

IMPLEMENTASI METODE KLASIFIKASI NAÏVE BAYES PADA ANALISIS SENTIMEN ULASAN APLIKASI *MOBILE LEGENDS* DI *GOOGLE PLAY STORE*

Hajjul Ikram¹, M. Aviciena Hasibuan², Herdi Rianda³, Angga Fahriza⁴, Suryadi⁵, Rivansyah Suhendra⁶

¹*Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Teuku Umar, Aceh Barat 23681, Indonesia*

ABSTRAK

Penelitian ini menerapkan metode klasifikasi *Naïve Bayes* untuk menganalisis sentimen ulasan pengguna terhadap aplikasi *Mobile Legends* di *Google Play Store*. Data ulasan dikumpulkan melalui teknik web scraping, lalu dilabeli dan diproses melalui tahap *preprocessing*, termasuk *case folding*, penghapusan *stopwords*, tokenisasi, dan *stemming*. Data dibagi menjadi data pelatihan dan pengujian, kemudian fitur-fitur diekstraksi menggunakan metode TF-IDF. Hasil penelitian menunjukkan bahwa model *Naïve Bayes* mencapai tingkat akurasi sebesar 85% dalam mengklasifikasikan sentimen ulasan. Evaluasi melibatkan presisi, *recall*, dan *F1-score*. Temuan penelitian ini memberikan wawasan yang mendalam tentang pandangan pengguna terhadap *Mobile Legends* dan memiliki potensi aplikasi dalam pemantauan sentimen serta pengembangan aplikasi mobile dan analisis sentimen berbasis teks di berbagai domain.

Kata kunci: *Analisis Sentimen, Text Mining, Naïve Bayes, Mobile Legend, Google Play Store*

PENDAHULUAN

Dalam era digital yang semakin maju, aplikasi *mobile* telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari (Guzsvinecz & Szűcs, 2023). Salah satu aplikasi mobile yang sangat populer dan berhasil menarik perhatian jutaan pengguna di seluruh dunia adalah *Mobile Legends*. Sebagai salah satu permainan daring (online) terbesar di dunia, *Mobile Legends* telah mencapai prestasi yang luar biasa dengan jumlah unduhan yang fantastis di *platform Google Play Store* (Ardianto *at al.*, 2020).

Namun, dengan popularitas yang tinggi, muncul pula kebutuhan untuk memahami persepsi dan tanggapan pengguna terhadap aplikasi ini. Seiring dengan peningkatan jumlah pengguna, ulasan dan pendapat pengguna di *Google*

Play Store menjadi sumber informasi yang berharga untuk mengevaluasi kualitas dan kinerja aplikasi *Mobile Legends*. Oleh karena itu, analisis sentimen ulasan pengguna dapat memberikan wawasan yang berharga untuk pengembang dan pemilik aplikasi dalam memahami kekuatan dan kelemahan aplikasi mereka (Kusnanda & Permana, 2023; Prasetya & Ependi, 2022).

Penelitian ini bertujuan adalah untuk melakukan analisis sentimen terhadap ulasan aplikasi *Mobile Legends* di *Google Play Store* menggunakan metode *Naïve Bayes*. Metode ini telah terbukti efektif dalam analisis sentimen dalam berbagai konteks dan dapat memberikan pemahaman yang mendalam tentang opininya pengguna. Dengan memanfaatkan teknik ini, kami berharap

dapat mengidentifikasi dan mengklasifikasikan ulasan pengguna *Mobile Legends* ke dalam kategori positif atau negatif, sehingga memberikan pemahaman yang lebih baik tentang preferensi dan kebutuhan pengguna (Alamo *et al.*, 2023; Lasamahu, 2021). Selain itu, penelitian ini juga memiliki tujuan untuk menyediakan pemahaman yang lebih mendalam tentang apa yang membuat pengguna merasa puas atau tidak puas dengan aplikasi *Mobile Legends*. Dengan memahami faktor-faktor ini, pengembang dapat melakukan perbaikan dan peningkatan yang lebih baik pada fitur-fitur dan pengalaman pengguna dalam upaya meningkatkan kepuasan pengguna secara keseluruhan (Ardianto *et al.*, 2020; Utami *et al.*, 2023).

Melalui analisis sentimen ulasan aplikasi *Mobile Legends* menggunakan metode *Naïve Bayes*, diharapkan penelitian ini dapat memberikan kontribusi yang signifikan dalam bidang teks mining terkhusus pada analisis sentimen dan memberikan wawasan berharga bagi pengembang aplikasi *Mobile Legends* (Punetha & Jain, 2023). Dengan pemahaman yang lebih baik tentang sudut pandang dan preferensi pengguna, diharapkan pengembang dapat mengambil langkah-langkah yang tepat untuk meningkatkan aplikasi mereka, meningkatkan mutu, dan memenuhi harapan pengguna dengan lebih baik.

METODOLOGI PENELITIAN

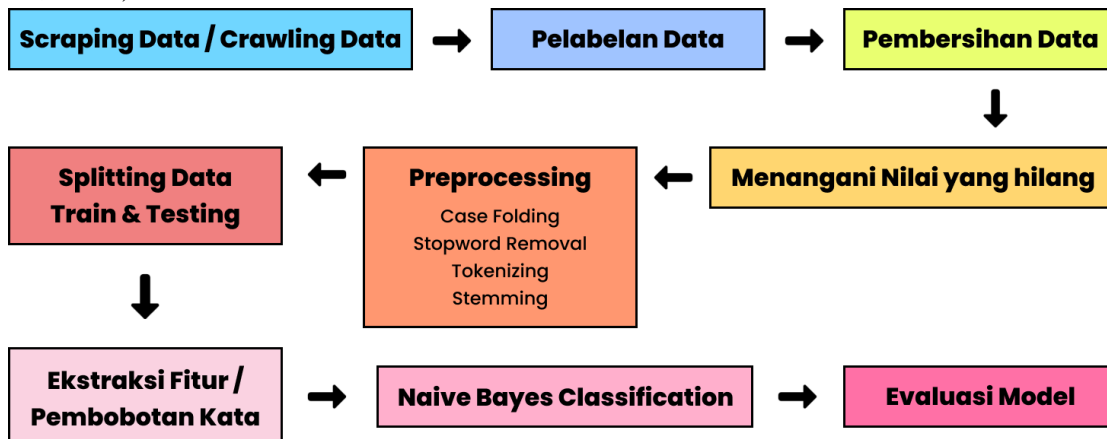
Metode yang digunakan dalam penelitian ini melibatkan serangkaian tahapan yang dirancang untuk menganalisis sentimen ulasan pengguna terhadap aplikasi *Mobile Legends* di

Google Play Store. Tahap pertama adalah pengumpulan data, di mana kami menggunakan teknik *web scraping* untuk mengambil ulasan-ulasan pengguna yang tersedia di *platform Google Play Store*. Selanjutnya, kami melabeli data tersebut sebagai positif, negatif, atau netral dalam tahap pelabelan data. Selama tahap *preprocessing*, kami melakukan beberapa langkah penting, termasuk *case folding* untuk mengubah semua huruf teks menjadi huruf kecil, penghapusan *stopwords* untuk menghilangkan kata-kata umum yang tidak relevan, tokenisasi untuk memecah teks menjadi token-token, dan *stemming* untuk mengubah kata-kata menjadi bentuk dasar mereka (Narwal & Aggarwal, 2022). Alur tahapan penelitian dapat dilihat pada Gambar 1.

Data kemudian dibagi menjadi dua subset yaitu data pelatihan dan data pengujian. Data pelatihan digunakan untuk melatih model klasifikasi *Naïve Bayes*. Selanjutnya, kami melakukan ekstraksi fitur menggunakan metode TF-IDF untuk mengubah teks ulasan menjadi representasi numerik yang dapat digunakan oleh model. Model klasifikasi *Naïve Bayes* kemudian diterapkan untuk mengklasifikasikan ulasan berdasarkan fitur-fitur yang diekstraksi, menghasilkan prediksi sentimen (Britto & Pacifico, 2020).

Tahap akhir penelitian adalah evaluasi hasil klasifikasi. Kami menggunakan berbagai metrik evaluasi seperti akurasi, presisi, recall, dan F1-score untuk mengevaluasi kinerja model (Han *et al.*, 2012). Metode yang kami terapkan dalam penelitian ini memberikan wawasan berharga tentang pandangan pengguna terhadap aplikasi

tersebut di *Google Play Store* (Musat et al., 2012).



Gambar 1. Tahapan Metodologi dalam Penelitian

HASIL DAN PEMBAHASAN

Web scraping sebagai proses mengambil data dari halaman web secara otomatis menggunakan program komputer, dengan tujuan mendapatkan informasi yang dapat dianalisis. Untuk pengambilan data kami menggunakan Teknik *scraping* untuk menghemat waktu pengerjaan program.

Library yang kami gunakan untuk

melakukan proses ini adalah *Google Play Scraper*, dengan menggunakan *Google Play Scraper*, pengguna dapat mengambil informasi seperti nama aplikasi, deskripsi, peringkat, ulasan pengguna, jumlah unduhan, kategori, harga, dan fitur lainnya dari aplikasi yang tersedia di *Google Play Store*. Hasil *scraping* data dapat dilihat pada gambar 2.

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	replyedAt	appVersion
0	1c5caec0-1ad4-46ba-bf50-66617c37b2fb	Hendra Mulya Sanjaya	https://play-ih.googleusercontent.com/a-AD_cM...	Tolong lah moonton, ini game kok habis update ...	1	13534	1.7.82.8551	2023-05-22 09:17:15	None	NaT	1.7.82.8551
1	70e90cef-914-4c80-49a1-b725929e701a	Dama Putra	https://play-ih.googleusercontent.com/a-AD_cM...	Game ini menarik, karena game ini bisa mengajai...	2	10235	1.7.82.8551	2023-05-18 10:08:48	Dear Hero inTerima kasih atas tanggapan Anda ...	2023-05-20 03:30:25	1.7.82.8551
2	d0651960-3b73-4da9-991a-1ea44c7f0d9e	Atef Fishing Vlog	https://play-ih.googleusercontent.com/aAAcHTL...	Untuk developer mohon perbaiki ... Baru update...	2	802	1.7.82.8551	2023-06-03 14:56:10	None	NaT	1.7.82.8551
3	8e36ff4c-39e-4264-b20f-37b6613b535a	Fawwaz Zakky	https://play-ih.googleusercontent.com/a-AD_cM...	Setelah update bukannya ada peningkatan malah ...	1	4053	1.7.82.8551	2023-05-30 15:13:47	None	NaT	1.7.82.8551
4	7670e45-dab4-40a2-aea5-c3e57044005	Yogi Siswoyo	https://play-ih.googleusercontent.com/a-AD_cM...	Gak tau kenapa mli kenapa jadi begini Stehlah up...	1	70	1.7.82.8551	2023-06-03 19:22:29	None	NaT	1.7.82.8551

Gambar 2. Hasil *Scraping* Data

Scraping kemudian dilakukan, pada program ini kami meminta sejumlah 1000 data berdasarkan reviews pada aplikasi *Mobile Legends* yang tersedia di *Google Play Store*. Berikut data teratas dari hasil tahapan *scraping* data. Data hasil *scraping* juga dapat di modifikasi sesuai keinginan, bisa diurutkan berdasarkan kolom yang diinginkan atau hanya menampilkan kolom yang dirasa

perlu. Semua dapat dilakukan ketika data sudah berhasil di *scraping*.

Hasil Pelabelan Data

Tahapan selanjutnya setelah pengumpulan data adalah pelabelan data. Pelabelan data pada *text mining* adalah proses memberikan klasifikasi atau label pada teks yang mencerminkan opini atau sentimen pengguna dalam bahasa

Indonesia, seperti positif, negatif, atau netral. Proses pelabelan data pada penelitian ini untuk memberikan nilai positif atau negatif berdasarkan score/bintang yang didapatkan dari penilaian user. Pada kasus ini, kami memberikan label none pada penilain dengan score 3 dikarenakan nilai dapat berupa tanggapan positif dan negatif.

Menangani *Missing Value*

Menangani *missing value* adalah proses atau langkah-langkah untuk mengatasi atau menangani nilai yang hilang atau tidak tersedia dalam dataset. Missing value dapat terjadi karena berbagai alasan, seperti kegagalan dalam

pengumpulan data, data yang tidak lengkap, atau data yang hilang karena kesalahan teknis. Sangat penting untuk menangani missing value karena dapat mempengaruhi analisis data dan menghasilkan hasil yang tidak akurat atau bias. Hasil penanganan missing value dapat dilihat pada gambar 4.

Kami juga menerapkan tahapan ini pada program kami untuk meningkatkan akurasi data yang kami miliki. Kami memulai dari tahapan pengecekan, menghitung jumlah yang hilang, meminta ringkasan statistik dataframe, dan menghapus nilai yang hilang pada kolom "Label". Gambar 3 menampilkan contoh hasil pelabelan data.

```
def pelabelan(score):
    if score < 3:
        return 'Negatif'
    elif score == 4 :
        return 'Positif'
    elif score == 5 :
        return 'Positif'
my_df['Label'] = my_df ['score'].apply(pelabelan)
my_df.head(20)
```

	content	score	Label
125	MOBILE LEGENDS Sangat seru dan bagus. tapi mas...	5	Positif
745	Semakin ke sini semakin aneh sistem pembagian ...	1	Negatif
821	Game aneh, pingnya selalu tinggi padahal game ...	1	Negatif
679	moonton tolong kembalikan assasin ke tempatnya...	5	Positif
928	Game yang bikin akal sehat sirna. Stress gara ...	1	Negatif
880	Karna ping yg gk jelas. D game lain d yt, apk ...	1	Negatif
892	Game sudah bagus, bahkan grafik juga sangat ba...	3	None

Gambar 3. Pelabelan Data

```
my_df.isnull().sum() #sum() untuk menghitung jumlah nilai yang hilang dalam setiap kolom
content      0
score        0
Label       117
dtype: int64

my_df.dropna(subset=['Label'],inplace = True)
#menghapus baris atau entri yang memiliki nilai yang hilang (NaN) dalam kolom 'Label' dari DataFrame my_
#subset=['Label'], kita menentukan bahwa hanya kolom 'Label' yang akan diperiksa untuk nilai yang hilang
#inplace=True digunakan untuk mengubah DataFrame my_df secara langsung, yaitu menghapus baris dengan nil

my_df.isnull().sum() #lihat outputnya udh hilang data NaN
content      0
score        0
Label        0
dtype: int64

[43] my_df.head(5)
```

	content	score	Label	text_clean
0	Why moonton? Tiap main selalu ping tiba² merah...	1	Negatif	why moonton tiap main selalu ping tiba merah p...
1	Moonton gbik, jaringan tidak stabil setelah up...	1	Negatif	moonton gbik jaringan tidak stabil seteah upd...
2	Moonton baik ya, contoh nya ML sekarang full H...	5	Positif	moonton baik ya contoh nya ml sekarang full hd... r
3	Game stresss, dari sistem nya kek tlo! masak d...	1	Negatif	game stresss dari sistem nya kek tlo! masak di...
4	Game bikin emosi minimal kalau ngasih tim itu ...	2	Negatif	game bikin emosi minimal kalau ngasih tim itu ...

Gambar 4. Penanganan *Missing Value*

Preprocessing

Preprocessing adalah serangkaian langkah yang diperlukan untuk mempersiapkan data mentah sebelum diproses lebih lanjut. Ini mencakup tokenisasi, normalisasi, penghapusan *stopwords*, stemming, dan lain-lain. Berikut adalah tahapan preprocessing yang terlibat dalam program kami:

1. Case Folding

Proses *case folding* adalah proses mengubah seluruh huruf menjadi huruf kecil. Pada proses ini karakter-karakter 'A'-'Z' yang terdapat pada data diubah kedalam karakter 'a'-'z'. Lihat pada gambar 7 untuk output yang kami dapatkan dari proses *Case Folding*.

2. Stopword Removal

Stopword removal adalah proses menghapus kata-kata yang umum dan tidak memberikan banyak informasi penting dalam analisis teks. Kata-kata tersebut disebut "*stop words*". *Stop words* biasanya termasuk kata-kata seperti "dan", "atau", "dari", "yang", "di", dan sejenisnya. Tujuan dari *stopword removal* adalah untuk meningkatkan efisiensi dan akurasi analisis teks dengan menghilangkan kata-kata yang sering muncul namun memiliki sedikit kontribusi dalam pemahaman konten atau makna teks.

3. Tokenizing

Tokenizing atau disebut juga tahap *Lexical Analysis* adalah proses pemotongan teks menjadi bagian-bagian yang lebih kecil, yang disebut token. Pada proses ini juga dilakukan penghilangan angka, tanda baca dan karakter lain yang dianggap tidak memiliki pengaruh terhadap pemrosesan teks. Pada proses ini kami menggunakan library NLTK, NLTK (*Natural Language Toolkit*) adalah sebuah library atau toolkit yang digunakan untuk pemrosesan bahasa alami (*natural language processing/NLP*) dalam bahasa pemrograman Python.

4. Stemming

Stemming melibatkan pemetaan dan penguraian kata ke bentuk kata dasarnya. Untuk melakukan stemming bahasa Indonesia, kita telah menyiapkan library Sastrawi di Python. Library Sastrawi menerapkan Algoritma Nazief dan Adriani yang berguna dalam proses stemming bahasa Indonesia.

Berikut tampilan proses melakukan stemming pada dataset yang kita miliki, proses ini akan memerlukan waktu untuk menampilkan *output* dari program yang dijalankan dikarenakan data yang diberikan akan sangat banyak. Hasil *stemming* terdapat pada gambar 5.

```

#-----STEMMING-----
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
#import swifter

# create Stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}
hitung=0

for document in data_clean['text_tokens']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = 1

print(len(term_dict))
print("-----")
for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    hitung+=1
    print(hitung,":",term,":",term_dict[term])

print(term_dict)
print("-----")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

#script ini bisa dipisah dari eksekusinya setelah pembacaan term selesai
data_clean['text_stemindo'] = data_clean['text_tokens'].apply(lambda x: ' '.join(get_stemmed_term(x)))
data_clean.head(20)

```

Gambar 5. Hasil Stemming

Hasil *Splitting Data Train & Testing*

Splitting dataset adalah proses membagi dataset menjadi dua *subset* yang berbeda, yaitu subset pelatihan (*training set*) dan subset pengujian (*testing set*). Tujuan dari *splitting dataset* adalah untuk melatih model pada subset pelatihan dan menguji performanya pada subset pengujian. Dengan memisahkan subset pengujian yang independen, kita dapat menguji kemampuan model untuk menggeneralisasi pada data yang belum pernah dilihat sebelumnya. Proses *splitting dataset* dapat dilakukan menggunakan berbagai metode dan *library* dalam pemrograman, tergantung pada bahasa pemrograman dan *framework* yang digunakan.

Hasil Ekstraksi Fitur (TF-IDF)

Ekstraksi fitur adalah cara mendapatkan sebuah informasi dalam suatu pola untuk membantu tugas

klasifikasi menjadi lebih mudah. Ekstraksi fitur merupakan proses mengubah data teks menjadi representasi numerik yang dapat digunakan oleh model pembelajaran mesin. Dalam konteks pemrosesan bahasa alami (*natural language processing/NLP*), ekstraksi fitur bertujuan untuk mengonversi teks menjadi vektor numerik yang dapat digunakan sebagai input untuk model pembelajaran *machine learning*.

Salah satu metode ekstraksi fitur yang populer dalam NLP adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF adalah metode yang menggabungkan dua konsep, yaitu frekuensi kata dalam dokumen (*Term Frequency/TF*) dan kebalikan frekuensi dokumen yang mengandung kata tersebut (*Inverse Document Frequency/IDF*). Gambar 6 adalah proses penerapan ekstraksi fitur.

```

[34] from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)

[35] print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

(706,)
(706,)
(177,)
(177,)

[36] from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()
vectorizer.fit(X_train)

CountVectorizer
CountVectorizer()

[37] X_train = vectorizer.transform(X_train)
X_test = vectorizer.transform(X_test)

[38] from sklearn.naive_bayes import MultinomialNB

nb = MultinomialNB()
nb.fit(tfidf_train, y_train)

MultinomialNB
MultinomialNB()
    
```

Gambar 6. Ekstraksi Fitur TF-IDF

Hasil Model Naïve Bayes Classification

Klasifikasi Bayesian merupakan proses klasifikasi statistik yang digunakan dalam memprediksi probabilitas keanggotaan suatu kelas berdasarkan *teorema teorema bayes* (Kusnanda & Permana, 2023). *Naïve Bayes* digunakan untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari "master" tabel keputusan (Chan *et al.*, 2020).

Pada program, kami mengimplementasikan MultinomialNB sebagai algoritma klasifikasinya. MultinomialNB (*Multinomial Naive Bayes*) adalah sebuah algoritma klasifikasi yang digunakan dalam pemrosesan bahasa alami (NLP) dan machine learning. Algoritma ini merupakan variasi dari *Naive Bayes* yang dikhususkan untuk data yang memiliki distribusi multinomial, seperti data yang dihasilkan dari representasi kata-kata atau kata-kata yang muncul dalam teks.

Dalam klasifikasi menggunakan MultinomialNB, setiap dokumen diwakili oleh vektor fitur yang berisi frekuensi kemunculan kata-kata dalam dokumen tersebut. Algoritma ini bekerja dengan menghitung probabilitas kemunculan kata-kata dalam setiap kelas

yang ada, dan kemudian menggunakan Teorema Bayes untuk menghitung probabilitas posterior kelas untuk setiap dokumen.

Hasil Evaluasi Model

Ketika menggunakan klasifikasi naïve bayes diperlukan tahapan evaluasi model untuk melihat hasil dari sebuah program ataupun dataset yang diteliti. Evaluasi model dilakukan dengan menggunakan instrumen *Confusion Matrix*. Tujuan penggunaan evaluasi model adalah untuk memahami sejauh mana model tersebut dapat melakukan tugas-tugas yang ditentukan dengan akurasi dan efisiensi yang memadai. Evaluasi model melibatkan penggunaan metrik evaluasi yang relevan, seperti akurasi, presisi, recall, dan F1-score.

Evaluasi model dilakukan dengan menguji model pada dataset yang telah ditentukan sebelumnya, baik dengan menggunakan dataset pengujian terpisah atau dengan menggunakan metode validasi silang. Dengan melakukan evaluasi model secara cermat, kita dapat menentukan sejauh mana model yang dibangun sudah memadai untuk tugas yang diberikan atau perlu ditingkatkan. Berdasarkan gambar 7 didapatkan hasil bahwasannya dari 1000 datatrain pada dataset yang diteliti, memiliki akurasi sebesar 84%, Presisi dengan nilai 85%, kemudian *recall* 96% dan juga *F1 score* 90%.

```

MultinomialNB Accuracy: 0.8457142857142858
MultinomialNB Precision: 0.8571428571428571
MultinomialNB Recall: 0.9635036496350365
MultinomialNB f1_score: 0.9072164948453608
confusion_matrix:
[[132  5]
 [ 22 16]]
=====

```

	precision	recall	f1-score	support
Negatif	0.86	0.96	0.91	137
Positif	0.76	0.42	0.54	38
accuracy			0.85	175
macro avg	0.81	0.69	0.72	175
weighted avg	0.84	0.85	0.83	175

Gambar 7. Hasil Evaluasi Model

Dari penelitian tersebut dapat disimpulkan bahwa metode *Naïve Bayes Classifier* dengan pembobotan tf-idf serta dengan algoritma MultinomialNB memiliki nilai akurasi tertinggi. Dengan menggunakan pembobotan tf-idf dan algoritma MultinomialNB dapat meningkatkan akurasi. Dari hasil tersebut dapat dinyatakan bahwa hipotesis terpenuhi.

KESIMPULAN

Penelitian ini berhasil mengimplementasikan metode klasifikasi *Naïve Bayes* dalam menganalisis sentimen ulasan pengguna terhadap aplikasi Mobile Legends di *Google Play Store*, dengan tingkat akurasi mencapai 85%. Hasil evaluasi menggunakan metrik presisi, recall, dan F1-score juga menggambarkan kinerja model dengan baik. Penelitian ini memberikan pemahaman yang dalam tentang pandangan pengguna terhadap aplikasi tersebut dan dapat menjadi alat efektif dalam pemantauan sentimen serta pengembangan lebih lanjut dalam konteks aplikasi mobile dan analisis sentimen berbasis teks. Penelitian ini memiliki beberapa batasan, termasuk keterbatasan dalam jumlah data yang dikumpulkan dan keakuratan pelabelan sentimen ulasan. Selain itu, fokus

penelitian terbatas pada aplikasi *Mobile Legends*, sehingga hasilnya mungkin tidak dapat langsung diterapkan pada aplikasi lain. Kemungkinan adanya perubahan dalam ulasan pengguna dari waktu ke waktu juga tidak dapat dihindari.

UCAPAN TERIMAKASIH

Ucapan terimakasih kami ucapkan kepada Ketua Laboratorium Program Studi Teknologi Informasi yang telah memberikan izin dalam penggunaan fasilitas untuk mengerjakan penelitian ini.

DAFTAR PUSTAKA

Alamo, E., Benosa, J.-A., Fernandez, C., & Paderan, M. P. (2023). Linguistic Swearing of Mobile Legend Players: A Pragmatic Study. *Journal of English Education and Linguistics*, 4(1), 43–56.

Ardianto, R., Rivanie, T., Alkhalifi, Y., Nugraha, F. S., & Gata, W. (2020). Sentiment analysis on E-sports for education curriculum using naive Bayes and support vector machine. *Jurnal Ilmu Komputer Dan Informasi*, 13(2), 109–122.

- Britto, L. F. S., & Pacifico, L. D. S. (2020). Evaluating Video Game Acceptance in Game Reviews using Sentiment Analysis Techniques. *Proceedings of the SBGames, Virtual*, 7–10.
- Chan, A. S., Fachrizal, F., & Lubis, A. R. (2020). Outcome prediction using Naïve Bayes algorithm in the selection of role hero mobile legend. *Journal of Physics: Conference Series*, 1566(1), 12041. IOP Publishing.
- Guzsvinecz, T., & Szűcs, J. (2023). Length and sentiment analysis of reviews about top-level video game genres on the steam platform. *Computers in Human Behavior*, 107955.
- Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*. <https://doi.org/10.1016/C2009-0-61819-5>
- Kusnanda, D., & Permana, A. (2023). Implementation of Naive Bayes Classifier (NBC) for Sentiment Analysis on Twitter in Mobile Legends. *International Journal of Science, Technology & Management*, 4(5), 1132–1138.
- Lasamahu, P. F. (2021). Analisis Sentimen Pengguna Game Online Mobile Legends dengan Naïve Bayes.
- Musat, C. C., Ghasemi, A., & Faltings, B. (2012). Sentiment analysis using a novel human computation game. *Proceedings of the 3rd Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources and Their Applications to NLP*, (CONF), 1–9. Association for Computational Linguistics.
- Narwal, R., & Aggarwal, H. (2022). Predicting online game-addicted behaviour with sentiment analysis using twitter data. *Machine Learning, Advances in Computing, Renewable Energy and Communication: Proceedings of MARC 2020*, 505–517. Springer.
- Prasetya, M. O., & Ependi, U. (2022). Sentimen Analisis Hero Mobile Legends Dengan Algoritma Naive Bayes. *Jurnal Mantik*, 6(3), 3226–3234.
- Punetha, N., & Jain, G. (2023). Bayesian game model based unsupervised sentiment analysis of product reviews. *Expert Systems with Applications*, 214, 119128.
- Utami, M. A. A. T., Silvianti, P., & Masjkur, M. (2023). Algoritme Support Vector Machine untuk Analisis Sentimen Berbasis Aspek Ulasan Game Online Mobile Legends: Bang-Bang. *Xplore: Journal of Statistics*, 12(1), 63–77.