

MOBA GAME REVIEW SENTIMENT ANALYSIS USING SUPPORT VECTOR MACHINE ALGORITHM

Alif F Panjalu¹, Syariful Alam², M. Imam Sulistyono³

^{1,2,3}Program Studi Teknik Informatika Sekolah Tinggi Teknologi Wastukencana Purwakarta
Email: ¹aliffajar10@wastukencana.ac.id, ²syarifulalam@wastukencana.ac.id, ³imams@wastukencana.ac.id

(Received: 21 July 2023, Revised: 26 July 2023, Accepted: 31 July 2023)

Abstract

MOBA as one of the many popular subgenres today, Mobile Legend, Arena of Valor, League of Legend and Lokapala based on the large number of downloads can be mentioned as popular MOBA games, but the rating of these four applications is below 4.0 on the Google Play platform Store, this happens because some users may think that this mobile MOBA game has several advantages, but also some disadvantages that affect ratings. This study aims to determine the results of sentiment analysis on mobile MOBA games using Google Play Store reviews. Then it is processed using Python programming to create a model with the linear kernel Support Vector Machine (SVM) algorithm to classify the dataset. From the results of the classification model test using 19,579 data, where there were 10,017 positive sentiment data and 9,562 negative sentiment data and the distribution of train data and test data was 70%: 30%, obtained an accuracy of 82.64% and then re-evaluated using the Cross Validation method using 5 times so that an accuracy of 83.38% is obtained

Keywords: MOBA, Sentiment Analysis, Machine Learning, SVM, Cross Validation

This is an open access article under the [CC BY](#) license.



**Corresponding Author: Alif Fajar Panjalu*

1. INTRODUCTION

In this modern era, games are heavily influenced by technological advances. The development of increasingly complex and captivating games that can be played connected using an internet connection that can be played simultaneously with millions of players in Indonesia and even throughout the world made possible by these technological advances. Games are still a big phenomenon today, mainly because of the widespread adoption of technology and the increasing penetration of the internet around the world. [1] Apart from that, games are developing not only because of the development of the internet but also the high interest of the game players.

One of them is the MOBA genre game which is still a trend today. MOBA stands for (Multiplayer Online Battle Arena) which is a game with the Real Time Strategy (RTS) genre which is currently loved by players today. MOBA itself tends to be played on PCs, but along with the development of the internet, developers are starting to venture into the world of gadgets[1]. The MOBA game is the trend so far, but

the rating of these four game applications is below 4.0, namely: Mobile Legend: Bang-Bang (3.7), Garena AoV: Arena of Valor (3.9), League of Legends: Wild Rift (3.4) and Lokapala (3.8) on Google Play Store Platforms.

The problem that will be discussed in this research is an application with a rating of less than 4.0 on the Google Play Store platform that most users are satisfied and most users are not satisfied with the game application. How do developers find out what features players don't like that need to be fixed immediately and what features players like so that these features can be improved Therefore in this research is how to take advantage of Google Play Store user reviews of MOBA games and sentiment analysis is carried out to gain insight from opinion about what players like and hate about the game.

Sentiment analysis or Opinion mining is the process of understanding, extracting and processing textual data automatically to obtain sentiment information contained in an opinion sentence[2]. Sentiment analysis is classifying the polarity of the text in documents, sentences, or opinions. Polarity

means whether the text in a document, sentence, or opinion has positive or negative aspects. [3] Sentiment analysis involves the use of Natural Language Processing (NLP) methods[4]. Natural Language Processing or Natural Language Processing is the creation of programs that have the ability to understand human language[5]. Sentiment analysis is one of the strategies used in the text categorization field. The technique of classifying text uses a machine learning model approach. Classification is a model/function that divides things into certain classes in order to determine the class of the item based on the training and testing processes that have been completed [6]. One classification approach is Support Vector Machine (SVM).

The SVM approach makes it possible to calculate linear problems by applying a mathematical transformation to the study space using the multiplication function [7]. SVM is a learning system that studies patterns by understanding a hypothetical space that seeks to construct linear functions in a high-dimensional feature space. This algorithm takes advantage of learning bias and optimization to get better results [8], The SVM concept can simply be interpreted as an attempt to find the best "hyperplane" to separate the two categories in the "input space". By measuring the margin value and determining the maximum point to find the best hyperplane [9]. SVM is a model that comes from statistical learning theory which will give better results than other methods [10], Support Vector Machine (SVM) outperforms the Nave Bayes, Rocchio, C4.5, and K-Nearest Neighbor classification algorithms [11]. Therefore in this study the SVM algorithm is used as algorithms that can support the text classification.

2. RESEARCH METHOD

The research methodology contains the steps taken by the researcher to conduct the research. In this study, after the researcher carried out the data collection stage, the researcher used the KDD method to analyze the sentiments of user reviews. KDD itself stands for Knowledge Discovery in Database. The KDD method itself has 5 stages, there are Selection, Pre-processing, Transformation, Data Mining, Evaluation, along with the research flowchart below:

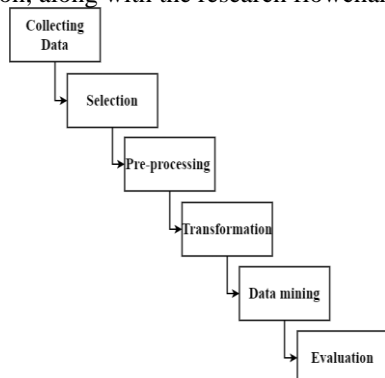


Figure 1. research methodology based on KDD

2.1 Collecting Data

Data collection was carried out by scraping using the Python programming on Google Colab with the help of the google-play-scraper library on the application distribution platform for the Android Google Play Store, data collection from case studies Mobile Legend: Bang-Bang, Garena AoV: Arena of Valor, League of Legends: Wild Rift and Lokapala, scraping was carried out on May 25, 2023 based on the review language, namely Indonesia, country of origin, namely Indonesia with the fields taken, namely: username, content, score, at, the following results of collecting data:

Table 1. example of scraping data results

userName	content	score	at
Rendi Febariansyah	Game Ini Bagus rame eventnya, suka banget 100%	5	25/ 05/ 2023 07:13
Muhammad Fathur Rizki	Update terus makin berat ini game, jelek 😞	2	25/ 05/ 2023 07:12

2.2 Selection

Selection is a stage to select the data needed in research. The existence of the selection stage can make the data processing carried out according to the objectives of the study [12]. There are 2 steps at this selection stage, the selection processes are as follows:

Labeling: In this labeling step, each row of data that has been processed in the previous step is categorized by adding a "label" field whose contents consist of positive sentiments by score 4-5 is labeled = "1" and negative by score 1-2 labeled = "-1" sentiments by score 3 which are categorized manually using Microsoft Excel. An example of labeling results may be seen in the table below:

Table 2. example of labeling

userName	content	score	at	label
Rendi Febariansyah	Game Ini Bagus rame eventnya, suka banget 100%	5	25/ 05/ 2023 07:13	1
Muhammad Fathur Rizki	Update terus makin berat ini game, jelek 😞	2	25/ 05/ 2023 07:12	-1

Subset field data: This step is carried out to sort out several data fields, this process is carried out by selecting the relevant fields for text classification, the researcher selects two data fields, namely the

"content" field as text data and the "label" field as the category of sentiment in text data. An example of the outcomes of subset field data may be seen in the table below:

Table 3. example of subset field data

content	label
Game Ini Bagus rame eventnya, suka banget 100%	1
Update terus makin berat ini game, jelek 😞	-1

2.3 Pre-processing

At this pre-processing stage, the "content" text data field is cleaned, as well as correcting word errors in the text data so that it helps in matching words, from several processes, namely: Clean Text, Normalize Word, Stopword Removal, Tokenization and Stemming, the following are the processes carried out in the pre-processing stage:

Clean text: this process is done by erasing other elements besides letters a to z such as numbers and other punctuation symbols in the text data, then swapping capital letters with non-capital letters.

Table 4. example of Clean Text

Raw Text	Clean Text
Game Ini Bagus rame eventnya, suka banget 100%	game ini bagus rame eventnya suka dah
Update terus makin berat ini game, jelek 😞	update terus makin berat ini game jelek

Normalized Word: this process is done by changing the words in the text data into a standard form, such as words that have the same meaning but are written in a different way, and then change them into standard word forms.

Table 5. example of Normalization Word.

Clean Text	Normalization
game ini bagus rame eventnya suka dah	game ini bagus ramai eventnya suka dah
update terus makin berat ini game jelek	update selalu semakin berat ini game buruk

Stopword Removal: this process is done by filtering out connecting or less meaningful words and words that rarely appear so that the feature space can be reduced when feature extraction is performed.

Table 6. example of Stopword Removal

Normalization	Stopword Removal
game ini bagus ramai eventnya suka dah	game bagus eventnya suka
update selalu semakin berat ini game buruk	update berat game buruk

Tokenization: this process is done by breaking down text or documents into small units is called tokens. Tokens are the smallest units in a text or document

that can be processed and analyzed separately. The token here is in the form of a word resulting from the process of splitting text based on spaces.

Table 7. example of Tokenization

Stopword	Tokenization
game bagus eventnya suka	game, bagus, eventnya, suka
update berat game buruk	update, berat, game, buruk

Stemming: this process is done by changing the words in the text into basic words or basic forms by removing affixes and the prefixes to the word.

Table 8. example of Stemming

Tokenization	Stemming
game, bagus, eventnya, suka	game bagus event suka
update, berat, game, buruk	update berat game buruk

2.4 Transformation

In the Transformation stage, it changes the form of text data into numeric data. This stage is the stage of adjusting the data used so that the data can be processed at the data mining stage. Because the SVM algorithm can only process numeric data in the form of tables [13], at this stage the authors use feature extraction Term Frequency-Inverse Document Frequency (TF-IDF) word weighting. Term Frequency or (TF), namely the weight of the term in a data, in this case news data, which is calculated based on the n or the level of occurrence in the data stack. TF is the number of words in one document, the greater the TF value of a document, the higher the document weight. The Inverse Document Frequency (IDF) function is to reduce the value of the previous TF. If a word appears frequently in the document, then the TF value or the weight will be smaller [7]. Dan hasil akhir dari metode ini adalah matriks output yang meliputi kata-kata unik dan nilai-nilai yang dihasilkan pada TF-IDF dari setiap kata pada seluruh data [14]. The value of the equation for determining TF (1) is shown below:

$$tf_{(t,d)} = f_{(t,d)} \quad (1)$$

Where $tf_{(t,d)}$ the appearance of the word t in the document d

Word weighting or what is called (term weighting) calculates the dimensions of the feature vector based on its (IDF) Inverse Document Frequency value. IDF can be formulated (2) as follows:

$$idf_t = \log \left(\frac{n}{df_t} \right) \quad (2)$$

Where idf_t looks for Inverse Document Frequency based on line and term df_t number of documents containing term t, n total documents.

As a result, the TF-IDF gives the weight of a word based on the TF value and the inverse value DF. Here are the equation values (3) for determining weight with TF-IDF:

$$tf_{(\tau,d)} * idf_{(\tau)} \quad (3)$$

An example of the results of the calculation of the TF-IDF can be seen in the table below:

Table 9. example of TF-IDF result

TF	n		DF	IDF	TF * IDF	
	D1	D2			D1	D2
game	1	1	2	0.602059	0.602059	0.602059
bagus	1	0	1	0.903089	0.903089	0
event	1	0	1	0.903089	0.903089	0
suka	1	0	1	0.903089	0.903089	0
update	0	1	1	0.903089	0	0.903089
berat	0	1	1	0.903089	0	0.903089
buruk	0	1	1	0.903089	0	0.903089

2.5 Data Mining

This Data Mining starts the process of processing data for text classification using Machine learning algorithms. In this study, researchers used the Linear kernel SVM algorithm as a machine learning model that classifies text, the process of making this classification model uses 70% of the train data and 30% of the test data using processed data at the Transformation stage, following the SVM with the kernel linear as formula (4):

$$K(x_i, x) = x_i^T * x \quad (4)$$

The hyperplane line is in the center of the nearest data's maximum margin value. To obtain a hyperline, use the following equation (5):

$$f(x) = x \cdot w + b \quad (5)$$

where $f(x)$ is a fictitious function that attempts to distinguish classes using a straight line. The feature vector of the input data is denoted by x . w denotes the weight vector that SVM must learn. b is the bias or offset that must also be learnt.

In this case, w_i is the weight vector (i weight vector data) that defines the amount and direction of the separation between the two classes in the i feature space data. While b is the bias constant (bias term) that controls the separator's location with regard to the x_i axis. The formula for comparing the two groups is shown below (6)(7):

$$x_i \cdot w_i + b \geq 1 \text{ for } y_i = 1 \quad (6)$$

$$x_i \cdot w_i + b \leq -1 \text{ for } y_i = -1 \quad (7)$$

2.6 Evaluation

And the last is the Evaluation stage, at this stage a model evaluation is carried out so that the performance of the model that has been made is obtained, as well as making a model with different training data so that, the following evaluations are below:

Confusion Matrix: is an evaluation process where after the classification model is obtained, the model is then tested using the Confusion matrix, the following is the Confusion matrix table:

Table 10. Confusion Matrix

	Prediction	
	Negative	Positive
Actual Negative	TN (True Negative)	FN (False Negative)
Actual Positive	FP (False Positive)	TP (True Positive)

where the actual TN (True Negative) data labeled as true Negative is predicted negative, FN (False Negative) is the actual data labeled Negative but predicted positive, TP (True Positive) is actual data labeled Positive is correctly predicted Positive, FP (False Positive) is actual data labeled Positive but predicted Negative.

The confusion matrix is a tool that can be used to calculate several scores on a performance matrix such as accuracy, precision, recall, F1. The following formula calculates precision positive (6), precision negative (7), recall positive (8), recall negative (9), F1 positive (10), F1 negative (11), and accuracy (12):

$$\text{Precision Positive (PP)} = \frac{TP}{(TP+FP)} \quad (6)$$

$$\text{Precision Negative (PN)} = \frac{TN}{(TN+FN)} \quad (7)$$

$$\text{Recall Positive (RP)} = \frac{TP}{(TP+FN)} \quad (8)$$

$$\text{Recall Negative (RN)} = \frac{TN}{(TN+FP)} \quad (9)$$

$$F1 \text{ Positive} = 2 * \frac{PP * RP}{PP + RP} \quad (10)$$

$$F1 \text{ Negative} = 2 * \frac{PN * RN}{PN + RN} \quad (11)$$

$$\text{Accuracy} = \frac{TP+TN}{\text{Total Test Data}} \quad (12)$$

K-Fold Cross Validation: is an evaluation process for dividing data into several parts (folds) as many as K to determine the train data and test data, this approach is generally used to produce model predictions and measure the accuracy of a predictive model when utilized in practice. K-fold cross validation is a cross validation approach that divides data into K equal-sized sections. To eliminate bias from the data, use k-fold=5 cross-validation [15].

After that a average accuracy will be obtained from the results of the train and test models (fold) as much as K, below is a visualization for the k-fold cross validation method:



Figure 2. Cross Validation 5 Folds
Source: Rukshan Manoratha (Towardsdatascience.com)

3. RESULT AND DISCUSSION

This section provides the results and discussion of research using the Knowledge Discovery in Database (KDD) data processing method to gain knowledge from data that has been processed through multiple stages of KDD, as well as the results and discussion below:

3.1 Collecting Data

The results of scraping data at the data collecting stage retrieved 5,000 Google Play Store reviews from each case study analyzed, namely Mobile Legends: Bang-Bang, Arena of Valor, League of Legends: Wild Rift, Lokapala, which add up all the totals. review data as much as 20000 data. Total data can be seen at table below:

Table 11 Collecting data

Nama	Total
Mobile Legends	5000
Arena of Valor	5000
League of Legends	5000
Lokapala	5000
Total Keseluruhan Data	20000

3.2 Selection

After the data scraping stage is carried out, it is continued at the selection stage where there are 2 processes, namely: labeling and subset field column, where the results of this labeling process result in a total of 10,204 positive sentiment review data and 9,796 negative sentiment review data, and the selection of the "content" field" as text data and "label" as a text data category field. Total data after selection stage can be seen at table below:

Table 12 Total data after Selection stage

Nama	Positif	Negatif	Total
Mobile Legends	2437	2563	5000
Arena of Valor	2575	2425	5000
League of Legends	2061	2939	5000

Lokapala	3281	1719	5000
Total Keseluruhan Data	10204	9796	20000

3.3 Pre-processing

After the selection stage, the data fields containing the reviews are processed at the pre-processing stage, where after going through 5 processes, namely clean text, Normalization Word, Stopword Removal, Tokenization and Stemming, the data is reduced to 19579 data, with 10017 positive labeled data and 9562 negative labeled data, can be seen at this table below:

Table 13. Total data after Pre-processing stage

Nama	Positif	Negatif	Total
Mobile Legends	2332	2545	4877
Arena of Valor	2544	2333	4877
League of Legends	2012	2918	4930
Lokapala	3191	1701	4892
Total Keseluruhan Data	10017	9562	19579

3.4 Transformation

Field review data that has gone through the pre-processing stage is converted into numerical form using TF-IDF word weighting so that 12953 feature terms (word features) are obtained from each word in the document, along with the number of word features obtained after processing TF-IDF:

Table 14. Total Word Feature using TF-IDF

	Dokumen Teks	Feature Term
jumlah	19759	12953

3.5 Data Mining

After the data is changed in the form of word weights, the data can be processed to the data mining stage. In this study, the researcher used the SVM kernel linear algorithm as the algorithm used to create machine learning models to be able to classify review sentiments, with the distribution of train data as much as 70% of the total. documents as many as 19579 documents. Total data can be seen at table below:

Table 15. Total Train and Test data for model SVM

	Documen (X)	label (Y)
Train	13705	13705
Test	5874	5874

3.6 Evaluation

After the data mining stage was carried out and the classification model was obtained, then the model was evaluated with test data as much as 30% of the total 19579 documents using the confusion matrix table, the following is the model test table using the confusion matrix:

5. REFERENCE

- [1]. D. Sinaga and C. Jatmoko, "ANALISIS SENTIMEN UNTUK MENGETAHUI KESAN PLAYER GAME MOBILE LEGENDS MENGGUNAKAN NAÏVE BAYES CLASSIFIER SENTIMENT ANALYSIS TO KNOW IMPRESSIONS PLAYER GAME MOBILE LEGENDS USING NAÏVE BAYES CLASSIFIER 1)," *Semnias Lppm*, pp. 540–547, 2020, [Online]. Available: www.netlytic.org
- [2]. A. Nurzahputra and A. Muslim, "Analisis Sentimen pada Opini Mahasiswa Menggunakan Natural Language Processing," *Seminar Nasional Ilmu Komputer*, pp. 114–118, Oct. 2016.
- [3]. H. Simorangkir and K. M. Lhaksana, "Analisis Sentimen pada Twitter untuk Games Online Mobile Legends dan Arena of Valor dengan Metode Naïve Bayes Classifier," *eProceedings of Engineering*, vol. Vol. 5, no. No. 3, pp. 8131–8140, Dec. 2018, [Online]. Available: <https://dev.twitter.com>.
- [4]. S. Fachri Pane and J. Ramdan, "Pemodelan Machine Learning : Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Menggunakan Data Twitter," *Jurnal Sistem Cerdas*, vol. Vol.05, no. No.01, pp. 12–20, 2022, [Online]. Available: <https://t.co/IEducGFuuJ>, doi: <https://doi.org/10.37396/jsc.v5i1.191>.
- [5]. A. Adjie Wicaksono, R. Yusuf, T. Aristi Saputri, and S. Dharma Wacana, "PENERAPAN NATURAL LANGUAGE PROCESSING BERBASIS VIRTUAL ASSISTANT PADA BAGIAN ADMINISTRASI AKADEMIK STMIK DHARMA WACANA," *IRobot*, vol. Vol.05, pp. 33–47, 2021, doi: <https://doi.org/10.53514/ir.v5i1.228>.
- [6]. W. Widayani and H. Harliana, "Perbandingan Kernel Support Vector Machine Dalam Melakukan Klasifikasi Penundaan Biaya Kuliah Mahasiswa," *Jurnal Sains dan Informatika*, vol. 7, no. 1, pp. 20–27, Jun. 2021, doi: 10.34128/jsi.v7i1.268.
- [7]. A. Nabillah, S. Alam, and M. G. Resmi, "Twitter User Sentiment Analysis Of TIX ID Applications Using Support Vector Machine Algorithm," 2022.
- [8]. R. Nanda, E. Haerani, S. K. Gusti, and S. Ramadhani, "Klasifikasi Berita Menggunakan Metode Support Vector Machine," *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 5, no. 2, 2022, doi: <https://doi.org/10.32672/jnkti.v5i2.4193>
- [9]. G. R. Ditami, E. F. Ripanti, and H. Sujaini, "Implementasi Support Vector Machine untuk Analisis Sentimen Terhadap Pengaruh Program Promosi Event Belanja pada Marketplace," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. Vol.8, no. No.3, pp. 508–516, Dec. 2022, doi: <https://doi.org/10.26418/jp.v8i3.56478>.
- [10]. D. Darwis, E. Shintya Pratiwi, A. Ferico, and O. Pasaribu, "PENERAPAN ALGORITMA SVM UNTUK ANALISIS SENTIMEN PADA DATA TWITTER KOMISI PEMBERANTASAN KORUPSI REPUBLIK INDONESIA," 2020, doi: <https://doi.org/10.21107/edutic.v7i1.8779>
- [11]. A. Zaiem Praghakusma and N. Charibaldi, "Komparasi Fungsi Kernel Metode Support Vector Machine untuk Analisis Sentimen Instagram dan Twitter (Studi Kasus: Komisi Pemberantasan Korupsi)," vol. 9, no. 2, pp. 33–42, 2021, doi: 10.12928/jstie.v8i3.xxx.
- [12]. S. C. Nurzanah, S. Alam, and T. I. Hermanto, "ANALISIS ASSOCIATION RULE UNTUK IDENTIFIKASI POLA GEJALA PENYAKIT HIPERTENSI MENGGUNAKAN ALGORITMA APRIORI (STUDI KASUS: KLINIK RAFINA MEDICAL CENTER)," *Jurnal Informatika dan Komputer) Akreditasi KEMENRISTEKDIKTI*, vol. 5, no. 2, 2022, doi: 10.33387/jiko.
- [13]. F. Romadoni, Y. Umaidah, and B. N. Sari, "Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 2, pp. 247–253, Jul. 2020, doi: 10.32736/sisfokom.v9i2.903.
- [14]. D. Hana Amalia and W. Yustanti, "Klasifikasi Buku Menggunakan Metode Support Vector Machine pada Digital Library," *Journal of Informatics and Computer Science*, vol. 03, 2021, [Online]. Available: <https://opac.unesa.ac.id/>, doi: <https://doi.org/10.26740/jinacs.v3n01.p55-61>
- [15]. H. Azis, P. Purnawansyah, F. Fattah, and I. P. Putri, "Performa Klasifikasi K-NN dan Cross Validation Pada Data Pasien Pengidap Penyakit Jantung," *ILKOM Jurnal Ilmiah*, vol. 12, no. 2, pp. 81–86, Aug. 2020, doi: 10.33096/ilkom.v12i2.507.81-86, doi: <https://doi.org/10.33096/ilkom.v12i2.507.81-86>