J. Indones. Math. Soc.
Vol. 32, No. 03 (2026), pp. 1–21.

A More Precise Elbow Method For Optimum K-means Clustering

Indra Herdiana[1*], Mokhamad Alfin Kamal[1], Triyani[1], Mutia Nur Estri[1], and Renny[1]

[1]Mathematics Study Program, Jenderal Soedirman University, Indonesia.

Abstract. K-means clustering is an unsupervised clustering method that requires an initial decision of number of clusters. One method to determine the number of clusters is the elbow method, a heuristic method that relies on a graphical plot of the within-cluster sum of squares against the number of clusters. The method uses the number based on the elbow point, the point closest to 90° that indicates the most optimum number of clusters. This research improves the elbow method such that the selection of cluster numbers is unbiased towards visual interpretation. We use the analytical geometric formula to calculate an angle between lines and real analysis principle of derivative to simplify the elbow point determination. We also consider every possibility of the elbow method graph behaviour such that the algorithm is universally applicable. The result is that the elbow point can be measured precisely with a simple algorithm that does not involve complex functions or calculations. This improved method gives an alternative of more reliable cluster determination method that contributes to more optimum k-means clustering, obtained by the fewest clusters with the lowest error.

Key words and Phrases: k-means clustering, elbow method, angle between lines.

1. INTRODUCTION

Data are valuable assets in this digital era, inseparable from peoples' lives. Every decision making process involves data. However, data alone cannot produce valuable information without proper management and interpretation. According to [1], profits are made by data management, not data possession, because data do not have an intrinsic value. Statistics is a branch of mathematics that deals with collection, management, analysis, interpretation, and visualisation of data. It is

*Corresponding author : indra.herdiana@unsoed.ac.id
2020 Mathematics Subject Classification:
Received: 26-11-2024, accepted: 07-03-2026.

the main key to data management, providing mathematical and scientific method to managing data. For example, [2] and [3] develop a data management system for demography and primary health facilities respectively that positively impact the society. These two are just a tiny fraction of the essential roles of statistics for human lives.

One of widely used data management method is clustering. Clustering is a method of grouping data to several clusters such that data points in the same cluster have a maximum similarity, while data points in different clusters have a maximum difference. This method is used in many sectors, such as manufacture, transport, sustainable energy, health, and public policy, since it makes data, that are not necessarily meaningful, become meaningful (see [4, 5, 6]). Despite the importance, no clustering methods that fit all [7]. Interestingly, different clustering methods applied to the same data set may result differently. One of notable clustering methods is k-means clustering.

K-means clustering is a clustering method which principle is minimising the distance between every object within a cluster and the centre of the cluster called a centroid. The used distance is the Euclidean distance since the data span Euclidean distance, and the centres of clusters are means on Euclidean spaces [8] [1]. In other words, objects are grouped into one cluster with the nearest centroid, indicating the maximum similarity centred in the centroid. Some examples of k-means use in data management for further application can be seen in [11, 12]. One factor that makes this method widely used is its simplicity [13]. K-means clustering uses simple calculations, such as the Euclidean distance, that are easily calculable by both manually and computationally. K-means clusering also converges fast [13]. However, although it converges faster than k-medoids clustering, k-means clustering needs an initial determination of number of clusters. This number must not be determined negligently.

There are several methods to determine such a number of clusters, such as the elbow method, gap statistics, the silhouette method, and canopy [14]. These methods have unique advantages, disadvantages, and compatibility. However, despite its simplicity, the elbow method has received criticism for its accuracy. The elbow method determines the number of clusters based on a graph that connects some number of clusters and corresponding sum of squared error (SSE) of each cluster. The optimum number of cluster is determined by the point forming an "elbow", the pointiest one, indicating that higher numbers of clusters do not significantly reduce the SSE. The criticism lies on the fact that this method relies on subjective visual interpretation where the elbow point is chosen according to the probably biased plot. The elbow point may be chosen inaccurately, thus affecting the clustering result. Here are some examples of the elbow method implementations.

[1] Some research use different distance for k-means clustering, such as Mahalanobis distance and contour similarity with some advantages [9] [10]. However, this paper mainly focuses on improving the elbow method and is not majorly influenced by distance choice

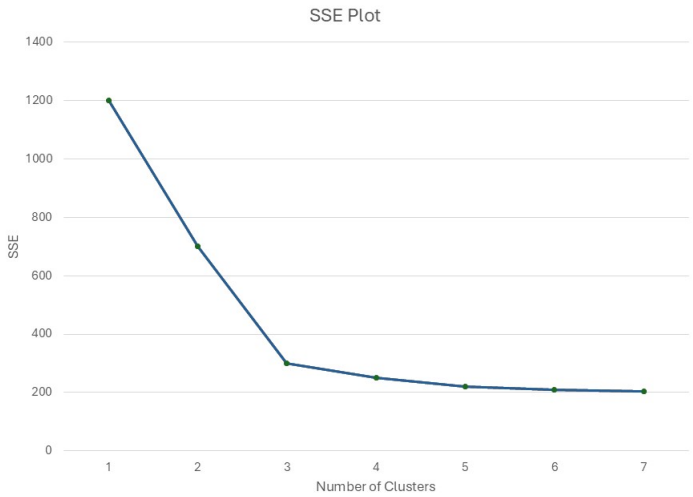

FIGURE 1. SSE plot with a clear elbow

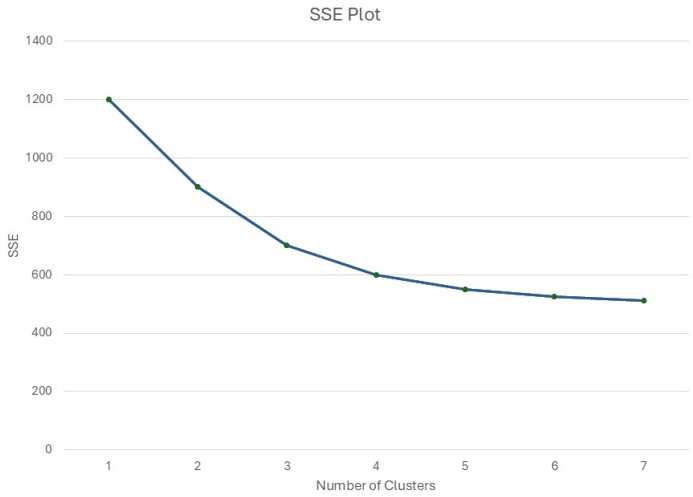

FIGURE 2. SSE plot with a unclear elbow

Figure 1 illustrates a clear elbow point; we can immediately choose $k = 3$ as the optimum number of clusters since the pointiest point is apparent. However,

there may be a case depicted by Figure 2 where the graph is almost smooth such that the pointiest point is not clearly shown. The elbow point is not clearly shown.

The issue extends beyond this point. Although the elbow points appears to be obvious as shown by 1, the "obvious elbow" is possible to be "a false elbow". It is because the scaling of the horizontal and vertical axis are often disproportional. In the case of Figure 1, the horizontal axis scale is 0 to 7, while the vertical axis scale is 0 to 1400. This distortion causes misinterpretations and lead to incorrect choices of optimum number of clusters, thus giving unwanted k-means clustering results.

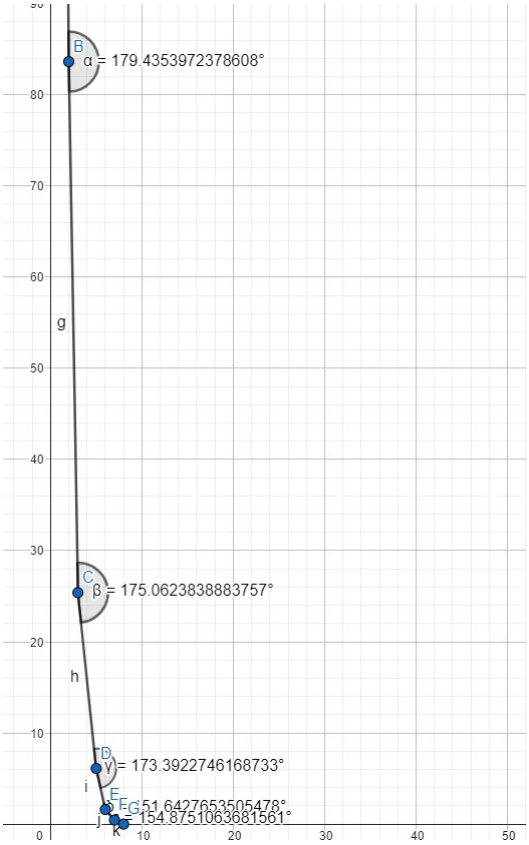

FIGURE 3. SSE plot with undistorted scale

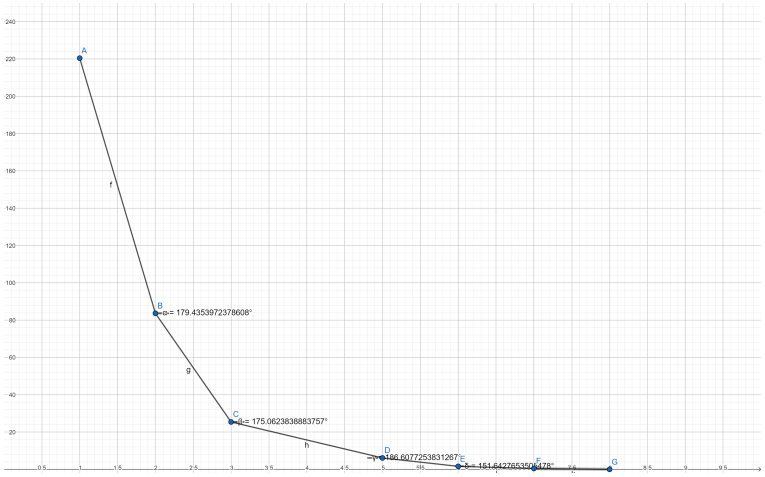

FIGURE 4. SSE plot with distorted scale

Figure 4 shows a distorted plot (that unfortunately often occurs during elbow method processes) where the vertical axis scale is not 1:1 proportional with the horizontal axis scale. In contrast, Figure 3 shows the plot with a proportional 1:1 scaling where it indicates absolute SSE decreases. Figure 4 gives impression that the optimum number of cluster is 3, while Figure 3 contradicts it.

Due to the complex subjectivity, k-means clustering often involves another method that is considered less subjective, thus neglecting improvisation of the elbow method. Meanwhile, analytical geometry and real analysis have hidden roles in precising this heuristic method. This paper aims to improve the method by changing the subjectivity into objectivity using some principles of analytical geometry and real analysis.

Shi etc. in [15] provides a method of determining the elbow point precisely using a principle of geometry, namely the cosine rule of a triangle. However, this method involves a long calculation of a triangle side length represented by the distance between two adjacent points. Moreover, this methods uses an inverse trigonometry function that is not a simple standard mathematical operation like addition and multiplication. Sinaga and Yang in [16] create an alternative k-means clustering algorithm where it finds the number of clusters independently without using initial cluster method determination. The method's disadvantage is similar to that of [15]. It involves a non-standard arithmetic operation namely natural logarithm.

The method proposed in this paper offers an advantage over the methods in [15, 16] that may give similar accuracy but computationally more expensive. It only involves standard arithmetic operations namely addition, subtraction, multiplication, and division, resulting in more efficient algorithm. Not all programming languages have built-in complex mathematical functions like trigonometric inverses

and natural logarithm; even C++ needs to add the package cmath in order to use these operations [17]. Therefore, using standard arithmetic functions that are included in all programming languages offer flexibility and simplicity.

The method uses the principle of measuring an angle between two lines formulated in analytical geometry. The real analysis principle of derivative is also used to optimise the algorithm such that non-standard operations occurring in the algorithm can be omitted. Furthermore, the proposed method also considers every possibility of the elbow method graph behaviour such that the elbow point is chosen more precisely. This method can be an alternative to optimise k-means clustering; results in [15] are even shown to be better than the silhouette method. In the research, the silhouette method exhibits inconsistency to different data distribution by giving different cluster numbers, unlike the elbow method, and is computationally more expensive.

2. MAIN RESULTS

This section discusses the detailed formulation of determining the elbow point using an analytical geometric approach. Data simulation is also provided using Python.

2.1. **Exact Elbow Point Formulation.**

The following theories about the elbow method, k-means clustering, and SSE are cited from [8, 14].

Let $\mathcal{X}$ be a collection of n continuous p-dimensional data. Let $\mathbf{X}_i$ denote the i-th data point on $\mathcal{X}$ where

$$\mathbf{X}_i = (x_1, x_2, \ldots, x_p)$$

and $i = 1, 2, \ldots, n$. Assume that $\mathcal{X}$ does not contain an outlier such that it is suitable for k-means clustering. Suppose that we cluster the data set into k clusters, and the determination of k utilises the elbow method.

We initiate the elbow method by plotting the sequence $(SSE(k))$ for $k = 1, 2, \ldots, n^2$ where $SSE(k)$ is the sum of squared errors obtained if the data set is grouped into k clusters by k-means clustering. It is defined by

$$SSE(k) = \sum_{j=1}^{k} \sum_{\mathbf{X}_i \in S_j} \|\mathbf{X}_i - \mathbf{C}_{S_j}\|^2$$

where S_j is the j-th cluster, $\mathbf{C}_{S_j}$ is the centroid of cluster S_j, and $\|\mathbf{X}_i - \mathbf{C}_{S_j}\|$ is the Euclidean distance between the data point $\mathbf{X}_i$ and its centroid $\mathbf{C}_{S_j}$. The SSE plotted here is obtained after iterating the k-means clustering such that no data points change their cluster membership. In other words, the plotted SSE is the most optimum SSE indicated by complete k-means clustering, not the initial SSE calculated when centroids are first-chosen randomly.

[2]The largest possible number of clusters is the number of data points since it is irrational to have more clusters than data points.

K-means clustering groups data points based on the nearest centroid using the Euclidean distance, and centroids change repeatedly by the optimising formula

$$\mathbf{C}_{S_j} = \frac{1}{N(S_j)} \sum_{\mathbf{X}_i \in S_j} \mathbf{X}_i,$$

where $N(S_j)$ denotes the number of data points in the cluster S_j. The change stops when no data points move to another cluster, thus implying the lowest possible SSE. If the number of clusters increases, the number of centroids also increases. Hence, distances between data points and their nearest centroids tend to shrink, resulting in a smaller SSE. By this deduction, we conclude that the sequence $(SSE(k))$ is monotonically decreasing as greater clusters imply smaller SSEs.

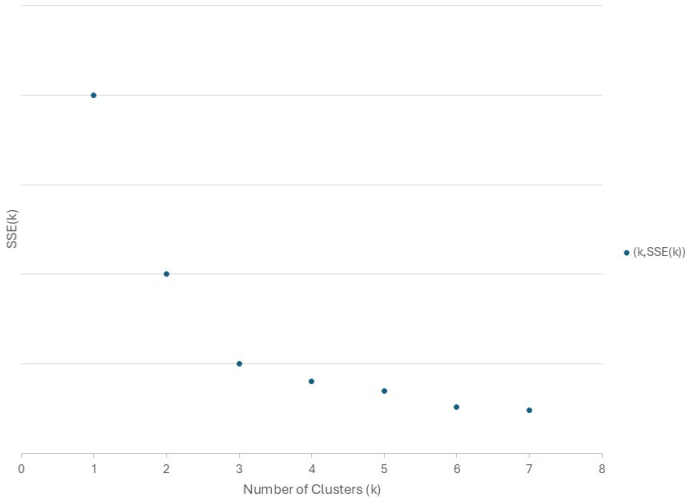

FIGURE 5. Initial SSE plotting

The sequence $SSE(k)$, as illustrated by Figure 5, is plotted monotonically decreasing. We connect adjacent points by a straight line such that all point (terms of $SSE(k)$) form a continuous function consisting of several straight lines. Every corner of the function graph forms an angle that is used to determine the elbow point.

The elbow point is the point in which there is no significant drop of SSE afterwards. The typical elbow method determines the angle closest to 90° since it indicates the last point with a significant drop of SSE i.e. the graph starts flattening beyond the point. However, since the elbow method, at the beginning, is a heuristic method, the closest-to-90° choice is rather ambiguous. Two disputes need to be addressed. First, the angle measurement is not mathematical. Second, the method

appears to ignore the possibility of facing-downwards corners chosen as the elbow point (illustrated by Figure 6). The point may indicate the closest one to 90◦ while there is still a significant SSE drop afterwards.

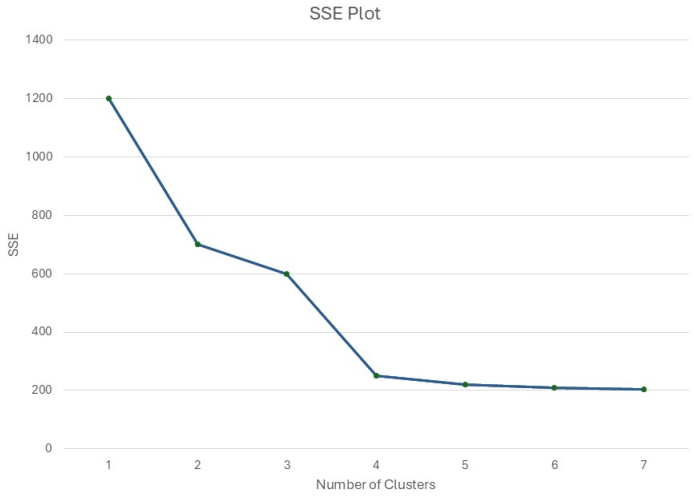

FIGURE 6. SSE plot with a facing-downwards corner

To address the first dispute, we construct a formula to measure the exact angle of the corners. We use the formula of angle between lines that uses line slopes.

Theorem 2.1. *Let l_1 and l_2 be lines with inclination θ_1 and θ_2 respectively. Let m_{l_1} and m_{l_2} denote the slope of l_1 and l_2 respectively where $m_{l_1} = \tan\theta_1$ and $m_{l_2} = \tan\theta_2$. The two lines intersect in a point, forming two supplementary angles, ϕ and ψ as illustrated by 7. The following holds:*

$$\tan\phi = \frac{m_{l_1} - m_{l_2}}{1 + m_{l_1} m_{l_2}}$$

and

$$\tan\psi = -\tan\phi.$$

The proof is constructed in [18] using the definition of slope, relation among interior and exterior angles of a triangle, and the property of tangent of sum of angles.

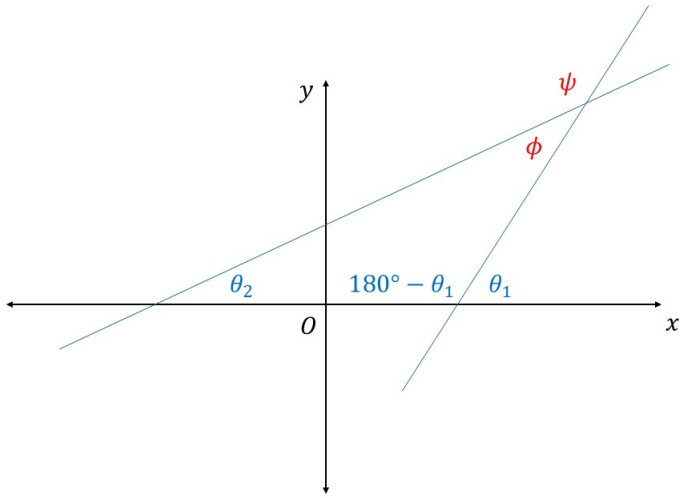

FIGURE 7. Formula of angles between two lines

Based on 7, ϕ and ψ are supplementary. Therefore, unless both are 90°, the former is located in the first quadrant (acute), and the latter is located in the second quadrant (obtuse), or vice versa. The angle located in the first quadrant has a positive tangent, whereas the one located in the second quadrant has a negative tangent.

We use the modified Theorem 2.1 to construct angle measurement formula of the SSE graph since the line behaviours are different. However, in this case, we first assume that the second dispute does not exist i.e. the lines forming the graph get more flattened after each term. The flattening condition is indicated by flattening lines i.e. slopes of the lines get closer to 0 as k increases (the slope of horizontal lines). In other words, we assume that

$$m_{l_k} > m_{l_{k-1}}$$

for $k = 1, 2, ..., n$.

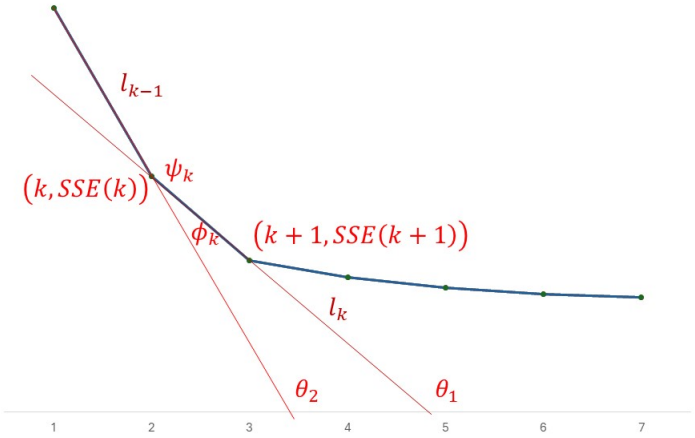

FIGURE 8. Implementation of Theorem 2.1 on SSE plotting

Theorem 2.2. *Let l_k denote the straight line connecting the point $(k, SSE(k))$ and $(k+1, SSE(k+1))$, and ψ_k denote the angle of the corner facing upwards. The elbow point is $(k, SSE(k))$ such that it satisfies*

$$\min\left(\tan\psi_k | k = 2, 3, ..., n-1\right) \tag{1}$$

where

$$\tan(\psi_k) = \frac{-SSE(k+1) + 2SSE(k) - SSE(k-1)}{1 + (SSE(k) - SSE(k-1))(SSE(k+1) - SSE(k))}.$$

To prove Theorem 2.2, we first consider the following proposition to prove that ψ_k is always between $90 \circ *and * 180 \circ *$. This fact ensures that the angle closest to $90 \circ *$ has the smallest tangent.

Proposition 2.3. *If ψ is the angle that faces upward, we have $90° < \psi < 180°$.*

Proof. If l_k denotes the straight line connecting the point $(k, SSE(k))$ and $(k+1, SSE(k+1))$, since $SSE(k+1) < SSE(k)$ for $k = 1, 2, ..., n$, we have

$$\begin{aligned} m_{l_k} &= \frac{SSE(k+1) - SSE(k)}{(k+1) - (k)} \\ &= SSE(k+1) - SSE(k) \\ &< 0. \end{aligned}$$

Moreover, we assume $m_{l_k} > m_{l_{k-1}}$ which implies $m_{l_k} - m_{l_{k-1}} > 0$. Consequently, since $m_{l_k} m_{l_{k-1}} > 0$, we have

$$\begin{aligned}\tan\phi_k &= \frac{m_{l_k} - m_{l_{k-1}}}{1 + m_{l_k} m_{l_{k-1}}} \\ &> 0\end{aligned}$$

and

$$\begin{aligned}\tan\psi &= -\tan\phi \\ &< 0.\end{aligned}$$

Since $\tan\psi < 0$, the angle ψ must be in quadrant II or III i.e. $90° < \psi < 180°$ or $180° < \psi < 270°$. However, since ψ and ϕ are supplementary, it must not exceed $180°$. Therefore, we have $90° < \psi < 180°$. □

Here is the proof of Theorem 2.2.

Proof. Recall that l_k denote the straight line connecting the point $(k, SSE(k))$ and $(k+1, SSE(k+1))$. As illustrated by Figure 8, we have θ_1 and θ_2 are the inclination of l_{k-1} and l_k respectively. Since $\psi + (180° - \theta_1) + \theta_2 = 180°$, we have

$$\begin{aligned}\tan\phi &= \tan\left(180° - ((180° - \theta_1) + \theta_2)\right) \\ &= \tan\left(\theta_1 - \theta_2\right) \\ &= \frac{\tan\theta_1 - \tan\theta_2}{1 + \tan\theta_1 \tan\theta_2} \\ &= \frac{m_{l_k} - m_{l_{k-1}}}{1 + m_{l_k} m_{l_{k-1}}}.\end{aligned}$$

Elbow point candidates are represented by the angle supplementary to ϕ, that is ψ. Since the angles are supplementary, the angles have opposites tangents i.e.

$$\begin{aligned}&\tan\psi = -\tan\phi \\ \iff &\tan\psi = -\frac{m_{l_k} - m_{l_{k-1}}}{1 + m_{l_k} m_{l_{k-1}}} \\ \iff &\tan\psi = \frac{m_{l_{k-1}} - m_{l_k}}{1 + m_{l_k} m_{l_{k-1}}}. \qquad (2)\end{aligned}$$

The endpoints, $(1, SSE(1)$ and $(n, SSE(n))$, do not form an angle. Therefore, Equation 2 only holds for $k = 2, 3, ..., n-1$.

By the formula of slope of a straight line, since l_k connects $(k, SSE(k))$ and $(k+1, SSE(k+1))$, we have

$$m_{l_k} = \frac{SSE(k+1) - SSE(k)}{(k+1) - k}$$

and

$$m_{l_{k-1}} = \frac{SSE(k) - SSE(k-1)}{k - (k-1)}.$$

Therefore, Equation 2 can be written as

$$\begin{aligned}\tan\psi_k &= \frac{m_{l_{k-1}} - m_{l_k}}{1 + m_{l_k} m_{l_{k-1}}} \\ &= \frac{\frac{SSE(k)-SSE(k-1)}{(k+1)-k} - \frac{SSE(k+1)-SSE(k)}{k-(k-1)}}{1 + (\frac{SSE(k+1)-SSE(k)}{(k+1)-k})(\frac{SSE(k)-SSE(k-1)}{k-(k-1)})} \\ &= \frac{SSE(k) - SSE(k-1) - (SSE(k+1) - SSE(k))}{1 + (SSE(k+1) - SSE(k))(SSE(k) - SSE(k-1))} \\ &= \frac{-SSE(k-1) + 2SSE(k) - SSE(k+1)}{1 + (SSE(k+1) - SSE(k))(SSE(k) - SSE(k-1))}.\end{aligned}$$

By Proposition 2.3, we have $90° < \psi < 180°$. Furthermore, since $\tan\psi$ is monotonically increasing within the interval[3], the angle ψ closest to 90° must be satisfied by the smallest (most negative) $\tan\psi$. Therefore, we conclude that the elbow point is $(k, SSE(k))$ that satisfies

$$\min\left(\tan\psi_k | k = 2, 3, ..., n-1\right).$$

□

2.2. **Alternative Method.** Theorem 2.2 holds if the second dispute is assumed to be untrue i.e. a facing downward corner, whose angle is the closest to 90∘, is probably chosen as the elbow point despite a still significant SEE drop afterwards. Some data sets cause the elbow method graph violating the assumption. In other words, there can be a case that $m_{l_k} \geq m_{l_{k-1}}$.

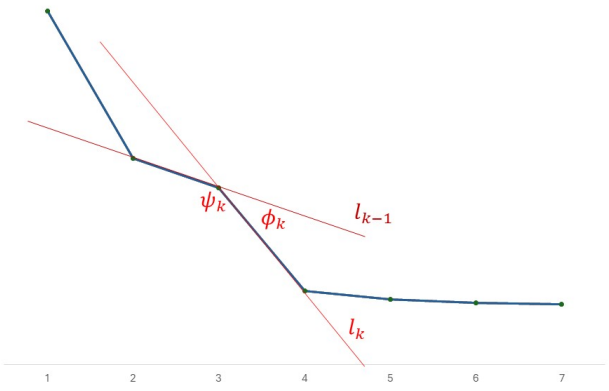

FIGURE 9. Facing-downwards corner impacts on elbow point determination

[3]The derivative of $\tan\psi$ with respect to ψ is $\sec^2\psi$ that is always positive for $90° < \psi < 180°$. Therefore, the function increases.

In Figure 9, the corner that is an intersection of l_1 and l_2 may be chosen if it is the closest angle to 90° if the usual elbow point criteria is used. However, it is clear that the corner must no be the elbow point as there a larger drop of SSE onwards. It does not necessarily indicate the sign of flattening SSE drops. In this case, $k = 4$ is the more appropriate elbow point since after $k = 3$, there is still a noticeable SSE drop.

Proposition 2.4. *If* $m_{l_k} \leq m_{l_{k-1}}$, *then* $SSE(k) - SSE(k+1) \geq SSE(k-1) - SSE(k)$.

Proof. The drop of $SSE(k)$ to $SSE(k+1)$ is $|SSE(k) - SSE(k+1)|$, or for simplicity, since $SSE(k)$ is monotonically decreasing, we have $SSE(k) - SSE(k+1)$ for $k = 1, 2, ..., n$.

Suppose that $m_{l_k} \leq m_{l_{k-1}}$. We then have

$$\begin{aligned}
& m_{l_k} \leq m_{l_{k-1}} \\
\iff & \frac{SSE(k+1) - SSE(k)}{(k+1) - k} \leq \frac{SSE(k) - SSE(k-1)}{k - (k-1)} \\
\iff & SSE(k+1) - SSE(k) \leq SSE(k) - SSE(k-1) \\
\iff & -(SSE(k+1) - SSE(k)) \geq -(SSE(k) - SSE(k-1)) \\
\iff & SSE(k) - SSE(k+1) \geq -SSE(k-1) - SSE(k) \qquad (3)
\end{aligned}$$

The Equation 3 indicates a higher drop after the point $(k, SSE(k))$. □

Figure 10 illustrates the Proposition 2.4.

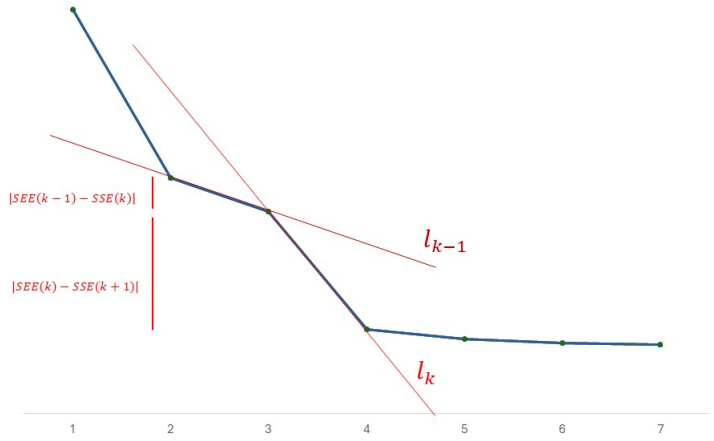

FIGURE 10. Further Effects of Facing-downwards corners

Figure 10 shows that l_k has a larger slope that of l_{k-1}. It indicates a higher SSE drop after the intersection of the two lines as shown by Equation 3. The corner of the intersection faces downwards.

By discovering the effect of ignoring the second dispute to the elbow point choice, we reconstruct Theorem 2.2 such that it is more universal. We use the same criteria (the closest-to-90°) with an additional condition: neglecting the point in Proposition 2.4. In this case, $k = 4$ is the more appropriate elbow point since after $k = 3$, there is still a noticeable SSE drop.

Theorem 2.5. *Suppose that* $SSE(k+1) \leq SSE(k)$ *for* $k = 1, 2, ..., n$. *Let* l_k *denote the straight line connecting the point* $(k, SSE(k))$ *and* $(k+1, SSE(k+1))$, *and* ψ_k *denote the angle of the corner facing upwards. The elbow point is* $(k, SSE(k))$ *such that it satisfies*

$$\min\left(\tan\psi_k | k = 2, 3, ..., n-1; m_{l_k} > m_{l_{k-1}}\right)$$

where

$$\tan(\psi_k) = \frac{-SSE(k+1) + 2SSE(k) - SSE(k-1)}{1 + (SSE(k) - SSE(k-1))(SSE(k+1) - SSE(k))}. \tag{4}$$

Theorem 2.5 adds a condition that is the negation of condition in Proposition 2.4 into Equation 1. Hence, points indicating a rise of SSE drop, satisfying the condition in Proposition 2.4 is neglected.

2.3. **Program Implementation and Simulation.** K-means clustering is often implemented by software, computationally, especially if it involves a large data set. Hence, we also provide an implementation of the constructed formula into computer programming. Here is the pseudo-code.

Algorithm 1 Elbow Method Pseudocode

Input a data set consisting of n m-dimensional data points $X_i = (x_1, x_2, ..., x_p)$
Input array $tanpsi$ with size n
Define function $d(X_i, Y_i)$
  for i = 1 to p do
    sum = sum + $(x_i - y_i)$^2
  end for
  return sqrt(sum)
Define function $SSE(k)$
  do k-means to data set with k cluster(s)
  define centroid C_i
  for i = 1 to k do
    sum = sum + $d(X_i, C_i)$
  end for
  return sum
for k = 2 to n-1 do
  if $SSE(k) - SSE(k+1) \geq SSE(k-1) - SSE(k)$ then
    $tanpsi[k] = 0$
  else
    $tanpsi[k] = (-SSE(k+1) + 2SSE(k) - SSE(k-1))/(1 + (SSE(k) - SSE(k-1))(SSE(k+1) - SSE(k)))$
  end if
end for
angle = minimum($tanpsi$)
do k-means clustering with k cluster(s) where $tanpsi[k]$ = angle

The algorithm begins with defining main variables used for k-means clustering and the elbow method namely a data set and an array containing list of $\tan\psi$. We define some functions to simplify the algorithm that are the Euclidean distance and the SSE. The Euclidean distance function involves a looping as a representative of consecutive summation of squared differences; the SSE function also involves a looping of Euclidean distances between every variable and its centroid.

The next step is creating a looping to put $\tan\psi_k$ consecutively into the defined array. The formula is of Equation 4. We also put a condition of Theorem 2.5 to ignore corners facing upwards. We put $\tan\psi_k = 0$ for this kind of points. Hence, the points will not be chosen as the elbow point since they will not become the minimum among negative tangents.

Lastly, we define the variable "angle" as the minimum of $\tan\psi_k$. The k-means clustering is then run with the number of clusters.

Some programming languages already include standard functions used in this algorithm, such as the Euclidean distance, SSE, and k-means itself. Python is one oh such programming languages. Here is the implementation in Python for some sample data set (we can change it to different data set).

Algorithm 2 The Elbow Method Implementation on Python

```
import math
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import numpy as np
# Sample 2D data
X = np.array([[1, 1], [1.5, 1.8], [5, 8], [8, 8], [10, 0.6], [9, 11], [0, 1], [3, 4]])
# Get the number of data points
dimension = X.shape[0]
# Calculate WCSS (Within-Cluster Sum of Squares)
wcss = []
for i in range(1, dimension + 1):
kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,
random_state=0)
kmeans.fit(X)
wcss.append(kmeans.inertia_)
# Plot the Elbow Method graph
plt.plot(range(1, dimension + 1), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('SSE')
plt.show()
# Calculate tan(psi)
tanpsi = [0]
for i in range(1, dimension - 1):
slope1 = wcss[i] - wcss[i - 1]
slope2 = wcss[i + 1] - wcss[i]
if slope2 ¡= slope1:
tanpsi.append(0)
else:
hasil = (slope1 - slope2) / (1 + (slope2 * slope1))
tanpsi.append(hasil)
tanpsi.append(0)
# Print tan(psi) values
for i in range(len(tanpsi)):
print(f"Tanpsi(i + 1) = {tanpsi[i]}")
# Get optimal number of clusters
optimal_k = tanpsi.index(min(tanpsi)) + 1
print(f"The most optimum number of clusters is {optimal_k}")
```

Here is the output of the given implementation on Python.

```
kmeans = KMeans(n_clusters=optimal_k, init='k-means++', max_iter=300,
n_init=10, random_state=0)
y_kmeans = kmeans.fit_predict(X)
# Plot centroids in the background
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300,
c='yellow', label='Centroids', zorder=1)
# Plot data points for each cluster
colors = ['red', 'blue', 'green', 'purple', 'orange', 'cyan']
for i in range(optimal_k):
plt.scatter(X[y_kmeans == i, 0], X[y_kmeans == i, 1], s=100, c=colors[i %
len(colors)], label=f'Cluster {i + 1}', zorder=2)
plt.title('Clusters of data points')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.show()
```

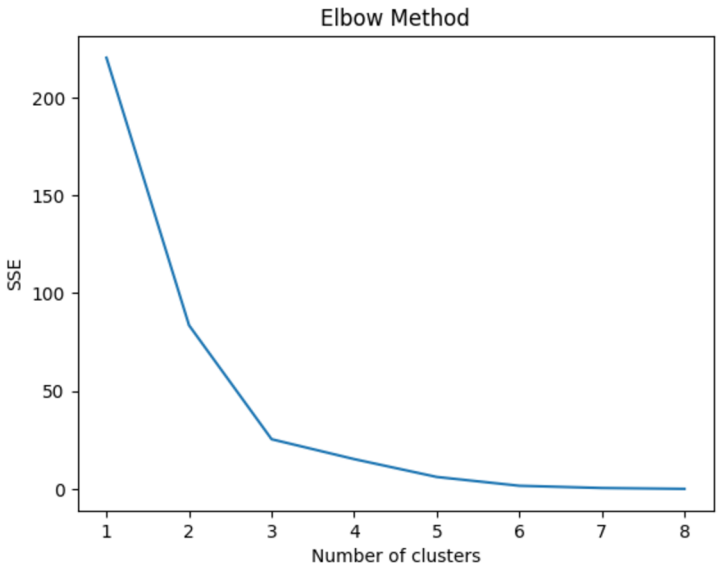

FIGURE 11. SSE plot of Python simulation (distorted)

Figure 11 shows that the most optimum number of cluster is 3. However, as described in Introduction, it could be misleading. It is answered by Figure 12 which is the output of the Python code for $\tan\psi$ calculation.

```
Tanpsi(1) = 0
Tanpsi(2) = -0.00985448658656615
Tanpsi(3) = -0.08105739083529678
Tanpsi(4) = -0.011118534000702132
Tanpsi(5) = -0.10994541365748306
Tanpsi(6) = -0.5434400756654507
Tanpsi(7) = -0.465472835468589
Tanpsi(8) = 0
The most optimum number of clusters is 6
```

FIGURE 12. Values of $\tan \psi_k$

Figure 12 shows that the smallest tangent is obtained when $k = 6$. It shows that the distorted SSE plot in Figure 11 is indeed misleading.

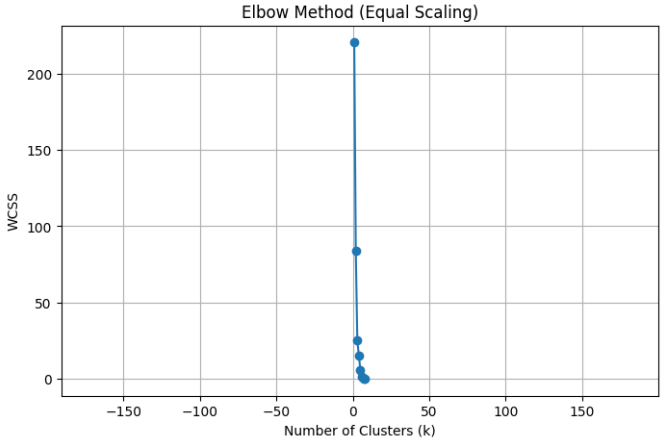

FIGURE 13. SSE plot of Python simulation (undistorted)

Figure 13 shows the undistorted plot of SSE. Despite the quite unclear plot, we can conclude that 3 is not the most optimum number of clusters since there is still indeed a significant drop afterwards. The plot starts flattening after $k = 6$. The result of the k-means clustering is shown in Figure 14.

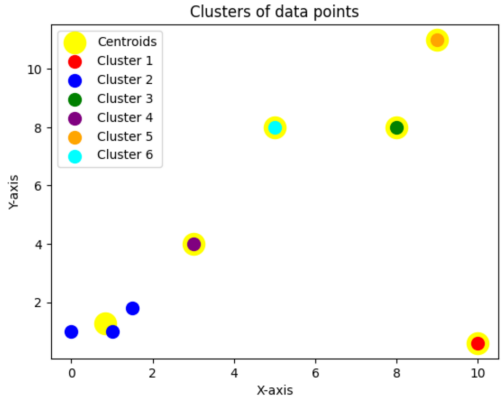

FIGURE 14. K-means implementation for $k = 6$

In comparison to Figure 14, Figure 15 shows the k-means clustering if there are 3 clusters, mistakenly chosen by the biased Figure 11. We see that the clustering still leaves a huge distance between data points and their corresponding centroids. Figure 14 shows a better clustering where the distances are minimum.

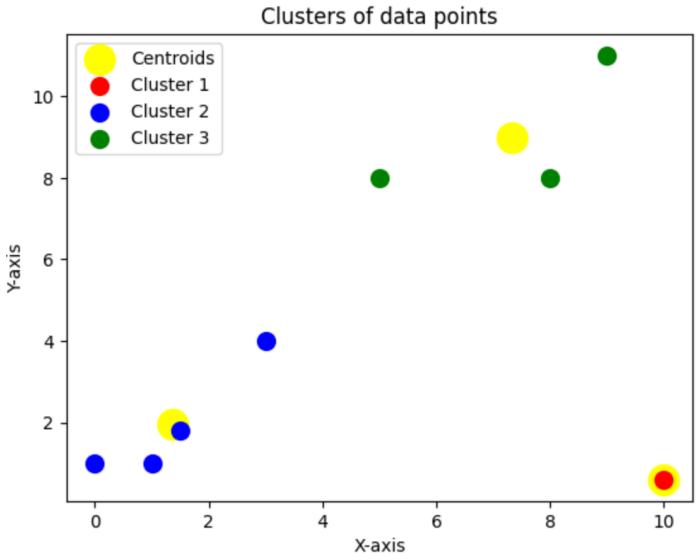

FIGURE 15. Figure example

3. CONCLUDING REMARKS

The elbow method, known as a heuristic method, is made exact using the formula of angle between lines. Suppose that $(k, SSE(k))$ are tuple points of number of clusters and corresponding SSE. The elbow point is the point that satisfy

$$\min\left(\tan\psi_k | k = 2, 3, ..., n-1; m_{l_k} > m_{l_{k-1}}\right)$$

where

$$\tan(\psi_k) = \frac{-SSE(k+1) + 2SSE(k) - SSE(k-1)}{1 + (SSE(k) - SSE(k-1))(SSE(k+1) - SSE(k))}$$

and $m_{l_k} = SSE(k+1) - SSE(k)$ for $k = 1, 2, ..., n$. The algorithm is simple such that it is easily implementable in programming. The precision provided by this method also eliminates the possibility of choosing an incorrect elbow point since it results the same regardless of the visual representation.

Data Availability Statement
Data used in the research are simulation data as an example of the main result algorithm implementation.

Declarations. The authors declare that this research was conducted in absence of any commercial or financial intervention that can influence the research results.

Author Contributions. Indra Herdiana: conceptualisation, methodology, formal analysis, resources, writing - original draft, project administration, funding acquisition. Mokhamad Alfin Kamal: conceptualisation, methodology, software, validation, visualisation. Triyani: validation, supervision, funding acquisition. Mutia Nur Estri: data curation, visualisation, writing - review and editing. Renny: investigation, resources, writing - review and editing. All authors discussed the final results.

Funding Statement. The research is funded by Institute of Research and Community Service (LPPM), Universitas Jenderal Soedirman, under the contract number 26.713 /UN23.35.5/PT.01/II/2024.

Acknowledgment. We would like to express gratitude towards Department of Mathematics of Jenderal Soedirman University and Institute of Research and Community Service (LPPM) of Jenderal Soudirman University for supporting this research work continuously.

REFERENCES

[1] J. M. Nolin, "Data as oil, infrastructure or asset? three metaphors of data as economic value," *Journal of Information, Communication and Ethics in Society*, vol. 18, pp. 28–43, 11 2019. https://doi.org/10.1108/JICES-04-2019-0044.

[2] D. Priyanti and S. Iriani, "Sistem informasi data penduduk pada desa bogoharjo kecamatan ngadirojo kabupaten pacitan," *Indonesian Journal of Network & Security*, vol. 2, pp. 55–61, 2013. http://dx.doi.org/10.55181/ijns.v2i4.181.

[3] N. K. Usada and A. Prabawa, "Analisis manajemen pengelolaan data sistem informasi puskesmas di tingkat dinas kesehatan di kabupaten bondowoso," *Jurnal Biostatistik, Kependudukan, dan Informatika Kesehatan*, vol. 2, p. 16, 2021. https://doi.org/10.7454/bikfokes.v2i1.1020.

[4] Z. Karaca, "The cluster analysis in the manufacturing industry with k-means method: An application for turkey," *Eurasian Journal of Economics and Finance*, vol. 6, pp. 1–12, 9 2018. https://doi.org/10.15604/ejef.2018.06.03.001.

[5] N. Negi and G. Chawla, "Clustering algorithms in healthcare," 2021. https://doi.org/10.1007/978-3-030-67051-1_13.

[6] N. Solikin, B. Hartono, S. Sugiono, and L. Linawati, "Farming in kediri indonesia: analysis of cluster k-means," *IOP Conference Series: Earth and Environmental Science*, vol. 1041, p. 012015, 6 2022. https://doi.org/10.1088/1755-1315/1041/1/012015.

[7] G. J. Oyewole and G. A. Thopil, "Data clustering: application and trends," *Artificial Intelligence Review*, vol. 56, pp. 6439–6475, 7 2023. https://doi.org/10.1007/s10462-022-10325-y.

[8] B. Everitt, *Cluster Analysis*. Wiley, 2011. https://doi.org/10.1002/9780470977811.

[9] J. Zhao, Y. Bao, D. Li, and X. Guan, "An improved k-means algorithm based on contour similarity," *Mathematics*, vol. 12, p. 2211, 7 2024. https://doi.org/10.3390/math12142211.

[10] A. Martino, A. Ghiglietti, F. Ieva, and A. M. Paganoni, "A k-means procedure based on a mahalanobis type distance for clustering multivariate functional data," *arXiv preprint*, 8 2017. https://doi.org/10.48550/arXiv.1708.00386.

[11] M. A. Sembiring, "Penerapan metode algoritma k-means clustering untuk pemetaan penyebaran penyakit demam berdarah dengue (dbd)," *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, vol. 4, p. 336, 10 2021. https://doi.org/10.54314/jssr.v4i3.712.

[12] A. Amelia, I. M. Nur, M. Rizky, and S. P. Milasari, "Poverty level grouping in west java province with the k-means clustering method," *Journal Of Data Insights*, vol. 1, pp. 51–61, 12 2023. https://doi.org/10.26714/jodi.v1i2.152.

[13] X. Li, L. Yu, L. Hang, and X. Tang, "The parallel implementation and application of an improved k-means algorithm," *J. Univ. Electron. Sci. Technol*, vol. 46, pp. 61–68, 2017. https://doi.org/10.3969/j.issn.1001-0548.2017.01.010.

[14] C. Yuan and H. Yang, "Research on k-value selection method of k-means clustering algorithm," *J*, vol. 2, pp. 226–235, 6 2019. https://doi.org/10.3390/j2020016.

[15] C. Shi, B. Wei, S. Wei, W. Wang, H. Liu, and J. Liu, "A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, p. 31, 12 2021. https://doi.org/10.1186/s13638-021-01910-w.

[16] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020. https://doi.org/10.1109/ACCESS.2020.2988796.

[17] J. Pitt-Francis and J. Whiteley, *Guide to Scientific Computing in C++*. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-73132-2.

[18] G. Fuller and D. Tarwater, *Analytic Geometry*. Pearson, 1992. https://books.google.co.id/books/about/Analytic_Geometry.html?id=cZbuPwAACAAJ&redir_esc=y.