in the level of public health, both long and short term. By using the K ¬- Means algorithm which can process data and determine stunting clustering, this can help in the process of analyzing stunting cases (Ranjawali et al., 2023).

Based on the problems that have been described, this research tries to provide a solution to these problems. The method applied in this research is the K-Means Method. In the K-Means research , data on stunting toddlers in Deli Serdang Regency will be grouped according to the level of occurrence of stunting cases and will measure the distance from the cluster center point or centroid . So the author took this research with the title " Measuring Centroid Distances in Determining Stunting Clusters".

Methods

The simple meaning of research is an effort to obtain facts by collecting and analyzing data (information) in a clear, thorough, systematic manner that can be accounted for. The steps taken by the author in forming a research framework are as follows:

The first step that researchers need to take is the data collection method using quantitative research methods. This research focuses on the results of interviews and observations, reports, journals, news and references from previous research results. Data collection was carried out by directly visiting the location of the research object, namely BKKBN Deli Serdang.

This stage is necessary because after collecting data, there is an initial process in data analysis which aims to clean, organize, change the data so that it is easy to understand and be processed by algorithms in processing the data . In the research "measuring centroid distances in determining stunting clusters", in this case *Preprocessing* plays an important role in processing the data obtained from researchers. The importance *of preprocessing* in ensuring the model is relevant to *the dataset* and obtaining accurate *cluster results regarding determining stunting clusters* based on their grouping.

K-Means Method

This research uses the *clustering method* with the choice of the K-Means algorithm to calculate the amount of data. After that, the distance calculation process is carried out at each cluster center point (*centroid*) using *Euclidean Distance* and *Manhattan Distance calculations* . Then implemented with the *Jupyter Notebook application* to present measurements of *centroid distances* in *stunting clusters* .

Testing

This stage is carried out by testing or executing the data using the *Euclidean* and *Manhattan distance calculation methods* . After that, the data is executed with the *Jupyter Notebook application* with the aim of getting accurate and accurate results.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Research plan

In the *clustering process, the K-Means* algorithm will be applied to help complete the data processing stage. This research uses the *Jupyter Notebook application* to determine *stunting clusters* and presents distance measurements from *the Centroid* . This makes it easier for the community health center or local government to determine the level of toddlers experiencing *stunting* .

Flow chart

A flowchart is a diagram that represents an algorithm, workflow or process in creating a program. Flowcharts are depicted in the form of symbols connected to each other with lines or arrows (Rosaly & Prasetyo, 2019) . By using a flowchart as the flow of a program in research, it will be clearer, more concise and reduce errors in interpretation. The flowchart of the *K-Means algorithm* can be shown in the image below:

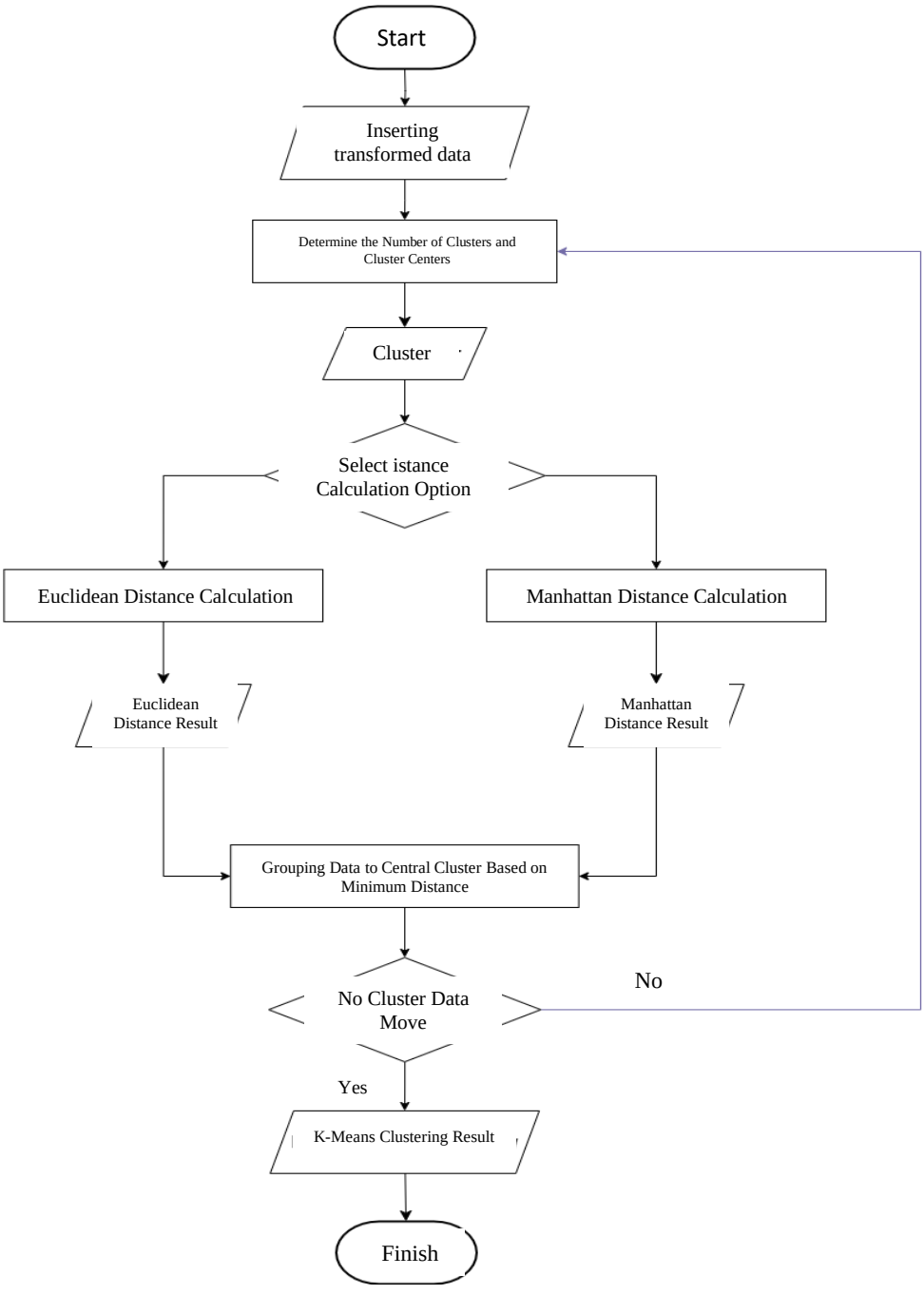

Figure 1. the K-Means Algorithm Process Flow *Clustering*

ISSN: 2716-3865 (Print), 2721-1290 (Online)

The flowchart starts, then enters the data that has been transformed/modified, after that determines the number of clusters and cluster center points, then the cluster results are calculated by comparing *the Euclidean distance calculations Distance* and *Manhattan Distance* , the results of these two methods are grouped into cluster centers based on the minimum or closest distance. If there is data that has moved *clusters then carry out the data mining* processing process again and if there is no data that has moved *clusters* then continue by presenting the results of the *K-Means Clustering Algorithm* and finish.

Results and Discussion

Results of Implementation of the K-Means Method

In the implementation stage of the *K-Means method* , determining the stunting cluster is first done by collecting data and several attributes such as age, weight and height as well as a *pre-processing process* including filtering the required data and normalizing the data for each number in each attribute. The next step, the model using the *K-Means method* is built by determining the number of clusters followed by calculating the centroid distance to measure the centroid distance. This model is used for training data and test data as an evaluation of each data to measure the level of accuracy and identify if errors occur. If the model has been well trained, the final step is to use it to determine clusters in each data and measure the best accuracy of centroid distances using the *Euclidean distance* and *Manhattan distance calculation methods* in each data. In the distance calculation process, each method will compare the best level of accuracy in order to obtain well-accurate analysis results. Therefore, utilizing the *K-Means algorithm* and the distance calculation method can provide information on grouping toddlers who are *stunted* and predict the future to minimize the level of *stunting* in toddlers.

Determination of K Values

The process of determining the number of *clusters* is determined by the number k=3, after that it goes through an algorithm calculation in *K-Means* based on the number of k for grouping data, then testing is carried out to get optimal accuracy results from distance calculations and comparing each of the distance calculation methods.

Initial Centroid Determination

Before calculating the distance, the first step is to determine the value of the cluster center *point or* initial *centroid* randomly . To get the *centroid value* , you can take the smallest number, middle number/ *mean* and largest number from each data attribute randomly. The *random* values obtained are in the 10th, 136th and 100th data as the initial *centroid values* c1, c2 and c3. Details of the initial *centroid data* in table consist of *centroid* , age, weight and height with a value of c=3, namely with many groups of three.

Table 1. Initial *centroid* value

CENTROID	AGE	HEAVY	TALL
c1	1	1	1
c2	0.491803279	0.533333333	0.791578947
c3	0	0	0.452631579

Calculating the Closest Distance to the *Cluster Center*

Calculation of the closest distance to the *cluster center point* has two similarities in distance comparison analysis. The results of this analysis were obtained for grouping *cluster data* by getting the same *cluster results* with different distance calculations.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

The distance calculations used to determine the closest distance to each *cluster are Euclidean distance* and *Manhattan distance* calculations. Following are the results of calculating the overall distance from each data which has three data groups/ *clusters* which can be seen in table below.

Table 2. Results of calculating *the Euclidean distance* from the initial *centroid*

Data i	cluster 1	cluster 2	cluster 3
1	0.129728283	0.596052317	1.393191891
2	0.196721311	0.598526561	1,394585544
3	0,196721311	0,598526561	1,394585544
4	0,238692171	0,551636493	1,334179625
5	0,557001926	0,46138732	1,119984418
6	0,323380189	0,435814902	1,215804237
7	0,3290994	0,400851667	1,194631531
8	0,196721311	0,598526561	1,394585544
9	0,304246172	0,470015608	1,252237632
10	0	0,720750318	1,516447226
11	0,222678654	0,570982735	1,353551128
12	0	0,720750318	1,516447226
13	0,317420368	0,427535832	1,217188549
14	0,11798061	0,627931757	1,421721202
15	0,736521012	0,044087163	0,794276127
…	…	…	…
…	…	…	…
551	1.010108035	0.291832319	0.531219092
552	0.924191844	0.205368201	0.607991967

In the next stage, this is done by determining a new *centroid* on *the Euclidean distance* by calculating the middle/ *mean value for each cluster* data value. The following are the results of the new *centroid* which can be seen in table below:

Table 3. New Centroid Center Euclidean Distance

Centroid	age	heavy	tall
c1	0.923175828	0.868006536	0.970681115
c2	0.456780924	0.482964349	0.790861244
c3	0.088654012	0.191491228	0.565706371

After obtaining the new centroid value and re-calculating the distance value of *the centroid data* . After that, the iteration continues until the *cluster member data values* do not change and there is no movement of *cluster data* to other *clusters* .

Manhattan **Distance Calculation**

Manhattan distance calculation method was used by obtaining 552 data. Following are the results of calculating the overall distance from each data which has three data groups/ *clusters* which can be seen below.

Table 4. Results of calculating *the Manhattan distance* from the initial *centroid*

Data i	*cluster* 1	*cluster* 2	*cluster 3*
1	0.192493529	0.990790912	2.354874892
2	0.196721311	0.986563129	2.35064711

ISSN: 2716-3865 (Print), 2721-1290 (Online)

3	0,196721311	0,986563129	2,35064711
4	0,293039977	0,890244464	2,254328444
5	0,773039977	0,507788323	1,774328444
6	0,505884383	0,677400058	2,041484038
7	0,523738855	0,659545585	2,023629566
8	0,196721311	0,986563129	2,35064711
9	0,426333046	0,756951395	2,121035375
10	0	1,183284441	2,547368421
11	0,263313201	0,91997124	2,28405522
12	0	1,183284441	2,547368421
13	0,483232672	0,700051769	2,064135749
14	0,146120219	1,037164222	2,401248202
15	1,186597642	0,062260569	1,360770779
…	…	…	…
…	…	…	…
551	1.635599655	0.452315214	0.911768766
552	1.499246477	0.315962036	1.048121944

To determine the new *centroid center* for *Manhattan distance* in the same way as for *Euclidean distance,* namely finding the *mean value* for each data *cluster* using *the cluster center*. The following are the results of the new *centroid* which can be seen in table below:

Table 5. New centroid center Manhattan Distance

Centroid	Age	Heavy	Tall
c1	0.917875158	0.866923077	0.971022267
c2	0.455060871	0.480389634	0.789770001
c3	0.087213115	0.191644444	0.562863158

Algorithm Loop and Results

The final stage in the data grouping process is carried out using the *K-Means algorithm* for repeated iterations so that the data does not experience *cluster movements* or changes in the data and produces final data or accurate data. In this research, the iteration went through two *K-Means algorithm calculations* which gave the same *cluster data values as the previous iteration.* So iterates again until the *cluster data value* reaches the final result. The final results of calculating *the Euclidean distance* and *Manhattan distance* in grouping data in the last iteration can be seen in tables 4.10 and 4.11. Before that, the following categories for determining the number of each *cluster member* can be seen in table below.

Tavle 6. Number of *Clusters*

Clusters	Information
C1	Light
C2	Critical
C3	Awfully

From the results of the *Euclidean distance calculation,* we get the results *of clustering* each data with a minimum distance. In *cluster* one, 169 data were obtained, then in *cluster* two there were 208 data and for *cluster* three there were 175 data. With a total of 552 data, the results of *the Euclidean distance calculation* in the last iteration can be seen in table below.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Table 7. *Euclidean Distance* Calculation Results from the Last Iteration

Data i	Cluster 1	Cluster 2	Cluster 3	Choice Clusters
1	0.147328513	0.629677358	1.032219928	1
2	0.232606412	0.633768363	1.024606919	1
3	0.232606412	0.633768363	1.024606919	1
4	0.113903873	0.580808643	0.983389036	1
5	0,345442218	0,479805406	0,819156043	1
6	0,072575748	0,466998503	0,866508179	1
7	0,057699165	0,433127951	0,836615329	1
8	0,232606412	0,633768363	1,024606919	1
9	0,074957197	0,498379755	0,900865489	1
10	0,274037358	0,755626669	1,156002333	1
11	0,128228452	0,600567492	1,003017852	1
12	0,274037358	0,755626669	1,156002333	1
13	0,048025204	0,457868641	0,861690575	1
14	0,170489808	0,660568387	1,064049345	1
15	0,476024184	0,04215788	0,424806185	2
…	…	…	…	…
…	…	…	…	…
551	0,747702834	0,257398081	0,159541591	3
552	0,659477721	0,168971811	0,24120025	2

In the last iteration, the *centroid center is obtained* for grouping each data and determining the number of *cluster results* . The results of the final *centroid center in the Euclidean* distance calculation process show that the *centroid value* at C1 has the highest value between C2 and C3. It can be concluded from this data that C1 is in the predetermined category with the "mild" category, C2 is in the "Severe" category and then C3 is in the "Very Severe" category from the clustering results. Following are the results of the last iteration of the *centroid center euclidean distance* which can be seen in table below.

Table 8. *Centroid* Center *Euclidean Distance* in the Last Iteration

Centroid	Age	Heavy	Tall
C1	0.848287904	0.779408284	0.941526004
C2	0.471311475	0.497621795	0.802322874
C3	0.173395785	0.277104762	0.638766917

Next, the distance calculation results for *the Manhattan distance* are obtained from each data *cluster* with a minimum distance. The first *cluster* produced 163 data, then in the second *cluster* 214 data and in the third *cluster* 175 data with a total of 552 data. The results of the last iteration of the *Manhattan distance calculation* can be seen in table below.

Table 9. *Manhattan* Distance Calculation Results from Iteration Final

Data i	Cluster 1	Cluster 2	Cluster 3	Optional Clusters
1	0.223719852	1.024836625	1,718403221	1
2	0.314433234	1,020608842	1,714175439	1
3	0.314433234	1,020608842	1,714175439	1
4	0.16599549	0.924290177	1.617856773	1
5	0.556967398	0.499481847	1.137856773	2

ISSN: 2716-3865 (Print), 2721-1290 (Online)

6	0.125877378	0,71144577	1,405012367	1
7	0,107525474	0,693591298	1,387157895	1
8	0,314433234	1,020608842	1,714175439	1
9	0,126035755	0,790997108	1,484563704	1
10	0,416213381	1,217330154	1,91089675	1
11	0,169055599	0,954016953	1,647583549	1
12	0,416213381	1,217330154	1,91089675	1
13	0,071836856	0,734097482	1,427664078	1
14	0,270093163	1,071209935	1,764776531	1
15	0,77038426	0,070567045	0,724299108	2
…	…	…	…	…
…	…	…	…	…
551	1,219386274	0,418269501	0.275297095	3
552	1.083033096	0.281916323	0.411650273	2

After determining the results of the *Manhattan distance calculation, the centroid center* for the last iteration can also be determined . By producing the final *centroid value.* It can be seen that the value of C1 is the highest value of C2 and C3. So the C1 value can be concluded in the "mild" category, the C2 value in the "Severe" category and for C3 in the "Very Severe" category. The *centroid* center in the *Manhattan distance calculation* in the last iteration can be seen in table below.

Table 10. Manhattan Centroid Center Distance in the Last Iteration

Centroid	Age	Heavy	Tall
c1	0.850749271	0.78807771	0.944959638
c2	0.47893366	0.499791277	0.803944909
c3	0.17470726	0.276038095	0.638357895

Python Implementation (*Jupyter Notebook*)

After designing and developing the system, the next stage is implementing *Python* . The aim of implementing *Python* with *Jupyter Notebook* is to adjust or evaluate whether it meets the expectations of the system that has been created by the researcher.

Data Import View

Import View is an important first step in data analysis using *Jupyter Notebook with the Python* programming language . To send data, the *Pandas library is used* by inputting the lines of code listed in *the Jupyter Notebook* . For use of *Jupyter Notebook* it is important to ensure that the data sent is in the correct directory. At this stage, the dataset is in EXCEL format which contains 552 *records* and 6 attributes.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

```
#Membaca File excel
data = pd.read_excel("euc_man.xlsx")
data
```

	NIK	NAMA	JK	UMUR	BERAT	TINGGI
0	120725**********	AY	L	56	16.0	99.0
1	121220**********	MR	L	49	17.5	100.0
2	120722**********	FN	P	49	17.5	100.0
3	120731**********	ff	P	58	14.0	99.0
4	120709**********	AA	L	58	9.8	80.0
...	...	...	...	...	...	...
547	121280**********	AN	L	18	8.0	75.0
548	060622**********	MG	L	8	6.5	64.0
549	121230**********	BA	P	26	9.8	81.0
550	120723**********	RS	L	16	7.9	75.5
551	120726**********	AL	L	21	8.4	77.5

552 rows × 6 columns

Figure 2. Data *Import* View

Data *Pre-processing* View

pre-processing process begins by deleting each column that is not needed in *the Dataframe* . By deleting columns such as 'NIK', 'NAME' and gender or 'JK'. The following can be seen in Figure 4.2 below.

```
#Menampilkan Atribut dan Menghapus Atribut Yang Tidak Diperlukan
features = ['UMUR','BERAT','TINGGI']
data = data.dropna(subset=features)
klaster = data[features].copy()
klaster
```

	UMUR	BERAT	TINGGI
0	56	16.0	99.0
1	49	17.5	100.0
2	49	17.5	100.0
3	58	14.0	99.0
4	58	9.8	80.0
...	...	...	...
547	18	8.0	75.0
548	8	6.5	64.0
549	26	9.8	81.0
550	16	7.9	75.5
551	21	8.4	77.5

552 rows × 3 columns

Figure 3. Required Attribute Display

Next, normalize each data attribute into a form that is appropriate to the *clustering process* . This aims to prepare data that is suitable for the machine learning process, thereby increasing the accuracy and performance of the model. The data normalization process can be seen in Figure 4.3 below.

```
#Proses Normalisasi Data
scaler = MinMaxScaler()
scaler.fit(klaster[['UMUR','BERAT','TINGGI']])
klaster[['UMUR','BERAT','TINGGI']] = scaler.transform(klaster[['UMUR','BERAT','TINGGI']])
klaster
```


ISSN: 2716-3865 (Print), 2721-1290 (Online)

	UMUR	BERAT	TINGGI
0	0.918033	0.900000	0.989474
1	0.803279	1.000000	1.000000
2	0.803279	1.000000	1.000000
3	0.950820	0.766667	0.989474
4	0.950820	0.486667	0.789474
...	...	...	...
547	0.295082	0.366667	0.736842
548	0.131148	0.266667	0.621053
549	0.426230	0.486667	0.800000
550	0.262295	0.360000	0.742105
551	0.344262	0.393333	0.763158

552 rows × 3 columns

Figure 4. Normalization Process Display Data

Clustering View

Clustering process is carried out using *Jupyter Notebook in the Python* programming language. The *clustering* process begins by determining the number of data groups What is desired is to create a *K-Means object.*

```
#Menentukan Banyaknya Cluster
kmeans = KMeans(n_clusters=3)
kmeans.fit(klaster[['UMUR','BERAT','TINGGI']])
```

```
KMeans
KMeans(n_clusters=3)
```

Figure 5. *K-Means* Process View

After determining the number of *clusters* that will be grouped into the data, the process that will continue is to calculate the distance from each data by calculating the *Euclidean distance* and *Manhattan distance* .

```
#Menghitung Jarak Dengan Euclidean Distance
def euclid(klaster,centroids):
    distances = centroids.apply(lambda x: np.sqrt(((klaster-x)**2).sum(axis=1)))
    return distances.idxmin(axis=1)
```

Figure 6. Euclidean Distance Calculation Process

```
#Proses Perhitungan Jarak Manhattan Distance
def manhat(klaster, centroidt):
    distances = pd.DataFrame()
    for col in centroidt.columns:
        distances[col] = klaster.apply(lambda row: abs(row - centroidt[col]).sum(), axis=1)
        min_indices = distances.idxmin(axis=1)
    return min_indices
```

Figure 7. Manhattan Distance Calculation Process

Evaluation View

The evaluation view is that model evaluation is carried out using *the Sum of Squared Errors* (SSE) which is called inertia in the context of *K-Means Clustering* . SSE is an evaluation metric to measure how far the data points in a *cluster are from the centroid* center point. Below, the highest to lowest SSE *Euclidean distance values* can be seen in Figure 4.7 below.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

```
inertias = []
k_range = range(1,10)
for k in k_range:
    km = KMeans(n_clusters=k)
    km.fit(klaster)
    sse_ = km.inertia_
    inertias.append(sse_)
```

```
print('Nilai SSE')
inertias
```

```
Nilai SSE
[429.8391347219467,
 127.54606586829607,
 16.90792640466273,
 12.064505741062096,
 8.949272815222974,
 6.554636494388472,
 5.726234042633598,
 4.992199775866506,
 4.419473700900811]
```

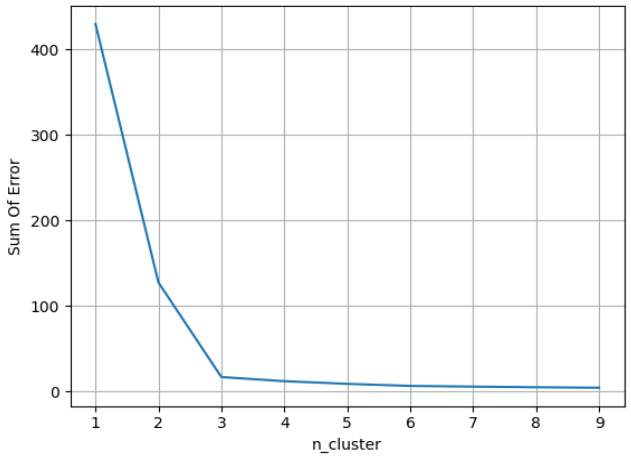

Figure 8. Euclidean Distance Evaluation Display

From the results of the SSE values in Figure 4.7 above, the highest value is 429.83913472119467 followed by a value of 127.54606586829607 after that up to a value of 16.90792640466273 which is the limit value for determining n _clusters or the number of *clusters* and the optimal result from Figure 4.7 is three *clusters* at *Euclidean distance* .

Furthermore, the highest to lowest SSE *Manhattan distance values* can be seen in Figure 4.8 below.

```
inertias = []
k_range = range(1,10)
for k in k_range:
    km = KMeans(n_clusters=k)
    km.fit(klaster)
    sse_ = km.inertia_
    inertias.append(sse_)
```

```
print('Nilai SSE')
inertias
```

```
Nilai SSE
[423.643482548034,
 129.74239383865506,
 17.099894093634045,
 12.539712425201554,
 9.424981745241238,
 6.651059082051848,
 5.861548714087825,
 5.114343045228228,
 4.468548523612905]
```

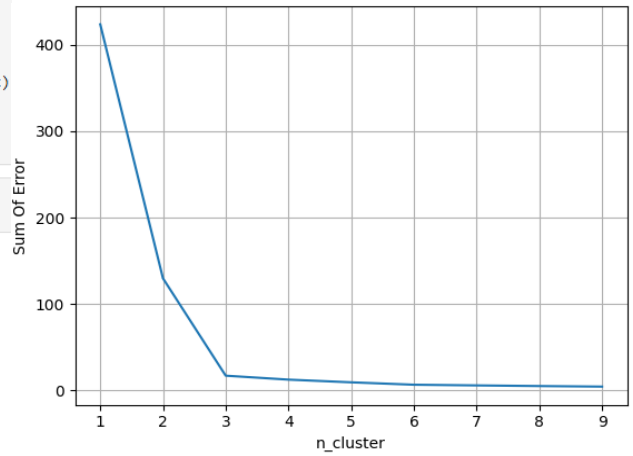

Figure 9. Manhattan Distance Evaluation View

From the results of the SSE values in Figure 4.8 above, the highest value is 423.643482548034 followed by a value of 129.74239383865506 after that up to a value of 17.099894093634045 which is the limit value in determining n _clusters or the number of *clusters* and the optimal result from Figure 4.8 is three *clusters* at *the Manhattan distance* .

Result Report Display

The results report display is the results obtained from the *Jupyter Notebook library* which will be saved as an excel file.

```
#Menyimpan Dataframe data Kedalam File Excel (xlsx)
data.to_excel('euclidean_distances.xlsx', index=False, sheet_name='Sheet1')
```

Figure 10. Euclidean Distance Results Report Display

```
#Menyimpan Dataframe df Kedalam File Excel (xlsx)
df.to_excel('manhattan_distances.xlsx', index=False, sheet_name='Sheet1')
```

Figure 11. Manhattan Distance Results Report Display

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Testing

Testing in this research uses the *K-Means method* by calculating *the Euclidean distance* and *Manhattan distance* to determine the closest centroid distance from each data *cluster* . In displaying *K-Means* data grouping using *Euclidean distance* and *Manhattan distance* calculations . Following are the results of iteration testing of each distance calculation, which can be seen in figures 4.11 and 4.12.

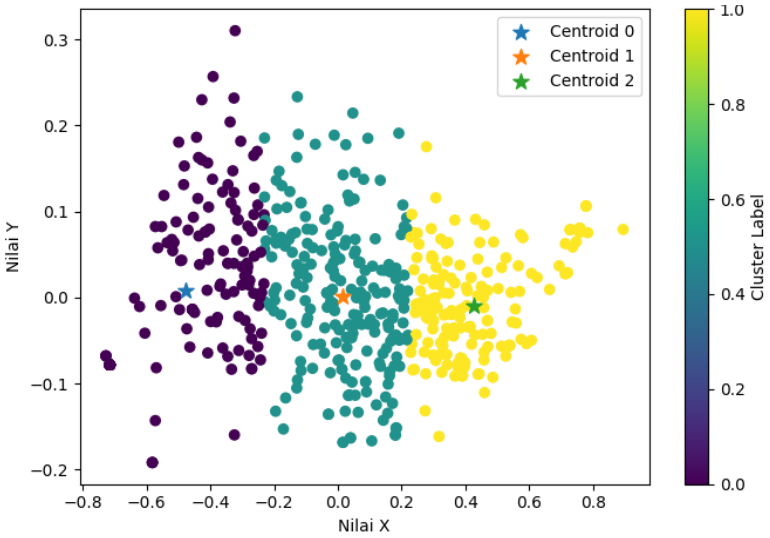

Figure 12. Cluster Iteration Results on Euclidean Distance

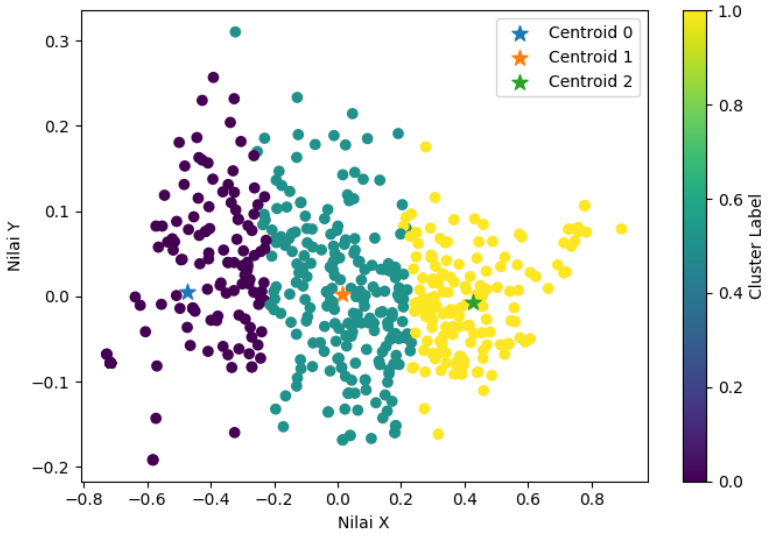

Figure 13. Cluster Iteration Results on Manhattan Distance

Based on figures 4.11 and 4.12, *the cluster* is divided into three, namely the first *cluster* is purple, followed by the second *cluster* is blue and the third *cluster* is yellow. Next, *the cluster results* from each distance calculation can be concluded The comparison of iterations shows that the iterations in calculating the *Euclidean distance* with a total of 8 iterations are the same as *the Manhattan distance* with 8 iterations. This shows that the distance calculations *for Euclidean distance* and *Manhattan distance* are the same in terms of iteration.

Following are the results of testing the closest distance from *Euclidean distance* and *Manhattan distance* which can be seen in table below.

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Table 11. Euclidean Distance and Manhattan Distance Test Results

NO	VARIABLES	EUCLIDEAN DISTANCE	MANHATTAN DISTANCE	CLOSEST DISTANCE
		Iteration 8	Iteration 8	
CLUSTER 1	AGE	0.848287904	0.850749271	0.848287904
	HEAVY	0.779408284	0.78807771	0.779408284
	TALL	0.941526004	0.944959638	0.941526004
CLUSTER 2	AGE	0.471311475	0.47893366	0.471311475
	WEIGHT	0.497621795	0.499791277	0.497621795
	HEIGHT	0.802322874	0.803944909	0.802322874
CLUSTER 3	AGE	0.173395785	0.17470726	0.173395785
	WEIGHT	0.277104762	0.276038095	0.276038095
	HEIGHT	0.638766917	0.638357895	0.638357895

Based on table 4.14 above, it can be concluded from *cluster* 1 for the variables age, weight and height in the *Euclidean distance calculation* that the centroid values are 0.848288, 0.779408, 0.941526 , whereas with use *manhattan distance* mark *centroid* obtained are 0.850749, 0.788078, 0.944960. Comparison results next to *cluster* 2 for variable age , weight and height at *Euclidean distance* are mark *centroid* amounted to 0.471311, 0.497622, 0.802323, while with use *manhattan distance* value *The centroids* obtained are 0.478934, 0.499791, 0.803945 and the results comparison finally in *cluster* 3 , namely with variable age, weight and height in the calculation process distance *Euclidean distance* with mark *centroid* of 0.173396, 0.277105, 0.638767, while in the calculation distance *manhattan distance* mark *centroid* obtained of 0.174707, 0.276038, 0.638358. From the results comparison in determine accuracy best with measurement distance *centroid* more dominated by calculations distance *euclidean distance* .

Following This results testing use *Mean Squared Error* with compare *euclidean distance* and *manhattan distance* can be seen in table 4.15 below This .

Table 12. Mean Squared Error (MSE) Test Results

ITERATION	Mean Squared Error(MSE)	
	EUCLIDEAN DISTANCE	MANHATTAN DISTANCE
1	0,061861306	0,148646255
2	0,037054203	0,087677999
3	0,031482287	0,07475772
4	0,030747202	0,072924851
5	0,030642513	0,072715327
6	0,030633207	0,072651345
7	0,030631203	0,072620172
8	0,030630301	0,07263329
Avg MSE	0,035460278	0,08432837

Based on from results testing table 4.15 above , value *mean squared error* on *Euclidean distance* dominates with mark smallest with iterations 1 to 8 , namely , 0.061861306, 0.037054203, 0.031482287, 0.030747202, 0.030642513, 0.030633207, 0.030631203 and 0.030630301 so that get the average MSE value is 0.035460278. Whereas mark *mean squared error* on *manhattan distance* own mark biggest with many iterations 1 to 8, namely, 0.148646255, 0.087677999, 0.07475772, 0.072924851, 0.072715327, 0.072651345, 0.072620172 and 0.07263329 so that get The average MSE value is 0.08432837. From the

ISSN: 2716-3865 (Print), 2721-1290 (Online)

results comparison average MSE Euclidean *distance* and *Manhattan* distance values smallest owned *euclidean distance* so that fulfil optimal MSE value .

By getting *the centroid* and *mean squared error* (MSE) results, you can determine the results of the number of *cluster data* from *Euclidean distance* and *Manhattan distance* . The number of *clusters* can be seen in Figure 4.13 below.

```
Hasil Cluster Euclidean Distance  Hasil Cluster Manhattan Distance
1    208                          1    214
2    175                          2    175
0    169                          0    163
Name: count, dtype: int64         Name: count, dtype: int64
```

Figure 14. *the Euclidean Distance* and *Manhattan Distance* Clusters

From *the cluster results* in Figure 4.13, the number of *cluster* 0 members with data on toddlers experiencing the impact of *stunting* with "Mild" status from *the Euclidean distance* is 169 data with an average age/month of 38-61, while *the Manhattan distance* is 163 data with an average age /month 38-61. The results for *cluster* 1 with data on toddlers who experienced the impact of *stunting* with the status "Severe" from *the Euclidean distance* were 208 data with an average age/month of 15-44, while *for Manhattan distance* there were 214 data with an average age/month of 15-58. The next results in *cluster* 2 with data on toddlers who experience *stunting* have the status "Very Severe" from *the Euclidean distance* as many as 175 data with an average age/month of 0-24, while in *the Manhattan distance* it has the same value, namely 175 data with an average age /month 0-24.

Conclusion

In the implementation of *K-Means* by calculating *Euclidean distance* and *Manhattan distance* on 552 data. With the results of *the centroid values*, the comparison of *Euclidean distance* and *Manhattan distance* from the test results determines that the optimal distance calculation used is *Euclidean distance with centroid* distance values of 0.848288, 0.779408, 0.941526 at *centroid* 1, then 0.471311, 0.497622, 0.802323 at *centroid* 2 and 0.173396, 0.2 77105 , 0.638767 at *centroid* 3. In the iteration process *euclidean distance* with iteration as many as 8, the same with *manhattan distance* with iteration as many as 8. This show that in matter iteration own balanced results. Analysis results comparison between *Euclidean distance* and *Manhattan distance* in context mark *mean squared error* (MSE). *Euclidean distance* shows mark more low compared to with *Manhattan distance* from every iterations carried out with as much eight iteration from every calculation distance . *Euclidean distance* has the MSE value dominates with mark smallest of 0.061861306 on iteration First until reach mark lowest 0.030630301 in iteration eighth with The average MSE Euclidean distance calculated from these iterations is 0.035460278. On the other hand, *Manhattan distance* shows a higher MSE value compared to *Euclidean distance* , with a maximum value of 0.148646255 in the first iteration and a minimum value of 0.07263329 in the first iteration. eighth. So the average MSE value is 0.08432837 from the first to the eighth iteration.

From the test results using *the Euclidean distance* and *Manhattan distance* with a total of 552 data. At *the Euclidean distance,* 169 toddlers experienced symptoms of mild *stunting* (*cluster* 0) with an average age/month of 38-61 , then 208 toddlers experienced symptoms of severe *stunting* (*cluster* 1) with The average age/month is 15-44 and 175 toddlers experience very severe symptoms of *stunting (cluster* 2) with an average age/month of 0-24 . Meanwhile, in *the Manhattan Distance,* 163 toddlers experienced symptoms of mild *stunting* (*cluster* 0) with an average age/month of 38-61 , then 214 toddlers experienced symptoms of severe *stunting (cluster* 1) with an average age/month of 15-58 and 175 toddlers experienced very severe symptoms *of stunting* (*cluster* 2) with an average age/month of 0-24. The clustering

ISSN: 2716-3865 (Print), 2721-1290 (Online)

results show that the distribution of *cluster members* for these two distance methods has significant similarities, although there are slight differences in the amount of data belonging to each *cluster* . However, this similarity shows consistency in applying *K-Means* using both distance methods.

The analysis results show that using *Sum of Squared Errors* can be used to determine the optimal number of *clusters using the K-Means method* with a case study of measuring *centroid distances* in determining *stunting clusters*. The application of *K-Means* with *Euclidean distance* and *Manhattan distance* was successful in grouping toddlers based on the severity of *stunting symptoms* (mild, severe, very severe). This shows the potential of this method to be used in further analysis of child health problems and evaluation of *stunting reduction*.

References

Apriyani, P., Dikananda, AR, & Ali, I. (2023). Application of the K-Means Algorithm in Clustering Child *Stunting Cases* in Tegalwangi Village. *Hello World Journal of Computer Science* , *2* (1),20–33. https://doi.org/10.56211/helloworld.v2i1.

Fadilah, A., Pangestu, MN, Lumbanbatu, S., & Defiyanti, S. (2022). Grouping Districts/Cities in Indonesia Based on Factors Causing *Stunting* in Toddlers Using the K-Means Algorithm. *JIKO (Journal of Informatics and Computers)*, *6*(2), 223. https://doi.org/10.26798/jiko.v6i2.581

Faujia, RA, Setianingsih, ES, & Pratiwi, H. (2022). K-Means Cluster Analysis and Agglomerative Nesting on Toddler *Stunting Indicators* in Indonesia. *National Seminar on Official Statistics , 2022* (1), 1249–1258. https://doi.org/10.34123/semnasoffstat.v2022i1.1511

Matdoan, MY, Matdoan, UA, & Saleh Far-Far, M. (2022). K-Means Algorithm for Classifying Provinces in Indonesia Based on Stunting Service Packages. PANRITA Journal of Science, Technology, and Arts, 1(2), 41–46. https://journal.dedication.org/pjsta

Nasution, MZ, & Hasibuan, MS (2020). Initial Centroid Search Approach to Improve K-Means Clustering Iteration Efficiency. Techno.Com, 19(4), 341–352. https://doi.org/10.33633/tc.v19i4.3875

Nuha, H. (2023). Mean Squared Error (MSE) and Its Use. Papers.Ssrn.Com, 52, 1–1. https://ssrn.com/abstract=4420880

Ranjawali, R., Talakua, AC, & Abineno, RT (2023). *CLUSTERING STUNTING IN TODDLERS USING THE K-MEANS METHOD AT KANATANG HEALTH CENTER* . 80–92.

Retno, S. (2019). Increasing the Accuracy of the K-Means Algorithm with Clustering Purity as the Initial Cluster Center Point (Centroid). Thesis, July 2019, 1–86. https://repository.usu.ac.id/bitstream/handle/123456789/16782/177038001.pdf?sequence=1&isAllowed=y

Rosaly, R., & Prasetyo, A. (2019). Understanding Flowcharts along with the Most Commonly Used Flowchart Functions and Symbols. https://Www.Nesabamedia.Com,2,2.https://www.nesabamedia.com/pengertian-flowchart/https://www.nesabamedia.com/pengertian-flowchart/

Sari, DJ, Handoko, W., & Parini, P. (2022). Sales Clustering to Determine the Best Selling Building Materials Using the K-Means Method at UD Maju Bersama. *JUTSI (Journal of Information Technology and Systems)*, *2*(2), 93–102. https://doi.org/10.33330/jutsi.v2i2.1690

ISSN: 2716-3865 (Print), 2721-1290 (Online)

Septiarini, A., Thaher, IA, & Puspitasari, N. (2022). Grouping Employee Performance Quality Using the K-Means Clustering Method. Computics : Journal of Computer Systems, 11(2), 131–141. https://doi.org/10.34010/komputika.v11i2.5518

Subayu, A. (2022). Application of the K-Means Method for Analysis of Nutritional *Stunting* in Toddlers: Systematic Review. *SNATI Journal*, *2*, 42–50. https://journal.uii.ac.id/jurnalsnati/article/view/24255/14152

Wanto Anjar, Muhammad Noor Hasan Siregar, APW, Dedy Hartama, NLWSRG, Darmawan Napitupulu, Edi Surya Negara, MRL, & Sarini Vita Dewi, CP (2020). *Data mining: Algorithms and Implementation* . Medan: Kita Write Foundation.

Wibowo Wahyu, Brodjol Sutijo Suprih Ulama, HAA (2020). Learn Python Programming Language. Surabaya: ITS Press.

Widodo, S., Brawijaya, H., & Samudi, S. (2021). Cervical Cancer Clustering Based on Euclidean and Manhattan Comparisons Using the K-Means Method. Budidarma Media Informatics Journal, 5(2), 687. https://doi.org/10.30865/mib.v5i2.2947

ISSN: 2716-3865 (Print), 2721-1290 (Online)