Jurnal Elektronik Ilmu Komputer Udayana
Volume 14, No 2. November 2025

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Twitter Sentiment Analysis Regarding the Influence of Political Figures Using the Distance – Weighted K – Nearest Neighbor Method

I Made Surya Adi Palguna[a1], Ngurah Agus Sanjaya ER[a2], I Made Widiartha[a3], Made Agung Raharja[a4]

[a]Department of Informatics, University of Udayana
Bali, Indonesia
[1]suryaadipalguna@gmail.com
[2]agus_sanjaya@unud.ac.id (Corresponding author)
[3]madewidiartha@unud.ac.id (Corresponding author)
[4]made.agung@unud.ac.id (Corresponding author)

Abstract

Social media such as Twitter has become an important platform for voicing public opinion, including in the political context. This study aims to classify public sentiment towards political figures on Twitter using the K – Nearest Neighbor (KNN) algorithm and its development, Distance – Weighted K – Nearest Neighbor (DWKNN). KNN is a nearest neighbor – based classification algorithm, but its performance is greatly influenced by the selection of the k value and does not consider the weight of the distance between neighbors. To overcome these limitations, DWKNN is applied by giving weights based on the distance of each neighbor to the test data. This study goes through the stages of data collection, data preprocessing, feature extraction, cross validation, algorithm implementation, and evaluation using three data balancing scenarios, namely baseline, oversampling, and undersampling. The evaluation results show that DWKNN provides the best performance in the baseline scenario with an accuracy of 68%, precision 52%, recall 47%, and F1 – score 48%, compared to KNN with an accuracy of 66%, precision 47%, recall 45%, and F1 – score 45%. These findings indicate that DWKNN is more effective in classifying public sentiment towards political figures than KNN.

Keywords: *Sentiment Analysis, Twitter, Political Figures, Distance – Weighted K – Nearest Neighbor, K – Nearest Neighbor*

1. Introduction

Elections are a democratic process where citizens (people) participate in decision – making (electing the president to regional officials) to fill political positions. Social media as a digital platform plays an important role in elections by allowing candidates to spread campaign messages, publish programs, and interact directly with voters, so they can access information about candidates and political parties and increase community involvement. [1] Social media platforms such as Twitter are a place to explore public opinion regarding political figures and related issues, discussions, and political sentiments.

One of the methods used for sentiment analysis is KNN. K – Nearest Neighbor (KNN) is one of the most well – known supervised learning algorithms in pattern classification that works by storing all training data and determining the class of new data (query) based on the majority of labels from a number of k closest neighbors in the feature space. [2] Several previous studies used this algorithm with the highest accuracy of 67.2% to 96% at values of k = 1, k = 5, k = 7 and k = 9, the highest precision of 56.94% to 85% at values of k = 5, the highest recall of 69.5% to 81% at values of k = 15, the highest F – score of 83%, and the highest AUC of 0.916. [3] [4] [5] [6] [7] [8] [9] [10] However, KNN is highly dependent on the selected k value where if the data is small or k is not appropriate, performance can decrease because it is too easily influenced by noisy, ambiguous, or outlier data. In addition, KNN uses majority voting without considering how close the neighbors are where closer neighbors are more relevant to determining the class, so the classification results can be less accurate if the distance between neighbors varies greatly. [2]

In order to overcome this problem, several weighted voting methods have been developed for KNN, one of which is using DWKNN. Distance – Weighted K – Nearest Neighbors (DWKNN) is a development of the KNN algorithm that uses weights based on distance to determine the class of test data (query). (Gou et al., 2012) Several previous studies have used this algorithm with the highest accuracy of 83.3% to 98.36% at k = 1, k = 3, and k = 9, the highest precision of 98.77%, the highest recall of 99.35%, and AUC of 99.03%. [11] [7] [12] This study aims to determine the optimal k value along with the results of the accuracy, precision, recall, and F1 – score of the Distance – Weighted K – Nearest Neighbor (DWKNN) algorithm with the K – Nearest Neighbor (KNN) algorithm in sentiment classification. It is expected that the results of this study practically, the results of this study can present the results of sentiment classification, namely positive, neutral, or negative. In addition, theoretically, this study can add to the study of the application of the Distance – Weighted K – Nearest Neighbor (DWKNN) algorithm in sentiment classification.

2. Reseach Methods

In this research methodology, the steps in this research will be explained. The description of the method used in this research is divided into several stages, namely (1) Data Collection, (2) Data Preprocessing, (3) Feature Extraction, (4) Cross Validation, (5) Implementation of KNN and DWKNN Algorithms, and (6) Testing and Evaluation.

2.1. Data Collection

In this sentiment analysis study, the type of data used in this study is textual data in the form of tweets whose data source in this study comes from the Twitter platform and is secondary data. Data is taken using the Twitter API using Tweet Harvest. Data is taken based on the dates January 22, 2024 to February 10, 2024 and is in Indonesian. The collected tweets are filtered based on the keywords "anies", "prabowo", "ganjar", "muhaimin", "gibran", and "mahfud". After the data is collected, each tweet is manually labeled with positive, negative, or neutral sentiment. The labeled data is then divided into training data and testing data with a ratio of 80:20.

2.2. Data Preprocessing

Data preprocessing on text is a process before doing text mining with the aim of obtaining the main features or main terms from text documents and to increase the relevance between words and documents as well as the relevance between words and categories. Preprocessing also functions to clean the collected text from noise, such as sorting which words are important for classification, removing stopwords, and so on. [3] Data preprocessing consists of (1) Cleaning Data, (2) Case Folding, (3) Normalization, (4) Tokenizing, (5) Stopword Removal, (6) Stemming, (7) Remove Outliers, and (8) Label Encoding.

2.3. Feature Extraction (TF – IDF)

Feature extraction with the TF – IDF (Term Frequency – Inverse Document Frequency) method in this study aims to convert preprocessed text data into a numeric form that represents the level of importance of each word in the document. The input of this process is text data that has gone through the preprocessing stage and is ready to be converted into a feature vector. The process begins by calculating the term frequency (TF) using the logarithm of the number of occurrences of the word in the document, then continues by calculating the inverse document frequency (IDF) based on the number of documents containing the word. The TF value is then multiplied by the IDF to obtain the final TF – IDF score which reflects the importance of a word in the context of the entire corpus. The output of this process is a numeric data in the form of a matrix, where each row represents a document and each column represents words that have been extracted as features, which are then used in the classification model training process.

2.4. Cross Validation (K – Fold Cross Validation)

Cross Validation with the K – Fold method is used in this study to evaluate model performance and reduce the risk of overfitting. The input of this process is in the form of feature data (X), labels (y), the classification model to be tested, and the k value that determines the number of folds in the validation process. In K – Fold Cross Validation, the data is divided into k parts that are more or less balanced where in each iteration, one part is used as test data and the rest as training data, until all parts have

p-ISSN: 2301-5373
e-ISSN: 2654-5101

been used as test data alternately. The output of this process is a collection of evaluation scores (accuracy) from each fold, which can be averaged.

2.5. Implementation of the KNN and DWKNN Algorithm

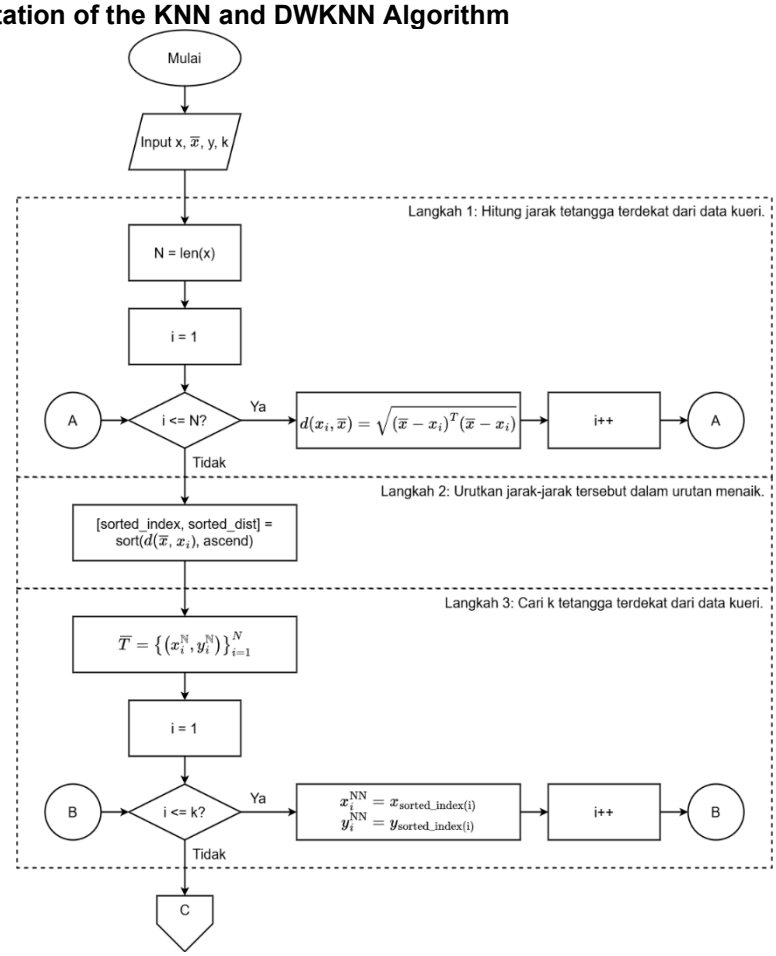

Figure 1. The process of calculating distances, sorting distances, and searching for the k nearest neighbors from query data of the KNN and DWKNN algorithms.

In the implementation of the K – Nearest Neighbors (KNN) algorithm as in Figure 1, and Figure 2, the input of this algorithm is in the form of test data (query) and a set of training data that has been represented in the form of a numeric vector. The process begins by calculating the distance between the test data and all training data using the Euclidean formula. After that, the distances are sorted in ascending order to obtain the k nearest neighbors. The output of this process is a class label selected through a class voting mechanism, where the label with the largest number of classes from the k nearest neighbors will be the final prediction result for the test data.

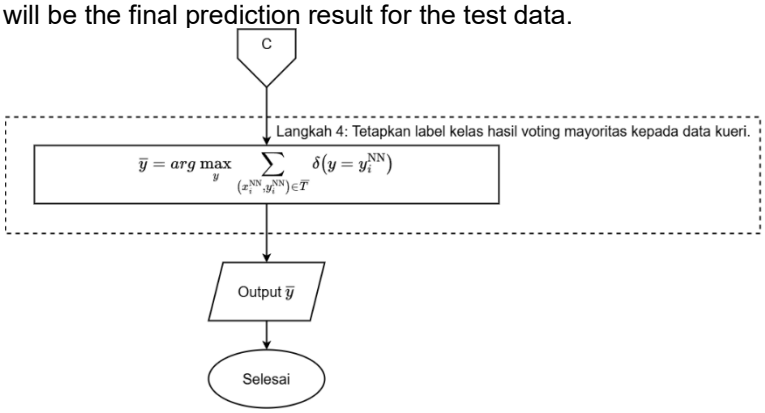

Figure 2. The process of predicting the final results of the KNN algorithm

In the implementation of the Distance Weighted K – Nearest Neighbors (DWKNN) algorithm as in Figure 1 and Figure 3, the input of this algorithm is in the form of test data (query) and a set of training

data that has been represented in the form of a numeric vector. The process begins by calculating the distance between the test data and all training data using the Euclidean formula. After that, the distances are sorted in ascending order to obtain the k nearest neighbors. Furthermore, each neighbor is given a weight based on a dual weighting function, which calculates the relative influence of each neighbor on the test data based on its distance position compared to the nearest and farthest neighbors. The output of this process is a class label selected through a weighted voting mechanism, where the label with the highest total weight from the k nearest neighbors will be the final prediction result for the test data.

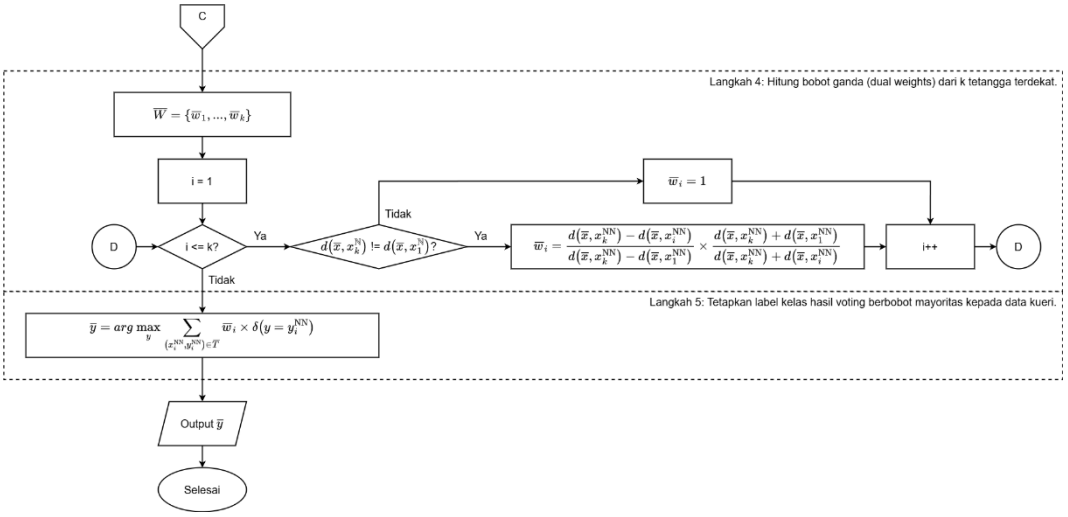

Figure 3. The process of predicting the final results of the DWKNN algorithm

2.6. Testing and Evaluation

To evaluate the performance of the classification methods used in this study, a series of tests were conducted by comparing the K – Nearest Neighbors (KNN) and Distance – Weighted KNN (DWKNN) algorithms at various k values, namely from k = 5 to k = 50 with an interval of 5. In addition, the effect of data balancing techniques was also analyzed using three scenarios, namely baseline (without balancing), oversampling, and undersampling. Each combination of classification methods, k values, and balancing scenarios was evaluated using four main performance metrics, namely accuracy, precision, recall, and F1 – score. This evaluation design aims to comprehensively understand how different algorithms and data imbalance handling strategies affect classification performance in the context of sentiment analysis in this study.

3. Result and Discussion

In Figure 4, we can see the test results by comparing the K – Nearest Neighbors (KNN) and Distance – Weighted KNN (DWKNN) algorithms at various k values, namely from k = 5 to k = 50 with three scenarios, namely baseline (without balancing), oversampling, and undersampling.

Table 1. Comparison results of KNN and DWKNN accuracy from k = 5 to k = 50 for baseline, oversampling, and undersampling

K	KNN			DWKNN		
	Baseline	Over Sampling	Under Sampling	Baseline	Over Sampling	Under Sampling
5	0,661506	0,512231	0,406433	0,636412	0,511642	0,380117
10	0,651895	0,493958	0,400585	0,66204	0,514589	0,397661
15	0,642819	0,473033	0,371345	0,678057	0,518126	0,391813
20	0,624666	0,465665	0,380117	0,684997	0,519894	0,391813
25	0,628938	0,45977	0,359649	0,682328	0,525494	0,397661
30	0,635878	0,453581	0,380117	0,681794	0,524609	0,394737
35	0,625734	0,451223	0,359649	0,682862	0,522841	0,394737
40	0,639082	0,444444	0,356725	0,682328	0,520778	0,394737
45	0,638548	0,440908	0,371345	0,680192	0,522546	0,406433
50	0,636946	0,43855	0,359649	0,680192	0,524315	0,406433

p-ISSN: 2301-5373
e-ISSN: 2654-5101

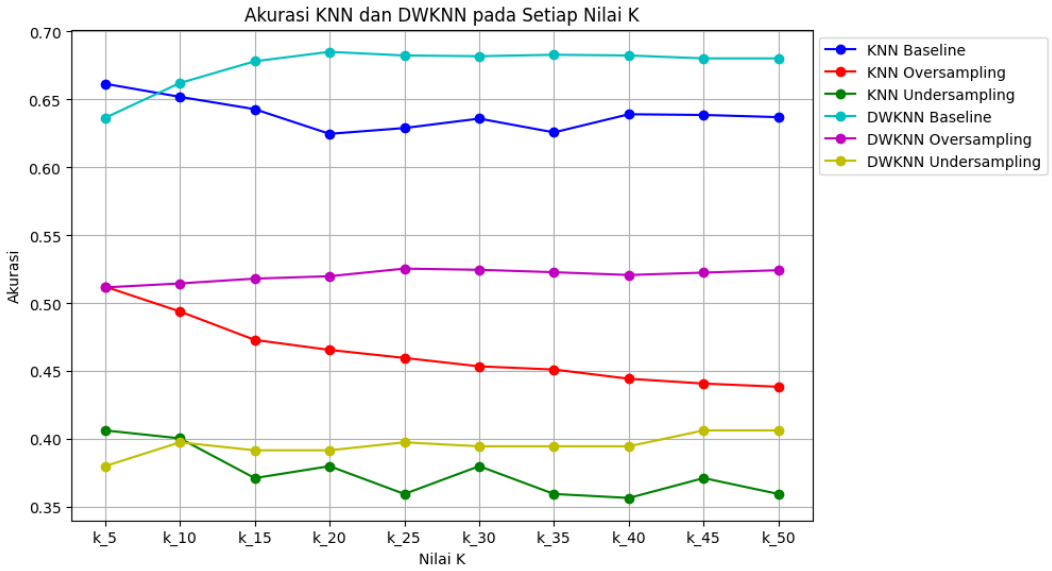

Figure 4. Comparison results of KNN and DWKNN accuracy from k = 5 to k = 50 for baseline, oversampling, and undersampling

Based on the evaluation results of the performance of the KNN and DWKNN algorithms in various data balancing scenarios (baseline, oversampling, and undersampling) obtained based on Table 1, it was found that DWKNN consistently showed higher accuracy than KNN for each k value tested, namely from k = 5 to k = 50 with an interval of 5. This indicates that giving weights based on distance in the DWKNN algorithm is able to increase the sensitivity of the model to the proximity of features in the vector space, thus providing a more accurate classification. In addition, when viewed from the data balancing scenario, the baseline method that represents the condition of an unbalanced class distribution produces higher accuracy compared to the oversampling and undersampling approaches in both algorithms. This can be caused by overfitting in oversampling due to duplication of minority data, as well as the loss of important information in undersampling due to reduction of majority data.

Based on the performance comparison of all combinations, the best results are obtained for each scenario where for the baseline scenario the DWKNN algorithm with the evaluation results in Table 5 obtained the highest accuracy of 68% at k = 20 while the KNN algorithm with the evaluation results in Table 2 obtained the best performance with an accuracy of 66% at k = 5. Then in the oversampling scenario the best results were achieved by DWKNN with the evaluation results in Table 6 with an accuracy of 53% at k = 25 while KNN with the evaluation results in Table 3 obtained the highest accuracy of 51% at k = 5. And also for the undersampling scenario DWKNN with the evaluation results in Table 7 recorded the best performance with an accuracy of 41% at k = 45, while KNN with the evaluation results in Table 4 obtained the highest results in the form of an accuracy of 41% at k = 5.

Table 2. KNN sentiment evaluation results with baseline with best k

Metrics	Negative	Neutral	Positive	Average
Accuracy				0,66
Precision	0,11	0,70	0,61	0,47
Recall	0,04	0,81	0,51	0,45
F1 – score	0,05	0,75	0,56	0,45

Table 3. KNN sentiment evaluation results with oversampling with best k

Metrics	Negative	Neutral	Positive	Average
Accuracy				0,51
Precision	0,55	0,47	0,51	0,51
Recall	0,46	0,35	0,72	0,51
F1 – score	0,50	0,40	0,60	0,50

Table 4. KNN sentiment evaluation results with undersampling with best k

Metrics	Negative	Neutral	Positive	Average
Accuracy				0,41
Precision	0,47	0,33	0,60	0,47
Recall	0,45	0,54	0,23	0,41
F1 – score	0,46	0,41	0,33	0,40

Table 5. DWKNN sentiment evaluation results with baseline with best k

Metrics	Negative	Neutral	Positive	Average
Accuracy				0,68
Precision	0,21	0,71	0,64	0,52
Recall	0,05	0,83	0,54	0,47
F1 – score	0,08	0,77	0,59	0,48

Table 6. DWKNN sentiment evaluation results with oversampling with best k

Metrics	Negative	Neutral	Positive	Average
Accuracy				0,53
Precision	0,57	0,52	0,51	0,53
Recall	0,39	0,38	0,80	0,53
F1 – score	0,47	0,44	0,62	0,51

Table 7. DWKNN sentiment evaluation results with undersampling with best k

Metrics	Negative	Neutral	Positive	Average
Accuracy				0,41
Precision	0,61	0,34	0,65	0,53
Recall	0,19	0,80	0,23	0,41
F1 – score	0,29	0,48	0,34	0,37

4. Conclusion

Based on the evaluation results that have been carried out on the KNN and DWKNN algorithms on the sentiment classification task with three data balancing scenarios (baseline, oversampling, and undersampling), the optimal k value is obtained which provides the best performance for each algorithm and scenario. The DWKNN algorithm shows the best performance in the baseline scenario with a value of k = 20, resulting in an accuracy of 68%, precision of 52%, recall of 47%, and F1 – score of 48%. Meanwhile, the KNN algorithm provides the best results in the baseline scenario with a value of k = 5, with an accuracy of 66%, precision of 47%, recall of 45%, and F1 – score of 45%. From these results, it can be concluded that the DWKNN algorithm not only produces higher accuracy, but also provides better average precision, recall, and F1 – score values than KNN.

References

[1] V. Friskila Angela, “ANALISIS PERAN MEDIA SOSIAL DALAM PENGARUH POLITIK MENJELANG PEMILU,” *Jurnal Ilmu Sosial dan Ilmu Politik Interdisiplin*, vol. 10, no. 1, pp. 555–564, Jun. 2023.

[2] J. Gou, L. Du, Y. Zhang, and T. Xiong, “A New Distance-weighted k-nearest Neighbor Classifier,” *Journal of Information & Computational Science*, vol. 9, no. 6, pp. 1429–1436, 2012.

[3] M. S. Alrajak, I. Ernawati, and I. Nurlaili, “ANALISIS SENTIMEN TERHADAP PELAYANAN PT PLN DI JAKARTA PADA TWITTER DENGAN ALGORITMA K-NEAREST NEIGHBOR (K-NN),” *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, vol. 1, no. 2, pp. 110–122, Aug. 2020.

[4] A. Asro’i and H. Februariyanti, “Analisis Sentimen Pengguna Twitter terhadap Perpanjangan PPKM Menggunakan Metode K-Nearest Neighbor,” *JURNAL KHATULISTIWA INFORMATIKA*, vol. 10, no. 1, pp. 17–24, Jun. 2022.

[5] A. Deviyanto and M. D. R. Wahyudi, “PENERAPAN ANALISIS SENTIMEN PADA PENGGUNA TWITTER MENGGUNAKAN METODE K-NEAREST NEIGHBOR,” *JISKa (Jurnal Informatika Sunan Kalijaga)*, vol. 3, no. 1, pp. 1–13, May 2018.
[6] S. Ernawati and R. Wati, “Penerapan Algoritma K-Nearest Neighbors Pada Analisis Sentimen Review Agen Travel,” *JURNAL KHATULISTIWA INFORMATIKA*, vol. 6, no. 1, pp. 64–69, Jun. 2018.
[7] W. Hardianti, F. Indriani, and R. Adi Nugroho, “ANALISIS PERBANDINGAN ALGORITMA DISTANCE-WEIGHTED KNN DAN ALGORITMA KNN PADA PREDIKSI MASA STUDI MAHASISWA,” *Seminar Nasional Ilmu Komputer (SOLITER)*, vol. 1, pp. 108–117, Oct. 2017.
[8] M. R. Huq, A. Ali, and A. Rahman, “Sentiment Analysis on Twitter Data using KNN and SVM,” *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 19–25, 2017.
[9] M. R. Irfan, M. A. Fauzi, Tibyani, and N. D. Mentari, “Twitter Sentiment Analysis on 2013 Curriculum Using Ensemble Features and K-Nearest Neighbor,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 5409–5414, Dec. 2018, doi: 10.11591/ijece.v8i6.pp5409-5414.
[10] R. Sari, “Analisis Sentimen Pada Review Objek Wisata Dunia Fantasi menggunakan Algoritma K-Nearest Neighbor (K-NN),” *Evolusi: Jurnal Sains dan Manajemen*, vol. 8, no. 1, pp. 10–17, Mar. 2020.
[11] A. R. Chrismanto, Y. Lukito, and A. Susilo, “Implementasi Distance Weighted K-Nearest Neighbor Untuk Klasifikasi Spam & Non-Spam Pada Komentar Instagram,” *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 6, no. 2, pp. 236–244, Aug. 2020, doi: 10.26418/jp.v6i2.39996.
[12] Y. Kurniawan Mangalik, T. Hamonangan Saragih, D. Turianto Nugrahadi, Muliadi, and M. Itqan Mazdadi, “ANALISIS SELEKSI FITUR BINARY PARTICLE SWARM OPTIMIZATION PADA KLASIFIKASI KANKER BERDASARKAN DATA MICROARRAY MENGGUNAKAN DISTANCE WEIGHTED K-NEAREST NEIGHBORS,” *JIP (Jurnal Informatika Polinema)*, vol. 9, no. 2, pp. 143–152, Feb. 2023.

This page is intentionally left blank.