ULIL ALBAB : Jurnal Ilmiah Multidisiplin
Vol.3, No.9, Agustus 2024

Comparison Of The Naïve Bayes Algorithm And The Decision Tree On Sentiment Analysis Of Student Comment Data In The Digital Teacher Assessment (DITA) Application

Ferat Kristanto[1], Agung Yuliyanto Nugroho[2]
[1]informatika, SMK Telkom Purwokerto
[2]Sistem Informasi, Universitas Cendekia Mitra Indonesia
E-mail:, feratk@smktelkom-pwt.sch.id[1] agungyuliyanto@unicimi.ac.id[2]

Article History:
Received: 15 Juli 2024
Revised: 26 Juli 2024
Accepted: 01 Agustus 2024

Keywords: *Naïve Bayes; Decision Tree; Lexicon Based; Sentiment Analysis*

Abstract: *Telkom Purwokerto Vocational High School (SMK) is a school managed by the Telkom Education Foundation. They use Digital Teacher Assessment (DITA) apps. Students provide comments to the teacher using the DITA application. The collected comment data will be grouped into three categories, namely positive, negative, and neutral. Based on the category of comments requires sentiment analysis in grouping these comments. Sentiment analysis uses lexicon-based. After getting sentiment analysis using lexicon-based, then the words are weighted using TF-IDF and then classified and evaluated. This study uses an algorithm naïve Bayes and a decision tree. So the results of the comparative research on the accuracy of the naïve Bayes algorithm and the decision tree with the decision tree algorithm have the highest level of accuracy, namely 99%. So it can be concluded that using the decision tree algorithm is better at classifying student comment sentiment analysis data.*

INTRODUCTION

Telkom Purwokerto Vocational High School (SMK) is a school managed by the Telkom Education Foundation. They use apps Digital Teacher Assessment (DITA) to continue to improve the quality of education they provide. The DITA application is used to collect input and assessments from students about their teacher's work to help improve their performance. Assessment is given at the end of the semester or at the end of each learning process.

The data collection process focused on student comments. The collected comment data will be grouped into three categories, namely positive, negative, and neutral comments. Based on the category of comments requires sentiment analysis in grouping these comments. Sentiment analysis is the computational study of opinions, feelings, and emotions expressed in texts. The fundamental task of sentiment analysis is to classify the polarity of documents, sentences, or opinion texts. Polarity has a meaning for what the text is for. A document, statement, or opinion has positive or negative aspects. Sentiment analysis is usually used to assess the public's likes and dislikes of goods and services.

Machine learning is a way to learn something quickly and easily using a computer. This is done by using special algorithms and methods to predict things, recognize patterns, and classify

things. Algorithms in machine learning that neural networks, decision trees, k-nearest neighbor, naïve bayes, random forests, and so on. This study uses an algorithm naïve bayes and decision tree in machine learning. The Naïve Bayes algorithm is a type of classification that uses probability and statistical methods. This method is faster and easier to use than other methods because it requires only a few (training data), An algorithm decision tree is a tree-based method used to classify data. This method can be improved by increasing and bagging, as it often overfits the data.

Lexicon-based is a way to analyze sentiment without having to train any data beforehand. This approach uses a predetermined list of words, each with a corresponding sentiment score. Easy to use and practical, making it a good choice for sentiment analysis of reviews or comments. Aftegetting sentiment analysis using lexicon-based, then the words are weighted using TF-IDF. TF-IDF is a data transformation process from textual data to numeric data for each word or feature that will be given a weight In a previous study conducted by Franly Salmon Pattiiha and Hendry entitled "Comparison of Methods K-NN, Naïve Bayes, Decision Tree for Twitter Tweet Sentiment Analysis Regarding Opinion on PT PAL Indonesia". The research objective is to analyze sentiment toward public opinion on Twitter social media by using data that has been collected into a dataset and processed using tools Rapidminer. This research uses the method naïve bayes, K-NN, and decision tree to make comparisons by looking at the level of accuracy of the three methods used. The results of the study show that the Naïve Bayes method has an accuracy rate of 84.08%, the K-NN method is 83.38% and the decision tree is 81.09%. The results of this study can show that the naïve Bayes method has a higher level of accuracy than other methods used with an accuracy rate of 84.08%.

Based on previous research, the research that will be conducted is to analyze the sentiment of students' comments towards teachers using a comparison algorithm naïve Bayes dan decision tree. The difference from previous research is to do a sentiment analysis of student comments using the process preprocessing and then use lexicon based for sentiment analysis compared to determining sentiment manually to find out the different results obtained by grouping positive, negative, and neutral sentiments and then using the TF-IDF word weighting and the results compare the level of accuracy in each algorithm.

RESEARCH METHOD

In this research, the correct steps are needed so that this research can run effectively. The following flow of this research can be seen in Figure 1.

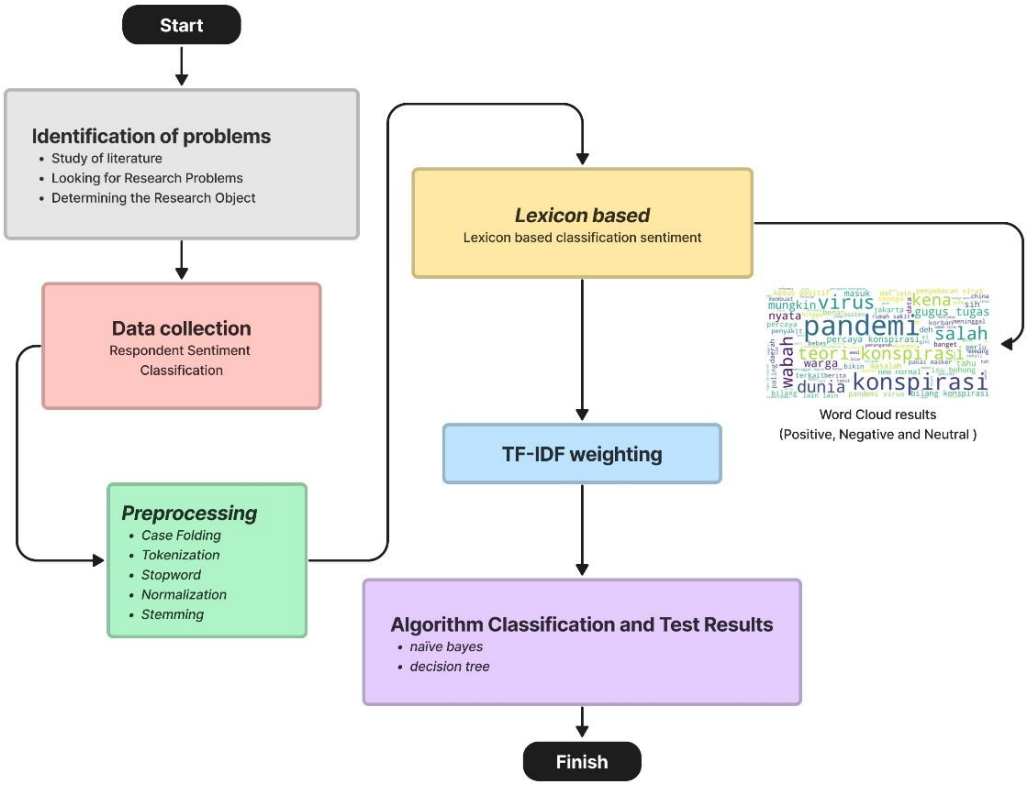


Figure 1. Research Flow

Based on Figure 1, the research flow has its stages, the explanation of the stages is as follows:

1. Identification of problems
   In this research the first step is problem identification, in problem identification there is a literature study, looking for research problems, and determining objects. The literature study stage is the process of finding, using, and studying various kinds of literature in the form of journals, books, papers, and so on related to sentiment analysis research. The problems and objects used in this research are how to do sentiment analysis of student comments in the application Digital Teacher Assessment (DITA) by using a lexicon-based, naïve bayes algorithm and decision tree. Students can comment on the DITA application to be addressed to the teacher. The data collection process focused on student comments. The collected comment data will be grouped into three categories, namely positive, negative, and neutral comments. Based on the category of comments requires sentiment analysis in grouping these comments.
2. Data collection
   At this stage data collection of students, comments were carried out in the DITA application. The data taken is then classified as sentiment in the manual sentiment labeling process.
3. Preprocessing
   There are many student commentary data that have been collected noise so it is necessary to have a stage of the elimination process noise so that the sentiment analysis process becomes more accurate . Channel preprocessing in this study can be seen in Figure 2.

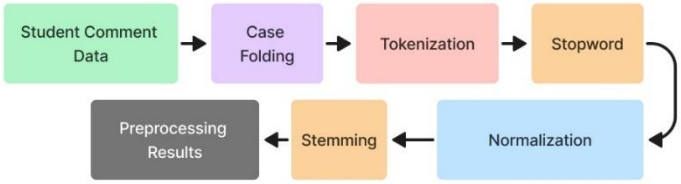


Figure 2. Preprocessing Flow

4. Lexicon Based
   After the data is preprocessed, the next step is data classification using lexicon-based. By grouping sentiment analysis into 3 groups, namely positive, negative, and neutral [11]. Results lexicon-based will be compared with manual sentiment data. Next results lexicon-based visualization is made.
5. TF-IDF weighting
   Word weighting TF-IDF calculates the weighted value of each word for each document. It is divided into two processes namely TF and IDF. TF (Term Frequency) counts the number of occurrences of each word in the document, and with the most occurrences of a word, the value of that word is the largest. IDF (Inverse Document Frequency) calculates the number of documents for each word that rarely appears in documents that are considered to have the greatest value. If the word has many occurrences of the word in the document, the result will have little value.
6. Algorithm Classification and Test Results
   The classification algorithm is carried out on a lexicon-based dataset with TF-IDF weighting and then processed from each naïve Bayes algorithm and decision tree.

The result of classifying algorithms according to lexicon-based sentiments is a comparison of the level of accuracy of each algorithm that uses algorithm performance measurement parameters, namely precision, recall, fl-measure, and accuracy.

The Naïve Bayes algorithm goes through training and classification stages in the classification process. At the training stage, the analysis process is carried out on sample documents, selecting the words that may appear in the sample document set as much as possible to represent the document. From the sample documents, the initial probabilities for each category are sought. For the classification stage of one document, the category value is determined based on the terms that appear in the classified documents.

The Decision Tree algorithm is a data mining method for classifying data. Variables or features are root nodes, internal nodes, and terminal nodes. The class label is the terminal node. To produce a decision tree, the Decision Tree method is often used. The data in the decision tree is expressed in the form of a table with attributes and records. Existing attributes are evaluated using statistical measures in the form of information acquisition, to measure the effectiveness of attributes when classifying a set of data samples.

This system is used by researchers to determine data accuracy by calculating precision (PREC) and recall (REC) values. The Confusion Matrix is then used to get the actual data results, which are then used to calculate classification predictions. The Confusion Matrix can be seen in Table 1.

Table 1. Confusion Matrix

	Labels or classes	
	Positve	Negative
Positve	True Positve	False Positive
Negative	False Negative	True Negative

Based on the confusion matrix in Table 1, it can be seen the various parameters of the algorithm performance measurement, namely precision, recall, in-measure, and accuracy.
Precision is a parameter to measure the accuracy of an algorithm. Calculate precision with the formula shown in equation.

$$precision = \frac{TP}{TP+FP} \quad (1)$$

The recall is a parameter to measure the completeness of an algorithm. Count recall with the formula shown in equation (2).

$$recall = \frac{TP}{TP+FN} \quad (2)$$

F-measure is the harmonic average of the precision and recall. The highest value is 1 and the lowest value is 0. Count f-measure with the formula shown in equation (3).

$$F - measure = \frac{2 \times precision \times recall}{precision+recall} \quad (3)$$

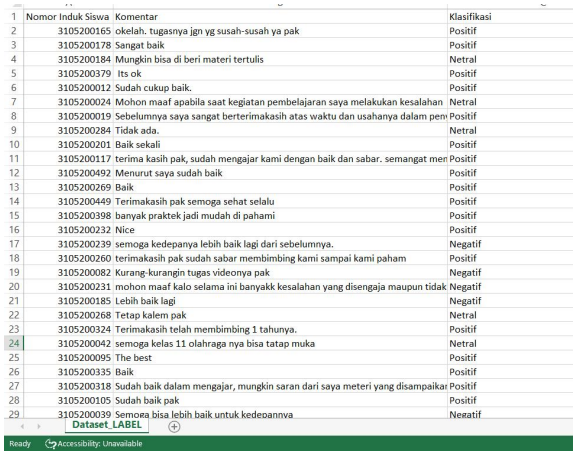
Accuracy is a calculation commonly used to evaluate the performance of an algorithm. Accuracy is calculated based on the ratio of the amount of data correctly predicted by the algorithm to the sum of all existing data datasets. Calculating accuracy with the formula shown in equation (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

RESULT AND DISCUSSION

This chapter discusses the results of sentiment analysis research on students' comments on teachers in the DITA application using the Lexicon Based and Naive Bayes algorithms and decision trees. The collection of data used in this study is data on student comments in the DITA application. Data processing uses the Jupyter Notebook application.

Data on student comments were taken and then classified as sentiments by manually labeling sentiments to produce sentiments which were grouped into 3 namely positive, negative, and neutral. Excel data with manually labeled CSV format is shown in Figure 3.



	Nomor Induk Siswa	Komentar	Klasifikasi
2	3105200165	okelah. tugasnya jgn yg susah-susah ya pak	Positif
3	3105200178	Sangat baik	Positif
4	3105200184	Mungkin bisa di beri materi tertulis	Netral
5	3105200379	Its ok	Positif
6	3105200012	Sudah cukup baik.	Positif
7	3105200024	Mohon maaf apabila saat kegiatan pembelajaran saya melakukan kesalahan	Netral
8	3105200019	Sebelumnya saya sangat berterimakasih atas waktu dan usahanya dalam pen	Positif
9	3105200284	Tidak ada.	Netral
10	3105200201	Baik sekali	Positif
11	3105200117	terima kasih pak, sudah mengajar kami dengan baik dan sabar. semangat men	Positif
12	3105200492	Menurut saya sudah baik	Positif
13	3105200269	Baik	Positif
14	3105200449	Terimakasih pak semoga sehat selalu	Positif
15	3105200398	banyak praktek jadi mudah di pahami	Positif
16	3105200232	Nice	Positif
17	3105200239	semoga kedepanya lebih baik lagi dari sebelumnya.	Negatif
18	3105200260	terimakasih pak sudah sabar membimbing kami sampai kami paham	Positif
19	3105200082	Kurang-kurangin tugas videonya pak	Negatif
20	3105200231	mohon maaf kalo selama ini banyakk kesalahan yang disengaja maupun tidak	Negatif
21	3105200185	Lebih baik lagi	Negatif
22	3105200268	Tetap kalem pak	Netral
23	3105200324	Terimakasih telah membimbing 1 tahunya.	Positif
24	3105200042	semoga kelas 11 olahraga nya bisa tatap muka	Netral
25	3105200095	The best	Positif
26	3105200335	Baik	Positif
27	3105200318	Sudah baik dalam mengajar, mungkin saran dari saya meteri yang disampaikar	Positif
28	3105200105	Sudah baik pak	Positif
29	3105200039	Semoga bisa lebih baik untuk kedepannya	Negatif

Dataset_LABEL

Ready Accessibility: Unavailable

Figure 3. Manual Labeling Comment Data

Based on the results of Figure 3, the comment data that will be processed is as many as 676 student comments, with the student ID number, comments, and classification fields. As shown in Figure 4.

	Nomor Induk Siswa	Komentar	Klasifikasi
0	3105200165	okelah. tugasnya jgn yg susah-susah ya pak :v	Positif
1	3105200178	Sangat baik,terba	Positif
2	3105200184	Mungkin bisa di beri materi tertulis	Netral
3	3105200379	Its ok	Positif
4	3105200012	Sudah cukup baik.	Positif
...	...	...	...
671	3105190386	Baikk	Positif
672	3105190548	Materi mudah di pahami	Positif
673	3105200351	Pembelajaran yang disampaikan oleh Pak Agus cu...	Positif
674	3105190500	cukup baik	Positif
675	3105190485	Baik	Positif

676 rows × 3 columns

Figure 4. Preliminary Data Display

The text preprocessing stage is necessary because the data collected from student comments contain a lot of noise, and does not contain useful information for the sentiment classification process such as symbols, punctuation marks, numbers, and the use of foreign languages. The preprocessing stages are case folding, tokenization, stopword, normalization, and stemming.

The case folding stage is used to change upper case letters to lower case or lower case letters. For example "Maybe" becomes "maybe". The following is the code for the case folding stages as shown in Figure 5.

```
#Preprocessing
#case folding
data['Komentar_lower']= data['Komentar'].str.lower()
data
```

Figure 5. Case Folding Process

The tokenization stage is used to remove characters that do not affect the sentiment classification process. Deleted characters such as comma (,), period (.), symbols, characters such

as letters, numbers, and separate words from each student's comments. The following is the code for the tokenization stages as shown in Figure 6.

```
#data cleansing
#Tokenization
def remove_komentar_special(text):
  #Hapus tab, new line
  text=str(text).replace('\\t', " ").replace('\\n'," ").replace('\\u', " ")  .replace('\\', " ").replace('\n', " ").replace('u\d
  #Hapus emoticon
  text=text.encode('ascii', 'replace').decode('ascii')
  text=re.sub(r"^rt[\s]+", '',text)
  #Hapus mention
  text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())
  #hapus url
  return text.replace("http://"," ").replace("https://"," ")
data['Komentar_lower'] = data['Komentar_lower'].apply(remove_komentar_special)
data

def remove_number(text):
    return re.sub(r"\d+","",text)
data['Komentar_lower'] = data['Komentar_lower'].apply(remove_number)

def remove_punctuation(text):
    return text.translate(str.maketrans("","",string.punctuation))
data['Komentar_lower'] = data['Komentar_lower'].apply(remove_punctuation)

def remove_whitespace_LT(text):
    return text.strip()
data['Komentar_lower'] = data['Komentar_lower'].apply(remove_whitespace_LT)

def remove_single_char(text):
    return re.sub(r"\b[a-zA-Z]\b","",text)
data['Komentar_lower'] = data['Komentar_lower'].apply(remove_single_char)

def word_tokenize_wrapper(text):
    return word_tokenize(text)
data['Komentar_token'] = data['Komentar_lower'].apply(word_tokenize_wrapper)

print('tokenizing result : \n')
data
```

Figure 6. Tokenization process

The stopword stage is used to delete words that are not important based on the stopword dictionary. The following is the code for the stopword stages as shown in Figure 7.

```
#Stopword
list_stopwords = set(stopwords.words('indonesian','english'))

def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

data['Komentar_token_Stopword'] = data['Komentar_token'].apply(stopwords_removal)

print(data['Komentar_token_Stopword'].head())
data
```

Figure 7. Stopword Process

The normalization stage is used to carry out the process of normalizing words for every word that contains non-standard or noisy words to become standard words and ready to be processed. The following is the code for the normalization stages as shown in Figure 8.

```
#Normalization
normalizad_word = pd.read_excel("normalisasi.xlsx")

normalizad_word_dict = {}

for index, row in normalizad_word.iterrows():
    if row[0] not in normalizad_word_dict:
        normalizad_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]

data['Komentar_normalized'] = data['Komentar_token_Stopword'].apply(normalized_term)

print(data['Komentar_normalized'].head())
data
```

Figure 8. Normalization Process

Stemming stages are used to return words to their basic form. For example “his task” becomes “task”. The following is the code for the stemming stages as shown in Figure 9.

```
#Stemming
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in data['Komentar_normalized']:
    for term in document:
        if term not in term_dict:
            term_dict[term]=' '

print(len(term_dict))
print("-------------------")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":" , term_dict[term])

print(term_dict)
print("-------------------")

def get_stemmed_term(document):
    return [term_dict[term] for term in document]
data['Komentar_token_stemmed'] = data['Komentar_normalized'].apply(get_stemmed_term)
data.head()
data
```

Figure 9. Stemming Process

The preprocessing process has been completed. Following are the results of the preprocessing stages from case folding, tokenization, stopword, normalization, stemming, and finally a new comment after going through the preprocessing stage which is shown in Figure 10.

kasi	Komentar_lower	Komentar_token	Komentar_token_kemunculan	Komentar_token_Stopword	Komentar_normalized	Komentar_token_stemmed	Komentar_baru
sitif	okelah tugasnya jgn yg susahsusah ya pak	[okelah, tugasnya, jgn, yg, susahsusah, ya, pak]	{'okelah': 1, 'tugasnya': 1, 'jgn': 1, 'yg': 1...	[okelah, tugasnya, jgn, yg, susahsusah, ya]	[okelah, tugasnya, jgn, yg, susahsusah, ya]	[oke, tugas, jgn, yg, susahsusah, ya]	oke tugas jgn yg susahsusah ya
sitif	sangat baikterba	[sangat, baikterba]	{'sangat': 1, 'baikterba': 1}	[baikterba]	[baikterba]	[baikterba]	baikterba
tral	mungkin bisa di beri materi tertulis	[mungkin, bisa, di, beri, materi, tertulis]	{'mungkin': 1, 'bisa': 1, 'di': 1, 'beri': 1, ...	[materi, tertulis]	[materi, tertulis]	[materi, tulis]	materi tulis
sitif	its ok	[its, ok]	{'its': 1, 'ok': 1}	[its, ok]	[its, ok]	[its, ok]	its ok
sitif	sudah cukup baik	[sudah, cukup, baik]	{'sudah': 1, 'cukup': 1, 'baik': 1}	[]	[]	[]	
...	...	...	...	...	...	...	...
sitif	baikk	[baikk]	{'baikk': 1}	[baikk]	[baikk]	[baikk]	baikk
sitif	materi mudah di pahami	[materi, mudah, di, pahami]	{'materi': 1, 'mudah': 1, 'di': 1, 'pahami': 1}	[materi, mudah, pahami]	[materi, mudah, pahami]	[materi, mudah, paham]	materi mudah paham
sitif	pembelajaran yang disampaikan oleh pak agus cu...	[pembelajaran, yang, disampaikan, oleh, pak, a...	{'pembelajaran': 1, 'yang': 1, 'disampaikan': ...	[pembelajaran, agus, mudah, dipahami]	[pembelajaran, agus, mudah, dipahami]	[ajar, agus, mudah, paham]	ajar agus mudah paham
sitif	cukup baik	[cukup, baik]	{'cukup': 1, 'baik': 1}	[]	[]	[]	
sitif	baik	[baik]	{'baik': 1}	[]	[]	[]	

Figure 10. The results of the preprocessing process

The classification process uses Lexicon based on data that has already gone through the preprocessing stage. As shown in Figure 10, namely using data in the new comment field. Lexicon-based is used for sentiment classification and to identify words with positive, negative, or neutral sentiments by calculating the polarity value. If the polarity value < 0 then the sentiment is negative, if the polarity value = 0 then the sentiment is neutral, and if the polarity value > 0 then the sentiment is positive as shown in Figure 11.

Komentar_token_stemmed	Komentar_baru	Compound_Score	Sentiments
[oke, tugas, jgn, yg, susahsusah, ya]	oke tugas jgn yg susahsusah ya	0.000	Netral
[baikterba]	baikterba	0.000	Netral
[materi, tulis]	materi tulis	0.000	Netral
[its, ok]	its ok	0.296	Positif
[]		0.000	Netral

Figure 11. Lexicon-based classification process

Based on the results of the classification using the lexicon-based, there are 3 sentiment groups, in which the most student comments are in the neutral group as much as 95.12% and the lowest negative group is 0.89% as shown in the graph in Figure 12.

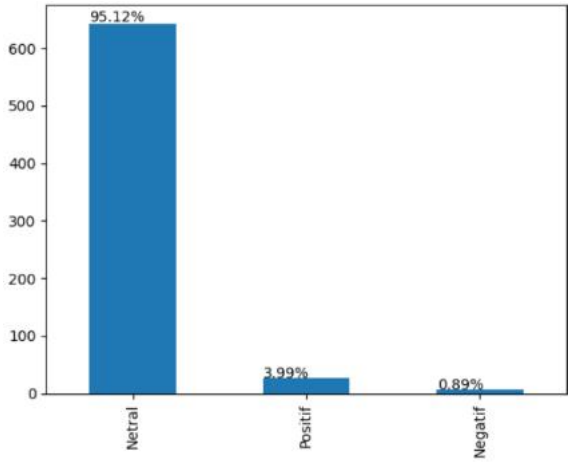


Figure 12. Graph of Lexicon-based Sentiment Classification Results

The results of the classification using the lexicon-based visualization are also made which are grouped based on positive, negative, and neutral sentiments. The visualization in question displays words belonging to the sentiment grouping [17] which can be seen in Figure 13 for positive (a), negative (b), and neutral (c).

Figure 13. Lexicon-based Sentiment Visualization Results

After the classification process uses the lexicon-based, new comments are carried out by the TF-IDF weighting process where the code and program appearance are shown in Figure 14.

```
#jumlah kalimat dan kata
document.shape
```

```
(676, 424)
```

```
#Menjumlahkan TF-IDF untuk setiap kalimat
result = np.sum(document,axis=1)
result.shape
```

```
(676,)
```

```
#Ditampilkan TF-IDF setiap kalimat dari kecil ke besar
sorted(result)
```

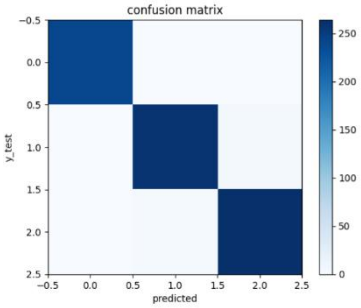
```
 1.0,
 1.0,
 1.0,
 1.0,
```

Figure 14. TF-IDF Weighting Results

Based on Figure 14 the results of the TF-IDF weighting, there are 676 comments and 424 words or terms that existed after the previous process. The result of the program code displays the TF-IDF value of each student's comment.

The results of the classification and evaluation of the naïve Bayes and decision tree algorithms using 60% test data and 40% testing data are obtained as follows:

The results of the decision tree algorithm obtained results which can be seen in Figure 15.

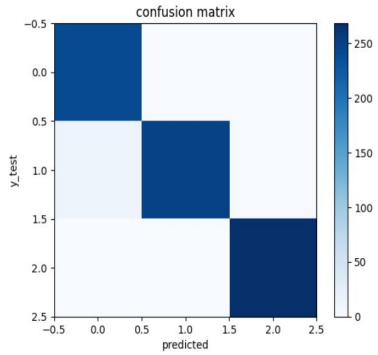



```
DecisionTree Accuracy : 0.9870466321243523
DecisionTree Precision : 0.9870646503799215
DecisionTree Recall: 0.9870466321243523
DecisionTree f1_score: 0.987046082841646
confusion matrix:
 [[240   0   0]
 [  0 258   6]
 [  0   4 264]]
==============================================================

              precision    recall  f1-score   support

     Negatif       1.00      1.00      1.00       240
      Netral       0.98      0.98      0.98       264
     Positif       0.98      0.99      0.98       268

    accuracy                           0.99       772
   macro avg       0.99      0.99      0.99       772
weighted avg       0.99      0.99      0.99       772
```

Figure 15. Results of the Decision Tree Algorithm

Based on Figure 15 the results of the decision tree algorithm with an accuracy value of 99% in the analysis of student comments and show the confusion matrix.

The results of the naïve Bayes algorithm obtained results which can be seen in Figure 16.




```
Naive Bayes Accuracy : 0.9805699481865285
Naive Bayes Precision : 0.9814544028556192
Naive Bayes Recall: 0.9805699481865285
Naive Bayes f1_score: 0.9805127186798257
confusion matrix:
 [[240   0   0]
 [ 13 249   2]
 [  0   0 268]]
==============================================================

              precision    recall  f1-score   support

     Negatif       0.95      1.00      0.97       240
      Netral       1.00      0.94      0.97       264
     Positif       0.99      1.00      1.00       268

    accuracy                           0.98       772
   macro avg       0.98      0.98      0.98       772
weighted avg       0.98      0.98      0.98       772
```

Figure 16. Naïve Bayes Algorithm Results

Based on Figure 16 the results of the naïve Bayes algorithm with an accuracy value of 98% in the analysis of student comments and the confusion matrix are shown.

The results of the naïve Bayes algorithm and decision tree using 60% test data and 40% testing data can be seen in the resulting confusion matrix, various algorithm performance measurement parameters can be known, namely precision, recall, fl-measure, and accuracy. A comparison of the naïve Bayes algorithm and the decision tree can be seen in Table 2.

Table 2. Naïve Bayes and Decision Tree Performance Comparison Results

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	99%	0.99	0.99	0.99
Naïve Bayes	98%	0.98	0.98	0.98

It can be concluded from Table 2, the results of the comparison of the accuracy of the naïve Bayes algorithm and the decision tree. The decision tree algorithm has the highest level of accuracy, namely 99%. So it can be concluded that using the decision tree algorithm is better at classifying student comment sentiment analysis data.

CONCLUSION

Based on the test results using Naïve Bayes and decision tree by using 60% training data and 40% testing data, several things are produced, as follows:

1. The results of the classification accuracy using the Naïve Bayes method obtained an accuracy of 98%, a precision of 98%, a recall of 98%, and an f1_score of 98%.
2. The results of the classification accuracy using the method decision tree obtained an accuracy of 99%, a precision of 99%, a recall of 99%, and an f1_score of 99%.
3. Overall the performance comparison of the Naïve Bayes method and decision tree shows that the result decision tree is better in classifying data. Based on the results of the analysis and conclusions, the researcher can give suggestions that for further research, in obtaining data, you can take data on other popular social media such as Facebook and Instagram so that the data obtained is more varied and can use different classification methods. so as to obtain more specific and good classification results.

REFERENCES

N. Tri Romadloni, I. Santoso, And S. Budilaksono, "Perbandingan Metode Naive Bayes, Knn Dan Decision Tree Terhadap Analisis Sentimen Transportasi Krl Commuter Line," *J. Ikra-Ith Inform.*, Vol. 3, No. 2, Pp. 1–9, 2019.

N. R. Muntiari And K. H. Hanif, "Klasifikasi Penyakit Kanker Payudara Menggunakan Perbandingan Algoritma Machine Learning," *J. Ilmu Komput. Dan Teknol.*, Vol. 3, No. 1, Pp. 1–6, 2022, Doi: 10.35960/Ikomti.V3i1.766.

M. K. Anam, B. N. Pikir, And M. B. Firdaus, "Penerapan Na ıve Bayes Classifier, K-Nearest Neighbor (Knn) Dan Decision Tree Untuk Menganalisis Sentimen Pada Interaksi Netizen Danpemeritah," *Matrik J. Manajemen, Tek. Inform. Dan Rekayasa Komput.*, Vol. 21, No. 1, Pp. 139–150, 2021, Doi: 10.30812/Matrik.V21i1.1092.

M. Alfi, R. Reynaldhi, And Y. Sibaroni, "Analisis Sentimen Review Film Pada Twitter Menggunakan Metode Klasifikasi Hybrid Svm, Naïve Bayes, Dan Decision Tree," Vol. 8, No. 5, Pp. 10127–10137, 2021.

Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou, And H. Jiang, "Refined Global Word Embeddings

Based On Sentiment Concept For Sentiment Analysis," *Ieee Access*, Vol. 9, No. 1, Pp. 37075–37085, 2021, Doi: 10.1109/Access.2021.3062654.

X. Fu, J. Yang, J. Li, M. Fang, And H. Wang, "Lexicon-Enhanced Lstm With Attention For General Sentiment Analysis," *Ieee Access*, Vol. 6, Pp. 71884–71891, 2018, Doi: 10.1109/Access.2018.2878425.

A. Fatihin, "Analisis Sentimen Terhadap Ulasan Aplikasi Mobile Menggunakan Metode Support Vector Machine (Svm) Dan Pendekatan Lexicon Based," 2022.

J. A. Septian, T. M. Fachrudin, And A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan Tf-Idf Dan K-Nearest Neighbor," *J. Intell. Syst. Comput.*, Vol. 1, No. 1, Pp. 43–49, 2019, Doi: 10.52985/Insyst.V1i1.36.

F. Salmon Pattiiha, "Perbandingan Metode K-Nn, Naïve Bayes, Decision Tree Untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap Pt Pal Indonesia," *J. Ris. Komputer)*, Vol. 9, No. 2, Pp. 2407–389, 2022, Doi: 10.30865/Jurikom.V9i2.4016.

H. Han, Y. Zhang, J. Zhang, J. Yang, And X. Zou, "Improving The Performance Of Lexicon-Based Review Sentiment Analysis Method By Reducing Additional Introduced Sentiment Bias," *Plos One*, Vol. 13, No. 8, Pp. 1–11, 2018, Doi: 10.1371/Journal.Pone.0202523.

A. S. Rusydiana, I. Firmansyah, And L. Marlina, "Sentiment Analysis Of Microtakaful Industry: Comparison Between Indonesia And Malaysia," *Int. J. Nusant. Islam*, Vol. 6, No. 1, Pp. 20–34, 2019, Doi: 10.15575/Ijni.V6i1.3004.

N. N. Wilim And R. S. Oetama, "Sentiment Analysis About Indonesian Lawyers Club Television Program Using K-Nearest Neighbor, Naïve Bayes Classifier, And Decision Tree," *Ijnmt (International J. New Media Technol.*, Vol. 8, No. 1, Pp. 50–56, 2021, Doi: 10.31937/Ijnmt.V8i1.1965.

I. Hilmy Khairi Idris, Mochammad Ali Fauzi, "Klasifikasi Teks Bahasa Indonesia Pada Dokumen Pengaduan Sambat Online Menggunakan Metode K-Nearest Neighbors Dan Chi-Square," *Syst. Inf. Syst. Informatics J.*, Vol. 3, No. 1, Pp. 25–32, 2017, Doi: 10.29080/Systemic.V3i1.191.

S. Panggabean, W. Gata, And T. A. Setiawan, "Analysis Of Twitter Sentiment Towards Madrasahs Using Classification Methods," *J. Appl. Eng. Technol. Sci.*, Vol. 4, No. 1, Pp. 375–389, 2022.

M. Rezapour, "Sentiment Classification Of Skewed Shoppers' Reviews Using Machine Learning Techniques, Examining The Textual Features," *Eng. Reports*, Vol. 3, No. 1, Pp. 1–13, 2021, Doi: 10.1002/Eng2.12280.

W. Li, P. Liu, Q. Zhang, And W. Liu, "An Improved Approach For Text Sentiment Classification Based On A Deep Neural Network Via A Sentiment Attention Mechanism," *Futur. Internet*, Vol. 11, No. 4, 2019, Doi: 10.3390/Fi11040096.

N. Z. Dina, "Tourist Sentiment Analysis On Tripadvisor Using Text Mining: A Case Study Using Hotels In Ubud, Bali," *African J. Hosp. Tour. Leis.*, Vol. 9, No. 2, Pp. 1–10, 2020.