

Implementation of Naive Bayes Algorithm on Sentiment Analysis Application

Martiti^{1,*} Christina Juliane¹

¹ Informatics Engineering Department, STMIK AMIK Bandung, Indonesia

*Corresponding author. Email: martiti@stmik-amikbandung.ac.id

ABSTRACT

Natural Language Processing (NLP) is one of the branches of artificial intelligence science where this branch science is the basis for developing sentiment analysis. The application in NLP in sentiment analysis includes Pre-processing text consisting of featured selection and tokenization. For the classification process, the determination of the algorithm is determined by comparing the results of the classification predictions of naïve Bayes, Weighted Instances, and Zero-R with the data that has been calculated for its frequency terms. The results of the testing analysis showed naïve Bayes had a stable accuracy after being tested with an accuracy value of 99.62% in the training data and 94% in the Testing Data, with an average classification failure of 0.13%. The results of the acquisition of words are used as a corpus for the construction of sentiment-level sentence analysis applications. The application development by the Naive Bayes algorithm was built using the PHP programming language and literary library. The method of application development using the waterfall starts from the analysis process to the application implementation. Based on testing the accuracy of 30 comments classified by the system, it produces an accuracy value of 86.66%. However, the accuracy of comments that have been classified as applications were retested using machine learning *Weka* which resulting in an accuracy value of 93.33%. The difference in accuracy is due to the Naive Bayes algorithm in utilizing the appearance of words to form a sentiment classification.

Keywords: *Opinion, sentiment, Pre-processing, TF-IDF, Naïve Bayes, Zero-R, Weighted Instance, Sastrawi.*

1. INTRODUCTION

Sentiment analysis or opinion mining is one technique that can be used to obtain a general picture of public perceptions of service quality in the form of services or products that tend to be positive, negative, or neutral [1]. However, to know the sentiments of a sentence, it requires the knowledge of sentence meaning and word function forming so that there is no multi-interpretation. On the other hand, a sentence or paragraph cannot be separated from ambiguous words when the sentence has a double meaning. This will certainly make it difficult to form the sentiment classes for sentence level. With so many words in Indonesian language, it will be difficult to recognize the sentiment pattern of each sentence formed by the word. Therefore, an application can do the data preparation so that the pattern of each sentence can be known and can be classified correctly by using the classification algorithm.

In 2014, Ahmad Fathan conducted a sentiment analysis study of public figures in Indonesia [2]. The object of his research is a tweet that discusses public figures to know the topics related to public figures or events that cause public sentiment to fall. The names of public figures analyzed were public figures with the highest survey results based on surveys of competent institutions in Indonesia. The Naïve Bayes classifier is used to obtain classification results from sentiments regarding public figures. Twitter data is classified by using the Naive Bayes method and the selected features are unigram, negation, and term-frequency.

The results of this study are the classification model and the results of the classification of tweet sentiments towards positive or negative. The results of classification testing accuracy with the term frequency feature were 79.91% while the TF-IDF feature was 79.68%. The classification using the RapidMiner tools with Naive Bayes and the term frequency feature was 73.81% while the TF-IDF feature was 71.11%.

The test results on the application built and on the RapidMiner tools show that accuracy with the term frequency feature gives better accuracy results than the accuracy with the TF-IDF feature [3].

Based on the previous research, this study tries to build a sentiment-level sentiment analysis application where this application has an accuracy value that has a similar accuracy value with Weka's machine learning. In addition, the application can classify the ambiguous sentences that are difficult to classify into positive, negative, or neutral classes where the accuracy value reaches 80%. Sentiment classification in this study uses the Naive Bayes method, where sentiment for training data comes from the public comments on the official social media account of Grab Indonesia. To test the accuracy of the application built, ambiguous sentences are used and have multiple interpretations. The classification model in this study uses word weighting techniques in which the research focuses on the unigram features and is calculated based on term frequency (Laplace smoothing).

2. RELATED WORK

In 2016, Syahmia Gusriani analyzed the research about the sentiment analysis based on public comments on online stores on Zalora and Berrybenka official social accounts on Facebook. The methodology starts from data collecting, pre-processing, feature selection, classification, and measurement accuracy. The Naive Bayes, KNN, and decision tree classification methods are used to compare the best classification prediction results through the KNIME tools. The results of the test analysis show Naive Bayes has a stability of accuracy after being tested with a minimum support value of 0.036 in the Frequent Itemset. Naive Bayes has an average accuracy of 90.1% [4].

The result of this research is the classification of comments class on Zalora and Berrybenka online stores so that the community's tendency towards the store can be identified. Based on the comments data obtained from Zalora and Berrybenka fan pages, the comments tend to contain negative comments. Customers who are disappointed with the products and services of online stores tend to submit their complaints on social media rather than giving praise for the products or services that have been obtained.

In this research, we will use the Naive Bayes method with unigram feature extraction and use term frequency weighting where the words from the unigram feature extraction are used as the reference in shaping sentence-level analysis sentiment applications. The tools are used in comparing accuracy values before determining the algorithm that will be used in the application. Each class classification that contains comments will also be tested for its accuracy by using Weka machine learning. For the comparative value of accuracy, three algorithms namely

Naive Bayes, Zero-R, and Weighted Instance will be employed. The algorithm with the highest accuracy value will be used as the foundation in building the sentiment-level sentence analysis applications.

3. RESEARCH METHODOLOGY

The study began with crawling comment data from Grab Indonesia official account. The comment data is used as training data to form the Indonesian corpus that is used as the database to recognize words. Then, the training data is processed by using the NLP techniques, including pre-processing data, stemming, and tokenization to form the basics word of absence. Then the results of the NLP process are used in Weka machine learning. The classification algorithm used in machine learning is Zero-R, Naive Bayes, and Weighted Instance. After all, the data that has been calculated accurately by using the three algorithms will be compared to see the best algorithm which build the sentiment-level sentence analysis application.

The study began by taking data from Grab Indonesia's official Twitter account. The reason for choosing comments on the Grab service is that online transportation services are popular in Indonesia. Grab is one of the most frequently downloaded online transportation services applications the Google Play Store as the number of downloads is more than 100 million and has the official social media account that has the second most commented out among other transportation services with 66.9% presentation (quoted from the YKLI survey about the 2018 online transportation survey).

The training data is collected by filtering keywords related to service combined with sentiment words. The following provisions are used in selecting training data among thousands of crawling comment data:

1. Determining the target/entity
 - a. services which are related to food delivery services or known as *GrabFood*, delivery services, or shuttle services or known as *GrabNow*.
 - b. customer complaints with keyword-related services including *send, order, order, package, and clarification*.
 - c. questions about Grab services and their applications.
2. Determine the positive/negative sentiment keywords that accompany a comment sentence.
 - a. positive service sentiment such as *arrive, quickly, already, come, easy, thank you*.
 - b. Negative service sentiment such as *old, disappointed, slow, late, difficult, difficult, severe, lying, wrong, cancel, sorry*.

The following table displays the comment data from the crawling data on Grab's official Twitter account:

Table 1. Comment on crawling data

No	Comment
1	@Inanyah it's so awful, the driver makes too many excuses, the location is too far and then they asked to cancel it. In the end, I can't order it and the app can't be used because they make the pouring rain as an excuse to cancel the order. They said the reason is they don't know the address, or the location is too far. What's the use of the map?

The next step is labeling the document. The 457 comments are divided into three documents, namely D1 for positive documents, D2 for neutral documents, and D3 for negative documents. Then, the data will be divided into training data and testing data. The percentage of training data sharing is 80% for the commented data and 20% for the testing data. The training data is then labeled manually as being negative, positive, or neutral by looking at the sentiment words contained while data testing is not labeled. Table 2 shows the distribution of comments based on data that has been obtained:

Table 2. The distribution of comments

N	Comment	Number of comments	Classification		
			Positive	Neutral	Negative
1	Data Training	400	140	180	80
2	Data Testing	57	Don't have the label		

Training data is used by classification algorithms (Naive Bayes, Weighted Instance, Zero-R) to form a classifier model. This model is a representation of knowledge that will be used to predict new data classes that have never existed. Data labeling is divided into three classes: positive, negative, and neutral as shown in Table 3 below:

Table 3. Labeled comments

Doc	Comment	Assumptions
D1	There's no need to go to the bank just for savings or to transfer money, all can be done by <i>Ovo</i> . it's so efficient and has no administration fee.	Positive
D2	I need help, I sign to be a Grab partner online, but I always got	Neutral

Doc	Comment	Assumptions
	rejected. After all, my STNK number is different from my ID card because I bought a used bike. What should I do?	
D3	There are too many ads on the app... please fix it first. How can you give any information if you cannot be contacted? Thank you.	Negative

Data testing is used to measure the extent to which the classifier successfully classifies correctly. Therefore, the data in testing set should not be in the training set so that it can be seen whether the classifier model is "right" in doing the classification. Table 4 shows the testing data used in the study.

Before conducting the learning process of classification algorithms in Weka machine learning, the data must go through the NLP stage to find the class polarity in words. That stage includes number filter, punctuation, tokenization, and stemming.

Table 4. Data testing

No	Comments	Assumption
1	Please check the driver's exact phone number. Do not just give them the phone number from Grab. Almost 95% of grab drivers' number whether it is GrabCar or GrabBike is inactive!!! There are too many ads on the app... please fix it first. How can you give any information if you can't be contacted? Thank you	?

The results of the process numbering filter and punctuation using python above are the following table 5:

Table 5. The result of Number Filter and Punctuation

No	Comment
<i>Before Punctuation, Number Filter, and Case Folding</i>	
1	@robert123 yeeeeeah awesome only on @grab the athletes can prove that the #VictoryIsNear 1:0 WE WIN
<i>After Punctuation, Number Filter, dan Case Folding</i>	
1	Yeeeeeah awesome only on the athletes can prove that we win

To perform the process of stemming and normalization, the data is processed using the python 3

scripts. The results of the stemming and normalization process is in Table 6 below:

Table 6. Stemming's Result

No	Comments
<i>After Punctuation, Number Filter, dan Case Folding</i>	
1	What in the world? I see on my notification that they have promo until the effective date, from the moment I enter the promo code I cannot redeem the code. It's a lie, even though I never use any promo. Please fix it soon.
<i>After Stemming Processing</i>	
1	What word GrabFood their promo in notification to word until effective date from moment enter code promo cannot it's a lie even though never use any promo code

Furthermore, after going through the process of stemming, comment data will pass the word normalization process. The process of word normalization itself uses the KBBI dictionary accessed via the page <https://kbbi.web.id/>. The following table is the result of normalization from the results of stemming:

Table 7. The results of the normalization's process

No	Comments
<i>After Stemming Processing</i>	
1	What word GrabFood their promo on notification too word until effective date from the moment enter promo code cannot it's a lie even though never use any promo codes
<i>After Normalization Processing</i>	
1	What word Grab Food their promo on notification too word until effective date from moment enter promo code cannot it's a lie even though never use any promo code

Before entering the process of breaking words or tokenizing, comments must be cleared of words that have no meaning or influence in the comments. These words are grouped into stop words. Stop words usually contain pronouns, conjunctions, and conjunctions. The tokenizing process is done by forming the unigram token. To do the tokenization process and calculate the frequency of occurrence of words, it uses Python to remove the stop word.

From the results of data preparation obtained by unigram tokens, the following are the number of tokens that appears in the document both training data and testing data:

Table 8. Frequency of Token in the document

No	Data	Document	Number of words
1	Data Training	Positive Token	289 kata
		Neutral Token	546 kata
		Negative Token	967 kata
2	Data Testing	-	246 kata

Each document in the training data has become a token or word given a sentiment class. Class labeling is divided into three classes: positive classes, neutral classes, and negated classes. The word labeling technique uses basic mathematical multiplication operations, where positive is represented by the number (+1), negative is represented by the number (-1), and neutral is symbolized by the sign (0). Before labeling the document, the frequency terms of each document are being calculated based on the rules. If there are words in the document, then it is given the number 1. If it is not, then it would be given the number 0.

Word weighting is very influential in determining the similarity between documents and queries. If the weight of each word can be determined precisely, it is expected that the results of the calculation of the similarity of the text will produce a good ranking of documents. Weighting is done on data training and Data Testing. The weighting process uses the TF-IDF formula, where the result that will be used to process the accuracy is the Term Frequency of the word. From the results of weighting, the words obtained words that do not have meaning in sentences that are often called stop words. Stop word is obtained from the calculation of TF-IDF which is worth 0 (zero).

Because	Now	Ovo
Only	Already	Wallet
Why	Each	WKWKW
New	Always	hahaha
Driver	Location	yahoo
Far	Grab	hai
And	detc	I am

Figure 1 Example of stop word results from word weighting

To determine whether the algorithm is suitable for classification, the training data will be tested using Weka machine learning. The greatest accuracy results will be used as a reference for classifying Data Testing.

Training data will be tested using machine learning as many as 799 words. The number of words was obtained from the result of tokenization which eliminates TF-IDF stop word results. The amount of Data Testing that will be used is 246 words. There is no class in testing data but each word has a frequency term which will later be matched to the results of the accuracy in the training data.

The training data that contains tokens that are given TF (term frequency) weights are then processed into AIFF format.

There are three methods chosen to determine which method with the best accuracy results, namely Naive Bayes, Weighted Instance, and Zero-R methods. The tests use Cross-Validation for the value of K = 10 for 5 times.

Table 9. The comparison of accuracy of three algorithm results

K-FOLD	Method					Average
	10	10	10	10	10	
Naive Bayes	99.62 3 %	99.6 23%	99.62 3%	99.62 3%	99.6 23 %	99.6 23 %
Zero-R	72.04 %	72.0 75%	72.07 5 %	72.07 55%	72.0 75 %	72.0 75 %
Weighted	72.07 55%	72.0 75 %	72.07 55%	72.07 5 %%	72.0 75 %	72.0 75 %

Table 10. Accuracy Values by class

Method	Accuracy Result				
	TP Rate	FP Rate	Precision	Recall	F-measure
Naive Bayes	0,996	0,004	0,996	0,996	0,996
Zero-R	0,721	0,721	?	0,721	?
Weighted	0,721	0,721	?	0,721	?

Table 11. Classification Failure Values

Method	Classification Failure	
	percentage	Words
Naive Bayes	0.3774 %	3
Zero-R	27.9245 %	222
Weighted Instance	0.1258 %	222

From the results of testing the accuracy of the training data obtained by the algorithm, the greatest accuracy value is the calculation with Naive Bayes where the accuracy value of the training data is 99.62%. So, this study uses the Naive Bayes algorithm as a basis for making sentiment analysis applications.

4. RESULT

Sentiment classification using Naive Bayes algorithm is done through two stages: the learning process stage and the classification stage.

The Naive Bayes classifier learning process begins with classifying sentiment. To classify sentiments, the pre-processing data in Table 12 contained sixteen training data that were manually labeled by humans. Labeling comments are divided into three classes, namely positive class, neutral class, and negative class. Labeling based on adjectives contained in comments. If a sentence contains more positive adjectives, the comments are categorized into positive classes, but if the number of negative adjectives is higher, then they will be categorized into negative classes. Meanwhile, the neutral classes are obtained from the number of negative and positive adjectives.

Table 12. Data of labeled comment

No	Comment	Assumption
1	<i>Saving bank transfer no need to go to bank all can be done by Ovo so efficient and no administration fee</i>	Positive
2	<i>Step forward</i>	Positive

To calculate Naive Bayes, it is needed to get the result of tokenization data called attribute data. Table 13 displays the attributes to see the results of the probability model training.

Table 13. Token Result Attribute

Attribute	Code	Attribute	Code	Attribute	Code
Active	A1	New	A9	Expensive	A17
Good	A2	Excited	A10	Onward	A18

Table 13 is used as the basis for forming the tabular classification data, where sentences that have the above attributes will be marked 1 (one) and none will be marked 0 (zero). The sentences are classified based on class P (Positive), N (Negative), and Neutral. The following table 14 displays training data by showing the appearance of each attribute in the training data. Number 1 represents the attribute that appears, while number 0 represents the attribute that does not appear in the data.

Table 14. Training Data

	A1	A2	A18	A22	A23	A24	Sentiment
D1	0	0	0	0	0	0	P
D2	0	0	0	0	0	0	N

From table 14, we can calculate by using the Naive Bayes method as follows:

1. The probability of each comment on the document
To calculate the probability for each comment you can use the formula:

$$P(ci) = \frac{fd(ci)}{|D|} \quad (1)$$

Information:

Fd (ci) = Number of comments that belong to the ci class

|D| = Number of training data

With the formula (Michael, 2010), it can be obtained the probability value of each positive, negative, and neutral comments as follows:

- a. Positive probability

In Table 14, there are four positive comments from the total of sixteen training data. The probability values are:

$$P(Positive) = \frac{4}{16}$$

- b. Negative probability

In Table 14, there are seven negative comments out of sixteen training data. The probability values are:

$$P(Negative) = \frac{7}{16}$$

- c. Neutral probability

In Table 14, there are five neutral comments from a total of sixteen training data. The probability values are:

$$P(Negative) = \frac{5}{16}$$

2. The probability value of words in document P with *Laplace Correlation*

The calculation of word probability will use additive smoothing techniques. This technique is used to handle probability values of 0 (zero). From many data in the training set, in each calculation, the data is added by 1 (one) and will not make a significant difference in the estimated probability so it can avoid the case of a probability value of 0 (zero). By using the Laplace Correlation or additive smoothing technique, you can calculate the probability value of words in a document as follows:

$$P(a_i|v_j) = \frac{n_i+1}{n+|vocabulary|} \quad (2)$$

Information:

n_i = frequency of occurrence of the word a_i in the document v_j

n = many words in categories v_j, |vocabulary| = number of words in training

From the formula, we can know the probability value of words in positive, negative, neutral documents, namely word "active" appears 0 times in positive documents. If the total vocabulary containing adjectives used in the training document is twenty-four words and the number of words in the positive document is five words, the probability value of the word "active" in the positive document is:

$$P(active|positive) = \frac{0 + 1}{5 + 24} = 0,034482$$

The probability value of the word "active" if it appears 1 time in a negative document is:

$$P(active|negative) = \frac{1 + 1}{14 + 24} = 0,052632$$

The probability value of the word " active " if it does not appear in a neutral document is:

$$P(active|neutral) = \frac{0 + 1}{5 + 24} = 0,034482$$

The following Table 15 displays the results of calculating the probability of words in each document in training data.

Table 15. The Value of P

Occurrence of Words	Class		
	Positive	Negative	Neutral
Active	0,034482	0,0526	0,0312
Good	0,034482	0,02631	0,0625
Busy	0,034482	0,0526	0,0312

The classification process with the Naive Bayes classifier algorithm begins with test data that has gone through the pre-processing stage and then calculates the Vmap value of each word in the document by utilizing the word probability value of the document in table 15. Next, set the Vmap value on the document that has the maximum Vmap value. From the largest Vmap value, the document is categorized as negative, positive, or neutral.

Table 16. Data Testing

No	Comment	Comment after Pre-processing	Class
1	<i>I don't like upgrading the app. It takes too much time. It takes too much internet quota, and the result is just the same.</i>	<i>Like upgrade repeat takes time takes internet quota result same.</i>	?

Using the word probability values in Table 15, the Vmap value of each comment on the test data can be calculated by using the formula:

$$V_{MAP} = \operatorname{argmax}_v \sum_{j \in V} P(v_j) \prod_{i=1}^n P(a_i | v_j) \quad (3)$$

The following is a calculation of the Vmap value in the test data:

1. The sentence is *like update takes time takes internet quota result same*

a. Vmap (“positive”)

P(“don’t_like”|positive) P(“upgrade”|positive)
 P(“repeat”|positive) P(“takes”|positive)
 P(“time”|positive) P(“takes”|positive) P(“internet quota”|positive) P(“result”|positive)
 P(“same”|positive) =

$$\frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} \times \frac{1}{29} = \frac{1}{261} = 0,0038314$$

b. Vmap (“negative”)

P(“don’t_like”|negative) P(“upgrade”|negative)
 P(“repeat”|negative) P(“takes”|negative) P(“time”|negative) P(“takes”|negative) P(“internet quota”|negative) P(“result”|negative) P(“same”|negative) =

$$\frac{2}{38} \times \frac{2}{38} \times \frac{1}{38} \times \frac{1}{38} \times \frac{1}{38} \times \frac{1}{38} \times \frac{1}{38} \times \frac{1}{38} \times \frac{1}{38} \times \frac{1}{38} = \frac{4}{342} = 0,011695$$

c. Vmap (“neutral”)

P(“don’t_like”|neutral) P(“upgrade”|neutral)
 P(“repeat”|neutral) P(“takes”|neutral) P(“time”|neutral) P(“takes”|neutral) P(“internet quota”|neutral) P(“result”|neutral) P(“same”|neutral) =

$$\frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times \frac{2}{32} = \frac{2}{288} = 0,0069444$$

From the above Vmap calculation results, it is found that the negative Vmap value is greater than the positive Vmap and neutral Vmap. It can be concluded that these comments are classified into negative sentiments.

By conducting the learning process of the Naive Bayes classifier in the training data, it can also be known the sentence classification stage in the test data by using the results of the word probability of the training data.

The calculation of the probability of words to a sentence can affect the class of sentiment of a comment. The more often the word appears in a comment, the greater the probability of the word to the positive, negative, or neutral class is. From that value, the Vmap value on the test data that does not have a sentiment class can be determined. The appearance of a word in a

sentence or a term called term frequency is very influential in determining the sentiment class in a sentence.

Sentence level sentiment analysis application is built by using PHP programming language and Sastrawi library. The purpose of this application development is to facilitate the process of text mining, namely the process of pre-processing text to classify Indonesian sentences using the Naive Bayes algorithm sentence level. The features available in the application are features for the pre-processing process and sentence classification features. The pre-processing feature consists of converting sentences into basic words called stemming special words for Indonesian sentences. In addition to the stemming process, there is a tokenization feature, which is a feature to break sentences into words and delete words that have no meaning (stop word list). The following Table 17 shows the functional requirements of the system created.

Table 17. Application Functional Needs

Code	Description
Actor: User	
KF -01	Input for stemming
KF - 02	See the results of stemming sentences
KF - 03	Input sentences for tokenization
KF - 04	See the results of sentence tokenization
KF - 05	Input sentences for sentiment tests
KF - 06	See the results of the sentiment test

This application has the features of tokenization and stemming features. Next, the sentiment analysis application interface is presented below:

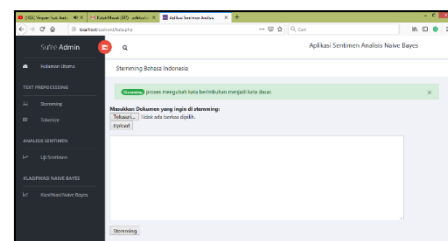


Figure 4 Page for stemming and tokenization



Figure 5 Page for classification

Accuracy testing was conducted to determine the level of accuracy of sentence or comment classification based on the applications made with manual

classification and machine learning classification. To do the accuracy testing, data testing and training data are needed. Training data is used as a reference when testing the accuracy of Weka's machine learning.

The following accuracy testing that will be carried out in the study includes:

1. testing the accuracy of the results of data testing classification according to the application of data testing that has been classified manually by 45 correspondents.
2. testing the accuracy of data testing classification results according to the application of the data testing that is tested by using Weka machine learning.

Based on the manual classification, the accuracy value of the application has been obtained. The appropriate number of sentences between applications made with manual classification is 26 sentences from a total of 30 test sentences. Then, the accuracy of the application of manual classification is as [2]

$$\text{Accuracy} = \frac{\text{The sum of true sentence}}{\text{The sum of all sentence}} \times 100 \% \quad (4)$$

$$\text{Accuracy} = \frac{26}{30} \times 100 \% = 86,66 \%$$

Based on the results of the classification of sentences carried out by the system, the appropriate number of sentences between the application and Weka's machine learning is 28 sentences (Sampling based on experienced), so the accuracy value of the application of Weka's machine learning is

$$\text{Accuracy} = \frac{28}{30} \times 100 \% = 93,33 \%$$

Based on the classification carried out by machine learning, the value of manual accuracy to the results of machine learning accuracy is the number of sentences that correspond between the results of machine learning classification with manual classification, which is 26 sentences from a total of 30 sentences testing.

$$\text{Accuracy} = \frac{26}{30} \times 100 \% = 86,66 \%$$

There are differences in the value of classification accuracy due to Naive Bayes algorithm in utilizing the probability of the appearance of words to classify a sentence.

Based on the research that has been done, it can be concluded that to be able to do sentiment classification analysis of sentence-level, the first stage is the process of pre-processing text. Data that must be prepared at the time of preparation is divided into two, namely training data and data testing. The training data used in this study is up to 457 sentences which were further broken down into words to determine the sentiment class. For training data tested using the Naive Bayes algorithm, the accuracy of 99.62% is obtained. So, with this accuracy, it made the

application of sentiment analysis can classify the sentences. The corpus used for applications is the corpus of training data results. Based on the test results, data testing using the built application obtained an accuracy of 86.66%. The difference in accuracy is due to the Naive Bayes algorithm using the probability of occurrence of words in the classification.

REFERENCES

- [1] Firmansyah, Dimas Ricky. (2013). Implementasi *Text Mining* Klasifikasi Objek Wisata Dengan Metode *Naive Bayes Classifier* di Dinas Pariwisata dan Kebudayaan Jawa Barat. Bandung : Universitas Komputer Indonesia.
- [2] Haddi, Emma., Liu, Xiaohui., & Yong Shi. (2013). *The Role of Text Pre-processing in Sentiment Analysis*. Netherlands : Elsevier B.V.
- [3] Hamzah, Amir. (2014). *Sentiment Analysis untuk Memanfaatkan Saran Kuesioner Dalam Evaluasi Pembelajaran dengan Menggunakan Naive Bayes Classifier*. Yogyakarta : IST AKPRIND.
- [4] Hemalatha, I., Varma, G. P Saradhi., & Govardhan, A. (2012). *Pre-processing the Informal Text of efficient Sentiment Analysis*. India : *International Journal of Emerging Trends & Technology in Computer Science*.
- [5] Hidayatullah, Ahmad Fathan. (2014). *Analisis Sentimen dan Klasifikasi Kategori Terhadap Tokoh Publik pada Twitter*. Yogyakarta : Universitas Gajah Mada.
- [6] Isah, Haruna. (2014). *Social Media Analysis for Product Safety using Text Mining and Sentiment Analysis*. Bradfor : University of Bradford.
- [7] Liu, Bing, Hu, Mingqing, and Cheng, Junsheng (2005). "Opinion Observer: Analyzing and Comparing Opinions on the Web." Proceedings of the 14th International World Wide Web Conference (WWW-2005), May 10-14, Chiba, Japan.
- [8] Hand, D.J., Mannila, JH, Smuth, P. (2001). *Principles of Data Mining*. London : The MIT Press.
- [9] Ian H. Witten, E.F. (2005). *Data Mining : Practical Machine Learning Tools and Techniques*, Second Edition. San Fransisco : Elsevier.
- [10] Adriani, M., Manurung, R. (2010). A survey of bahasa Indonesia NLP research conducted at.
- [11] Aster T. (2013). *Sentiment Analysis*. In T. Aster, Aster Analytics Foundation Guide (pp. 336-340). Teradata Corporation.
- [12] Putranti, Noviah Dwi., & Winarko, Edi. (2014). *Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan Maximum Entropy dan Support Vector Machine*. Yogyakarta : Universitas Gajah Mada.