SENTIMENT ANALYSIS OF YOUTUBE COMMENTS ABOUT INDONESIAN LGBT USING SUPPORT VECTOR MACHINE AND NAÏVE BAYES ALGORITHMS

[1]Hapsari Ratri Sasodro, [2]Yonathan Purbo Santosa
[1,2]Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Katolik Soegijapranata
[2]yonathansantosa@unika.ac.id

ABSTRACT

YouTube is a social media that is widely used by content creators to publish their work, including LGBT content. Because of this content, many viewers end up expressing their opinions through comments. This research aims to see which is the best algorithm between Support Vector Machine using two kernels, linear kernel and RBF kernel or Naive Bayes using multinomial naive bayes seen from confusion matrix. Also, to see which pre-processing is best used for sentiment analysis by dividing pre-processing into several parts. Support Vector Machine using RBF kernel is the best algorithm in this research with 77% accuracy with precision for sentiment -1 74%, recall 72% and f1-score 72%. For sentiment 0, 70% for precision, 81% for recall, and 75% for f1-score. And the last, for sentiment 1, with 90% precision, 77% recall and 83% f1-score. In addition, pre-processing using stemming-tokenizing is the best pre-processing used for sentiment analysis in this research based on the highest average number.

Keywords: Sentiment Analysis, YouTube, LGBT, Support Vector Machine, Naïve Bayes.

BACKGROUND

Technological advances that are happening right now have a lot of impact on everyday life, whether it's a positive or negative impact. One of the advancements in technology that is now widely used by many people is social media. Many types of social media are used, including Facebook, Instagram, Twitter, Telegram, YouTube and many more. Social media helps many people to connect with one another, but not infrequently social media is also used for bad things that are detrimental.

YouTube is a social media that is widely used by content creators to publish their work. Starting from music, news, daily life, tutorials, and more. As with other social media where users are free to express themselves, many content creators ultimately do not hesitate to reveal some of their personal things, one of which is their sexual orientation, whether they are lesbian, gay, bisexual or transgender.

Because of the content, many audiences also end up expressing their opinions through comments. There are those who firmly reject this sexual orientation, there are those who support it, and there are also those who do not support or reject it, in other words, they are neutral parties.

Therefore, this research was conducted with the aim of analyzing comments on YouTube about LGBT people in Indonesia through sentiment analysis using the Support Vector Machine and Naïve Bayes methods algorithm.

LITERATURE STUDY

Giovani, Angelina Puput, et al [1]. The authors see that at this time, a lot of e-learning is used for learning and one of them is Ruang Guru. Therefore, research is carried out to see if an application is successful or not. The research used is sentiment analysis by taking data from comments on Twitter social media. A total of 513 tweets were obtained, then data cleaning was carried out and received positive sentiments of 338 tweets and negative sentiments of 175 tweets. The data was extracted using the Naive Bayes (NB) algorithm, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), and feature selection with the Particle Swarm Optimization (PSO) algorithm. This study compares the NB, SVM, K-NN methods without using feature selection with the NB, SVM, K-NN methods that use feature selection and compares the Area Under Curve (AUC) values of these methods to find out the most optimal algorithm. The results obtained from this study are that the SVM (PSO) algorithm has the highest accuracy and performance values when compared to NB, SVM, K-NN, NB (PSO), SVM (PSO), and K-NN (PSO). While the author's evaluation of this research is that the data used only takes from one source, namely Twitter, so in future research it can be a reference for using different data sources. I will use this research as a reference for my future research.

Fanissa, et al [2]. This study was conducted to analyze public reviews on TripAdvisor on tourism in the city of Malang using sentiment analysis and classifying it into two classes, namely positive or negative. In this research, the author uses the Naive Bayes method with Query Expansion Ranking feature selection to reduce the number of features in the classification process. The data used in this study is the amount of training data and test data, namely 200 training data (100 positive data and 100 negative data) and 30 test data documents. The process of sentiment analysis consists of preprocessing, feature selection using the Query Expansion Ranking method, and classification using Naive Bayes. From the test results, the Query Expansion Ranking algorithm produces the highest accuracy of 86.6 at 75% feature selection. The author's evaluation of this research is for improvement, it is recommended for further research to pay more attention to ambiguous words, abbreviations, compound words and sarcastic sentences. I will use this research as a reference for my future research.

Ernawati, et al [3]. The internet has a lot of influence on people, especially when they have positive or negative opinions, so in this study, these comments are analyzed, especially comments about travel agents. The research was conducted by classifying opinions by analyzing sentiment through a text mining approach, and requires a method that is able to classify opinions accurately and the authors choose to use the K-NN algorithm. The data used for this study were obtained from comments found on the https://www.trustpilot.com/categories/travel_holidays page. The scope of this research is a review of travel agents processing data using the K-Nearest Neighbor

(K-NN) algorithm which uses 100 positive reviews and 100 negative reviews with six words related to sentiment, namely: Fast, Good, Great, Buruk, Cancel, and Tunggu. From the results of research conducted, the K-NN algorithm gets an accuracy value of 87.00%.

Rahman Isnain, et al [4]. The author conducted this research by looking at the many comments on social media about the Lockdown policy carried out by the Jakarta government. The author conducts a sentiment analysis of 2000 comments in Indonesian language contained on Twitter social media using the Support Vector Machine method with Tf-Idf feature extraction. With a test that will later see how the values of accuracy, precision, recall and F1-Score will be. Researchers found that by using the SVM algorithm, it resulted in better accuracy, and could also be developed by combining the SVM method with the Firefly method as an optimization method to get more accurate results.

Hikmawan, et al [5]. The author conducts an analysis of public sentiment about government policies with the keywords "'Jokowi" and "covid" on Twitter social media. The method that I use is the Gata Framework which is used for preprocessing, and Rapidminer is also used to analyze and compare three classification methods, namely Naive Bayes, Support Vector Machine, and K-NN. From this study it can be concluded that using the SVM algorithm produces the best accuracy among the other 2 algorithms. And in the future, larger and more complex datasets are needed as well as preprocessing improvements for non-standard Indonesian.

Ratino, et al [6]. The author conducts a sentiment analysis to find out the sentiment of every comment on Instagram social media towards COVID-19 information. The methods used are Support Vector Machine and Naïve Bayes with the addition of Particle Swarm Optimization (PSO). The data that the author uses are obtained from comments on Instagram social media. The results of this study indicate that using the Support Vector Machine with the addition of Particle Swarm Optimization has a higher accuracy than the Naive Bayes algorithm. And as for the evaluation, you can use the K-NN algorithm or Decision Tree to test the algorithm more, and other Feature Selection methods can also be used, such as Chi Square, Information Gain, Luxicon Based Feature, and others in order to compare the results.

Maureen Pudjajana, et al [7]. The author conducts a sentiment analysis about pornography and the existence of homosexuals in Indonesia through comments on Twitter. The twitter data was analyzed by sentiment analysis as text mining using the Naïve Bayes method. With this research, it can be seen that the results of positive and negative sentiments on tweet test data and based on the results of these tests can be conveyed to Twitter users at large to use Twitter appropriately. In addition, the Naive Bayes calculation is compared with k-Nearest Neighbor (k-NN) to determine the level of accuracy. The results obtained from this study are that Naive Bayes has better accuracy after a comparison with the k-NN algorithm with negative sentiment dominating over positive sentiment. With this research, I became interested in researching this topic with different problems later.

Najiyah, et al [8]. The author conducts a sentiment analysis about covid-19 uploaded to social media facebook, twitter, and instagram by dividing it into 3 classes, namely positive, negative and neutral. The dataset used is a collection of comments on Facebook, Twitter, and Instagram, totaling 1177 datasets with the distribution of 560 positive datasets, 355 negative datasets and 262 neutral datasets. The method used is the Probabilistic Neural network classification method. Before doing the classification, the preprocessing in this study includes tokenization, normalization, removing emoticons, Convert Negation, Stopword removal and TF-IDF using the python language with several libraries such as keras, tensor flow and pandas. The results obtained from this study are the use of this method turned out to produce better results than the method used by the previous author.

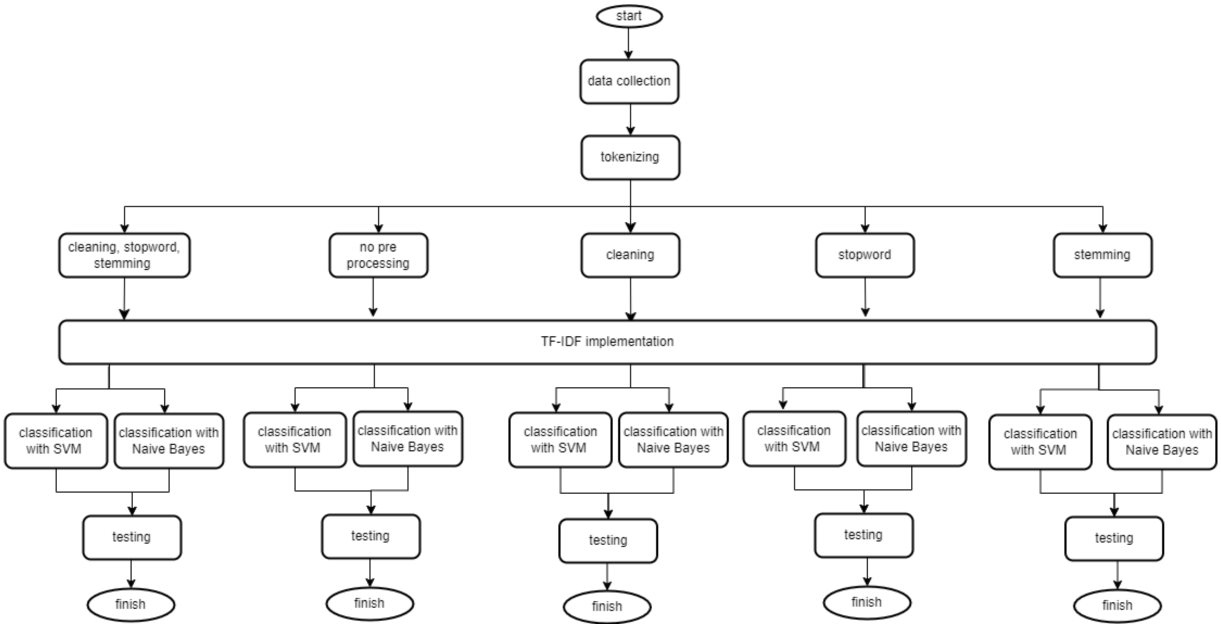
Sari, et al [9]. The author looks for the best decision for tourist attraction visitors using sentiment analysis by looking at previous visitor reviews. This study uses the K-Nearest Neighbor method with preprocessing used, namely tokenize, stopword filter and bi-gram. The data used are 50 positive reviews and 50 negative reviews originating from the review site www.tripadvisor.com. This study produces an accuracy that is not too large, less than 80%. And the evaluation from the author is that you will get greater accuracy if you use several algorithms such as SVM.

Siringoringo, et al [10]. The author conducts a sentiment analysis for the Denim brand of Trendy Shoes. This research has several stages, starting from data collection, initial processing, data transformation, feature selection and classification stages using the Support Vector Machine. The dataset is obtained from several online buying and selling sites in Indonesia. The results of this study, it is found that the method that has better performance is SVM when compared to k-NN.

Based on the literature study above, this research will use literature [4] and [7] as the basis of research. The basis of why literature [4] and [7] are the basis of research is that the dataset is obtained from the same source, twitter. With the same form of comment writing, it can be considered that the datasets from the two literatures are the same. In addition, the topic raised by literature [4], homosexuality, has a correlation with the topic that will be raised by this research, that is LGBT. The dataset that will be used in this research is taken from comments on YouTube. By comparing the two dataset sources, it is concluded that comments about homosexuals on twitter and LGBT on YouTube have the same form of writing, so the dataset is considered the same. On this basis, this research will conduct sentiment analysis by comparing two algorithms, namely Support Vector Machine uses two types of kernels to find out the final results, which are linear kernels and RBF kernels [7] and Naïve Bayes use multinomial naïve bayes [4] to see which is the best algorithm based on the results of the confusion matrix (precision, recall, f1-score and accuracy) and analyze which pre-processing is best used for sentiment analysis using the Support Vector Machine and Naïve Bayes algorithms.

RESEARCH METHODOLOGY

The research methodology of the research is presented in Gambar 1



Gambar 1. Research methodology

In this research, the first step is to collect the dataset. The dataset is collected from kaggle, which is a commentary on YouTube. After doing data collection, there will be two methods, which are classification by using data pre-processing such as data cleaning, tokenizing, and steaming at the same time and splitting pre-processing into four parts, tokenizing which is a must step, cleaning-tokenizing, stopword-tokenizing and stemming-tokenizing. After obtaining the pre-processed data, the next step is to implement TF-IDF, which converts text into a usable vector. Then, classification is carried out using the Support Vector Machine and Naïve Bayes algorithms and the last is testing to see the final results of the classification.

Data collection

Dataset collection in this research is made by collecting datasets from the kaggle website https://www.kaggle.com/datasets/edwinaayuchristy/youtube-comments-on-lgbt-in-indonesia. In addition, the dataset is also taken manually or copy the comments on some YouTube content that contains content about LGBT to Microsoft Excel and then manually labeled whether the comment is positive, negative or neutral by using labeling 1 for positive, -1 for negative and 0 for neutral. The number of datasets obtained is 4000 data with the total number of positive comments 774, negative comments 1534 and neutral comments 1692. After that, the data that was previously in the form of xlsx was converted into csv so that it could be processed by the program.

Pre-processing

Data cleaning

Data cleaning is the process of cleaning comments from things that are not really needed in research. In this research, the cleaning process carried out is cleaning hashtags, next characters, punctuation marks, extra whitespace, and converting to lowercase. Cleaning in this research is done so that the data processing that is carried out becomes more optimal.

Tokenizing

Tokenizing is the process of separating text into chunks called tokens for analysis, in other words tokenizing is done to make it easier for computers to read text in any form. In this research, tokenizing is done by separating words per word in comments by using the nltk module to perform tokenizing and also using RegexpTokinizer (Regular expression) to control text token generation.

In this research, the tokenizing process is very important, because without this tokenizing process, the data that will go to the next process cannot be used because the computer does not understand or cannot capture the meaning of the inputted text. Therefore, in this research in finding the best pre-processing for the analysis sentiment, tokenizing must be included in every pre-processing.

Filtering(stopword removal)

Stopwords are common words that usually appear in large numbers and have no meaning, for example "yang", "dan", "di", "dari", etc. The purpose of stopwords is to remove words that have low information and only focus on words that are important instead. Filtering is done so that words contained in the stopword list are not analyzed and can get maximum accuracy. This research uses the nltk module to download stopwords. By using the nltk.corpus module, previously downloaded stopwords are imported to be used to create a stopword list in Indonesian.

Stemming sastrawi

Stemming is the process of reducing words to their basic or original form. Since the data used in this research is in Indonesian, stemming must use the Python Sastrawi library. Python Sastrawi library is a python library to reduce words in Indonesian to their basic or original form and the algorithm applied is the Nazief and Adriani algorithm.

Before stemming, Sastrawi is installed first. The version used is Sastrawi-1.0.1. Next, the stemming step is carried out using StemmerFactory which is imported from the sastrawi package that has been downloaded previously.

Implementation TF-IDF

Term frequency-inverse document frequency is a text vectorizer that combines two concepts, Term Frequency (TF) and Document Frequency (DF) which are used to convert text into vectors for easy use. Term frequency is the number of occurrences of important terms in a document. TF will show how important a term is in a document and also represents each text of the data which will become a matrix with its rows being the number of documents and its columns being the number of different terms in all documents.

Document frequency is the number of documents that contain a particular term or in other words this document frequency shows how common the term is. Inverse document frequency is the weight of a term, with the aim of reducing the weight of a term if the occurrence of the term is spread throughout the document.

In this research, the implementation of TF-IDF is using the library from sklearn, namely sklearn.feature_extraxtion by importing TFidfVectorized to implement TF-IDF. The parameter used for TF-IDF implementation is text_string which is a string form of text that has gone through the stemming process.

Classification with Support Vector Machine

The first thing to do before performing classification in this study is to split the dataset into two, which are training data and test data, with a distribution of 90% for training data and 10% for test data. This step is needed to avoid overfitting, overfitting is a situation where the machine learning model fits the trainer data too well so that the adjustment with additional reliable data fails.

After the data is split, the next thing to do is to declare the variable "svm" as a data model that uses a Support Vector Classifier or SVC. In this research, modeling uses two types of kernels, which are the linear kernel and the RBF (Gaussian Radial Basis Function) kernel. After data modeling is done, the next thing to do is prediction using a confusion matrix by utilizing sklearn, namely sklearn_metrics to import classification_report and confusion_matrix.

Classification with Naïve Bayes

The same thing that is done in SVM, in this Naïve Bayes algorithm, the first thing to do is to split the dataset into two parts, which are train and test data. With the same proportion, which is 90% for train data and 10% for test data. In this research, multinomial naïve bayes are the method used to perform classification. Modeling is done using sklearn, namely MultinomialNB() which is declared in the "nb" variable. The next step is to predict using a confusion matrix by utilizing sklearn, namely sklearn_metrics to import classification_report and confusion_matrix.

Testing

In this testing step, what is to calculate the accuracy, recall, precision and f1-score values. Which is obtained from splitting the dataset and classification results using the Support Vector Machine and Naïve Bayes algorithms.

The testing carried out using the confusion matrix method is as below:

1. *Accuracy*

   It means the ratio of correct predictions (positive and negative) to the overall data. Accuracy itself can be calculated by :

$$\frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{1}$$

According to function (1), TP is True positive, TN is True Negative while FP is False Positive and FN is False Negative and the results are multiplied by 100 in order to get the total percentage.

2. *Recall*

Recall is the ratio of true positive predictions compared to the overall true positive data. Recall can be calculated by :

$$\frac{TP}{TP + FN} \times 100 \tag{2}$$

According to function (2), TP is True positive, while FN is False Negative and the results are multiplied by 100 in order to get the total percentage.

3. *Precision*

Precision is the ratio of true positive predictions compared to the overall predicted true positive results. Precision can be calculated by :

$$\frac{TP}{FP + TP} \times 100 \tag{3}$$

According to function (3), TP is True positive, while FP is False Positive. These results are multiplied by 100 in order to get the total percentage.

4. *F1 - score*

It is a weighted average comparison of precision and recall. F1 - score can be calculated using function (4):

$$\frac{2 \times (recall \times precision)}{(recall + presicion)} \tag{4}$$

RESULT

Determining the best algorithm

After going through the process starting from pre-processing, then the application of TF-IDF, continued with the classification algorithm until it ends in testing, the results are obtained as written in the table below.

Tabel 1. The results of Support Vector Machine testing use a linear kernel

Sentiment	Precision	Recall	f1-score	

-1	70%	72%	71%	
0	71%	60%	65%	
1	72%	84%	77%	
accuracy				**71%**

Based on tabel 1. it can be seen that the accuracy of SVM using linear kernels is 71% with precision for sentiment -1 70%, recall 72% and f1-score 71%. For sentiment 0, 71% for precision, 60% for recall, and 65% for f1-score. And the last, for sentiment 1, with 72% precision, 84% recall and 77% f1-score.

Tabel 2. The results of Support Vector Machine testing use a RBF kernel

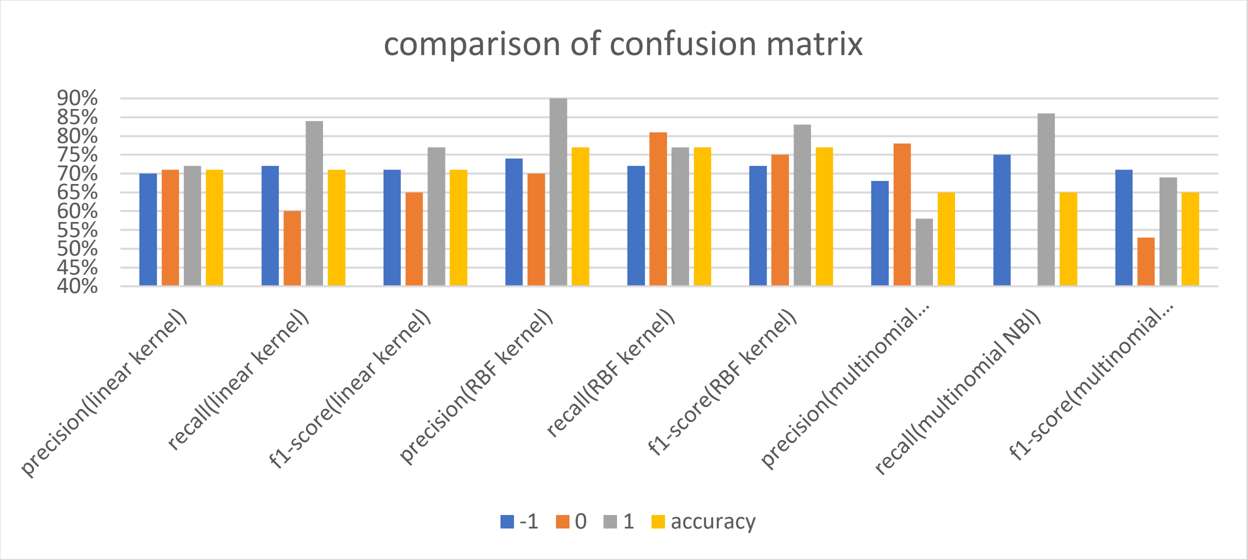
Sentiment	**Precision**	**Recall**	**f1-score**	
-1	74%	72%	72%	
0	70%	81%	75%	
1	90%	77%	83%	
accuracy				**77%**

Based on tabel 2. it can be seen that the accuracy of SVM using the RBF kernel is 77% with precision for sentiment -1 74%, recall 72% and f1-score 72%. For sentiment 0, precision 70%, recall 81%, and f1-score 75%. And finally, for sentiment 1, with 90% precision, 77% recall and 83% f1-score.

Tabel 3. The result of multinomial Naïve Bayes

Sentiment	**precision**	**Recall**	**f1-score**	
-1	68%	75%	71%	
0	78%	40%	53%	
1	58%	86%	69%	
accuracy				**65%**

Based on tabel 3. it can be seen that the accuracy of multinomial naïve bayes is 65% with precision for sentiment -1 68%, recall 75% and f1-score 71%. For sentiment 0, precision 78%, recall 48%, and f1-score 53%. And the last, for sentiment 1, with 58% precision, 86% recall and 69% f1-score.



Gambar 2. Comparison of confusion matrix

Determining the best pre-processing

Tokenizing (no pre-processing)

Tabel 4. Results of tokenizing pre-processing testing using a linear kernel SVM

Sentiment	precision	Recall	f1-score	
-1	71%	72%	72%	
0	72%	65%	68%	
1	78%	86%	82%	
accuracy				**74%**

Based on tabel 4. it can be seen that the accuracy of SVM using linear kernels is 74% with precision for sentiment -1 71%, recall 72% and f1-score 72%. For sentiment 0, precision 72%, recall 65%, and f1-score 68%. And the last, for sentiment 1, with 78% precision, 86% recall and 82% f1-score.

Tabel 5. Results of tokenizing pre-processing testing using a RBF kernel SVM

Sentiment	precision	Recall	f1-score	
-1	72%	69%	71%	
0	70%	81%	75%	
1	76%	79%	86%	
accuracy				**77%**

Based on tabel 5. it can be seen that the accuracy of SVM using the RBF kernel is 77% with precision for sentiment -1 72%, recall 69% and f1-score 71%. For sentiment 0, precision 70%, recall 81%, and f1-score 75%. And the last, for sentiment 1, with 76% precision, 79% recall and 86% f1-score.

Tabel 6. Results of tokenizing pre-processing testing using multinomial naïve bayes

Sentiment	**precision**	**Recall**	**f1-score**	
-1	73%	76%	75%	
0	78%	31%	45%	
1	55%	92%	69%	
accuracy				**64%**

Based on tabel 6. it can be seen that the accuracy of multinomial naïve bayes is 64% with precision for sentiment -1 73%, recall 76% and f1-score 75%. For sentiment 0, the precision is 78%, recall is 31%, and f1-score is 45%. And the last, for sentiment 1, with 55% precision, 99% recall and 69% f1-score.

Tabel 7. Results of tokenizing pre-processing testing using a sigmoid kernel SVM

Sentiment	**precision**	**Recall**	**f1-score**	
-1	69%	71%	70%	
0	67%	62%	64%	
1	72%	77%	75%	
accuracy				**69%**

Based on tabel 7. it can be seen that the accuracy of SVM using sigmoid kernels is 69% with precision for sentiment -1 69%, recall 71% and f1-score 70%. For sentiment 0, precision 67%, recall 62%, and f1-score 64%. And the last, for sentiment 1, with 72% precision, 77% recall and 75% f1-score.

Tabel 8. Results of tokenizing pre-processing testing using a polynomial kernel SVM

Sentiment	**precision**	**Recall**	**f1-score**	
-1	75%	62%	68%	
0	67%	84%	74%	
1	96%	81%	87%	
accuracy				**76%**

Based on tabel 8. it can be seen that the accuracy of SVM using polynomial kernels is 76% with precision for sentiment -1 75%, recall 62% and f1-score 68%. For sentiment 0, precision 67%, recall 84%, and f1-score 74%. And the last, for sentiment 1, with 96% precision, 81% recall and 87% f1-score.

Cleaning-tokenizing

Tabel 9. Results of cleaning-tokenizing pre-processing testing using a linear kernel SVM

Sentiment	**precision**	**recall**	**f1-score**	
-1	69%	69%	69%	
0	71%	65%	68%	
1	76%	83%	79%	
accuracy				**72%**

Based on tabel 9. it can be seen that the accuracy of pre-processing, cleaning and tokenizing SVM using linear kernel is 72% with precision for sentiment -1 69%, recall 69% and f1-score 69%. For sentiment 0, 71% precision, 65% recall, and 68% f1-score. And the last, for sentiment 1, with 76% precision, 83% recall and 79% f1-score.

Tabel 10. Results of cleaning-tokenizing pre-processing testing using a RBF kernel SVM

Sentiment	**precision**	**recall**	**f1-score**	
-1	72%	71%	72%	
0	68%	79%	73%	
1	92%	75%	83%	
accuracy				**75%**

Based on tabel 10. it can be seen that the accuracy of pre-processing, cleaning and tokenizing SVM using RBF kernel is 75% with precision for sentiment -1 72%, recall 71% and f1-score 72%. For sentiment 0, precision 68%, recall 79%, and f1-score 73%. And the last, for sentiment 1, with 92% precision, 75% recall and 83% f1-score.

Tabel 11. Results of cleaning-tokenizing pre-processing testing using multinomial naïve bayes

Sentiment	**precision**	**recall**	**f1-score**	
-1	74%	76%	75%	
0	82%	32%	46%	
1	55%	92%	69%	
accuracy				**65%**

Based on tabel 11. it can be seen that the accuracy of pre-processing, cleaning and tokenizing using multinomial naïve bayes is 65% with precision for sentiment -1 74%, recall 76% and f1-score 75%. For sentiment 0, the precision is 82%, recall is 32%, and f1-score is 46%. And the last, for sentiment 1, with 55% precision, 92% recall and 69% f1-score.

Tabel 12. Results of cleaning-tokenizing pre-processing testing using a sigmoid kernel SVM

Sentiment	**precision**	**Recall**	**f1-score**	
-1	69%	70%	69%	
0	72%	61%	66%	
1	73%	84%	78%	
accuracy				**71%**

Based on tabel 12. it can be seen that the accuracy of SVM using sigmoid kernels is 71% with precision for sentiment -1 69%, recall 70% and f1-score 69%. For sentiment 0, precision 72%, recall 61%, and f1-score 66%. And the last, for sentiment 1, with 73% precision, 84% recall and 78% f1-score

Tabel 13. Results of cleaning-tokenizing pre-processing testing using a polynomial kernel SVM

Sentiment	**precision**	**Recall**	**f1-score**	
-1	77%	67%	71%	
0	69%	83%	76%	
1	96%	86%	90%	
accuracy				**79%**

Based on tabel 13. it can be seen that the accuracy of SVM using polynomial kernels is 79% with precision for sentiment -1 77%, recall 67% and f1-score 71%. For sentiment 0, precision 69%, recall 83%, and f1-score 76%. And the last, for sentiment 1, with 96% precision, 86% recall and 90% f1-score.

Stopword-tokenizing

Tabel 14. Results of stopword-tokenizing pre-processing testing using a linear kernel SVM

Sentiment	**precision**	**recall**	**f1-score**	
-1	70%	71%	70%	

0	69%	65%	67%	
1	78%	82%	80%	
accuracy				**72%**

Based on tabel 14. it can be seen that the accuracy of stopword pre-processing and SVM tokenizing using a linear kernel is 72% with precision for sentiment -1 70%, recall 71% and f1-score 70%. For sentiment 0, 69% precision, 65% recall, and 67% f1-score. And the last, for sentiment 1, with 78% precision, 82% recall and 80% f1-score.

Tabel 15. Results of stopword-tokenizing pre-processing testing using a RBF kernel SVM

Sentiment	**precision**	**recall**	**f1-score**	
-1	72%	72%	72%	
0	69%	82%	75%	
1	96%	74%	84%	
accuracy				**76%**

Based on tabel 15. it can be seen that the accuracy of stopword pre-processing and SVM tokenizing using the RBF kernel is 76% with precision for sentiment -1 72%, recall 72% and f1-score 72%. For sentiment 0, 69% precision, 82% recall, and 75% f1-score. And the last, for sentiment 1, with 96% precision, 74% recall and 84% f1-score.

Tabel 16. Results of stopword-tokenizing pre-processing testing using a multinomial naïve bayes

Sentiment	**precision**	**recall**	**f1-score**	
-1	72%	82%	76%	
0	79%	39%	52%	
1	58%	86%	70%	
accuracy				**67%**

Based on tabel 16. it can be seen that the accuracy of pre-processing stopwords and tokenizing using multinomial naïve bayes is 67% with precision for sentiment -1 72%, recall 82% and f1-score 76%. For sentiment 0, the precision is 79%, recall is 39%, and f1-score is 52%. And the last, for sentiment 1, with 58% precision, 86% recall and 70% f1-score.

Tabel 17. Results of stopword-tokenizing pre-processing testing using a sigmoid kernel SVM

Sentiment	**precision**	**Recall**	**f1-score**	

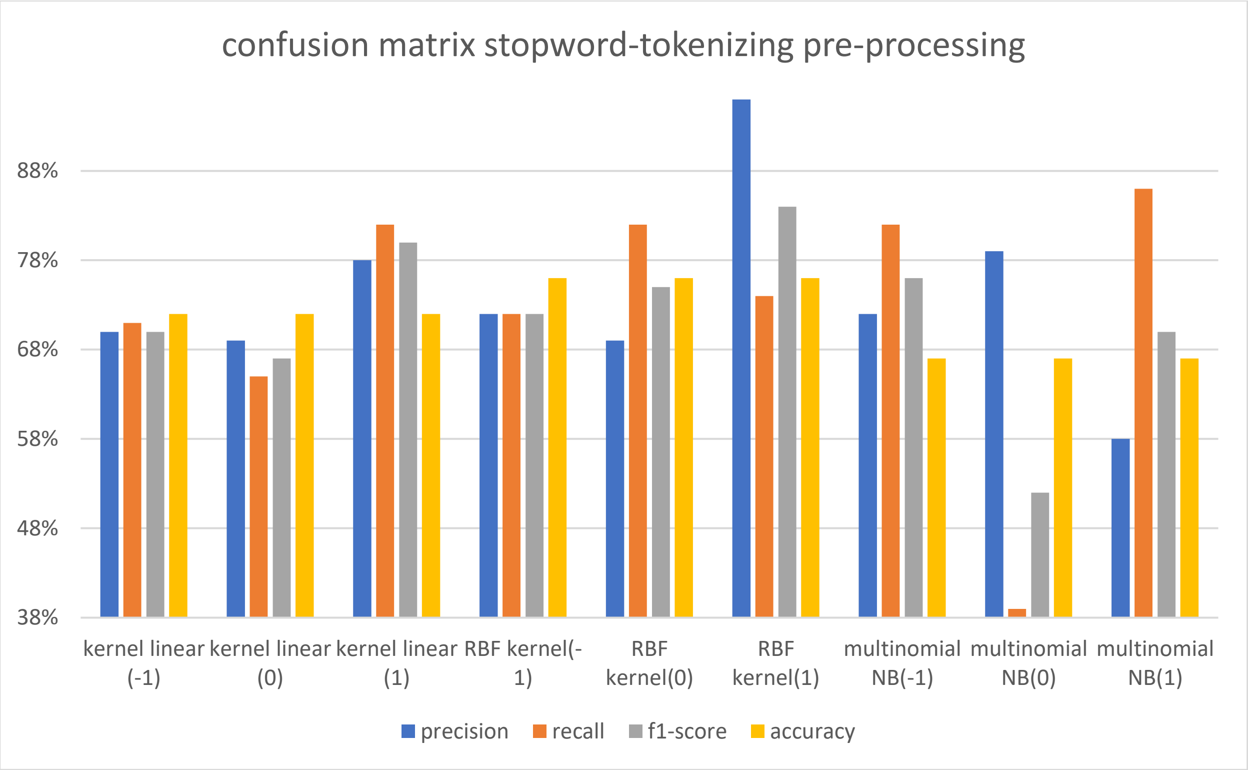
-1	69%	68%	69%	
0	65%	63%	64%	
1	74%	77%	75%	
accuracy				**69%**

Based on tabel 17. it can be seen that the accuracy of SVM using sigmoid kernels is 69% with precision for sentiment -1 69%, recall 68% and f1-score 69%. For sentiment 0, precision 65%, recall 63%, and f1-score 64%. And the last, for sentiment 1, with 74% precision, 77% recall and 75% f1-score.

Tabel 18. Results of stopword-tokenizing pre-processing testing using a polynomial kernel SVM

Sentiment	**precision**	**Recall**	**f1-score**	
-1	74%	61%	67%	
0	66%	84%	74%	
1	96%	82%	88%	
accuracy				**76%**

Based on tabel 18. it can be seen that the accuracy of SVM using polynomial kernels is 76% with precision for sentiment -1 74%, recall 61% and f1-score 67%. For sentiment 0, precision 66%, recall 84%, and f1-score 74%. And the last, for sentiment 1, with 96% precision, 82% recall and 88% f1-score.



Gambar 3. Confusion matrix stopword-tokenizing pre-processing

Stemming-tokenizing

Tabel 19. Results of stemming-tokenizing pre-processing testing using a linear kernel SVM

Sentiment	**precision**	**recall**	**f1-score**	
-1	69%	72%	71%	
0	73%	64%	68%	
1	75%	83%	79%	
accuracy				**72%**

Based on tabel 19. it can be seen that the accuracy of SVM pre-processing stemming and tokenizing using a linear kernel is 72% with precision for sentiment -1 69%, recall 72% and f1-score 71%. For sentiment 0, 73% precision, 64% recall, and 68% f1-score. And lastly, for sentiment 1, with 75% precision, 83% recall and 79% f1-score.

Tabel 20. Results of stemming-tokenizing pre-processing testing using a RBF kernel SVM

Sentiment	**precision**	**recall**	**f1-score**	
-1	74%	71%	73%	
0	68%	82%	74%	
1	92%	75%	83%	
accuracy				**76%**

Based on tabel 20. it can be seen that the accuracy of SVM pre-processing stemming and tokenizing using the RBF kernel is 76% with precision for sentiment -1 74%, recall 71% and f1-score 73%. For sentiment 0, the precision is 68%, recall is 82%, and f1-score is 74%. And lastly, for sentiment 1, with 92% precision, 75% recall and 83% f1-score.

Tabel 21. Results of stemming-tokenizing pre-processing testing using a multinomial naïve bayes

Sentiment	**precision**	**recall**	**f1-score**	
-1	71%	76%	74%	
0	84%	40%	54%	
1	57%	90%	70%	
accuracy				**67%**

Based on tabel 21. it can be seen that the accuracy of pre-processing stemming and tokenizing using multinomial naïve bayes is 67% with precision for sentiment -1 71%, recall 76% and f1-score 74%. For sentiment 0, the precision is 84%, recall is 40%, and f1-score is 54%. And the last, for sentiment 1, with 57% precision, 90% recall and 70% f1-score.

Tabel 22. Results of stemming-tokenizing pre-processing testing using a sigmoid kernel SVM

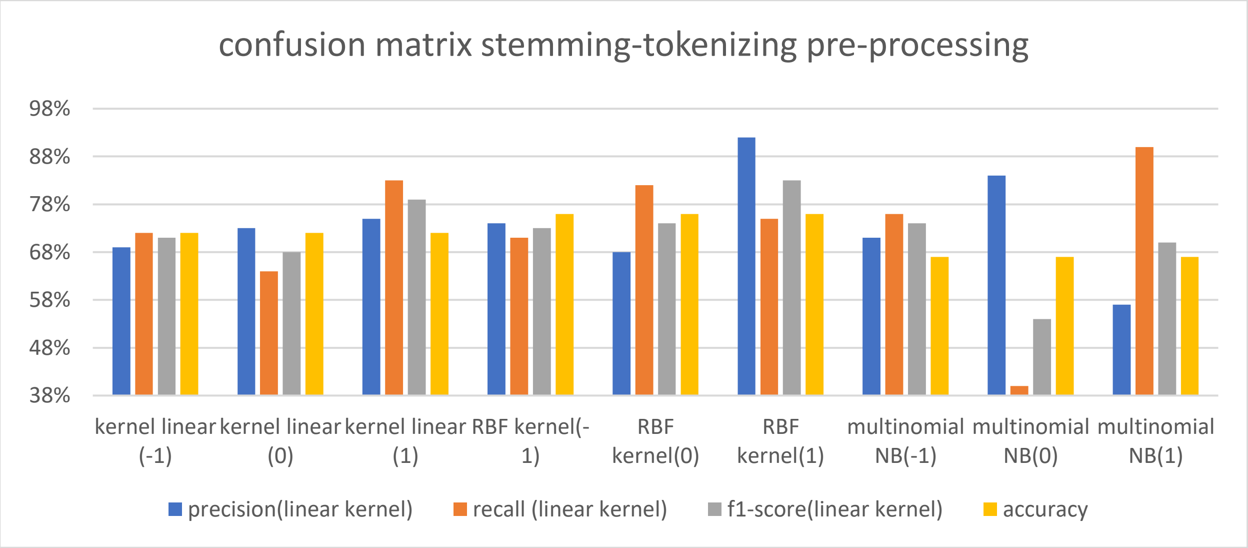
Sentiment	**precision**	**Recall**	**f1-score**	
-1	67%	72%	69%	
0	73%	64%	68%	
1	73%	79%	76%	
accuracy				**71%**

Based on tabel 22. it can be seen that the accuracy of SVM using sigmoid kernels is 71% with precision for sentiment -1 67%, recall 72% and f1-score 69%. For sentiment 0, precision 73%, recall 64%, and f1-score 69%. And the last, for sentiment 1, with 73% precision, 79% recall and 76% f1-score.

Tabel 23. Results of stemming-tokenizing pre-processing testing using a polynomial kernel SVM

Sentiment	precision	Recall	f1-score	
-1	80%	58%	67%	
0	64%	86%	73%	
1	96%	81%	87%	
accuracy				76%

Based on tabel 23. it can be seen that the accuracy of SVM using polynomial kernels is 76% with precision for sentiment -1 80%, recall 58% and f1-score 67%. For sentiment 0, precision 64%, recall 86%, and f1-score 73%. And the last, for sentiment 1, with 96% precision, 81% recall and 87% f1-score.



Gambar 4. Confusion matrix stemming-tokenizing pre-processing

Tabel 24. Average accuracy per pre processing

Pre processing	Accuracy
Tokenizing (no pre-processing)	71,67%
Cleaning and tokenizing	70,67%
Stopword and tokenizing	71,67%
Stemming and tokenizing	71,67%

Based on tabel 4. it can be seen the average of the three accuracies of linear SVM kernel, SVM kernel RBF, and multinomial naïve bayes from each pre-processing. This average accuracy will be used to determine which pre-processing is best for performing sentiment analysis in this research.

Tabel 25. Average precision per pre processing

Pre processing	Precision
Tokenizing (no pre-processing)	71,67%
Cleaning and tokenizing	73,22%
Stopword and tokenizing	73,67%
Stemming and tokenizing	73,67%

Based on tabel 5. it can be seen the average precision of linear SVM kernel, SVM kernel RBF, and multinomial naïve bayes from each pre-processing. This average precision will be used to determine which pre-processing is best for performing sentiment analysis in this research.

Tabel 26. Average recall per pre processing

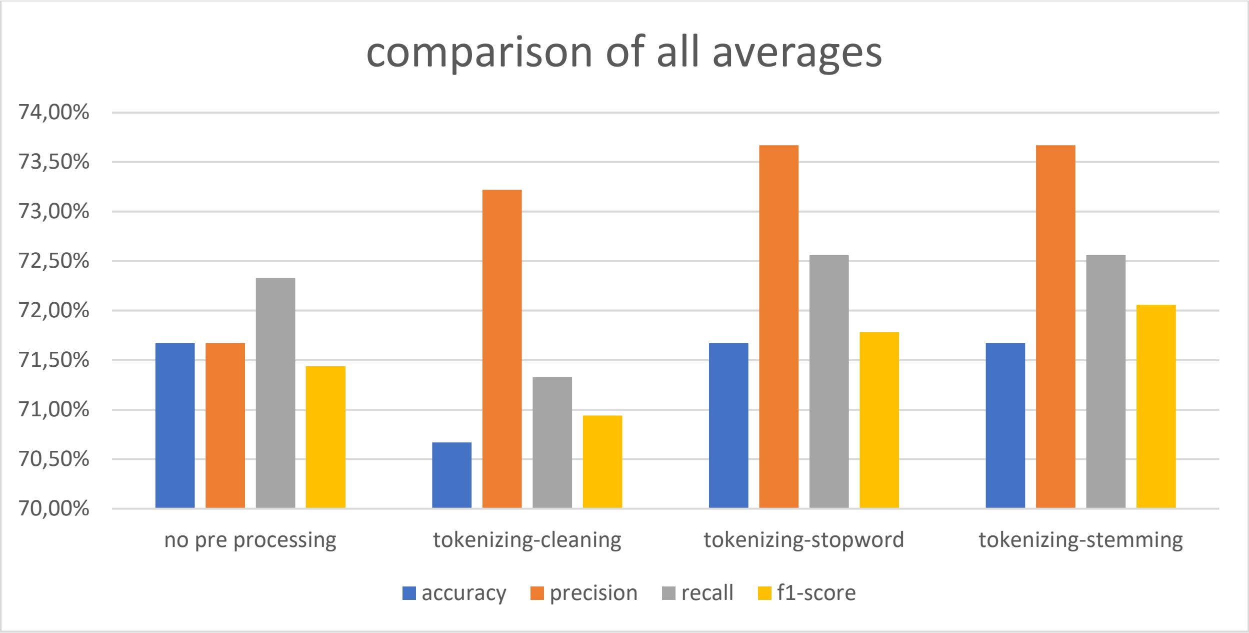
Pre processing	Recall
Tokenizing (no pre-processing)	72,33%
Cleaning and tokenizing	71,33%
Stopword and tokenizing	72,56%
Stemming and tokenizing	72,56%

Based on tabel 6. it can be seen the average recall of linear SVM kernel, SVM kernel RBF, and multinomial naïve bayes from each pre-processing. This average recall will be used to determine which pre-processing is best for performing sentiment analysis in this research.

Tabel 27. Average f1-score per pre processing

Pre processing	F1-score
Tokenizing (no pre-processing)	71,44%
Cleaning and tokenizing	70,94%
Stopword and tokenizing	71,78%
Stemming and tokenizing	72,06%

Based on tabel 7. it can be seen the average of f1-score of linear SVM kernel, SVM kernel RBF, and multinomial naïve bayes from each pre-processing. This average f1-score will be used to determine which pre-processing is best for performing sentiment analysis in this research.



Gambar 5. Comparison chart of all averages

DISCUSSION

The best algorithm

Based on tabel 1. until tabel 3. and gambar 2 , the accuracy results of the linear SVM kernel is 71%, SVM kernel RBF 77%, and multinomial naïve bayes 65%. Meanwhile, when viewed and compared per sentiment, for sentiment -1 in precision SVM kernel RBF is superior with 74%, sentiment 0 for precision multinomial naïve bayes is superior with 78%, and with a percentage of 90% precision sentiment 1 SVM kernel RBF is a higher percentage than the others. Furthermore, for recall, for sentiment -1, multinomial naïve bayes is higher with 75%. For sentiment 0, SVM kernel RBF is higher with 81%, and for sentiment 1, multinomial naïve bayes is higher with 86%. And finally, for f1-score on sentiment -1, it can be seen that SVM kernel RBF are higher with 72%. For sentiment 0, SVM kernel RBF is higher with 75%, and for sentiment 1 it is seen that SVM kernel RBF is higher with 83%. Thus, if we look at the number of things that are higher, then using SVM kernel RBF is clearly better when compared to SVM using linear kernel or multinomial naïve bayes.

The best pre-processing

Based on tabel 24. and gambar 5. which contains the average accuracy of each pre processing, it can be said that pre processing using only tokenizing, combining tokenizing with stopwords and combining tokenizing with stemming gets the highest average of 71.67% when compared to combining tokenizing with cleaning which only gets an average of 70.67%. Next, based on tabel 25. and gambar 5. which contains the average precision of each pre processing, it can be said that pre processing using the combination of tokenizing with stopwords and combining tokenizing with stemming gets the highest average score of 73.67% when compared to only using tokenizing which gets 71.67% and combining tokenizing with cleaning which gets 73.22%. Furthermore, based on tabel 26. and gambar 5. which contains the average recall of each pre-processing, it can be said that pre-processing using a combination of tokenizing with stopwords and a combination of tokenizing with stemming gets the highest average of 72.56% when compared to only using tokenizing 72.33% and a combination of tokenizing with cleaning 71.33%. Lastly, based on tabel 27. and gambar 5. which contains the average f1-score of each pre processing, it can be said that pre processing using a combination of tokenizing with stemming gets the highest average of 72.06% when compared to only using tokenizing 71.44%, combining tokenizing with cleaning 70.94% and combining tokenizing with stopwords 71.78%.

Based on the highest average number, it is found that there are two top pre-processing, which are the pre-processing using a combination of tokenizing with stemming and a combination of tokenizing with stopwords. Tabel 14 to 16 and gambar 3, are the contents of the confusion matrix of the tokenizing-stopword pre-processing combination which is then averaged from all modeling. While tabel 19 to 21 and gambar 4, are the contents of the confusion matrix from the combination of tokenizing-stemming pre-processing which then from all modeling is averaged. Seen from the highest average number, it can be said that pre-processing using a combination of tokenizing with stemming is the best pre-processing in this research.

Pre processing using tokenizing gets a high average accuracy because at the tokenizing step there is a word-by-word separation and it makes it easier for the computer to read. And by adding the stemming process, it can also make it easier to analyze because words that previously had front or back affixes have been converted into raw words or their roots. Therefore, pre-processing using tokenizing and stemming-tokenizing can get a high average accuracy and become the best pre-processing done in the analysis of sentiments in this research.

Table 7, 8, 12, 13, 17, 18, 22 and 23, are the confusion matrix of SVM using sigmoid and polynomial kernels. It can be seen, if using a sigmoid kernel produces a confusion matrix that is less than or equal to the results of the confusion matrix using a linear kernel and much higher than the confusion matrix using multinomial naïve bayes. Whereas SVM using polynomial kernels unexpectedly can produce a confusion matrix that is quite high when compared to other modeling. However, this does not affect the results of this research at all, because in this research only SVM uses linear kernel and RBF kernel to see the final results.

CONCLUSION

Based on the results of research and testing of Support Vector Machine algorithm using linear kernel and RBF kernel and Naïve Bayes using multinomial naïve bayes in classifying YouTube comments about LGBT in Indonesia using Indonesian comments, it can be concluded that Support Vector Machine using RBF kernel is the best algorithm in this research with 77% accuracy with precision for sentiment -1 74%, recall 72% and f1-score 72%. For sentiment 0, 70%

for precision, 81% for recall, and 75% for f1-score. And the last, for sentiment 1, with 90% precision, 77% recall and 83% f1-score. In addition, pre-processing using stemming-tokenizing is the best pre-processing used for sentiment analysis in this research based on the highest average number.

This research can certainly still be further developed by adding pre-processing in order to get results on a more optimal confusion matrix, and can also be done with other pre-processing combinations in order to see which combination is best for sentiment analysis. In future research, it can be re-experimented why SVM using sigmoid kernel produces less than or equal to the results of the confusion matrix using linear kernel and higher than the confusion matrix using multinomial naïve bayes, as well as why SVM using polynomial kernel can produce a higher confusion matrix than other modeling. In addition, more attention should be paid to abbreviations or idioms in the dataset so that the analysis process can get the best results.

DAFTAR PUSTAKA

[1] A. P. Giovani, Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "ANALISIS SENTIMEN APLIKASI RUANG GURU DI TWITTER MENGGUNAKAN ALGORITMA KLASIFIKASI," *Jurnal TEKNOINFO*, vol. Vol. 14, No. 2, 2020, 116-124, 2020, doi: 10.33365/jti.v14i2.679.

[2] S. Fanissa, M. A. Fauzi, and S. Adinugroho, "Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking," vol. Vol. 2, p. hlm. 2766-2770, Agustus 2018, https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/1962.

[3] S. Ernawati and R. Wati, "Penerapan Algoritma K-Nearest Neighbors Pada Analisis Sentimen Review Agen Travel," *URNAL KHATULISTIWA INFORMATIKA*, vol. VOL. VI, Jun. 2018, https://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/article/view/3802/.

[4] A. Rahman Isnain, A. Indra Sakti, D. Alita, and N. Satya Marga, "SENTIMEN ANALISIS PUBLIK TERHADAP KEBIJAKAN LOCKDOWN PEMERINTAH JAKARTA MENGGUNAKAN ALGORITMA SVM," *JDMSI*, vol. Vol. 2, pp. 31–37, 2021, https://ejurnal.teknokrat.ac.id/index.php/JDMSI/article/view/1021.

[5] S. Hikmawan, A. Pardamean, and S. Nur Khasanah, "Sentimen Analisis Publik Terhadap Joko Widodo Terhadap Wabah Covid-19 Menggunakan Metode Machine Learning," *Jurnal Kajian Ilmiah (JKI)*, vol. Vol. 20, p. Halaman: 167-176, Mei 2020, https://ejurnal.ubharajaya.ac.id/index.php/JKI/article/view/117.

[6] Ratino, NoorHafidz, S. Anggraeni, and W. Gata, "Sentimen Analisis Informasi Covid 19 menggunakan Support Vector Machine dan Naïve Bayes," *Jurnal JUPITER*, vol. Vol. 12, pp. 1–11, Bulan Tahun 2020, https://jurnal.polsri.ac.id/index.php/jupiter/article/view/2388.

[7] A. Maureen Pudjajana and D. Manongga, "SENTIMEN ANALISIS TWEET PORNOGRAFI KAUM HOMOSEKSUAL INDONESIA DI TWITTER DENGAN NAIVE BAYES," *Jurnal SIMETRIS*, vol. Vol. 9, Apr. 2018, https://jurnal.umk.ac.id/index.php/simet/article/view/1922.

[8] I. Najiyah and I. Haryanti, “SENTIMEN ANALISIS COVID - 19 DENGAN METODE PROBABILISTIC NEURAL NETWORK DAN TF - IDF,” *JURNAL RESPONSIF*, vol. Vol. 3, pp. 100–111, Feb. 2021, https://ejurnal.ars.ac.id/index.php/jti/article/view/488.

[9] R. Sari, “Analisis Sentimen Pada Review Objek Wisata Dunia Fantasi menggunakan Algoritma K-Nearest Neighbor,” *Jurnal Sains dan Manajemen*, vol. Vol 8, Mar. 2020, https://ejournal.bsi.ac.id/ejurnal/index.php/evolusi/article/view/7371.

[10] R. Siringoringo and Jamaluddin, “Text Mining dan Klasterisasi Sentimen Pada Ulasan Produk Toko Online,” *Jurnal Penelitian Teknik Informatika Universitas Prima Indonesia (UNPRI)*, vol. Volume 2, Apr. 2019, http://jurnal.unprimdn.ac.id/index.php/JUTIKOMP/article/view/456.

[11] R. Rumajar, “Analisis Sentimen Tweet Sicepat Menggunakan SVM,” 2022. Accessed: Apr. 16, 2023. [Online]. Available: https://www.kaggle.com/code/ranodeyansarumajar/analisis-sentimen-tweet-sicepat-menggunakan-svm

[12] A. Munna, “Analisis Sentimen Aplikasi Gojek di Playstore Menggunakan Python dengan Algoritma Random Forest, SVM dan Naive Bayes,” 2022. Accessed: Jun. 23, 2023. [Online]. Available: https://medium.com/@aliyatulmunna7/analisis-sentimen-aplikasi-gojek-di-playstore-menggunakan-python-dengan-algoritma-random-forest-2e5504090f9b

[13] K. Khasanahh, “Analisis Sentimen Ulasan Aplikasi Shopee di google play store Menggunakan Metode Klasifikasi Algoritma Naive Bayes,” 2023. Accessed: Jun. 15, 2023. [Online]. Available: https://github.com/KhuswatunHasanahh/sentimen-shopee/blob/main/Analisis_Sentimen_Ulasan_Aplikasi_Shopee_di_google_play_store_Menggunakan_Metode_Klasifikasi_Algoritma_Naive_Bayes.ipynb