Comparison of Scenario Pre-processing Performance on Support Vector Machine and Naïve Bayes Algorithms for Sentiment Analysis

Novrido Charibaldi[1], Nabila Valinka Pusean[2], Budi Santosa[3]

[1,2,3]*Informatics, Veterans National Development University Yogyakarta, Indonesia*

[1]novrido@upnyk.ac.id (*)

[2]valinkanabila@gmail.com

[3]dissan@upnyk.ac.id

Received: 2022-12-22; Accepted: 2023-01-24; Published: 2023-01-28

Abstract— Television shows need a rating in their assessment, but public opinion is also required to complete it. Sentiment analysis is necessary for its completion. An essential step in sentiment analysis is pre-processing because, in public opinion, there are still many inappropriate writings. This study aims to compare the performance results using different pre-processing scenarios to get the best pre-processing performance on Support Vector Machine (SVM) and Naïve Bayes (NB) on sentiment analysis about the television show *X Factor Indonesia*. The stages used to start from literature study, problem analysis, design, data collection, pre-processing with two scenarios, word weighting with TF-IDF, classification using SVM and NB, then resulting accuracy from Confusion Matrix. The findings of this research are that optimal performance can be achieved using a comprehensive pre-processing scenario. This scenario should include the following steps: case-folding, removing emoji, cleansing, removing repetition characters, word normalization, negation handling, stopwords removal, stemming, and tokenization, with an accuracy of 79.44% on the SVM algorithm. This research shows that the complete pre-processing of the SVM algorithm is better in terms of accuracy, precision, recall, and F1-score.

Keywords— Sentiment Analysis; Pre-processing; Support Vector Machines; Naïve Bayes; Television Program.

I. INTRODUCTION

Every day, television presents various types of programs, such as talk shows, variety shows, documentaries, movies, news, and more. A certain level of popularity is required to maintain that impression and attract viewers. Television viewing ratings can be seen through the rating. The rating results will be made into a report and evaluation that will determine whether the program is still broadcast or replaced. Therefore, the rating value is highly important for the survival of a television show, but having a rating does not guarantee that the show is of high quality. As a result, public opinion is required because it can be used to perform sentiment analysis in predicting whether a television show's value is neutral, positive, or negative examples for journal articles [1].

Researchers [2] have carried out using Retweet Analysis and NB. This study did not normalize words and lemmatization as pre-processing stages, so the resulting accuracy was not optimal, namely 61%. This accuracy is also affected by small data and unbalanced data for each class.

The results showed that the best pre-processing performance was obtained by using a combination of cleaning and stemming; and word normalization, cleaning, and stemming with an accuracy value of 77.77%. However, repetition and negation have not been carried out in this study. There is also sample journal articles in [3] on the effect of pre-processing using NB, Maximum Entropy, and SVM. The best accuracy was obtained by NB, which was 91.81%. This happened after carrying out several pre-processing steps such as case folding, removing emoji, stopwords removal, stemming, and tokenization. Accuracy is also affected by word vectorization using CBOW, Skipgram, and Bigram. However, this research only carried out five pre-processing steps, so further research is needed using other pre-processing steps.

This study will focus on finding out which of the pre-processing scenarios in the algorithm gives the best accuracy using SVM, NB, and TF-IDF as word weighting. This study will compare the pre-processing 1st scenario, including case folding, removing emoji, stopwords removal, stemming, and tokenization. While the pre-processing 2nd scenario includes case folding, removing emoji, cleansing, removing repetition characters, normalizing words, negation handling, removing stopwords, stemming, and tokenization. SVM and NB are compared because sample journal articles in [3] have compared these algorithms. In this study, a comparison of the SVM and NB algorithms will also be carried out. SVM and NB is also a model in Supervised Learning. Supervised handling techniques label data where the system knows the output data pattern. This creates a model that Supervised Learning is more accurate than the Unsupervised Learning model because the expected output is known in advance. The SVM method can also be used with small data examples [4], and NB does not require large data sets. SVM and NB algorithms also do not require much memory, so the training time is relatively fast for journal articles [5]. The goal to be achieved in this study is to compare the performance using different pre-processing scenarios on SVM and NB.

DOI : http://dx.doi.org/10.25139/inform.v8i1.5667

II. Research Methodology

In this study, four test scenarios were carried out that focused on looking for performance pre-processing scenarios on the SVM or NB algorithms that had the best accuracy. Sentiment analysis testing was carried out with 900 tweets from XFactorID television, divided into 80% data train and 20% data test, resulting in 720 train data and 180 test data. Data from tweets are retrieved using automated scraping from 2021 to 2022.

Data collection is the first research stage, followed by labeling and separating train data and data from other sources. After being separated, it then underwent pre-processing. There are two scenarios at the pre-processing stage: pre-processing in 1st scenario and pre-processing in 2nd scenario. After pre-processing is carried out, the next step is weighting with TF-IDF. After obtaining the weight of each word, then the next step is classification. The SVM and NB algorithms are used in the classification process of this research. From the classification results, we will evaluate the performance of sentiment analysis using the Confusion Matrix to determine the accuracy, precision, recall, and F1-score performance of the system being built.

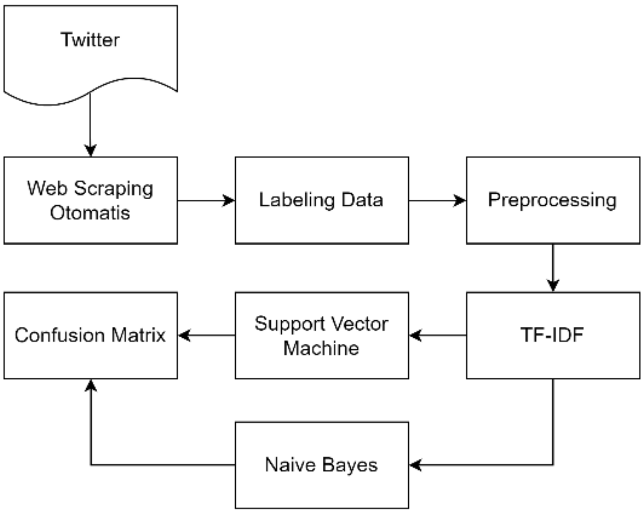

Figure 1. Research Methodology

As shown in Figure 1, which is the research methodology flow, there are seven steps, web scraping, data labeling, pre-processing, TF-IDF, SVM, NB, and Confusion Matrix.

A. Data collection

Data retrieval is done by an automatic scraping method. Data scraping is done using the Snscrape library. The data taken is from the tweets that include the keyword "XFactorID". Tweet data is collected beginning in July 2021 and ending in 2022. The data taken is 900, divided by 80% and 20%, so the training data is 720, and the test data is 180.

TABLE I
DATA COLLECTION

Category	**Quantity**
Positive	364
Neutral	279
Negative	257

B. Data Labeling

Labeling is based on three sentiment classes: positive, negative, and neutral. This labeling was done manually by a student majoring in Indonesian Literature at Yogyakarta State University. The purpose of labeling by Language Department students in Indonesia is to ensure that sentiment results are more accurate because more expert does labeling.

TABLE II
DATA LABELING

Tweets	**Label**
Keren lo vin,lagu dangdut classic menjadi lebih fresh dan kekinian	Positive
Masih menunggu kemungkinan @DnrWidianto menyanyikan "Belum Ada Judul"(Iwan Fals)	Neutral
kualitas terkalahkan dengan kuantitas!!!!!!!!!	Negative

As shown in Table II, the examples of data labeling that given manually. Three examples of data labeling received positive, neutral, and negative labels.

C. Pre-processing

Pre-processing is preparing and cleaning raw data that has already passed labelling [6]. The pre-processing used is case folding, removing emoji, cleansing, removing repetition characters, word normalization, negation handling, stopwords removal, stemming, and tokenization. Case fold a document, the letters in it will be changed to lowercase, as seen in the examples for journal articles on [2] if the delete emoji command is used on the document. The emoji symbol will be removed from it. Examples of this command may be seen in [3] journal articles. During cleaning, URLs, usernames, punctuation, and unnecessary spaces, such as those found in examples for journal articles [7], will be removed. Remove repetition character is the removal of repeated letters in a word, examples for journal articles on [8]. This process uses the help of the Regex library from Python to find and delete more than one character. Normalization is the process by which non-standard words become standard, according to the available dictionaries and examples for journal articles [9]. Negation handling is the change of word after word that does not become an antonym, examples for journal articles on [9]. Stopword removal removes words that do not have useful information, examples for journal articles [3]. Stemming is removing modulated words and creating basic examples for journal articles [10]. Tokenization is solving sentences based on each word that composes them, examples for journal articles on [2].

The use of pre-processing sequences is based on research [11] which consists of case folding, removing emoji, normalizing words, stopword removal, and stemming. Then, for the process of cleansing and removing repetition characters, the process was added to the sequence after removing the emoji due to complete data cleaning. The negation handling process is in the order after the normalized word; because the order is after, the data has become a more standard word. Then do negation handling and stopwords removal so that the word "no" after negation handling becomes erased. This study will compare two pre-processing scenarios, 1st scenario, and 2nd scenario.

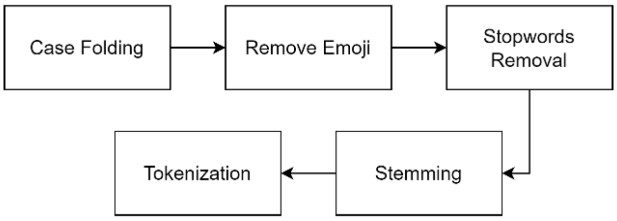

Figure 2. Pre-processing 1st scenario

As shown in Figure 2, the pre-processing 1st scenario was taken from sample journal articles in [3], including case folding, removing emoji, stopwords removal, stemming, and tokenization. After the data is carried out in pre-processing 1st scenario, the results are shown in Table III.

TABLE III
PRE-PROCESSING 1ST SCENARIO

Before	After
Msh menunggu kemungkinan @DnrWidianto menyanyikan "Belum Ada Judul"	*["msh", "tunggu", "mungkin", "@DnrWidianto", "nyanyi", "belum", "ada", "judul"]*

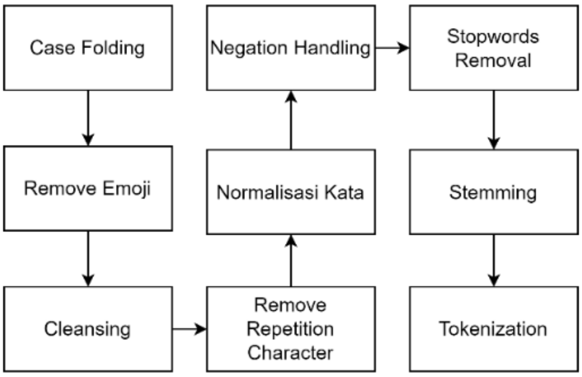

Figure 3. Pre-processing 2nd scenario

As shown in Figure 3, the pre-processing 2nd scenario was taken from sample journal articles in [3] which included case folding, cleansing, word normalization, stopwords removal, stemming, and tokenization and also added several pre-processing steps such as removing emoji, removing character repetition and negation. This is done so that the pre-processing scenario is complete. After the data is carried out in pre-processing 2nd scenario, results are shown in Table IV.

TABLE IV
PRE-PROCESSING 2ND SCENARIO

Before	After
Msh menunggu kemungkinan @DnrWidianto menyanyikan "Belum Ada Judul"	*["masih", "tunggu", "mungkin", "nyanyi", "belum", "ada", "judul"]*

D. TF-IDF Weighting

TF-IDF refers to bringing together two concepts: Term Frequency and Document frequency. Term Frequency is a concept where weighting is applied by searching for the frequency of a term appearing in the text. Document Frequency is the number of documents or texts in which a word appears. The lower the frequency of appearance, the lower the value, examples for journal articles on [12].

In calculating the Term Frequency, all the words are usually considered significant. Therefore, it is necessary to calculate the TF-IDF, where the score can be obtained using Equation (1). Equation (1) is the multiplication of several word frequencies with the number of document intervals for each word.

$$W_{t,d} = tf_{t,d} \times idf_t = tf_{t,d} \times \log \frac{N}{df_t} \qquad (1)$$

E. Support Vector Machine

The SVM aims to find the best hyperplane to separate classes. Effective separation means that the hyperplane has the maximum margin to the points the closest training of the two classes because the larger margin reduces error generalization of classifiers, examples for journal articles on [13]. SVM has proven to be one of the most powerful learning algorithms for text categorization, examples for journal articles [14].

$$f(x) = \sum_{i=1}^{m} a_i y_i K(x, x_i) + b \qquad (2)$$

The decision function using Equation (2) is the result of the sigma weight value of each data point multiplied by the class of each data multiplied by the kernel function plus the bias value. The characteristics of the SVM, as explained in the previous section, are summarized as follows, examples for journal articles on [5] :

- In principle, an SVM is a linear classifier.
- Pattern recognition is performed by transforming data in the input space to a higher dimensional space, and optimization is performed in the new vector space. This differentiates the SVM from typical pattern recognition solutions, which perform parameter optimization in the resulting transformed space with lower dimensions than the input space.
- Implementing a Structural Risk Minimization (SRM) strategy.
- The working principle of the SVM is basically only able to handle the classification of two classes.

DOI : http://dx.doi.org/10.25139/inform.v8i1.5667

This study used a sequential learning algorithm to process training data of the SVM, known as a simple algorithm, and used a short time compared to other algorithms.

F. Naïve Bayes

Naive Bayes is a collection of probabilistic classification algorithms based on the theorem of Bayes, namely, seeing opportunities in the future based on past experiences [15]. The Naive Bayes algorithm is a simple and efficient algorithm that is commonly used for text classification and data analysis. Some of the advantages of using this algorithm include [5]:

- Simplicity: Naive Bayes is a simple algorithm that is easy to implement and understand, which makes it an ideal choice for beginners or when working with limited computational resources.
- Efficiency: Naive Bayes requires minimal computational resources and is relatively fast when compared to other algorithms. This makes it an ideal choice for large-scale text data analysis.
- Low data requirements: Naive Bayes requires relatively little training data, making it an ideal choice when working with limited data.
- Good results: Naive Bayes has been found to provide very good results when used for text data analysis, as it is able to handle large amounts of data, and it can be used in many different applications, such as sentiment analysis, spam detection, and more.

As you have cited in your statement, many examples of journal articles and studies have reported the good performance of the Naive Bayes algorithm for text data analysis [14]. However, it's important to note that the choice of algorithm depends on the specific context and research goal. Other algorithms should be considered as well. This algorithm may not be scalable for large data sets [16].

$$P(m_i|n_j) = \frac{P(n_j|m_i)\,P(m_i)}{P(n_j)} \qquad (3)$$

Posterior or opportunity for class *i* when the word j appears. Equation (3) is multiplied by the Conditional probability or opportunity for word j in class *i* with Prior or opportunity for class *i* and divided by Evidence or the chance for the word to appear.

Sample journal articles in [17] look at the probability of the word appearing in the Equation. They can be eliminated because this opportunity does not affect the comparison of the classification results for each category. So the form of Equation (4). To find the Prior using Equation (5).

$$P(m_i|n_j) = P(m_i) \times P(n_j|m_i) \qquad (4)$$

$$P(m_i) = \frac{N_m}{N} \qquad (5)$$

Meanwhile, to get the Posterior value can be done by multiplication between the prior and the total conditional probability using Equation (6).

$$P(m_i|n_j) = P(m_i) \times P(n_j|m_i) \times \dots \times P(n_n|m_i) \qquad (6)$$

G. Confusion Matrix

Testing is done using the Confusion Matrix. The final result of the testing process carried out at this stage will get the value of accuracy, precision, recall, and F1-score of sentiment classification using the SVM and NB methods. A confusion matrix is a table that shows the correct classification of the number and quantity of test data [18]. In the confusion matrix, calculations are obtained to calculate accuracy, precision, recall, and the F1-score. The accuracy value using Equation (7) relates to how accurate the model is in correctly classifying the test data.

$$Acc = \frac{TPositive + TNegative + TNetral}{Total} \times 100 \qquad (7)$$

The precision value compares the correctly predicted data and the predicted data. The recall value compares the data that is correctly predicted and the correct data. In contrast, the F1-score compares the weighted average precision and recall.

III. Result and Discussion

This section analyzes the results using various classifiers with the SVM and Naive Bayes algorithms. The pre-processing scenario described in this section consists of two scenarios: 1st scenario and 2nd scenario.

Data was collected using scraping techniques from the Twitter social media put the word "XFactorID" from 2021 to 2022. The process starts by initializing variables and preparing other variables to store retrieved data [19][20]. Furthermore, data collection is carried out with the scrape library, entering the criteria and limits determined. This library is used because the Twitter API cannot reach old tweets. The data fetched is added to the storage variable. The data taken is only in the form of tweets. The retrieved data is then stored in a **.csv* file to facilitate the next process.

After data collection and labeling, the next step is pre-processing. In the pre-processing 1st scenario, the steps involved are case folding, removing emojis, stopwords removal, stemming, and tokenization. Whereas in the pre-processing 2nd scenario, the pre-processing stage passed case folding, removing emoji, cleansing, removing repetition characters, normalizing words, negation handling, stopwords removal, stemming, and tokenization. The pre-processing 2nd scenario takes a long time the process compared to the pre-processing 1st scenario. This is because there are steps to match each word into the existing dictionary.

After pre-processing 1st and 2nd scenarios, both are then carried out TF-IDF weighting, which means giving weight to each term in the data. In this study, the TF-IDF weighting

DOI : http://dx.doi.org/10.25139/inform.v8i1.5667

process was used with the TfidfVectorizer then the weight was stored in the pickle file. It is from this pickle file that will be implemented in application implementation.

The next step after weighting with TF-IDF is classification. The SVM classification uses the SVM model obtained from the scikit-learn library. This model is suitable for completing the text data classification. The model has been developed, and it will be saved in the form of a pickle file and implemented in the application implementation process. Table V compares the accuracy, precision, recall, and F1-score performance from pre-processing 1st and 2nd scenarios obtained using the SVM algorithm. The highest accuracy is shown in Table V using the pre-processing 2nd scenario.

TABLE V
CONFUSION MATRIX 1ST SCENARIO PRE-PROCESSING USING THE SVM METHOD

		Prediction			Total
		Positive	Negative	Neutral	
Actual	Positive	32	4	5	41
	Negative	8	30	14	52
	Neutral	6	9	72	87
Total		46	43	91	

Table V shows three true positives, there are 32, 30, and 72. The value of the True Positive can then be determined from the accuracy value.

TABLE VI
CONFUSION MATRIX 2ND SCENARIO PRE-PROCESSING USING SVM METHOD

		Prediction			Total
		Positive	Negative	Neutral	
Actual	Positive	32	7	2	41
	Negative	10	35	7	52
	Neutral	2	9	76	87
Total		44	51	85	

Table VI shows True Positives 32, 35, and 76. The value of the True Positive can then be determined based on its accuracy. So from Table V-VI, it can be calculated to look for accuracy, precision, recall, and F1-score.

TABLE VII
THE TEST RESULT OF THE SVM ALGORITHM

Method	Acc	Precision	Recall	F1-score
SVM 1st scenario	74,44%	72,81%	72,82%	72,52%
SVM 2nd scenario	79,44%	76,91%	76,94%	76,83%

Table VII compares the accuracy, precision, recall, and F1-score performance from pre-processing 1st and 2nd scenarios obtained using the SVM algorithm. The highest accuracy is shown in the Table VII using the pre-processing 2nd scenario.

The SVM algorithm's accuracy obtained pre-processing 1st scenario is 74.44%. While in 2nd scenario, there is an increased accuracy of 79.44%. In the 1st scenario, the precision value is 72.81%. Whereas in 2nd scenario, there is an increase in precision to 76.91%. In 1st scenario, obtained recall value is 72.82%. Whereas in 2nd scenario, there is an increase in recall to 76.94%. This is because 2nd scenario can clean data, whereas 1st scenario cannot do so. Pre-processing 2nd scenario includes case folding, cleansing, word normalization, stopwords removal, stemming, and tokenization, as well as several pre-processing steps such as removing emoji, removing repetition characters, and negation handling. Pre-processing 2nd scenario also includes case folding, cleansing, word normalization, stopword removal, and stemming. Furthermore, adding a step in pre-processing 2nd scenario, including cleansing, removing repetition characters, word normalization, and negation handling, increase the SVM classifier's accuracy.

When utilizing NB for classification, the Complement NB model was used. This model is suitable for completing the text data classification. The model has been developed, saved in the form of a pickle file, and implemented in the application's code during the implementation process.

TABLE VIII
CONFUSION MATRIX 1ST SCENARIO PRE-PROCESSING USING NB METHOD

		Prediction			Total
		Positive	Negative	Neutral	
Actual	Positive	32	4	5	41
	Negative	11	27	14	52
	Neutral	7	10	70	87
Total		50	41	89	

Table VIII shows three true positives, and there are 32, 27, and 70. The value of the True Positive can then be determined from the accuracy value.

TABLE IX
CONFUSION MATRIX 2ND SCENARIO PRE-PROCESSING USING NB METHOD

		Prediction			Total
		Positive	Negative	Neutral	
Actual	Positive	36	3	2	41
	Negative	15	28	9	52
	Netural	4	9	74	87
Total		55	40	85	

Table IX shows three true positives. There are 36, 28, and 74. The value of the True Positive can then be determined based on its accuracy. So from Table VIII-IX, it can be calculated to look for accuracy, precision, recall, and F1-score.

TABLE X
THE TEST RESULT OF THE NAIVE BAYES ALGORITHM

Method	Acc	Precision	Recall	F1-score
NB 1st scenario	71,67%	69.05%	70,13%	69,60%
NB 2nd scenario	76,67%	74.16%	75,56	73,96%

Table X compares the accuracy, precision, recall, and F1-score performance from pre-processing 1st and 2nd scenarios obtained using the NB algorithm. The highest accuracy is shown in Table X using the pre-processing 2nd scenario.

The accuracy value is obtained in the 1st NB scenario algorithm, namely 71.67%. While in 2nd scenario, there is an

DOI : http://dx.doi.org/10.25139/inform.v8i1.5667

increase to 76.67%. In 1st scenario, the precision value is 69.05%. While in 2nd scenario, there is an increase of 74.16%. In 1st scenario, the recall value is 70.13%. Meanwhile, in 2nd scenario, there is an increase of 75.56%. Pre-processing 2nd scenario includes case folding, cleansing, word normalization, stopwords removal, stemming, and tokenization, as well as several pre-processing steps such as removing emoji, removing repetition characters, and negation handling, can clean data, which cannot be done by pre-processing 1st scenario. Furthermore, adding a step pre-processing 2nd scenario, which includes cleansing, removing repetition characters, word normalization, and negation handling, increases the NB classifier's accuracy.

A complete pre-processing scenario makes it easier for the sentiment analysis system to classify comments to increase accuracy, precision, recall, and the F1-score. This is because using a complete pre-processing scenario allows the system to find the appropriate base words and still analyze negative meanings. As well as classifying with an SVM can also increase accuracy, precision, recall, and F1-score compared to NB. This is because the SVM can classify text with the ability to generalize well in high-dimensional feature space. Meanwhile, NB classifies texts by looking at opportunities from the past.

IV. Conclusion

Accuracy, precision, recall, and F1-score are the outcomes of the tests conducted using the SVM algorithm. The pre-processing 2nd scenario yielded the highest possible scores, respectively 79.44%, 76.91%, 76.94%, and 76.83%, respectively. While the NB algorithm has the best accuracy, precision, recall, and F1-score obtained from the pre-processing 2nd scenario, namely 74.16%, 75.56%, and 73.96%. So the best accuracy was obtained by the pre-processing 2nd scenario consisting of case folding, removing emoji, cleansing, removing repetition characters, normalizing words, negation handling, stopwords removal, stemming, and tokenization using the SVM algorithm. The testing results also revealed that using more pre-processing scenarios can help find the appropriate base word while also analyzing the negative meaning. Using SVM to classify text with capabilities generalizes well in high-dimensional feature spaces. In previous research, the Naive Bayes algorithm obtained the highest accuracy due to the influence of weightings, such as TF-IDF, Skipgram, and BiGram. While in this study did not use Skipgram and Bigram as a weighting process. So this is what causes the difference in accuracy results. The weakness of this study is that it does not pay attention to word order in sentiment analysis. So the results of the sentiment analysis have not been classified accurately.

For further research, comparing pre-processing scenarios on other algorithms, such as deep learning methods, is advisable. In addition, it is recommended to try to experiment with collaborating with other weights, such as the N-Gram, to provide greater accuracy of the weights.

References

[1] W. E. Nurjanah, R. S. 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. Univ. Brawijaya*, vol. 1, no. 12, pp. 1750–1757, 2017.

[2] T. F. Berlian, A. Herdiani, and W. Astuti, "Analisis Sentimen Opini Masyarakat Terhadap Acara Televisi pada Twitter dengan Retweet Analysis dan Naïve Bayes Classifier," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 8660–8669, 2019.

[3] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Comput. Math. Organ. Theory*, vol. 25, no. 3, pp. 319–335, 2019.

[4] J. Cervantes, X. Li, and W. Yu, "SVM classification for large data sets by considering models of classes distribution," *Proc. - 2007 6th Mex. Int. Conf. Artif. Intell. Spec. Sess. MICAI 2007*, pp. 51–60, 2007.

[5] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Syst.*, vol. 226, p. 107134, 2021.

[6] A. V. Vitianingsih, Z. Othman, S. S. K. Baharin, A. Suraji, and A. L. Maukar, "Application of the Synthetic Over-Sampling Method to Increase the Sensitivity of Algorithm Classification for Class Imbalance in Small Spatial Datasets," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 5, pp. 676–690, 2022.

[7] J. A. Septian, T. M. Fahrudin, and A. Nugroho, "Journal of Intelligent Systems and Computation 43," pp. 43–49, 2019.

[8] F. Anugerah and A. Djunaidy, "Improving the Performance of Repeated Character Preprocessing in Recognizing Words in the Indonesian Sentiment Classification," vol. 7, no. 9, pp. 1–9, 2017.

[9] G. A. Buntoro, "Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter," *INTEGER J. Inf. Technol.*, vol. 1, no. 1, pp. 32–41, 2017.

[10] V. S and J. R, "Text Mining: open Source Tokenization Tools – An Analysis," *Adv. Comput. Intell. An Int. J.*, vol. 3, no. 1, pp. 37–47, 2016.

[11] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 406, 2021.

DOI : http://dx.doi.org/10.25139/inform.v8i1.5667

[12] N. N. Wilim and R. S. Oetama, “Sentiment Analysis About Indonesian Lawyers Club Television Program Using K-Nearest Neighbor, Naïve Bayes Classifier, And Decision Tree,” *IJNMT (International J. New Media Technol.*, vol. 8, no. 1, pp. 50–56, 2021.

[13] R. Inglehart, “Chapter 10. From Elite-Directed To Elite-Directing Politics: The Role Of Cognitive Mobilization, Changing Gender Roles, And Changing Values,” *Cult. Shift Adv. Ind. Soc.*, pp. 335–370, 2019.

[14] A. M. Rahat, A. Kahir, and A. K. M. Masum, “Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset,” *Proc. 2019 8th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2019*, pp. 266–270, 2020.

[15] B. M. Pintoko and K. M. L., “Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naive Bayes Classifier,” *e-Proceeding Eng.*, vol. 5, no. 3, pp. 8121–8130, 2018.

[16] A. Prabhat and V. Khullar, “Sentiment classification on big data using Naïve bayes and logistic regression,” *2017 Int. Conf. Comput. Commun. Informatics, ICCCI 2017*, 2017.

[17] Imam Fahrur Rozi, Imam Fahrur Rozi, and Muhammad Balya Iqbal Alfahmi, “PENGEMBANGAN APLIKASI ANALISIS SENTIMEN TWITTER MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER (Studi Kasus SAMSAT Kota Malang),” *J. Inform. Polinema*, pp. 149–154, 2018.

[18] D. Normawati and S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” *J. Sains Komput. Inform.*, vol. 5, no. 2, pp. 697–711, 2021.

[19] M. Z. Sarwani and D. A. Sani, “Social Media Analysis Using Probabilistic Neural Network Algorithm to Know Personality Traits,” *J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 6, no. 1, pp. 61–64, 2021.

[20] M. Z. Sarwani, D. A. Sani, and F. C. Fakhrini, “Personality Classification through Social Media Using Probabilistic Neural Network Algorithms,” *Int. J. Artif. Intell. Robot.*, vol. 1, no. 1, p. 9, 2019.

DOI : http://dx.doi.org/10.25139/inform.v8i1.5667