Comparative Analysis of Naïve Bayes Algorithm Performance in English and Indonesian Text Sentiment Classification on Duolingo Application in Playstore

Andi Serlina[1], Abdul Rahim[2*], Arbansyah[3]

[1,2,3] Informatics Engineering, Science and Technology, Muhammadiyah University of East Kalimantan, Samarinda, East Kalimantan, Indonesia
E-mail: [1]2111102441008@umkt.ac.id, [2*]ar622@umkt.ac.id, [3]arb381@umkt.ac.id

(Received: 5 Feb 2025, revised: 24 Feb 2025, accepted: 25 Feb 2025)

Abstract

Text classification is an important topic in Natural Language Processing (NLP), especially when conducting research on user reviews on language learning apps such as Duolingo. This study compares the effectiveness of the Naïve Bayes algorithm in identifying sentiment in English and Indonesian reviews on the Duolingo app on Playstore. The approach includes data collection, text preparation (case folding, tokenization, stopword removal, and stemming), and Naïve Bayes algorithm evaluation for each dataset. Model performance was evaluated using accuracy, precision, recall, and F1-score. The Naïve Bayes method obtained 84% accuracy on the English dataset with a 90:10 data split and 67% accuracy on the Indonesian dataset with the same split ratio. The difference in the results obtained is due to several variables, including the use of informal language, slang, and more complicated word variants in Indonesian, which make proper classification more difficult for the model to achieve.

Keywords: Naïve Bayes, Sentiment Classification, Text Mining, Duolingo, NLP.

I. INTRODUCTION

Text classification is significant in Natural Language Processing (NLP), particularly when assessing user-generated material in various applications, including language learning sites like Duolingo [1] Duolingo is one of the most popular language-learning applications, with over 500 million downloads and 27 million user reviews globally [2][3]. The app's efficacy is aided by modern computational approaches, such as sentiment classification algorithms that evaluate user input. The Naïve Bayes algorithm is a popular choice for large-scale text classification due to its simplicity and computing efficiency [4][5]. However, its performance may be influenced by linguistic differences across languages. English and Indonesian have distinct structural variances and expression patterns, which might complicate sentiment classification. Given the rising usage of language learning applications such as Duolingo in the Play Store, it is vital to analyze how the Naïve Bayes algorithm performs in sentiment analysis in these two languages [6].

Sentiment analysis is a branch of research that uses Natural Language Processing (NLP), a technique for automatically extracting, understanding, and processing data in unstructured text to extract sentiment information in an opinion or opinion phrase [7]. This research will not only discuss sentiment analysis. However, it will focus on comparing the performance of the Naïve Bayes algorithm in classifying English and Indonesian user reviews on the Duolingo application in Playstore. This analysis aims to determine how well the Naïve Bayes method can operate well in multilingual situations. This will contribute to building apps more responsive to user requests from different language backgrounds.

This research compares the performance of the Naïve Bayes algorithm in sentiment classification for English and Indonesian texts. Although Naïve Bayes is extensively utilized for its simplicity and efficiency, its performance may vary based on linguistic structures and expression patterns. Previous research, such as Cognetta (2023), has shown that structural and linguistic variations between languages provide major hurdles for text classification [8]. Most research has focused on languages with comparable grammatical structures, creating a vacuum in knowing how Naïve Bayes works in languages with diverse morphological difficulties. For example, while Indonesian has a simpler sentence structure than English, it has complicated affixation, which may affect classification accuracy. This study compares Naïve

Bayes' sentiment classification performance for both languages, aiming to close the gap.

Text mining is an information exploration method where users interact with a collection of documents using data mining techniques, one of which is classification, whose purpose is to predict or categorize data into certain groups [9]. Several algorithms are employed for this purpose, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, and Naïve Bayes [10]. Naïve Bayes is one of them that is frequently used because of how well it handles text classification jobs [11].

Previous research titled “Text Classification: Naive Bayes Classifier with Sentiment Lexicon” developed an initial keyword identification model and collected sentiment lexicons related to specific sentiment domains for Vietnamese and other languages. The method used sentiment lexicon as training data for Naïve Bayes classification in social media content analysis. Based on the results, the approach used had the best accuracy, with a value of 98.2% for Vietnamese and 96.1% for English. [12].

Previous research has shown that the Naïve Bayes method is good at classifying Indonesian text. For example, Indrayuni (2019) used this approach to categorize cosmetic product reviews in Indonesian, with an accuracy of 90.50% [13]. While this study supports the algorithm's capabilities in a single-language setting, it does not investigate its performance when applied to languages with distinct structures and expression patterns, such as English and Indonesian. This study compares the efficiency of Naïve Bayes in sentiment classification across two languages, emphasizing how linguistic variations affect classification accuracy.

From the background explanation above, this research raises the title “Comparative Analysis of Naïve Bayes Algorithm Performance in English and Indonesian Text Sentiment Classification on Duolingo Application in Playstore”. The findings are intended to help to a better understanding of the algorithm's multilingual performance and give guidance to developers looking for effective sentiment classification algorithms for language-learning applications.

II. RESEARCH METHODOLOGY

Figure 1 shows the research process starting from data scraping, data pre-processing, labelling, tf-idf and evaluation of the naive bayes model carried out to analyze the comparison of two languages in the Duolingo application in Playstore.

The flow will be explained in more detail at the points below.

A. Data Collection

The initial step in this investigation is data collecting. On August 2, 2024, 5,000 customer reviews were taken from the Duolingo application page on the Play Store (2,500 in English and 2,500 in Indonesian). The data collecting procedure was carried out using Google Colab, which was chosen for its ability to handle large-scale data extraction rapidly while requiring little local processing resources.

To assure data quality, a screening method was used to exclude irrelevant or spam reviews. This was accomplished by removing duplicate entries, very brief evaluations (less than three lines), and non-textual information like emojis or special characters. Additionally, reviews that were identified as non-English or non-Indonesian were eliminated to ensure linguistic uniformity in the dataset.

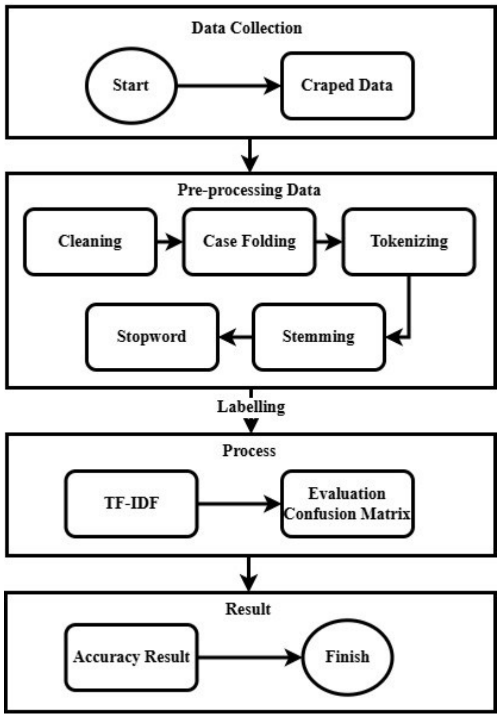

Figure 1. Research Stages

B. Pre-processing Data

The following process is data pre-processing. This stage is important in data analysis and machine learning, as it converts raw data into clean data for the model [14]. After the cleansing process, the dataset is developed by giving positive and negative labels to the data. The steps involved in data pre-processing are as follows:

1. Case Folding

   This stage changes all text to lowercase to promote consistency and avoid capitalization variances from influencing the classification process. Without case folding, words like "Good" and "good" would be considered separate tokens, potentially lowering model performance [15].
2. Tokenize

   Tokenization is the process of separating text data into token form [16].
3. Stopword

   This phase eliminates frequently used terms that have no major significance in sentiment classification, such as "the," "is," or "and" in English and "dan," "yang," or "adalah" in Indonesian. The NLTK stopword list was

DOI: 10.34148/teknika.v14i1.1207
ISSN 2549-8037, EISSN 2549-8045

utilized for English, while the Sastrawi stopword list was used for Indonesian [17].

4. Stemming
To preserve data integrity, stemming reduces words to their basic form. The Porter Stemmer method was employed for English text, whereas Sastrawi was used for Indonesian to handle affixes and word variants properly [18].

C. Labelling

After the pre-processing stage, user review data is labelled for sentiment analysis. The labels used in this research are 0 for negative sentiment and 1 for positive sentiment. Labelling is done automatically, using models and lexicons matching the specified language. VADER lexicon is used to label sentiments on English review data. Vader is a lexical technique that serves as a paradigm for mood analysis, and emotional depth can be used to evaluate various data points. The way Vader works is built on human knowledge and judgment [19].

The IndoBERT model is used to label sentiment on Indonesian-language review data. IndoBERT is successful in Indonesian NLP tasks. The algorithm was trained on an Indonesian language dataset containing about 4 billion words and 250 million phrases. Although it has the same architecture as BERT, the difference is in the dataset used for unsupervised training [20].

The selection of VADER and IndoBERT is based on previous research confirming their use in automated sentiment labeling, thus reducing the need for manual annotation. These methods provide a consistent and scalable strategy for labeling large datasets while achieving accurate classification results. This research excludes neutral ratings by removing such sentiments to distinguish between positive and negative views aiming to improve the reliability and interpretability of the model through binary classification (positive vs. negative). The labeled data is then used to train and test the Naïve Bayes algorithm in sentiment classification.

D. TF-IDF Feature Extraction

The following phase is feature extraction with the TF-IDF (Term Frequency-Inverse Document Frequency) approach. This approach aids in converting the dimensions of raw data into numerical characteristics for machine learning models. It computes Term Frequency (TF), which is how frequently a term appears in a document, and Inverse Document Frequency (IDF), which represents the word's importance across all documents. The TF-IDF score is the product of these two values, and it identifies key terms for text classification. This feature extraction produces a numerical representation of the text that can be used as input to the Naive Bayes method [21].

For example, a word like "exciting" that appears frequently in one document but infrequently in others will have a high TF-IDF score, indicating its relevance in that particular text.

E. Naive Bayes Classification

At this stage, the testing stage will be carried out using the Python programming language to find the accuracy and confusion matrix results by dividing the data with three comparison ratios, namely 90:10, 80:20 and 70:30. After obtaining accuracy results from testing Indonesian and English data, a comparison of Naïve Bayes performance will be carried out on the two results.

F. Model Evaluation

Evaluation is carried out in this study to measure the performance of the Naive Bayes algorithm between English and Indonesian in performing sentiment analysis on user reviews by looking for Accuracy, Precision, Recall and F1-score values.

The calculation values from the confusion matrix data are shown in Table 1. This data is generated after testing the model with the dataset [22].

Table 1. Confusion Matrix

Predicted Class	True Class	
	Positive	**Negative**
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

The three classification ranks are determined using the four parameters listed in the table above, namely:

1. System accuracy refers to how well the system can categorize the data. The following Formula 1 can be used to determine accuracy:

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \times 100\% \quad (1)$$

2. Precision measures the proportion of all papers found relevant to all documents found. The following Formula 2 can be used to determine precision:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

3. The ratio of all papers found relevant to all documents that are actually relevant is called recall. The following Formula 3 can be used to determine recall

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

4. F1-score will combine the two metrics, namely precision and recall, by calculating their numbers into one (Formula 4).

$$F1 - score = \frac{(2 \times Recall \times Precision)}{Recall+Precision} \quad (4)$$

III. RESULTS AND DISCUSSION

A. Data Collection

This study collected 2,500 user reviews in each of the languages (English and Indonesian) collected, to verify that the reviews were from different people. The data was extracted using a technique in Google Colab that uses a random library to eliminate bias by randomly selecting reviews from the available dataset. This reduces the possibility of over-representation of favorable or negative evaluations. The scraping results obtained can be seen in Table 2 below.

Table 2. Data Scrapping

Content
Gak mencet apa apa tiba tiba salah
lumayan lah buat pengalaman bahasa Inggris
Bagus sihhhh, tapi ada unsur lgbt, so sad

Table 2 above shows some reviews obtained from scraping results with the attributes used in the classification of this research, namely user names and reviews.

B. Data Pre-processing

In the initial stage, the data will go through a pre-processing stage, which will be cleaned to serve as the measurement value of the data. The following is the process used in the data pre-processing stage:

1. Case Folding

This stage will change the review to lowercase, remove spaces at the beginning and end of the sentence, and remove symbols such as exclamation points, commas, and others, which can be seen in Table 3.

Table 3. Case Folding Result

Input	Output
Sangat cocok untuk anda yang ingin belajar bahasa Inggris ataupun yang lainnya.	*sangat cocok untuk anda yang ingin belajar bahasa inggris ataupun yang lainnya*
ini bener² bisa buat belajar bahasa saya suka belajar di duolingo	*ini bener bisa buat belajar bahasa saya suka belajar di duolingo*
Very nice app, I learn two languages here.	very nice app i learn two languages here

Table 3 above shows examples of data set results before and after going through the Case Folding process.

2. Stopword

At the stopword stage, we will remove unimportant words such as personal pronouns, conjunctions, prepositions, auxiliary verbs, adverbs and others, which can be seen in Table 4.

Table 4. Stopword Result

Input	Output
sangat cocok untuk anda yang ingin belajar bahasa inggris ataupun yang lainnya	*cocok belajar bahasa inggris*
ini bener bisa buat belajar bahasa saya suka belajar di duolingo	*bener belajar bahasa suka belajar duolingo*
very nice app i learn two languages here	nice app learn two languages

Table 4 above shows examples of data set results before and after going through the Tokenizing process.

3. Tokenizing

The next stage is tokenization, which is the stage of separating each word contained in the review in the form of an array, as can be seen in Table 5.

Table 5. Tokenizing result

Input	Output
cocok belajar bahasa inggris	[*cocok*, *belajar*, *bahasa*, *inggris*]
bener belajar bahasa suka belajar duolingo	[*bener*, *belajar*, *bahasa*, *suka*, *belajar*, duolingo]
nice app learn two languages	[nice, app, learn, two, languages]

Table 5 above shows examples of data set results before and after going through the Stopword process.

4. Stemming

The last stage of pre-processing is the stemming process; at this stage, the removal of affix words contained in the front or back of the word, such as 'mem', 'nya', 'ber', and others can be seen in Table 6.

Table 6**.** Stemming Result

Input	Output
cocok belajar bahasa inggris	*Cocok ajar bahasa inggris*
Bener belajar bahasa suka belajar duolingo	*bener ajar bahasa suka ajar* duolingo
Nice app learn two language	nice app learn two language

Table 6 above shows examples of data set results before and after going through the Stemming process.

C. Labelling

After going through the data pre-processing stage, the next step is to give positive and negative labels to the cleaned data. The amount of English data is 2,500, with positive label results of 1990 data and negative label results of 510 data the amount of Indonesian data is also 2,500 data with positive label results of 1,626 and negative label results of 874. Figure 2 below displays the percentage of data that has been labelled in the form of a bar chart.

DOI: 10.34148/teknika.v14i1.1207

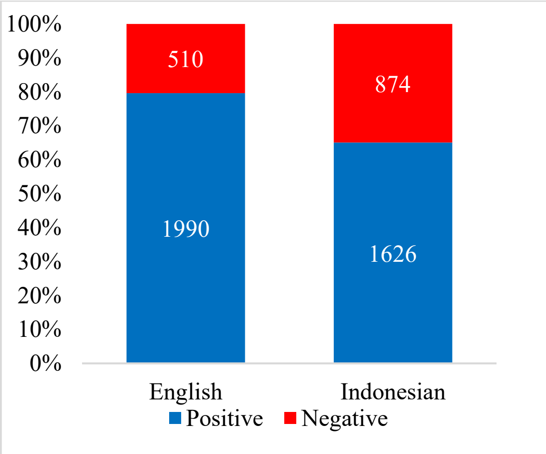

Figure 2. Bar Chart Percentage

With the bar chart displayed above, it can be seen how many positive labels and negative labels are in each data set.

D. TF-IDF Feature Extraction

In this study, text feature extraction is performed using the Term Frequency - Inverse Document Frequency (TF-IDF) approach, which measures the relevance of each word in the document based on its frequency of occurrence and how seldom the term appears in the whole corpus. The results of the TF-IDF calculation are shown as numerical vectors representing the weight of each word in the document.

Based on the calculation results obtained, the TF-IDF results are shown in Figure 3:

```
[7.21500802 7.72583364 7.43815157 ... 8.13129875 8.13129875 8.13129875]
  (0, 196)      0.18183118108837298
  (0, 245)      0.15730160737007295
  (0, 302)      0.17460696100513726
  (0, 435)      0.044736819557172895
  (0, 547)      0.2514493040050572
  (0, 587)      0.2016796355489454
  (0, 652)      0.11316998395063718
  (0, 831)      0.13855768581211184
  (0, 911)      0.16293965240674899
  (0, 1139)     0.10767954603762404
  (0, 1184)     0.23001467449792506
  (0, 1518)     0.10812664250564114
  (0, 1597)     0.15881037653468114
  (0, 1881)     0.23001467449792506
```

Figure 3. TF-IDF Result

The first number (0) in the matrix above represents the dataset's document index, while the second represents the word index in the TF-IDF feature vector. The last result in each row is the TF-IDF weight, representing the word's relevance in the document.

E. Split Data

After feature extraction, the data is split into two sets: training and testing data. The training data is used to train the Naive Bayes model, while the test data is used to assess the performance of the model. Data division in this study was carried out with three ratios, namely 90:10, 80:20 and 70:30. The reason for using these ratios is that some previous studies have stated that data with a ratio of 90:10 and 80:20 provides high accuracy results. However, overfitting often occurs because it has a relatively small amount of test data [23]. While the 70:30 ratio often gives consistent accuracy results, using more test data allows the model to capture many patterns in the data [24]. Table 6 shows the data sets with the ratio of 90:10, 80:20, and 70:30.

Table 6. Data Ratio Split

Split	Ratio 90:10	Ratio 80:20	Ratio 70:30
Training	2,250	2,000	1,750
Testing	250	500	750

Table 6 shows the amount of training data and testing data in each comparison ratio.

F. Model Evaluasi

This level is evaluated using a confusion matrix. Table 7 compares the results of all the confusion matrix evaluations for each situation.

Table 7. Accuracy Comparison

Ratio	90:10	80:20	70:30
English	84%	82%	83%
Indonesian	67%	67%	65%

Table 7 above shows the accuracy results based on various ratios. For English, the highest accuracy was achieved at a 90:10 ratio of 84%, followed by 70:30 at 83% and 80:20 at 82%. Meanwhile, the accuracy in Bahasa Indonesia remained consistent at 65% to 67% across all ratios. Table 7 shows that applying the Naïve Bayes algorithm in the 90:10 situation achieved an accuracy rate of 84% for English and 67% for Indonesian. Figures 4 and 5 show instances of the confusion matrix from modelling each text.

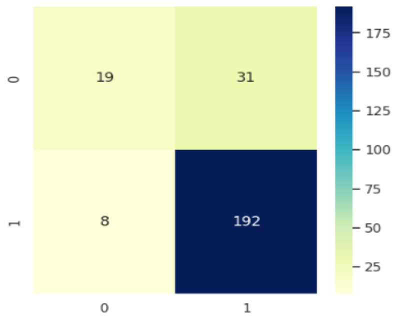

Figure 4. English Confusion Matrix

From Figure 4 above, the following information can be explained:

a. A total of 19 negatively labelled data are correctly predicted as True Negative (TN)
b. A total of 192 positively labelled data are correctly predicted as True Positive (TP)
c. A total of 8 positively labelled data are predicted incorrectly as False Negative (FN)
d. A total of 31 negatively labelled data are wrongly predicted as False Positive (FP)

The information obtained from the confusion matrix values above will be used to calculate the performance of the Naïve Bayes model in the form of accuracy, precision, recall and f1-score. Below is the calculation process to assess the performance of the Naïve Bayes model.

$$Accuracy = \frac{19+192}{19+192+8+31} \times 100\% = 84\%$$

$$Precision = \frac{177}{177+32} \times 100\% = 86\%$$

$$Recall = \frac{177}{177+22} \times 100\% = 96\%$$

$$F1-score = \frac{(2 \times 0.86 \times 0.98)}{0.86+0.98} = 90\%$$

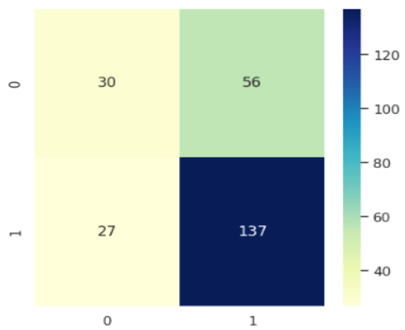

Figure 5. Indonesian Confusion Matrix

From Figure 5, the following information can be explained:

a. A total of 30 negatively labelled data are correctly predicted as True Negative (TN)
b. A total of 137 positively labelled data are correctly predicted as True Positive (TP)
c. A total of 27 positively labelled data are predicted incorrectly as False Negative (FN)
d. A total of 56 negatively labelled data are wrongly predicted as False Positive (FP)

The information obtained from the confusion matrix values above will be used to calculate the performance of the Naïve Bayes model in the form of accuracy, precision, recall and f1-score. Below is the calculation process to assess the performance of the Naïve Bayes model.

$$Accuracy = \frac{30+137}{30+137+27+56} \times 100\% = 67\%$$

$$Precision = \frac{137}{137+56} \times 100\% = 71\%$$

$$Recall = \frac{137}{137+27} \times 100 = 84\%$$

$$F1-score = \frac{(2 \times 0.71 \times 0.83)}{0.71+0.83} = 77\%$$

From the results of the calculations that have been carried out, Naive Bayes has a different performance when classifying user evaluations of the Duolingo application in English and Indonesian. The results of Naive Bayes performance comparison between English and Indonesian can be seen in the table below:

Table 10. Result Comparison

Language	Accuracy	Precision	Recall	F1-Score
English	84%	86%	96%	90%
Indonesian	67%	71%	84%	77%

According to Table 10, the Naïve Bayes algorithm works better on English text than Indonesian. The accuracy of English was 84%, whereas Indonesian was 67%. Precision was 86% for English and 71% for Indonesian, with recall of 96% and 84%, respectively. In terms of F1-score, English scored 90%, which was greater than Indonesian's 77%.

Although Naïve Bayes works well on English text, the vast difference in results with Indonesian is influenced by sarcasm, slang and mixed feelings, such as the example review "The app is good, but it crashes often and frustrates me," which can be categorized as positive simply because of the word "good," even though the context is negative. In addition, the model often misclassifies Indonesian reviews due to the casual language, slang, and confusing phrase patterns used. Words like "*mantul*" (*mantap betul*) often have a positive connotation, but this can change depending on the context, making it difficult for the model to handle them correctly.

IV. CONCLUSIONS

Based on the discussion described in the previous chapter, the following conclusions can be drawn. The Naïve Bayes algorithm has better performance in The Naïve Bayes method performs better in sentiment classification for English text than Indonesian. This is seen from the better accuracy of English text (84%) compared to Indonesian (67%). This discrepancy is caused by several variables, including the usage of informal language, slang, and more complicated word variants in Indonesian, which make proper classification more difficult for the model to achieve.

To increase the accuracy of Indonesian sentiment classification, future research can include more complicated models such as BERT or LSTM and contextual feature extraction approaches such as Word2Vec or FastText. Lexicon enrichment and improved pre-processing procedures can also aid in model performance.

REFERENCES

[1] P. Lavanya and E. Sasikala, "Deep Learning Techniques on Text Classification Using Natural Language Processing (NLP) In Social Healthcare Network: A Comprehensive Survey," in *2021 3rd International Conference on Signal Processing and*

DOI: 10.34148/teknika.v14i1.1207

Communication (ICPSC), 2021, pp. 603–609. doi: 10.1109/ICSPC51351.2021.9451752.

[2] Fristia Nopano, "10 Aplikasi Smartphone Terbaik untuk Belajar Bahasa Inggris," Telkom University Language Center.

[3] D. Winoto, V. D. Aditia, C. Sorisa, R. Priskila, and V. H. Pranatawijaya, "Analisis sentimen pada ulasan pengguna terhadap aplikasi pembelajaran bahasa Duolingo: Menggunakan algoritma Naïve Bayes dan K-Nearest Neighbor," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, pp. 3230–3236, 2024.

[4] A. Wibisono, "Filtering Spam Email Menggunakan Metode Naive Bayes," *Jurnal Teknologi Pintar*, vol. 3, no. 4, 2023.

[5] R. Hidayat, R. N. Rahman, M. R. Perdana, and A. Arbansyah, "Analisis Sentimen Aplikasi Identitas Kependudukan Digital (IKD) Menggunakan Metode Naïve Bayes," *Jurnal Sistem Informasi dan Ilmu Komputer*, vol. 2, no. 1, pp. 129–140, 2024.

[6] F. Alifiana, M. F. Asnawi, I. A. Ihsannudin, M. A. M. Baihaqy, and D. Asmarajati, "Analisis Sentimen Aplikasi Duolingo Menggunakan Algoritma Naïve Bayes dan Support Machine Learning," *Device*, vol. 13, no. 2, pp. 223–230, 2023.

[7] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," *J. Teknol. Inf. dan Ilmu Komput*, vol. 8, no. 1, p. 147, 2021.

[8] M. Cognetta, S. Moon, L. Wolf-sonkin, and N. Okazaki, *Parameter-Efficient Korean Character-Level Language Modeling*. 2023. doi: 10.18653/v1/2023.eacl-main.172.

[9] Ahmad Hussein Ababneh, "Investigating the relevance of Arabic text classification datasets based on supervised learning," *Journal of Electronic Science and Technology*, vol. 20, no. 2, Jun. 2022.

[10] T. A. Azzahra *et al.*, "Perbandingan Efektivitas Naïve Bayes dan SVM dalam Menganalisis Sentimen Kebencanaan di Youtube," *Jurnal Media Informatika Budidarma*, vol. 8, no. 1, pp. 312–322, 2024.

[11] K. Fithriasari, R. W. Mayasari, N. Iriawan, and W. S. Winahju, "Surabaya Government Performance Evaluation Using Tweet Analysis," *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics*, pp. 31–42, 2020.

[12] C. C. Le, P. W. C. Prasad, A. Alsadoon, L. Pham, and A. Elchouemi, "Text classification: Naïve bayes classifier with sentiment Lexicon," *IAENG Int J Comput Sci*, vol. 46, no. 2, pp. 141–148, 2019.

[13] E. Indrayuni, "Klasifikasi Text Mining Review Produk Kosmetik Untuk Teks Bahasa Indonesia Menggunakan Algoritma Naive Bayes," *Jurnal Khatulistiwa Informatika*, vol. 7, no. 1, 2019.

[14] R. Rahmadani, A. Rahim, and R. Rudiman, "Analisis Sentimen Ulasan 'Ojol The Game' di Google Play Store Menggunakan Algoritma Naive Bayes Dan Model Ekstraksi Fitur TF-IDF Untuk Meningkatkan Kualitas Game," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, 2024.

[15] S. D. Prasetyo, S. S. Hilabi, and F. Nurapriani, "Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN," *Jurnal KomtekInfo*, pp. 1–7, 2023.

[16] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast wordpiece tokenization," *arXiv preprint arXiv:2012.15524*, 2020.

[17] A. P. Wibawa, H. K. Fithri, I. A. E. Zaeni, and A. Nafalski, "Generating Javanese Stopwords List using K-means Clustering Algorithm.," *Knowl. Eng. Data Sci.*, vol. 3, no. 2, pp. 106–111, 2020.

[18] W. B. Demilie, "Implemented Stemming Algorithms for Information Retrieval Applications," 2020.

[19] Y. Asri, W. N. Suliyanti, D. Kuswardani, and M. Fajri, "Pelabelan Otomatis Lexicon Vader dan Klasifikasi Naive Bayes dalam menganalisis sentimen data ulasan PLN Mobile," 2022, *Petir*.

[20] R. Merdiansah, S. Siska, and A. A. Ridha, "Analisis sentimen pengguna X Indonesia terkait kendaraan listrik menggunakan IndoBERT," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 221–228, 2024.

[21] E. Hasibuan and E. A. Heriyanto, "Analisis Sentimen Pada Ulasan Aplikasi Amazon Shopping Di Google Play Store Menggunakan Naive Bayes Classifier," *Jurnal Teknik dan Science*, vol. 1, no. 3, pp. 13–24, 2022.

[22] R. D. Yahya, S. A. Wibowo, and N. Vendyansyah, "Analisis Sentimen Untuk Deteksi Ujaran Kebencian Pada Media Sosial Terkait Pemilu 2024 Menggunakan Metode Support Vector Machine," 2024.

[23] Muhammad Ardiansyah Sembiring, Mustika Fitri Larasati Sibuea, and Ika Rahmanda Sitorus, "Penerapan Naive Bayes Untuk Mengetahui Status Gizi Balita," *Journal Of Science And Social Research*, vol. 6(2), Jun. 2023.

[24] M. R. E. Rayhan, R. Rudiman, and F. Y. Fendy, "Perbandingan Metode K–Nearest Neighbor (KNN) dan Naive Bayes Terhadap Analisis Sentimen Pada Pengguna E-Wallet Aplikasi Dana Menggunakan Fitur Ekstraksi TF-IDF," *Jurnal Teknologi Informasi: Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika*, vol. 18, no. 2, pp. 139–159, 2024.