

Analisis Sentimen Pengguna Terhadap Ulasan Aplikasi Duolingo Menggunakan Metode *Logistic Regression*

Made Dinda Radityaswari ^{a1}, Luh Gede Astuti ^{a2}

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹ radityaswari.2308561072@student.unud.ac.id
² lg.astuti@unud.ac.id

Abstract

The ability to speak foreign languages, especially English, has become an important skill in the era of globalization and digitalization. However, according to the EF English Proficiency Index 2024, Indonesia ranks 80th out of 116 countries. One of the widely used solutions is the Duolingo application, a gamified language learning platform that has been downloaded over 500 million times. This research aims to analyze user sentiment toward the Duolingo application through reviews on Google Play Store using logistic regression. The data used consists of 8.648 reviews that have been labeled as positive and negative sentiment. The research process includes the stages of data preprocessing, dividing data into test and training data, weighting using TF-IDF, and classification using Logistic Regression algorithm with the parameter `class_weight='balanced'` to handle class imbalance, and evaluation using a confusion matrix. The evaluation results show that the model can achieve an accuracy of 89.83%, with a precision value of 73.91%, recall of 88.18%, and f1-score of 78.49%. This research shows that Logistic Regression with TF-IDF weighting is effective in sentiment analysis.

Keywords: Duolingo, Analysis Sentiment, Logistic Regression, TF-IDF, Confusion Matrix

1. Pendahuluan

Kemampuan berbahasa asing adalah salah satu keterampilan utama yang diperlukan di era globalisasi dan digitalisasi saat ini. Bahasa asing, terutama Bahasa Inggris menjadi hal yang penting baik itu dalam dunia pendidikan, bisnis, pariwisata, maupun dalam ilmu pengetahuan dan teknologi. Namun, pada kenyataannya tingkat kemahiran dalam berbahasa asing masih tergolong rendah di banyak negara. Berdasarkan laporan dari EF English Proficiency Index 2024, Indonesia berada di peringkat ke-80 dari 116 negara untuk kemampuan berbahasa Inggris dan termasuk ke kategori *low proficiency*[1]. Hal ini menunjukkan bahwa mayoritas masyarakat Indonesia menghadapi tantangan dalam menguasai bahasa asing. Banyak faktor yang memengaruhi hal tersebut seperti terbatasnya akses untuk pendidikan bahasa asing, metode pembelajaran yang kurang menarik, dan biaya kursus bahasa asing yang mahal. Oleh karena itu, diperlukan sebuah solusi untuk mengatasi permasalahan tersebut. Salah satunya adalah aplikasi Duolingo yang merupakan sebuah aplikasi pembelajaran bahasa asing dengan pendekatan gamifikasi yang dirilis pada tahun 2011 oleh Luis von Ahn dan Severin Hacker [2]. Hingga tanggal 19 Juni 2025 aplikasi Duolingo telah diunduh lebih dari 500 juta kali pada Google Play Store dengan lebih dari 30 juta ulasan pengguna. Dengan banyaknya data ulasan yang tersedia, penelitian ini berfokus untuk menganalisis sentimen positif dan negatif untuk memahami persepsi pengguna terhadap aplikasi Duolingo.

Berdasarkan penelitian yang dilakukan oleh Fatma Alifiana dkk dengan judul “Analisis Sentimen Aplikasi Duolingo Menggunakan Algoritma Naive Bayes dan Support Machine Learning” yang bertujuan untuk membandingkan algoritma *Naive Bayes* dan *Support Vector Machine* untuk mengklasifikasikan data ulasan pengguna pada aplikasi Duolingo di Google Play Store [2]. Penelitian dilakukan dengan menggunakan 1500 data ulasan lalu dianalisis dengan algoritma *Naive Bayes* yang mendapatkan skor akurasi sebesar 81%, presisi 80%, dan recall 98%, lalu

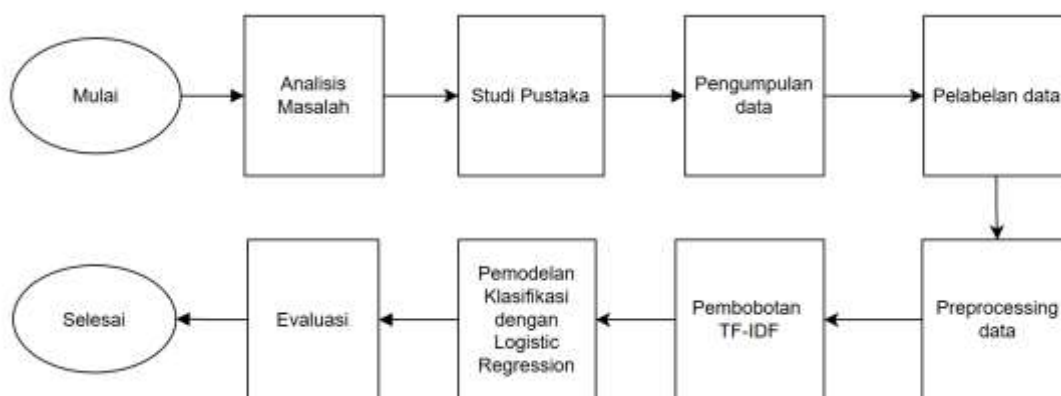
dilakukan pula analisis dengan algoritma *Support Vector Machine* yang mendapatkan skor akurasi sebesar 85%, presisi 87%, dan recall 94% [2]. Terdapat pula penelitian yang dilakukan pada tahun 2024 dengan judul “Analisis Sentimen Ulasan Aplikasi Access by KAI Menggunakan Algoritma Machine Learning” yang membandingkan penggunaan algoritma *Logistic Regression*, *Random Forest*, dan *Naive Bayes* dengan menggunakan 9.124 data yang berasal dari Google Play Store yang diklasifikasikan menjadi ulasan positif, negatif, dan netral [3]. Dari penelitian tersebut didapatkan hasil bahwa algoritma *Logistic Regression* memiliki nilai akurasi tertinggi yaitu 84% kemudian algoritma *Random Forest* nilai akurasinya adalah 78% dan algoritma *Naive Bayes* nilai akurasinya adalah 73% [3].

Pada penelitian ini akan menerapkan *machine learning* dengan metode *Logistic Regression* dimana pada *preprocessing* datanya menggunakan pembobotan TF-IDF dan pelabelan ulasan terdiri dari positif dan negatif sebagai dasar untuk menganalisis sentimen ulasan pengguna pada aplikasi Duolingo.

2. Metode Penelitian

2.1 Alur Penelitian

Penelitian ini dimulai dengan menganalisis permasalahan lalu dilanjutkan dengan studi pustaka untuk mengumpulkan materi tentang penerapan algoritma *Logistic Regression*. Kemudian dilanjutkan dengan mengumpulkan dataset ulasan aplikasi Duolingo pada Google Play Store. Lalu dilakukan pelabelan dataset yang akan digunakan pada penelitian. Selanjutnya dilakukan tahapan *preprocessing* data dan dilanjutkan dengan melakukan pembobotan dengan metode TF-IDF. Kemudian dilakukan pelatihan dengan menggunakan *Logistic Regression* dan tahap terakhir adalah evaluasi untuk menghitung performa dari model yang digunakan. Gambaran alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

2.2 Pengumpulan Data

Data yang digunakan pada penelitian adalah data sekunder yang merupakan ulasan aplikasi Duolingo yang diambil dari Google Play Store. Data yang digunakan sebanyak 8.648 data yang diambil dari ulasan pada tanggal 1 Mei 2025 hingga 20 Juni 2025 dengan ulasan berbahasa Indonesia. Data yang diambil berupa *score*, tanggal ulasan, dan ulasan.

2.3 Pelabelan Data

Data yang digunakan pada penelitian ini belum terdapat label sentimen positif maupun negatif, tetapi terdapat *score* yang dapat dijadikan acuan untuk melabelkan data. Data dengan *score* 1, 2, dan 3 dilabelkan menjadi sentimen negatif dan data dengan *score* 4 dan 5 dilabelkan menjadi sentimen positif [4].

2.4 Preprocessing Data

Preprocessing data dilakukan dengan beberapa tahapan dengan tujuan agar data yang digunakan dapat lebih mudah dipahami dan diproses oleh komputer. Proses *preprocessing* data pada penelitian ini dilakukan dari *cleaning* data sampai *stemming*.

- a. *Cleaning data*
Tahapan pertama pada *preprocessing* data adalah *cleaning* data yang dilakukan untuk menghilangkan informasi yang tidak dibutuhkan dari data [5]. *Cleaning* data dilakukan dengan menghapus emoji, link, karakter spesial dan simbol, angka, kata yang berulang, dan spasi yang berlebihan yang diganti dengan spasi kosong.
- b. *Case Folding*
Case folding adalah tahapan untuk membuat semua ulasan menjadi huruf kecil atau *lowercase*. *Case folding* dilakukan untuk menyamakan format penulisan kata, sehingga kata yang sama tetapi berbeda di penggunaan huruf kapital tidak akan dianggap sebagai kata yang berbeda [5].
- c. *Tokenisasi*
Tahapan ketiga adalah tokenisasi yang dilakukan dengan memisahkan setiap kata pada ulasan. Pada tokenisasi, spasi tidak dianggap sebagai sebuah kata sehingga pemisahan dilakukan berdasarkan spasi tanpa menjadikan spasi sebagai bagian dari hasil tokenisasi [5].
- d. *Normalisasi*
Tahapan keempat adalah normalisasi yang dilakukan dengan melakukan perbaikan terhadap kata-kata yang tidak baku maupun kata-kata yang mengalami kesalahan penulisan [5]. Pada penelitian ini dibuat sebuah kamus yang berisi daftar kata-kata yang tidak baku maupun kata-kata yang mengalami kesalahan penulisan beserta dengan padanan kata dalam bentuk baku dan kata dengan ejaan yang benar.
- e. *Stopword Removal*
Tahapan kelima adalah *stopword removal* yang dilakukan untuk menghapus kata-kata yang dianggap tidak penting seperti kata ganti, kata sambung, maupun kata-kata yang terlalu umum [5]. Contoh katanya adalah “yang”, “adalah”, “saya”, dan lainnya.
- f. *Stemming*
Tahapan terakhir dari *preprocessing* data adalah *stemming* yang dilakukan dengan mengubah setiap kata menjadi bentuk dasar dengan menghilangkan imbuhan yang terdapat pada kata tersebut seperti awalan, sisipan, maupun akhiran [5].

2.5 TF-IDF

TF-IDF merupakan singkatan dari *Term Frequency-Inverse Document Frequency* adalah metode yang digunakan untuk menghitung kata dalam sebuah dokumen dengan menentukan nilai masing-masing kata dalam dokumen tersebut [6]. Terdapat langkah-langkah yang dilakukan untuk menerapkan TF-IDF, yaitu:

- a. *Term Frequency (TF)*
Term Frequency atau TF adalah langkah untuk menghitung frekuensi suatu kata dalam dokumen yang dipengaruhi oleh panjang teks atau dokumen [6]. Persamaan TF dapat dilihat pada persamaan 1 [6].

$$tf(t,d) = \text{count of } t \text{ in documents} \quad (1)$$

b. *Document Frequency* (DF)

Document Frequency atau DF adalah langkah untuk menghitung seberapa sering setiap kata muncul berdasarkan jumlah kemunculannya dalam dokumen [6]. Persamaan DF dapat dilihat pada persamaan 2 [6].

$$df(t) = \text{occurrence of } t \text{ in documents} \quad (2)$$

c. *Inversi Document Frequency* (IDF)

Inverse Document Frequency atau IDF adalah langkah untuk menentukan bobot dari sebuah dokumen atau teks berdasarkan dari hasil yang didapat dari TF dan DF [6]. Persamaan IDF dapat dilihat pada persamaan 3 dan persamaan TF-IDF dapat dilihat pada persamaan 4 [6].

$$idf(t) = \log\left(\frac{d}{df}\right) \quad (3)$$

Dimana :

$t = \text{term}$ (kata)

$d = \text{document}$ (jumlah dokumen)

$DF = \text{Document Frequency}$ (frekuensi dokumen berdasarkan kata)

$$W = TF \times (IDF + 1) \quad (4)$$

Dimana :

$W = \text{weight}$ (bobot kata)

$TF = \text{Term Frequency}$ (frekuensi kemunculan kata)

$IDF = \text{Inverse Document Frequency}$

2.6 Logistic Regression

Logistic Regression adalah metode analisis yang digunakan untuk melihat hubungan antara satu variabel yang ingin diprediksi dengan satu atau lebih variabel bebas [7]. Jika variabel yang ingin diprediksi hanya memiliki dua kemungkinan (biner), maka digunakan *logistic regression* biner [7]. Pada penelitian ini, ulasan akan dikategorikan ke dalam kelas positif dan negatif, sedangkan untuk kelas netral diabaikan karena fungsi sigmoid yang digunakan pada *logistic regression* menghasilkan output dalam rentang 0 sampai 1 yang merepresentasikan probabilitas data dikategorikan ke dalam salah satu dari dua kelas [6]. Persamaan sigmoid dapat dilihat pada persamaan 5 [6].

$$\text{sigmoid}(z) = \frac{1}{1+e^{-z}} \quad (5)$$

Dimana:

$z = \text{data input}$

2.7 Evaluasi

Pada penelitian ini evaluasi dilakukan dengan menggunakan metode *confusion matrix*. *Confusion matrix* bekerja dengan cara membandingkan hasil prediksi dari model dengan label asli pada data, yang merepresentasikan antara informasi yang sebenarnya dengan hasil prediksi klasifikasi [8]. *Confusion matrix* dapat dilihat pada gambar 2 [8].

	Actual Values	
Predicted Value	1 (Positive)	0 (Negative)
1 (Positive)	TP (True Positive)	FP (False Positive)
0 (Negative)	FN (False Negative)	TN (True Negative)

Gambar 2. *Confusion Matrix*

Confusion matrix menghasilkan beberapa metrik evaluasi yaitu *F1-score* yang merupakan nilai rata-rata harmonik dari *precision* dan *recall*, *accuracy* yang digunakan untuk menilai sejauh mana model mampu melakukan prediksi secara tepat, *recall* yang digunakan untuk mengukur rasio jumlah data positif yang berhasil dikenal dengan benar dibandingkan dengan seluruh data yang seharusnya positif, dan *precision* yang digunakan untuk menunjukkan proporsi data positif yang diprediksi dengan benar dibandingkan dengan seluruh data yang diprediksi sebagai positif oleh sistem [9]. Perhitungan *F1-score*, *accuracy*, *recall*, dan *precision* dapat dilihat pada persamaan berikut [9]:

$$F1 - score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (6)$$

$$Accuracy = \frac{TP + TN}{Total Data} \quad (7)$$

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

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

3. Hasil dan Diskusi

3.1 Pengumpulan dan Pelabelan Data

Pada penelitian ini digunakan data ulasan aplikasi Duolingo yang diambil dari Google Play Store dari tanggal 1 Mei 2025 hingga 20 Juni 2025 sebanyak 8.648 data. Data yang diambil berupa *score*, tanggal ulasan, dan ulasan. Data dengan *score* 1, 2, dan 3 dilabelkan menjadi sentimen negatif dan data dengan *score* 4 dan 5 dilabelkan menjadi sentimen positif sehingga dihasilkan data dengan sentimen negatif sebanyak 896 data dan data dengan sentimen positif sebanyak 7.752 data.

3.2 Preprocessing Data

Pada *preprocessing* data dimulai dengan melakukan *cleaning* yang dilakukan dengan menghapus emoji, link, karakter spesial dan simbol, angka, kata yang berulang, dan spasi yang berlebihan yang diganti dengan spasi kosong. Hasil *cleaning* dapat dilihat pada tabel 1.

Tabel 1. Hasil *Cleaning* Data

Sebelum Cleaning	Sesudah Cleaning
game ini sangat bagus dan aku sangat suka sama game ini, game yang sangat" bagus, truss kalo dia menasihati aku, aku merasa kalo aku ini adalah orang sangat pintar, Wow!!! aku tidak percaya! aku mendapatkan super duolingo thank you	game ini sangat bagus dan aku sangat suka sama game ini game yang sangat bagus truss kalo dia menasihati aku aku merasa kalo aku ini adalah orang sangat pintar Wow aku tidak percaya aku mendapatkan super duolingo thank you

Selanjutnya dilakukan tahap *case folding* untuk membuat semua ulasan menjadi huruf kecil atau *lowercase*. Hasil *case folding* dapat dilihat pada tabel 2.

Tabel 2. Hasil *Case Folding* Data

Sebelum Case Folding	Sesudah Case Folding
game ini sangat bagus dan aku sangat suka sama game ini game yang sangat bagus truss kalo dia menasihati aku aku merasa kalo aku ini adalah orang sangat pintar Wow aku tidak	game ini sangat bagus dan aku sangat suka sama game ini game yang sangat bagus truss kalo dia menasihati aku aku merasa kalo aku ini adalah orang sangat pintar wow aku tidak

Sebelum <i>Case Folding</i>	Sesudah <i>Case Folding</i>
percaya aku mendapatkan super duolingo thank you	percaya aku mendapatkan super duolingo thank you

Selanjutnya dilakukan tahap tokenisasi yang dilakukan dengan memisahkan setiap kata pada ulasan. Hasil tokenisasi dapat dilihat pada tabel 3.

Tabel 3. Hasil Tokenisasi Data

Sebelum Tokenisasi	Sesudah Tokenisasi
game ini sangat bagus dan aku sangat suka sama game ini game yang sangat bagus truss kalo dia menasihati aku aku merasa kalo aku ini adalah orang sangat pintar wow aku tidak percaya aku mendapatkan super duolingo thank you	['game', 'ini', 'sangat', 'bagus', 'dan', 'aku', 'sangat', 'suka', 'sama', 'game', 'ini', 'game', 'yang', 'sangat', 'bagus', 'truss', 'kalo', 'dia', 'menasihati', 'aku', 'aku', 'merasa', 'kalo', 'aku', 'ini', 'adalah', 'orang', 'sangat', 'pintar', 'wow', 'aku', 'tidak', 'percaya', 'aku', 'mendapatkan', 'super', 'duolingo', 'thank', 'you']

Selanjutnya dilakukan tahap normalisasi untuk melakukan perbaikan terhadap kata-kata yang tidak baku maupun kata-kata yang mengalami kesalahan penulisan dengan membuat sebuah kamus yang berisi daftar kata-kata yang tidak baku maupun kata-kata yang mengalami kesalahan penulisan beserta dengan padanan kata dalam bentuk baku dan kata dengan ejaan yang benar. Hasil normalisasi dapat dilihat pada tabel 4.

Tabel 4. Hasil Normalisasi Data

Sebelum Normalisasi	Sesudah Normalisasi
['game', 'ini', 'sangat', 'bagus', 'dan', 'aku', 'sangat', 'suka', 'sama', 'game', 'ini', 'game', 'yang', 'sangat', 'bagus', 'truss', 'kalo', 'dia', 'menasihati', 'aku', 'aku', 'merasa', 'kalo', 'aku', 'ini', 'adalah', 'orang', 'sangat', 'pintar', 'wow', 'aku', 'tidak', 'percaya', 'aku', 'mendapatkan', 'super', 'duolingo', 'thank', 'you']	['permainan', 'ini', 'sangat', 'bagus', 'dan', 'aku', 'sangat', 'suka', 'sama', 'permainan', 'ini', 'permainan', 'yang', 'sangat', 'bagus', 'terus', 'kalau', 'dia', 'menasihati', 'aku', 'aku', 'merasa', 'kalau', 'aku', 'ini', 'adalah', 'orang', 'sangat', 'pintar', 'wow', 'aku', 'tidak', 'percaya', 'aku', 'mendapatkan', 'super', 'duolingo', 'terima kasih', 'kamu']

Selanjutnya dilakukan tahap *stopword removal* untuk menghapus kata-kata yang dianggap tidak penting seperti kata ganti, kata sambung, maupun kata-kata yang terlalu umum dengan menggunakan *library* NLTK atau *Natural Language Toolkit* untuk mengambil daftar *stopwords* berbahasa Indonesia. Hasil *stopword removal* dapat dilihat pada tabel 5.

Tabel 5. Hasil *Stopword Removal* Data

Sebelum <i>Stopword Removal</i>	Sesudah <i>Stopword Removal</i>
['permainan', 'ini', 'sangat', 'bagus', 'dan', 'aku', 'sangat', 'suka', 'sama', 'permainan', 'ini', 'permainan', 'yang', 'sangat', 'bagus', 'terus', 'kalau', 'dia', 'menasihati', 'aku', 'aku', 'merasa', 'kalau', 'aku', 'ini', 'adalah', 'orang', 'sangat', 'pintar', 'wow', 'aku', 'tidak', 'percaya', 'aku', 'mendapatkan', 'super', 'duolingo', 'terima kasih', 'kamu']	['permainan', 'bagus', 'suka', 'permainan', 'permainan', 'bagus', 'menasihati', 'orang', 'pintar', 'wow', 'percaya', 'super', 'duolingo', 'terima kasih']

Tahap terakhir dari proses *preprocessing* data adalah tahap *stemming* yang dilakukan dengan mengubah setiap kata menjadi bentuk dasar dengan menghilangkan imbuhan yang terdapat pada kata tersebut dengan menggunakan *library* Sastrawi. Hasil *stemming* dapat dilihat pada tabel 6.

Tabel 6. Hasil *Stemming* Data

Sebelum <i>Stemming</i>	Sesudah <i>Stemming</i>
['permainan', 'bagus', 'suka', 'permainan', 'permainan', 'bagus', 'menasihati', 'orang', 'pintar', 'wow', 'percaya', 'super', 'duolingo', 'terima kasih']	['main', 'bagus', 'suka', 'main', 'main', 'bagus', 'nasihat', 'orang', 'pintar', 'wow', 'percaya', 'super', 'duolingo', 'terima kasih']

3.3 Pembagian Data Latih dan Data Uji

Setelah dilakukan *preprocessing* data, data akan dibagi menjadi data latih dan data uji. Perbandingan untuk data latih dan data uji adalah 80 : 20 [2]. Sehingga dari 8.648 data, data latih yang digunakan adalah sebanyak 6.918 data dan data uji yang digunakan adalah sebanyak 1.730 data. Hasil pembagian data dapat dilihat pada gambar 3.

```
from sklearn.model_selection import train_test_split

X=df['stemming']
Y=df['sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

print("Jumlah data latih:", len(X_train))
print("Jumlah data uji:", len(X_test))

Jumlah data latih: 6918
Jumlah data uji: 1730
```

Gambar 3. Pembagian Data Latih dan Data Uji

3.4 Pembobotan TF-IDF

Pembobotan TF-IDF dilakukan untuk menghitung kata dalam sebuah dokumen dengan menentukan nilai masing-masing kata dalam dokumen tersebut. Pembobotan TF-IDF dilakukan dengan menggunakan *library TfidfVectorizer*. Implementasi TF-IDF dapat dilihat pada gambar 4.

```
from sklearn.feature_extraction.text import TfidfVectorizer

X_train = X_train.astype(str).tolist()
X_test = X_test.astype(str).tolist()

vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

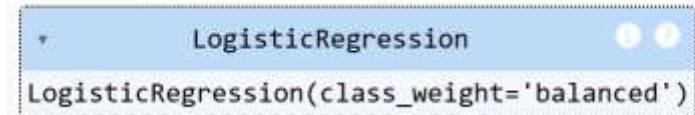
Gambar 4. Implementasi TF-IDF

3.5 Pemodelan Klasifikasi dengan *Logistic Regression*

Tahap selanjutnya adalah melakukan klasifikasi menggunakan metode *logistic regression* dengan menggunakan parameter *class weight = balanced*. Pada penelitian ini jumlah data untuk sentimen positif lebih banyak dibandingkan dengan sentimen negatif, maka digunakan *class weight* untuk mengatasi ketidakseimbangan antar kelas dengan menetapkan bobot yang lebih pada kelas yang memiliki jumlah data yang kecil yaitu sentimen negatif [10] [11]. Implementasi klasifikasi menggunakan *logistic regression* dapat dilihat pada gambar 5.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

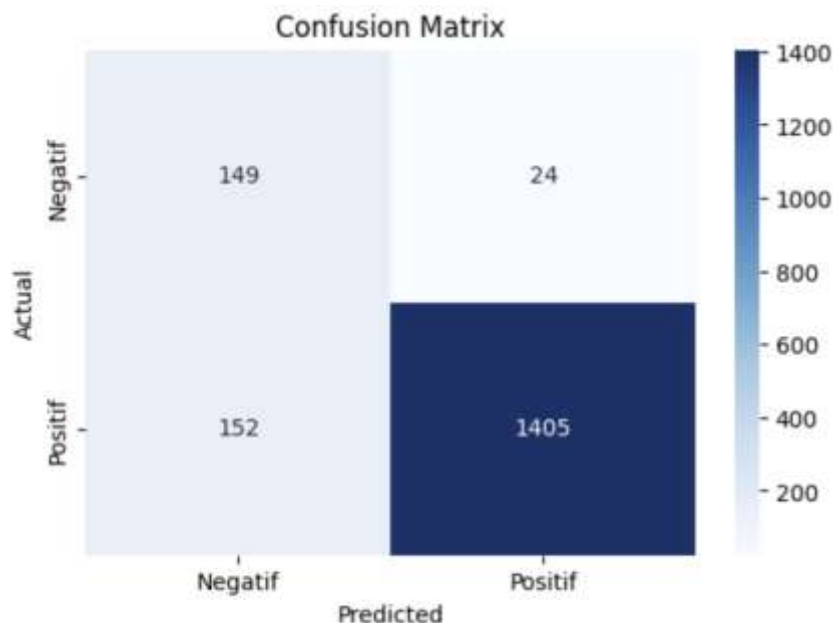
model = LogisticRegression(class_weight='balanced')
model.fit(X_train_tfidf, y_train)
```



Gambar 5. Implementasi *Logistic Regression*

3.6 Evaluasi

Pada penelitian ini evaluasi dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* menampilkan informasi berupa hasil perbandingan antara prediksi dari *logistic regression* dengan label asli dari data. Berdasarkan evaluasi menggunakan *confusion matrix* didapatkan hasil bahwa terdapat 1.405 data positif yang diprediksi positif, 152 data positif yang diprediksi negatif, 149 data negatif yang diprediksi negatif, dan 24 data negatif yang diprediksi positif. Hasil *confusion matrix* dapat dilihat pada gambar 6.



Gambar 6. Hasil *Confusion Matrix*

Setelah didapatkan hasil *confusion matrix* lalu akan menghasilkan beberapa metrik evaluasi

seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil evaluasi dapat dilihat pada gambar 7.

Hasil Klasifikasi:				
	precision	recall	f1-score	support
Negatif	0.4950	0.8613	0.6287	173
Positif	0.9832	0.9024	0.9411	1557
accuracy			0.8983	1730
macro avg	0.7391	0.8818	0.7849	1730
weighted avg	0.9344	0.8983	0.9098	1730

Gambar 7. Hasil Evaluasi

Berdasarkan hasil evaluasi terhadap *logistic regression* untuk menganalisis sentimen ulasan aplikasi Duolingo didapatkan bahwa skor *accuracy* sebesar 89,83%, skor *precision* sebesar 73,91%, skor *recall* sebesar 88,18%, dan skor *f1-score* sebesar 78,49%.

4. Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat disimpulkan bahwa algoritma *logistic regression* menghasilkan akurasi yang baik untuk analisis sentimen ulasan aplikasi Duolingo pada ulasan di Google Play Store. Data yang digunakan yaitu sebanyak 8.648 data lalu dilakukan pelabelan data menjadi sentimen positif dan sentimen negatif. Kemudian dilakukan *preprocessing* data dan dilanjutkan dengan membagi data menjadi data latih dan data uji dengan rasio 80:20 lalu dilakukan pembobotan TF-IDF. Selanjutnya data diklasifikasikan menggunakan algoritma *logistic regression* dengan menggunakan parameter *class weight = balanced* sehingga menghasilkan skor *accuracy* 89,83%. Hasil evaluasi juga menunjukkan nilai *precision* sebesar 73,91%, nilai *recall* sebesar 88,18%, dan nilai *f-1 score* sebesar 78,49%. Penelitian selanjutnya diharapkan untuk dapat melakukan analisis sentimen dengan menggunakan algoritma yang lain atau menggunakan metode *deep learning*.

Daftar Pustaka

- [1] "EF EPI EF English Proficiency Index A Ranking of 116 Countries and Regions by English Skills." Accessed: Jun. 20, 2025. [Online]. Available: <https://www.ef.com/wwen/epi/>
- [2] F. Alifiana, F. Asnawi, I. A. Ihsannudin, M. Alif, M. Baihaqy, and D. Asmarajati, "ANALISIS SENTIMEN APLIKASI DUOLINGO MENGGUNAKAN ALGORITMA NAÏVE BAYES DAN SUPPORT MACHINE LEARNING," *JURNAL DEVICE*, vol. 13, no. 2, pp. 223–230, Nov. 2023.
- [3] M. A. S. Nugroho, D. Susilo, and D. Retnoningsih, "ANALISIS SENTIMEN ULASAN APLIKASI "ACCESS BY KAI" MENGGUNAKAN ALGORITMA MACHINE LEARNING," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 7, no. 2, pp. 820–827, 2024, doi: 10.37600/tekinkom.v7i2.1854.
- [4] R. Maulana, A. Voutama, and T. Ridwan, "ANALISIS SENTIMEN ULASAN APLIKASI MYPERTAMINA PADA GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA NBC," *Jurnal Teknologi Terpadu*, vol. 9, no. 1, pp. 42–48, 2023.
- [5] N. M. G. S. Nirmala and N. A. S. ER, "Analisis Sentimen dengan Logistic Regression untuk Deteksi Kata pada Livin' by Mandiri," *JNATIA*, vol. 2, no. 4, pp. 869–878, Aug. 2024.
- [6] E. Darmaja, V. C. Mawardi, and N. J. Perdana, "REVIEW SENTIMEN ANALISIS APLIKASI SOSIAL MEDIA DI GOOGLE PLAYSTORE MENGGUNAKAN METODE LOGISTIC REGRESSION," Dec. 2021.
- [7] Ash Shiddicky and Surya Agustian, "Analisis Sentimen Masyarakat Terhadap Kebijakan Vaksinasi Covid-19 pada Media Sosial Twitter menggunakan Metode Logistic Regression," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3,

- no. 2, pp. 99–106, Aug. 2022, doi: 10.37859/coscitech.v3i2.3836.
- [8] N. Hendrastuty, A. Rahman Isnain, and A. Yanti Rahmadhani, “Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine,” *Jurnal Informatika : Jurnal Pengembangan IT (JPIT)*, vol. 6, no. 3, pp. 150–155, Sep. 2021, [Online]. Available: <http://situs.com>
- [9] H. Hidayatullah and Y. Umaidah, “PENERAPAN NAÏVE BAYES DENGAN OPTIMASI INFORMATION GAIN DAN SMOTE UNTUK ANALISIS SENTIMEN PENGGUNA APLIKASI CHATGPT,” Jun. 2023.
- [10] geeksforgeeks, “How Does the class_weight Parameter in Scikit-Learn Work?” Accessed: Jul. 30, 2025. [Online]. Available: <https://www.geeksforgeeks.org/machine-learning/how-does-the-classweight-parameter-in-scikit-learn-work/>
- [11] Masriah and W. Tisno Atmojo, “Analisis Sentimen Terhadap Ulasan pada Aplikasi Astro Menggunakan Algoritma Support Vector Machine,” *Jurnal Komtika (Komputasi dan Informatika)*, vol. 9, no. 1, pp. 11–23, May 2025, doi: 10.31603/komtika.v9i1.13129.