JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

Analisis Sentimen Untuk Mengukur Ulasan Pengguna Aplikasi Mobile Legend Menggunakan Algoritma Naive Bayes, SVM, Random Fores, Decision Tree, dan Logistic Regression

Nevita Cahaya Ramadani
Magister Ilmu Komputer, Universitas Amikom Purwokerto
e-mail: nevitacahayar1@gmail.com

Abstrak

Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna aplikasi Mobile Legend di Google Play Store menggunakan algoritma machine learning, termasuk Naïve Bayes, Support Vector Machine (SVM), Random Forest, Decision Tree, dan Logistic Regression. Dari hasil scraping data pada 3989 ulasan, ditemukan kecenderungan bahwa pengguna Mobile Legend di Indonesia cenderung bersikap netral terhadap aplikasi ini, dengan 1265 komentar netral, 1115 komentar positif, dan 1204 komentar negatif. Melalui evaluasi model, SVM menunjukkan akurasi tertinggi sebesar 87%, mengungguli algoritma lainnya seperti Random Forest, Naïve Bayes, Decision Tree, dan Logistic Regression. Kesimpulan penelitian ini menunjukkan bahwa SVM merupakan pilihan yang sangat baik untuk melakukan analisis sentimen terhadap ulasan pengguna aplikasi Mobile Legend. Hasil ini dapat menjadi panduan bagi pengembang untuk memahami respons pengguna dan meningkatkan kualitas aplikasi berdasarkan temuan analisis sentimen.

Kata kunci: Analysis, Machine Learning, Netral

Abstract

This research aims to analyze the sentiment of user reviews of the Mobile Legend application on the Google Play Store using machine learning algorithms, including Naïve Bayes, Support Vector Machine (SVM), Random Forest, Decision Tree, and Logistic Regression. From the results of scraping data on 3989 reviews, it was found that Mobile Legend users in Indonesia tend to have a neutral attitude towards this application, with 1265 neutral comments, 1115 positive comments and 1204 negative comments. Through model evaluation, SVM shows the highest accuracy of 87%, outperforming other algorithms such as Random Forest, Naïve Bayes, Decision Tree, and Logistic Regression. The conclusion of this research shows that SVM is an excellent choice for conducting sentiment analysis on user reviews of the Mobile Legend application. These results can serve as a guide for developers to understand user responses and improve application quality based on sentiment analysis findings.

Keywords: Analysis, Machine Learning, Neutral

1. INTRODUCTION

Dalam era digital saat ini, aplikasi mobile telah menjadi bagian integral dari kehidupan sehari-hari banyak pengguna. Mobile Legends, sebagai salah satu game mobile yang populer, mendapatkan perhatian besar dari pengguna di seluruh dunia. Keberhasilan suatu aplikasi tidak hanya dapat diukur dari performa teknisnya, tetapi juga dari pengalaman pengguna yang diungkapkan dalam ulasan.

Analisis sentimen merupakan metode yang dapat memberikan wawasan mendalam tentang bagaimana pengguna merasakan dan merespons aplikasi. Dalam konteks ini, algoritma machine learning, seperti *Naive Bayes, SVM, Random Forest,*

Decision Tree, dan *Logistic Regression*, digunakan untuk mengukur sentimen dari ulasan pengguna Mobile Legends.

Seperti pada penelitian sebelumnya oleh Giovani et al [1] yang berjudul ”Analisis Sentimen Aplikasi Ruang Guru di Twitter”. Tujuan penelitian ini adalah untuk mengetahui keberhasilan suatu aplikasi dengan melakukan analisis sentimen terhadap aplikasi. Menghasilkan bahwa aplikasi optimasi terbaik dalam model ini adalah algoritma PSO berbasis SVM dengan nilai akurasi sebesar 78,55% dan AUC sebesar 0,853. Penelitian ini berhasil mendapatkan algoritma yang efektif dan terbaik dalam mengklasifikasikan komentar positif dan komentar negatif terkait dengan aplikasi Ruang Guru.

Penelitian selanjutnya oleh Adhi Putra [2] yang berjudul ”Analisis Sentimen Pada Ulasan Pengguna Aplikasi Bibit Dan Bareksa Dengan Algoritma KNN”. Penelitian ini bertujuan untuk menganalisa sentimen pada ulasan pengguna aplikasi investasi online yaitu bibit dan bareksa. Berdasarkan hasil yang diperoleh dari tahapan modelling dengan menggunakan algoritma knearest neighbors dan perbandingan 60:40 untuk data training dan data testing, maka nilai akurasi precision dan recall yang dihasilkan dari tiap aplikasi yaitu untuk bibit 85,14% , 91,91%, dan 76,44% sedangkan untuk bareksa yaitu 81,70% , 87,15%, 75,73%.

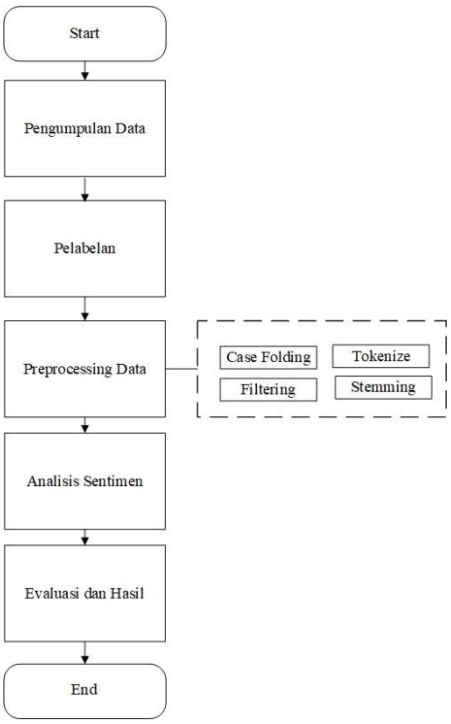
Penelitian selanjutnya oleh Wahyudi & Kusumawardana [3] yang berjudul ”Analisis Sentimen pada review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine”. Tujuan penelitian ini adalah untuk menganalisis review pengguna aplikasi Grab pada Google Play Store, menggunakan analisis sentimen. analisis review pengguna ini menggunakan metode *Support Vector Machine* (SVM). Menghasilkan analisis menggunakan *Support Vector Machine* menghasilkan akurasi 85,54% dan Hasil Review positif yang paling sering diulas adalah “ovo”, sedangkan review negatif yang paling sering diulas adalah “driver”.

Penelitian ini bertujuan untuk mengaplikasikan berbagai algoritma machine learning untuk mengukur sentimen pengguna terhadap aplikasi Mobile Legends. Dengan menganalisis ulasan pengguna, kita dapat memberikan wawasan tentang bagaimana aplikasi ini diterima oleh masyarakat. Hasil evaluasi model akan membantu dalam menentukan keefektifan dan kehandalan masing-masing algoritma dalam mengklasifikasikan sentimen. Kesimpulan dari penelitian ini diharapkan dapat memberikan pandangan yang berguna untuk pengembangan dan pemeliharaan aplikasi Mobile Legends.

2. RESEARCH METHOD

Penelitian ini, menganalisis review pengguna aplikasi Mobile Legend yang terdapat pada *Google Playstore* menggunakan algoritma *Support Vectore Machine* (SVM), *Naïve Bayes*, *Random Forest*, *Decision Tre*e dan *Logistic Regression*, dengan tahapan penelitian yang dapat dilihat pada gambar 1.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index



Gambar 1. Metode Penelitian

a. Pengumpulan Data

Data yang digunakan adalah review aplikasi Mobile Legend di playstore. Pengumpulan data menggunakan teknik *scraping* dengan library python *google play scraper*. Data yang diambil sejumlah 10.000 review dengan sorting most relevant. Data ini diambil pada tanggal 16 Januari 2024.

b. Pelabelan

Proses pelabelan ini dilakukkan supaya algoritma *Naïve Bayes, Support Vectore Machine, Random Forest, Decision Tree*, dan *Logistic Regression* dapat mempelajari dataset. Kelas yang terdiri dari 3 yaitu 1 untuk Positif, -1 untuk Negatif, dan 0 untuk Netral [4] .

c. Preprocessing Data

Proses preprocessing data merupakan langkah penting dalam analisis sentiment dan pengolahan teks secara umum. Tujuan dari preprocessing data adalah untuk membersihkan dan menyiapkan data teks sehingga dapat diolah dengan baik oleh model machine learning. Beberapa Langkah umum dalam preprocessing data teks, yaitu [5]:

- *Casefolding*

  Yaitu menghapus url dari kolom konten, merubah teks jadi *lower case*, menghapus mention, menghapus hashtag, menghapus next karakter, menghapus tanda baca, menghapus *extra whitespace*.

- *Tokenize*

  Yaitu suatu proses memecah teks atau kalimat menjadi unit yang lebih kecil yang disebut sebagai "token." Token dapat berupa kata, frasa, atau entitas lainnya, seperti angka atau simbol. Tujuan utama dari tokenisasi adalah untuk mempermudah analisis dan pemrosesan teks.

- *Filtering* (*Stopword Removal*)

  Yaitu tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting).

  Stopword adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopword* dalam bahasa Indonesia adalah “yang”, “dan”, “di”, “dari”, dll. Makna di balik penggunaan *stopword* yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya.

- *Stemming*

  Yaitu teknik pemrosesan bahasa alami yang menurunkan infleksi kata ke bentuk akarnya, sehingga membantu dalam pra-pemrosesan teks, kata, dan dokumen untuk normalisasi teks.

d. Analisis Sentimen

Pada analisis sentimen dilakukan dengan 3 algoritma yaitu algoritma *Naïve Bayes, algoritma Support Vectore Machine, Random Forest, Decision Tree,* dan *Logistic Regression*. Berikut adalah penjelasan mengenai masing-masing algoritma:

- *Naïve Bayes*

  Yaitu metode klasifikasi yang berdasarkan teorema probabilitas Bayes. Ini termasuk dalam kategori algoritma pembelajaran mesin yang menggunakan metode pembelajaran supervisi. Algoritma ini populer untuk tugas klasifikasi teks dan pengelompokan dokumen, tetapi juga dapat diterapkan pada berbagai jenis data [6].

- *Supper Vectore Machine* (SVM)

  Yaitu algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. Tujuan utama dari SVM adalah untuk membangun pemisah atau hiperplane yang optimal dalam ruang berdimensi tinggi untuk memisahkan instance dari kelas yang berbeda [7].

- *Random Forest*

  Yaitu suatu algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi, regresi, dan pengurutan. Algoritma ini merupakan bentuk ensemble learning yang menggabungkan beberapa model pembelajaran (pohon keputusan) untuk membuat prediksi yang lebih akurat dan stabil [8].

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

Setelah dilakukkan pelabelan selanjutnya dilakukan proses *feature extraction* dilakukkan dengan mengubah teks ke representasi vector dengan TF-IDF (*Term Frequency-Inverse Document Frequency*). Rumus untuk menghitung TF-IDF, sebagai berikut [9]:

$$IDFi = \log\left(\frac{D}{dfi}\right)$$

Dimana D merupakan jumlah semua dokumen sedangkan dfi adalah jumlah dokumen yang mengandung TF. Rumus TF-IDF sebagai berikut.

$$TFIDF = TFxIDF$$

Setelah itu dilakukan pembagian data latih dan data uji sebesar 80% dan 20%.

e. Evaluasi dan Hasil

Evaluasi performa adalah proses mengukur sejauh mana suatu sistem atau model berhasil atau gagal dalam mencapai tujuan tertentu. Dalam konteks *machine learning*, evaluasi performa adalah cara untuk mengukur seberapa baik model dapat melakukan tugas tertentu dalam klasifikasi. Untuk mengevaluasi performa suatu model digunakan sebuah *confusion matrix* yang berisi informasi asli dan yang diprediksi. Dari matrix tersebut dapat diketahui akurasi, presisi, dan recall, seperti pada tabel 1.

Tabel 1. *Confusion Matrix*

Kelas		**Predicted Class**		
		Positif	Negatif	Netral
Actual Class	Positif	True Positif (TP)	False Negatif (FN)	False Netral (FNet)
	Negatif	False Positif (FP)	True Negatif (TN)	False Netral (FNet)
	Netral	False Positif (FP)	False Negatif (FNet)	True Netral (TNet)

Penjelasan pada tabel 1 dapat dilihat sebagai berikut:

- *True Positif* (TP):
  Instance yang sebenarnya berada di kelas Positif dan diprediksi dengan benar sebagai kelas Positif oleh model.
- *True Negatif* (TN):
  Instance yang sebenarnya berada di kelas Negatif dan diprediksi dengan benar sebagai kelas Negatif oleh model.
- *True Netral* (TNet):
  Instance yang sebenarnya berada di kelas Netral dan diprediksi dengan benar sebagai kelas Netral oleh model.
- *False Positif* (FP):
  Instance yang sebenarnya berada di kelas Negatif atau Netral, tetapi salah diprediksi sebagai kelas Positif oleh model.

- *False Negatif* (FN):
  Instance yang sebenarnya berada di kelas Positif atau Netral, tetapi salah diprediksi sebagai kelas Negatif oleh model.
- *False Netral* (FNet):
  Instance yang sebenarnya berada di kelas Positif atau Negatif, tetapi salah diprediksi sebagai kelas Netral oleh model.

Dari *confusion matrix* dapat digunakan untuk menghitung nilai *accuracy*, *precision, recall* dan *F1*, berikut rumus dari *accuracy*.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Perhitungan *precision* untuk mengetahui perbandingan jumlah kelas positif yang diprediksi benar dibandingkan dengan jumlah semua kelas positif. Berikut umus *precision*.

$$precision = \frac{TP}{TP + FP}$$

Perhitungan *recall* dilakukkan untuk mengetahui perbandingan jumlah kelas positif yang diprediksi benar dibandinhkan dengan kelas yang benar positif. Berikut rumus *recall*.

$$recall = \frac{TP}{TP + FN}$$

Perhitungan *F-Measure* diperoleh dari *precision* dan *recall*. Berikut rumus *F1*.

$$F1 = \frac{2\ x\ precision\ x\ recall}{precision + recall}$$

3. RESULTS AND ANALYSIS

3.1. Pengumpulan Data

Pengumpulan data ini dilakukkan dengan teknik *scrapping python google play scrapper*, seperti pada gambar 2.

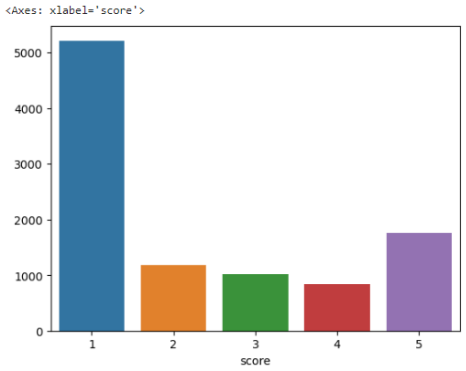
JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

	content	score	Year	Month	Day
8206	Untuk penyimpanan semakin update semakin membe...	1	2023	8	8
6717	Parahhhh...makin kesini makin parah pengaturan...	1	2023	8	8
6720	Saya pemain lama emel, makin lama emel sering ...	3	2023	8	9
6616	Tolong kembalikan performa Mobile Legends sepe...	4	2023	8	9
6615	Tolong diperbaiki sinyal dan kapasitas penyimp...	5	2023	8	10
...	...	...	...	...	...
8207	Fitur 'highlight' setelah update menghilang. D...	4	2024	1	14
1300	Ram 6 bisa lag mulu..... Padahal penyimpanan...	1	2024	1	14
8672	Game bagus tapi bosan Mau nanya Tujuan ngasih ...	1	2024	1	14
8482	Ehh min klu ngasih tim it yg waras dikit lah b...	1	2024	1	15
8995	Aqw kasik ulasan Bintang 1 aja . Game tambah l...	1	2024	1	15

10000 rows × 5 columns

Gambar 2. Pengumpulan Data

Untuk mempermudah dalam melihat nilai *score* maka dilakukan visualisasi *score*, seperti pada gambar 3.



Gambar 3. Visualisasi Score

Pada gambar 3 terlihat *score* dengan nilai 1 lebih banyak dibandingkan yang lain.

3.2. Pelabelan

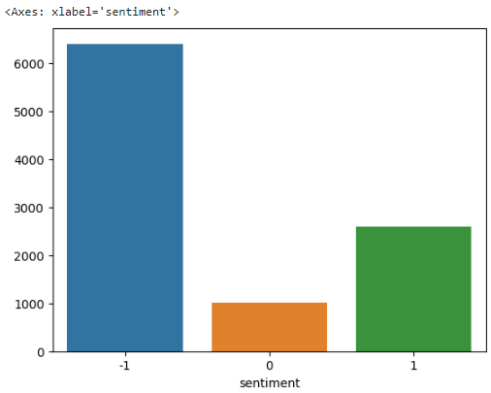
Pada proses pelabelan menambahkan kolom sentimen dengan kriteria *score* 1–2 menjadi negatif dengan kode -1, *score* 3 adalah netral dengan kode angka 0 dan *score* 4–5 adalah positif dengan kode 1, seperti pada gambar 4.

	content	score	Year	Month	Day	sentiment
8206	Untuk penyimpanan semakin update semakin membe...	1	2023	8	8	-1
6717	Parahhhh...makin kesini makin parah pengaturan...	1	2023	8	8	-1
6720	Saya pemain lama emel, makin lama emel sering ...	3	2023	8	9	0
6616	Tolong kembalikan performa Mobile Legends sepe...	4	2023	8	9	1
6615	Tolong diperbaiki sinyal dan kapasitas penyimp...	5	2023	8	10	1

Gambar 4. Pelabelan

Untuk mempermudah dalam melihat pelabelan *score* maka dilakukan visualisasi pelabelan *score*, seperti pada gambar 5.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index



Gambar 5. Visualisasi Pelabelan

Pada gambar 5 terlihat *score* dengan nilai -1 lebih banyakk dibandingkan yang lain.

3.3. Preprocessing

preprocessing data digunakan untuk membersihkan dan menyiapkan data teks sehingga dapat diolah dengan baik oleh model *machine learning*.

- *Case Folding*

Yaitu menghapus url dari kolom konten, merubah teks jadi *lower case*, menghapus mention, menghapus hashtag, menghapus next karakter, menghapus tanda baca, menghapus *extra whitespace*, seperti pada tabel 2.

Tabel 2. *Case Folding*

Dataset	Aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog, gameplay ku rusak dibuat mooton,kalau aku jago sigapapa,aku tidak jago mooton ini ada apa si ini winrate udah habis-habisan,masa gameplay pun mau dirusak,dapat yang tim yang tidak kompak terus,dapat lawan tidak kira-kira skill nya, kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim-tim yang egois,saya bukan pro player,diluar pemikiran saya kalo dapat tim yang itu-itu terus
CaseFolding	aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog gameplay ku rusak dibuat mooton kalau aku jago sigapapa aku tidak jago mooton ini ada apa si ini winrate udah habis habisan masa gameplay pun mau dirusak dapat yang tim yang tidak kompak terus dapat lawan tidak kira kira skill nya kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim tim yang egois saya bukan pro player diluar pemikiran saya kalo dapat tim yang itu itu terus

- *Tokenize*

Yaitu proses memecah teks atau kalimat menjadi unit yang lebih kecil yang disebut sebagai “token”, seperti pada tabel 3.

Tabel 3. *Tokenize*

Dataset	Aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog, gameplay ku rusak dibuat mooton,kalau aku jago sigapapa,aku tidak jago mooton ini ada apa si ini winrate udah habis-habisan,masa gameplay pun mau dirusak,dapat yang tim yang

	tidak kompak terus,dapat lawan tidak kira-kira skill nya, kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim-tim yang egois,saya bukan pro player,diluar pemikiran saya kalo dapat tim yang itu-itu terus
CaseFolding	aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog gameplay ku rusak dibuat mooton kalau aku jago sigapapa aku tidak jago mooton ini ada apa si ini winrate udah habis habisan masa gameplay pun mau dirusak dapat yang tim yang tidak kompak terus dapat lawan tidak kira kira skill nya kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim tim yang egois saya bukan pro player diluar pemikiran saya kalo dapat tim yang itu itu terus
Tokenizer	['aneh', 'penyimpanan', 'lag', 'langsung', 'relog', 'gameplay', 'ku', 'rusak', 'mooton', 'jago', 'sigapapa', 'jago', 'mooton', 'si', 'winrate', 'udah', 'habis', 'habisan', 'gameplay', 'dirusak', 'tim', 'kompak', 'lawan', 'skill', 'nya', 'kaya', 'gini', 'player', 'kemajuan', 'performa', 'dibarengin', 'tim', 'tim', 'egois', 'pro', 'player', 'diluar', 'pemikiran', 'kalo', 'tim']

- *Filtering*

Yaitu membuang kata kurang penting atau *wordlist* menyimpan kata penting, seperti pada tabel 4.

Tabel 4. *Filtering*

Dataset	Aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog, gameplay ku rusak dibuat mooton,kalau aku jago sigapapa,aku tidak jago mooton ini ada apa si ini winrate udah habis-habisan,masa gameplay pun mau dirusak,dapat yang tim yang tidak kompak terus,dapat lawan tidak kira-kira skill nya, kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim-tim yang egois,saya bukan pro player,diluar pemikiran saya kalo dapat tim yang itu-itu terus
CaseFolding	aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog gameplay ku rusak dibuat mooton kalau aku jago sigapapa aku tidak jago mooton ini ada apa si ini winrate udah habis habisan masa gameplay pun mau dirusak dapat yang tim yang tidak kompak terus dapat lawan tidak kira kira skill nya kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim tim yang egois saya bukan pro player diluar pemikiran saya kalo dapat tim yang itu itu terus
Tokenizer	['aneh', 'penyimpanan', 'lag', 'langsung', 'relog', 'gameplay', 'ku', 'rusak', 'mooton', 'jago', 'sigapapa', 'jago', 'mooton', 'si', 'winrate', 'udah', 'habis', 'habisan', 'gameplay', 'dirusak', 'tim', 'kompak', 'lawan', 'skill', 'nya', 'kaya', 'gini', 'player', 'kemajuan', 'performa', 'dibarengin', 'tim', 'tim', 'egois', 'pro', 'player', 'diluar', 'pemikiran', 'kalo', 'tim']
Filtering	['aneh', 'simpan', 'lag', 'langsung', 'relog', 'gameplay', 'ku', 'rusak', 'mooton', 'jago', 'sigapapa', 'jago', 'mooton', 'si', 'winrate', 'udah', 'habis', 'habis', 'gameplay', 'rusak', 'tim', 'kompak', 'lawan', 'skill', 'nya', 'kaya', 'gin', 'player', 'maju', 'performa', 'dibarengin', 'tim', 'tim', 'egois', 'pro', 'player', 'luar', 'pikir', 'kalo', 'tim']

- *Stemming*

Yaitu teknik pemrosesan bahasa alami yang menurunkan infleksi kata ke bentuk akarnya, sehingga membantu dalam pra-pemrosesan teks, kata, dan dokumen untuk normalisasi teks, seperti pada tabel 5.

Tabel 5. *Stemming*

Dataset	Aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog, gameplay ku rusak dibuat mooton,kalau aku jago sigapapa,aku tidak jago mooton ini ada apa si ini winrate udah habis-habisan,masa gameplay pun mau dirusak,dapat yang tim yang

	tidak kompak terus,dapat lawan tidak kira-kira skill nya, kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim-tim yang egois,saya bukan pro player,diluar pemikiran saya kalo dapat tim yang itu-itu terus
CaseFolding	aneh padahal penyimpanan masih banyak masa lag sedikit langsung relog gameplay ku rusak dibuat mooton kalau aku jago sigapapa aku tidak jago mooton ini ada apa si ini winrate udah habis habisan masa gameplay pun mau dirusak dapat yang tim yang tidak kompak terus dapat lawan tidak kira kira skill nya kalau kaya gini terus kapan setiap player ada kemajuan performa kalau dibarengin sama tim tim yang egois saya bukan pro player diluar pemikiran saya kalo dapat tim yang itu itu terus
Tokenizer	['aneh', 'penyimpanan', 'lag', 'langsung', 'relog', 'gameplay', 'ku', 'rusak', 'mooton', 'jago', 'sigapapa', 'jago', 'mooton', 'si', 'winrate', 'udah', 'habis', 'habisan', 'gameplay', 'dirusak', 'tim', 'kompak', 'lawan', 'skill', 'nya', 'kaya', 'gini', 'player', 'kemajuan', 'performa', 'dibarengin', 'tim', 'tim', 'egois', 'pro', 'player', 'diluar', 'pemikiran', 'kalo', 'tim']
Filtering	['aneh', 'simpan', 'lag', 'langsung', 'relog', 'gameplay', 'ku', 'rusak', 'mooton', 'jago', 'sigapapa', 'jago', 'mooton', 'si', 'winrate', 'udah', 'habis', 'habis', 'gameplay', 'rusak', 'tim', 'kompak', 'lawan', 'skill', 'nya', 'kaya', 'gin', 'player', 'maju', 'performa', 'dibarengin', 'tim', 'tim', 'egois', 'pro', 'player', 'luar', 'pikir', 'kalo', 'tim']
Stemming	aneh simpan langsung relog gameplay rusak mooton jago sigapapa jago mooton winrate udah habis habis gameplay rusak kompak lawan skill kaya player maju performa dibarengin egois player luar pikir kalo

3.4 Analisis Sentimen

Pada analisis sentimen dilakukan dengan 5 algoritma yaitu algoritma *Naïve Bayes*, *algoritma Support Vectore Machine, Random Forest, Decision Tree,* dan *Logistic Regression.* Sebelum dilakukkan analisis, terlebih dahulu melihat ulasan positif, negative, dan netral dengan *wordcloud*, seperti pada gambar 6,7,8.

Gambar 6. *Wordcloud Positif*

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

Gambar 7. *Wordcloud Negatif*

Gambar 8. *Wordcloud Netral*

Setelah melihat *wordcloud* positif, negative dan netral. Selanjutnya mengubah nilai menjadi TF-IDF. *Term frequency-inverse document frequency* adalah *text vectorizer* yang mengubah teks menjadi vektor yang dapat digunakan. Ini menggabungkan 2 konsep, *Term Frequency* (TF) dan *Document Frequency* (DF). Selanjutnya dalam sentiment analysis ini tahapannya adalah split data menjadi data training dan data testing sebesar 80% dan 20%, seperti pada tabel 6.

Tabel 6. Split Data

	Data Pelatihan	Pengujian Data	Total
Jumlah Data	15952	3989	19941

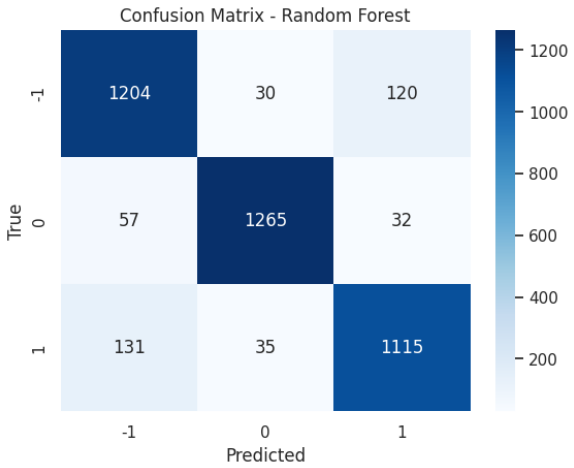
3.5 Evaluasi dan Hasil

Evaluasi performa adalah proses mengukur sejauh mana suatu sistem atau model berhasil atau gagal dalam mencapai tujuan tertentu. Evaluasi hasil dapat dilihat pada gambar berikut:

a. *Random Forest*

Sebelum dilakukan perhitungan *accuracy*, terlebih dahulu menghitung *confusion mat*rix, seperti pada gambar 9.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index



Gambar 9. CM *Random Forest*

Dari hasil data pada gambar 9, data dengan jumlah 3989 ulasan pengguna diperoleh 1115 ulasan dinyatakan benar positif, 1204 komentar dinyatakan benar negatif, dan 1265 komentar dinyatakan benar netral. Selanjutnya dilakukkan perhitungan *classification report* yang meliputi *Precision score*, *recall score*, *f1-score* dan nilai *accuracy*, seperti pada gambar 10.

Classification Report:

	precision	recall	f1-score	support
-1	0.76	0.96	0.85	1294
0	0.92	0.90	0.91	1289
1	0.94	0.72	0.81	1256
accuracy			0.86	3839
macro avg	0.87	0.86	0.86	3839
weighted avg	0.87	0.86	0.86	3839

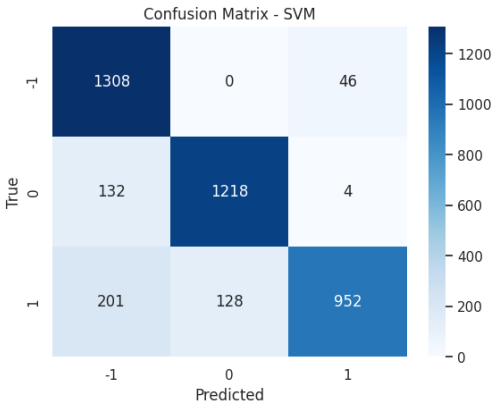
Gambar 10. CR *Random Forest*

Pada gambar 10 menghasilkan nilai accuracy sebesar 86% dengan waktu komputasi 28.98 detik yang terdiri dari waktu waktu training 28.78 detik dan waktu prediksi 0.20 detik.

b. *Support Vector Machine*

Sebelum dilakukan perhitungan *accuracy*, terlebih dahulu menghitung *confusion matrix*, seperti pada gambar 11.

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index



Gambar 11. CM *Support Vector Machine*

Dari hasil data pada gambar 11, data dengan jumlah 3989 ulasan pengguna diperoleh 952 ulasan dinyatakan benar positif, 1308 komentar dinyatakan benar negatif, dan 1218 komentar dinyatakan benar netral. Selanjutnya dilakukkan perhitungan c*lassification report* yang meliputi *Precision score*, *recall score*, *f1-score* dan nilai *accuracy*, seperti pada gambar 12.

Classification Report:

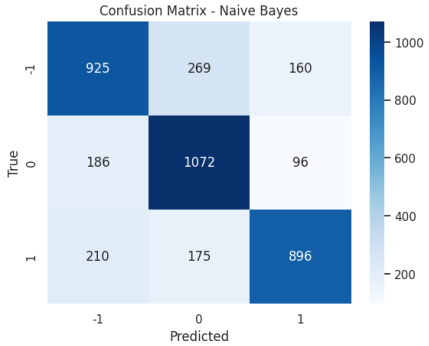
	precision	recall	f1-score	support
-1	0.80	0.97	0.87	1354
0	0.90	0.90	0.90	1354
1	0.95	0.74	0.83	1281
accuracy			0.87	3989
macro avg	0.88	0.87	0.87	3989
weighted avg	0.88	0.87	0.87	3989

Gambar 12. CR *Support Vector Machine*

Pada gambar 12 menghasilkan nilai *accuracy* sebesar 87% dengan waktu komputasi 60.08 detik yang terdiri dari waktu waktu training 48.62 detik dan waktu prediksi 11.46 detik.

c. *Naïve Bayes*

Sebelum dilakukan perhitungan *accuracy*, terlebih dahulu menghitung *confusion matrix*, seperti pada gambar 13.



Gambar 13. *CM Naïve Bayes*

JSI : Jurnal Sistem Informasi *(E-Journal), VOL. 16, NO. 1, April 2024*
ISSN Print : 2085-1588
ISSN Online :2355-4614
LINK: https://jsi.ejournal.unsri.ac.id/index.php/jsi/index

Dari hasil data pada gambar 12, data dengan jumlah 3989 ulasan pengguna diperoleh 896 ulasan dinyatakan benar positif, 925 komentar dinyatakan benar negatif, dan 1072 komentar dinyatakan benar netral. Selanjutnya dilakukkan perhitungan *classification report* yang meliputi *Precision score, recall score, f1-score* dan nilai *accuracy*, seperti pada gambar 13.

Classification Report:

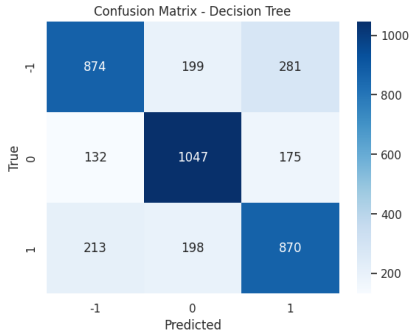
	precision	recall	f1-score	support
-1	0.70	0.68	0.69	1354
0	0.71	0.79	0.75	1354
1	0.78	0.70	0.74	1281
accuracy			0.73	3989
macro avg	0.73	0.72	0.73	3989
weighted avg	0.73	0.73	0.72	3989

Gambar 13. CR *Naïve Bayes*

Pada gambar 13 menghasilkan nilai *accuracy* sebesar 73% dengan waktu komputasi 0.01 detik yang terdiri dari waktu waktu training 0.01 detik dan waktu prediksi 0.0 detik.

d. *Decision Tree*

Sebelum dilakukan perhitungan *accuracy*, terlebih dahulu menghitung *confusion matrix*, seperti pada gambar 14.



Gambar 14. CM *Decision Tree*

Dari hasil data pada gambar 14, data dengan jumlah 3989 ulasan pengguna diperoleh 870 ulasan dinyatakan benar positif, 874 komentar dinyatakan benar negatif, dan 1047 komentar dinyatakan benar netral. Selanjutnya dilakukkan perhitungan *classification report* yang meliputi *Precision score, recall score, f1-score* dan nilai *accuracy*, seperti pada gambar 15.

Classification Report:

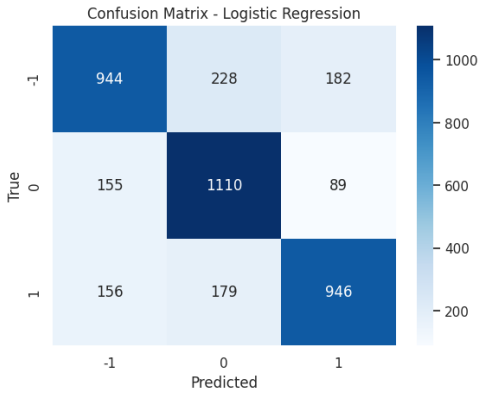
	precision	recall	f1-score	support
-1	0.72	0.65	0.69	1354
0	0.73	0.77	0.75	1354
1	0.65	0.68	0.67	1281
accuracy			0.70	3989
macro avg	0.70	0.70	0.70	3989
weighted avg	0.70	0.70	0.70	3989

Gambar 15. CR *Decision Tree*

Pada gambar 15 menghasilkan nilai accuracy sebesar 70% dengan waktu komputasi 4.39 detik yang terdiri dari waktu waktu training 4.39 detik dan waktu prediksi 0.0 detik.

e. *Logistic Regression*

Sebelum dilakukan perhitungan *accuracy*, terlebih dahulu menghitung *confusion matrix*, seperti pada gambar 16.



Gambar 16. CM *Logistic Regression*

Dari hasil data pada gambar 16, data dengan jumlah 3989 ulasan pengguna diperoleh 946 ulasan dinyatakan benar positif, 944 komentar dinyatakan benar negatif, dan 1110 komentar dinyatakan benar netral. Selanjutnya dilakukkan perhitungan *classification Report* yang meliputi *Precision score, recall score, f1-score* dan nilai *accuracy*, seperti pada gambar 17.

Classification Report:

	precision	recall	f1-score	support
-1	0.75	0.70	0.72	1354
0	0.73	0.82	0.77	1354
1	0.78	0.74	0.76	1281
accuracy			0.75	3989
macro avg	0.75	0.75	0.75	3989
weighted avg	0.75	0.75	0.75	3989

Gambar 15. CR *Decision Tree*

Pada gambar 15 menghasilkan nilai *accuracy* sebesar 75% dengan waktu komputasi 4.51 detik yang terdiri dari waktu waktu training 4.51 detik dan waktu prediksi 0.0 detik.

4. CONCLUSION

Berdasarkan data ulasan pengguna yang diperoleh dari hasil s*crapping google playstore* aplikasi Mobile Legend kecenderungan ulasan yang disampaikan mengandung komentar netral. Pengguna Mobile Legend bersikap netral terhadap aplikasi ini. Dari 3989 data pengujian, terdapat 1265 komentar dengan sentimen netral, kemudian 1115 komentar dengan sentimen positif, dan 1204 komentar dengan sentimen negatif. Di sini dapat disimpulkan bahwa pengguna Mobile Legend di Indonesia bersikap netral terhadap aplikasi. Berdasarkan hasil akurasi algoritma *Support Vectore Machine* sebesar 87%,

lebih tinggi dibandingkan akurasi Algoritma *Random Forest, Naïve Bayes, Decision Tree* dan Algoritma *Logistic Regression* maka dapat disimpulkan bahwa algoritma *Support Vector Machine* sangat baik digunakan untuk melakukan analisis sentiment ulasan pengguna aplikasi Mobile Legend.

Saran penelitian ini adalah mencoba menggunakan perbandingan algoritma lain untuk mengahsilkan performa akurasi yang lebih baik dalam menganalisis sentiment ulasan pengguna seperti algoritma KNN, BERT (*Bidirectional Encoder Representations from Transformers*) dan LSTM (*Long Short Term Memory*)

REFERENCES

[1] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi," *Jurnal Teknoinfo*, vol. 14, no. 2, p. 115, 2020, doi: 10.33365/jti.v14i2.679.

[2] A. D. Adhi Putra, "Analisis Sentimen pada Ulasan pengguna Aplikasi Bibit Dan Bareksa dengan Algoritma KNN," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 2, pp. 636–646, 2021, doi: 10.35957/jatisi.v8i2.962.

[3] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *Jurnal Informatika*, vol. 8, no. 2, pp. 200–207, 2021, doi: 10.31294/ji.v8i2.9681.

[4] G. Nugroho, D. Murdiansyah, and K. Lhaksmana, "Analisis Sentimen Pemilihan Presiden Amerika 2020 di Twitter Menggunakan Naïve Bayes dan Support Vector Machine," *e-Proceeding of Engineering*, vol. 8, no. 5, pp. 10106–10115, 2021.

[5] Suripto, Rr. N. Rahmanita, and A. S. Kirana, "Teknik Pre-processing dan Classification dalam Data Science – Master of Industrial Enginering." 2022.

[6] M. H. Widianto, "Algoritma Naive Bayes | BINUS UNIVERSITY BANDUNG - Kampus Teknologi Kreatif," *Binus.Ac.Id*. 2019.

[7] Samsudiney, "Penjelasan Sederhana tentang Apa Itu SVM? | by Samsudiney | Medium," *Https://Medium.Com/*. pp. 2–7, 2019.

[8] Trivusi, "Algoritma Random Forest_ Pengertian dan Kegunaannya," *Trivusi*. p. 1, 2022.

[9] Delta Sierra, "Algoritma TF — IDF," *The Medium Blog*. p. https://dltsierra.medium.com/algoritma-tf-idf-633e, 2019.