JOURNAL OF BIG DATA SCIENCE

P-ISSN xxxx-xxxx
E-ISSN xxxx-xxxx

Volume 1 Number 1
September 2023

Naïve Bayes Method for Text-Based Sentiment Analysis on Social Media

Rezki Syaputra[1]*, Ria Andryani[2], Deni Erlansyah[3], Rupianti[4]

[1]Director, PT. Lingkaran Sistem Intelektual (LSI)
[2,3,4]Information System, Universitas Bina Darma

*rezkis@lingkaran.id

Soekarno Hatta street No. 3, Palembang, South Sumatra 30113, Indonesia

Received September 25th, 2023; **Revised** September 29th, 2023; **Accepted** September 30th, 2023

Abstract

Scientometrics is the study of measurement and analysis of science, innovation and technology through scientific publications. One form of measurement that can be taken is the network of authors measurement. This study uses author network analysis as a measurement tool performed in scientific studies. The purpose of this study was to observe the Authorsip network formed among professors at Bina Darma University, in order to determine which professors and departments are the most productive in producing yearbook articles or magazine. The method used in this study is the centrality of graphic degrees. Software used to view Gephi 0.9.2. The data used in this study are published data for the year 2015-2020. Based on the results of this study, it can be concluded that the agent with the highest central value is the EU with a value of 28, where the EU is the agent. with the largest number of publications. Meanwhile, the actor who has an influence or relationship and frequently collaborates on publications with the highest score on Betweenness Centrality is AM with a score of 61500.94.

Keywords: *Centrality Detection, Social Network Analysis, SNA, Data Visualization, Gephi*.

1. INTRODUCTION

One indicator of the progress of science and technology in a country is the number of published and utilized research results. Data on international scientific publications in Indonesia indexed by Scopus as of 2017 are at 12,098 publications, Indonesia is currently ranked third at the Asean level. Publication is one of the tasks that must be carried out by a lecturer, Scientometrics is the study of measurement and analysis of science, technology and innovation [1], [2]. Authors in a publication can be modeled as a network (graph) in which the main object is the author/author represented as a set of nodes, with relationships relationship between one and two authors when writing together. is a representation of the relationship (edge) [3], [4].

Social network analysis (SNA) looks at social relationships related to network theory, including nodes representing individual actors in the network and relationships representing relationships between networks. Individual [5], [6]. SNA is a method used to analyze the structure of social networks with various elements in an interconnected social environment. The network analysis approach has been widely used in various domains, such as: analyzing and detecting research network communities on Research Gate social media [7]. In addition, social network analysis has also been widely used in various social studies such as friendship networks [8], [9], hoax detection [10], and YouTube video searches with a network analysis approach with various approaches such as degree centrality [11], betweenness centrality, closeness centrality and community detection [12]–[14].

Centrality in SNA is a measure to see the position of an actor/group in a sociogram. Actor Degree Centrality is the number of direct relations owned by an actor. Betweeness Centrality is one way to measure centrality in a social network [15]. Interaction between 2 or more actors sometimes depends on other actors in the network. Actors who act as intermediaries between 2 or more actors are often considered to have a bigger role in the flow of information because they control the interactions between

these actors. Betweenness of an actor is the number of presences of an actor in the geodesic (shortest path) of each other pair of actors compared to the number of geodesic pairs of actors in the network. Individuals with the highest intermediate value are considered to have the most control over the flow of information in the network [15]. Another measure of centrality is closeness. Closeness measures the closeness between actors/nodes. The original idea of this measure is that an agent is called the center of the network if it can interact more easily and quickly with other agents. [15]. In terms of information flow, a hub close to other actors is more efficient because they can access information faster.

The centrality method in social network analysis can be used to determine the structure of the authentication network. Authors in a publication can be referred to as actors (nodes), while faculties can be referred to as networks (graphs). Actors here can be seen in relation to other actors in a network, how central the prominent actors (nodes) are in the network so that we can find out which lecturers and faculties are the most productive in producing publications. Based on this phenomenon, this article will present an authoritative network analysis that is applied to determine scientometrics and lecturer performance in the publication of scientific papers.

2. RESEARCH METHODOLOGY

2.1 Research Design

In this study, researchers used the Naïve Bayes Algorithm method in sentiment analysis. The research flow for sentiment analysis can be seen in Figure 1. which is processed using the Python tool.

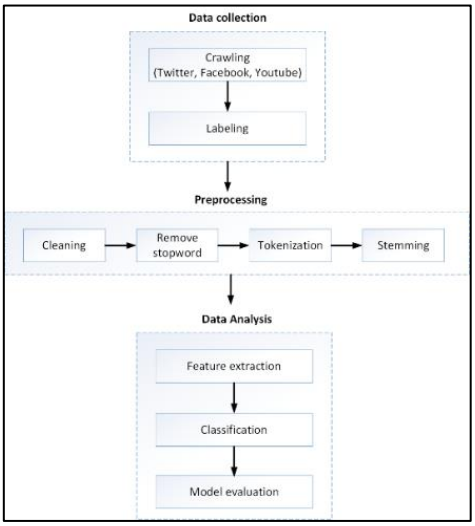

Figure 1 : Research Freamwork

2.2 Dataset

This study uses a dataset with a CSV file type with an excel file extension. CSV (Comma Separated Values) is a data format in a database where each record is separated by a comma (,) or a semicolon (;). The data used is data taken from the social media Youtube, Facebook, and Twitter, where for YouTube social media the researchers took data from the official CNN Indonesia account with the title of the special presidential staff, the data is 21 thousand, for social media Facebook the researchers took data from the page. official Kompas TV with the title of Joko Widodo presisden's special staff with 1400 data and for Twitter social media the researchers took data with keywords (stafsus, special staff, and stafsus_presiden) with a total of 1010 data. The number of datasets used by researchers from three social media is 3000 data. with 80% training data and 20% test data.

2.3 Labeling Data

The results of the data crawling process are in the excel file in Figures 2, 3, and 4 above, where the data labeling process will be carried out to determine the classification of opinions or views from the results of the crawled comment data earlier. In this labeling process, it is divided into 3 classes. They

are positive class, negative class and neutral class. An example of the data labeling process is shown in Table 1 below.

Table 1 : Data Labelling Results

Comment	Source	Label
Dodol....wamen, stafsus, stafsusmilenial, komisaris a,b,c...yg sifatnya bagi2 kue kekuasaan dan.tidak perlu,itu yg perlu dirampingkan.	Twitter	-1
Demokrasi butuh oposisi milenial. Oposisi partai politik.	Youtube	0
Pa jokowi pasti sdh mempertimbangkan & aq sangat menghargai nya. Krn org muda yg berdedikasi lebih baik dr yg tua tp bebal & bengal.	Facebook	1

In this case, class negative label -1 states that the comments are words that contain hate speech or hate speech, while positive class labeled 1 are words that do not contain hate speech elements and neutral class labeled 0 are neutral words. does not contain hate speech or contains hate speech. Each data will go through the preprocessing process by changing the form of unstructured data into structured data according to needs such as overcoming repetitive words, standard words, foreign words, and characters that have no meaning.

2.4 Data Analysis

At this stage, data analysis that has been taken is carried out, through several stages. The following are the stages that will be passed.

2.4.1 Preprocessing

Preprocessing is the stage of the process for cleaning data from unnecessary words or comments and words that have no meaning [16]. This process is carried out in accordance with the contents of the data from the data collection process or data crawling from social media Youtube, Facebook, Twitter [16]. The steps of the preprocessing process have the following sequence:

a. Cleaning

Cleaning is the process of removing symbols, punctuation, capital letters and numbers that often appear in comments from Twitter, Facebook and Youtube users so that the data becomes ineffective and has no meaning such as: (# \\ S + "," ", x,!, ()) [17]–[19]. This process is carried out using a program, so that this cleaning runs automatically before saving the decoded results in the form of an excel file, the researcher creates the dataset in excel form because the Python programming language has a library that provides read and write services for csv or excel file types. An example of applying the cleaning process can be seen in Table 2 below.

Table 2 : Sample Cleaning Result Data

Before Cleaning	After Cleaning
Dodol....wamen, stafsus, stafsusmilenial, komisaris a,b,c...yg sifatnya bagi2 kue kekuasaan dan.tidak perlu,itu yg perlu dirampingkan.	Dodol wamen stafsus stafsusmilenial komisaris a b c yg sifatnya bagi2 kue kekuasaan dan tidak perlu itu yg perlu dirampingkan
Demokrasi butuh oposisi milenial. Oposisi partai politik.	Demokrasi butuh oposisi milenial Oposisi partai politik
Pa jokowi pasti sdh mempertimbangkan & aq sangat menghargai nya. Krn org muda yg berdedikasi lebih baik dr yg tua tp bebal & bengal.	Pa jokowi pasti sdh mempertimbangkan & aq sangat menghargai nya Krn org muda yg berdedikasi lebih baik dr yg tua tp bebal & Bengal

b. Remove Stopword

Remove Stopword is the process of removing meaningless words or words that have no meaning such as the word and, or, you, me [19]. An example of the implementation process at the Remove Stopword stage can be seen in Table 3 below.

Table 3 : Example of Remove Stopword Results Data

Before Remove Stopword	After Remove Stopword
Dodol wamen stafsus stafsusmilenial komisaris a b c yg sifatnya bagi2 kue kekuasaan dan tidak perlu itu yg perlu dirampingkan	komisaris sifatnya bagi kue kekuasaan dan tidak perlu itu yg perlu dirampingkan
Demokrasi butuh oposisi milenial Oposisi partai politik	Demokrasi butuh oposisi milenial Oposisi partai politik
Pa jokowi pasti sdh mempertimbangkan & aq sangat menghargai nya Krn org muda yg berdedikasi lebih baik dr yg tua tp bebal & Bengal	jokowi pasti sdh mempertimbangkan sangat menghargai org muda berdedikasi lebih baik tua

c. Tokenization

Tokenization is the process of breaking a sentence into pieces which are called tokens [20], [21]. A token can be thought of as a form of a word, phrase, or a meaningful element. An example of the process at the tokenization stage can be seen in Table 4 below.

Table 4 : Example of Remove Stopword Results Data

Before Tokenization	After Tokenization
komisaris sifatnya kue kekuasaan tidak perlu dirampingkan	[‘komisaris’, ‘sifatnya’, ‘kue’, ‘kekuasaan’, ‘tidak’, ‘perlu’, ‘dirampingkan’]
demokrasi butuh oposisi milenial oposisi partai politik	[‘demokrasi’, ‘butuh’, ‘oposisi‘, ‘milenial’ ,’oposisi’ ‘partai’, ‘politik’].
jokowi pasti mempertimbangkan sangat menghargai muda berdedikasi lebih baik tua	[‘jokowi’, ‘pasti’, ‘mempertimbangkan’, ‘sangat’, ‘menghargai’, ‘muda’, ‘berdedikasi’, ‘lebih’, ‘baik’, ‘tua’]

d. Stemming

Stemming is the process of changing a word into its basic form by removing the affixes before and after the word [22]. An example of the stemming application process can be seen in Table 5 below.

Table 5: Example of Stemming Result Data

Before Stemming	After Stemming
[‘komisaris’, ‘sifatnya’, ‘kue’, ‘kekuasaan’, ‘tidak’, ‘perlu’, ‘dirampingkan’]	komisaris sifat kue kuasa tidak perlu ramping
[‘demokrasi’, ‘butuh’, ‘oposisi, ‘milenial’,’oposisi’ ‘partai’, ‘politik’].	demokrasi butuh oposisi milenial oposisi partai politik
[‘jokowi’, ‘pasti’, ‘mempertimbangkan, ‘sangat’, ‘menghargai’, ‘muda’, ‘berdedikasi’, ‘lebih’, ‘baik’, ‘tua’]	jokowi pasti pertimbangkan sangat hargai muda dedikasi lebih baik tua

2.5 Methods

Data passed through the text processing stage can then move to the next stage, namely classification using the Naïve Bayes Classifier algorithm (Septian and Shidik, 2017; Xu, Pan, and Xia, 2020). Data in text form will appear in two text classification results containing positive, neutral and negative. The following is the calculation of the Naïve Bayes Classifier algorithm.

- The first step of the Naïve Bayes Classification process is to calculate the probability of each class from the overall training data.
- Checking process. This process involves determining the accuracy of the model built during training, often using data known as test sets to predict labels. The Naïve Bayes Classifier method consists of two stages in the text classification process, the training phase and the classification phase. At the training stage, the analysis procedure is performed on a sample document in the form of a selection of words, namely words that can appear in a set of sample documents representing that document. The next step is to determine the probability for each type based on the document sample. The following is a sample calculation of the Naïve Bayes classifier.

Table 6 : Example of Stemming Result Data

Set	Document	Word	Label
1	Dokumen 1	Pa jokowi pasti sdh mempertimbangkan & aq sangat menghargai nya. Krn org muda yg berdedikasi lebih baik dr yg tua tp bebal & bengal.	1
2	Dokumen 2	Selamat kpd para anak muda pilihan Pak Dhe....semoga bisa berkontribusi utk membangun NKRI	1
3	Dokumen 3	Demokrasi butuh oposisi milenial. Oposisi partai politik.	0
4	Dokumen 4	Wkwkwk, gak terima ya bapak kalo pak Jokowi memilih stafsus milenial? Apakah anda iri karena sebagai sesepuh tidak terpilih ?	-1
5	Dokumen 5	Staff Khusus Demokrasi Perlu Didukung dengan Penuh dari pemerintah dan masyarakat	1

In the example above, the negative class label -1 states that the comments are words that contain hate speech or hate speech, while the positive class labeled 1 are words that do not contain hate speech elements and the neutral class labeled 0 are words that contain hate speech elements. neutral does not contain hate speech or contain hate speech. The following below are the calculation results for testing data.

$$P(H|X) = \frac{P(X|H)P(H)}{P(H)} \qquad 1$$

Calculate the prior probability of the positive, neutral, and negative classes.

y(pos)=2/4
y(net)=1/4
y(neg)=1/4

Then calculate the maximum likelihood value using the formula.

$$posterior = \frac{prior\ x\ likelihood}{evidence} \qquad 2$$

y(baik |pos) = (2+1)/(12+23) = 3/35
y(iri|pos) = (0+1) (12+23) = 1/35
y(demokrasi |pos) = (0+1)(12+23)= 1/35

y(baik |net) = (0+1)/(12+23) = 1/35
y(iri|net) = (0+1)(12+23)= 1/35
y(demokrasi |net) = (1+1)(12+23)= 2/35

y(baik |neg) = (0+1)/(12+23) = 1/35
y(iri|neg) = (1+1)(12+23)= 2/35
y(demokrasi|neg) = (0+1)(12+23)= 1/25

y(pos|d5) = 2/4*3/35*1/35*1/35= 3,49854E-05
y(net|d5) = 1/4*1/35*1/35*2/35 = 1,16618E-05
y(neg|d5) = 1/4*1/35*2/35*1/25 = 1,63265E-05
y(pos|d5) > y(neg|d5) dan y(net|d5)

The result of the above calculation is that the positive class on d5 has the highest value, so class d5 has a POSITIVE class.

2.5.1 Feature Extraction with Python

In the feature extraction process, the first process carried out by the system after tokenization is to convert the dataset into a vector representation using the library provided by Python called the Count Vectorizer library. For example, research uses 3 comments, including:
(D1) “Demokrasi butuh oposisi milenial. Oposisi partai politik”
(D2) “Selamat kpd para anak muda pilihan Pak Dhe...semoga bisa berkontribusi utk membangun NKRI”
(D3) “Pa jokowi pasti sdh mempertimbangkan & aq sangat menghargai nya. Krn org muda yg berdedikasi lebih baik dr yg tua tp bebal & bengal”

After the system preprocesses, there are 4 standard words from the 3 sentences above, namely "Democracy", "Congratulations", "Youth", and "Dedication". After the above steps, each document is displayed as a vector with elements, when the word is present. in the document, the value is given 1, if not, then it is given a value of 0. For example, it is shown in Table 6 below.

Table 6 : Making Word Vector

	Demokrasi	Selamat	Muda	Dedikasi
D1	1	1	0	0
D2	0	0	2	1
D3	1	0	0	1

Documents that have been converted into word vectors will then be calculated using the TF-IDF formula, using this formula will produce a word vector that has a weighted value. TF or Term Frequency itself is the number of times the words appear from a term in the document concerned, while IDF or Inverse Document Frequency is a calculation of how terms are spread or widely distributed in the collection of documents concerned. The process of calculating word weight is done by first calculating the TF or Term Frequency. You can see an example in Table 7 below.

Table 7 : TF (Term Frequency) Calculation Process

	D1	D2	D3
Demokrasi	1	0	1
Selamat	1	0	0
Muda	0	2	0
Dedikasi	0	1	1

After the TF weight calculation process is complete, then the process of determining the DF or Document Frequency is carried out, namely the number of terms (t) appearing.

Table 8 : DF (Document Frequency) Calculation Process

T (Term)	*DF (Document Frequency)*
Demokrasi	2
Selamat	1
Muda	2
Dedikasi	2

Then after the TF and DF processes then proceed to calculate the IDF (Inverse Document Frequency) value by calculating the value from the log of D results or the number of documents in this case there are 3 tweets, of the 3 documents divided by the value of the DF (Document Frequency). Then it will produce a calculation value like Table 9 below.

Table 9 : IDF (Inverse Document Frequency) Process

T (Term)	*DF (Document Frequency)*	*D/DF*	*IDF (Inverse Document Frequency)*
Demokrasi	2	1.5	log 1,5 = 0,176
Selamat	1	3	log 3 = 0,477
Muda	2	1.5	log 1,5 = 0,176
Dedikasi	2	1.5	log 1,5 = 0,176

After getting the IDF (Inverse Document Frequency) value, then proceed with calculating the TF-IDF. As in Table 10 Weighted Word Vector Examples below.

Table 10 : Example of TF-IDF Calculation Process

Q	TF									
	D1	D2	D3	DF	D/DF	IDF	IDF+1	D1	D2	D3
Demokrasi	1	0	1	2	1.5	0.176	1.176	1.176	0	1.176
Selamat	1	0	0	1	3	0.477	1.477	1.477	0	0
Muda	0	2	0	2	1.5	0.176	1.176	0	2.35	0
Dedikasi	0	1	1	2	1.5	0.176	1.176	0	1.176	1.176
								2.653	3.528	2.352

The results of the word vectors that have been weighted can be seen in Table 11 below.

Table 11 : Weighted Word Vector Examples

	Demokrasi	Selamat	Muda	Dedikasi
D1	1.176	1.477	0	0
D2	0	0	2.352	1.176
D3	1.176	0	0	1.176

3. RESULTS AND DISCUSSION

3.1 Implementation Of Naïve Bayes Classification In Python

Feature extraction process and the Naïve Bayes classification process which will later be compressed into one pipeline vectorizer class => transfomer => classifier. The classification process runs with the help of a library in the Python3 programming language which has the name scikit-learn library for the classification process, besides that there are numpy and pandas libraries as data reading. For the scikit-learn library used here are Pipeline, Count Vectorizer, Naïve Bayes, Multinomial NB, Confusion Matrix, Tfidf Transformer, and f1 Score. The first step in working on the feature extraction and classification process is to install the necessary libraries. Furthermore, after all libraries are installed, it is continued with the process of declaring all libraries that will be used. The program code for the declaration is in Figure 2 below.

```
import pandas as pd
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC, SVC
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report,confusion_matrix, accuracy_score, f1_score, precision_score, recall_score
```

Figure 2 : The library declaration function used

After completing the declaration of the library, it is continued with the process of taking a dataset that will be used as training data using the Pandas library. For the program code in Figure 3 below.

```
1 data = pd.read_excel
2 len(data)
```

Figure 3 : Calls Up The Data Set

Furthermore, the process of making a pipeline class in which there are 3 steps, namely changing the dataset from which Twitter data is crawled into a vector representation (converting letters to numbers) using the Count Vectorizer library with weighting using word vectors in the Tfidf Transformer library, the last stage is classification using the Multinominal Naive Bayes library. The process of implementing the three pipeline class creation processes is in Figure 4 below.

```
#Multinomial Naive Bayes
pipeline_mnb = Pipeline([
    ('vect', CountVectorizer()),
    ('tfidf', TfidfTransformer(use_idf=True, smooth_idf=True)),
    ('clf', MultinomialNB(alpha=1))
])

txt = data['cleantext'].values.astype('U')
#X_train, X_test, y_train, y_test = train_test_split(data['cleantext'], data['label'],  test_size=0.33, random_state = 0)
X_train, X_test, y_train, y_test = train_test_split(txt, data['label'],  test_size=0.33, random_state = 0)
pipeline_mnb.fit(X_train, y_train)
```

Figure 4 : The Pipiline Class Implementation Process

In the process of classifying this data, the researcher used randomized test data from 20% or 0.2 of the training data. The process of classifying this data is carried out using probability calculations from each class, so that new researchers can get clear results from the predicted data input. The final stage after carrying out all the classification processes, then it can be calculated from the performance of the algorithm used.

3.2 Model Testers

3.2.1 Test the Twitter Data Model

To determine the level of performance of the Naïve Bayes Algorithm, the researchers tested the model. The results of the classification will later be displayed in the form of confusion matrix. The table displayed in this confusion matrix consists of predicted class and actual class. The model of confusion matrix can be seen in Table 12.

Table 12. Confusion Matrix Model

		Predict Class	
		Class A	Class B
Actual Class	Positif	TP	FP
	Negatif	FN	TN

To find out the value of the accuracy of the model, it is obtained from the right amount of data the clarification results are divided by the total of the data, as in Figure 5 below.

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN}$$

Figure 5 : value of the accuracy of the model

In the test process this model will produce a confusion matrix with a size of 2x2 which can be seen in Table 13 below.

Table 13. Confusion Matrix Results

		Predict Class	
		Positif	Negatif
Actual Class	Positif	358	96
	Negatif	143	228

As in Table 13 above, the confused matrix matrix with a size of 2 x 2 each column represents the value of each class, namely the positive class and the negative class. To calculate the process of calculating the value of precision, recall and f-1 score in this system can be seen in Figure 6 below.

```
txt = data['cleantext'].values.astype('U')
#X_train, X_test, y_train, y_test = train_test_split(data['cleantext'], data['label'],  test_size=0.33, random_state = 0)
X_train, X_test, y_train, y_test = train_test_split(txt, data['label'],  test_size=0.33, random_state = 0)
pipeline_mnb.fit(X_train, y_train)
predictions = pipeline_mnb.predict(X_test)

print("Accuracy: {}".format(accuracy_score(y_test, predictions)))
print("F1 Score: {}".format(f1_score(y_test, predictions)))
print("Precision score: {}".format(precision_score(y_test, predictions)))
print("Recall score: {}".format(recall_score(y_test, predictions)))
print("Confusion matrix: {}".format(confusion_matrix(y_test, predictions)))
```

Figure 6 : The process of calculating the value of precision, recall and F-1 score

The results of the precision, recall, and f-1 score have an assessment measure of 0-1. The higher the value, the better, in the sense that the closer to the number 1 value from 0, the better the system. The results of the process of evaluating the overall model of this system are shown in Figure 7 below.

```
Accuracy: 0.7103
F1 Score: 0.6561
Precision score: 0.7037
Recall score: 0.6146
Confusion matrix:
[[358  96]
 [143 228]]
              precision    recall  f1-score   support

           0       0.71      0.79      0.75       454
           1       0.70      0.61      0.66       371

avg / total       0.71      0.71      0.71       825
```

Figure 9. Model Test Results

By knowing the value of precision, recall, and f-1 Score in the performance of the entire system, we can find out the ability of the system to find the accuracy or truth of the information requested by the user with the answer results issued by the system and inform the success rate of a system. in determining back information or the accuracy value is 71%. The results of the precision, recall, and f-1 score values in each class are shown in Table 14 below.

Table 14 : Results of the Value Precision, Recall, and F-1 score

Klasifikasi	*Precision*	*Recall*	*F-1 Score*
Positif	0.71	0.79	0.75
Negatif	0.70	0.61	0.66

It can be seen from the model test results in Table 4.9 that the accuracy and recall values of each class can be seen. The level of the system's processing power to find out the degree of accuracy between the information desired by the user as an active class is "71". %", and for the negative layer "70, %. The system's processing success rate in retrieving the positive layer information is "79%", for the negative layer it is "61%". With these values, it can be said that the performance of the system depends on the success of the system in retrieving positive and negative information.

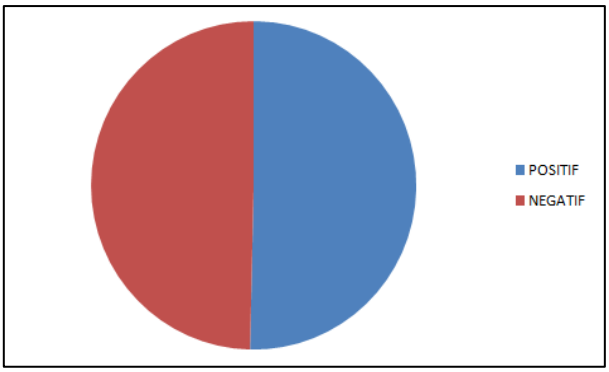

Figure 10 : Graph of Sentiment Analysis Model Test Results on Twitter

3.2.2 Test the Facebook Data Model

In the test process this model will produce a confusion matrix with a size of 2x2 which can be seen in Table 15 below.

Table 15 : Confusion Matrix Results

		Predict Class	
		Positif	Negatif
Actual Class	Positif	558	105
	Negatif	261	334

Get the results from the accuracy value and 2x2 confusion matrix in Figure 11 below.

```
Accuracy: 0.7100
F1 Score: 0.7496
Precision score: 0.7668
Recall score: 0.7331
Confusion matrix:
[[138  66]
 [ 79 217]]
             precision    recall  f1-score   support

          0       0.64      0.68      0.66       204
          1       0.77      0.73      0.75       296

avg / total       0.71      0.71      0.71       500
```

Figure 11. Model Test Results

The accuracy value obtained from testing the model is 71.0% whose calculation process is based on the number of values of the diagonal confusion matrix divided by the entire amount of data. Because the amount of data in each training data class is not balanced, the amount of accuracy value is not the most important. The results of the precision, recall, and f-1 score values in each class are shown in Table 16 below.

Table 16 : Results of the Value Precision, Recall, and F-1 score

Klasifikasi	*Precision*	*Recall*	*F-1 Score*
Positif	0.64	0.68	0.66
Negatif	0.77	0.73	0.75

From the model evaluation results in Table 16, it can be seen that the accuracy value and the recall value of each class can be considered as the level of the system's processing ability to find out the accuracy level between the information. information desired by the user because the positive class is "64%" and for the negative class it is "77%". The success rate of the system processing to retrieve the positive layer information is "68%", for the negative layer it is "73%". With these values, it can be said that the performance of the system depends on the success of the system in retrieving positive and negative information.

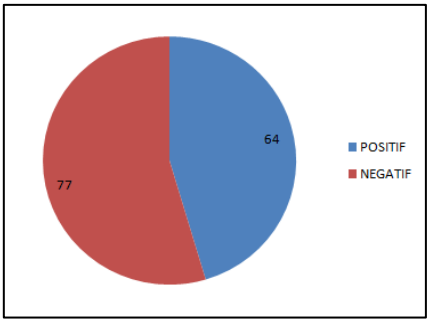

Figure 12 : Graph of Sentiment Analysis Model Test Results on Facebook

2.2.3 Test the YouTube Data Model

In the test process this model will produce a confusion matrix with a size of 2x2 which can be seen in Table 17 below.

Table 17 : Confusion Matrix Results YouTube

		Predict Class	
		Positif	Negatif
Actual Class	Positif	440	144
	Negatif	356	268

The accuracy value obtained from testing the model is 70.0% whose calculation process is based on the number of values of the diagonal confusion matrix divided by the entire amount of data. Because the amount of data in each training data class is not balanced, the amount of accuracy value is not the most important. The results of the precision, recall, and f-1 score values in each class are shown in Table 18 below.

Table 18. Results of the Value Precision, Recall, and F-1 score

Klasifikasi	*Precision*	*Recall*	*F-1 Score*
Positif	0.71	0.59	0.64
Negatif	0.69	0.73	0.75

Model evaluation results in Table 4.10 show, it can be seen that the precision and recall values of each class can be seen, the level of the system's processing ability to find out the level of precision between the information that the user wants is the positive class is " 71%", and for the negative class. grade is "69%". The system's success rate in retrieving information on positive classes is "59%" and on negative classes is "73%". With these values, it can be said that the performance of the system depends on the success of the system in retrieving positive and negative information.

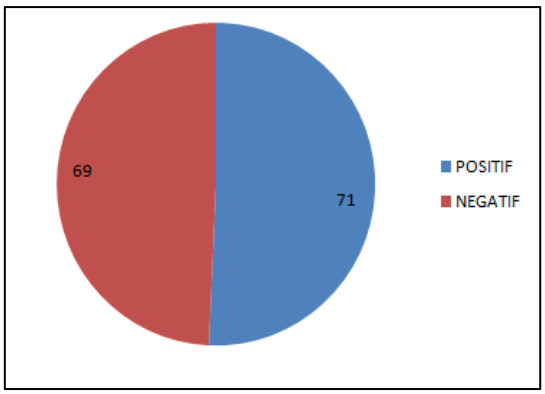

Figure 13 : Graph of Sentiment Analysis Model Test Results on Youtube

4. CONCLUSION

Based on the research conducted and the results of the discussion described in the previous chapters, the following conclusions can be drawn:

1) The Naive Bayes method can classify data in the form of text, especially text that comes from Twitter (tweet), Facebook, Youtube.
2) The number of words in each training class greatly affects the classification results on the testing data, therefore the data balance needs to be maintained.
3) Non-standard vocabulary can affect the classification results of a testing class if a training class has more data on the number of non-standard words than other training classes.

REFERENCES

[1] A. Darko, A. P. C. Chan, X. Huo, and D.-G. O. Manu, "A Scientometric Analysis and Visualization of Global Green Building Research," *Build. Environ.*, vol. 149, pp. 501–511, 2019.
[2] B. Zhong, H. Wu, H. Li, S. Sepasgozar, H. Luo, and L. He, "A Scientometric Analysis and Critical Review of Construction Related Ontology Research," *Autom. Constr.*, vol. 101, pp. 17–31, 2019.
[3] A. Higaki, T. Uetani, S. Ikeda, and O. Yamaguchi, "Co-authorship Network Analysis in Cardiovascular Research Utilizing Machine Learning (2009–2019)," *Int. J. Med. Inform.*, vol. 143, p. 104274, 2020.
[4] V. M. Patel *et al.*, "Collaborative Patterns, Authorship Practices and Scientific Success in Biomedical Research: a Network Analysis," *J. R. Soc. Med.*, vol. 112, no. 6, pp. 245–257, 2019.
[5] D. N. Sari, D. Syamsuar, and E. S. Negara, "Structure Community Analysis on Social Network," in *In The 6th International Conference on Information Technology and Business Application (ICIBA2017)1*, Pusat Penerbitan dan Percetakan Universitas Bina Darma Press (PPP-UBD Press) Palembang, 2017, pp. 1–7.
[6] J. Kim and M. Hastak, "Social Network Analysis: Characteristics of Online Social Networks After a Disaster," *Int. J. Inf. Manage.*, vol. 38, no. 1, pp. 86–89, 2018.
[7] E. S. Negara, D. Kerami, I. M. Wiryani, and T. B. M. Kusuma, "Researchgate Data Analysis to Measure the Strength of Indonesian Research," *Far East J. Electron. Commun.*, vol. 17, no. 5, pp. 1177–1183, 2017.
[8] R. Andryani, E. S. Negara, and D. Triadi, "Social Media Analytics: Data Utilization of Social Media for Research," *J. Inf. Syst. Informatics*, vol. 1, no. 2, pp. 193–205, 2019.
[9] W. Anjar *et al.*, "Data Mining: Algoritma dan Implementasi," *Yayasan Kita Menulis*, 2020.
[10] D. F. Brianna, E. S. Negara, and Y. N. Kunang, "Network Centralization Analysis Approach in the Spread of Hoax News on Social Media," in *In 2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, IEEE, 2019, pp. 303–308.
[11] R. Amanda and E. S. Negara, "Analysis and Implementation Machine Learning for YouTube Data Classification by Comparing the Performance of Classification Algorithms," *J. Online Inform.*, vol. 5, no. 1, pp. 61–72, 2020.
[12] E. S. Negara, D. Triadi, and R. Andryani, "Topic Modelling Twitter Data with Latent Dirichlet Allocation Method," in *International Conference on Electrical Engineering and Computer Science (ICECOS)*, BATAM, 2019, pp. 386–390.
[13] T. Sutabri, A. Suryatno, and E. S. Negara, "Improving Naïve Bayes in Sentiment Analysis for Hotel Industry in Indonesia," in *In 2018 Third International Conference on Informatics and Computing (ICIC)*, IEEE, 2018, pp. 1–6.
[14] E. S. Negara and R. Andryani, "A Review on Overlapping and Non-Overlapping Community Detection Algorithms for Social Network Analytics," *Far East J. Electron. Commun.*, vol. 18, no. 1, pp. 1–27, 2018, doi: 10.17654/ec018010001.
[15] J. Zhang and Y. Luo, "Degree Centrality, Betweenness Centrality, and Closeness Centrality in Social Network," in *In Proceedings of the 2017 2nd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2017)*, 2017, pp. 300–303.
[16] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations," *Organ. Res. Methods*, vol. 25, no. 1, pp. 114–146, 2022.
[17] R. Ardianto, T. Rivanie, Y. Alkhalifi, F. S. Nugraha, and W. Gata, "Sentiment Analysis on E-Sports for Education Curriculum Using Naive Bayes and Support Vector Machine," *J. Ilmu Komput. dan Inf.*, vol. 13, no. 2, pp. 109–122, 2020, doi: 10.21609/jiki.v13i2.885.
[18] K. Sahu, Y. Bai, and Y. Choi, "Supervised Sentiment Analysis of Twitter Handle of President Trump with Data Visualization Technique," in *In 2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2020, pp. 0640–0646.
[19] A. F. Hidayatullah and M. R. Ma'arif, "Pre-processing Tasks in Indonesian Twitter Messages," in *Journal of Physics: Conference Series 801*, IOP Publishing, 2016, p. 012072. doi: 10.1088/1742-6596/755/1/011001.
[20] A. Rai and S. Borah, "Study of various methods for tokenization," in *Applications of Internet of Things*, 2021, pp. 193–200.

[21] G. N. R. Prasad, "Identification of Bloom's Taxonomy level for the given Question paper using NLP Tokenization technique," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 13, pp. 1872–1875, 2021.

[22] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 874, p. 012017, 2020, doi: 10.1088/1757-899X/874/1/012017.