

Naive Bayes Classifier with SMOTE for Sentiment Analysis of Blibli App Reviews on The Google Play Store

Yogiek Indra Kurniawan^{*1}, Rafli Hudanul Sidiq², Azzam Dicky Umar Widadi³, Afiftha Ravi Aufa Yubiharto⁴, Akhmad Khahlil Gibran^{5,6}, Maulana Rizki Aditama^{7,8}, Fx Anjar Tri Laksono^{9,10}

^{1,2,3,4}Informatics, Engineering Faculty, Universitas Jenderal Soedirman, Indonesia

^{5,7,9}Geology Engineering, Engineering Faculty, Universitas Jenderal Soedirman, Indonesia

⁶Department of Petroleum Geology and Sediments, Faculty of Earth Sciences, King Abdulaziz University, Saudi Arabia

⁸Department of Earth and Environment, University of Manchester, United Kingdom

¹⁰Department of Geology and Meteorology, Institute of Geography and Earth Sciences, Faculty of Sciences, University of Pécs, Hungary

Email: ¹yogiek@unsoed.ac.id

Abstrak

In the digital age, online shopping has become prevalent, with platforms like the Google Play Store enabling users to download and review mobile applications. This study aims to analyze the sentiment of user reviews for the Blibli application on the Google Play Store using the Naive Bayes Classifier, a simple yet effective algorithm for text classification tasks. A total of 2500 recent reviews were scraped using Google Colaboratory and the Python programming language. Data preprocessing steps included cleaning, stopword removal, tokenization, and stemming, followed by addressing class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE). The dataset was divided into training and testing sets in an 80:20 ratio. The Naive Bayes algorithm with SMOTE was employed for sentiment classification, yielding an accuracy of 90%, precision of 90%, recall of 92%, and an F1-score of 91%. These results demonstrate the model's reliability in distinguishing between positive and negative sentiments, with a slight bias towards positive sentiments. Additionally, word cloud visualizations were generated to highlight frequently occurring words in both positive and negative reviews. The findings provide valuable insights for Blibli application developers and stakeholders, aiding in the assessment of user satisfaction and identification of areas for improvement. This research underscores the efficacy of the Naive Bayes Classifier in sentiment analysis and the utility of Google Colaboratory for data processing tasks.

Kata Kunci: *Analysis Sentiment, Blibli, Google Play Store, Naive Bayes, Python, SMOTE*

1. INTRODUCTION

Shopping is one of the things that people do quite often. Goods purchased start from basic needs to goods for self-satisfaction. Currently, shopping is easy to do, namely by shopping online. This online shopping is very helpful for people in getting goods that are far from the location where they live. In online shopping there are also various attractive promotions so that people prefer to shop online compared to coming to the store directly. Now online shopping platforms are diverse. These platforms can be easily downloaded such as the Google Play Store.

In the ever-evolving digital age, mobile-based applications have become an integral part of everyday life. One of the largest platforms for mobile app distribution is the Google Play Store, which allows users to not only download apps, but also leave reviews and ratings. One of the features found in the Play Store is the rating and review feature where users of products from the Play Store can give their opinions on the products they have used (Rizkya et al., 2023). These user reviews are an invaluable source of information for app developers as they provide insights into user satisfaction, app performance, and suggestions for improvement. Business companies generally use it to detect sentiment in social data, gauge brand reputation, and understand the customer and their needs (Demircan et al., 2021). Lots of users use various online resources to express their views and opinions (Wankhade et al., 2022). However, with such many reviews, manual analysis becomes an almost impossible task. Hence, an

automated approach is required to analyze the sentiment of these reviews. Analyzing the sentiment tendency of consumer evaluation can not only provide a reference for other consumers but also help businesses on e-commerce platforms to improve service quality and consumer satisfaction (Yang et al., 2020).

Sentiment analysis focuses on identifying and extracting opinions or sentiments from text. This technique is used to determine whether text is positive and negative. One of the popular algorithms used for sentiment analysis is the Naive Bayes Classifier. This algorithm is known for its simplicity and good performance in various text classification tasks.

The Naïve Bayes method is a classification method in machine learning that excels in using training data samples to estimate parameters involved in the classification process rapidly, resulting in high accuracy (Pasaribu & Sriani, 2023). based on previous research written by (Pasaribu & Sriani, 2023) entitled "The Shopee Application User Reviews Sentiment Analysis Employing Naïve Bayes Algorithm" concluded that the sentiment analysis of user reviews using naïve bayes with training data and 8:2 test data gets an accuracy of 86.00% which shows superior positive sentiment analysis results. In the previous analysis conducted by (Kosasih & Alberto, 2021) with the title "Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier" get the results of sentiment analysis which can be concluded that Sentiment analysis using TF-IDF method and Naive Bayes Classifier based on reviews from buyers (regardless of rating feature). The data collected were 1000 game product reviews on the online shopping site Shopee, divided into 700 training data and 300 test data. Based on the research results, the accuracy rate is 80.2223% and the f1 value is 0.691372. Then from another journal analysis with the title "App Review Sentiment Analysis Shopee Application In Google Play Store Using Naive Bayes Algorithm" written by (Pratmanto et al., 2020) with 200 review data taken consisting of 100 positive reviews and 100 negative reviews applying data mining to its analysis with naive bayes with partitioning techniques resulting in 96.667% accuracy, 100% precision, 93.33% recall, and AUC 1.00 which can be said to be included in a very good classification. from other research in other journals with the title "Sentiment Analysis of Online Transportation Service using the Naïve Bayes Methods" written by (Tika Adilah et al., 2020) obtained the final result using the naive Bayes method for data mining or text mining classification, namely 81.00% accuracy.

In this study, a sentiment analysis of user reviews on the Bilibli application on the google play store using naive bayes will be carried out by taking user review data using google colab. Collaboratory, or Colab for short, is a Google Research tool that allows developers to write and run Python code in their browser. Google Colab is an amazing tool for hands-on learning activity. It is a little Jupyter notebook that requires no installation and includes a fantastic free edition that gives you free access to Google computer resources like GPUs and TPUs (Srivastava et al., 2020). around 2500 data which will be divided for testing data and training data. The benefits of this research are expected to be able to find out the response of application users conveyed through opinions that are both positive and negative.

In line with these challenges, this research aims to conduct a comprehensive sentiment analysis of user reviews for the Bilibli application on the Google Play Store. The primary objective is to develop and evaluate a text classification model capable of accurately distinguishing between positive and negative sentiments. The Naive Bayes Classifier is selected as the core algorithm due to its proven effectiveness and computational efficiency in handling large-scale textual data.

To enhance classification performance, particularly in handling class imbalance often present in real-world review datasets, the Synthetic Minority Over-sampling Technique (SMOTE) is incorporated into the modeling process. This approach ensures that the minority class is sufficiently represented, leading to more balanced model training and improved generalization capabilities.

Specifically, the study seeks to achieve three key goals:

- a. Data Acquisition and Preprocessing – to collect a substantial dataset of Bilibli user reviews and transform raw text into structured, analyzable form through cleaning, tokenization, stemming, and stopword removal.
- b. Model Development and Evaluation – to implement Naive Bayes with and without SMOTE, and compare performance using accuracy, precision, recall, and F1-score metrics.

- c. Insight Generation – to visualize frequent terms in positive and negative reviews, enabling developers and stakeholders to identify strengths, weaknesses, and areas for improvement.

By fulfilling these objectives, this research not only provides a practical sentiment analysis framework for e-commerce application reviews but also contributes to the broader field of natural language processing by demonstrating the synergy between classic machine learning algorithms and data balancing techniques.

2. METHOD

2.1. Problem Formulation

This study aims to analyze the sentiment of user reviews for the Blibli application on the Google Play Store. The primary objective is to classify reviews as positive or negative using the Naive Bayes Classifier. This method leverages the simplicity and efficiency of the Naive Bayes algorithm in handling text classification tasks. The research stages are outlined to ensure a systematic approach, including data collection, preprocessing, model training, and evaluation.

Sentiment analysis, a crucial aspect of natural language processing (NLP), involves determining the emotional tone behind a series of words. In this case, it helps in understanding customer feedback by categorizing reviews into positive or negative sentiments. This classification aids the Blibli application developers and stakeholders in assessing user satisfaction and identifying areas for improvement. Sentiment analysis is one of the key analyses that is currently used with the aim of classifying sentiments and opinions generated by human beings and in text (A., 2021).

2.2. Naïve Bayes Classifier

In the research of text mining, document classification is a growing field. Even though we have many existing classifying approaches, Naïve Bayes Classifier is simple and effective at classification. At this stage the data are analyzed, then models are applied according to the type of data. The model proposed in this study is Naive Bayes.

The Naive Bayes Classifier is based on Bayes' theorem, which provides a way of calculating the posterior probability, from the prior probability, and the likelihood. The theorem is formulated as follows:

$$P(C_k | X) = \frac{P(C_k) \cdot P(X | C_k)}{P(X)} \quad (1)$$

$P(C_k | X)$ = This is the probability of class C_k given the data X .

$P(C_k)$ = This is the prior probability of class C_k .

$P(X | C_k)$ = This is the likelihood of the data X given that it is from class C_k .

$P(X)$ = This is the prior probability of the data X .

The "naive" aspect of the classifier comes from the assumption that the features (in this case, words in a review) are conditionally independent given the class. This assumption simplifies the computation significantly, making the algorithm very efficient, especially for large datasets.

2.3. Research Stages

The research involves several stages, each critical to the overall success of the sentiment analysis. The following are the research steps that have been carried out which can be seen in Figure 1.

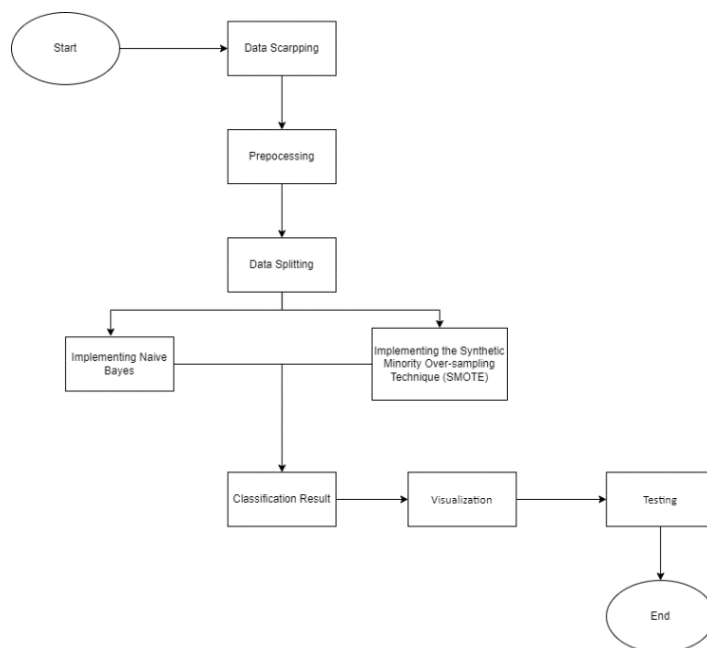


Fig 1. Flowchart

Figure 1 illustrates the overall research workflow applied in this study, starting from data acquisition to sentiment classification and result visualization. The process begins with data collection through web scraping of 2,500 recent Blibli application reviews from the Google Play Store using Google Colaboratory and the Python programming language. The collected raw data then undergoes a comprehensive preprocessing stage, including cleaning to remove irrelevant or neutral content, stopword removal, tokenization, and stemming to obtain the root form of words. To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied before splitting the dataset into training and testing subsets using the hold-out method with an 80:20 ratio. The preprocessed data is subsequently used to train and evaluate the Naive Bayes Classifier, both with and without SMOTE, by measuring accuracy, precision, recall, and F1-score. Finally, the workflow concludes with sentiment visualization using word clouds to highlight the most frequent terms in positive and negative reviews, providing both quantitative and qualitative insights into user feedback.

2.3.1. Data Collection

Data collection in this study was conducted using an automated web scraping process to ensure efficiency, accuracy, and reproducibility. The Google-play-scraper library, implemented in the Python programming language, was utilized within the Google Colaboratory environment to extract review data directly from the Blibli application page on the Google Play Store. This library was selected due to its ability to handle structured retrieval of app metadata and user reviews without the need for manual interaction. The scraping process targeted the most recent 2,500 reviews to ensure the dataset reflected current user experiences and sentiments.

Each review record consisted of multiple attributes, including review text, review date, user rating, and additional metadata. However, only the textual content of the review and the sentiment implied by the rating were considered relevant for this analysis. To maintain data quality, duplicate entries and incomplete reviews were automatically removed during the scraping process. Furthermore, to preserve compliance with platform policies and ethical research standards, the scraping was performed without violating user privacy, as no personal identifiable information (PII) was collected.

This automated collection method offers several advantages: it reduces human error, ensures scalability for larger datasets, and allows for periodic replication to capture longitudinal sentiment trends. By relying on a programmatic approach, the dataset can be updated in the future to evaluate changes in user sentiment as the Blibli application undergoes feature updates or service improvements.

2.3.2. Data Preprocessing

The following stage is preprocessing. Preprocessing is stage in which we clean the data before extracting its features. Preprocessing can help to avoid data interference, imperfect data, and inconsistent data (Rozaq et al., 2022). Pre-processing is a process of changing the form of data that has not been structured into structured data as needed, for further mining processes (Nurcahyono et al., 2020).

- a. Cleaning: Removal of noise and irrelevant data, particularly neutral value data.
- b. Stopword Removal: Elimination of common words that do not contribute to the sentiment, based on a predefined stopword list.
- c. Tokenizing: Breaking down text into individual tokens for easier processing.
- d. Stemming: Reducing words to their root form using the “Sastrawi” library.
- e. SMOTE (Synthetic Minority Over-sampling Technique): Used to address class imbalance by increasing the number of samples in the minority class.

2.3.3. Data Splitting

Data splitting is a crucial step in the experimental design to ensure that the performance evaluation of the classification model is unbiased and representative of real-world scenarios. In this study, the hold-out method was employed, where the dataset was randomly divided into two subsets: 80% for training and 20% for testing. The training set is used to build and optimize the Naive Bayes model, while the testing set is reserved exclusively for evaluating the model’s generalization capability on unseen data.

To maintain reproducibility and minimize the influence of random sampling bias, a fixed random seed was applied during the splitting process. This ensures that the same data partitioning can be replicated in subsequent experiments or comparative studies. Furthermore, the split was stratified according to class labels to preserve the original sentiment distribution in both the training and testing sets, especially after applying the Synthetic Minority Over-sampling Technique (SMOTE) for class balancing.

The rationale for selecting an 80:20 ratio lies in balancing the need for sufficient training data to optimize model learning while retaining enough testing data to produce statistically meaningful performance metrics. By implementing this structured splitting process, the study ensures that evaluation metrics, such as accuracy, precision, recall, and F1-score, accurately reflect the classifier’s ability to handle real-world sentiment analysis tasks without overfitting to the training data.

2.4. Classification

The classification stage in this study focuses on applying the Naive Bayes Classifier to categorize user reviews into positive or negative sentiment classes. Naive Bayes was selected due to its simplicity, low computational cost, and proven effectiveness in text classification tasks, particularly when dealing with large and sparse datasets such as user-generated reviews. The classifier operates based on Bayes’ theorem, assuming conditional independence between features, which in this context are the tokenized words from the preprocessed text.

Prior to classification, the textual data was transformed into a numerical representation suitable for the model, typically through term frequency-based vectorization. Each review was represented as a feature vector, where the presence and frequency of specific terms contribute to the probability estimation of each sentiment class. The Naive Bayes algorithm then calculates the posterior probability for each class given the observed features and assigns the review to the class with the highest probability.

To address the inherent imbalance between positive and negative reviews, two model configurations were evaluated: one without class balancing and another with the Synthetic Minority Over-sampling Technique (SMOTE) applied prior to training. This allowed for an assessment of the impact of data balancing on classification performance. The models were evaluated using a confusion matrix to determine true positives, true negatives, false positives, and false negatives, and performance was quantified using accuracy, precision, recall, and F1-score metrics.

By combining the Naive Bayes algorithm with SMOTE, this classification framework aims to achieve a robust and balanced predictive capability, ensuring that minority sentiment classes are

adequately represented in the final model, which is essential for reliable sentiment analysis in real-world applications.

2.5. Visualization

The visualization stage in this study aims to provide an intuitive and interpretable representation of the sentiment analysis results, enabling stakeholders to gain rapid insights from the processed data. Word cloud generation was employed to illustrate the most frequently occurring terms within positive and negative review categories. In this visualization, the size of each word corresponds to its relative frequency in the dataset, allowing key themes, issues, and strengths highlighted by users to be identified at a glance (Erfina & Alamsyah, 2023).

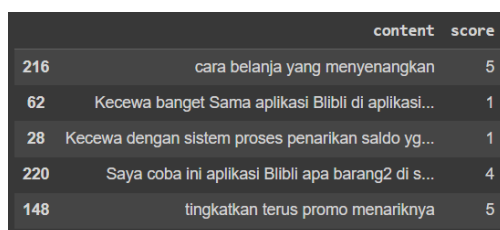
This method is particularly useful in exploratory data analysis, as it condenses large volumes of textual information into a compact and easily digestible form. For positive reviews, frequently occurring words often reflect satisfaction with product quality, usability, and promotional offers, while negative reviews highlight common pain points such as delivery issues, technical errors, or unsatisfactory customer service. By visually contrasting these two categories, developers and decision-makers can pinpoint areas for enhancement and prioritize improvements based on user sentiment trends.

The use of visualization not only complements the quantitative evaluation metrics—such as accuracy, precision, recall, and F1-score—but also bridges the gap between data science outputs and actionable business insights. Furthermore, this approach supports continuous monitoring of sentiment changes over time, as updated visualizations can be generated periodically to track the impact of feature updates, marketing campaigns, or service modifications. In the broader context of computer science, such visual analytics play a crucial role in enhancing the interpretability of natural language processing (NLP) models and facilitating evidence-based decision-making in software development and user experience optimization.

3. RESULTS

3.1. Web Scrapping

Web scrapping data is a process carried out to collect review data from the Google Play Store using the Python programming language on the Bilibli link on the Google Play Store obtained 2500 review data from the latest. This process is carried out by installing the Google-play-scraper library which is used to scrape review data on the Google Play Store by simply entering the ID of the application whose review you want to retrieve which is found in the application link (Agustina et al., 2022). The following is a dataset that has been taken and 5 examples of data are displayed which can be seen in figure 2.



	content	score
216	cara belanja yang menyenangkan	5
62	Kecewa banget Sama aplikasi Bilibli di aplikasi...	1
28	Kecewa dengan sistem proses penarikan saldo yg...	1
220	Saya coba ini aplikasi Bilibli apa barang2 di s...	4
148	tingkatkan terus promo menariknya	5

Fig 2. Scrapping Result

3.2. Preprocessing

Text preprocessing consists of things like punctuation and symbol removal, stop words and slang removal, and stemming.(Hariguna & Rachmawati, 2019). Data preprocessing, or data pre-processing, is a series of steps or stages carried out on raw data before the data is used for further analysis or model development. The main goal of data preprocessing is to improve data quality, ensure the accuracy of analysis results, and address problems or deficiencies that may arise in the raw data (Sudipa et al., 2024).

3.2.1. Cleaning

Data cleaning is a process carried out to remove noise from data that is inconsistent or could be called irrelevant. For cleaning is to remove data that has a neutral value. The following are the results of data cleaning which can be seen in figure 3.

Before			
	content	score	Label
0	mudah mencari barang mudah cara bayarnya...	5	Positif
1	pertama coba ,nanti rate lagi	5	Positif
2	harganya plus pengiriman masih mahal	3	NaN
After			
	content	score	Label
0	mudah mencari barang mudah cara bayarnya...	5	Positif
1	pertama coba ,nanti rate lagi	5	Positif
3	Aplikasi jelek, cancel pesanan aja gabisa. Gam...	1	Negatif

Fig 3. Data Cleaning

Figure 3 presents the outcome of the data cleaning process applied to the collected Blibli application reviews. In this stage, irrelevant information, noise, and neutral-valued reviews were removed to ensure that the dataset only contained text that contributes to sentiment classification. Noise in this context includes elements such as excessive punctuation, special symbols, URLs, and emoticons that do not provide meaningful sentiment cues for the analysis. Removing neutral reviews was essential to maintain a clear separation between positive and negative sentiment classes, thereby improving the classifier's learning process and predictive accuracy.

The cleaning process was implemented programmatically using Python-based text preprocessing functions, which ensured reproducibility and minimized human error. By eliminating inconsistent or incomplete entries, the dataset became more uniform and semantically relevant for further stages, such as tokenization and feature extraction. This refinement step not only enhanced the quality of the training and testing data but also reduced the risk of introducing bias or irrelevant patterns into the model.

Ultimately, the cleaned dataset forms a reliable foundation for subsequent sentiment analysis, as it ensures that the Naive Bayes Classifier is trained on high-quality, sentiment-rich content, thereby improving the robustness and validity of the classification results.

3.2.2. Stopword Removal

Stopword Removal is part of the text preprocessing stage which aims to remove irrelevant words in a sentence based on the stopwords list. Stopword to delete words that are not needed or unwanted in this project is to filter words using the stopwords library filter (Jaka Harjanta & Herlambang, 2020). Stop word removal can be thought of as an entity selection routine, in which entities that do not contribute to correct ranking decisions are considered spurious words and are removed from the entity space accordingly (Thwe et al., 2021). The following are the results of stopwords removal which can be seen in the figure 4.

text_clean	text_Stopword
mudah mencari barang mudah cara bayarnya	mudah mencari barang mudah bayarnya
pertama coba nanti rate lagi	coba rate
aplikasi jelek cancel pesanan aja gabisa gamau...	aplikasi jelek cancel pesanan aja gabisa gamau...
keren mudah digunakan banyak fitur dan promosi	keren mudah fitur promosi
aplikasi jujur dan cepet responsenya	aplikasi jujur cepet responsenya

Fig 4. Stopword Removal

3.2.3. Tokenizing

Tokenization is a procedure for recovering the elements of interest in a sequence of data. This term is commonly used to describe an initial step in the processing of programming languages, and also for

the preparation of input data in the case of artificial neural networks (Friedman, 2023). Sentences are categorized into words or tokens by ignoring whitespaces and other symbols (Vangara* et al., 2020). Tokenization refers to splitting the text into meaningful smaller units known as tokens (Uddin et al., 2022). The following are the results of tokenizing which can be seen in the figure 5.

text_stopword	text_tokens
mudah mencari barang mudah bayarnya	[mudah, mencari, barang, mudah, bayarnya]
coba rate	[coba, rate]
aplikasi jelek cancel pesanan aja gabisa gamau...	[aplikasi, jelek, cancel, pesanan, aja, gabisa...]
keren mudah fitur promosi	[keren, mudah, fitur, promosi]
aplikasi jujur cepet responsenya	[aplikasi, jujur, cepet, responsenya]

Fig 5. Tokenizing

3.2.4. Stemming

Stemming is the next process after filtering to get root words from each token. The stemming method used in developing corpus was “Sastrawi” (Mutiara et al., 2021). The following are the results of stemming which can be seen in the figure 6.

```
diaplikasinya : aplikasi
halo : halo
menurun : turun
listrik : listrik
jawabannya : jawab
opsi : opsi
produknya : produk
dikasih : kasih
```

Fig 6. Stemming

Figure 6 illustrates the results of the stemming process applied during the text preprocessing stage. Stemming is a linguistic normalization technique aimed at reducing words to their root or base form, thereby minimizing lexical variation and improving the consistency of the dataset. In this study, the Indonesian “Sastrawi” stemming library was utilized due to its effectiveness in handling the morphological structure of the Indonesian language, which is characterized by extensive use of affixes.

The process removes prefixes, suffixes, and infixes from each token, converting various morphological forms of a word into a single canonical representation. For example, the words “membeli” and “pembelian” are both reduced to the root word “beli”. This normalization is particularly important in sentiment analysis, as it allows semantically equivalent terms to be treated as identical features, thereby reducing feature sparsity and enhancing model performance.

By applying stemming, the dimensionality of the feature space is reduced, which not only optimizes computational efficiency but also improves the classifier’s ability to generalize patterns across similar words. This step ensures that the Naive Bayes model focuses on the semantic essence of the text rather than superficial morphological differences, ultimately contributing to higher accuracy and more reliable sentiment predictions.

3.3. SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) was applied in this study to address the class imbalance problem commonly found in real-world sentiment datasets, where positive reviews often significantly outnumber negative ones (Xu et al., 2020).. Class imbalance can lead to biased model training, where the classifier tends to favor the majority class, thus reducing its ability to correctly predict minority class instances. SMOTE mitigates this issue by generating synthetic samples for the minority class rather than simply duplicating existing ones, thereby enriching the dataset with more diverse and representative examples (Hermanto et al., 2020).

In the SMOTE process, new synthetic samples are created by selecting a minority class instance and generating additional instances along the line segments that connect it to its k nearest minority neighbors in the feature space. This approach ensures that the synthetic data points are plausible and lie within the same decision region as the original minority samples. The technique was applied after text preprocessing and feature extraction but before splitting the dataset into training and testing sets, ensuring that the training data was balanced while avoiding data leakage into the testing set.

Figures 7 and 8 visually demonstrate the distribution of sentiment classes before and after applying SMOTE. As shown, the dataset initially exhibited a disproportionate number of positive reviews, which was corrected post-SMOTE to achieve an approximately equal distribution between positive and negative sentiments. This balancing process is crucial for improving the classifier's recall for the minority class, leading to more reliable and fair performance metrics.

```
Label
Positif    1859
Negatif    562
Name: count, dtype: int64
```

Fig 7. Before Smote

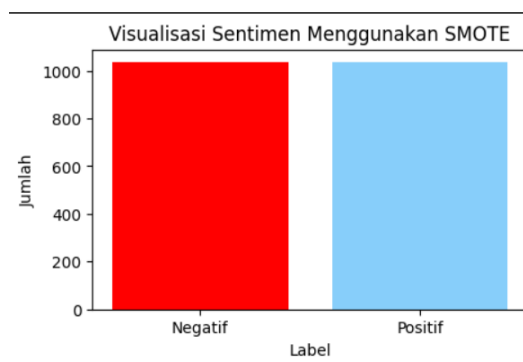


Fig 8. SMOTE

By integrating SMOTE with the Naive Bayes classification approach, this study ensures that the model's predictive capabilities are not compromised by skewed class distributions, thereby enhancing both robustness and generalization in sentiment analysis tasks.

3.4. Splitting Data

The process of dividing training data using the hold-out method divides training data by 80% and testing data by 20%. The data-sharing process uses the Python programming language which can be seen in the figure 9.

```
#membagi data menjadi data training dan testing dengan test_size = 0.20 dan random state nya 0
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(data_clean['content'], data_clean['Label'],
                                                  test_size = 0.20,
                                                  random_state = 0)
```

Fig 9. Splitting Data

Figure 9 depicts the process of dividing the preprocessed dataset into training and testing subsets using the hold-out validation method. In this study, 80% of the data was allocated for training the Naive Bayes model, while the remaining 20% was reserved for testing its generalization capability on unseen data. This division ensures that the evaluation metrics—such as accuracy, precision, recall, and F1-score—reflect the model's true predictive performance rather than its ability to simply memorize the training data.

3.5. Classification

Naïve Bayes technique is utilized for both categorization and training. Testing using the confusion matrix was carried out to test the model implemented on training data and testing data. Based on the test results using the confusion matrix, the results are shown in the table 1 and 2.

Table 1. Confusion Matrix Result

Prediction Data	Actual Data	
	Positive	Negative
Positive	96	21
Negative	23	349

Table 2. Confusion Matrix Result with SMOTE

Prediction Data	Actual Data	
	Positive	Negative
Positive	249	25
Negative	19	165

The results of the naive Bayes algorithm obtain accuracy, precision, recall and f1-score values which can be seen in the figure 10 and 11.

```
MultinomialNB Accuracy: 0.9092783505154639
MultinomialNB Precision: 0.8205128205128205
MultinomialNB Recall: 0.8067226890756303
MultinomialNB f1_score: 0.8135593220338982
```

Fig 10. Accuracy, Precision, Recall, and f1-score Results

```
MultinomialNB Accuracy: 0.9039301310043668
MultinomialNB Precision: 0.9087591240875912
MultinomialNB Recall: 0.9291044776119403
MultinomialNB f1_score: 0.9188191881918818
```

Fig 11. Accuracy, Precision, Recall, and f1-score Results with SMOTE

The following are the sentiment results from the Blibli application review on Google Playstore, the results of which can be seen in the figure 12 and 13.

	precision	recall	f1-score	support
Negatif	0.82	0.81	0.81	119
Positif	0.94	0.94	0.94	366
accuracy			0.91	485
macro avg	0.88	0.87	0.88	485
weighted avg	0.91	0.91	0.91	485

Fig 12. Sentiment Results

	precision	recall	f1-score	support
Negatif	0.91	0.93	0.92	268
Positif	0.90	0.87	0.88	190
accuracy			0.90	458
macro avg	0.90	0.90	0.90	458
weighted avg	0.90	0.90	0.90	458

Fig 13. Sentiment Results with SMOTE

4. DISCUSSIONS

The results of the study provide insights into the sentiment analysis of Blibli application reviews from the Google Play Store. Through a comprehensive web scraping and data preprocessing workflow, extract, clean, and analyze 2500 reviews to determine user sentiment.

Sentiment analysis using the Naive Bayes algorithm yielded promising results, with an accuracy rate indicating that the model correctly classified a significant portion of the reviews. Specifically, the confusion matrix showed 96 true positive predictions and 345 true negative predictions, alongside 23 false positives and 21 false negatives then with SMOTE showed 249 true positive predictions and 165 true negative predictions, alongside 19 false positives and 25 false negatives. These results demonstrate the model's capability to distinguish between positive and negative sentiments effectively.

The accuracy, precision, recall, and F1-score metrics further validate the model's performance. With an accuracy of 0.90, a precision of 0.82, a recall of 0.80, and an F1-score of 0.81 then use SMOTE with an accuracy of 0.90, a precision of 0.90, a recall of 0.92, and an F1-score of 0.91, the model demonstrates strong reliability in predicting both positive and negative sentiments. These metrics indicate that the model is particularly effective at identifying positive sentiments, as reflected in the high recall value. This slight bias towards positive sentiment predictions could be attributed to the inherent nature of app reviews, where satisfied users are more likely to leave reviews than dissatisfied users.

The results obtained from this study indicate that the Naive Bayes Classifier, when combined with the Synthetic Minority Over-sampling Technique (SMOTE), provides robust and balanced performance in classifying sentiments from e-commerce application reviews. The implementation of SMOTE successfully mitigated the class imbalance problem, as reflected in the improved precision (from 0.82 to 0.90) and recall (from 0.80 to 0.92). This improvement suggests that the model became more capable of identifying minority class instances, which is critical in sentiment analysis tasks where negative feedback, although less frequent, often contains valuable insights for service improvement.

The application of word cloud visualization further complements the quantitative findings by revealing recurring terms in both positive and negative sentiments. These visual insights enable developers and stakeholders to quickly identify recurring user concerns, such as technical issues or delivery problems, and strengths such as ease of use and promotional benefits.

From a broader perspective in computer science, this research underscores the importance of integrating data balancing techniques into traditional machine learning models for natural language processing tasks. The methodological framework presented here—encompassing automated data collection, comprehensive preprocessing, class balancing, and sentiment visualization—can be adapted to various domains, including social media analytics, customer feedback systems, and opinion mining for policy-making. The significance of this study lies in its contribution to improving model fairness and predictive reliability, which are crucial for building user-centric and data-driven decision-making tools in the e-commerce sector and beyond.

5. CONCLUSION

This study investigated the application of the Naive Bayes Classifier, enhanced with the Synthetic Minority Over-sampling Technique (SMOTE), for sentiment analysis of 2,500 user reviews of the Blibli application collected from the Google Play Store. A systematic workflow was employed, including automated web scraping, text preprocessing (cleaning, stopword removal, tokenization, and stemming), and class balancing prior to model training. The evaluation demonstrated that integrating SMOTE improved the model's ability to classify both majority and minority classes, achieving 90% accuracy, 90% precision, 92% recall, and 91% F1-score. These results confirm that the proposed approach effectively addresses class imbalance and enhances classification reliability.

The visual analysis using word clouds provided additional qualitative insights, highlighting recurring positive and negative themes that can inform targeted application improvements. The combination of quantitative and qualitative findings underscores the practical utility of this framework for e-commerce platforms seeking to understand and respond to user feedback more effectively.

From a computer science perspective, this research contributes to the advancement of text classification methodologies by demonstrating the synergistic benefits of combining classic machine

learning algorithms with data balancing techniques. The proposed framework is adaptable to other domains involving large-scale textual datasets, such as social media monitoring, public opinion mining, and customer service analytics. Future research may explore the integration of deep learning architectures, multi-class sentiment classification, or multilingual analysis to further enhance performance and broaden applicability.

DAFTAR PUSTAKA

- A., P. P. (2021). Performance Evaluation and Comparison using Deep Learning Techniques in Sentiment Analysis. *Journal of Soft Computing Paradigm*, 3(2), 123–134. <https://doi.org/10.36548/jscp.2021.2.006>
- Agustina, N., Citra, D. H., Purnama, W., Nisa, C., & Kurnia, A. R. (2022). Implementasi Algoritma Naive Bayes untuk Analisis Sentimen Ulasan Shopee pada Google Play Store. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 2(1), 47–54. <https://doi.org/10.57152/malcom.v2i1.195>
- Demircan, M., Seller, A., Abut, F., & Akay, M. F. (2021). Developing Turkish sentiment analysis models using machine learning and e-commerce data. *International Journal of Cognitive Computing in Engineering*, 2(October), 202–207. <https://doi.org/10.1016/j.ijcce.2021.11.003>
- Erfina, A., & Alamsyah, M. R. N. R. (2023). Implementation of Naive Bayes classification algorithm for Twitter user sentiment analysis on ChatGPT using Python programming language. *Data and Metadata*, 2, 2–11. <https://doi.org/10.56294/dm202345>
- Friedman, R. (2023). *Tokenization in the Theory of Knowledge*. 380–386.
- Hariguna, T., & Rachmawati, V. (2019). Community Opinion Sentiment Analysis on Social Media Using Naive Bayes Algorithm Methods. *IJIIS: International Journal of Informatics and Information Systems*, 2(1), 33–38. <https://doi.org/10.47738/ijiis.v2i1.11>
- Hermanto, Kuntoro, A. Y., Asra, T., Pratama, E. B., Effendi, L., & Ocanitra, R. (2020). Gojek and Grab User Sentiment Analysis on Google Play Using Naive Bayes Algorithm and Support Vector Machine Based Smote Technique. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012102>
- Jaka Harjanta, A. T., & Herlambang, B. A. (2020). Extraction Sentiment Analysis Using naive Bayes Algorithm and Reducing Noise Word applied in Indonesian Language. *IOP Conference Series: Materials Science and Engineering*, 835(1). <https://doi.org/10.1088/1757-899X/835/1/012051>
- Kalmukov, Y. (2021). Using Word Clouds for Fast Identification of Papers' Subject Domain and Reviewers' Competences 15. *Proceedings of University of Ruse*, 60, 114–119.
- Kosasih, R., & Alberto, A. (2021). Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier. *ILKOM Jurnal Ilmiah*, 13(2), 101–109. <https://doi.org/10.33096/ilkom.v13i2.721.101-109>
- Mutiara, A. B., Wibowo, E. P., & Santosa, P. I. (2021). Improving the accuracy of text classification using stemming method , a case of non - formal Indonesian conversation. *Journal of Big Data*, 1–16. <https://doi.org/10.1186/s40537-021-00413-1>
- Nurcahyono, D., Putra, W. P., Najib, A., & Tulili, T. R. (2020). Analysis sentiment in social media against election using the method naive Bayes. *Journal of Physics: Conference Series*, 1511(1). <https://doi.org/10.1088/1742-6596/1511/1/012003>
- Pasaribu, N. A., & Sriani. (2023). The Shopee Application User Reviews Sentiment Analysis Employing Naïve Bayes Algorithm. *International Journal Software Engineering and Computer Science (IJSECS)*, 3(3), 194–204. <https://doi.org/10.35870/ijsecs.v3i3.1699>
- Pratmanto, D., Rousyati, R., Wati, F. F., Widodo, A. E., Suleman, S., & Wijianto, R. (2020). App Review Sentiment Analysis Shopee Application in Google Play Store Using Naive Bayes Algorithm. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012043>

- Rizkya, A. T., Rianto, R., & Gufroni, A. I. (2023). Implementation of the Naive Bayes Classifier for Sentiment Analysis of Shopee E-Commerce Application Review Data on the Google Play Store. *International Journal of Applied Information Systems and Informatics (JAISI)*, 1(1), 31–37.
- Rozaq, A., Yunitasari, Y., Sussolaikah, K., & Sari, E. R. N. (2022). Sentiment Analysis of Kampus Mengajar 2 Toward the Implementation of Merdeka Belajar Kampus Merdeka Using Naïve Bayes and Euclidean Distance Methods. *International Journal of Advances in Data and Information Systems*, 3(1), 30–37. <https://doi.org/10.25008/ijadis.v3i1.1233>
- Srivastava, M., Yadav, V., & Singh, S. (2020). Implementation of Web Application for Disease Prediction Using AI. *BOHR International Journal of Data Mining and Big Data*, 1(1), 5–9. <https://doi.org/10.54646/bijdmdbd.002>
- Sudipa, I. G. I., Darmawiguna, I. G. M., Dendi, I. M., & Sanjaya, M. (2024). *Buku ajar data mining* (Issue January).
- Thwe, P., Aung, Y. Y., & Lwin, C. C. (2021). Naïve Bayes Classifier for Sentiment Analysis. *International Journal Of All Research Writings*, 3(7), 32–35.
- Tika Adilah, M., Supendar, H., Ningsih, R., Muryani, S., & Solecha, K. (2020). Sentiment Analysis of Online Transportation Service using the Naïve Bayes Methods. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012093>
- Uddin, M. N., Hafi, M. F. Bin, Hossain, S., & Islam, S. M. M. (2022). Drug Sentiment Analysis using Machine Learning Classifiers. *International Journal of Advanced Computer Science and Applications*, 13(1), 92–100. <https://doi.org/10.14569/IJACSA.2022.0130112>
- Vangara*, R. V. B., Thirupathur, K., & Vangara, S. P. (2020). Opinion Mining Classification using Naive Bayes Algorithm. *International Journal of Innovative Technology and Exploring Engineering*, 9(5), 495–498. <https://doi.org/10.35940/ijitee.e2402.039520>
- Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. In *Artificial Intelligence Review* (Vol. 55, Issue 7). Springer Netherlands. <https://doi.org/10.1007/s10462-022-10144-1>
- Xu, Z., Shen, D., Nie, T., & Kou, Y. (2020). A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. *Journal of Biomedical Informatics*, 107(June), 103465. <https://doi.org/10.1016/j.jbi.2020.103465>
- Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. *IEEE Access*, 8, 23522–23530. <https://doi.org/10.1109/ACCESS.2020.2969854>