**Accredited SINTA 2 Ranking**
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

**JURNAL RESTI**
**(Rekayasa Sistem dan Teknologi Informasi)**
Vol. 8 No. 5 (2024) 623 - 630 e-ISSN: 2580-0760

# Quantum-Enhanced K-Medoids Clustering: Comparative Analysis of Stroke Medical Data

Rikardo Hotman Siahaan[1*], Swingly Purba[2], Jeremia Siregar[3], Marvin Frans Sakti Hutabarat[4], Rasmi Sitohang[5]
[1,2,3] Informatics Engineering Study Program, Institut Sains dan Teknologi TD Pardede, Medan, Indonesia
[4]Electrical Engineering Study Program, Institut Sains dan Teknologi TD Pardede, Medan, Indonesia
[5]Industrial Engineering Study Program, Institut Sains dan Teknologi TD Pardede, Medan, Indonesia
[1]ricardoricardo3010@gmail.com, [2]swingly.purba61@gmail.com, [3]jeremiasiregar@istp.ac.id, [4]marvin.hutabarat@gmail.com,
[5]rasmisitohang83@gmail.com

## *Abstract*

*Stroke is a severe medical condition that occurs when the blood supply to parts of the brain is interrupted or reduced, resulting in brain tissue that lacks oxygen and nutrients. This causes brain cells to start to die in minutes. Early prevention reduces the risk of stroke. In this study, a quantum computing approach is used to improve the performance of the K-Medoids method. A comparative analysis of these methods was carried out with a focus on their performance, especially on the accuracy of the test results. The investigation was carried out using a data set of stroke patient medical records. The data set was tested using the classical and K-Medoids methods with a quantum computing approach utilizing Manhattan distance calculations. The findings of this research reveal improvements in the K-Medoids algorithm with Manhattan distance calculation influenced by the integration of a quantum computing framework. In particular, the simulation test results show an increase in accuracy from the classical K-Medoids method to the K-Medoids method with a quantum computing approach, from 52% to 64%. These results highlight that the performance of the K-Medoids method with a quantum computing approach is superior to that of the classical K-Medoids method.*

*Keywords: K-Medoids; Manhattan; Performance; Quantum Computing; Stroke*

*How to Cite:* R. Siahaan, S. Purba, J. Siregar, M. F. S. Hutabarat, and R. Sitohang, "Quantum-Enhanced K-Medoids Clustering: Comparative Analysis of Stroke Medical Data", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 8, no. 5, pp. 623 - 630, Oct. 2024.]
*DOI*: https://doi.org/10.29207/resti.v8i5.5814

## 1. Introduction

Quantum computing is a field of computer science that utilizes principles of quantum mechanics to perform computations. Unlike classical computing, which uses bits as the basic unit of information (represented as 0 or 1), quantum computing employs qubits (quantum bits) that can exist in states of 0, 1, or superpositions of both. Some key concepts in quantum computing include superposition, where qubits can be in superposition, meaning they can represent multiple values simultaneously, and entangled states, where quantum entangled qubits are interconnected, allowing changes to one qubit to directly affect another, even if they are separated over a distance [1], [2], [3].

Clustering technology is a technique used to group data or objects based on specific similarities or patterns. This technology is frequently used in various fields such as data analysis, image processing, and pattern recognition. In the context of data analysis, clustering technology is used to organize data into groups with similar characteristics or properties. For example, in market analysis, clustering can be used to group customers based on their preferences or shopping behaviors. In the field of image processing, clustering technology can be used to group pixels in an image based on their color or texture.

This can assist in visual analysis, object recognition, or image compression. Clustering technology commonly utilizes specialized algorithms such as K-Means [4], hierarchical clustering [5], K-Medoids [6], [7], DBSCAN [8] and many others. Its primary objective is to uncover hidden structures or patterns within the data efficiently and effectively.

The primary reference for this research is research [9], which conducted an analysis of two data clustering techniques: K-means and K-medoids algorithms using

Received: 17-05-2024 | Accepted: 18-06-2024 | Published Online: 07-10-2024

various distance metrics. In addition, data transformation techniques were applied, including scaling, range adjustment, and the Yeo-Johnson method. The results of the transformed data were then utilized in the K-means and K-medoids algorithms to produce accurate data clustering. The results generated from the transformed data were found to be superior to those obtained from the raw data. The Yeo-Johnson transformation method was identified as the most effective for the K-means, K-medoids, and Rank k-medoids algorithms, achieving average accuracies of 67.58%, 69.58%, and 72.64% respectively.

The main issue of this research is to find an alternative to the K-Medoids method for clustering using quantum computing. The classical K-Medoids method utilizes classical operations, whereas the K-Medoids method with a quantum computing approach uses qubits, which can be in states 0, 1, or a superposition of 0 and 1 simultaneously. The aim of this research is to develop the K-Medoids method with quantum computing to achieve better clustering results in terms of accuracy. The novelty of this research lies in the clustering method using the K-Medoids algorithm with a quantum computing approach.

## 2. Research Methods

This research uses the K-Medoids Data Mining method by calculating the Manhattan Distance with a Quantum computing approach to obtain more accurate results in grouping stroke medical record data. This research compares the classical K-Medoids method with the K-Medoids method with a quantum computing approach to get better grouping results in predicting stroke.

### *2.1 Related Research*

Several studies on the K-Medoids method, such as research [10], aim to improve the efficiency of the k-medoids algorithm by proposing a new approach called BanditPAM++, which can reduce the number of computations required to determine the medoids. BanditPAM++ employs a strategy that combines techniques from bandit mastery learning with the PAM (Partitioning Around Medoids) algorithm, enabling this approach to adaptively select the most informative medoid-candidate pairs for further analysis. Testing BanditPAM++ on several datasets has shown that this approach is faster than the classical k-medoids algorithm, while still providing comparable clustering quality. This research makes a significant contribution to the development of faster and more efficient clustering algorithms, particularly important for large-scale data analysis.

Research [11] aims to enhance the efficiency of k-medoid clustering algorithms by reducing the number of iterations required to achieve convergence towards optimal medoids, employing a method known as the purity Algorithm. This approach utilizes specific strategies or techniques to minimize the number of iterations needed for convergence, particularly focusing on addressing issues of algorithm slowness due to excessive iterations, especially with large or complex datasets. The study's contribution lies in advancing more efficient and effective clustering algorithms through the application of the Purity Algorithm in the context of k-medoid clustering, promising significant benefits in handling clustering problems at larger scales with reduced computational time.

Research [12] aims to compare the clustering methods of K-Means and K-Medoids in the context of Twitter data analysis. K-means and K-medoids are two commonly used approaches to clustering data. K-Means use the mean as the center representation of clusters, whereas K-Medoids use actual data points (medoids) as the center representation of clusters. This study compares the performance, accuracy, and stability of these two clustering methods, particularly in Twitter data. The findings of this research provide valuable information on which clustering method is more effective and suitable for social media data such as Twitter, which can be beneficial for sentiment analysis applications, trend identification, and user behavior modeling.

This study [13] discusses the occurrence of earthquakes in Indonesia, a consequence of the country's location at the meeting point of multiple tectonic plates, leading to a complex arrangement of plates. Earthquakes vary in impact based on their magnitude and depth. The research aimed to classify earthquake data in Indonesia by magnitudes and depths using clustering techniques, specifically the k-medoids and k-means algorithms. The k-medoids algorithm employs the clustering large application (CLARA) approach, grouping data into clusters with medoids as centroids.

On the other hand, k-means divide the data into clusters based on the proximity to the closest average. The study concluded that the CLARA algorithm yielded the best clustering results for earthquake data in Indonesia, identifying five clusters with member totals ranging from 199 to 2392 objects in clusters 1 to 5 respectively. This research provides insight into earthquake patterns in Indonesia and underscores the importance of data mining techniques for disaster risk management and mitigation efforts.

This research [14] highlights the development and availability of a fast implementation of the FasterPAM algorithm using Rust, which includes optional parallelization, as well as an easy-to-use Python wrapper. The k-medoids clustering algorithm is described as particularly valuable in scenarios where the input data are not continuous and the Euclidean distance may not be applicable. By providing this software package, the authors aim to increase the accessibility of the k-medoids algorithm for data scientists in various domains. Furthermore, the availability of source code is expected to support researchers in data mining efforts to advance clustering algorithms.

This study [15] focuses on achieving global deterministic optimization of the K-Medoids clustering problem. The researchers propose a branch-and-bound (BB) scheme that utilizes a specialized Lagrangian relaxation method from the 1970s to establish a lower bound at each node in the branch-and-bound tree. The lower bounding method ensures maximum gap control at the root node and features a closed-form solution that can be analytically derived without explicitly solving optimization problems, facilitating easy parallel computation. Additionally, by leveraging this lower bounding technique, finite convergence to the global optimal solution is ensured by branching exclusively on medoid regions.

The study also introduces several custom-bound tightening techniques aimed at reducing search space and computational expense. Through extensive computational experiments in 28 machine learning datasets, the researchers demonstrate that their algorithm can achieve a provable global optimal solution with an optimality gap of 0.1% within a 4-hour timeframe, even with datasets containing up to one million samples. Furthermore, the algorithm consistently produces objective values that are equal to or better than those obtained using heuristic methods. The study includes a theoretical proof of global convergence for the proposed algorithm, highlighting its effectiveness and reliability in solving the clustering problem.

The study [16] applies the RFM (Recency, Frequency, Monetary) model combined with K-Means, K-Medoids and DBSCAN algorithms to analyze transaction data and identify potential customers. The dataset, consisting of 334,641 transactions, is converted into 1661 RFM data lines. The algorithms (K-Means, K-Medoids, and DBSCAN) are sensitive to initial cluster center or parameter values, which are randomly initialized. Clustering is performed with two to six clusters, and the trial process involves setting random centroid values (for K-Means and K-Medoids) or Epsilon and Min Points (for DBSCAN) to identify clusters representing potential customers.

Cluster validation is carried out using the Davies-Bouldin Index and Silhouette Index methods. The study finds that K-Means exhibits the highest validity level compared to K-Medoids and DBSCAN, with the Davies-Bouldin index and Silhouette index yields of 0.33009058 and 0.912671056, respectively. The optimal number of clusters identified by these indices is 2, with each algorithm (K-Means, K-Medoids, DBSCAN) categorizing customers into Dormant and Golden customer classes.

Research [17] introduced a new BanditPAM method for clustering k-medoids with almost linear computing time. This method utilizes a multi-armed bandits approach to efficiently select candidate medoids, reducing algorithm complexity significantly compared to traditional k-medoids clustering techniques. Experimental results show that BanditPAM speeds up the clustering process and provides competitive results in terms of clustering quality compared to other methods.

Research [18] discusses applying the RFM (Recency, Frequency, Monetary) model for customer segmentation using three clustering methods: K-Means, K-Medoids, and DBSCAN. This research aims to compare the effectiveness of the three methods in grouping customers based on their RFM value. The study results show that each method has advantages and disadvantages regarding accuracy and interpretability, with DBSCAN offering the ability to detect more complex patterns than K-Means and K-Medoids.

Research [19] compared the performance of the K-Means and K-Medoids algorithms in text mining for student thesis classification. This study uses the Davies-Bouldin Index to evaluate the quality of clusters both algorithms produce. The research results show that although K-Means are faster in the clustering process, K-medoids tend to provide more stable and better results in terms of cluster quality. This research highlights the importance of selecting appropriate algorithms for text-mining applications, especially in the context of academic document classification.

Research [20] combined the results of K-Medoids clustering with particle swarm optimization (PSO) to increase the effectiveness of the measles immunization program for toddlers. This study aims to identify certain patterns in immunization data and optimize vaccine distribution strategies based on clustering results. By integrating PSO, this research succeeded in increasing accuracy and efficiency in grouping immunization data, as well as providing better recommendations for the management and implementation of immunization programs. The results show that this combined method can significantly improve immunization outcomes and the effectiveness of health programs.

Research [21] compared the performance of the K-Means algorithm with three different distance metrics: Euclidean Distance, Canberra Distance, and Manhattan Distance. This study aims to evaluate the impact of each distance metric on the effectiveness of K-Means clustering in various data scenarios. The results show that each distance metric affects clustering results differently, with Euclidean Distance providing more consistent results in most cases. At the same time, Canberra Distance and Manhattan Distance have advantages in certain situations. This research helps select appropriate distance metrics to improve K-Means clustering results based on different data characteristics.

### *2.2 Research Stages*

The steps for completing this research are depicted in Figure 1.

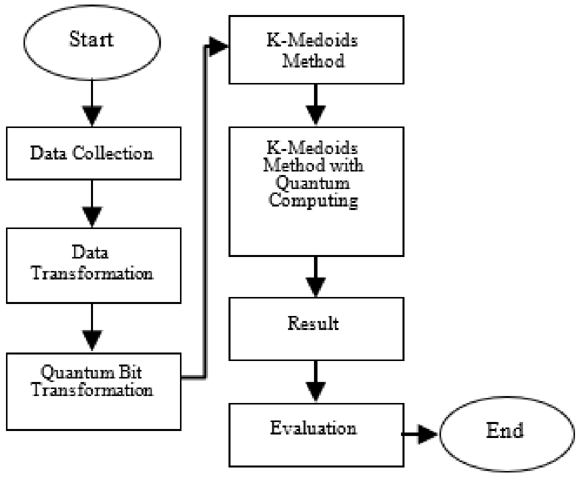

Figure 1. Research Stages

The research stages begin with data collection, where the necessary dataset to predict stroke disease is gathered. This dataset includes medical records of stroke patients, encompassing factors such as age, hypertension, heart disease, marital status, average glucose level, body mass index (BMI), and smoking status, all of which are useful in predicting the likelihood of a stroke.

Next, the data transforms into a binary representation (1 or 0) to facilitate further processing. This stage involves normalizing or encoding categorical variables into binary variables and preparing the data for clustering methods.

Following this, the binary data is transformed into quantum bits (qubits), the fundamental units of information in quantum computing, allowing information to be stored and manipulated quantumly. This transformation enables the use of the K-Medoids algorithm within a quantum computing environment.

The K-Medoids method is a data clustering technique used to group data into clusters based on similarities between data points, with the central point of each cluster (medoid) serving as its representative. When applied in a quantum computing context, the K-Medoids method can be optimized or executed on quantum computers to improve the speed or efficiency of data clustering.

The results of the clustering process using both the K-Medoids method and the K-Medoids method with quantum computing are then analyzed, revealing the structure and patterns within data identified by the clustering algorithms.

Finally, the evaluation stage assesses the performance of the K-Medoids method and the K-Medoids method with quantum computing in predicting stroke disease. This evaluation may include metrics such as accuracy to determine how effectively the models predict stroke conditions.

The data from the stroke medical records undergoes a transformation process based on specific rules to ensure standardized representation. This process involves using an algorithm that operates with qubits, which can take values of 0 or 1, or even 0 and 1 simultaneously. The notation used for this process is Dirac's notation, characterized by the symbols ">" and "<". Quantum computing approaches are employed to modify or enhance the values of epoch and accuracy.

The age data is categorized into distinct groups: Toddler Period (0–5 years), Childhood (5–11 years), Early Adolescence (12–16 years), Late Adolescence (17–25 years), Early Adulthood (26–35 years), Late Adulthood (36–45 years), Early Old Age (46–55 years), Late Old Age (56–65 years), and Senior Age (over 65 years). If an individual falls into the category from Early Old Age to Senior Age, the corresponding value is set to 1; otherwise, it is set to 0.

Hypertension is represented using binary values: 1 for individuals with hypertension and 0 for those without it. Similarly, heart disease is also represented using binary values: 1 for individuals with heart disease and 0 for those without it. Marital status is simplified into a binary format where 1 represents individuals who are married (or have ever been married), and 0 represents those who have never been married.

The average glucose level is categorized using a binary system: if the average glucose level is below 200, the value is set to 0; if it is 200 or higher, the value is set to 1, indicating the presence of sugar. The Body Mass Index (BMI) is also categorized: if the BMI falls within the range of 18.5 to 22.9, indicating normal body weight, the value is set to 0; if the BMI is outside this range, the value is set to 1. Smoking status is similarly represented with a binary system: if an individual is a current or former smoker, the value is set to 1, while if an individual has never smoked, the value is set to 0.

These rules ensure that the stroke medical records are standardized and can be used effectively for analysis and further processing in a consistent format. The results of the encoding into binary code can be seen in Table 1.

For example, a sample is taken from dataset number 1, namely the binary code 101111101; the code has a meaning, namely: attribute age = old-senior, hypertension = no, heart disease = yes, ever married = yes, average glucose = yes, BMI = abnormal, smoking status = not smoking, and target = yes. The data in Table 1 are changed in qubit form like Table 2.

Table 1. Stroke disease medical record dataset

| Age | Hypertension | Heart disease | Ever Married | Average Glucose | BMI | Smoking Status | Stroke |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |

Table 2. Qubit Data for Stroke Medical Records

| Age | Hypertension | Heart Disease | Ever Married | Average Glucose | BMI | Smoking Status | Stroke |
|---|---|---|---|---|---|---|---|
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ |
| $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |

| Age | Hypertension | Heart Disease | Ever Married | Average Glucose | BMI | Smoking Status | Stroke |
|---|---|---|---|---|---|---|---|
| $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |
| $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ |

### *2.3 K-Medoids with Quantum Computing*

In developing K-Medoids using quantum computing with Manhattan distance calculations, the steps for the K-Medoids algorithm with Manhattan distance are as follows. First, in the initialization step, k medoids are randomly selected from the data as initial cluster centres, which can be chosen randomly from the data or using a specific initialization strategy. Then, in the assignment step, the Manhattan distance between each data point and each medoid is calculated, and then each data point is assigned to the cluster with the closest medoid based on the Manhattan distance as shown in Equation 1.

$$d(x_i, m_j) = \sum_{k=1}^{n} \left| |x_{ik}> - |m_{jk}> \right| \qquad (1)$$

Next, in the update step, for each cluster, a non-medoid data point is selected as a candidate to become a new medoid. The total Manhattan distance from all data points in the cluster to that candidate medoid is calculated for each candidate medoid. The candidate medoid with the lowest total distance is selected as the new medoid for that cluster, and this step is repeated for each cluster.

## 3. Results and Discussions

The findings of this research include a development model of the K-Medoids clustering method to a K-Medoids method with a quantum computing approach using Manhattan distance calculations. The attribute values and medoid values are transformed into a quantum computing form. In this study, the clustering of stroke medical records data was performed using both the K-Medoids method and the K-Medoids method with Manhattan distance calculations. The test results showed an improvement in clustering accuracy from 52% to 64%.

The simulation results of testing the K-Medois algorithm have an accuracy of 52% with epoch 2. The following are the results of testing the epoch-1 and epoch-2 data, which can be seen in Tables 3 and 4.

Table 3. K-Medoids Epoch-1 Test Results

| C1 | C2 | Shortest Distance | Cluster | Data Real | Values |
|---|---|---|---|---|---|
| 4 | 4 | 4 | 1 | 1 | False |
| 5 | 3 | 3 | 2 | 1 | True |
| 4 | 2 | 2 | 2 | 1 | True |
| 3 | 3 | 3 | 1 | 1 | False |
| 4 | 2 | 2 | 2 | 1 | True |
| 4 | 4 | 4 | 1 | 1 | False |
| 2 | 0 | 0 | 2 | 1 | True |
| 6 | 4 | 4 | 2 | 1 | True |
| 3 | 3 | 3 | 1 | 1 | False |
| 6 | 4 | 4 | 2 | 1 | True |
| 6 | 4 | 4 | 2 | 1 | True |
| 4 | 4 | 4 | 1 | 1 | False |
| 3 | 3 | 3 | 1 | 1 | False |
| 6 | 4 | 4 | 2 | 1 | True |
| 4 | 6 | 4 | 1 | 1 | False |
| 3 | 1 | 1 | 2 | 0 | False |
| 4 | 4 | 4 | 1 | 0 | True |
| 3 | 1 | 1 | 2 | 0 | False |
| 3 | 1 | 1 | 2 | 0 | False |
| 2 | 0 | 0 | 2 | 0 | False |
| 3 | 3 | 3 | 1 | 0 | True |
| 4 | 2 | 2 | 2 | 0 | False |
| 4 | 4 | 4 | 1 | 0 | True |
| 6 | 4 | 4 | 2 | 0 | False |
| 2 | 4 | 2 | 1 | 0 | True |
| Total of Shortest Distance | | 70 | Accuracy | | 48 % |

Table 4. K-Medoids Epoch-2 Test Results

| C1 | C2 | Shortest Distance | Cluster | Data Real | Values |
|---|---|---|---|---|---|
| 6 | 3 | 3 | 2 | 1 | True |
| 7 | 4 | 4 | 2 | 1 | True |
| 6 | 3 | 3 | 2 | 1 | True |
| 5 | 2 | 2 | 2 | 1 | True |
| 6 | 3 | 3 | 2 | 1 | True |
| 6 | 3 | 3 | 2 | 1 | True |
| 4 | 5 | 4 | 1 | 1 | False |
| 4 | 3 | 3 | 2 | 1 | True |
| 5 | 2 | 2 | 2 | 1 | True |
| 4 | 5 | 4 | 1 | 1 | False |
| 4 | 3 | 3 | 2 | 1 | True |
| 6 | 3 | 3 | 2 | 1 | True |
| 5 | 2 | 2 | 2 | 1 | True |
| 4 | 5 | 4 | 1 | 1 | False |
| 2 | 1 | 1 | 2 | 1 | True |
| 5 | 4 | 4 | 2 | 0 | False |
| 6 | 3 | 3 | 2 | 0 | False |
| 5 | 4 | 4 | 2 | 0 | False |
| 5 | 4 | 4 | 2 | 0 | False |
| 4 | 5 | 4 | 1 | 0 | True |
| 5 | 2 | 2 | 2 | 0 | False |
| 6 | 3 | 3 | 2 | 0 | False |
| 6 | 3 | 3 | 2 | 0 | False |
| 4 | 3 | 3 | 2 | 0 | False |
| 4 | 1 | 1 | 2 | 0 | False |
| Total of Shortest Distance | | 75 | Accuracy | | 52 % |

The simulation results of the test of the K-Medoids algorithm with quantum computing show a precision of

64% with Epoch 2. The following are the results of the test of the epoch-1 and epoch-2 data, which can be seen in Tables 5 and 6.

Table 5. K-Medoids with Quantum Computing Epoch-1 Test Results

| C1 | C2 | C1 (Dec) | C2 (Dec) | Shortest Distance | Cluster | Data Real | Values |
|---|---|---|---|---|---|---|---|
| $\begin{bmatrix}0\\0\end{bmatrix}$ | $\begin{bmatrix}4\\4\end{bmatrix}$ | 0,00 | 5,66 | 0,00 | 1 | 1 | False |
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 1,41 | 4,24 | 1,41 | 1 | 1 | False |
| $\begin{bmatrix}3\\2\end{bmatrix}$ | $\begin{bmatrix}3\\2\end{bmatrix}$ | 3,61 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\2\end{bmatrix}$ | $\begin{bmatrix}3\\2\end{bmatrix}$ | 3,61 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}4\\4\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}4\\4\end{bmatrix}$ | $\begin{bmatrix}4\\4\end{bmatrix}$ | 5,66 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}0\\0\end{bmatrix}$ | $\begin{bmatrix}4\\4\end{bmatrix}$ | 0,00 | 1,41 | 0,00 | 1 | 1 | False |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}4\\4\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 5,66 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}4\\4\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}4\\4\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 5,66 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}5\\5\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 7,07 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 1,41 | 1,41 | 1,41 | 1 | 0 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}0\\0\end{bmatrix}$ | $\begin{bmatrix}4\\4\end{bmatrix}$ | 0,00 | 1,41 | 0,00 | 1 | 0 | True |
| $\begin{bmatrix}4\\4\end{bmatrix}$ | $\begin{bmatrix}0\\0\end{bmatrix}$ | 5,66 | 1,41 | 1,41 | 2 | 0 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}4\\4\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 0 | False |
| Total of Shortest Distance | | | | 31,02 | Accuracy | | 56 % |

Table 6. K-Medoids with Quantum Computing Epoch-2 Test Results

| C1 | C2 | C1 (Dec) | C2 (Dec) | Shortest Distance | Cluster | Data Real | Values |
|---|---|---|---|---|---|---|---|
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 1,41 | 2,83 | 1,41 | 1 | 1 | False |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | 2,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\end{bmatrix}$ | 2,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}0\\0\end{bmatrix}$ | 1,41 | 1,41 | 1,41 | 1 | 1 | False |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}4\\4\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 5,66 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 1,41 | 1,41 | 1,41 | 1 | 1 | False |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}4\\4\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 5,66 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 1 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}2\\2\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 2,83 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}4\\4\end{bmatrix}$ | $\begin{bmatrix}3\\3\end{bmatrix}$ | 5,66 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}0\\0\end{bmatrix}$ | $\begin{bmatrix}1\\1\end{bmatrix}$ | 0,00 | 1,41 | 0,00 | 1 | 0 | True |
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}0\\0\end{bmatrix}$ | 1,41 | 1,41 | 1,41 | 1 | 0 | True |
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 1,41 | 1,41 | 1,41 | 1 | 0 | True |
| $\begin{bmatrix}3\\3\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 4,24 | 1,41 | 1,41 | 2 | 0 | False |
| $\begin{bmatrix}1\\1\end{bmatrix}$ | $\begin{bmatrix}2\\2\end{bmatrix}$ | 1,41 | 1,41 | 1,41 | 1 | 0 | True |
| Total of Shortest Distance | | | | 32,43 | Accuracy | | 64 % |

The findings of this research demonstrate significant advances in the K-Medoids method, facilitated by the application of quantum computing methodology. Simulation results show that the K-Medoids method achieves an accuracy rate of 64% when quantum computing is applied. Test results show an increase in accuracy of 12% compared to the classic K-Medoids method. The following compares the grouping results using the classical K-Medoids method with Manhattan distance calculations and the K-Medoids method with the quantum computing approach presented in Table 7.

Table 7. Comparison Results of 2 Methods

| Methods | Accuracy (%) | Epoch |
|---|---|---|
| K-Medoids Classic | 52 | 2 |
| K-Medoids with Quantum Computing | 64 | 2 |

## 4. Conclusions

The researchers achieved a significant breakthrough by effectively utilizing quantum computing to advance the K-Medoids data mining method. In particular, the experimental results reveal that the K-Medoids method with Manhattan distance calculation shows a better level of accuracy than the K-Medoids clustering method without quantum computing, demonstrating their proficiency in effectively grouping data in datasets. This research marks an important contribution in this field, introducing a new dimension through the use of quantum computing in the development of the K-Medoids data mining method with Manhattan distance calculations.

## Acknowledgements

We want to thank the TD Pardede Institute of Science and Technology, Medan, Indonesia, for their support until the completion of this publication.

## References

[1] S. Solikhun, S. Efendi, M. Zarlis, and P. Sihombing, “Optimizing Quantum Perceptron Architecture Using Quantum Circuits for Data Classification,” *ICIC Express Lett.*, vol. 16, no. 11, pp. 1139–1146, 2022. https://doi.org/10.24507/icicel.16.11.1139.

[2] K. Hulliyah and S. Solikhun, “Q-Madaline: Madaline Based On Qubit,” Resti, vol. 7, no. 5, pp. 1003–1008, 2023. https://doi.org/10.29207/resti.v7i5.5080.

[3] R. Buaton and S. Solikhun, “The Application of Numerical Measure Variations in K-Means Clustering for Grouping Data,” MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput., vol. 23, no. 1, pp. 103–112, 2023. https://doi.org/10.30812/matrik.v23i1.3269.

[4] T. Bezdan et al., “Hybrid fruit-fly optimization algorithm with k-means for text document clustering,” Mathematics, vol. 9, no. 16, pp. 1–19, 2021. https://doi.org/10.3390/math9161929.

[5] N. Kumar Sahu, “DBSCAN & Hierarchical Clustering Algorithm: Analysis Cyber Crime Data,” 2022. https://doi.org/2022/11.03.44.

[6] L. Zahrotun, U. Linarti, B. H. T. Suandi As, H. Kurnia, and L. Y. Sabila, “Comparison of K-Medoids Method and Analytical Hierarchy Clustering on Students’ Data Grouping,” Int. J. Informatics Vis., vol. 7, no. 2, pp. 446–454, 2023. https://doi.org/10.30630/joiv.7.2.1204.

[7] E. Schubert and P. J. Rousseeuw, Fast and eager k-medoids clustering: O(k) runtime improvement of the PAM, CLARA, and CLARANS algorithms, vol. 101, no. 124020371. 2021. https://doi.org/10.1016/j.is.2021.101804.

[8] E. L. Cahapin, B. A. Malabag, C. S. Santiago, J. L. Reyes, G. S. Legaspi, and K. L. Adrales, “Clustering of students admission data using k-means, hierarchical, and DBSCAN algorithms,” Bull. Electr. Eng. Informatics, vol. 12, no. 6, pp. 3647–3656, 2023. https://doi.org/10.11591/eei.v12i6.4849.

[9] S. A. Abbas, A. Aslam, A. U. Rehman, W. A. Abbasi, S. Arif, and S. Z. H. Kazmi, “K-Means and K-Medoids: Cluster Analysis on Birth Data Collected in City Muzaffarabad, Kashmir,” IEEE Access, vol. 8, pp. 151847–151855, 2020. https://doi.org/10.1109/ACCESS.2020.3014021.

[10] M. Tiwari, R. Kang, D. Lee, S. Thrun, and C. Piech, “BanditPAM ++: Faster k -medoids Clustering,” no. NeurIPS, 2023.

[11] R. K. Dinata, S. Retno, and N. Hasdyna, “Minimization of the Number of Iterations in K-Medoids Clustering with Purity Algorithm,” Rev. d’Intelligence Artif., vol. 35, no. 3, pp. 193–199, 2021. https://doi.org/10.18280/ria.350302.

[12] C. Oktarina, K. A. Notodiputro, and I. Indahwati, “Comparison of K-Means Clustering Method and K-Medoids on Twitter Data,” Indones. J. Stat. Its Appl., vol. 4, no. 1, pp. 189–202, 2020. https://doi.org/10.29244/ijsa.v4i1.599.

[13] I. H. Rifa, H. Pratiwi, and R. Respatiwulan, “Clustering of Earthquake Risk in Indonesia Using K-Medoids and K-Means Algorithms,” Media Stat., vol. 13, no. 2, pp. 194–205, 2020. https://doi.org/10.14710/medstat.13.2.194-205.

[14] E. Schubert and L. Lenssen, “Fast k-medoids Clustering in Rust and Python,” J. Open Source Softw., vol. 7, no. 75, p. 4183, 2022. https://doi.org/10.21105/joss.04183.

[15] J. Ren, K. Hua, and Y. Cao, “Global Optimal K-Medoids Clustering of One Million Samples,” Adv. Neural Inf. Process. Syst., vol. 35, no. NeurIPS, 2022. [16] J. Ren, K. Hua, and Y. Cao, "Applying the RFM Model with K-Means, K-Medoids, and DBSCAN Algorithms for Customer Segmentation in Transaction Data," in Advances in Neural Information Processing Systems (NeurIPS), vol. 35, no. NeurIPS, 2022.

[17] M. Tiwari, M. J. Zhang, J. Mayclin, S. Thrun, C. Piech, and I. Shomorony, “BanditPAM: Almost linear time k-medoids clustering via multi-armed bandits,” Adv. Neural Inf. Process. Syst., vol. 2020-December, no. NeurIPS 2020, pp. 1–12, 2020.

[18] R. W. Sembiring Brahmana, F. A. Mohammed, and K. Chairuang, “Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods,” Lontar Komput. J. Ilm. Teknol. Inf., vol. 11, no. 1, p. 32, 2020. https://doi.org/10.24843/lkjiti.2020.v11.i01.p04.

[19] S. Ramadhani, D. Azzahra, and T. Z, “Comparison of K-Means and K-Medoids Algorithms in Text Mining based on Davies Bouldin Index Testing for Classification of Student’s Thesis,” Digit. Zo. J. Teknol. Inf. dan Komun., vol. 13, no. 1, pp. 24–33, 2022. https://doi.org/10.31849/digitalzone.v13i1.9292.

[20] J. Kuang, G. Xu, A. Jian, H. Jatnika, and H. Jamaludin, “Application of K-Medoids Cluster Result with Particle Swarm Optimization (PSO) in Toddler Measles Immunization Cases Application of K-Medoids Cluster Result with Particle Swarm Optimization (PSO) in Toddler Measles Immunization Cases,” pp. 2–10, 2021. https://doi.org/10.1088/1742-6596/1933/1/012036.

[21] M. Faisal, E. M. Zamzami, and Sutarman, “Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance,” *J. Phys. Conf. Ser.*, vol. 1566, no. 1, 2020. https://doi.org/10.1088/1742-6596/1566/1/012112.