# Stemming's Impact on Indonesian Tweet Sentiment Analysis

**1Stanley Pudjowibowo, 2Daniel Hartanto**

**3Jonathan Kevin Giustino Trenggono**

1,2,3Informatics Engineering Major, Faculty of Computer Science, Soegijapranata Catholic University

120k10006@student.unika.ac.id, 220k10008@student.unika.ac.id, 320k10009@student.unika.ac.id

**Abstract**

*Text mining, information retrieval, and other natural language processing studies have all made substantial use of stemming. There is evidence, however, that stemming has minimal influence on text categorization accuracy. As a result, the goal of this study is to see how stemming affects sentiment analysis of Indonesian tweets. Furthermore, this research explores how two conditions on a pre-preprocessing task differ from one another when stemming is used and when it is not used. our result shows in SVM stemming in pre-processing using TF-IDF is 0.23% lower than non stemming and 0.5% higher when using the CountVectorizer method. In Multinomial Naive Bayes stemming increased accuracy by 0.73% in TF-IDF and 1.85% higher with the CountVectorizer method. Finally, this study shows that stemming, whether used with the SVM or Naive Bayes algorithms, has negligible and inconsistent effect on accuracy.*

**Keywords:** Stemming, Sentiment Analysis,Tweet Classification

## Introduction

Sentiment analysis is crucial in the context of Indonesian tweets because it captures the variety of thoughts and feelings expressed by millions and tens of thousands of users in the Indonesian language. In light of Indonesia's large Twitter user base, it makes sense to perform sentiment analysis in order to ascertain the typical viewpoint of Indonesians.

Sentiment analysis is the process of evaluating and classifying the emotions included in a text. It enables us to extract useful information from volumes of millions of bytes, allowing us to highlight new societal trends, consumer preferences, and public opinion. Pre-processing of text is one of the most important things that must be done in text mining, which must be done to prepare the data and also clean the data so that it can be processed. One of the processes that is usually used in pre-processing is stemming.

The data that is used in this research is gained from GitHub which is named Twitter Pilkada DKI 2017 which contains opinions about the 2017 Jakarta governor election. This work compares what is higher between stemming with using the Sastrawi algorithm, and not using stemming. stemming supposedly to increase the accuracy but if stemming doesn't give more effect then it can be removed from preprocessing.

The remaining part of this essay has a structure as seen below. Section 2 contains several works on Stemming, which form the basis of discussion in sections 3 and 4 of the paper. The suggested method for this work is discussed in section 5,and how this effort came to an conclusion

## Related Works

In recent development, some stemming algorithms has been developed specifically for Indonesian language such as Nazief-Adriani, Vega, Arifin Setiono, and Sastrawi. Some research papers conclude that the Nazief-Adriani Algorithm stemming process does not significantly affect the accuracy of language processing tasks[1], [2]. Other papers have an opposing conclusion that results using stemming algorithms have better accuracy compared to results with other methods [3]–[5]. However, Rizki et al [6] were able to determine that despite Nazief-Adriani Algorithms having a faster processing time than other algorithms, they are only marginally more accurate than the alternatives.

Sastrawi Algorithm, another stemming technique, outperformed Nazief-Adriani and Arifin Setiono Algorithm in terms of accuracy [7]. Yusliani et al [8] conclude that while Sastrawi Algorithm provides a good result, this algorithm has a downside which is a slow processing time. Agastasya [9] also said that while stemmer uses a lot of processing time it does not consistently boost accuracy. The Sastrawi algorithm also has a flaw in processing non-formal Indonesian language, thus improvements are required to generate accurate results [10].

Because on research both Hidayatullah [1], [2] works said that stemming using Nazief-Adriani doesn't affect the accuracy of language pre-processing tasks and based on Mustikasari [7] which said that the Sastrawi algorithm gives better results than Nazief-Adriani. so, this research wants to compare the result with Sastrawi stemming and compared it to the result with no stemming to find out which one is better.

## Proposed Method

The suggested methodology contains the following tasks: dataset, text pre-processing, term weighting, classification methodology, and performance evaluation explained below :

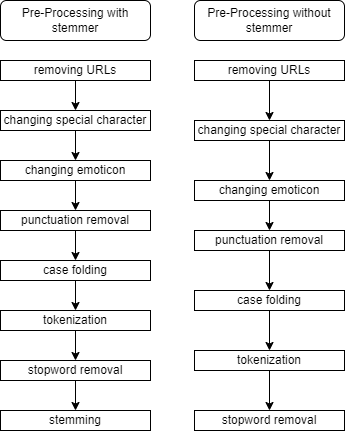
### Dataset

The sentiment analysis dataset that we took from Github was collected from the social networking site Twitter related to the implementation of the 2017 DKI Jakarta Governor Election, consisting of 900 tweet documents from 3 candidate pairs, namely Agus-Sylvi, Ahok-Djarot, & Anies-Sandi.The dataset contains 900 tweets which has 2 sentiment classes with 450 as positive data and 450 as negative data.

### Text Pre-processing

The pre-processing tasks suggested in this study include eliminating URLs, Twitter symbols and special characters, case folding, stemming, and stopwords:

1. Removing URLs that are found in tweets dataset.
2. Transforming emoticon into representative words and step replacing space inside bracket into underscore
3. Changing all username and hashtag into only @username and #hashtag
4. Removing unnecessary punctuation
5. All the words in the dataset transformed into lowercase
6. Separating word contained on the dataset by tokenization
7. Removing stopword on the dataset
8. Using of Sastrawi Stemmer on one batch of our dataset and one without



1. Pre-Processing Model

### C. Sastrawi Algorithm

Actually, it is a stemmer library called Sastrawi. TThe link is available on a source code provider site and could be accessed at https://github.com/sastrawi/sastrawi. The set of rules of Sastrawi stemmer are as following [7] :

1. First will check whether the word will not be stemmed is on dictionary root words or not. If exist, so the process will stop at this step.
2. If a word does not appear in the dictionary, it is assumed to be an affix word, in which case its suffix lah, kah, ku, mu, nya, kah, tah, or pun is deleted.
3. Next removing derivative affixes –i, -kan, -an, then delete be-, di-, ke-, me, pe-, se- and te-.
4. If at these steps resulted root word is not found in the dictionary, then the word is checked whether included in ambiguous table in the last column or not.
5. At last, when all the above steps failed, then algorithm returns the word to its original word.

### D. Term weighting

1. *TF-IDF*

A statistical tool used in natural language processing to assess a word's significance in a document is called TF-IDF. To give words that are common in one document but uncommon across the entire document collection more weight, it combines term frequency (TF) with inverse document frequency (IDF). Using this method, key phrases for document ranking and information retrieval can be found.

1. *CountVectorizer*

CountVectorizer is a text preprocessing technique used in NLP and machine learning. It converts text documents into a matrix representation where each document is represented by a vector. It tokenizes the text, creates a vocabulary of unique terms, and counts the occurrences of each term in each document. The resulting matrix provides a numerical representation of the text data, suitable for machine learning tasks like document classification and sentiment analysis.

### E. Classification Method

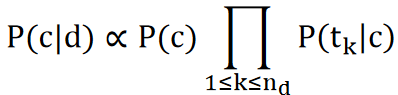
We use 2 type of classification method, SVM and Multinomial Naive Bayes both are widely used for their performance in classification

1. *SVM*

Support Vector Machines (SVMs) are classification and regression algorithms used for breaking down a dataset into different subclasses or predicting continuous values. Support vector machines use kernel functions to transform high-dimensional data that may take on nonlinear patterns so that individual decisions can be made easily. This can improve the margin in between classes, leading to increased accuracy in prediction or maximization of some objective function. SVMs are capable of handling both multi-class classification tasks as well as multivariational classification tasks. They provide an effective method for solving difficult characterization problems since they find optimal decision boundaries or hyperplanes for different attributes in a dataset.

1. *Multinomial Naive Bayes*

A probability-based classifier for text categorization applications is Multinomial Naive Bayes. The probability that a sample belongs to a class is determined as a function of the likelihood of witnessing feature values under the assumption that features are independent. For discrete data like word counts, this classifier lends itself well to natural language processing applications. It is simple in assumption, computationally efficient, and performs well when carrying out text classification tasks with fast learning machines. The likelihood that document d belongs to class c is calculated as



1. Multinomial Naive Bayes

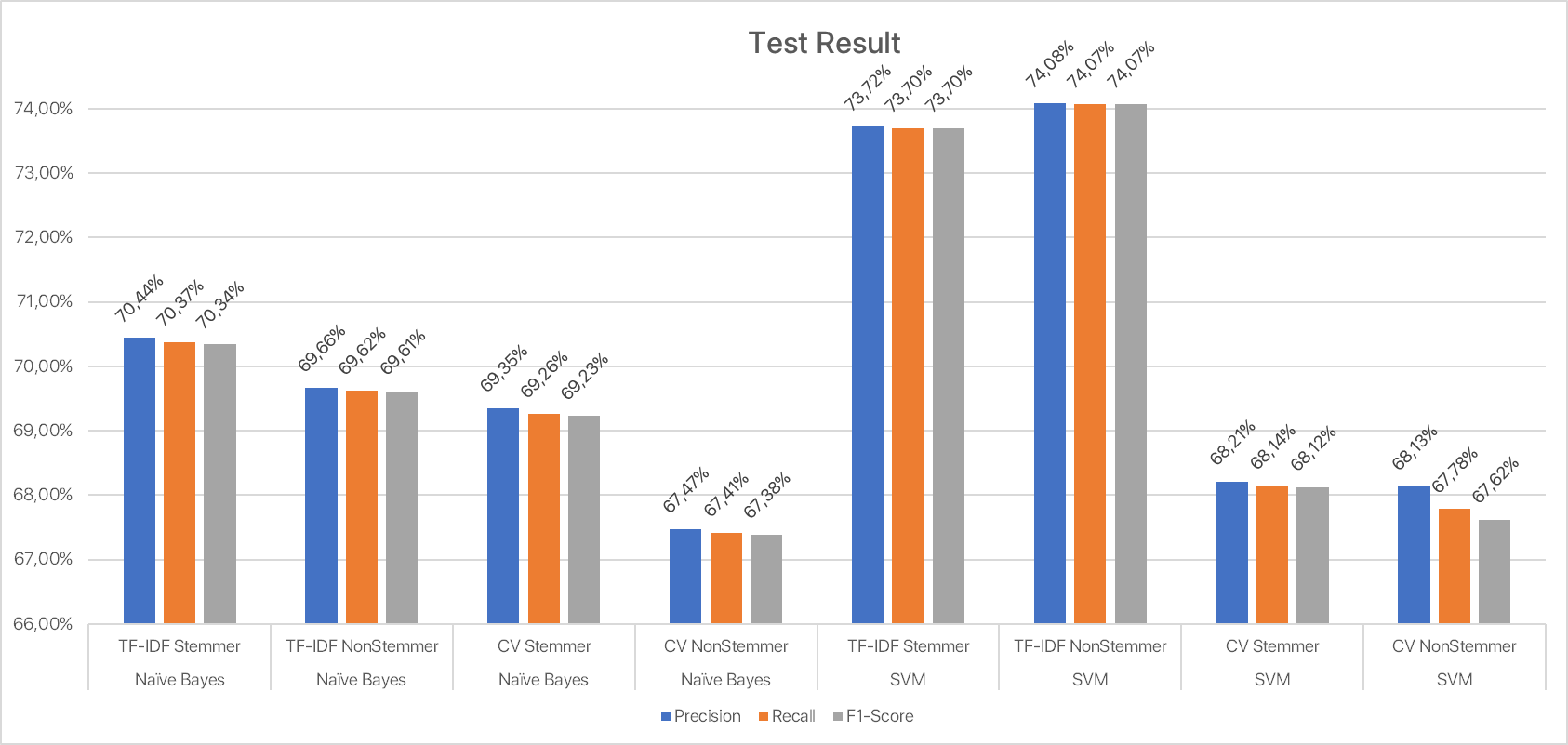
## IV. Result and Discusion

This experiment shows that stemming does not improve performance in every situation. when using stemming, 70.34% accuracy was achieved on multinomial naive bayes with TF-IDF. if not do any stemming on multinomial naive bayes with TF-IDF then accuracy shed into 69.62%. but when using Count vectorizer with multinomial naive bayes and Sastrawi stemmer this results will achieve an accuracy of 69.23%, but when using count vectorizer and multinomial naive bayes but not using stemmer this results got 67.38%

SVM and TF-IDF use stemming gets a total 73.7%. apart from that SVM and TF-IDF accuracy increased reach up to 74.07% when doing not using stemming. using Count vectorizer, and SVM get 68.12% when using SVM with stemmer, and 67.62% getting accuracy when using SVM without stemmer.

1. Result Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification** | **Method** | **Precision** | **Recall** | **F1-Score** |
| Naïve Bayes | TF-IDF Stemmer | 70,44% | 70,37% | 70,34% |
| Naïve Bayes | TF-IDF NonStemmer | 69,66% | 69,62% | 69,61% |
| Naïve Bayes | CV Stemmer | 69,35% | 69,26% | 69,23% |
| Naïve Bayes | CV NonStemmer | 67,47% | 67,41% | 67,38% |
| SVM | TF-IDF Stemmer | 73,72% | 73,70% | 73,70% |
| SVM | TF-IDF NonStemmer | 74,08% | 74,07% | 74,07% |
| SVM | CV Stemmer | 68,21% | 68,14% | 68,12% |
| SVM | CV NonStemmer | 68,13% | 67,78% | 67,62% |



1. Result Graph

## V. Conclusion

This work has compared between two stages of pre-processing, first the pre-process steps involved are stemming and another one does not involve stemming. Based on the experiments, this work made a conclusion that Stemming task does not raise the result accuracy either using SVM or Naive Bayes algorithm. However, the difference in accuracy between preprocessing which involves stemming and does not involve it is not too high. The accuracy difference for SVM stemming in pre-processing using TF-IDF is 0.23% lower than non stemming and 0.5% higher when using the CountVectorizer method. In Multinomial Naive Bayes stemming increased accuracy by 0.73% in TF-IDF and 1.85% higher with the CountVectorizer method. This result proves that stemmer addition in sentiment analysis is negligible and within margin of error.

Future research may alter the outcome with broader data sets that contain more variety. It is important to include more topics, such as sports, product reviews, consumer satisfaction, etc., as well as combine them with a new preprocessing method, in order to improve the accuracy of the results. This is due to the fact that we may now integrate data from other topics with election-related data and other new preprocessing techniques, changing the analytic tool's result.

## Daftar Pustaka

[1] A. F. Hidayatullah, “The Influence of Stemming on Indonesian Tweet Sentiment Analysis,” *Proceeding Electr. Eng. Comput. Sci. Inform.*, vol. 2, no. 1, pp. 127–132, Jun. 2015, doi: 10.11591/eecsi.v2.791.

[2] A. F. Hidayatullah, C. Ratnasari, and S. Wisnugroho, “Analysis of Stemming Influence on Indonesian Tweet Classification,” *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 14, p. 665, Jun. 2016, doi: 10.12928/telkomnika.v14i2.3113.

[3] T. Mardiana, T. B. Adji, and I. Hidayah, “Stemming Influence on Similarity Detection of Abstract Written in Indonesia,” *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 14, no. 1, Art. no. 1, Mar. 2016, doi: 10.12928/telkomnika.v14i1.1926.

[4] M. S. H. Simarangkir, “STUDI PERBANDINGAN ALGORITMA - ALGORITMA STEMMING UNTUK DOKUMEN TEKS BAHASA INDONESIA,” *J. Inkofar*, vol. 1, no. 1, Art. no. 1, Aug. 2017, doi: 10.46846/jurnalinkofar.v1i1.2.

[5] J. Jumadi, D. S. Maylawati, L. D. Pratiwi, and M. A. Ramdhani, “Comparison of Nazief-Adriani and Paice-Husk algorithm for Indonesian text stemming process,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1098, no. 3, Art. no. 3, Mar. 2021, doi: 10.1088/1757-899X/1098/3/032044.

[6] A. S. Rizki, A. Tjahyanto, and R. Trialih, “Comparison of stemming algorithms on Indonesian text processing,” *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 17, no. 1, p. 95, Feb. 2019, doi: 10.12928/telkomnika.v17i1.10183.

[7] D. Mustikasari, I. Widaningrum, R. Arifin, and W. H. E. Putri, “Comparison of Effectiveness of Stemming Algorithms in Indonesian Documents:,” presented at the 2nd Borobudur International Symposium on Science and Technology (BIS-STE 2020), Magelang, Indonesia, 2021. doi: 10.2991/aer.k.210810.025.

[8] N. Yusliani, R. Primartha, and M. Diana, “Multiprocessing Stemming: A Case Study of Indonesian Stemming,” *Int. J. Comput. Appl.*, vol. 182, pp. 15–19, Feb. 2019, doi: 10.5120/ijca2019918476.

[9] I. M. A. Agastya, “PENGARUH STEMMER BAHASA INDONESIA TERHADAP PEFORMA ANALISIS SENTIMEN TERJEMAHAN ULASAN FILM,” *J. Tekno Kompak*, vol. 12, no. 1, Art. no. 1, Feb. 2018, doi: 10.33365/jtk.v12i1.70.

[10] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, “Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation,” *J. Big Data*, vol. 8, no. 1, Art. no. 1, Jan. 2021, doi: 10.1186/s40537-021-00413-1.