Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Sentiment Analysis of Mobile Legends Game using Naïve Bayes, K-Nearest Neighbors and Support Vector Machine Algorithm

Samuel Surya Sanjaya[1*], April Kurniawan Jaya[2], Rikky Candra[3], Stefven Zang[4]

[1,2,3,4] Institut Bisnis dan Teknologi Pelita Indonesia ,Jl. Jend Ahmad Yani Sukajadi, Pekanbaru, Riau, telp. 0761 24418
samuel.ss@student.pelitaindonesia.ac.id [1], april.kj@student.pelitaindonesia.ac.id [2], rikky.candra@student.pelitaindonesia.ac.id [3], stefven.zang@student.pelitaindonesia.ac.id [4]*

Abstract

Sentiment analysis of Mobile Legends: Bang Bang (MLBB) user reviews is very important for understanding public satisfaction and perspectives. Therefore, this study aims to analyze and compare the performance of three Machine Learning algorithms: Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) in classifying user review sentiments. A supervised machine learning approach was applied using 6,000 reviews obtained from a secondary Kaggle dataset, involving Data Preprocessing and Feature Extraction (TF-IDF) stages, followed by an 80:20 Data Split for model training. The comparison of metric results shows that the Support Vector Machine (SVM) model provides the best overall performance, achieving 79.88% Accuracy and 78.06% F1-Score, although NB slightly outperforms in the Precision metric. In conclusion, SVM's performance proves this algorithm is superior in classifying Indonesian-language mobile game review sentiments, providing strategic insights for MLBB developers in making service improvement decisions.

***Keywords**: Analisis Sentimen; K-Nearest Neighbors; Klasifikasi Teks; Mobile Legends; Naïve Bayes; Support Vector Machine*

1. Introduction

Mobile Legends: Bang Bang (MLBB) is a highly popular Multiplayer Online Battle Arena (MOBA) game in Indonesia, attracting millions of active users. This popularity generates a large volume of reviews on the Google Play Store, which serves as a direct reflection of user experience and satisfaction[1] . Therefore, sentiment analysis of user reviews is very important for developers to understand public perspectives and as a basis for service improvement. Given the large amount of review data, an automated approach based on Machine Learning is essential for interpreting sentiments accurately and quickly[2]. This study uses and compares Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) algorithms to classify user sentiments. The results of this analysis are expected to provide strategic input for MLBB developers in improving application and service quality[3], [4].

The Research Problems in this study are as follows:
1. How are user review sentiments of the Mobile Legends game on Google Play Store classified into positive, negative, and neutral categories?
2. How do the performances of Naïve Bayes, K-Nearest Neighbors, and Support Vector Machine algorithms compare in classifying Mobile Legends user review sentiments?

The Research Objectives are:
1. To analyze and classify user sentiments toward the Mobile Legends: Bang Bang application based on reviews on the Google Play Store.
2. To compare the performance of Naïve Bayes, K-Nearest Neighbors, and Support Vector Machine algorithms in sentiment classification using model performance metrics.

This research has scientific significance in the form of contributions to knowledge development in sentiment analysis, particularly the application of comparing three Machine Learning models on Indonesian-language mobile game review data[5], [6]. Additionally, this research provides practical insights for game developers in utilizing sentiment analysis results to make appropriate improvement decisions.

2. Literature Review

2.1. Sentiment Analysis

Sentiment Analysis is a technique in the field of Natural Language Processing (NLP) whose function is to identify and categorize views or emotional expressions conveyed by users regarding an entity, such as a product, service, or game[7]. In this research, sentiment analysis aims to categorize Mobile Legends user reviews into positive, negative, or neutral sentiment categories. Previous research involving a comparison between Naïve Bayes, Support Vector Machine (SVM), and Random Forest algorithms on sentiment analysis of Ajaib application reviews showed that all three models have competitive classification capabilities. Performance evaluation of these models yielded accuracy in the range of 85% for Random Forest and Naïve Bayes, while the SVM model recorded 83% accuracy[8].

2.2. Text Classification

Text classification is the process of grouping documents into categories based on their content. In sentiment analysis, this classification determines the polarity of opinions (positive, negative, or neutral)[9]. This research follows a structured workflow, starting from secondary Data Collection from Kaggle and Data Exploration. The process continues with Text Preprocessing to clean and prepare the text, including Case Folding, Text Cleaning, Tokenization, Stopword Removal, and Stemming[10]. The prepared data then goes through Feature Extraction using Count-Vectorizer and TF-IDF Weighting, followed by Data Splitting (80% training, 20% testing). The main process is Algorithm Model Classification using three machine learning models: Naïve Bayes (NB), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), to analyze public sentiment. The entire process concludes with Model Evaluation and Comparison using a Confusion Matrix. These data preparation and feature extraction stages have a significant influence on improving the final results of the classification model.

2.3. Naïve Bayes Algorithm

Naïve Bayes (NB) is an algorithm often utilized to predict outcomes or calculate the probability of an event. In sentiment analysis, this algorithm operates based on the fundamental assumption that the occurrence of a word is not influenced by the occurrence of other words in the review, or in other words, the words are considered mutually independent of one another[11]. In this research, Naïve Bayes is used to build a classification model tasked with determining and categorizing the sentiment (positive, negative, or neutral) contained in Mobile Legends user reviews.

2.4. K-Nearest Neighbors Algorithm

K-Nearest Neighbors (KNN) is an algorithm that classifies user reviews based on proximity to existing training data[12]. In this research, after Mobile Legends review data is processed and transformed into numerical representation using Count-Vectorizer and TF-IDF, KNN will calculate the distance between reviews in that feature space. New reviews (test data) are then classified as Positive, Negative, or Neutral sentiment based on the polarity most commonly possessed by the Knearest reviews. This algorithm serves as one of the main comparison models to determine the performance of sentiment classification for Indonesian-language mobile game reviews[13].

2.5. Support Vector Machine Algorithm

Support Vector Machine (SVM) is a classification model that works by finding the optimal hyperplane, which is a separating plane with the largest margin, to separate data classes[14]. In the context of this sentiment research, SVM will construct boundaries between reviews labeled as Positive, Negative, and Neutral in the high-dimensional feature vector space generated from TF-IDF. The advantage of SVM lies in its ability to reliably handle data with many features (such as text data), so this model is expected to provide strong sentiment classification results and will be carefully compared with Naïve Bayes and KNN[15].

2.6. Mobile Legends: Bang Bang App

Mobile Legends: Bang Bang (MLBB) is a Multiplayer Online Battle Arena (MOBA) genre game that is highly popular and has millions of active users in Indonesia. MLBB's popularity among smartphone users generates a very large volume of reviews on the Google Play Store. These reviews are a direct reflection of public experience and satisfaction with the quality of the game and services provided. Therefore, MLBB user reviews on the Google Play Store serve as the primary data source in this research, whose sentiment will be analyzed using Support Vector Machine (SVM), Naïve Bayes, and K-Nearest Neighbors (KNN) algorithms to provide strategic insights to developers.

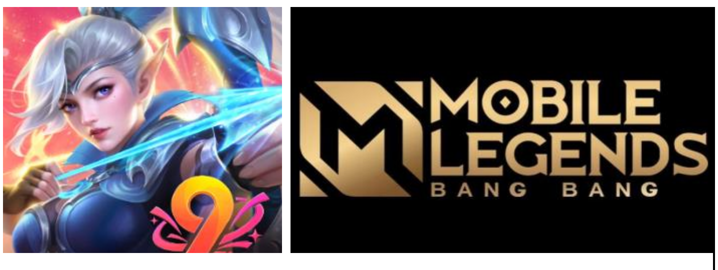


Fig 1: Mobile Legends: Bang Bang (MLBB) App display on Google Play Store

2.7. User Reviews as a Source

Reviews left by users on application distribution platforms, such as the Google Play Store, are a very important data source and are frequently utilized in research to assess the quality and acceptance level of an application[16]. This review data provides authentic and spontaneous insights from users regarding features, usability aspects, and various problems they experience while using the application. Thus, these reviews serve as the primary material in this research to conduct sentiment analysis on the Mobile Legends: Bang Bang game.

3. Research Methods

This research applies a supervised machine learning approach with a text classification method that aims to categorize sentiments of Mobile Legends: Bang Bang user reviews obtained from the Google Play Store into positive, negative, and neutral categories. The research process begins with Data Collection, followed by Data Exploration to understand its characteristics. The data then goes through Data Preprocessing for cleaning and Feature Extraction to transform text into numerical representations ready for processing[17]. After that, Data Splitting (Train-Test Split) is performed into training and testing data. The next stage is Algorithm Model Classification, where training is conducted using three main algorithms: Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). The performance of these three models is then assessed through Model Evaluation and Comparison, and the entire research sequence concludes with the formulation of Conclusions.

3.1. Naïve Bayes Algorithm

Naïve Bayes is a popular algorithm in machine learning, developed based on Bayes' Theorem. This algorithm is a simple probabilistic technique that has the fundamental assumption that each feature affecting the class value is independent (not interrelated) of one another. The characteristic of this method is a simple Bayesian network that assumes each attribute can be treated separately and does not influence each other. With an easily understood probability approach, Naïve Bayes is capable of performing classification by calculating the probability of occurrence of an event based on the independent condition of each of its attributes[18].

$$\boldsymbol{P(x \mid y) = \frac{P(y \mid x)\, P(x)}{P(y)}} \tag{1}$$

with the description:

Table 1: Description of Notations in the Naïve Bayes Algorithm

y :	kelas atau kategori yang ingin diprediksi atau diklasifikasikan (misalnya: Positif, Negatif).
x :	bukti atau fitur yang diamati dari data (misalnya: kata-kata dalam sebuah ulasan).
P(x | y) :	Disebut Likelihood. Ini adalah probabilitas untuk melihat bukti x, jika diketahui kelasnya adalah y.
P(y | x) :	Disebut Peluang Posterior. Ini adalah probabilitas yang ingin dicari: probabilitas bahwa data tersebut termasuk kelas y, setelah bukti x diamati. Inilah hasil prediksi klasifikasi.
P(x) :	Disebut Peluang Bukti. Ini adalah probabilitas untuk melihat bukti x secara keseluruhan. Nilai ini berfungsi sebagai normalisasi dalam rumus.

3.2. K-Nearest Neighbors Algorithm

K-Nearest Neighbors (KNN) is a supervised learning algorithm that works based on the principle of similarity between data. This algorithm classifies new data by examining a certain number of k nearest data points in the training dataset, then determines the class based on the majority of those nearest neighbors. KNN is a lazy learner because it does not require a complex model training process; calculations are performed during prediction by measuring the distance between data, usually using Euclidean distance or Cosine similarity. In text analysis, KNN uses vector representations such as TF-IDF to measure similarity between documents or comments[19].

$$D(x_i, y_i) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

with the description:

Table 2: Description of Notations in the K-Nearest Neighbors Algorithm

$D(x_i, y_i)$:	nilai jarak atau ketidakmiripan antara dua titik data, yaitu titik data x dan titik data y. Semakin kecil nilainya, semakin mirip kedua data tersebut.
x_i :	sebuah titik data dari set data yang sudah ada dan telah diketahui labelnya (data latih). Dalam konteks KNN, x adalah salah satu "tetangga" yang akan dibandingkan.
y_i :	sebuah titik data baru yang labelnya belum diketahui (data uji). Ini adalah data yang akan diklasifikasikan.
i :	indeks yang digunakan untuk mengakses fitur atau dimensi tertentu dari data. Jika data teks diubah menjadi vektor, i menunjuk ke fitur ke-i (misalnya, bobot TF-IDF dari kata ke-i).
n :	total jumlah fitur atau dimensi yang dimiliki oleh data x dan y. Dalam klasifikasi teks dengan Count-Vectorizer atau TF-IDF, n adalah jumlah total kata unik dalam kamus (vocabulary size).

3.3. Support Vector Machine Algorithm

Support Vector Machine (SVM) is a supervised learning technique that is highly favored in machine learning due to its high level of accuracy and quality. SVM is a popular algorithm for classification, with the main capability of identifying a hyperplane that optimally separates different classes. The SVM implementation process requires sequential training and testing stages, starting with converting text data into vector form and using the TF-IDF method for weighting. This algorithm forms a classification model using training data and can predict classes for new data that have never existed before[20].

$$f(x_d) = \sum_{i=1}^{ns} a_i y_i \vec{x_i}\,\vec{x_d} + b \quad (3)$$

with the description:

Table 3: Description of Notations in the Support Vector Machine Algorithm

ns :	total jumlah support vector yang ditemukan selama proses pelatihan model SVM. Support vector adalah titik-titik data (ulasan) dari kelas yang berbeda yang berada paling dekat dengan hyperplane pemisah. Jumlah ns seringkali jauh lebih kecil dari total data latih.
ai :	koefisien pengali Lagrange atau nilai bobot yang terkait dengan setiap support vector ke-i. Nilai a_i menunjukkan seberapa besar pengaruh support vector tersebut terhadap posisi dan orientasi hyperplane. Hanya support vector yang memiliki $a_i > 0$.
yi :	label kelas dari support vector ke-i. Dalam analisis sentimen, ini biasanya adalah polaritas sentimen (+1 untuk Positif, -1 untuk Negatif, atau sejenisnya). Nilai ini digunakan untuk menentukan arah pemisahan hyperplane.
$\vec{x_i}$:	vektor fitur dari support vector ke-i. Ini adalah data ulasan yang paling krusial yang menentukan batas pemisah antar kelas.
$\vec{x_d}$:	vektor fitur dari data baru yang labelnya belum diketahui (data uji). Fungsi keputusan SVM akan menggunakan data ini untuk memprediksi kelasnya.
b :	nilai bias atau intercept dari hyperplane. Nilai b menentukan posisi hyperplane relatif terhadap titik asal (origin) di ruang fitur. Ini adalah konstanta yang meminimalkan error klasifikasi.

3.4. Sources and Data Collection

The data in this research comes from secondary data downloaded through the Kaggle platform. The dataset contains a collection of user reviews of the Mobile Legends application and has been provided by a third party so it can be directly used in research without the need for manual data collection.

The dataset used consists of 6,000 raw data points and comprises 11 attributes, namely:

1. reviewId, a unique code for each review.
2. userName, the name of the user who wrote the review.
3. userImage, a link to the user's profile photo.
4. content, the text content of the review which is the main focus in sentiment analysis.
5. score, the star rating given by the user.
6. thumbsUpCount, the number of other users who found the review helpful.
7. reviewCreatedVersion, the app version when the review was created.
8. at, the date and time the review was posted.
9. replyContent, a response from the developer (if any).
10. repliedAt, the time when the response was given.
11. appVersion, the app version used by the user.

The dataset is downloaded in CSV or Excel format, then further processed through data cleaning stages, attribute checking, handling missing data, and text processing before being used in the sentiment analysis stage.

.

3.5. Research Stages

This research encompasses a series of structured steps with the main objective of categorizing (classifying) opinions or feelings (sentiments) from reviews given by Mobile Legends application users on the Google Play Store. The arrangement of each step is designed to create a sentiment classification model that has accurate and reliable results. The models used, namely Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), go through a training process using prepared data. After the models are developed, their capability in classifying sentiments is tested using testing data.

These stages include:

1. Data Collection Data is obtained through secondary data acquired from the Kaggle website. A total of 6,000 reviews were collected.
2. Data Exploration Includes initial data analysis and cleaning to understand the characteristics of raw data.
3. Data Preprocessing The obtained review data is then cleaned through the preprocessing stage. This process includes Case Folding (converting text to lowercase) and Text Cleaning (removing URLs, numbers, or special characters). Subsequently, Tokenization (breaking text into individual words), Stopword Removal (removing unimportant words), and Stemming (finding root words) are performed. The overall purpose of this preprocessing process is to simplify text and reduce noise (interference) in the data.
4. Feature Extraction Clean text data is transformed into numerical representations such as TF-IDF that can be processed by algorithm models.
5. Data Splitting (Train-Test Split) The data is then divided into two parts: 80% training data and 20% testing data. Training data is used to build the model, while testing data is used to evaluate its performance.
6. Algorithm Model Classification Implementation of model training with NB, KNN, and SVM algorithms.
7. Model Evaluation and Comparison Model performance is tested using testing data, and the results from all three algorithms are evaluated and compared.

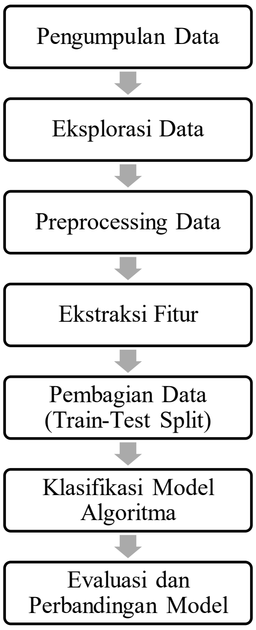


Fig 2: Research Flowchart

This diagram shows a structured research workflow, starting from data collection, followed by data exploration, data preprocessing, and feature extraction. The next stage is data splitting (dataset division), followed by algorithm model classification, before finally conducting model evaluation and comparison.

3.6. Tools and Software

In this research, the researcher uses several software and supporting tools to support the data analysis process and development of sentiment classification models. The main tools used are as follows:

1. Visual Studio Code (VS Code):

Visual Studio Code is a powerful and lightweight source code editor developed by Microsoft. In the context of this research, VS Code functions as an Integrated Development Environment (IDE) used to write, manage, and debug Python scripts related to data processing, feature extraction, and algorithm training (NB, KNN, and SVM). VS Code was chosen because it provides syntax highlighting features, an integrated terminal, and extension support that facilitates scientific work.

Fig 3: Visual Studio Code Logo

2. Pustaka Python:

The libraries used during data processing, feature extraction, and model training include:
a) Pandas, for manipulation, loading, and analysis of tabular data (dataframes).
b) NumPy, to support numerical computation, especially in matrix and array operations.
c) Matplotlib and Seaborn, for data visualization, diagram creation, and confusion matrices display in evaluation reports.
d) Re, for string operations and text cleaning using Regular Expressions.
e) Scikit-learn (sklearn), the main machine learning library used to split data (train_test_split), text feature extraction (TfidfVectorizer), implementation of classification algorithms (Naïve Bayes, KNN, SVM), evaluation and calculation of model performance metrics.
f) Sastrawi, used for Indonesian language text preprocessing (StopWordRemover and Stemmer).

g) Imbalanced-learn (imblearn), can be used optionally/conditionally to handle data imbalance through Oversampling techniques (SMOTE). h) Other Supporting Libraries, for example: pickle, os. These are needed for saving models (pickle) and file system-related operations (os).

Fig 4: Python Libraries Logo

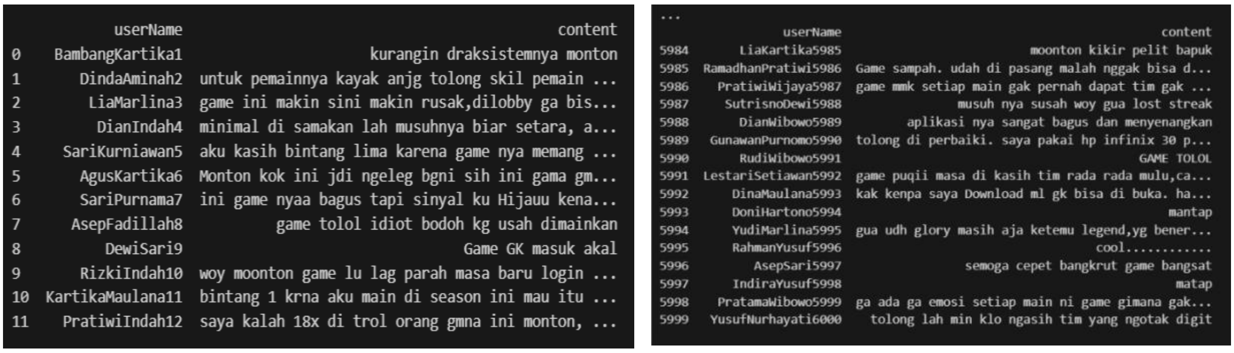
4. Results and Discussion

Naïve Bayes, KNN, and SVM models were implemented to classify Mobile Legends application user reviews into three sentiment categories: positive, neutral, and negative. The dataset consists of 6,000 reviews which are secondary data obtained from the Kaggle website.

4.1. Data Splitting

Data splitting (data slicing) is the stage of dividing the dataset into several different parts. This step is performed with the main purpose of supporting the development (training) process and testing model performance. In general, the dataset is divided into:

1. Training Data: Functions to train the model to be able to recognize and learn patterns present in the data.
2. Testing Data: Used to test and assess the working capability (performance) of the trained model.
3. Validation Data (optional): Used (if needed) to refine model parameters (hyperparameter tuning) during development.



	userName	content
0	BambangKartika1	kurangin draksistemnya monton
1	DindaAminah2	untuk pemainnya kayak anjg tolong skil pemain ...
2	LiaMarlina3	game ini makin sini makin rusak,dilobby ga bis...
3	DianIndah4	minimal di samakan lah musuhnya biar setara, a...
4	SariKurniawan5	aku kasih bintang lima karena game nya memang ...
5	AgusKartika6	Monton kok ini jdi ngeleg bgni sih ini gama gm...
6	SariPurnama7	ini game nyaa bagus tapi sinyal ku Hijauu kena...
7	AsepFadillah8	game tolol idiot bodoh kg usah dimainkan
8	DewiSari9	Game GK masuk akal
9	RizkiIndah10	woy moonton game lu lag parah masa baru login ...
10	KartikaMaulana11	bintang 1 krna aku main di season ini mau itu ...
11	PratiwiIndah12	saya kalah 18x di trol orang gmna ini monton, ...

...

	userName	content
5984	LiaKartika5985	moonton kikir pelit bapuk
5985	RamadhanPratiwi5986	Game sampah. udah di pasang malah nggak bisa d...
5986	PratiwiWijaya5987	game mmk setiap main gak pernah dapat tim gak ...
5987	SutrisnoDewi5988	musuh nya susah woy gua lost streak
5988	DianWibowo5989	aplikasi nya sangat bagus dan menyenangkan
5989	GunawanPurnomo5990	tolong di perbaiki. saya pakai hp infinix 30 p...
5990	RudiWibowo5991	GAME TOLOL
5991	LestariSetiawan5992	game puqii masa di kasih tim rada rada mulu,ca...
5992	DinaMaulana5993	kak kenpa saya Download ml gk bisa di buka. ha...
5993	DoniHartono5994	mantap
5994	YudiMarlina5995	gua udh glory masih aja ketemu legend,yg bener...
5995	RahmanYusuf5996	cool...........
5996	AsepSari5997	semoga cepet bangkrut game bangsat
5997	IndiraYusuf5998	matap
5998	PratamaWibowo5999	ga ada ga emosi setiap main ni game gimana gak...
5999	YusufNurhayati6000	tolong lah min klo ngasih tim yang ngotak digit

Fig 5: Sample User Review Dataset

4.2. Results of Pre-processing Implementation

Secondary data obtained from the Kaggle website was cleaned and prepared through a series of preprocessing stages. This process begins with Case Folding (converting all text to lowercase) and Text Cleaning (removing irrelevant elements such as URLs, numbers, and special characters). Subsequently, the text is broken down into individual words through Tokenization, followed by Stopword Removal to eliminate common words, and Stemming (finding root words). All these stages are executed to ensure the text data is clean, standardized, and ready for feature extraction in the next stage.

4.2.1 Case Folding

Table 4: Implementation Example of Case Folding

No	Before Case Folding	After Case Folding
1	Banyak drak sistem nya masa Roam Saber mati 12	banyak drak sistem nya masa roam saber mati 12

4.2.2 Text Cleaning

Table 5: Implementation Example of Text Cleaning

No	Before Text Cleaning	After Text Cleaning
1	Banyak drak sistem nya masa Roam Saber mati 12	Banyak drak sistem nya masa Roam Saber mati

4.2.3 Tokenization

Table 6: Implementation Example of Tokenization

No	Before Tokenization	After Tokenization
1	Banyak drak sistem nya masa Roam Saber mati 12	Banyak, drak, sistem, nya, masa, Roam, Saber, mati

4.2.4 Stopword Removal

Table 7: Implementation Example of Stopword Removal

No	Before Stopword Removal	After Stopword Removal
1	Banyak drak sistem nya masa Roam Saber mati 12	banyak drak sistem nya masa roam saber mati

4.2.5 Stemming

Table 8: Implementation Example of Stemming

No	Before Stemming	After Stemming
1	Banyak drak sistem nya masa Roam Saber mati 12	banyak drak sistem nya masa roam saber mati 12

4.3. Results of the Naïve Bayes

The Naïve Bayes model was implemented to classify user reviews into positive, negative, or neutral categories. Model training was conducted by splitting the data into training data and test data with an 80:20 ratio. The Naïve Bayes algorithm used is Multinomial Naïve Bayes because it is suitable for text in the form of words or discrete features. Hyperparameter tuning was performed by searching for the optimal alpha value using a validation set, and an optimal alpha of 0.9 was obtained.

Confusion matrix:

Table 9: Confusion Matrix Evaluation of Naïve Bayes

	Positive Prediction	Netral Prediction	Negatif Prediction
Positive Actual	251	16	130
Netral Actual	9	0	45
Negatif Actual	38	5	684

4.4. Results of the K-Nearest Neighbors

The K-Nearest Neighbors model was implemented to classify user reviews into positive, negative, or neutral categories. Model training was conducted by splitting the data into training data and test data with an 80:20 ratio. The KNN algorithm used is KNeighborsClassifier which predicts classes based on k nearest neighbors. Hyperparameter tuning was performed by searching for the optimal combination of k, metric, and weights using cross-validation, and optimal parameters were obtained: k=85, metric='cosine', and weights='distance.

Confusion matrix:

Table 10: Confusion Matrix Evaluation of K-Nearest Neighbors

	Positive Prediction	Netral Prediction	Negatif Prediction
Positive Actual	268	0	129
Netral Actual	15	0	39
Negatif Actual	101	0	626

4.5. Results of the Support Vector Machine

The Support Vector Machine model was implemented to classify user reviews into positive, negative, or neutral categories. Model training was conducted by splitting the data into training data and test data with an 80:20 ratio. The SVM algorithm used is SVC (Support Vector Classifier) which searches for an optimal hyperplane to separate classes with maximum margin. Hyperparameter tuning was performed by searching for the optimal combination of C and kernel using GridSearchCV, and optimal parameters were obtained: C=50.0, kernel='linear', and class_weight='balanced'.

Confusion matrix:

Table 11: Confusion Matrix Evaluation of Support Vector Machine

	Positive Prediction	Netral Prediction	Negatif Prediction
Positive Actual	288	6	103
Netral Actual	9	0	45
Negatif Actual	73	1	653

4.6. Result of Model Evaluation

Model performance evaluation was carried out using several metrics commonly used in classification testing, namely:

1. Accuracy: Measures the percentage of data successfully classified correctly by the model out of the total data.
2. Precision: Assesses how accurate the positive predictions made by the model are compared to all data predicted as positive.
3. Recall: Calculates the portion of data that should be positive that is successfully found and correctly classified by the model.
4. F1-Score: A single value that reflects the harmony (harmonic mean) between the model's precision and recall capabilities.

Table 12: Evaluation Result of All Model

	Naïve Bayes	KNN	SVM
Akurasi	79.37%	75.89%	79.88%
Presisi	77.53%	72.18%	76.54%
Recall	79.37%	75.89%	79.88%
F1-Score	77.57%	73.93%	78.06%

5. Conclusion

Sentiment analysis of 6,000 Mobile Legends user reviews obtained from the Kaggle secondary dataset was successfully implemented using a supervised machine learning approach. From the comparison results of the three algorithms, the Support Vector Machine (SVM) model overall proved to provide the best performance, with the highest values in Accuracy (79.88%), Recall (79.88%), and F1-Score (78.06%). It should be noted that the Naïve Bayes model slightly excels in the Precision metric with a value of 77.53%. Nevertheless, SVM's performance makes it the most superior algorithm in classifying Indonesian-language mobile game review sentiments compared to Naïve Bayes and K-Nearest Neighbors. These results provide practical insights for MLBB developers to understand public perspectives and serve as a strategic basis for making service improvement and application quality decisions.

Acknowledgement

The researcher expresses deep appreciation and gratitude to Prof. Dr. Amries Rusli Tanjung, S.E., M.M., Ak., as Rector of the Pelita Indonesia Institute of Business and Technology, Irwan, M. Kom, as Dean of the Faculty of Computer Science, and Gusrianty, M. Kom, as Coordinator of the Informatics Engineering Study Program, for their support and facilities during the implementation of this study. Appreciation is also directed to Dewi Nasien, M. Sc. Ph. D., as the lecturer for the related course. Sincere appreciation is extended to parents for their prayers, moral and material support, and endless motivation. Final thanks are extended to all fellow students for their assistance, ideas, and encouragement during the writing of this research.tian.

References

[1] M. A. A. T. Utami, P. Silvianti, and M. Masjkur, "Algoritme Support Vector Machine untuk Analisis Sentimen Berbasis Aspek Ulasan Game Online Mobile Legends: Bang-Bang," *Xplore: Journal of Statistics*, vol. 12, no. 1, pp. 63–77, Jan. 2023, doi: 10.29244/xplore.v12i1.1064.

[2] R. Wahyudi and G. Kusumawardhana, "Analisis Sentimen pada review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *JURNAL INFORMATIKA*, vol. 8, no. 2, 2021, [Online]. Available: http://ejournal.bsi.ac.id/ejurnal/index.php/ji

[3] A. M. Ndapamuri, D. Manongga, and A. Iriani, "Analisis Sentimen Ulasan Aplikasi Tripadvisor Dengan Metode Support Vector Machine, K-Nearest Neighbor, Dan Naïve Bayes," vol. 8, no. 1, p. 2023, 2023.

[4] Tinaliah and T. Elizabeth, "Analisis Sentimen Ulasan Aplikasi PrimaKu Menggunakan Metode Support Vector Machine," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 4, 2022, [Online]. Available: http://jurnal.mdp.ac.id

[5] C. Zaenudin Zidane, Tukino, and F. Nurapriani, "Journal of Artificial Intelligence and Engineering Applications Sentiment Analysis of User Reviews for the POLRI Super App on Google Play Store Using the Naïve Bayes Algorithm," 2025. [Online]. Available: https://ioinformatic.org/

[6] S. Alfaris and Kusnawi, "Komparasi Metode KNN dan Naïve Bayes Terhadap Analisis Sentimen Pengguna," *Indonesian Journal of Computer Science Attribution*, vol. 12, no. 5, pp. 2023–2766, 2023.

[7] E. Indrayuni, A. Nurhadi, and D. A. Kristiyanti, "Implementasi Algoritma Naïve Bayes, Support Vector Machine, dan K-Nearest Neighbors untuk Analisa Sentimen Aplikasi Halodoc," *Faktor Exacta*, vol. 14, no. 2, p. 64, Aug. 2021, doi: 10.30998/faktorexacta.v14i2.9697.

[8] J. W. Iskandar and Y. Nataliani, "Comparison of Naïve Bayes, SVM, and k-NN for Aspect-Based Gadget Sentiment Analysis," *Jurnal RESTI*, vol. 5, no. 6, pp. 1120–1126, Dec. 2021, doi: 10.29207/resti.v5i6.3588.

[9] T. Septiani Gumilar, R. Astuti, and Y. A. Wijaya, "ANALISIS SENTIMEN ULASAN APLIKASI LITA DI PLAY STORE MENGGUNAKAN ALGORITMA NAÏVE BAYES," 2024.

[10] K. Kevin, M. Enjeli, and A. Wijaya, "Analisis Sentimen Pengunaaan Aplikasi Kinemaster Menggunakan Metode Naïve Bayes," *Jurnal Ilmiah Computer Science*, vol. 2, no. 2, pp. 89–98, Jan. 2024, doi: 10.58602/jics.v2i2.24.

[11] N. Aditiya *et al.*, "Sistemasi: Jurnal Sistem Informasi Analisis Sentimen Kepuasan Masyarakat terhadap Aplikasi 'INFO BMKG' menggunakan Naïve Bayes, SVM, dan KNN Sentiment Analysis of Public Satisfaction with the 'INFO BMKG' Application using Naïve Bayes, SVM, and KNN," 2025. [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[12] K. Mustaqim, F. A. Amaresti, and I. N. Dewi, "Analisis Sentimen Ulasan Aplikasi PosPay untuk Meningkatkan Kepuasan Pengguna dengan Metode K-Nearest Neighbor (KNN)," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 1, pp. 11–20, Jun. 2024, doi: 10.29408/edumatic.v8i1.24779.

[13] N. Alvionika, S. Faisal, R. Rahmat, and A. F. N. Masruriyah, "Analisis Sentimen Pada Komentar Instagram Provider By.U Menggunakan Metode K-Nearest Neighbors (KNN)," *Jurnal Algoritma*, vol. 21, no. 2, pp. 50–63, Nov. 2024, doi: 10.33364/algoritma/v.21-2.1672.

[14] A. Julia and D. Nasien, "Cyberbullying Sentiment Analysis On Tweets In X (Twitter): A Case Study Using Support Vector Machine Method," 2025. [Online]. Available: www.___.

[15] V. Fitriyana *et al.*, "Analisis Sentimen Ulasan Aplikasi Jamsostek Mobile Menggunakan Metode Support Vector Machine," 2023.

[16] L. Rhomaningtias, A. Khairunisa, S. Shella, M. Wara, and K. M. Hindrayani, "ANALISIS SENTIMEN ULASAN APLIKASI SMILE INDONESIA MENGGUNAKAN METODE NAÏVE BAYES DAN SUPPORT VECTOR MACHINE (SVM)," 2025, doi: 10.52972/hoaq.vol16no1.

[17] A. Rahman Isnain, N. Hendrastuty, and L. Andraini, "Comparison of Support Vector Machine and Naïve Bayes on Twitter Data Sentiment Analysis," vol. 6, no. 1, 2021.

[18] R. D. Pebrianti, "ANALISIS SENTIMEN MASYARAKAT PLATFORM X TERHADAP KORUPSI PT. PERTAMINA (PERSERO) MENGGUNAKAN METODE SVM," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 2, Apr. 2025, doi: 10.23960/jitet.v13i2.6399.

[19] A. Baita and N. Cahyono, "ANALISIS SENTIMEN MENGENAI VAKSIN SINOVAC MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN K-NEAREST NEIGHBOR (KNN)," 2021.

[20] K. A. Rokhman, Berlilana, and P. Arsi, "PERBANDINGAN METODE SUPPORT VECTOR MACHINE DAN DECISION TREE UNTUK ANALISIS SENTIMEN REVIEW KOMENTAR PADA APLIKASI TRANSPORTASI ONLINE," 2021.