 ISSN:2302-8149
e-ISSN:2540-9719

Data Balancing Approach Using Combine Sampling on Sentiment Analysis With K-Nearest Neighbor

[1]Evlyn Pricilia Kondy, [2]Siswanto Siswanto*, [3]Nirwan Ilyas
[1,2,3]Statistics, Fakultas of Mathematics and Natural Science, Hasanuddin University
[1,2,3]Perintis Kemerdekaan KM 10, Tamalanrea Indah, Tamalanrea, Makassar City, South Sulawesi, Indonesia 90245
*e-mail: siswanto@unhas.ac.id

(***received*:** 13 March 2024, ***revised*:** 24 June 2024, ***accepted*:** 21 July 2024)

Abstract

One of the topics that has been discussed on twitter is the rules regarding the removal of masks. However, there's a chance that the data from Twitter contains unequal data classes. An unequal amount of data can cause the classification process to malfunction. Combining under- and oversampling techniques is known as combined sampling, and it is a data-balancing strategy. The research's data consists of Indonesian tweets using the hashtag "The Policy of Removing Masks." In this study, the classification approach was K-Nearest Neighbor, while the oversampling and undersampling techniques were SMOTE and Tomek Links. The purpose of this research is to classify sentiment using the K-Nearest Neighbor algorithm and to use combined sampling to balance the amount of training data in the two classes that are not yet balanced. 234 training data with a positive sentiment and 652 training data with a negative sentiment were obtained after the data was divided. Due to an imbalance in the quantity of training data between the two classes, the positive class's data is minor and the negative class's data is major. The quantity of training data is 613 in the positive class and 613 in the negative class obtained following the combine sampling. Following the balancing of data between the two classes, sentiment classification was performed, yielding an accuracy of 60.4%, precision of 78.5%, and recall of 65%. The reason for the accuracy number of 60.4% is that machine learning misinterpreted a tweet regarding Indonesia's mask removal policy, leading to incorrect classification.

Keywords: combine sampling, smote, tomek links, k-nearest neighbor, the policy of removing masks

1 Introduction

The advance of internet technology makes it easier for people to access news and express their thoughts. Twitter is one of the social media platforms where users can post brief messages with comments, news, and opinions [1]. Twitter users can tweet about their opinions. As more and more tweets are made about a subject, it may eventually become trending on twitter. The policy regarding the removal of masks is one of the trending topics that was debated on twitter. The pros and cons of this approach are discussed throughout society. Sentiment analysis is required to process and examine opinions regarding the mask removal policy in text form. Sentiment analysis uses sentiment results to categorize texts as positive, neutral, or negative [2]. In sentiment analysis, numerous classification algorithms are used, such as K-Nearest Neighbor (KNN), decision tree, naive bayes, support vector machine (SVM), and others [3], [4].

The KNN algorithm is a supervised algorithm that employs one of the most basic and straightforward classification technique of any machine learning algorithm [5]. This algorithm performs well in sentiment analysis in Indonesian and can handle enormous amounts of data [6]. However, this technique is significantly reliant on the value selection *k*. When the parameter value *k* is too small, conditions overfitting occurs, and when the parameter value *k* is too large, too many points from other classes are included [5]. One method for improving the performance of the KNN algorithm is to balance the quantity of data in the obtained classes [7]. Data from twitter may have an unequal amount of data in classes. Unbalanced data classes are classified as major data classes or minor data classes [8]. The major data class has been labeled with more items than the minor data class [9]. Because the data tends to support the majority class, data imbalance can lead to classification errors

 ISSN:2302-8149
e-ISSN:2540-9719

[10]. Combine sampling method can be used to overcome data imbalance. Combine sampling is a method that combines oversampling with SMOTE and undersampling using Tomek Links [11].

SMOTE method works by replicating data in the minor class until the data in both classes are balanced [12]. SMOTE can overcome the overfitting problem caused by technique oversampling [13]. Besides from that, SMOTE can improve prediction accuracy for minority groups Next, Tomek Links will apply data removal to remove noise that may interfere with the classification process [14]. The goal of this research is to balance the amount of training data on the sentiment of the mask-free policy in Indonesia with implementation combine sampling and classify the sentiment of the mask removal policy in Indonesia using the KNN algorithm, and then measure the KNN algorithm's performance percentage.

2 Literature Review

K-Nearest Neighbor is one of the classification techniques in machine learning that groups new data based on *k*. Several studies comparing the performance of KNN with other classification methods are research by [15] compares algorithm support vector machine (SVM) and KNN to classify sentiment regarding vaccine Sinovac obtained through twitter. The SVM method has a performance accuracy value of 70%, while the KNN algorithm has a value of 56%. The accuracy of the KNN method is lower than that of the SVM algorithm. Other research by [16] compares algorithm multinomial naïve bayes and KNN for classifying review movies in Gujarati using TF-IDF feature extraction and count vectorizer. The algorithm multinomial naive bayes with TF-IDF feature extraction produced the best results (accuracy, precision, recall, and F1-score).

The two research above show that the KNN algorithm performs less than the SVM algorithm and multinomial naive bayes. The data balancing stage can be performed during the data preprocessing stage to increase the performance of the KNN algorithm. Combine sampling is a data balancing strategy that combines oversampling and undersampling approaches. Previous studies used the combined sampling method to solve data imbalance are research by [17] uses a combination of SMOTE and Tomek Links on the algorithm random forest to carry out classification on the diabetes dataset obtained through Kaggle. Random forest algorithm performance improved to 86,4% using a combination of SMOTE and Tomek Links, compared to 81,9% with SMOTE alone and 76% without data balancing. Then research by [18] compares the performance of the C5.0 algorithm without, with SMOTE, and with a combination of SMOTE and Tomek Links. When applying a combination of SMOTE and Tomek Links, the accuracy decreases from 0,9991 to 0,9775 as compared to using only the C5.0 method. The value sensitivity also decreases, from 0,9 when using the C5.0 algorithm to 0,06088 when applying a combination of SMOTE, Tomek Links, and the C5.0 algorithm. However, in terms of value specificity, the best result was obtained by combining SMOTE and Tomek Links with the C5.0 algorithm, which was 0,99973. Research by [19] compared the use of SVM without and with a combination of SMOTE and Tomek Links in the review KitaBisa application. After using a combination of SMOTE and Tomek Links, accuracy went from 72% to 98%, precision increased from 76% to 98%, recall increased from 72% to 98%, and F1-score ascended from 64% to 98%.

3 Research Method

Data for this research was collected from Twitter using the keyword mask removal policy between May 1, 2022, and October 31, 2022. 12.000 Indonesian tweets were used as data. The following are the steps of this research:

3.1 Crawling Data Twitter

Crawling data is a program or script that collects content or information from a website [20], [21]. Crawling techniques can be used to extract data from social media. Twitter crawling is an implementation of the crawling technique to acquire information from social media platforms such as twitter [22]. Twitter has developed an Application Programming Interface (API) for obtaining and processing data. The Application Programming Interface (API) allows computer applications to communicate and exchange information. Twitter API to make it easier for other developers to access the information contained on twitter [23]. The information obtained is a keyword-adaptable. Account information can be retrieved by crawling Twitter, including mentions, retweets, favorites, follows, and friends [24].

 ISSN:2302-8149
e-ISSN:2540-9719

3.2 Preprocessing Data

Text data preprocessing is the first step in transforming a text document into structured data, which can be processed further in the text mining process [25]. Preprocessing text data involves five phases, notably [26]:

1. Data Cleansing
   Data cleansing is a method of cleaning data or characters that can interfere with data processing, such as hashtag, username, URL, and punctuation, so that the results produced are simply the letters "a" to "z" [4].
2. Case Folding
   Case folding is the process of transforming every character to lowercase. Case folding avoids sensitive cases in preprocessing text data [27].
3. *Tokenizing*
   This stage will involve tokenizing or splitting the substance of a sentence into separate words (terms) to create a token that can be evaluated [25].
4. Stopword Removal
   The technique tries to delete words that do not contribute to text classification or convey a relevant message in the text or phrase [25].
5. Stemming
   Stemming is a rule-based process in IR systems that converts words in a document to their basic form [27]. Method stemming tries to delete prefixes, suffixes, insertions, and confixes (prefix-suffix combinations) [25].

3.3 Term Weighting Using Term Frequency-Inverse Document Frequency

Term weighting is the process of assigning weight (term) to words in a text document that will be processed. The phases in word weighting with TF-IDF are as follows [28]:

1. Term Frequency (TF) refers to the frequency with which a word appears in a document..
2. Document Frequency (DF) is the number of documents containing specific words.
3. Inverse Document Frequency (IDF) refers to the inverse frequency with which particular words (terms) occur in a written document. The IDF equation is as follows [28]:

$$idf_t = \ln\left(\frac{N}{df_{(t)}}\right) + 1 \quad (1)$$

   Equation (1) is used to calculate the IDF value for each term where N is the number of text documents, and $df_{(t)}$ is the number of documents that contain term t.
4. Term Frequency-Inverse Document (TF-IDF) is a multiplication of term frequency (TF) and inverse document frequency (IDF) that calculates weights and determines the importance of a term from a document. When a term appears frequently in a document, the weight value will be bigger; nevertheless, if the term appears frequently in multiple documents, it will have a lower weight value. The equation for TF-IDF is provided below [28]:

$$Tf.idf_{t,d} = tf_{td} \times idf_{(t)} \quad (2)$$

   Equation (2) is used to calculate the TF-IDF value for each term in the document where tf_{td} is the term frequency of t in document d and $idf_{(t)}$ is the inverse document frequency of word t.

3.4 Split Data into Training and Test Data

Data splitting is the partition of data into two or more subsets. Data splitting plays an important role in machine learning, particularly for developing data models [29]. When two data sets are divided, training and test data are typically generated [30]. Training data is used as a reference while developing a classification model. The test data refers to data used to evaluate the classification model's performance [31]. There is no standard criterion for determining the appropriate ratio of training data to test data, however, multiple empirical analyses demonstrate that using 70-80% training data and 20-30% test data yields the best results [32]. As a result, this study will use a 70:30 ratio for training and test data.

 ISSN:2302-8149
e-ISSN:2540-9719

3.5 Data Balancing With Combine Sampling

Combine sampling is a method that uses oversampling and undersampling to balance the amount of data in both classes [33]. There are numerous ways for oversampling, including SMOTE (Synthetic Minority Oversampling Technique), random oversampling, adaptive synthetic (ADASYN), and others. The SMOTE approach is one method for overcoming overfitting caused by oversampling [34]. SMOTE can also improve prediction accuracy for minority classes [35]. SMOTE replicates data on a minority class basis, with k nearest neighbors. The equation for generating fresh synthetic data is as follows. [36]:

$$x_{syn(ij)} = x_{ij} + \left(x_{knn(ij)} - x_{ij}\right) \times w \quad (3)$$

Equation (3) is used to calculate the synthetic data to ith data term-j that will be created where x_{ij} is ith data term-j that will be replicated, $x_{knn(ij)}$ is the ith data term-j with the shortest distance from x_{ij}, and w is a random number between 0 and 1.

After SMOTE generates synthetic data, the next step is to check the data for noise on the border using the technique of undersampling [14]. Data noise is data that is misclassified and must be deleted since it can interfere with the classification process [37]. There are several ways for undersampling, including the Neighborhood Cleaning Rule (NCL), Tomek Links, random undersampling, edited nearest neighbors, and so on [38]. Tomek Links was employed as the method of undersampling in this research. Tomek Links is a technique for undersampling that acts as a cleaning mechanism, removing samples from two different classes. Tomek Links are defined as follows [33]:

1. Two samples, x and y, belong to separate classes, and $d(x, y)$ is their distance.
2. (x, y) is called Tomek Links with no sample z until $d(x, z) < d(x, y)$ or $d(y, z) < d(y, x)$.

3.6 Classification Using K-Nearest Neighbor Algorithm

K-nearest neighbor (KNN) is a classification technique that uses k-training data to determine the object's closest distance [39]. This algorithm calculates the distance between items in the test and training data [40]. The KNN algorithm is a nonparametric machine learning technique. Because it is nonparametric, the resulting class determination line is adaptable and nonlinear [41]. Aside from that, this algorithm has self-learning capabilities and can adapt to fresh training data [42]. The Euclidean distance approach is used in this research. Distance Euclidean is used to quantify the distance between two objects represented by a straight line in Euclidean space. The Euclidean distance method finds the shortest distance between two places (straight distance). Euclidean distance equation is as follows [43]:

$$d(x_k, y_k) = \sqrt{\sum_{i=1}^{n} (x_{ik} - y_{ik})^2} \quad (4)$$

Equation (4) is used to calculate the Euclidean distance between the kth training data and kth testing data where x_{ki} is the point of training data in term i document k and y_{ik} is the point of testing data in term i document k.

The phases of data classification with the KNN algorithm are as follows [44]:

1. Determine the distance between test and training data using the Euclidean distance approach.
2. Sort the distance between test and training data by the smallest value.
3. Determines the parameter value k, where k is the number of nearest neighbors in the test data.
4. Label the test data based on the majority label of its nearest neighbors.

The KNN algorithm also depends on choosing the number of parameters *k*. *K* represents the number of nearest neighbors from the test data. When *k* is 1, the classification results are very rigid because it only takes into account one nearest neighbor. However, if *k* is too many, the classification results are vague, so parameter values are needed in k appropriate method to obtain classification with a good level of accuracy [45]. As a result, this study will use values *k* ranging from 2 to 10, and the score *k* with the highest accuracy value will be used as a parameter for categorizing mask-free policy sentiments in Indonesia.

 ISSN:2302-8149
e-ISSN:2540-9719

3.7 Classification Algorithm Perfomance Percentage Test

Measuring classification system performance is critical [46]. Classification system performance refers to how successfully the system classifies data [47]. The performance of a classification algorithm can be tested using a confusion matrix. A confusion matrix is a way to calculate accuracy, precision, and sensitivity (recall) in process data mining. To calculate the classification algorithm's performance, there are four components in the confusion matrix, namely [48]:

1. True positive (TP) is the amount of positive data that is correctly classified as positive data..
2. True Negative (TN) is the number of negative data that are correctly classified as negative data.
3. False positive (FP) is the number of positive data but incorrectly classified as negative data.
4. False negative (FN) is the number of negative data but incorrectly classified as positive data.

The confusion matrix can be used to measure the classification performance of a model [23]. The following three measurements of the classification algorithm's performance are as follows [49]:

1) Accuracy
Accuracy is used to calculate the accuracy of the classification of a document that has data balanced in each category. **Equation (5)** is used for calculating accuracy as follows [49]:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (5)$$

2) Precision
Precision is the ratio of the number of data that are predicted to be truly positive out of all data that is classified as positive data. **Equation (6)** for calculating precision as follows [49]:

$$precision = \frac{TP}{TP + FP} \times 100\% \quad (6)$$

3) Recall
Recall is used to see the accuracy between the positive data class produced by the system and the actual class. **Equation (7)** for calculating precision as follows [49]:

$$recall = \frac{TP}{TP + FN} \times 100\% \quad (7)$$

4 Results and Analysis

Results and discussion on the stages of balancing training data using combine sampling, as well as the classification of mask-removing policy sentiment in Indonesia using the KNN algorithm, are as follows:

4.1 *Crawling* Data *Twitter*

The data in this study was gathered through results crawling Twitter with the keyword policy of eliminating masks from 1 May 2022 to 31 October 2022, with up to 12,000 tweets speaking Indonesian. **Table 1** displays the outcomes of data collecting.

Table 1. Twitter crawl data

No	Publication Date	Username	Tweet
1	5/15/2022 23:57	@HukumDan	@KRMTRoySuryo2 @elonmusk @jokowi Di negara sendiri pake masker, di negeri orang lepas masker 😁
2	5/16/2022 0:12	@winterouz	@jmj_cross kalo disini pocongnya udah new normal sih, jd pada lepas masker
3	5/16/2022 0:33	@MatchWonBy	@badutminton Senyum Greysia tak pernah lepas dari bibirnya, sekalipun harus tertutup oleh masker. Tatap matanya berbinar, melihat juniornya berjuang dengan begitu semangatnya hingga akhir laga. Secercah asa bangkitnya srikandi merah putih mulai tumbuh di benaknya.

 ISSN:2302-8149
e-ISSN:2540-9719

No	Publication Date	Username	Tweet
			"terima kasih, adik-adikku,"
⋮	⋮	⋮	⋮
12000	10/31/2022 23:14	@i_m_henpunzel	@BY0100 Klo pake masker GUANTENG POL pas lepas oh yaudin b aja wkwkwkkwkw

Table 1 is the result of data crawling twitter consisting of 12,000 tweets. The next stage is manual labeling. Manual labeling is the process of labeling all collected data based on its category. There are two categories in this research, namely the positive class and the negative class. Based on manual labeling results, the amount of data labeled was 1,267 data. **Figure 1** shows the bar chart sentiment class tweet mask removal policy.

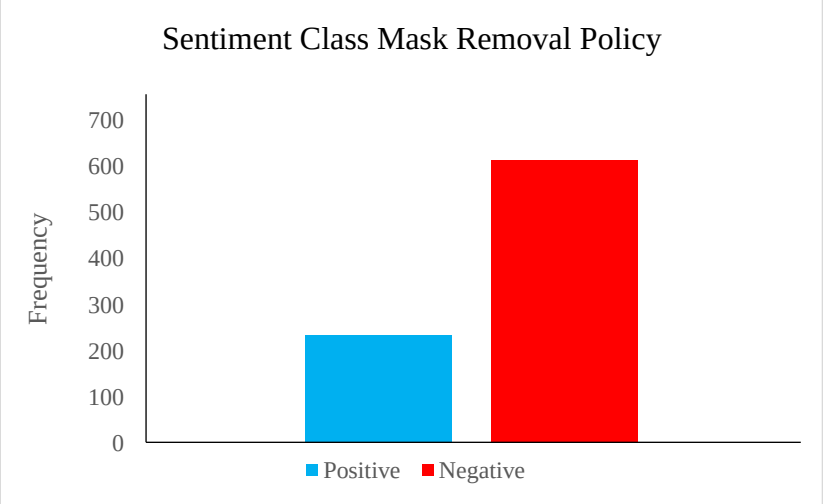

Figure 1. Bar chart sentiment class data mask removal policy

Figure 1 represents the amount of data categorized as positive and negative. Because the negative class contains more data than the positive class, the negative class is referred to as major data and the positive class as minor data. Because the amount of data in the two classes is not balanced, it is necessary to combine sampling at the data preprocessing stage.

4.2 Text Data Preprocessing

Text data preprocessing is the first step in transforming a text document into structured data, which can then be processed further in the text mining process. Text data preparation has five phases, which are as follows:

1. Data Cleansing
   Data cleansing is the process of removing data or characters that may interfere with data processing, such as URLs, usernames, hashtags, numerals, and punctuation marks and leaving only the letters a-z. **Table 2** shows a data structure before and after the method of data cleansing.

Table 2. Data structure before and after processing data cleansing

No	Text Before Data Cleansing Process	Text After Data Cleansing Process
1	@txtdrjkt udah saatnya kita lepas masker	udah saatnya kita lepas masker
2	@Eka4Liong Sampai skrg belum berani ke mall disaat weekend . Belum berani lepas masker. Belum berani ke tempat keramaian .	Sampai skrg belum berani ke mall disaat weekend Belum berani lepas masker Belum berani ke tempat keramaian
3	Dimanapun, kapanpun kamu berada, selalu gunakan masker! Jangan lepas maskermu. Dan selalu patuhi	Dimanapun kapanpun kamu berada selalu gunakan masker Jangan lepas maskermu Dan selalu patuhi di setiap

 ISSN:2302-8149
e-ISSN:2540-9719

No	Text Before Data Cleansing Process	Text After Data Cleansing Process
	#ProkesPutusRantaiKovid di setiap aktifitasmu https://t.co/UrZIcjLqLw	aktifitasmu
⋮	⋮	⋮
1267	@pandji yuk sdh saatnya lepas masker, muka ganteng gue ketutup mulu.. bodo amat apa yg diomongin mereka berdua	yuk sdh saatnya lepas masker muka ganteng gue ketutup mulu bodo amat apa yg diomongin mereka berdua

Table 2 shows a data structure before and after the method of data cleansing. Documents that have gone through this process do not include assumed characters like URLs, usernames, hashtags, numbers, or punctuation. The next step is case folding.

2. Case Folding

Case folding is a procedure for reducing cases in a document. **Table 3** shows the data structure before and after going through the process of case folding.

Table 3. Data structure before and after processing case folding

No	Text Before Case Folding Process	Text After Case Folding Process
1	udah saatnya kita lepas masker	udah saatnya kita lepas masker
2	Sampai skrg belum berani ke mall disaat weekend Belum berani lepas masker Belum berani ke tempat keramaian	sampai skrg belum berani ke mall disaat weekend belum berani lepas masker belum berani ke tempat keramaian
3	Dimanapun kapanpun kamu berada selalu gunakan masker Jangan lepas maskermu Dan selalu patuhi di setiap aktifitasmu	dimanapun kapanpun kamu berada selalu gunakan masker jangan lepas maskermu dan selalu patuhi di setiap aktifitasmu
⋮	⋮	⋮
1267	yuk sdh saatnya lepas masker muka ganteng gue ketutup mulu bodo amat apa yg diomongin mereka berdua	yuk sdh saatnya lepas masker muka ganteng gue ketutup mulu bodo amat apa yg diomongin mereka berdua

Table 3 shows the data structure prior to and during the process of case folding. Documents that have gone through the process of case folding into one standard form, in which all words are transformed to lowercase in order to eliminate case-sensitive classification.

3. Tokenizing

Tokenizing is the process of separating the substance of a sentence into individual words (terms) so that it can be evaluated. **Table 4** shows a data structure before and after tokenization.

Table 4. Data structure before and after processing tokenizing

No	Text Before Tokenizing Process	Text After Tokenizing Process
1	udah saatnya kita lepas masker	['udah', 'saatnya', 'kita', 'lepas', 'masker']
2	sampai skrg belum berani ke mall disaat weekend belum berani lepas masker belum berani ke tempat keramaian	['sampai', 'skrg', 'belum', 'berani', 'ke', 'mall', 'disaat', 'weekend', 'belum', 'berani', 'lepas', 'masker', 'belum', 'berani', 'ke', 'tempat', 'keramaian']
3	dimanapun kapanpun kamu berada selalu gunakan masker jangan lepas maskermu dan selalu patuhi di setiap aktifitasmu	['dimanapun', 'kapanpun', 'kamu', 'berada', 'selalu', 'gunakan', 'masker', 'jangan', 'lepas', 'maskermu', 'dan', 'selalu', 'patuhi', 'di', 'setiap', 'aktifitasmu']
⋮	⋮	⋮
1267	yuk sdh saatnya lepas masker muka ganteng gue ketutup mulu bodo amat apa yg diomongin mereka berdua	['yuk', 'sdh', 'saatnya', 'lepas', 'masker', 'muka', 'ganteng', 'gue', 'ketutup', 'mulu', 'bodo', 'amat', 'apa',

 ISSN:2302-8149
e-ISSN:2540-9719

		'yg', 'diomongin', 'mereka', 'berdua']

Table 4 shows the data structure before and after the tokenization procedure. Following this stage, tokens are obtained in the form of individual words (terms), which are then processed for stopword removal and stemming.

4. Stopword Removal

Stopword removal is a procedure that reduces the number of words and conjunctions that exist in a document but does not add substantial significance to its contents. **Table 5** shows the data structure before and after the process of stopword removal.

Table 5. Data structure before and after processing stopword removal

No	Text Before Stopword Removal Process	Text After Stopword Removal Process
1	['udah', 'saatnya', 'kita', 'lepas', 'masker']	['udah', 'saatnya', 'kita', 'lepas', 'masker']
2	['sampai', 'skrg', 'belum', 'berani', 'ke', 'mall', 'disaat', 'weekend', 'belum', 'berani', 'lepas', 'masker', 'belum', 'berani', 'ke', 'tempat', 'keramaian']	['sampai', 'skrg', 'belum', 'berani', 'ke', 'mall', 'disaat', 'weekend', 'belum', 'berani', 'lepas', 'masker', 'belum', 'berani', 'ke', 'tempat', 'keramaian']
3	['dimanapun', 'kapanpun', 'kamu', 'berada', 'selalu', 'gunakan', 'masker', 'jangan', 'lepas', 'maskermu', 'dan', 'selalu', 'patuhi', 'di', 'setiap', 'aktifitasmu']	['dimanapun', 'kapanpun', 'kamu', 'berada', 'selalu', 'gunakan', 'masker', 'jangan', 'lepas', 'maskermu', 'dan', 'selalu', 'patuhi', 'di', 'setiap', 'aktifitasmu']
⋮	⋮	⋮
1267	['yuk', 'sdh', 'saatnya', 'lepas', 'masker', 'muka', 'ganteng', 'gue', 'ketutup', 'mulu', 'bodo', 'amat', 'apa', 'yg', 'diomongin', 'mereka', 'berdua']	['yuk', 'sdh', 'saatnya', 'lepas', 'masker', 'muka', 'ganteng', 'gue', 'ketutup', 'mulu', 'bodo', 'amat', 'apa', 'yg', 'diomongin', 'mereka', 'berdua']

Table 5 shows the data structure before and after the process of stopword removal. Following this stage, word tokens are obtained that appear infrequently yet hold significant significance in the material.

5. Stemming

Stemming removes prefixes, suffixes, infixes, and confixes. **Table 6** shows the data structure before and after the procedure stemmed.

Table 6. Data structure before and after process stemming

No	Text Before Stemming Process	Text After Stemming Process
1	['udah', 'saatnya', 'kita', 'lepas', 'masker']	['udah', 'saat', 'kita', 'lepas', 'masker']
2	['sampai', 'skrg', 'belum', 'berani', 'ke', 'mall', 'disaat', 'weekend', 'belum', 'berani', 'lepas', 'masker', 'belum', 'berani', 'ke', 'tempat', 'keramaian']	['sampai', 'skrg', 'belum', 'berani', 'ke', 'mall', 'saat', 'weekend', 'belum', 'berani', 'lepas', 'masker', 'belum', 'berani', 'ke', 'tempat', 'ramai']
3	['dimanapun', 'kapanpun', 'kamu', 'berada', 'selalu', 'gunakan', 'masker', 'jangan', 'lepas', 'maskermu', 'dan', 'selalu', 'patuhi', 'di', 'setiap', 'aktifitasmu']	['mana', 'kapan', 'kamu', 'ada', 'selalu', 'guna', 'masker', 'jangan', 'lepas', 'masker', 'dan', 'selalu', 'patuh', 'di', 'tiap', 'aktifitasmu']
⋮	⋮	⋮
1267	['yuk', 'sdh', 'saatnya', 'lepas', 'masker', 'muka', 'ganteng', 'gue', 'ketutup', 'mulu', 'bodo', 'amat', 'apa', 'yg', 'diomongin', 'mereka', 'berdua']	['yuk', 'sdh', 'saat', 'lepas', 'masker', 'muka', 'ganteng', 'gue', 'tutup', 'mulu', 'bodo', 'amat', 'apa', 'yg', 'diomongin', 'mereka', 'dua']

 ISSN:2302-8149
e-ISSN:2540-9719

Table 6 compares the data structure before and after the stemming process. The stemming process generates word tokens that do not include prefixes, suffixes, infixes, or confixes.

4.3 Term Weighting with Term Frequency-Inverse Document Frequency

Following the preprocessing of text input, TF-IDF calculations are performed to determine the weight value for each fundamental word generated from the stemming process. Table 7 shows the word weighting values for each term. **Table 7** displays the TF-IDF values for each term in the full document.

Table 7. Term weighting values with tf-idf

No	Abis	Aib	Aja	Udah	Aktivitas	Akut	⋯	Yuk
1	0	0	0	2,44	0	0	⋯	0
2	0	0	0	0	0	0	⋯	0
3	0	0	0	0	0	0	⋯	0
4	0	0	3,14	0	0	0	⋯	0
5	0	1,06	0	0	0	0	⋯	0
⋮	…	…	…	…	…	…	…	…
1265	0	0	0	0	6,07	0	⋯	0
1266	0	5,25	0	0	0	0	⋯	0
1267	0	0	0	0	0	0	⋯	5,95

Table 7 shows the TF-IDF value for each term in the entire document. **Table 7** displays 400 terms as columns and 1267 documents as rows. The higher the TF-IDF number, the more essential the term is. The high TF-IDF score is due to the vast number of occurrences of the word in a document and the low frequency of occurrence in the overall dataset. The generated TF-IDF values will be utilized as input for combined sampling and classification.using the KNN algorithm.

4.4 Separation of Training and Test Data

Data sharing plays a vital role in machine learning, particularly when developing data models. When two data sets are divided, training and test data are often generated. Training data is used as a reference while developing a classification model. The test data is used to evaluate the classification model's performance. In this study, the dataset will be separated into two parts: 70% for training data, which will be used to create a classification model using the KNN algorithm, and 30% for test data, which will be used to validate the classification model created. **Table 8** shows the proportions of training and test data.

Table 8. Proportion of training data and test data

Data Type	Classification Class		Total
	Positive	Negative	
Data Latih	234	652	886
Data Uji	123	258	381
Total	357	910	1.267

Table 8 shows the quantity of 886 training data that will be used to develop the classification model and 381 test data, which will be used to evaluate the classification model that has been formed. 886 of the training data, 234 were positive and 652 were negative. The amount of training data between the two classes is not equal. As a result, using combine sampling, the amount of training data for the two classes will be balanced, but the amount of test data in the two classes will not.

 ISSN:2302-8149
e-ISSN:2540-9719

4.5 Balancing the Amount of Training Data with Combine Sampling

The final step before classification is to balance the amount of training data in both the positive and negative classes. Combine sampling comprises phases of oversampling using SMOTE, followed by steps of undersampling using Tomek Links to balance imbalanced data. **Table 9** shows the percentage of training data before and after combine sampling.

Table 9. Proportion of training data before and after *combine sampling*

	Positive	Negative	Total
Before	234	652	886
Combine Sampling	**613**	**613**	**1.226**

Table 9 shows the percentage of training data following the combine sample stage. After the combine sampling stage, there were 1266 training data points, 613 of which were positive and 613 were negative. A bar chart illustrating the proportion of training data before and after the combine sampling stage follows:

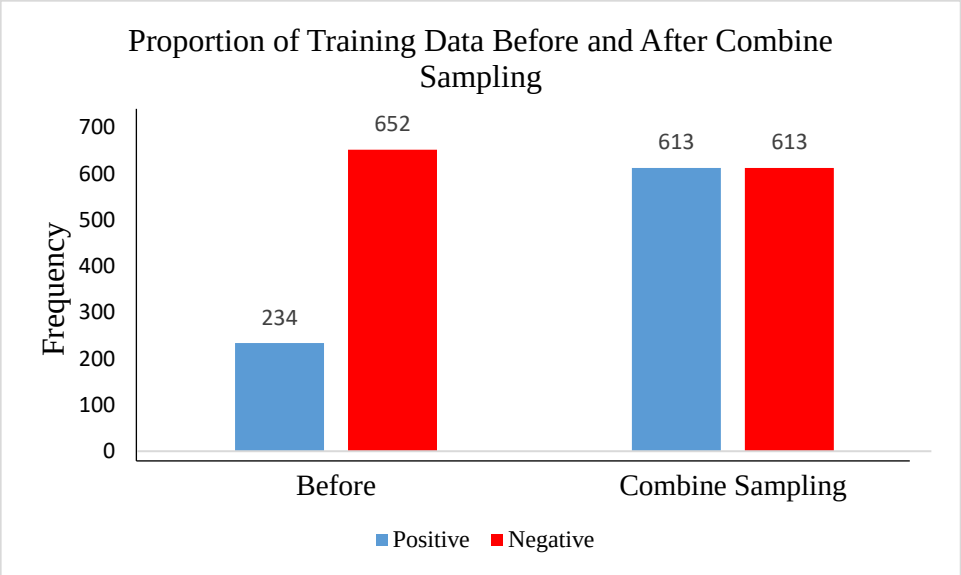

Figure 2. Bar chart proportion of training data before and after combine sampling

Figure 2 shows the amount of training data, which is 234 for the positive data class and 652 for the negative data class, before the combine sampling stage. Because the amount of data in the two classes is not evenly distributed, combined sampling will be performed. The negative data class is significant since it contains more data than the positive data class, whereas the positive data class is minor. The first step of combined sampling is oversampling using SMOTE. Data in the minor class, namely the positive data class, will be replicated based on resemblance to its nearest neighbors, ensuring that the amount of data in both classes is the same. The next step is under sampling with Tomek Links. Data in major and minor classes will be analyzed, and any data points that are close together but belong to other classes will be eliminated. Following the combined sampling method, a balanced amount of training data was acquired, with the positive and negative classes having the same amount of data (613 data). **Table 10** shows training data that has been balanced using combined sampling.

Table 10. Train data after combine sampling

No	Abis	Aib	Aja	Udah	Aktivitas	Akut	⋯	Yuk	Sentimen
1	0	0	0	0	0	0	⋯	0	Positive

 ISSN:2302-8149
e-ISSN:2540-9719

No	Abis	Aib	Aja	Udah	Aktivitas	Akut	⋯	Yuk	Sentimen
2	0	0	0	0	0	0	⋯	0	Negative
3	0	0	0	0	0	0	⋯	0	Negative
4	0	0	3,13	0	0	5,44	⋯	0	Negative
5	0	0	0	0	0	0	⋯	0	Negative
⋮	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋮
1224	0	0	0	0	0	0	⋯	0	Positive
1225	0	0	0	1,92	0	0	⋯	0	Positive
1226	0	0	0	0,39	0	0	⋯	0	Positive

Table 10 shows training data following the combine sample stage, which includes 400 terms with each TF-IDF value. After collecting a balanced amount of data from both classes, the K-Nearest Neighbor technique is used to classify the results.

4.6 Classification Using K-Nearest Neighbor Algorithm

The classification stage continues the stage of balancing the amount of training data in the two classes. The KNN method has no requirements for picking the values *k* that must be used. To avoid inflexible classification findings, this study used values ranging from 2 to 10 instead of 1 to 10. The following step is to compute the accuracy value for each parameter selection. The *k* value with the highest accuracy will be used as a parameter for the sentiment classification of the mask removal policy in Indonesia. **Table 11** shows the accuracy values for each parameter.

Table 11. Accuracy values for each parameter *k*

k value	Accuracy
2	0,60
3	0,50
4	0,53
5	0,48
6	0,53
7	0,49
8	0,52
9	0,49
10	0,53

Table 11 compares the parameter value *k* with the best accuracy to the value *k* that the other is equal to. Aside from that, it can be shown that in the case of mask-free policy sentiment in Indonesia, value *k* even has a greater accuracy value than value *k* odd. So, the parameter *k* that will be employed in this research is 2. The next stage is to calculate the percentage of the KNN algorithm's performance in classifying test data using accuracy, precision, and recall.

4.7 Classification Algorithm Perfomance Percentage Test

The classification model's performance is evaluated using a confusion matrix based on prediction results from test data. **Table 12** shows the confusion matrix prediction results for the mask removal policy using KNN.

Table 12. Confusion Matrix KNN prediction results

Actual Class	**Prediction Class**	
	Positive	**Negative**
Positive	186	100
Negative	51	44

 ISSN:2302-8149
e-ISSN:2540-9719

Based on the confusion matrix in **Table 12**, 186 TP or positive sentiment data were properly predicted out of 381 test data. 44 TN data or negative sentiment data were accurately predicted, 100 FN data or positive sentiment data were wrongly predicted as negative sentiment, and 51 FP data or data Negative sentiment was incorrectly predicted as positive sentiment.

A Confusion matrix on **Table 12** will be used to calculate the performance percentage of the KNN algorithm using **Equations (5)**, **(6)**, and **(7)**:

1. The accuracy value is calculated using **Equation (5)**

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$
$$= \frac{186 + 44}{186 + 44 + 51 + 100} = \frac{230}{381} \times 100\% = 60{,}4\ \%$$

2. The precision value is calculated using **Equation (6)**

$$\text{precision} = \frac{TP}{TP + FP} \times 100\%$$
$$= \frac{186}{186 + 51} \times 100\% = \frac{186}{237} \times 100\% = 78,5\%$$

3. The recall (sensitivity) is calculated using **Equation (7)**

$$\text{recall} = \frac{TP}{TP + FN} \times 100\%$$
$$= \frac{186}{186 + 100} \times 100\% = \frac{186}{286} \times 100\% = 65{,}0\%$$

Table 13 displays the performance of the KNN algorithm in classifying data from the mask removal policy in Indonesia.

Table 13. KNN classification performance

	Accuracy	Precision	Recall
k = 2	60,4%	78,5%	65%

The KNN algorithm produces an accuracy of 60,4%. This means that 60,4% of the positive and negative sentiment data is accurately predicted from the whole data. Precision is 78,5% defined as the percentage of positive sentiment data that is accurately predicted relative to the total data projected as positive sentiment. The recall amount 65% is the percentage of positive sentiment data that is accurately anticipated relative to the entire quantity of positive sentiment data. The rationale for getting an accuracy value of 60,4% is the meaning that each person wishes to express. Machine learning misinterpreted Indonesia's mask removal policy, resulting in misclassification.

5 Conclusion

Combine sampling gives a balanced quantity of training data in each class, with 613 data in the positive class and 613 data in the negative class. Next, the performance or classification performance utilizes an algorithm K-Nearest Neighbor on the sentiment of the policy of removing masks in Indonesia, with an accuracy of 60,4%, a precision percentage of 78,5%, and a percentage recall of 65%. The reason for attaining an accuracy value of 60,4% is the meaning that for each person to convey a tweet via twitter, machine learning misinterpreted Indonesia's mask removal policy, resulting in misclassification.

Reference

[1] Y. Tresnawati, "Analisis Sentimen Pada Twitter Menggunakan Pendekatan Agglomerative Hierarchial Clustering," Universitas Sanata Dharma, 2017. [Online]. Available: https://123dok.com/document/y6e7jl4z-analisis-sentimen-twitter-menggunakan-pendekatan-agglomerative-hierarchical-clustering.html

[2] A. R. Isnain, J. Supriyanto, and M. P. Kharisma, "Implementation of K-Nearest Neighbor (K-

 ISSN:2302-8149
e-ISSN:2540-9719

NN) Algorithm For Public Sentiment Analysis of Online Learning," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 2, p. 121, Apr. 2021, doi: 10.22146/ijccs.65176.

[3] S. Mulyani, S. A. Thamrin, and S. Siswanto, "Analisis Sentimen Masyarakat Pada Kebijakan Vaksinasi Covid-19 Di Twitter Menggunakan Metode Mesin Vektor Pendukung Dengan Kernel Radial Basis Function Berbasis Fitur Leksikon," *Jambura J. Probab. Stat.*, vol. 3, no. 2, pp. 110–119, 2022, doi: 10.34312/jjps.v3i2.16663.

[4] N. Rezki, S. A. Thamrin, and S. Siswanto, "Sentiment Analysis of Merdeka Belajar Kampus Merdeka Policy Using Support Vector Machine With Word2Vec," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 17, no. 1, pp. 0481–0486, 2023, doi: 10.30598/barekengvol17iss1pp0481-0486.

[5] R. T. Prasetio, "Seleksi Fitur dan Optimasi Parameter k-NN Berbasis Algoritma Genetika Pada Dataset Medis," *J. RESPONSIF*, vol. 2, no. 2, pp. 213–221, 2020, [Online]. Available: http://ejurnal.ars.ac.id/index.php/jti

[6] A. Prayoga Permana, K. Ainiyah, and K. Fahmi Hayati Holle, "Analisis Perbandingan Algoritma Decision Tree, kNN, dan Naive Bayes untuk Prediksi Kesuksesan Start-up," *JISKa*, vol. 6, no. 3, pp. 178–188, 2021, [Online]. Available: http://repository.uin-malang.ac.id/9921/

[7] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," 2004. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1007730.1007735

[8] S. Choirunnisa and J. Lianto, "Hybrid Method of Undersampling and Oversampling for Handling Imbalanced Data," in *2018 International Seminar On Research Of Information Technology and Intelligent Systems*, 2018, pp. 276–280.

[9] M. Mustaqim, B. Warsito, and B. Surarso, "Combination of synthetic minority oversampling technique (Smote) and backpropagation neural network to handle imbalanced class in predicting the use of contraceptive implants," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 5, no. 2, pp. 116–127, Jul. 2019, doi: 10.26594/register.v5i2.1705.

[10] H. Ali, M. N. M. Salleh, R. Saedudin, K. Hussain, and M. F. Mushtaq, "Imbalance class problems in data mining: A review," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 3, pp. 1552–1563, Jun. 2019, doi: 10.11591/ijeecs.v14.i3.pp1552-1563.

[11] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. Akmal Khalid, "Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset," in *IEEE International Conference on Control and Automation, ICCA*, IEEE Computer Society, Oct. 2020, pp. 803–808. doi: 10.1109/ICCA51439.2020.9264517.

[12] C. M. F. Andriani and D. Susilaningrum, "Klasifikasi Waiting Time for Pilot di Pelabuhan Tanjung Perak Menggunakan Metode Regresi Logistik – Synthetic Minority Oversampling Technique (SMOTE)," *J. Sains dan Seni ITS*, vol. 12, no. 1, pp. 111–118, 2023.

[13] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, and F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 3, pp. 677–690, Jul. 2022, doi: 10.30812/matrik.v21i3.1726.

[14] L. Ganda, R. Putra, K. Marzuki, and H. Hairani, "Correlation-based feature selection and Smote-Tomek Link to improve the performance of machine learning methods on cancer

 ISSN:2302-8149
e-ISSN:2540-9719

disease prediction," *Eng. Appl. Sci. Res.*, vol. 50, no. 6, pp. 577–583, 2023, doi: 10.14456/easr.2023.59.

[15] A. Baita and N. Cahyono, "Analisis Sentimen Mengenai Vaksin SINOVAC Menggunakan Algoritma Support Vector Machine (SVM) DAN K-Nearest Neighbor (KNN)," *Inf. Syst. J.*, vol. 4, no. 2, pp. 42–46, 2021.

[16] P. Shah, P. Swaminarayan, and M. Patel, "Sentiment analysis on film review in Gujarati language using machine learning," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1. Institute of Advanced Engineering and Science, pp. 1030–1039, Feb. 01, 2022. doi: 10.11591/ijece.v12i1.pp1030-1039.

[17] H. Hairani, A. Anggrawan, and D. Priyanto, "Improvement Performance of the Random Forest Method on Unbalanced Diabetes Data Classification Using Smote," *Int. J. Informatics Vis.*, vol. 7, no. 1, pp. 258–264, 2023.

[18] K. Alamat, W. Nugraha, D. Risdiansyah, D. Purwaningtias, T. Hidayatulloh, and S. Suhada, "Kombinasi Tomek Link dan SMOTE Untuk Mengatasi Ketidakseimbangan Kelas Pada Credit Card Fraud," *J. Larik*, vol. 2, no. 2, pp. 32–40, 2022, [Online]. Available: http://jurnal.bsi.ac.id/index.php/larik

[19] I. N. Switrayana, D. Ashadi, H. Hairani, and A. Aminuddin, "Sentiment Analysis and Topic Modeling of Kitabisa Applications using Support Vector Machine (SVM) and Smote-Tomek Links Methods," *Int. J. Eng. Comput. Sci. Appl.*, vol. 2, no. 2, pp. 81–91, Sep. 2023, doi: 10.30812/ijecsa.v2i2.3406.

[20] E. Gusniawan Pradana, "Implementasi Web Crawler Untuk Mencari Harga Barang Termurah Dari Berbagai Situs E-Commerce Indonesia," *J. Teknol. Pint.*, vol. 2, no. 9, pp. 1–11, 2022.

[21] J. Budiarto, "Identifikasi Kebutuhan Masyarakat Nusa Tenggara Barat pada Pandemi Covid-19 di Media Sosial dengan Metode Crawling (Requirements Identification for NTB People in pandemic covid-19 at Social Media Using Crawling Method)," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 2, no. 4, pp. 244–250, 2021.

[22] P. Y. Saputra, "Implementasi Teknik Crawling Untuk Pengumpulan Data Dari Media Sosial Twitter," *J. Din.*, vol. 8, no. 2, pp. 160–168, 2017, [Online]. Available: www.quicksprout.com

[23] M. Yusran, S. Rasyid, E. Sagita, R. N. D. Julia, and Siswanto, "Sentiment Analysis of Sustainable Development Goals on Twitter with Classifying Decision Tree C5.0 and Classification and Regression Tree," *Int. J. Acad. Appl. Res.*, vol. 6, no. 6, pp. 104–110, 2022, [Online]. Available: www.ijeais.org/ijaar

[24] T. D. Dikiyanti, A. M. Rukmi, and M. I. Irawan, "Sentiment analysis and topic modeling of BPJS Kesehatan based on twitter crawling data using Indonesian Sentiment Lexicon and Latent Dirichlet Allocation algorithm," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1821/1/012054.

[25] W. Astuti, D. Djoko, and A. Widodo, "Pemetaan Tindak Kejahatan Jalanan di Kota Semarang Menggunakan Algoritma K-Means Clustering," *J. Tek. Elektro*, vol. 8, no. 1, pp. 5–7, 2016.

[26] C. Sains Teknologi, S. Pakpahan, A. Manullang, and K. Kunci, "Analisis Sentimen Integritas KPK Tahun 2021 Pencegahan Korupsi pada Twitter KPK menggunakan Metode K-Nearest Neighbor dan Naive Bayes," *Citra Sains Teknol.*, vol. 2, no. 1, pp. 63–73, 2022.

[27] M. S. Bahri, A. Hermawan, E. Pricilia Kondy, and R. Joyce Semida, "Performance

 ISSN:2302-8149
e-ISSN:2540-9719

Comparison of Supporting Vector Machine Method without or with Particle Swarm Optimization Based on Sentiment Analysis WhatsApp Review," *Int. J. Acad. Appl. Res.*, vol. 6, no. 6, pp. 94–101, 2022, [Online]. Available: www.ijeais.org/ijaar

[28] W. E. Nurjanah, R. Setya Perdana, and M. A. Fauzi, "Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode K-Nearest Neighbor dan Pembobotan Jumlah Retweet," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1750–1757, 2017.

[29] R. Adinugroho, "Perbandingan Rasio Split Data Training dan Data Testing Menggunakan Metode LSTM Dalam Memprediksi Harga Indeks Saham Asia," 2022. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/67314

[30] S. Rabbani, D. Safitri, N. Rahmadhani, A. A. F. Sani, and M. K. Anam, "Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 153–160, Oct. 2023, doi: 10.57152/malcom.v3i2.897.

[31] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional," *J. Tekno Kompak*, vol. 15, no. 1, pp. 131–145, 2021.

[32] A. T. Putra, E. Kardinata, H. Junaedi, F. Chandra, and J. Santoso, "Ekstraksi Relasi Antar Entitas di Bahasa Indonesia Menggunakan Neural Network," *J. Inf. Syst. Hosp. Technol.*, vol. 3, no. 02, pp. 49–54, Oct. 2021, doi: 10.37823/insight.v3i02.156.

[33] E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset," *Sensors*, vol. 22, no. 9, May 2022, doi: 10.3390/s22093246.

[34] R. M. Sari and A. Prasetyo, "Penerapan Synthetic Minority Oversampling Technique terhadap Data Perokok Anak di Nusa Tenggara Barat Tahun 2021," *Inferensi*, vol. 6, no. 2, p. 133, Sep. 2023, doi: 10.12962/j27213862.v6i2.18472.

[35] A. K. Duggal and M. Dave, "A Comparative Study of Load Balancing Algorithms in a Cloud Environment ..," in *Advances in Computing and Intelligent Systems Algorithms for Intelligent Systems Series*, 2019, pp. 115–126. [Online]. Available: http://www.springer.com/series/16171

[36] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Inf. Sci. (Ny).*, vol. 465, pp. 1–20, Oct. 2018, doi: 10.1016/j.ins.2018.06.056.

[37] D. C. R. Novitasari, M. F. Rozi, and R. Veriani, "Klasifikasi Kelainan Pada Jantung Melalui Citra Iris Mata Menggunakan Fuzzy C-Means Sebagai Pengambil Fitur Iris dan Klasifikasi Menggunakan Support Vector Machine," *INTEGER J. Inf. Technol.*, vol. 4, no. 1, pp. 1–10, 2019.

[38] D. Devi, S. K. Biswas, and B. Purkayastha, "A Review on Solution to Class Imbalance Problem: Undersampling Approaches," in *2020 International Conference on Computational Performance Evaluation (ComPE)*, 2020, pp. 626–631.

[39] L. M. Sinaga, Sawaluddin, and S. Suwilo, "Analysis of classification and Naïve Bayes algorithm k-nearest neighbor in data mining," in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/725/1/012106.

 ISSN:2302-8149
e-ISSN:2540-9719

[40] R. Damarta, A. Hidayat, and A. S. Abdullah, "The Application of K-Nearest Neighbors Classifier For Sentiment Analysis of PT PLN (Persero) Twitter Account Service Quality," in *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1722/1/012002.

[41] Anggi Priliani Yulianto and S. Darwis, "Penerapan Metode K-Nearest Neighbors (kNN) pada Bearing," *J. Ris. Stat.*, vol. 1, no. 1, pp. 10–18, Jul. 2021, doi: 10.29313/jrs.v1i1.16.

[42] S. Dyah Fritama, Y. Raymond Ramadhan, and M. Andayani Komara, "Analisis Sentimen Review Produk Acne Spot Treatment di Female Daily Menggunakan Algoritma K-Nearest Neighbor," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 1, pp. 134–143, 2023, doi: 10.30865/klik.v4i1.1070.

[43] A. Habibie and I. Rachmawati, "Analisis Preferensi Konsumen Dalam Memilih Smartphone di Indonesia Consumer Analysis of Preferences in Choosing Smartphone in Indonesia," in *e-Proceeding of Management*, 2020, pp. 114–124.

[44] Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, "Image classification based on quantum K-Nearest-Neighbor algorithm," *Quantum Inf. Process.*, vol. 17, no. 9, pp. 1–18, Sep. 2018, doi: 10.1007/s11128-018-2004-9.

[45] Indrayanti, D. Sugianti, and M. Karomi, Al Adib, "Optimasi Parameter Pada Algoritma K-Nearest Neighbour Untuk Klasifikasi Penyakit Diabetes Mellitus," in *Prosiding SNATIF*, 2017, pp. 823–829.

[46] S. Ruuska, W. Hämäläinen, S. Kajava, M. Mughal, P. Matilainen, and J. Mononen, "Evaluation of the Confusion Matrix Method in the Validation of an Automated System For Measuring Feeding Behaviour of Cattle," *Behav. Processes*, vol. 148, pp. 56–62, Mar. 2018, doi: 10.1016/j.beproc.2018.01.004.

[47] A. . Ihsan, "Reduksi Atribut Pada Algoritma K-Nearest Neighbor (KNN) Dengan Menggunakan Algoritma Genetika," 2018. [Online]. Available: https://repositori.usu.ac.id/handle/123456789/3878

[48] G. Zeng, "On the Confusion Matrix In Credit Scoring and Its Analytical Properties," *Commun. Stat. - Theory Methods*, vol. 49, no. 9, pp. 2080–2093, 2019, doi: 10.1080/03610926.2019.1568485.

[49] B. Juba and H. S. Le, "Precision-Recall versus Accuracy and the Role of Large Data Sets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 4039–4048. [Online]. Available: www.aaai.org