IJISCS (International Journal of Information System and Computer Science)
IJISCS, Volume 8, Number 2, August 2024
E ISSN: 2598-246X; P ISSN: 2598-0793, pp. 115-127
Accredited SINTA 4 Number 79/E/KPT/2023
https://jurnal.ftikomibn.ac.id/index.php/ijiscs

SENTIMENT ANALYSIS ON MYPERTAMINA APPLICATION REVIEWS USING NBC AND SVM WITH NEGATION HANDLING

Amir Hamzah[1]*, Renna Yanwastika Ariyana[2]
[1,2]Departement of Informatics, Institute of Science & Technology AKPRIND, Yogyakarta, Indonesia

*Corresponding author
amir@akprind.ac.id

Article history:
Received June 28, 2021
Revised July 12, 2021
Accepted July 16, 2021

Keywords:
Negation handling;
MyPertamina;
NBC;
SVM.

Abstract
Pertamina has issued a cashless application for fuel purchases since July 2019, named as MyPertamina. The application aims to make it easier for customers to make payments in transactions at fuel stations. MyPertamina application can currently be downloaded on Google Playstore. Since its release until now, MyPertamina has been downloaded as many as 10 million with a rating of 3.1 and 339 thousand reviews. Unfortunately the low rating and user reviews dominated by negative comments show that the app's performance is still not satisfactory. The reviews data can be converted into valuable information by using entiment analysis. Many researchers have applied sentiment analysis to MyPertamina user comment data. However, there have been no studies that apply the handling of negation in MyPertamina reviews, even though negative comments are very often found the word negation, i.e 'tidak','jangan','belum' and 'bukan' that will change the sentiment of next sentiment word. Untreated negation words will lead to errors in classification which in turn will decrease accuracy. This study applies the handling of negation words using First Sentiment Word (FSW) and Fixed Window Length (FWL) methods. The classification algorithms used are Naïve Bayes Classifier (NBC) and Support Vector Machine (SVM). In this work, we analized 1000 comments consisting of 390 positive comments and 610 negative comments. The results showed that the best performance of negation handling is FSW. This method has improved accuracy by 2.5% and F1 by 1.5% using NBC algorithm and has improved accuracy by 2.9% nad F1 by 3.4% using SVM algorithm.

1.0 INTRODUCTION

Every company today is required to utilize information technology in improving service to customers. As a State-Owned Enterprise Pertamina developed a non-cash fuel payment service, called MyPertamina. MyPertamina is a digital financial service application that is integrated with the LinkAja application. To get a MyPertamina account, users can download the application first from the Google Play Store (for Android phones) or App Store (for iPhone). The breakthrough app is intended to provide a convenient way for customers to transact [1]. This application also provide prizes in the form of earning points and redeeming points [2]. To support government policies in the distribution of fuel oil, this application is also intended to control the distribution of subsidized fuel to be right on target [3]. However, since it was first created on July 1, 2022, this application has encountered several obstacles so that many users are disappointed[4]. Complaints submitted by many users include the number of bugs, difficulty in registration, authentication and smoothness in use [5], and many officials lack understanding of the use of the system [6]. Currently, the MyPertamina application on Google Playstore has a rating of 3.1 with 339 thousand reviews and has been downloaded by around 10 million users. The number of downloads is still far from the potential of motorized vehicle users, which is around 140 million [7]. However, the number of reviews can be elaborated further to determine the direction of the sentiment revealed from users, whether positive, negative or neutral. The study will use sentiment analysis to evaluate reviews in order to improve the system.

Sentiment analysis is the processing of text data that aims to analyze, process, extract textual data in the form of responses to an object or event and determine whether user sentiment includes positive and negative sentiments [8]. Sentiment analysis has been applied by many researchers on the MyPertamina application, among others, by [2], [9]-[10] and [11]. However, most sentiment analysis accuracy is still about 60%. This may be due to almost all researchers have not included negation handling factors in their analysis. According to [12] one of the causes of low accuracy in sentiment analysis is the absence of semantic analysis. Without semantic analysis, the ambiguity of meaning arising from negative words will cause the classification to be wrong. For example, on sentences "*aplikasi sangat tidak bagus*" which has a negative sentiment when the tokenization process generally uses unigram units, then word "tidak" will be ignored so that sentiment will turn into positive sentiment. In fact, in the comments of MyPertamina users who tend to be negative, the percentage of the emergence of negation words such as: "tidak", "bukan", and "belum" tends to be very high. For that reason, the handling of the word negation is indispensable in improving sentiment analysis performance. This study aims to apply the technique of handling the word negation in sentiment analysis. It is expected that with the application of negation word handling, the accuracy of sentiment analysis can be improved.

2.0 THEORETICAL

MyPertamina currently has a rating of 3.1 and product reviews of 339 thousand. The large number of reviews and low ratings have encouraged researchers to apply sentiment analysis to myPertamina reviews. The application of sentiment analysis in MyPertamina application product reviews has been widely carried out. Some researchers apply the NBC classification method, including [2], [10] and [11]. Setya Ananto & Hasan [2] analyzed 1.289 data with 285 positive data and 1004 negative data. The classification results obtained 77.4% accuracy, 49.9% precision and 76.8% recall. Nabilla et al. [9] conducted an analysis on 1001 *tweet* data sourced from twitter consisting of 494 positive and 507 negative. The results of the classification with NBC get 71% accuracy. The study with the higher number of reviews data was [11] that used 3948 data, and [10] that used 5.722 data. The studies with the highest data reviews are [13] that used 8000 data with details of 4300 negatives, 1575 neutrals and 1325 positives. The latter uses the SVM method to perform classification. However, the analysis results only showed 67% accuracy, 69% recall and 57% precision. It can be seen from the algorithm used by the researchers mostly using NBC and SVM. According to the [14] NBC method, it is the fastest method compared to other classification methods.

The application of negation handling in sentiment analysis has been widely done by researchers on various text and various languages, including dutch [15] and english [16]. The increasing number of text reviews in Indonesian has encouraged researchers to study the importance of handling negation for sentiment analysis on Indonesian objects. Ramadhan et al. (2022) have applied negation handling to Covid-19 data collected from Twitter as many as 902 data (441 positive, 195 negative and 266 neutral). The results showed a slight increase in accuracy from 59.1% to 59.6%. Another negatio handlling study on Twitter conducted by [18] has increased accuracy by about 3.17%. Implementation of negation handlling using modified syantactic rule have done by [19]. By applying their method to 1000 hotel review data (500 positive and 500 negative) they got an increase in accuracy of 3,3%.

2.1. Naïve Bayes Classifier (NBC)

NBC's method for opinion classification views opinions as a set of attributes $(a_1, a_2, a_3, \dots a_n)$. Suppose the opinion to be classified into V categories, then V is the set of categories. Classification is done by finding the maximum value of V_{MAP} according to (1) [20].

$$V_{MAP} = {}^{argmax}_{v_j \in V} P(v_j \vee a_1, a_2, a_3, \ldots a_n) \quad (1)$$

By applying *Bayes*' theorem, equation (1) can be written as (2)

$$V_{MAP} = \begin{matrix} argmax \\ v_j \in V \end{matrix} \frac{P(a_1,a_2,a_3,\ldots a_n \vee v_j)P(v_j)}{P(a_1,a_2,a_3,\ldots a_n)} \quad (2)$$

Considering that value $P(a_1, a_2, a_3, \ldots a_n)$ is a constant for every v_j , so that equation (2) can be written as (3).

$$V_{MAP} = {}^{argmax}_{v_j \in V} P(a_1, a_2, a_3, \ldots a_n \vee v_j)P(v_j) \quad (3)$$

Assuming that within each category, each attribute is independent of each other, equation (3) can be written as (4).

$$P(a_1, a_2, a_3, \ldots a_n | v_j) = \prod_i P(a_i \vee v_j) \quad (4)$$

Thus the purpose function (1) becomes (5).

$$V_{MAP} = {}^{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (5)$$

In the process of training, probability P(vj) and $P(a_i|v_j)$ can be calculated by formula (6) and (7).

$$P(v_j) = \frac{docs_j \vee}{training\_V} \quad (6)$$

$$P(a_i|v_j) = \frac{n_i + 1}{n + num\_token} \quad (7)$$

where:

$docs_j$	= the number of documents in category j
training_V	= the number of documents used in the *training* process
n_i	= the number of occurrences of the word a_i on category v_j
n	= the number of tokens that appear in the category v_j
num_token	= the number of unique words in all *training* data

2.2. Support Vector Machine (SVM)

Support vector machines (SVM) are included in supervised learning. The *SVM* find a separator function that can separate two data sets from two different classes. The concept can be explained simply that SVM tries to find the best hyperplane that functions as a separator for two classes in the input space by maximizing the distance between classes [21] as illustrated in Figure 1.

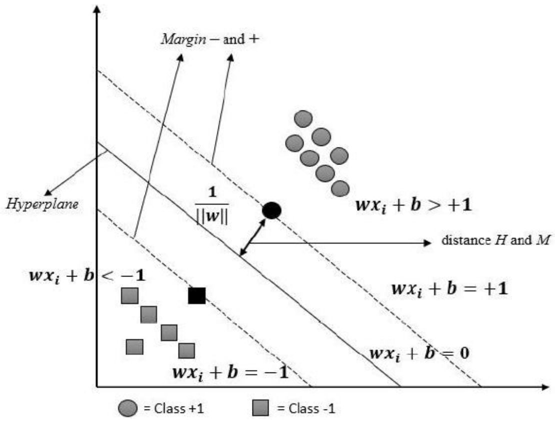

Figure 1. SVM Concept Illustration

The SVM concept is described as an attempt to find the best hyperplane that functions as a separator for two classes -1 and +1. The best separating hyperplane between the two classes is found by measuring the hyperplane margins and finding the maximum point. Trying to find the location of the hyperplane is the core of the learning process in SVM. The hyperplane equation assumes that both classes -1 and +1 can be perfectly separated by the d-dimensional hyperplane defined in equation (8).

$$(w^T x_i + b) = 0 \quad (8)$$

The pattern which belongs to class -1 (negative), can be formulated as a *pattern* that satisfies (9).

$$(w^T x_i + b) < -1 \qquad (9)$$

While the pattern which belongs to class +1 (positive), is formulated by inequality (10).

$$(w^T x_i + b) > +1 \qquad (10)$$

The most significant margin can be found by maximizing the value of the distance between the hyperplane and its closest point. This can be formulated as a *Quadratic Programming* (QP) *problem*, namely finding the minimum point of equation (11) by constraints of equation (12).

$$\min \tau(w) \; = \frac{1}{2}||w|^2 \qquad (11)$$

$$(w^T x_i + b)y_i - 1] \geq 0, \nabla_i \qquad (12)$$

This problem can be solved by *Lagrange Multiplier*

$$L(w, b, \alpha_i) = \frac{||w||^2}{2} + \sum_{i=1}^{m} \alpha_i \left[(w^T x_i + b)y_i - 1\right] \qquad (13)$$

with i =(1,2,3,…m)

α_i is a *Lagrange multipliers*, which are zero or positive . The optimal value of equation (13) can be calculated by minimizing *L* respect to w^T and b.

In summary, after the final solving of the problem we will get the result like equation (14)

$$\text{Maximize} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x^T{}_i x_j \qquad (14)$$

$$\text{Subject to} \quad \sum_{i=1}^{m} \alpha_i . y_i = 0, \qquad \alpha \geq 0$$

2.3 Negation Handling

The key to sentiment analysis is the determination of polarity, which determines whether the sentiment is positive or negative. Negation is part of linguistics that will change the value of the polarity of the text (Dadvar et al., 2011). Therefore the handling of negation becomes very important and determines the accuracy of text classification.

In Indonesian, researchers have different opinions in determining the constituent negation. Generally, negation constituents include "tidak", "tak", "bukan", "jangan", "tanpa", and "belum" [23]. Indonesian word negation generally comes in the form of syntactic negation; which reverses the word sentiment. Previous study on negation was performed by [18], [24] and [19] using the static window, punctuation mark, and POS methods. However, static window method and punctuation mark method caused an increase in the number of unnecessary features, because the words added to the new feature were not restricted. Words are added without considering whether they are affected by the negation or not. Moreover, in applications reviews such as myPertamina, the sentences used usually are not in standard form, even in slang words like "okay bossque". Review in slang sentences does not have a structure that makes it difficult to apply the POS method. Therefore we prefer to use a lexicon-based approach, because the Lexicon approach is better for unstructured comment [25].

Currently, the methods that have been widely used in handling lexicon-based negation are the First Sentiment Word (FSW) method, which is by flipping the sentiment exactly one word after the word negation, and the Fixed Window Length (FWL) method, which is reversing the sentiment of n words after the word negation [24] .

3.0 METHODOLOGY

This research was conducted through 3 stages, namely data collection, pre-processing and data analysis. The flowchart of the research process is presented in Figure 2.

3.1 Data Collection

The data used in this research was crawled from MyPertamina user reviews posted on Google PlayStore. The collected data still contains unnecessary data, such as id-user, date of posting, photo caption, etc. **Data cleaning** is meant to remove unnecessary information so that only comments are left that will be processed in the next step. Dataset used Indonesian Language that have been labelled as positive and negative. The **data labelling** process applies a lexicon-based algorithm. The list of lexicons used is 10,218 lexicons consisting of 3.609 positive lexicons and 6.609 negative lexicons [25]. From the positive lexicons and negative lexicon we selects only lexicons that come from a single word.

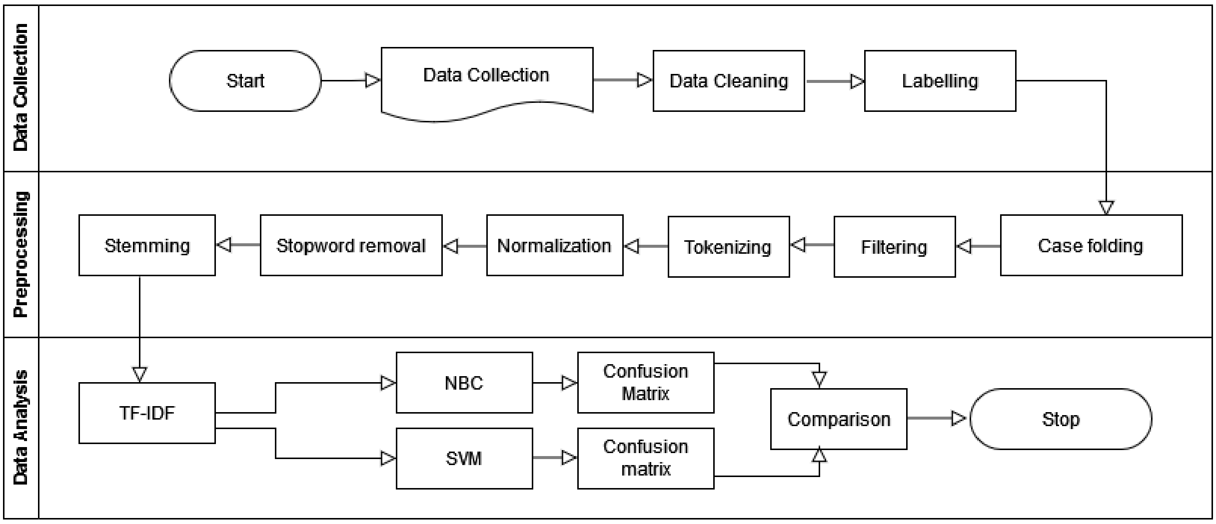

Figure 2. Flowchart of Research Steps

3.2 Preprocessing

Preprocessing is very important in sentiment analysis, because preprocessing turning raw data into clean data that ready for sentiment analysis processes. The preprocessing steps consist of:

1) *Case folding*: Because not all text documents are letter-consistent, this process change the letter characters in the comment to all lowercase characters.
2) *Filtering*: In this process adjustments are made by removing special characters such oter characters ($,%,*, and so on). This process also eliminates words that do not match the parsed results. For example usernames that start with the symbol "@" or hashtags "#".
3) *Tokenizing:* Tokenizing process break the review down into word units. The tokenizing process is carried out by looking at every space in the review. Based on these spaces the words can be separated.
4) *Normalization:* The process of converting non-standard words into standard words. For example the words 'nggak', 'tdk' will be converted into 'tidak', the word 'dpt' will converted into 'dapat'.
5) *Stopword removal :* Stopword is a word that appears frequently and has no informational value because it cannot distinguish documents, such as 'ini', 'yang', 'dari', 'ke' etc. This stage serves to eliminate words that have no influence in the later classification process.
6) *Stemming:* This step converts the word to its root word with the goal of lowering the dimensionality of the words in the collection.

3.3 TF-IDF

TF-IDF is feature extraction technique to measure the weighting of words in a document. The measure called *term frequency-inverse document frequency* (TF-IDF) is defined as tf_{ij}*idf_i , where tf_{ij} denote the number of occurrences of term t_i in document d_j , and idf_i denote the inverse document frequency of term i. If **N** is the total number of documents in the collection and **idf_i** is the number of documents containing term i, then the **TF-IDF** weighting can be written as equation (1)[26].

$$w_{i.j} = tf_{ij}xlog(\frac{N}{df_i}) \quad (1)$$

3.4 Data Labelling

The dataset taken from scrapping and cleaning process is still a raw data that does not yet have a target class, so it is necessary to label the dataset. The process of labeling dokumen is done by lexicon-based approach. We use a list of positive laxicons such as Table 4 and a list of negative lexicons such as Table 5. For data labeling, 3 algorithms are used, namely lexicon-based labeling without negation, lexcon-based labeling with FSW method negation and labeling with FWL method negaso. Algorithms are presented in algorithm 1, algorithm 2 and algorithm 3. This method detects the negation word in the text and reverses the sentiment of the word that follows it. For FSW we reverse the first word sentimet, while for FWL it reverses n'th sentiment word behind the negation word.

Table 4. Pos-Lexicon Library

Word	Sentiment weight
puas	+1
nyaman	+1
ok	+1
lancar	+1
..	..
hebat	+1

Table 5. Neg-Lexicon library

Word	Sentiment weight
parah	-1
payah	-1
jelek	-1
ribet	-1
..	..
susah	-1

Algorithm 1 **Find_Label_Of_Dokumen**
Input : dokumen , a list of *Pos_Lexicon_List*, *Neg_Lexicon_List*
Output : lebel of dokumen

```
Count_Pos=0; count_Neg=0;
foreach w in word_of_Doc_List do
        If w in Pos_Lexicon_List do
                count_Pos =count_Pos +1
            else
                If w in Neg_Lexicon_List do
                        count_Neg=count_Neg-1
                endif
          endif
If  (count_Pos + count_Neg) > 0 do
        return 'positive'
      else
        retun 'negative'
 endif
```

Algorithm 2 **Find_Label_Of_Dokumen_with_Negation_Handling_FSW**
Input : documen , *Pos_Lexicon_List*, *Neg_Lexicon_List, NEG_word_List*
Output : lebel of dokumen

```
word_of_Doc_List =documen.split()  // split dokumen into list
Count_Pos=0; count_Neg=0; n=len(word_of_doc_List)
for i  in rangen (n)  do
        If  word_of_Doc_List[i]  in NEG_word_List  do
        If  word_of_Doc_List[i+1]  in Pos_Lexicon_List do
                  count_Pos =count_Pos -1
            Else
                If word_of_Doc_List[i+1}  in Neg_Lexicon_List do
                        count_Neg=count_Neg+1
                endif
          endif
If  (count_Pos + count_Neg) > 0 do
```

```
        return 'positive'
    else
        retun 'negative'
endif
```

```
Algorithm 3   Find_Label_Of_Dokumen_with_Negation_Handling_FWL
      Input   : documen , Pos_Lexicon_List, Neg_Lexicon_List,
                NEG_word_List, L
     Output   : lebel of dokumen

word_of_Doc_List =documen.split()  // split dokumen into list
Count_Pos=0; count_Neg=0; n=len(word_of_doc_List)
for i  in rangen (n)  do
        If  word_of_Doc_List[i]  in NEG_word_List  do
        If  word_of_Doc_List[i+L]  in Pos_Lexicon_List do
                count_Pos =count_Pos -1
          Else
              If word_of_Doc_List[i+L}  in Neg_Lexicon_List do
                    count_Neg=count_Neg+1
              endif
         endif
If  (count_Pos + count_Neg) > 0 do
        return 'positive'
    else
        retun 'negative'
endif
```

3.5 Evaluation

To evaluate the performance of the classification algorithm, the confision matrix is arranged as Figure 3:

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure 3. Confusion Matrix of classification results

From the confusion matrix we derive various classification performance parameters, including:

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (17)$$

To compare the performance of two algorithms we combine Precision and Recall in one measure, namely as F1 formulated in equation (18). F1-score helps to measure Recall a nd Precision at the same time.

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} \quad (18)$$

4.0 RESULANTS

4.1 Preparing Dataset

The data collection process is carried out using scrapping techniques on the google playstore (https://play.google.com/store/apps/details?id=com.dafturn.mypertamina). The data taken in this study is part of consumer reviews and ratings. The code for scrapping process can be seen in Figure 3.

```
!pip install google-play-scraper
from google_play_scraper import Sort,reviews
result,continuation_token=reviews('com.dafturn.mypertamina',lang='id
',country='id',sort=Sort.NEWEST,count=1000,filter_score_with=None)
```

Figure 3. Code for Scrapping Data

The data collected from scrapping is still contains attributes that may not be needed for data analysis, such as *reviewId*, *userName*, *userImage*, etc (see Table 2). For this reason, a data cleaning process is needed so that only what we need is left, namely content. The program code to drop attributes that are not needed is as shown in Figure 4 below. Review data consisting of content is stored in a csv file as formatted in table 3.

```
dataset.drop(['reviewId','userName','userImage','score','thumbsUpCount'
,'reviewCreatedVersion','at','replyContent','replyAt','appVersion'])
```

Figure 4. Code for Dropping Unnecessary Attributes

Table 2. Sample of Original Dataset

reviewId	a1f9a3fe-9d7b-4136-b3c7-d091e36a718f
userName	Reffi Siregar
userImage	https://play-lh.googleusercontent.com/a-/ALV-UjVPU4YrdhJFoMahlt226mjinn3B1nzZZRxAoQ3y3m-wq7U
content	Dah isi banyak dgn harapan point dapat banyak juga error sistemnya... Sama sekali ga bisa di akses... Kaya maen maen Tp sewaktu isi dikit ok ok aja ga ada masalah... Jadi kaya aneh gitu...
score	1
thumbsUpCount	0
reviewCreatedVersion	4.2.3
at	10/25/2023 3:54
replyContent	
repliedAt	
appVersion	4.2.3

Table 3. Sample of Dataset Save in CSV

No	reviews
1	Aplikasi bagus tapi banyak errornya
2	Dah isi banyak dgn harapan point dapat banyak juga error sistemnya... Sama sekali ga bisa di akses... Kaya maen maen Tp sewaktu isi dikit ok ok aja ga ada masalah... Jadi kaya aneh gitu...
3	Update terossss lg dk kios malah nunggu update bangke
4	Program cukup bisa membantu
5	smoga beruntung mendapat hadiah.. tukar poin

After the application of the data labeling algorithm, we will get reviews that have been labeled. Examples of rivew that have been labeled are presented in Table 4.

Table 4. Sentiment Review after Labelling

No	Reviews	Sentiment
1	Aplikasi bagus tapi banyak errornya	Negative
2	Dah isi banyak dgn harapan point dapat banyak juga error sistemnya... Sama sekali ga bisa di akses... Kaya maen maen Tp sewaktu isi dikit ok ok aja ga ada masalah... Jadi kaya aneh gitu...	Negative
3	Update terossss lg dk kios malah nunggu update bangke	Negative
4	Program cukup bisa membantu	Positive
5	smoga beruntung mendapat hadiah.. tukar poin	poistive

The results of labeling data using three algorithm produce the following dataset (Table 5).

Table 5. Dataset for Classification

No	Reviews	Negative	Positive	Total
1	Without Negation Handling	398	602	1000
2	With Negation Handling FSW	390	610	1000
3	With Negation Handling FWL n=2	385	615	1000

4.2 Pre-Preprocessing

Pre-processing was done on datasets that have been labeled positive and negative using a lexikon-based algorithm. As an illustration if we have a review as follows:

"Aplikasi ribet bikin pusing, sdh lama pakai app ini dan sdh dpt kode QR...stelah di UPDATE malah suruh daftar ulang..GIMANA?! "

The preprocessing steps can be seen in Table 6.

Table 6. The Proprocessing Step and Results

Preprocessing step	Results
Casefolding	aplikasi ribet bikin pusing, sdh lama pake app ini dan sdh dpt kode qr...stelah diupdate malah disuruh daftar ulang..gimana?!
Filtering	aplikasi ribet bikin pusing sdh lama pake app ini dan sdh dpt kode qr stelah diupdate malah disuruh daftar ulang gimana
Tokenizing	[aplikasi],[ribet],[bikin],[pusing],[sdh], [lama], [pake], [app], [ini], [dan], [sdh], [dpt],[kode],[qr],[stelah],[diupdate],[malah],[disuruh], [daftar], [ulang],[gimana]
Normalization	[aplikasi],[ribet],[bikin],[pusing],[sudah], [lama], [pakai], [app], [ini], [dan], [sudah], [dapat],[kode],[qr],[setelah],[diupdate],[malah],[disuruh], [daftar], [ulang],[gimana]
Stopword Removal	[aplikasi],[ribet],[bikin],[pusing],[lama],[pakai],[app],[dapat],[kode],[qr],[update],[malah], [disuruh], [daftar], [ulang],[gimana]
Stemming	[aplikasi],[ribet],[bikin],[pusing],[lama],[pakai],[app],[dapat],[kode],[qr],[update],[malah], [suruh], [daftar], [ulang],[gimana]

The final step in Pre-processing is to compile a list of tokens for all documents in training. Furthermore,from the list of tokens, a term document matrix is created that records the frequency of occurrence of terms in the document. The TF document-term matrix is presented in Table 6. From Matrix TF, we converted into normalized TF-IDF matrix in Table 7. This final matrix is then ready to be classify using NBC and SVM algorithm.

Table 6. TF-IDF Weighting of Document

Doc	aplikasi	sangat	bantu	ribet	ok	bbm	daftar	pusing	…	sulit	class
Doc-1	1	1	1	0	0	0	0	0	0	0	+1
Doc-2	2	0	0	1	0	0	0	0	0	1	-1
Doc-3	0	0	0	1	0	0	1	1	0	0	-1
…											
Doc-n	1	0	0	0	1	0	0	0	0	0	+1

Table 7. Normalized TF-IDF Weighting of Document

Doc	aplikasi	sangat	bantu	ribet	ok	bbm	daftar	pusing	…	sulit	class
Doc-1	0.5775	0.5775	0.577	0	0	0	0	0	0	0	+1
Doc-2	0.6220	0	0	0.311	0	0	0	0	0	0.3110	-1
Doc-3	0	0	0	0.422	0	0	0.422	0.4223	0	0	-1
…											
Doc-n	0.7071	0	0	0	0.707	0	0	0	0	0	+1

4.3 Analysis and Evaluation

Classification is carried out using two classification methods, namely the NBC and SVM methods. The classification results are the result of various combinations of training data and testing data, namely a combination of traning:testing, i.e 60:40, 70:30, 80:20, and 90:10. Each combination will produce a confusion matrix from which evaluation parameters are set, namely precision, recall, accuracy and F1 value. Table 8 shows the comparison of performance between Non-Negation Handling and With Negation Handling (FSW) for NBC algorithm. A comparison of the FWL algorithm with classification using NBC is presented in table 9 for n=2 and n=3.

Table 8. Comparison of Algorithm Performance (NBC)

Combination Training:Testing	Without Negation Handling				With Negation Handling FSW			
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
60:40	67.2%	**68.4%**	62.5%	67.8%	69.4%	63.8%	64.8%	66.5%
70:30	**68.3%**	65.2%	63.7%	66.7%	**69.7%**	67.8%	65.5%	68.7%
80:20	68.2%	67.2%	**65.6%**	**66.7%**	69.2%	**67.2%**	**68.1%**	**68.2%**
90:10	64.7%	65.7%	64.1%	65.2%	65.6%	66.8%	66.7%	66.2%

Table 9. Comparison of Algorithm Performance (NBC) Using FWL

Combination Training:Testing	With Negation Handling FWL n=2				With Negation Handling FWL n=3			
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
60:40	65.3%	**66.2%**	61.5%	65.7%	64.2%	64.2%	62.3%	64.2%
70:30	67.2%	65.3%	64.3%	66.2%	65.2%	**66.3%**	63.4%	65.7%
80:20	67.7%	64.3%	**65.8%**	**66.0%**	66.2%	61.2%	**64.3%**	**63.6%**
90:10	63.2%	64.5%	63.8%	63.8%	59.5%	63.2%	66.7%	61.3%

SVM classification performance is presented in Table 10 and Table 11 for comparison without Negation handling and with negation handling (FSW) and comparison between negation handlling FWL with n=2 and n=3

Table 10. Comparison of Algorithm Performance (SVM)

Combination Training:Testing	Without Negation Handling				With Negation Handling FSW			
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
60:40	69.1%	67.9%	69.4%	68.5%	72.3%	70.2%	64.8%	71.2%
70:30	71.2%	68.3%	70.3%	69.7%	71.3%	73.2%	65.5%	72.2%
80:20	**72.3%**	**70.4%**	**72.3%**	**71.3%**	**74.3%**	75.1%	**75.2%**	**74.7%**
90:10	67.2%	66.5%	67.4%	66.8%	65.6%	68.5%	71.2%	67.0%

Table 11. Comparison of Algorithm Performance (SVM) Using FWL

Combination Training:Testing	With Negation Handling FWL n=2				With Negation Handling FWL n=3			
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
60:40	70.2%	61.5%	69.8%	67.8%	66.8%	62.3%	67.3%	70.2%
70:30	68.3%	64.3%	69.2%	68.2%	67.3%	63.1%	67.7%	68.3%
80:20	**69.3%**	**65.8%**	**70.2%**	**69.3%**	**68.2%**	**64.3%**	**68.7%**	**69.3%**
90:10	66.4%	63.8%	67.3%	65.2%	63.2%	66.7%	64.2%	66.4%

It can be seen from Table 8, Table 9, Table 10 and Table 11 that the parameter will reach a maximum mostly in combinations of 80:20. Therefore, we take a summary of the comparison of accuracy and F1 from a combination of 80:20. The results are as in table 12.

Table 12. Summary Comparison of Classification Performance

Dataset	NBC		SVM	
	Accuracy	F1	Accuracy	F1
Without Negation Handling	65.6%	66.7%	72.3%	71.3%
With Negation Handling FSW	**68.1%**	**68.2%**	**75.2%**	**74.7%**
With Negation Handling FWL n=2	65.8%	66.0%	70.2%	69.3%
With Negation Handling FWL n=3	64.3%	63.6%	68.7%	69.3%

From table 12 we concluded that the application of negation handling has increased the classification performance, both NBC and SVM algorithms. The NBC algorithm has increased accuracy by 2.5% and F1 by 1.5%, while the SVM algorithm has increased accuracy by 2.9 and increased F1 by 3.4%.

5.0 CONCLUSION

5.1 Conclusion

The conclusion of this study is that negation handling has been able to increase the performance of the classification algorithm, either FSW or FWL. In the application of the FWL algorithm, the use of n=2 results in better performance compared to n=3. In general, the performance parameters of the FWL method are still less than the FSW method. Compared with classification without negation handling, the NBC algorithm has increased accuracy by 2.5% and F1 by 1.5%, while the SVM algorithm has increased accuracy by 2.9% and increased F1 by 3.4%.

5.2 Recommendations

Although this study has shown that the handling of negation words has been proven to improve classification performance, the average accuracy has not been too high. This is possible because there is too many comments in the form of informal expressions such as "crash trus, g bsa d gnakan". Data like this was addressed in step 4 pre-processing (normalization) to change nonstandard words like 'tdk' or 'gak'', but too many variations of unstructured forms. In the next study, better normalization procedures can be proposed so that sentences become more easily detected in the library of sentiment word.

REFERENCES

[1] U. Khaidarni, A. H. Fauzi, H. Nisa, R. G. Gumelar, and A. Muldi, "Analisis Difusi Inovasi Terhadap Aplikasi Mypertamina," *Jurnal Ilmiah Wahana Pendidikan*, 2023.

[2] F. Setya Ananto and F. N. Hasan, "Implementasi Algoritma Naïve Bayes Terhadap Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store," *Jurnal ICT : Information Communication & Technology*, vol. 23, no. 1, 2023.

[3] A. M. Ginting, "KEBIJAKAN PEMBATASAN KONSUMSI BBM BERSUBSIDI MELALUI APLIKASI MYPERTAMINA," *KAJIAN SINGKAT TERHADAP ISU AKTUAL DAN STRATEGIS*, vol. 14, no. 13, 2022.

[4] R. A. Softina, F. M. Amin, and N. Wahyudi, "Analisis Faktor yang Mempengaruhi Innovation Resistance dan Intention to Use Terhadap Penerapan Pembayaran Non

Tunai," *JURNAL SISTEM INFORMASI BISNIS*, vol. 12, no. 1, 2022, doi: 10.21456/vol12iss1pp26-35.

[5] M. A. Abdurrazzaq, "Analisis Ulasan Aplikasi MyPertamina Menggunakan Topic Modeling dengan Latent Dirichlet Allocation," *KALBISCIENTIA Jurnal Sains dan Teknologi*, vol. 10, no. 1, 2023, doi: 10.53008/kalbiscientia.v10i1.694.

[6] A. A. Sinurat, C. Hendriyani, and F. Damayanti, "MyPertamina Application To Increase Consumer Engagement," *The International Journal of Business Review (The Jobs Review)*, vol. 5, no. 1, 2022, doi: 10.17509/tjr.v5i1.48470.

[7] A. U. Hasanah, B. Waspodo, and E. Rahajeng, "Analysis of MyPertamina Application User Satisfaction Using End User Computing Satisfaction Method," *Journal of Software Engineering Ampera*, vol. 4, no. 1, 2023, doi: 10.51519/journalsea.v4i1.375.

[8] C. G. Indrayanto, D. E. Ratnawati, and B. Rahayudi, "Analisis Sentimen Data Ulasan Pengguna Aplikasi MyPertamina di Indonesia pada Google Play Store menggunakan Metode Random Forest," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 3, 2023.

[9] Nabilla Saumi Putri, "Analisis sentimen review aplikasi mypertamina pada twitter menggunakan metode naïve bayes classifier," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 4, no. 1, 2023, doi: 10.37859/coscitech.v4i1.4789.

[10] R. Maulana, A. Voutama, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC," *Jurnal Teknologi Terpadu*, vol. 9, no. 1, 2023, doi: 10.54914/jtt.v9i1.609.

[11] Gilbert, Syariful Alam, and M. Imam Sulistyo, "ANALISIS SENTIMEN BERDASARKAN ULASAN PENGGUNA APLIKASI MYPERTAMINA PADA GOOGLE PLAYSTORE MENGGUNAKAN METODE NAÏVE BAYES," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 2, no. 3, 2023, doi: 10.55123/storage.v2i3.2333.

[12] N. Yusliani, M. Diana MARIESKA, E. Lestari, M. Ridho Putra SULFA, and W. Arimurti, "The Development of Indonesian Sentiment Analysis with Negation Handling," 2020.

[13] M. Mustasaruddin, E. Budianita, M. Fikry, and F. Yanto, "Klasifikasi Sentiment Review Aplikasi MyPertamina Menggunakan Word Embedding FastText dan SVM (Support Vector Machine)," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 3, 2023, doi: 10.30865/json.v4i3.5695.

[14] S. L. Ting, W. H. Ip, and A. H. C. Tsang, "Is Naïve Bayes a Good Classifier for Document Classification?," 2011. [Online]. Available: https://www.researchgate.net/publication/266463703

[15] B. Heerschop, P. Van Iterson, A. Hogenboom, F. Frasincar, and U. Kaymak, "Analyzing sentiment in a large set of Web data while accounting for negation," in *Advances in Intelligent and Soft Computing*, 2011, pp. 195–205. doi: 10.1007/978-3-642-18029-3_20.

[16] A. Hogenboom, P. Van Iterson, B. Heerschop, F. Frasincar, and U. Kaymak, "Determining negation scope and strength in sentiment analysis," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2011, pp. 2589–2594. doi: 10.1109/ICSMC.2011.6084066.

[17] V. P. Ramadhan, P. Purwanto, and F. Alzami, "Sentiment Analysis of Community Response Indonesia Against Covid-19 on Twitter Based on Negation Handling," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 7, no. 2, pp. 161–168, Jun. 2022, doi: 10.22219/kinetik.v7i2.1429.

[18] R. Amalia, M. A. Bijaksana, and D. Darmantoro, "Negation handling in sentiment classification using rule-based adapted from Indonesian language syntactic for Indonesian text in Twitter," in *Journal of Physics: Conference Series*, 2018. doi: 10.1088/1742-6596/971/1/012039.

[19] T. G. Prahasiwi and R. Kusumaningrum, "Implementation of negation handling techniques using modified syntactic rule in Indonesian sentiment analysis," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2019. doi: 10.1088/1742-6596/1217/1/012115.

[20] A. R. Hamzah, "Sentiment Analysis untuk Opini Akademik Menggunakan Naive Bayes Classifier dan Information Gain," *Seminar Nasional Call For Paper Fakultas Teknik Universitas Batik Surakrta*, pp. 1–22, 2021.

[21] D. I. Sumantiawan, J. E. Suseno, and W. A. Syafei, “Sentiment Analysis of Customer Reviews Using Support Vector Machine and Smote-Tomek Links For Identify Customer Satisfaction,” *J. Sistem Info. Bisnis*, vol. 13, no. 1, pp. 1–9, Jun. 2023, doi: 10.21456/vol13iss1pp1-9.

[22] M. Dadvar, C. Hauff, and F. De Jong, “Scope of Negation Detection in Sentiment Analysis.” [Online]. Available: http://wwwhome.ewi.utwente.nl/~dadvarm/dir2011/negation.txt

[23] Sudaryono, *Negasi dalam bahasa Indonesia : Suatu Tinjauan Sintaksis Semantik, Jakarta: Depdiknas.* Jakarta: Depdiknas , 1992.

[24] A. M. Ningtyas and G. B. Herwanto, “The Influence of Negation Handling on Sentiment Analysis in Bahasa Indonesia,” in *Proceedings of 2018 5th International Conference on Data and Software Engineering, ICoDSE 2018*, Institute of Electrical and Electronics Engineers Inc., Jul. 2018. doi: 10.1109/ICODSE.2018.8705802.

[25] F. Koto and G. Y. Rahmaningtyas, “Inset lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs,” in *Proceedings of the 2017 International Conference on Asian Language Processing, IALP 2017*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 391–394. doi: 10.1109/IALP.2017.8300625.

[26] W. B. Trihanto, R. Arifudin, and A. Muslim, “Information Retrieval System for Determining The Title of Journal Trends in Indonesian Language Using TF-IDF and Naïve Bayes Classifier,” *Scientific Journal of Informatics*, vol. 4, no. 2, pp. 2407–7658, 2017, [Online]. Available: http://journal.unnes.ac.id/nju/index.php/sji