

Implementasi NLP(*Natural Language Processing*) Dasar pada Analisis *Sentiment Review Spotify*

Marsha Auriel Prasetya¹, Miftakhul Wulandari², Siti Alvi Nikmah³

Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas
Nusantara PGRI Kediri

E-mail: ¹marshaauriel@gmail.com, ²miftakhulwulandari06@gmail.com, ³sitialvinikmah@gmail.com

Abstrak – Dalam Penelitian ini, analisis sentiment terhadap ulasan Spotify dilakukan dengan menggunakan teknik *Natural Language Processing*(NLP). Langkah-langkah prapemrosesan khusus diimplementasikan untuk meningkatkan kinerja klasifikasi sentiment, yang bertujuan untuk mencapai hasil yang lebih unggul dibandingkan dengan solusi berbasis NLP (*Natural Language Processing*). Prosedur prapemrosesan ini sangat penting dalam mengategorikan emosi secara efektif. Analisis sentiment yang dihasilkan mengelompokkan sentiment kedalam tiga kelas : *negative*, *netral* dan *positif*. Kelas-kelas turunan ini menjadi sasaran analisis komparatif yang menggunakan metodologi *machine learning*. Penilaian komparatif menentukan bahwa metode *Naïve bayes* dan *Logistic Regression* menunjukkan kinerja klasifikasi yang lebih sukses dibandingkan dengan pendekatan lainnya. Untuk mengatasi ketidakseimbangan data dalam dataset, berbagai metode penanganan ketidakseimbangan digunakan dan disandingkan satu sama lain melalui investigasi ini. Penelitian ini berkontribusi pada pemahaman yang lebih komprehensif tentang analisis sentiment dalam konteks ini. Pada kedua metode ini juga pasti terdapat perbedaan hasil antara algoritma *naïve bayes* dan juga algoritma *logistic regression*, untuk mengetahui algoritma mana yang lebih unggul untuk mendapatkan akurasi tertinggi dalam pengolahan dataset spotify ini, maka harus dilakukan antara perbandingan algoritmat *Naive bayes* dan algoritma *Logistic Regression*.

Kata Kunci — Analisis Sentimen , Klasifikasi , *Machine Learning* ,NLP, spotify

1. PENDAHULUAN

Di era digital saat ini, hiburan memiliki peran penting dalam kehidupan sehari-hari. Salah satu hiburan yang sering kita lakukan yaitu mendengarkan *music* hal ini merupakan salah satu hiburan yang dapat membantu kita membangkitkan semangat beraktifitas atau membantu menghilangkan penat saat melakukan aktivitas pekerjaan. Salah satu contoh media *music* yang *trend* saat ini adalah *Spotify*. *Spotify* merupakan layanan *streaming music* terkemuka yang memiliki jutaan pengguna. Aplikasi ini didirikan oleh Daniel Ek dan Martin Lorentzon pada tahun 2006 di Stockholm, Swedia. Aplikasi ini diluncurkan secara resmi pada tanggal 7 Oktober 2008. *Spotify* memungkinkan *user* untuk memilih lagu, album, dan *podcast* dari berbagai *genre music*. Pengguna aplikasi ini sering kali mengungkapkan pengalaman, preferensi dan kritik melalui ulasan tertulis, yang menawarkan *repository* data yang kaya akan *sentiment* yang bernilai.

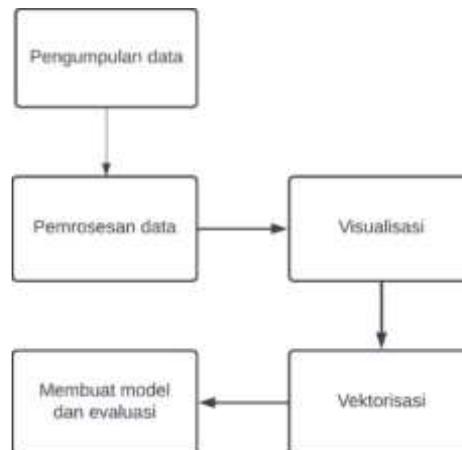
Analisis *sentiment* adalah sebuah proses untuk menentukan *sentiment* atau opini dari seseorang yang diwujudkan dalam bentuk teks dan bias dikategorikan sebagai sentiment positif atau negatif[1]. Analisis *sentiment* dalam ulasan *Spotify* tidak hanya memberikan wawasan tentang kepuasan dan ketidakpuasan pengguna, tetapi juga menawarkan sudut pandang yang unik untuk memahami preferensi dan tren *music* yang terus berkembang. Penelitian ini mempelajari ranah analisis *sentiment* yang diterapkan pada ulasan *Spotify*, yang bertujuan untuk mengeksplorasi interaksi *user* dan aplikasi *music* di era digital. Melalui analisis dan interpretasi dengan *sentiment*, penelitian ini berkontribusi pada pemahaman yang lebih dalam tentang hubungan yang rumit antara teknologi, *music*, dan emosi manusia [2].

Penelitian ini bertujuan untuk melakukan analisis *sentiment* berbasis kecerdasan buatan pada ulasan *Spotify* menggunakan kumpulan data yang diperoleh dari salah satu penyedia sumberdaya terbuka secara online. Dalam penelitian ini menawarkan pendekatan terintegrasi yang dimulai dari prapemrosesan hingga klasifikasi, dapat memperkuat keefektifan analisis *sentiment* dalam konteks ulasan *music* di aplikasi *Spotify*.

2. METODE PENELITIAN

Dalam penelitian ini penulis menggunakan dua metode untuk mencapai dan melihat akurasi tertinggi dari metode tersebut. Metode yang dipakai yaitu *Naïve Bayes* dan *Logistic Regression*, disini penulis juga menggunakan pendekatan berbasis NLP (*Natural Language Processing*) dimana pendekatan ini mengimplementasikan teknik pemrosesan bahasa alami untuk meningkatkan kinerja dari kedua metode yang dipakai.

Dalam pendekatan NLP (*Natural Language Processing*) ini melibatkan beberapa guna untuk mendapatkan akurasi terbaik, Pada Gambar 1 berikut merupakan tahapan atau langkah-langkah dalam melakukan penelitian:



Gambar 1. Alur Penelitian

Berikut merupakan penjelasan dari gambar 1 yang tertera diatas:

2.1 Pengumpulan data

Dataset yang digunakan dalam penelitian ini diperoleh dari salah satu *platform* yang menyediakan berbagai sumber daya terbuka yang biasanya digunakan oleh para peneliti yaitu kaggle, disini peneliti menggunakan database yang berisi data-data dari *spotify*. Beberapa informasi yang ada di dalam dataset *spotify* diantara yaitu *time submitted*, *reviews*, *rating*, *total thumbsup*, dan *review reply*.

2.2 Pemrosesan data

Tahap ini dilakukan untuk menyiapkan data yang akan dianalisis. Pemrosesan data ini penting untuk dilakukan guna memastikan bahwa data yang digunakan dalam analisis adalah data yang berkualitas sehingga dapat menghasilkan akurasi yang akurat. Penyiapan dataset ini melalui beberapa *preprocess* diantaranya yaitu:

- 1) *Expanding contractions*, tahap ini berisi proses untuk mengubah kontraksi dari sebuah kata menjadi bentuk utuh. Kontraksi kata sendiri merupakan bentuk singkat dari dua kata atau lebih yang digabungkan dengan menghilangkan satu atau beberapa huruf, misal kata *I'm* merupakan kata yang berasal dari kata *I am*.
- 2) *Case Folding*, yaitu mengubah teks menjadi huruf kecil, untuk membantu menormalkan teks, yang bervariasi berbeda seperti ada huruf besar kecil di awal dan ditengah ataupun di akhiran teks.
- 3) *Removing stopwords* adalah kata yang dianggap umum atau tidak informatif, kata ini biasanya dihapus dari teks sebelum melakukan analisis. Contoh dari kata *stopwords* ini yaitu *and a, an, in, of* dan lainnya[3].
- 4) Lematisasi, proses ini digunakan untuk memproses bahasa alami yang bertujuan untuk mengubah kata kedalam bentuk dasar mereka, misalnya kata dasar *better* adalah *good*.

2.3 Visualisasi

Tahap ini akan memuat tentang visualisasi dari *dataset* yang digunakan. Pada tahap ini penulis memvisualisasikan *Rating* dari *dataset* dimana *rating* ini terbagi menjadi 3 yaitu *good*, *bad rating*, dan *netral*, selain itu peneliti juga memberikan perintah *wordcloud* dimana kata yang lebih sering digunakan dalam *dataset* akan berukuran besar.

2.4 Vektorisasi

Vektorisasi merupakan proses untuk mengubah data *non-numeric* menjadi *numeric*, hal ini dilakukan dengan merepresentasikan setiap kata dalam teks sebagai *vector*, lalu menggabungkan *vector* yang ada untuk mewakili keseluruhan teks. Proses ini melibatkan perhitungan kemunculan setiap kata dalam dataset. Vektorisasi TF-IDF (*Term Frequency Inverse Document Frequence*) merupakan teknik yang umum digunakan dalam pemrosesan bahasa alami ini [4]. TF-IDF merupakan proses yang digunakan untuk melakukan transformasi data dari sebuah teks menjadi sebuah numerik guna dilakukan untuk pembobotan pada tiap kata [5].

2.5 Membuat model dan evaluasi

Setelah *preprocessing* data dilakukan, dan model klasifikasi pembelajaran mesin telah dibuat untuk kolom Kediri, 13 Januari 2024

rating, dilanjutkan dengan menerapkan metode *naive bayes* dan juga *logistic regression*. Metode *naive bayes* merupakan metode klasifikasi dengan nilai yang ditentukan oleh probabilitas data [6], sedangkan *logistic regression* merupakan sebuah model statistik yang digunakan untuk menentukan apakah *independent variable* mempunyai pengaruh terhadap sebuah *binary dependent variable* [7]. Nantinya peneliti akan membandingkan kedua metode tersebut kedalam program yang kami buat guna mengetahui perbedaan akurasi dari tiap metode tersebut. Dalam perbandingan tersebut kinerja metode-metode tersebut diukur menggunakan parameter klasifikasi akurasi, *precision*, *recall* dan *F1-score*. Klasifikasi *naive bayes* merupakan klasifikasi yang bersifat *supervised learning*, kinerja dari metode ini mempunyai waktu klasifikasi yang singkat sehingga cepat dalam memproses sistem analisis *sentiment* [8]. Berikut merupakan rumus untuk menghitung *naive bayes* secara manual:

$$P(H|X) = \frac{P(H|X)P(H)}{P(X)} \dots\dots\dots(1)$$

Keterangan:

- X : Data kelas tidak diketahui
- H : Data hipotesis X adalah kelas yang terpisah
- P(H|X) : Probabilitas hipotesis H berdasarkan kondisi
- P(H) : Probabilitas hipotesis H
- P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis
- P(X) : Probabilitas X

Logistic Regression memiliki sejumlah keunggulan yang membuatnya menjadi pilihan yang baik dalam berbagai tugas klasifikasi. Salah satu keunggulan utamanya adalah interpretabilitas model atau hasilnya mudah diinterpretasikan, memungkinkan pemahaman yang jelas tentang hubungan antara fitur dan probabilitas kelas yang dihasilkan. Metode ini juga termasuk kedalam algoritma klasifikasi, berikut merupakan rumus perhitungan manual dari metode *logistic regression*:

$$P(Y = 1) = \frac{1}{1+e^{-(b_0+b_1x_1+b_2+\dots+b_k+X_k)}} \dots\dots\dots(2)$$

Keterangan:

- P(Y=1) : probabilitas *variable* dependen Y sama dengan 1
- e : basis logaritma *natural*
- b₀ : intercept (konstanta)
- b₁, b₂ : koefisien yang menunjukkan seberapa besar perubahan dalam *variable independent* (X₁,X₂,...X_k)

Setelah melalui proses pembuatan model untuk perhitungan akurasi dari kedua metode *naive bayes* dan juga *logistic regression*, tahap selanjutnya yaitu Evaluasi hasil pemodelan dari perbandingan kedua metode untuk menilai tingg akurasi dari tiap metode.

3. HASIL DAN PEMBAHASAN

3.1 Pengambilan Data

Pengumpulan data untuk penelitian ini diambil dari Kaggle, yang diambil dari ulasan pengguna aplikasi *Spotify* di *Play Store* menggunakan teknik web *scraping* dan terkumpul 61596 ulasan yang berisi ulasan baik, ulasan buruk, dan ulasan netral.

	Time submitted	Review	Rating	Total thumbsup	Reply
0	7/9/2022 13:08	I buy any music that I like it is super	5	0	NaN
1	7/9/2022 13:05	improve recommend songs and to find similar so...	5	0	NaN
2	7/9/2022 13:00	Android user - there are loads of glitches wit...	2	0	NaN
3	7/9/2022 13:00	I can't listen to my DOWNLOADED playlist while...	3	0	NaN
4	7/9/2022 12:58	It always crashing down, unable to play and ju...	1	0	NaN
-	-	-	-	-	-
61564	1/1/2022 9:26	Too many ads! There are 3, 30 second ads back ...	1	1	NaN
61565	1/1/2022 9:17	I've been using Spotify for about 3+ years now...	5	26	NaN
61566	1/1/2022 8:44	the only star Spotify doesn't get is the fact ...	4	0	NaN
61567	1/1/2022 8:19	This app has rmmies from Music Artist that f...	5	0	NaN
61568	1/1/2022 5:58	It selects music based on your taste makes bes...	5	4	NaN

61569 rows x 5 columns

Gambar 2 . Pengambilan data

3.2 Pemrosesan Data

Setelah mengumpulkan data Langkah berikutnya yaitu persiapan data yang diperoleh dan diolah, tahapan awal yaitu pemrosesan data ini mencakup kegiatan membersihkan data agar siap untuk langkah selanjutnya yaitu Visualisasi. Dalam pemrosesan data dibagi menjadi beberapa Langkah: *Expanding Contractions, Case Folding, Stopword, Lemmatisasi*.

a. *Expanding Contractions*

Dengan memperluas kontraksi ke bentuk penuhnya untuk meningkatkan akurasi analisis karena kontraksi dapat bersifat ambigu dan dapat menimbulkan kebingungan dalam konteks tertentu. Oleh karena itu, perluasan kontraksi merupakan langkah pra-pemrosesan yang penting dalam banyak aplikasi NLP (*Natural Language Processing*) untuk memastikan analisis data teks yang akurat dan konsisten.

Tabel 1 *Expanding Constraction*

<i>Before</i>	<i>After</i>
<i>Can't</i>	<i>Cannot</i>
<i>I'm</i>	<i>I am</i>

b. *Case Folding*

Mengubah teks menjadi huruf kecil, untuk membantu menormalkan teks, yang bervariasi berbeda seperti ada huruf besar kecil di awal dan ditengah ataupun di akhiran teks. Mengubah semua teks menjadi huruf kecil membantu memastikan konsistensi data, yang dapat meningkatkan akurasi dan keandalan model NLP (*Natural Language Processing*) [9]. Misalnya, jika sebagian teks menggunakan huruf besar dan sebagian lagi menggunakan huruf kecil, akan lebih sulit bagi model untuk mengidentifikasi pola dan membuat prediksi yang akurat.

Tabel 2 *Case Folding*

<i>Before</i>	<i>After</i>
<i>I Cannot Listen To My Downloaded</i>	<i>i cannot listen to my downloaded</i>

c. *Removing Stopword*

Menghapus *stopwords*, *stopwords* sendiri adalah kata-kata yang dianggap umum atau tidak informatif dan biasanya dihapus[5], dengan melakukan pengurangan kosakata, normalisasi dengan mengurangi jumlah variabilitas dalam data. Dan selanjutnya konsistensi dengan mengubah teks, tanda baca dapat mengurangi jumlah variasi kata atau frasa yang sama serta meningkatkan akurasi dan kualitas model NLP (*Natural Language Processing*)

Tabel 3 *Stopword*

<i>the</i>	<i>and</i>	<i>a</i>	<i>an</i>	<i>in</i>	<i>to</i>
------------	------------	----------	-----------	-----------	-----------

d. Lematisasi

Dan yang terakhir adalah Lematisasi adalah proses mereduksi kata menjadi bentuk dasar, tujuan dari *Lematisasi* adalah mengubah kata menjadi bentuk kanonik atau kamusnya yang membantu mengurangi dimensi data dan meningkatkan akurasi model NLP (*Natural Language Processing*). Dengan mereduksi semua variasi kata ke bentuk dasarnya, kita dapat memperlakukan berbagai bentuk kata yang sama sebagai satu istilah, lemmatisasi dapat membantu mengatasi masalah ketersebaran, yang dapat terjadi ketika berbagai bentuk kata yang sama diperlakukan sebagai istilah yang terpisah, sehingga menyebabkan *overfitting* dan berkurangnya akurasi. Secara keseluruhan, lemmatisasi adalah teknik yang berguna untuk mengurangi dimensi data dan meningkatkan akurasi dan efisiensi model NLP (*Natural Language Processing*).

Tabel 4 Lematisasi

<i>Before</i>	<i>After</i>
<i>Running</i>	<i>Run</i>

3.4 Vektorisasi

Vektorisasi mengacu pada proses mengubah data *non-numeric*, yaitu seperti teks atau gambar, menjadi vektor numerik yang dapat digunakan sebagai masukan ke model pembelajaran mesin atau algoritma lainnya. Dalam pemrosesan bahasa alami NLP (*Natural Language Processing*), misalnya vektorisasi digunakan untuk mengubah data teks, seperti kalimat atau dokumen, menjadi vektor numerik yang menangkap makna teks sebagai vektor, lalu menggabungkan vektor-vektor tersebut untuk mewakili keseluruhan teks.

a. *Count* vektorisasi

Count vektorisasi atau vektorisasi hitungan adalah teknik dalam menerjemahkan bahasa alami NLP (*Natural Language Processing*) untuk mengubah sebagian teks menjadi representasi numerik yang dapat digunakan dalam algoritma *machine learning*, prosesnya melibatkan pengungkapan setiap kata dalam dokumen dan membuat vektor di mana setiap elemen memiliki jumlah kata tertentu dalam dokumen, kemudian dapat digunakan sebagaimana untuk berbagai algoritma *machine learning*.

Tabel 5 Vektorisasi

Before	After
“the”, “song”, “is”, “amazing”, “melody”	[1, 1, 1, 1, 1]

Vektorisasi menunjukkan bahwa setiap kata muncul sekali dalam kalimat atau merepresentasikan frekuensi kemunculan kata-kata dalam kalimat tersebut. Namun vektorisasi hitungan memiliki beberapa batasan, seperti ukuran vektor yaitu jumlah dimensi dalam vektor yang merepresentasikan kata-kata, frekuensi dominasi kata-kata umum yang sering muncul dan besarnya jumlah kata-kata unik dalam teks.

b. TF-IDF Vektorisasi

Vektorisasi TF-IDF (*Term Frekuensi-Invers Dokumen Frekuensi*) adalah teknik yang umum digunakan dalam pemrosesan bahasa alami NLP (*Natural Language Processing*) untuk mengubah kumpulan dokumen teks menjadi format numerik yang dapat dengan mudah diproses oleh algoritma *machine learning*[7]. Vektorisasi TF-IDF melibatkan dua langkah utama yaitu perhitungan Term Frekuensi (TF) dan perhitungan Frekuensi Dokumen *Invers*(IDF).

Tabel 6 TD-IDF

Term Frekuensi (TF)	Frekuensi Dokumen <i>Invers</i>
$TF(t,d) = (\text{Berapa kali istilah } t \text{ muncul di dokumen } d) / (\text{jumlah total istilah di dokumen } d)$	$IDF(t) = \log_e (\text{jumlah dokumen} / \text{jumlah dokumen dengan term } t \text{ di dalamnya})$

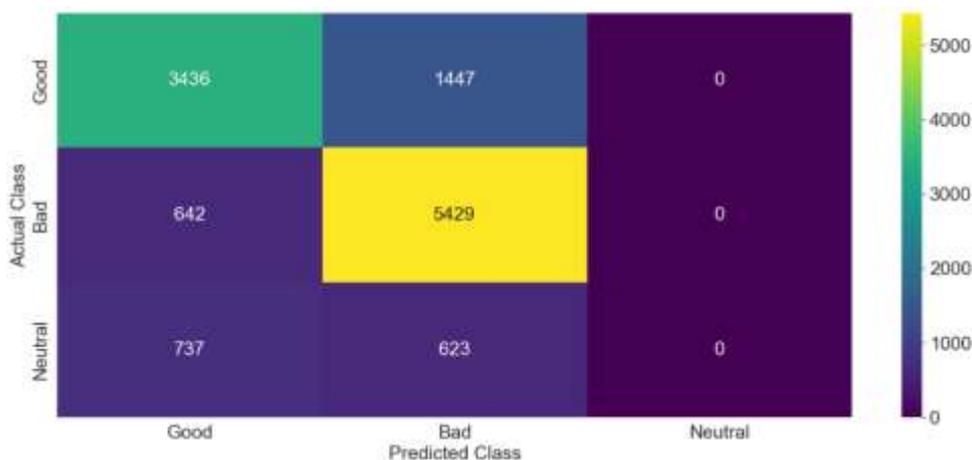
3.5 Evaluasi

Pada tahapan Evaluasi menunjukkan sejauh mana model yang telah dibuat dapat memenuhi akurasi hasil evaluasi hasil penelitian [10]. Hasil akurasi teks opini menggunakan algoritma *Naive Bayes* dan *Logistic Regression* disajikan pada tabel 7:

Tabel 7 Evaluasi Hasil Akurasi

No	Algoritma	Akurasi(%)	<i>Precision</i> (%)	<i>Recall</i> (%)
1	<i>Naive Bayes</i>	77.0	75.4	77.6
2	<i>Logistic Regression</i>	79.0	74.7	77.3
	Selisih	2.0	0.7	0.3

Berdasarkan tabel no 7 evaluasi dapat disimpulkan bahwa nilai *performansi* akurasi model algoritma *Naive Bayes* dengan nilai akurasi 77% memiliki akurasi lebih pendek 2% dibandingkan dengan algoritma *Logistic Regression* dengan nilai lebih baik yaitu 79% dalam konfigurasi *dataset* dengan 61596 data ulasan. Berikut merupakan gambar tabel *matrix confusion* yang dapat memberikan gambaran tentang seberapa baik model dari algoritma *Naive Bayes* dan *Logistic Regression*.



Gambar 6. Tabel Matrix Confusion Algoritma Naive Bayes.

Berdasarkan percobaan, hasil pengujian dari 12.314 data dengan algoritma *Naive Bayes* pada tabel *Confusion Matrix* ditunjukkan pada tabel 8.

Tabel 8 Accuracy Naive Bayes

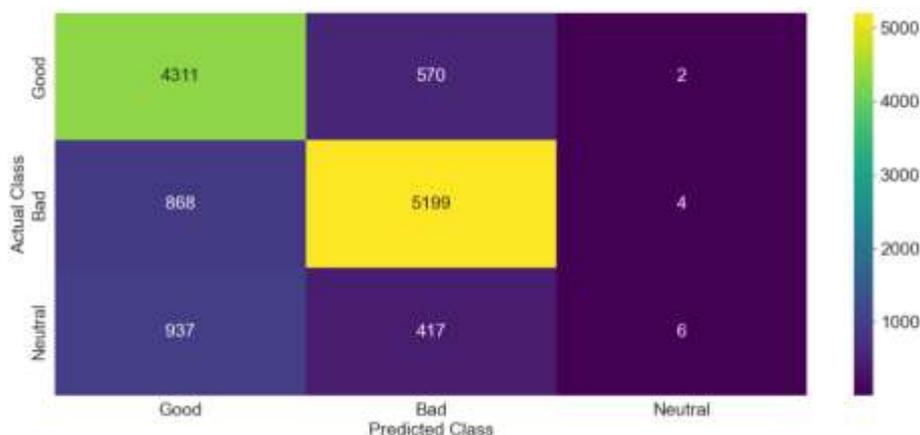
Accuracy 77%		
	TP	TN
Prediksi Good	4311	572
Prediksi Bad	5199	872
Prediksi Neutral	6	6

Berdasarkan tabel *Accuracy Naive* terlihat bahwa jumlah TP sebanyak 9.516 data opini dan TN sebanyak 1.450 opini. Berikut 872 pernyataan TN dan 572 FP hasilnya disajikan pada tabel Hasil Perhitungan *Naive Bayes*

Tabel 9 Hasil Perhitungan Naive Bayes

Dalam Persen (%)	
Accuracy	43.00
Sensitivity	88.00
Specificity	14.00
Npg	86.00
Npb	85.00
Npn	50.00

Berikut adalah tabel *Confusion Matrix* hasil pengujian data testing menggunakan *Logistic Regression* yang dapat memberikan gambaran tentang seberapa baik model dari algoritma *Logistic Regression*.



Gambar 7. Tabel Confusion Matrix Algoritma Logistic Regression

Tabel 10 Accuracy Logistic Regression

Accuracy 79 %		
	TP	TN
Prediksi Good	4309	1494
Prediksi Bad	5320	969
Prediksi Neutral	62	160

Berdasarkan tabel 10 Accuracy Logistic Regression terlihat bahwa jumlah TP sebanyak 9.691 data opini dan TN sebanyak 2623 opini. Berikut 969 pernyataan TN dan 1494 FP hasilnya disajikan pada tabel berikut:

Tabel 11 Hasil Perhitungan Logistic Regression

Dalam Persen (%)	
Accuracy	42.00
Sensitivity	74.00
Specificity	15.00
Npg	74.00
Npb	84.00
Npn	72.00

Setelah *rating spotify review* dibagi menjadi kategori *good*, *bad*, *neutral* sehingga ditampilkan nilai akurasinya, maka dapat diuji tingkat akurasinya untuk melihat performa dari hasil pengujian data diatas, hasil akhir membuktikan jika nilai akurasi algoritma *Logistic Regression* lebih tinggi daripada algoritma *Naive Bayes*.

4. SIMPULAN

Implementasi NLP (*Natural Language Processing*) Dasar pada Analisis *Sentiment Review Spotify* yang menggunakan algoritma *Naive Bayes* dan juga *Logistic Regression* mendapatkan hasil terbaik dengan menggunakan algoritma *Logistic Regression* dikarenakan pada algoritma *Logistic Regression* ini, algoritma melakukan proses pencarian bobot dengan melakukan perhitungan probabilitas pada data dan mendapatkan hasil nilai akurasi sebesar 79% sedangkan pada algoritma *Naive Bayes* memiliki nilai akurasi sebesar 76% jumlah data yang digunakan dalam program ini yaitu sebanyak 61596 data ulasan *spotify*. Data *training* yang digunakan untuk penelitian ini sebesar 49255 sedangkan untuk data testing pada penelitian ini sebesar 12314 data.

5. SARAN

Saran untuk penelitian lebih lanjut yaitu untuk menutup kekurangan penelitian dan memperluas analisis *sentiment* dengan mempertimbangkan pendekatan NLP (*Natural Language Processing*) yang lebih maju dalam analisis *sentiment* untuk hasil yang mendalam dan lebih baik. Selain itu database yang dipakai dapat diubah menjadi *database realtime*, atau setiap kali *running* hasil akan selalu berbeda karena program dijalankan secara *realtime*.

DAFTAR PUSTAKA

- [1] M. D. Rhajendra and N. Trianasari, "Analisis Sentimen Ulasan Aplikasi Spotify Untuk Peningkatan Layanan Menggunakan Algoritma Naive Bayes," *eProceedings Manag.*, vol. 8, no. 5, 2021.
- [2] G. M. T. D. N. D. N. E. D. S. F. N. P. P. Biancofiore, "Aspect Based Sentiment Analysis in Music: acase study with Spotify," *SAC*, 2022, [Online]. Available: <https://dl.acm.org/doi/10.1145/3477314.3507092>
- [3] G. Ginabila and A. Fauzi, "Analisis Sentimen Terhadap Pemutar Musik Online Spotify Dengan Algoritma Naive Bayes dan Support Vector Machine," *J. Ilm. Ilk. Komput. Inform.*, vol. 6, no. 2, pp.111–122, 2023.
- [4] M. simsek Buğra Kağan kayhan, "Enhancement of Sentiment Analysis Classification Performance of Spotify," *IJANSER*, vol. Vol. 7 No., 2023, [Online]. Available: <https://as-proceeding.com/index.php/ijanser/article/view/1423>
- [5] Nadhif Sanggara Fathullah, Y. A. (2020, Februari 2). Analisis sentimen terhadap rating dan ulasan film dengan menggunakan metode klasifikasi Naive Bayes dengan fitur Lexicon-based. 4.

- [6] Jeremy Andre Septian, T. M. (2019). Analisis Sentimen Pengguna Twitter Terhadap polemik persepakbolaan indonesia menggunakan pembobotan TF-IDF dan K-Nearest Neighbor. *vol. 1*.
- [7] A. S. Rahayu, A. Fauzi, and R. Rahmat, “Komparasi Algoritma Naïve Bayes Dan Support Vector Machine (SVM) Pada Analisis Sentimen Spotify,” *J. Sist. Komput. dan Inform.*, vol. 4, no. 2, pp. 349–354, 2022.
- [8] Kelvin, J. B. (2022, Februari 2). Analisis perbandingan sentimen corona virus disease 2019 (covid19) pada Twitter menggunakan metode Logistic Regression dan Support vector machine (SVM). 5.
- [9] K. P. J. Sitompul, A. R. Pratama, and K. A. Baihaqi, “Komparasi algoritma Naive Bayes, support vectore machine dan logistic regression pada analisis sentimen pengguna aplikasi transportasi online,” *Klik-kumpulan j. ilmu komput.*, vol. 10,no. 1, pp. 27–38, 2023.
- [10] Rahmadanti, T. F. (2021). Klasifikasi Pengguna Shopee Berdasarkan Promosi Menggunakan Naive Bayes. *Generation Journal/Vol.5 No.2*, 81-90.