COMPARISON OF K-MEANS CLUSTERING METHOD AND K-MEDOIDS ON TWITTER DATA*

Cahyani Oktarina[1], Khairil Anwar Notodiputro[2‡], Indahwati[3]

[1]Department of Statistics, IPB University, cahyanioktarina10@gmail.com
[2] Department of Statistics, IPB University, khairilnotodiputro@gmail.com
[3] Department of Statistics, IPB University, indah.stk@gmail.com
‡corresponding author

Indonesian Journal of Statistics and Its Applications (eISSN:2599-0802)
Vol 4 No 1 (2020), 189 - 202

Abstract

The presidential election is one of the political events that occur in Indonesia once in five years. Public satisfaction and dissatisfaction with political issues have led to an increase in the number of political opinion tweets. The purpose of this study is to examine the performance of the k-means and k-medoids method in the Twitter data and to tweet about the presidential election in 2019. The data used in this study are primary data taken from Muhyi's research, then mining the text against data obtained. Because this data has been processed by Muhyi to analyze the electability of the 2019 presidential candidate pairs, for this journal needs a preprocessing was carried out to analyze the tendency of tweets to side with the candidate pairs of one or two. The difference in the pre-processing of this research with previous research is that there is a cleaning of duplicate data and normalizing. The results of this study indicate that the optimal number of clusters resulting from the k-means method and the k-medoid method are different.

Keywords: *text mining, clustering, k-means, k-medoids, twitter*.

* Received Dec 2019; Accepted Feb 2020; Published online on Feb 2020

1. Introduction

As technology and information become increasingly sophisticated, many agencies or organizations produce and store large amounts of data in their data base. The most popular method used to extract data base or big data is called data mining. According to Simhachalam & Ganesan (2016), data mining is defined as an analysis process to find valid and unexpected relationships between data sets and convert data into data structures so that they are easy to understand and useful for users. To find out the relationship of the data base, data analysis techniques are needed. Data mining analysis techniques generally consist of prediction techniques, description techniques and inference techniques.

Grouping is one of the description techniques of data mining analysis. In general there are two methods of grouping, the method of hierarchy and the method of non-hierarchy. One of the popular non-hierarchical clustering methods used is the k-means method. K-means is also known as hard clustering which can group objects with clear boundaries, meaning that they can group objects into certain groups and not members of other groups (Sivarathri & Govardhan, 2014). The k-means method is a partition-based method that attempts to partition data into two or more groups using the mean value as the center of the cluster. In addition to the k-means method there is also the k-medoids method which is a partition-based method that uses medoids as the center of the cluster. Medoids is the most centralized cluster data object (Arora et al., 2016), so this method is more robust to outliers than the k-means method (Tiwari & Singh, 2012).

Previously there were studies comparing the performance of the k-means algorithm with k-medoids, one of which was Arora et al. (2016). Both algorithms are implemented using a dataset of 10000 transactions obtained from KEEL (Knowledge Extraction Evolutionary Learning). The results show that the cluster produced by the k-means algorithm shows an overlap while the k-medoids algorithm is less overlapping than the k-means algorithm. Problem statistics methods when the data is large. This is the issue that will be discussed in this paper using the example of Twitter data.

Large data will cause the noise is also large. If the noise is large, it will affect the results of the grouping. This will cause the results of grouping to be not optimal. One way that is done to overcome large noise is to do pre-processing.

In this study the determination of similarity between objects using euclidean distances, where the concept of distance requires freedom between changes (Mattjik & Sumertajaya, 2011). Regarding the comparison of the performance of the k-means and k-medoids algorithms, a new study of the two methods is needed. The problem that arises from this research is the data used does not pay attention to the correlation between the variables used. The grouping is done with the condition of the correlation between changes, so the characteristics of the groups formed are not optimal. One approach that can be taken to overcome correlated variables is to use principal component analysis, which can then be analyzed using cluster analysis (Mattjik & Sumertajaya, 2011). In addition to overcoming correlation between changes, principal component analysis can also handle plots in a multidimensional space. Previous research also suggests examining correlations. Thus researchers interested in studying the problem to complement the results of previous studies.

The data used in this study is text data that is a tweet about the 2019 presidential election. Hanna et al. (2013) explains that the increase in the number of political opinion tweets on Twitter is caused by a political event that occurs, for example the General Election. The data that has been collected is then processed using text mining and then the tweets are grouped. The number of variables that are formed depends on the tweet used.

This study aims to examine the performance of the k-means and k-medoids methods in the Twitter data and to group tweets about the 2019 presidential election.

2. Methodology

2.1 Twitter Social Media

Social media is an internet-based application built with Web 2.0 technology and allows the exchange of user-generated content (Kaplan & Haenlein, 2010). One of the most popular social media right now is Twitter. Twitter is used to exchange ideas, ideas, gather information, and see the activities of users that are followed (Java et al., 2007). Ideas sent via Twitter are called tweets. Tweets are stored in the Application Programming Interface (API) feature that can be accessed by users. The desired information can be found based on the keywords entered so that the tweets obtained are in accordance with the topics discussed. Suppose the keyword used is "pilpres 2019 ", then tweets that have the word "pilpres 2019 " will be picked up by the system. Withdrawing Twitter data needs to get permission from Twitter to get the API access code. To get the access code you need to register the application using a Twitter account which can be done at the following link, https: ///dev.twitter.com/apps. There are four access codes, namely consumer key, consumer secret, access token, and access secret. Data withdrawal can be done if you already get the access code by integrating Twitter API and R Studio. During data withdrawal the internet connection must always be activated.

2.2 Preprocessing

Preprocessing is the most influential stage on the quality of data generated from text mining. This stage is done to transform unstructured data into structured data, so that further analysis can be done. All data obtained is done by pre-processing, namely cleaning by deleting text containing punctuation, username, @, hashtag, url, http, links, numbers, and changing tweets into lowercase letters called case folding. Furthermore, making tweets into words is called tokenizing. In addition, stopward removal is carried out and continued with stemming. Stopward is a word that has zero value information or unimportant words such as "dan", "di", "pada", etc. Whereas stemming is the process of changing the form of words into basic words. Stemming works by searching for the basic words of each word and eliminating affixes, so that mining of the text is optimal (Munková et al., 2013). As well as normalizing data that is changing nonstandard words or abnormal words into standard words in accordance with the Big Indonesian Dictionary (KBBI).

2.3 Term Frequency Inverse Document Frequency (TF-IDF)

After doing the pre-processing then convert text data to numeric by making a weighting matrix. The weighting terms used are term frequency (TF) and inverse document frequency (IDF). TF weight is the appearance of the term in a tweet, if it appears once

in a tweet then the value is 1. If it does not appear at all then the value is 0, while IDF weight is the logarithm of dividing the number of tweets by the frequency of tweets containing words. The TF-IDF value is a multiplication of two components, namely the TF and IDF values.

2.4 Cluster Analysis

Cluster analysis is one of the multiple variable techniques whose main purpose is to group objects based on their similarity in characteristics. The characteristics of objects in a group have a high degree of similarity, while the characteristics of objects in a group with other groups have a low level of similarity. In other words, diversity within a group is minimum while diversity among groups is maximum (Mattjik & Sumertajaya, 2011). The similarity or dissimilarity between objects can be measured using distance measurements. In cluster analysis there are several measures of distance that are often used to measure the degree of similarity of objects including euclidean distance, minkowski, and manhattan/city block (Johnson & Wichern, 2007). In this study the similarity between objects is measured using the euclidean distance. Mathematically can be written with the following equation:

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} \tag{1}$$

Where d_{ij} is The distance between the i-th object and the j-th object, x_{ik} is the value of the i-th object in the k-th variable, x_{jk} is the value of the j-object in the k-th variable, p is number of observed variables.

2.5 Principal Component Analysis

One approach that can be taken to overcome correlated variables is to use principal component analysis, which can then be analyzed using cluster analysis (Mattjik & Sumertajaya, 2011). Principal component analysis is one of the multiple variable analysis used to form a new variable which is a combination of the initial variables linearly. In other words, the analysis of the main components reduces the origin variable p to a new dimension q where $q < p$ (Johnson & Wichern, 2007). These new variables are called the main component which is a linear combination of the original variables. The information contained in the main component is a combination of all variables with a certain weight. The main components are uncorrelated and the information does not overlap (Mattjik & Sumertajaya, 2011). The i-th main component model of p variables with a pairs of eigenvalues and eigenvectors (λ_1, a_1), (λ_2, a_2), … , (λ_p, a_p) can be written as follows:

$$Y_i = a_{1i}x_1 + a_{2i}x_2 + \cdots + a_{pi}x_P = \boldsymbol{a}'\boldsymbol{x} \tag{2}$$

The diversity of the j-th major component is

$$var(Y_i) = \lambda_i \; ; j = 1,2, \dots, p \tag{3}$$

2.6 K-means method

In 1967 Mac Queen introduced k-means which is a partition-based clustering analysis method. K-means is a method that attempts to partition data into two or more groups using the mean value as the center of the cluster. For example $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\}$ is data to be analyzed and $\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \dots, \boldsymbol{v_c}\}$ is the center of the data $\boldsymbol{X}$ group in the dimension $(\mathbb{R}^p)$. Where n is the number of objects, p is the number of variables and c is the number of partitions or groups (Cebeci & Yildiz, 2015). The center of the cluster can be calculated using the following formula:

$$v_{ij} = \frac{1}{n_i}\sum_{k=1}^{ni} x_{kj} \tag{4}$$

Where v_{ij} is center of the i-th group on variable j-th, n_i is the number of objects that belong to the i-th group, x_{kj} is observation value of the k-th object to the j-th variable. Data grouping can be written as follows:

$$\mu_{ik} = \begin{cases} 1, d = min\{d_{ik}^2(x_k, v_i)\} \\ 0, d\ yang\ lainnya \end{cases} \tag{5}$$

Where μ_{ik} is value of the k-th object membership into the i-th cluster, v_i is average value of the i-th group center, x_k is k-th object observation value. The purpose of this grouping is to minimize diversity within a group and maximize diversity between groups. In other words this grouping aims to minimize objective functions (Cebeci & Yildiz, 2015). Mathematically the objective function can be written with the following equation:

$$J(\boldsymbol{X}, \boldsymbol{V}) = \sum_{\boldsymbol{k=1}}^{ni}\sum_{\boldsymbol{i=1}}^{\boldsymbol{c}} \mu_{ik} d_{ik}^2 \tag{6}$$

Where ni is the number of objects that belong to the i-th group, c is the number of cluster, d_{ik}^2 is euclidean distance between the object to the center of the i-th cluster.

2.7 K-medoids method

K-medoids is a partition-based, non-hierarchical clustering method that uses medoids as the center of the cluster. The algorithm that is often used in k-medoids is partitioning around medoids (PAM). This algorithm aims to minimize the distance of the object to the object medoids. For example $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\}$ is data to be analyzed and $\boldsymbol{P} = \{\boldsymbol{o_1}, \boldsymbol{o_2}, \dots, \boldsymbol{o_c}\}$ is the object as its medoids. Determination of replacing or not medoids objects with non-medoids objects depends on the cost function calculated during the iteration (Han et al., 2011). For example $\boldsymbol{X_{nxp}}$ is data consisting of n objects and p variables. The distance between the i-th object and the j-th object is denoted by $d(i, j)$. The allocation of each j-th object to one of the initial medoids can be written as follows:

$$z_{ij} = \begin{cases} 1, d = min\{d_{ij}^2(x_j, O_i)\} \\ 0, d\ yang\ lainnya \end{cases} \tag{7}$$

Where z_{ij} is membership value of the j-th object into the i-th medoids, O_i is i-medoids object x_j is j-th object observation value. The optimization model in k-medoids was first discovered by Vinol (1969) in Kaufman & Rousseeuw (2009) which can be written as follows:

$$minimize \sum_{i=1}^{n} \sum_{j=1}^{n} d(i,j) z_{ij} \tag{8}$$

A group will be formed by allocating each object to the closest initial medoids. The distance between the jth object to the initial medoids is defined as follows:

$$\sum_{i=1}^{n} d(i,j) z_{ij} \tag{9}$$

If all objects have been allocated to the nearest medoids, then calculate the total distance using the following formula:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d(i,j) z_{ij} \tag{10}$$

2.8 Data

The data used in this study are real data in the form of primary data taken from Muhyi (2019). The time of data retrieval and the number of tweets obtained in this case are presented in Table 1. The keywords used are @KHMarufAmin, @jokowi, Jokowi, midget, @prabowo, @sandiuno, Prabowo, and Kampret. Total data obtained were 27083 tweets. Tweets are from Twitter users in all provinces in Indonesia.

Table 1: Time of data retrieval and number of tweets.

No	Date	Number of Tweets
1	26 February 2019	2900
2	8 March 2019	15601
3	17 March 2019	8582

2.9 Data Analysis Procedure

Data analysis uses software R 3.4.3. The steps of analysis are as follows :

1. Pre-processing. This stage aims to transform unstructured data into structured data.
2. Weighting with Frequency Inverse Document Frequency (TF-IDF). TF-IDF produces a matrix that is used as a variable in clustering applications.
3. Manually mining sentiments. Sentiment is carried out after the pre-processing cleaning stage. Grouping in this study was grouped based on the sentiments obtained.

4. Conduct analysis of the main components to overcome the correlated variables.
5. Grouping tweets with the k-means and k-medoids methods.
   Steps for grouping the k-means method:
   a) Determine the starting point of each cluster randomly from the specified variable.
   b) Calculate the distance of each object with each center point of the cluster using Euclidean distance.
   c) Grouping each object into a group based on the minimum distance.
   d) Repeat step (a) and the process stops if no changes to the membership of the group are formed.

   Steps for grouping the k-medoids method:
   a) Determine the initial medoids of each group that are randomly determined from the specified variable.
   b) Assign each object to each group with the closest medoids object.
   c) Randomly retrieve an object that is not medoids, O_{random}
   d) Calculate the cost value, S from the exchange value of the O_j medoids object with O_{random}
   e) If $S < 0$ then exchange O_j with $O_{random}$for new data values from k medoids. S is the difference between the total new distance and the previous total distance.
   f) The process stops if there is no change in membership of the cluster formed.
6. Interpretation of cluster results.

3. Results

3.1 Preprocessing

Preprocessing is the most influential stage on the quality of data generated from text mining. This stage is done to identify unstructured data into structured data, so that further analysis can be done. The data used in this case is text, that is a tweet written by Twitter users. Before the data is analyzed, pre-processing of the tweets is obtained. Because this data has been preprocessed by Muhyi (2019) to analyze the electability of the two pairs of presidential candidates in 2019, for this thesis needs to be reprocessed to analyze the grouping of tweets according to the tendency of tweets to side with candidate pairs 1 or 2.

The difference in the pre-processing of this research with previous research is that there is cleaning of duplicate data and normalizing. From the total data obtained, as many as 27083 tweets are then cleaned by removing duplicate tweets. There are 21647 duplicate tweets. Tweets are handled by being deleted from the database so as not to interfere at the pre-processing stage. Furthermore, pre-processing of 5436 tweets was performed.

Pre-processing in this study aims to transform unstructured data into structured data, so that further analysis can be done. Examples of pre-processing results are presented in Table 2.

The pre-processing stages carried out in this study are as follows:

1. Cleaning. At this stage the text containing elements http, link, url, hashtag, mention, punctuation, and not the alphabet is replaced with spaces.
2. Case folding. This phase is carried out aiming to facilitate text comparisons

in data processing. At this stage all text containing capital letters is converted to lowercase letters.

Table 2: Example of pre-processing results.

Process	The Original Word	Transformation Results
Remove punctuation	ma?ruf	maruf
Remove mention	@prabowo	prabowo
Remove hastag	#KitaHarusMenang	(lost)
Clean number	presiden 2019	presiden
Remove URL	https://t.co/8WBqjBuYyN"	(lost)
Case folding	INDONESIA MAJU	indonesia maju
Tokenizing	kabar baik	“kabar” “baik”
Stopwords	Jika kita terus mendukung	mendukung
Stemming	kedamaian	damai
Normalisasi	jokodok	jokowi

3. Tokenizing. At this stage the decomposition process is carried out which aims to separate the tweets into separate words which are also called tokens.
4. Stopwords. At this stage the word deletion that has information of zero value or words that do not have the tendency to be negative or positive is carried out.
5. Stemming. At this stage the word is changed into the basic word form.
6. Normalization. At this stage the word abnormal or nonstandard is changed to the standard word.

Table 3: Frequency of occurrence of words that appear more than 200 times.

Word	Frequency
prabowo	2059
jokowi	1644
sandiuno	1419
gerindra	569
tidak	474
indonesia	346
pdip	316
menang	303
dukung	206
presiden	201

At this pre-processing stage there is a reduction in the number of words. The total number of words after cleaning is 10666 words. There are 7284 words deleted or used as stopwords, then stemming and normalizing. The stemming and normalization phase can reduce the number of words by 73 percent. So there are 1931 words left to be analyzed. The frequency of occurrence of words that appeared more than 200

times is presented in Table 3. The highest frequency of words is the word prabowo which appears 2059 times and the second highest is the word jokowi as much as 1644 times.

3.2 TF-IDF Weighting

After pre-processing the next step is giving TF-IDF weight to each word and changing it in matrix form. The matrix is used as a variable for the application of cluster analysis. The TF-IDF value is a multiplication of two components, namely the TF and IDF values. For example three tweets that have been pre-processed.
Tweet 1: jokowi dukung coblos
Tweet 2: prabowo maju
Tweet 3: jokowi optimis

TF weight is the appearance of the term in a tweet, if it appears once in a tweet then the value is 1. If it does not appear at all then the value is 0. This weighting is presented in Table 4. The first tweet is given a value of 1 if there is the word jokowi, support, and punch. While other words are given a value of 0 because it is not contained in the first tweet.

Table 4: Weighting of TF.

Tweet	Word					
	jokowi	dukung	coblos	prabowo	maju	optimis
1	1	1	1	0	0	0
2	0	0	0	1	1	0
3	1	0	0	0	0	1

IDF weight is the logarithm of dividing the number of tweets by the frequency of tweets containing words. If the word appears in a lot of tweets, the IDF value is getting smaller, and vice versa. This weighting is presented in Table 5. The IDF weighting values for each word are as follows:

$$IDF_i = log\left(\frac{N}{df_i}\right) = log\left(\frac{3}{1}\right) = 0.48$$

$$IDF_i = log\left(\frac{N}{df_i}\right) = log\left(\frac{3}{2}\right) = 0.18$$

So, the word jokowi has an IDF value of 0.18 because it appears in two tweets, while the other word has an IDF value of 0.48 because it appears in one tweet.

Table 5: Weighting of IDF.

Tweet	Word					
	jokowi	dukung	coblos	prabowo	maju	optimis
1	0.18	0.48	0.48	0	0	0
2	0	0	0	0.48	0.48	0
3	0.18	0	0	0	0	0.48

After the TF and IDF weights are known, the next step is to calculate the TF-

IDF weights. The weight of TFIDF is the value of the frequency of the i-th word against j-tweets. The TF-IDF weights are presented in Table 6 which is the multiplication between the TF and IDF weights. The TF-IDF weight values for the word jokowi are as follows:

$TFIDF = TFxIDF = 1x\ 0.18 = 0.18$

TFIDF weight values for other words are as follows:

$TFIDF = TFxIDF = 1x\ 0.48 = 0.48$

Table 6: Weighting of TF-IDF.

Tweet	Word					
	jokowi	dukung	coblos	prabowo	maju	optimis
1	0.18	0.48	0.48	0	0	0
2	0	0	0	0.48	0.48	0
3	0.18	0	0	0	0	0.48

3.3 Grouping Tweets

The results of simulation studies that have been done previously show that it cannot be distinguished between the k-means method and the k-medoids method from the simulation data generated. Besides the correlation between the variables used in this study an average of 0.004.

The purpose of the grouping in this study is to group the tweets by exploring the similarity of words based on the weighted values obtained. The grouping is based on sentiment labeling done manually which is objective with the following labeling criteria, the first label shows tweets in favor of Jokowi, the second label shows tweets in favor of Prabowo and the third label tweets that show other opinions.

Table 7: Percentage of sentiment labeling results.

Label	Number of tweets
1	54.27
2	23.80
3	21.93

Table 7 presents the percentage of sentiment labeling results. The highest percentage were tweets that favored Jokowi, which was 54.27 percent, while tweets that sided with Prabowo were 23.80 percent, and tweets with other opinions amounted to 21.93 percent. Before clustering, the optimum number of groups for each method must be known.

Evaluation of cluster results can be seen from the comparison between the distance within the cluster and the distance between the buttons (S_w/S_b). A small S_w/S_b value indicates the optimum number of clusters. The S_w/S_b value of the k-means method decreases until the cluster is 5, when the cluster is 6 the S_w/S_b values increase and decreases again until the cluster is 15. The evaluation results can be seen from the graph of the S_w/S_b value as illustrated in Figure 1 When the number of hordes equals 6 the graph drops steadily. This can also be seen from the percentage decrease in the

value of S_w/S_b as illustrated in Figure 2. When the number of clusters equals 6 graphs the percentage decrease in the value of S_w/S_b k-means method starts to stabilize so that it can be concluded that the optimum number of clusters produced by the k-method means is 6 groups.

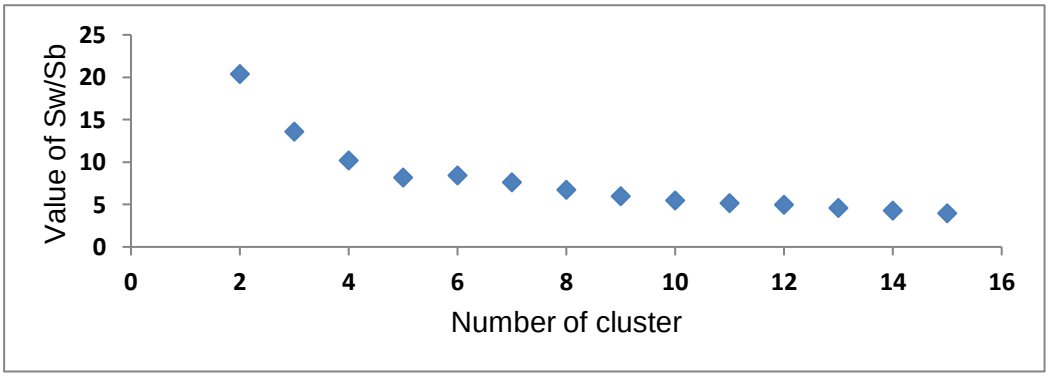

Figure 1: Graph of The S_w/S_b Value of K-Means Method.

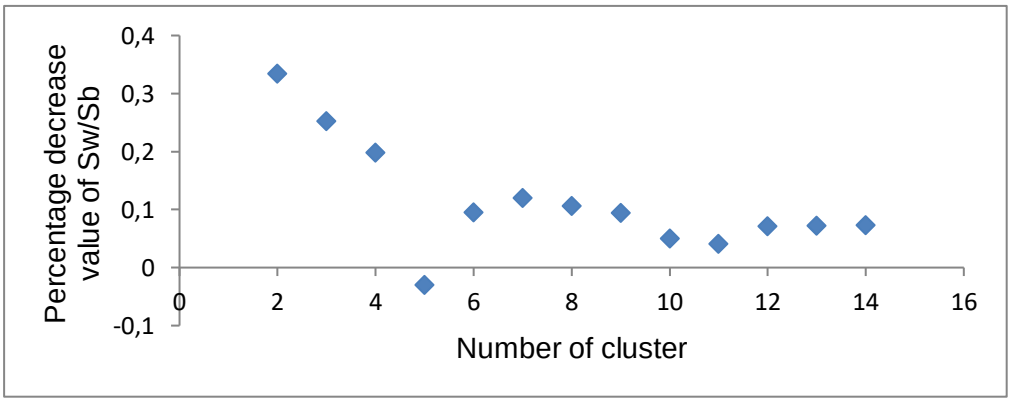

Figure 2: Graph of Percentage Decrease in The Value of S_w/S_b K-Means Method.

The S_w/S_b value of the k-medoid method does not converge until the group is 15. Decreasing the value of S_w/S_b starts stable compilation of 7 switch groups. This can be seen from the graph S_w/S_b values that can be seen in Figure 3. When the number of groups equals 7 the graph decreases to begin to stabilize. This can also be seen in Figure 4 namely the graph of the percentage decrease in the value of S_w/S_b. By counting the number of groups that are equal to 7, counting the number of decreases, S_w/S_b, the k-medoids method starts to stabilize so that it can calculate the optimal number of groups produced by the k-medoids method as many as 7 groups.

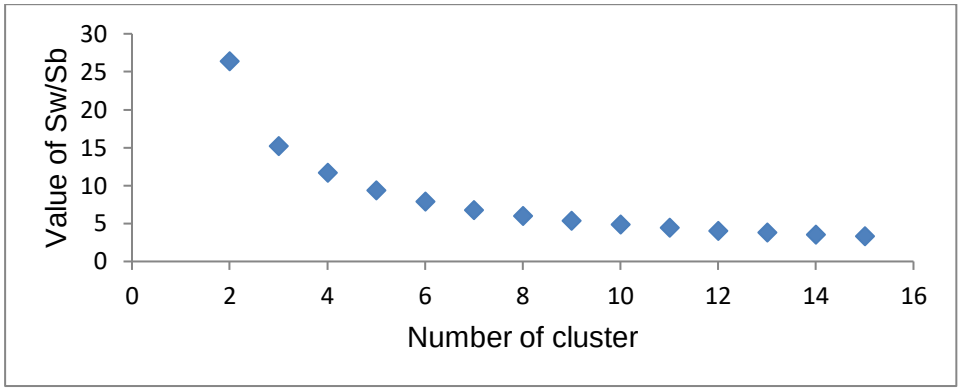

Figure 3: Graph of The S_w/S_b Value of K-Medoids Method.

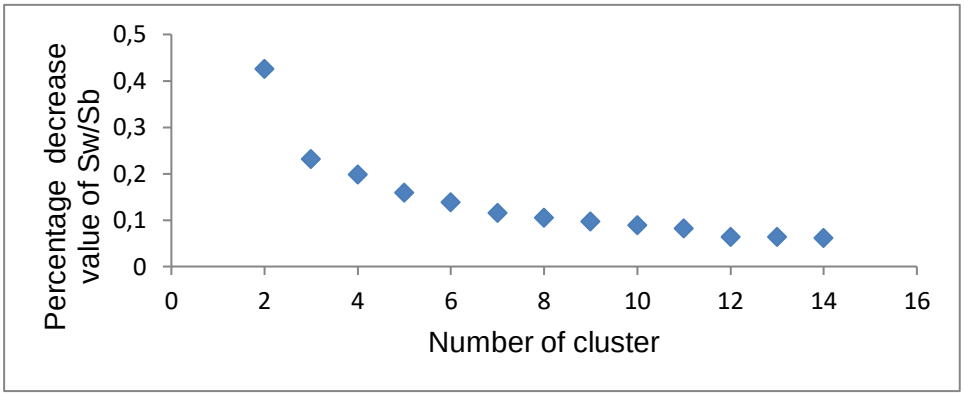

Figure 4: Graph of Percentage Decrease in The Value of S_w/S_b K-Medoids Method.

In this case the optimum number of clusters resulting from the k-means method and the k-medoids method differ. The optimum number of clusters produced by the k-means method is 6 clusters, while the k-medoids method is 7 clusters.

Table 8: Percentage results of grouping of tweets using the k-means method with a group of six.

Cluster	Number of tweets	Label			Results
		1	2	3	
1	2300	54.01	23.07	22.92	1
2	1017	39.46	39.50	21.04	1 or 2
3	528	64.44	33.33	2.22	1
4	213	53.05	24.41	22.54	1
5	480	55.42	19.58	25.00	1
6	898	14.48	19.31	66.21	3

Table 8 presents the percentage of tweeting results based on sentiment labeling using the k-means method and Table 9 presents the percentage of tweeting results using the k-medoids method. From the results obtained it appears that there is a group that sided with the two labels. This can be seen from the absence of a majority percentage. Whereas groups that produce a percentage value of more than 50% on labeling indicate that the group is in favor of the label. As mentioned previously, the

first label shows tweets that are in favor of Jokowi, the second label shows tweets that are in favor of Prabowo and the third label is tweets that show other opinions. The first label shows tweets in favor of Jokowi, the second label shows tweets in favor of Prabowo and the third label shows tweets that show other opinions.

Table 9: Percentage results of grouping of tweets using the k-medoids method with the number of groups of seven.

Cluster	Number of tweets	Label			Results
		1	2	3	
1	4281	54.87	23.11	22.02	1
2	216	53.24	24.07	22.69	1
3	487	22.18	27.10	50.72	3
4	213	53.05	24.41	22.54	1
5	17	41.18	58.82	0.00	2
6	13	7.69	61.54	30.77	2
7	209	27.27	54.07	18.66	2

4. Conclusion and Suggestion

4.1 Conclusion

Both of these methods can be used to group text data into tweets based on predetermined parameters. The optimum number of clusters produced from the k-means method and the k-medoids method is not the same. The optimum number of clusters produced by the k-means method is 6 clusters, while the k-medoids method is 7 clusters.

4.2 Suggestion

The clustering in this study uses the eucliden distance. The next researcher can use another distance measure. Labeling sentiments manually is expected to be done by people who are experts in their fields and separating tweets containing the word Jokowi only and tweets containing the word Prabowo alone.

References

Arora, P., Deepali, & Varshney, S. (2016). Analysis of k-means and k-medoids algorithm for big data. *Procedia Computer Science*, *78*: 507–512. https://doi.org/10.1016/j.procs.2016.02.095

Cebeci, Z., & Yildiz, F. (2015). Comparison of k-means and fuzzy c-means algorithms on different cluster structures. *Agrárinformatika/Journal of Agricultural Informatics*, *6*(3): 13–23. https://doi.org/10.17700/jai.2015.6.3.196

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hanna, A., Wells, C., Maurer, P., Friedland, L., Shah, D., & Matthes, J. (2013). Partisan alignments and political polarization online: A computational approach to understanding the French and US presidential elections. *Proceedings of the 2nd Workshop on Politics, Elections and Data*, 15–22.

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, 56–65.

Johnson, R. A., & Wichern, D. W. (2007). *Applied multivariate statistical analysis* (Vol. 6). New Jersey (US): Pearson Education.

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, *53*(1): 59–68. https://doi.org/10.1016/j.bushor.2009.09.003

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Mattjik, A., & Sumertajaya, I. (2011). *Sidik Peubah Ganda: Menggunakan SAS*. Bogor (ID): IPB Press.

Muhyi, F. (2019). *Penggunaan Twitter sebagai Penyedia Peubah Penyerta dalam Pendugaan Area Kecil [tesis]*. Bogor (ID): IPB University.

Munková, D., Munk, M., & Vozár, M. (2013). Data pre-processing evaluation for text mining: transaction/sequence model. *Procedia Computer Science*, *18*: 1198–1207. https://doi.org/10.1016/j.procs.2013.05.286

Simhachalam, B., & Ganesan, G. (2016). Performance comparison of fuzzy and non-fuzzy classification methods. *Egyptian Informatics Journal*, *17*(2): 183–188.

Sivarathri, S., & Govardhan, A. (2014). Experiments on Hypothesis “Fuzzy K-Means is better than K-Means for Clustering.” *International Journal of Data Mining & Knowledge Management Process*, *4*(5): 21–34. https://doi.org/10.5121/ijdkp.2014.4502

Tiwari, M., & Singh, R. (2012). Comparative investigation of k-means and k-medoid algorithm on iris data. *International Journal of Engineering Research and Development*, *4*(8): 69–72.