Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.2, April 2025, pp. 265~272
e-ISSN: 2548-6861

Implementation of SVM Algorithm to Predict Song Popularity based on Sentiment Analysis of Lyrics

Quiin Latifah Almatin Lubis [1]*, Arif Akbarul Huda [2]*
* Informatika, Universitas Amikom Yogyakarta
quiin@students.amikom.ac.id [1], arif.akbarul@amikom.ac.id [2]

Article Info

Article history:

Received 2024-11-27
Revised 2025-01-08
Accepted 2025-01-21

Keyword:

Sentiment Analysis,
Song Lyrics,
Support Vector Machine,
Popularity Prediction

ABSTRACT

Independent musicians face significant challenges in enhancing the visibility and appeal of their work amid intense competition on music streaming platforms. Although numerous studies have been conducted to analyze and predict song popularity, most of them focus on English-language songs. This creates a research gap for Indonesian-language songs, particularly in the context of predicting popularity based on lyrics. The dataset used includes 652 Indonesian songs from 2017 to 2024. The research methodology includes data pre-processing, feature extraction using TF-IDF, handling data imbalance with SMOTE, implementing SVM, and model optimization. The results show an improvement in model accuracy from 84% to 89% after parameter optimization using GridSearchCV. In the model evaluation with 5-fold cross-validation, an average accuracy of 86.19% with a standard deviation of 0.90% was obtained. Precision, Recall, and F1-score metrics for the Less Popular class are 0.98, 0.85, and 0.91; for the Moderately Popular class, 0.79, 0.95, and 0.86; and for the Very Popular class, 0.92, 0.86, and 0.89. The implementation of the model in a Streamlit application allows for the prediction of song popularity based on lyrics, providing valuable insights for musicians in choosing word choices that can potentially increase the popularity of their songs.

This is an open access article under the CC–BY-SA license.

I. PENDAHULUAN

Musik telah menjadi elemen yang sangat penting dalam kehidupan manusia, baik sebagai hiburan, ekspresi budaya, maupun media untuk menyampaikan pesan emosional. Berbagai genre musik terus berkembang dan mendapatkan perhatian dari masyarakat luas. Di era digital yang semakin berkembang ini, industri musik telah mengalami transformasi signifikan dengan hadirnya platform *streaming* musik seperti Spotify, Apple Music, Joox, Youtube, dan lainnya yang telah mengubah cara konsumen mengakses dan menikmati musik[1], memungkinkan pendengar untuk mengakses ribuan lagu dengan mudah dan kapan saja. Selain itu, keberadaan platform ini juga memberikan peluang baru bagi para pelaku industri musik, termasuk pencipta lagu, produser, dan label rekaman, untuk menganalisis data pengguna dan lagu secara mendalam demi memahami faktor-faktor yang mempengaruhi popularitas lagu[2]. Meskipun demikian, kemunculan platform streaming juga membawa tantangan baru bagi para musisi, terutama bagi musisi independen.

Salah satu tantangan bagi musisi independen adalah menciptakan karya yang sesuai dengan tren dan selera pendengar. Tantangan ini dipengaruhi oleh berbagai faktor, seperti karakteristik lagu yang meliputi genre, tempo, struktur, melodi, vokal, lirik, dan kualitas produksi, karakteristik pendengar, strategi pemasaran melalui platform *streaming* musik, media sosial, *influencer*, dan konser, serta faktor eksternal seperti tren musik terbaru, peristiwa budaya, dan keadaan ekonomi[2]. Di antara faktor-faktor tersebut, lirik memainkan peran penting, karena memiliki pengaruh signifikan terhadap daya tarik sebuah lagu[3]. Lirik yang disusun dengan kosakata yang tepat dan relevan dapat membuat sebuah lagu lebih mudah diterima dan diingat oleh pendengar[4]. Oleh karena itu, dalam proses penciptaaan lirik, musisi independen perlu mempertimbangkan pemilihan diksi yang sesuai agar dapat meningkatkan peluang kepopuleran lagu mereka di platform streaming.

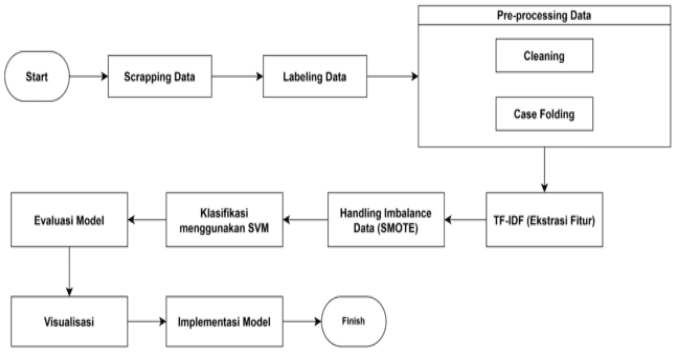
Beberapa penelitian sebelumnya telah dilakukan untuk memahami faktor-faktor yang mempengaruhi popularitas lagu, terutama melalui analisis lirik. Penelitian yang dilakukan oleh Hana Agatha dan rekan-rekannya[3] menggunakan algoritma BERT untuk menganalisis sentimen lirik lagu berbahasa Inggris dalam memprediksi popularitas lagu. Penelitian ini menghasilkan akurasi sebesar 87% setelah menggunakan metode *oversampling*. Algoritma BERT dinilai mampu menangkap konteks lirik secara mendalam, tetapi memiliki kekurangan dalam hal kebutuhan komputasi yang tinggi. Adapun penelitian lainnya[5] yang mengambil pendekatan dengan menggabungkan fitur audio dan lirik untuk klasifikasi lagu hit berbahasa inggris. Model yang memanfaatkan *audio descriptors* dan *embedding* BERT untuk lirik ini menghasilkan akurasi 76%.

Dalam penelitian ini, peneliti fokus pada lagu berbahasa Indonesia, dengan menggunakan algoritma *Support Vector Machine* (SVM) yang dapat menangani masalah analisis sentimen yang telah dibuktikan pada penelitian sebelumnya. Beberapa penelitian telah melakukan analisis sentimen menggunakan SVM dan mendapatkan hasil akurasi yang tinggi. Penelitian oleh Helen Sastypratiwi, dkk[6] memanfaatkan algoritma SVM dengan *Particle Swarm Optimization* (PSO) untuk mengklasifikasikan emosi pada lirik lagu dalam bahasa Indonesia. Model ini menunjukkan performa terbaik dengan akurasi sebesar 92,13% dalam mengklasifikasikan lima emosi dasar seperti cinta, senang, marah, takut, dan sedih. Pada penelitian[7] menggunakan algoritma SVM untuk analisis sentimen dan mendapatkan akurasi tertinggi sebesar 91,27% menggunakan metode TF-IDF, dan mendapatkan akurasi 88,59% menggunakan metode *Bags of Word* (BoW). Penelitian lainnya[8] membandingkan dua algoritma yaitu *Naïve Bayes dan* SVM, menemukan bahwa SVM lebih unggul dengan hasil akurasi sebesar 89,41% sedangkan hasil akurasi *Naïve Bayes* 82.08%.

Tujuan penelitian ini adalah untuk membangun sistem yang dapat menganalisis sentimen lirik lagu dan memanfaatkan hasil analisis tersebut untuk memprediksi kepopuleran lagu. Berbeda dengan penelitian sebelumnya, penelitian ini mempertahankan setiap kata dalam lirik tanpa menghapus kata-kata yang biasanya dianggap sebagai *stopwords*. Dalam konteks lirik lagu, setiap kata memiliki rasa dan makna yang penting, berperan sebagai elemen yang mendukung daya tarik lagu. Pendekatan ini diharapkan dapat lebih akurat menangkap nuansa diksi yang berkontribusi terhadap popularitas lagu, serta mengatasi keterbatasan pada studi terdahulu yang mungkin mengabaikan kata-kata tersebut. Penelitian ini bertujuan memberikan panduan praktis bagi musisi independen dalam memilih diksi yang tepat untuk meningkatkan potensi popularitas lagu mereka di platform digital. Fokus penelitian ini adalah lagu-lagu berbahasa Indonesia yang dirilis antara 2017 hingga 2024. Diharapkan, penelitian ini dapat menghasilkan model yang lebih efektif dalam memprediksi kepopuleran lagu dan memberikan wawasan yang berguna bagi musisi, khususnya yang independen, dalam menciptakan karya yang sesuai dengan preferensi pendengar dan tren musik yang sedang berkembang. Selain itu, penelitian ini diharapkan turut berkontribusi pada kemajuan teknologi analisis sentimen dalam industri musik.

II. METODE

Tahapan penelitian ini mengikuti beberapa langkah yang dirancang untuk memastikan bahwa proses penelitian berjalan sesuai dengan tujuan yang telah ditetapkan. Alur penelitian lengkap dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian

A. Scraping Data

Data yang digunakan dalam penelitian ini diperoleh melalui teknik *scraping*. Data *track*, *artist*, dan *popularity* diambil dari platform Spotify[9] yang menyediakan informasi lengkap tentang lagu-lagu dan popularitasnya. Sementara itu, data lirik lagu diperoleh dengan menggunakan teknik *scraping* dari situs web https://lirik.kapanlagi.com/. Peneliti menggunakan l*ibrary spotipy, requests, BeautifulSoup* dalam melakukan tahapan ini. Proses pengambilan data dilakukan secara bertahap, dengan total 652 data lagu yang diambil dari rentang waktu 2017 hingga 2024.

B. Labeling Data

Dalam penelitian ini, lagu-lagu diklasifikasikan berdasarkan indeks popularitasnya. Indeks popularitas ini memiliki rentang skor dari 0 hingga 84, yang kemudian digunakan untuk mengelompokkan lagu menjadi tiga kategori yaitu kurang populer, cukup populer, dan sangat populer. Lagu dengan skor popularitas antara 0-42 dilabeli sebagai kurang populer, lagu dengan skor antara 43-58 dilabeli sebagai cukup populer, dan lagu dengan skor antara 59-84 dilabeli sebagai sangat populer. Proses klasifikasi ini didasarkan pada analisis statistik deskriptif, di mana skor untuk label kurang populer ditentukan dari kuartil bawah, label cukup populer dari kuartil tengah, dan label sangat populer dari kuartil atas.

C. Pre-processing Data

Tahap ini bertujuan untuk membersihkan data dari *noise*, sehingga data yang telah dibersihkan siap untuk diproses lebih lanjut dan dapat menghasilkan hasil yang lebih

akurat[10]. Beberapa langkah yang dilakukan dalam tahap ini antara lain:

1) Cleaning Data

Tahap ini dilakukan dengan tujuan untuk menghilangkan karakter atau elemen yang tidak relevan dengan analisis. Proses ini mencakup penghapusan karakter yang tidak sesuai dengan aturan, termasuk tanda baca, angka, *URL*, *hashtag*, dan *username*, serta huruf atau simbol di luar rentang alfabet a-z, untuk memperoleh teks yang lebih bersih dan sesuai untuk analisis[11].

2) Case Folding

Pada tahap ini, setiap huruf dalam teks diubah menjadi huruf kecil (*lowercase*) untuk mengurangi redundansi data[12]. Proses ini menyelaraskan teks ke dalam format yang seragam, memastikan konsistensi, dan mempermudah tahap klasifikasi.

D. Ekstraksi Fitur (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) adalah metode ekstraksi fitur yang memberikan bobot tertentu pada setiap kata[13]. Metode ini berfungsi untuk mengukur seberapa signifikan sebuah kata dalam menggambarkan suatu kalimat, berdasarkan frekuensi kemunculannya. Semakin tinggi nilai TF-IDF, semakin sering kata tersebut muncul dalam dokumen, sedangkan nilai TF-IDF yang rendah menunjukkan kata tersebut jarang muncul. Perhitungan TF-IDF dilakukan menggunakan rumus 1, 2, dan 3.

$$TF_{t,d} = \frac{Jumlah\ kemunculan\ term\ t\ dalam\ dokumen\ d}{Total\ jumlah\ term\ dalam\ dokumen\ d} \quad (1)$$

$$IDF_t = \log\left(\frac{N}{df_t}\right) \quad (2)$$

$$TF-IDF_{t,d} = TF_{t,d} \times IDF_t \quad (3)$$

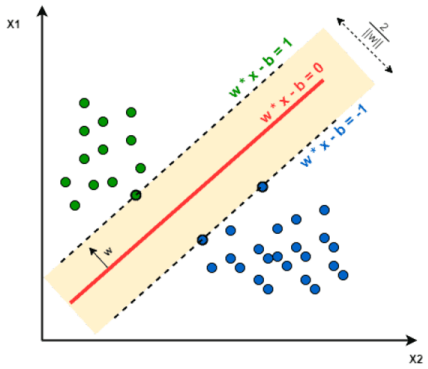
Keterangan :
TFt,d = Jumlah frekuensi kata yang muncul dalam dokumen
IDFt = Jumlah inverse frekuensi dokumen tiap kata
N = Jumlah total dokumen dalam kumpulan data
dft = Jumlah dokumen yang mengandung kata t setidaknya satu kali.

E. Handling Imbalanced data (SMOTE)

Pada penelitian ini, peneliti menerapkan metode *Synthetic Minority Over-sampling Technique* (SMOTE) untuk menangani ketidakseimbangan data. SMOTE menghasilkan sampel baru dari kelas minoritas guna menyeimbangkan distribusi data[14]. Teknik ini membuat data sintetis dengan menggabungkan informasi dari sampel yang ada dalam kelas minoritas, sehingga menciptakan distribusi data yang lebih seimbang dan dapat meningkatkan kinerja model.

F. Support Vector Machine

Support Vector Machine adalah algoritma yang umum digunakan untuk masalah klasifikasi. Algoritma ini bekerja dengan membedakan dua kelas dengan cara mencari *hyperplane* optimal yang memaksimalkan jarak (margin) antara titik data terdekat dari masing-masing kelas, sebagaimana ditunjukkan pada Gambar 2. Penggunaan SVM dalam penelitian ini didasarkan pada kemampuannya menangani data berdimensi tinggi seperti teks, kinerjanya yang konsisten pada dataset berukuran sedang, dan keefektifannya dalam klasifikasi *multiclass* melalui pendekatan *one-vs-rest* [15]. Metode *one-vs-rest* membagi masalah klasifikasi *multiclass* menjadi beberapa masalah biner.



Gambar 2. Algoritma SVM

G. Evaluasi Model

Tahap ini bertujuan untuk menilai kinerja model yang telah dibangun. Metrik evaluasi yang digunakan dalam penelitian ini adalah *confusion matrix*, yang menjadi acuan untuk mengukur hasil analisis sentimen data[16]. *Confusion matrix* memberikan gambaran mengenai seberapa baik model dalam mengklasifikasikan data secara akurat, dengan membandingkan prediksi model terhadap label yang sebenarnya. Berikut adalah rumus-rumus yang digunakan untuk menghitung *accuracy*, *precision*, *recall*, dan *f1-score* [17], [18]:
Akurasi adalah indikator yang mengukur tingkat kebenaran secara keseluruhan dari sebuah model dalam memprediksi label.

$$Accuracy = \frac{TP+TN}{(TP+FP+TN+FN)} \quad (4)$$

Presisi adalah indikator yang mengukur keberhasilan model dalam memprediksi elemen positif dan tingkat keandalannya.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Recall adalah indikator yang mengukur ketepatan model dalam memprediksi elemen positif.

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

F1-score adalah indikator yang mengukur kinerja keseluruhan model dengan menghitung rata-rata harmonis antara presisi dan recall.

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

H. Visualisasi

Pada tahap visualisasi, peneliti menerapkan *word cloud* sebagai alat untuk menampilkan kata-kata yang paling sering muncul[19]. Visualisasi ini memberikan gambaran yang lebih jelas tentang pola penggunaan kata, sehingga memudahkan analisis kecenderungan diksi yang secara signifikan berkontribusi terhadap popularitas lagu.

I. Implementasi Model

Peneliti menggunakan Streamlit untuk mengimplementasikan model dalam aplikasi berbasis web. Streamlit adalah *framework* Python yang memungkinkan pembuatan antarmuka pengguna yang interaktif dan mudah digunakan tanpa memerlukan pengkodean frontend yang kompleks[20], [21].

III. HASIL DAN PEMBAHASAN

A. Hasil Scraping Data

Pada Gambar 3, terlihat hasil *scraping* data yang diperoleh dari platform Spotify dan situs web *https://lirik.kapanlagi.com/*. Data yang dikumpulkan terdiri dari 652 lagu dari rentang waktu 2017 hingga 2024. Kolom yang diperoleh meliputi: *track* (judul lagu), *artist* (nama artis), *popularity* (jumlah pemutaran dan frekuensi pemutaran terbaru), dan *lyrics* (lirik lagu).

	track	artist	popularity	lyrics
0	Untungnya, Hidup Harus Tetap Berjalan	Bernadya	84	Persis setahun yang lalu\nKu dijauhkan dari ya...
1	Kata Mereka Ini Berlebihan	Bernadya	81	Ku tak pernah ikat rambutku lagi semenjak kaub...
2	Lama-Lama	Bernadya	81	Berusaha tetap terjaga\nTunggu kamu selesaikan...
3	Kini Mereka Tahu	Bernadya	79	Dari dulu kulebih-lebihkan semua\nPadahal yang...
4	Mati-Matian	Mahalini	78	Mahalini baru saaja merilis lagu terbarunya ya...

Gambar 3. Hasil *Scraping*

B. Hasil Labeling Data

Proses pelabelan dilakukan dengan menggunakan skor popularitas yang dihitung melalui analisis statistik deskriptif yang dapat dilihat pada Gambar 4.

	popularity
count	652.000000
mean	51.558282
std	11.505745
min	1.000000
25%	43.000000
50%	49.000000
75%	59.000000
max	84.000000

Gambar 4. Statistik Deskriptif

Batas kategori ditentukan berdasarkan kuartil data, dengan kuartil bawah pada skor 43, kuartil tengah pada skor 49, dan kuartil atas pada skor 59. Berdasarkan skor tersebut, lagu-lagu dikelompokkan ke dalam tiga kategori:

- Kurang Populer dengan skor popularitas 0–42.
- Lumayan Populer dengan skor popularitas 43–58.
- Sangat Populer dengan skor popularitas 59–84.

Hasil dari proses *labeling* ditampilkan pada Gambar 5.

	track	artist	popularity	lyrics	sentiment
0	Untungnya, Hidup Harus Tetap Berjalan	Bernadya	84	Persis setahun yang lalu\nKu dijauhkan dari ya...	sangat populer
1	Kata Mereka Ini Berlebihan	Bernadya	81	Ku tak pernah ikat rambutku lagi semenjak kaub...	sangat populer
2	Lama-Lama	Bernadya	81	Berusaha tetap terjaga\nTunggu kamu selesaikan...	sangat populer
3	Kini Mereka Tahu	Bernadya	79	Dari dulu kulebih-lebihkan semua\nPadahal yang...	sangat populer
4	Mati-Matian	Mahalini	78	Mahalini baru saaja merilis lagu terbarunya ya...	sangat populer

Gambar 5. Hasil *Labeling*

C. Hasil Pre-processing Data

Proses *pre-processing* bertujuan untuk membersihkan teks lirik dari elemen-elemen yang tidak diperlukan dan menyelaraskan format data agar siap untuk analisis lebih lanjut. Proses ini sangat penting karena kualitas data yang bersih dan konsisten akan mempengaruhi efektivitas dan akurasi model saat dilatih.

Contoh hasil *pre-processing* pada lirik lagu dapat dilihat pada Tabel I. Awalnya, teks lirik berisi informasi tambahan seperti deskripsi lagu, tanggal rilis, dan sumber platform. Setelah dilakukan *cleaning*, hanya bagian lirik lagu yang relevan yang dipertahankan. Selanjutnya, pada tahap *case folding*, teks lirik diubah menjadi huruf kecil secara keseluruhan.

TABEL I
HASIL PRE-PROCESSING

Tanpa Pre-processing	Mahalini baru saaja merilis lagu terbarunya yang berjudul Mati Matian pada 29 Maret 2024. Lagu tersebut dapat disaksikan di Channel Youtube HITS Records. Yuk simak lirik lagunya! VERSE 1 Kita adalah dua insan penuh cinta Di awal tercipta kisah kita Manis tuturmu buatku terpana Bagiku kau sempurna
Cleaning	Kita adalah dua insan penuh cinta Di awal tercipta kisah kita Manis tuturmu buatku terpana Bagiku kau sempurna
Case Folding	kita adalah dua insan penuh cinta di awal tercipta kisah kita manis tuturmu buatku terpana bagiku kau sempurna

D. Hasil TF-IDF

Dalam penelitian ini, digunakan pembobotan TF-IDF untuk mengubah teks menjadi data numerik. TF-IDF (*Term Frequency-Inverse Document Frequency*) mengukur pentingnya suatu kata dalam dokumen. Skor TF-IDF dihitung berdasarkan frekuensi kemunculan kata dalam dokumen (TF) dan seberapa unik kata tersebut muncul di seluruh kumpulan

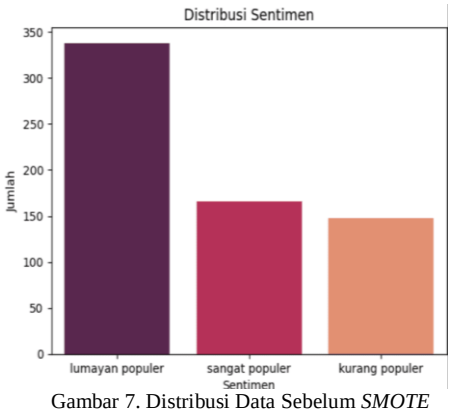
dokumen (IDF). Pada Gambar 6, (0, 6206) mengacu pada dokumen ke-0 dan indeks kata ke-6206, sementara nilai 0.0705 adalah skor TF-IDF yang menunjukkan seberapa penting kata tersebut dalam dokumen. Nilai TF-IDF yang tinggi menunjukkan kata-kata yang sangat khas dan berkontribusi pada lirik lagu.

```
(0, 6206)    0.07052761191200128
(0, 7253)    0.0673489120340327
(0, 8610)    0.12840842333069474
(0, 3864)    0.03742671782007375
(0, 3493)    0.08246216634879858
(0, 1604)    0.07500773776301332
(0, 1391)    0.02922879249559284
(0, 7589)    0.04108216346726508
(0, 1787)    0.0673489120340327
(0, 8407)    0.04247360122820692
(0, 3599)    0.06488332085802766
(0, 6686)    0.028631384849752372
(0, 2632)    0.059535784182186 27
(0, 3622)    0.06488332085802766
(0, 2274)    0.02598451764831235
(0, 4811)    0.07500773776301332
(0, 201)     0.07052761191200128
(0, 3951)    0.07176487843262785
(0, 6124)    0.040605417698042344
(0, 1205)    0.07500773776301332
(0, 6264)    0.16052008696673448
(0, 6426)    0.06286878618302066
(0, 6558)    0.06116552138455878
(0, 6797)    0.049565669400066642
(0, 4151)    0.06286878618302066
:    :
(651, 3932)  0.0692158189847196
(651, 508)   0.0692158189847196
```

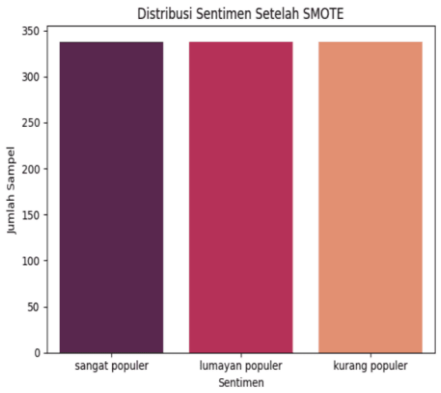
Gambar 6. Hasil TF-IDF

E. Hasil SMOTE

Dalam penelitian ini, digunakan teknik SMOTE (*Synthetic Minority Over-sampling Technique*) untuk mengatasi masalah ketidakseimbangan kelas pada data. Teknik ini bertujuan untuk meningkatkan jumlah sampel dari kelas minoritas dengan cara mensintesis sampel baru berdasarkan sampel yang sudah ada. Sebelum penerapan SMOTE, data memiliki distribusi yang tidak merata, di mana kelas sangat populer hanya memiliki 166 sampel, dan kelas kurang populer sebanyak 148 sampel, sementara kelas lumayan populer memiliki 338 sampel. Setelah SMOTE diterapkan, jumlah sampel pada setiap kelas menjadi seimbang, yaitu 338 sampel per kelas, sebagaimana diperlihatkan pada Gambar 7 dan 8. Langkah ini dilakukan untuk membantu model dalam menangani ketidakseimbangan kelas sehingga kinerjanya lebih optimal.



Gambar 7. Distribusi Data Sebelum *SMOTE*



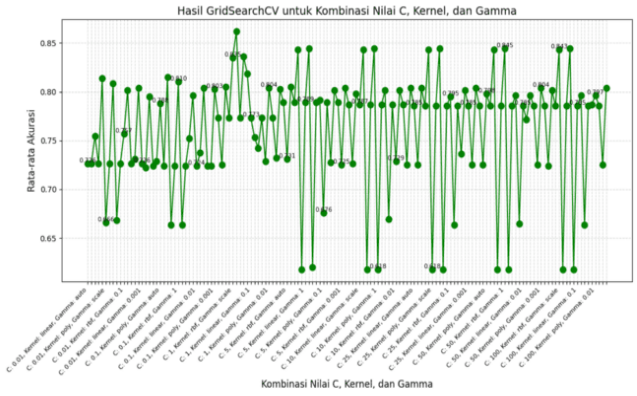
Gambar 8. Distribusi Data Setelah *SMOTE*

F. Support Vector Machine

Setelah dilakukan proses *handling imbalanced* data, dataset dibagi menjadi dua bagian, yaitu 80% untuk data latih sebanyak 811 data dan 20% untuk data uji sebanyak 203 data. Model klasifikasi yang digunakan adalah SVM dengan pendekatan *One-vs-Rest Classifier* untuk menangani klasifikasi multi-kelas. Untuk meningkatkan performa model, dilakukan *hyperparameter tuning* menggunakan *GridSearchCV* dengan 5-fold *cross validation.* Proses ini bertujuan menemukan kombinasi parameter terbaik dengan menguji beberapa nilai berikut:

- C: 0.01, 0.1, 1, 5, 10, 25, 50, 100
- Kernel: linear, rbf, poly
- Gamma*: auto, scale*, 0.001, 0.01, 0.1, 1

Hasil optimasi menunjukkan bahwa kombinasi parameter terbaik diperoleh pada C=1, *kernel=poly*, dan *gamma=scale*. Setelah model dioptimasi menggunakan parameter tersebut, akurasi model meningkat dari 84% menjadi 89% pada data uji. Selanjutnya, dilakukan validasi silang (*cross-validation*) sebanyak 5-fold untuk mengevaluasi kestabilan performa model. Hasil *cross validation* menunjukkan bahwa model memiliki rata-rata akurasi sebesar 86,19% dengan standar deviasi 0,90%. Hal ini mengindikasikan bahwa performa model stabil dan konsisten di berbagai lipatan validasi, serta membuktikan bahwa SVM mampu melakukan klasifikasi teks secara efektif. Visualisasi hasil optimasi *hyperparameter tuning* menggunakan *GridSearchCV* ditampilkan Gambar 9.



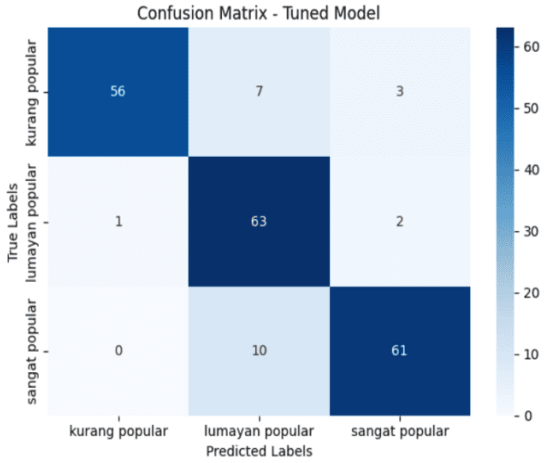
Gambar 9. Grafik Hasil *GridSearchCV*

G. Evaluasi Model

Evaluasi model dilakukan untuk mengukur performa algoritma SVM sebelum dan setelah proses optimasi menggunakan *hyperparameter tuning*. Hasil evaluasi menunjukkan bahwa akurasi model mengalami peningkatan sebesar 5%, dari 84% menjadi 89% setelah *tuning*. Perbandingan performa model ditampilkan pada Tabel II, yang mencakup metrik *precision*, *recall*, dan, *f1-score* untuk setiap kelas. Visualisasi *confussion matrix* hasil optimasi dapat dilihat pada Gambar 10.

TABEL II
PERBANDINGAN HASIL SEBELUM DAN SETELAH *HYPERPARAMETER TUNING*

Tahap Evaluasi	Kelas	*Precision*	*Recall*	*F1-Score*	*Acc*
Sebelum Tuning	Kurang Populer	0.98	0.76	0.85	
	Lumayan Populer	0.71	0.97	0.82	
	Sangat Populer	0.92	0.80	0.86	
	Rata-rata	0.87	0.84	0.84	**0.84**
Setelah Tuning	Kurang Populer	0.98	0.85	0.91	
	Lumayan Populer	0.79	0.95	0.86	
	Sangat Populer	0.92	0.86	0.89	
	Rata-rata	0.90	0.89	0.89	**0.89**



Gambar 10. *Confussion Matrix*

H. Visualisasi

Pada tahap ini, dilakukan visualisasi untuk menampilkan kata-kata yang paling sering muncul menggunakan *Word Cloud*. Hasil visualisasi untuk setiap kategori dapat dilihat pada Gambar 11-13. Visualisasi ini menggambarkan perbedaan kata-kata yang digunakan pada setiap tingkat popularitas. Hal ini dapat memberikan wawasan yang berguna bagi musisi independen dalam memilih diksi yang tepat untuk meningkatkan peluang kesuksesan lagu mereka.



Gambar 11. *WordCloud* Lirik Sangat Populer



Gambar 12. *WordCloud* Lirik Lumayan Populer

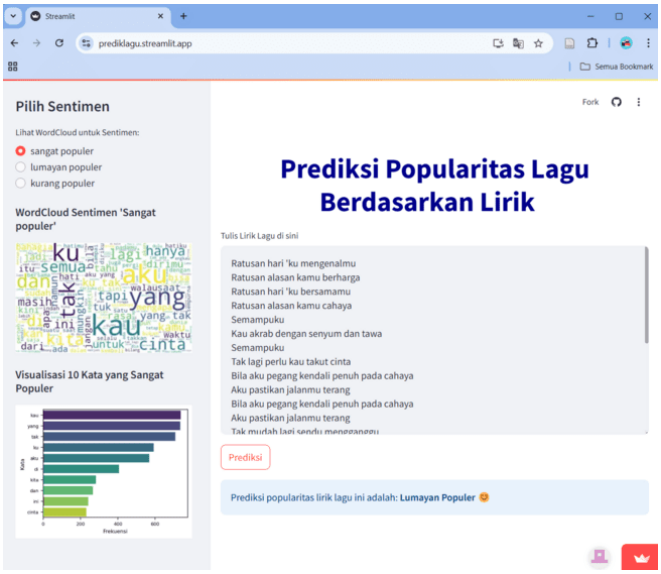


Gambar 13. *WordCloud* Lirik Kurang Populer

I. Implementasi Model

Untuk implementasi model peneliti menggunakan Streamlit untuk membangun aplikasi web yang dapat memprediksi popularitas lagu berdasarkan liriknya. Aplikasi ini memungkinkan pengguna untuk menginput lirik lagu, yang kemudian dianalisis dan diprediksi tingkat popularitasnya menggunakan model SVM yang telah dilatih. Selain itu, aplikasi ini dilengkapi dengan sidebar untuk melihat *word cloud* berdasarkan sentimen dan visualisasi 10 kata yang sangat populer. Hasil prediksi popularitas lagu ini dapat dilihat dengan memilih sentimen yang sesuai dan kemudian memprediksi popularitasnya berdasarkan lirik yang

diinput oleh pengguna. Hasil dari implementasi ini dapat dilihat pada Gambar 14.



Gambar 14. Hasil Implementasi Model

IV. KESIMPULAN

Penelitian ini telah berhasil mengembangkan sistem prediksi kepopuleran lagu berbahasa Indonesia menggunakan algoritma *Support Vector Machine* (SVM) berdasarkan analisis sentimen lirik. Serangkaian metode, termasuk *pre-processing* data, ekstraksi fitur dengan TF-IDF, penanganan ketidakseimbangan data menggunakan SMOTE, dan optimasi model, telah menghasilkan peningkatan akurasi dari 84% menjadi 89%. Hasil ini menunjukkan bahwa pemilihan kata dalam lirik memiliki korelasi yang signifikan dengan tingkat popularitas lagu, memberikan wawasan berharga bagi para pelaku industri musik, khususnya musisi independen dalam proses penciptaan karya mereka. Selain itu, visualisasi *word cloud* yang dihasilkan membantu mengidentifikasi pola diksi yang berbeda di antara kategori popularitas lagu, menegaskan pentingnya penggunaan kata yang tepat untuk meningkatkan daya tarik lagu.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan. Pertama, fokus yang hanya pada analisis lirik tidak memperhitungkan elemen penting lainnya seperti genre musik, tempo, struktur lagu, melodi, dan kualitas produksi, yang juga mempengaruhi popularitas lagu. Kedua, dataset yang terbatas pada lagu-lagu dari tahun 2017 hingga 2024 mungkin belum sepenuhnya merepresentasikan perubahan preferensi musik yang dinamis. Untuk penelitian lebih lanjut, disarankan untuk menggabungkan analisis fitur audio dengan analisis lirik, memperluas cakupan dataset, dan mempertimbangkan faktor kontekstual seperti tren musik serta pengaruh media sosial. Selain itu, eksplorasi model pembelajaran mesin lanjutan seperti *deep learning* berpotensi meningkatkan performa prediksi secara signifikan. Penelitian ini diharapkan dapat menjadi landasan bagi pengembangan studi lanjutan sekaligus memberikan kontribusi praktis bagi industri musik.

DAFTAR PUSTAKA

[1] Kompasiana.com, "Analisis Tren Musik di Platform Streaming," KOMPASIANA. Accessed: Nov. 06, 2024. [Online]. Available: https://www.kompasiana.com/juliarni53946/66908c53c925c472f5369972/analisis-tren-musik-di-platform-streaming

[2] S. Marlia, K. Setiawan, and C. Juliane, "Analysis of Music Features and Song Popularity Trends on Spotify Using K-Means and CRISP-DM," *SISTEMASI*, vol. 13, no. 2, p. 595, Mar. 2024, doi: 10.32520/stmsi.v13i2.3757.

[3] H. Agatha, F. P. Putri, and A. Suryadibrata, "Sentiment Analysis on Song Lyrics for Song Popularity Prediction Using BERT Algorithm".

[4] R. A. Nender, P. Rumengan, and G. Latuni, "Struktur Musik Lagu-Lagu Koes Plus Dan Pengaruhnya Terhadap Popularitas Dan Kelestariannya," *Kompetensi*, vol. 1, no. 01, pp. 228–237, Dec. 2022, doi: 10.53682/kompetensi.v1i01.1803.

[5] I. Sharma, "Hit Song Classification With Audio Descriptors And Lyrics".

[6] H. Sastypratiwi, H. Muhardi, and M. Noveanto, "Klasifikasi Emosi Pada Lirik Lagu Menggunakan Algoritma Multiclass SVM dengan Tuning Hyperparameter PSO," *J. MEDIA Inform. BUDIDARMA*, vol. 6, no. 4, p. 2279, Oct. 2022, doi: 10.30865/mib.v6i4.4609.

[7] D. D. Nur Cahyo *et al.*, "Sentiment Analysis for IMDb Movie Review Using Support Vector Machine (SVM) Method," *Inf. J. Ilm. Bid. Teknol. Inf. Dan Komun.*, vol. 8, no. 2, pp. 90–95, Mar. 2023, doi: 10.25139/inform.v8i2.5700.

[8] B. Rakajati and E. Y. Hidayat, "Perbandingan Metode Naive Bayes dan Support Vector Machine Pada Klasifikasi 22 Bahasa Daerah," *J. MEDIA Inform. BUDIDARMA*, vol. 8, no. 1, p. 221, Jan. 2024, doi: 10.30865/mib.v8i1.7236.

[9] "Home | Spotify for Developers." Accessed: Nov. 12, 2024. [Online]. Available: https://developer.spotify.com/

[10] G. Cahyani, W. Widayani, S. D. Anggita, Y. Pristyanto, I. Ikmah, and A. Sidauruk, "Klasifikasi Data Review IMDb Berdasarkan Analisis Sentimen Menggunakan Algoritma Support Vector Machine," *J. MEDIA Inform. BUDIDARMA*, vol. 6, no. 3, p. 1418, Jul. 2022, doi: 10.30865/mib.v6i3.4023.

[11] D. Oktavia and Y. R. Ramadahan, "Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (SVM)".

[12] T. Safitri, Y. Umaidah, and I. Maulana, "Analisis Sentimen Pengguna Twitter Terhadap Grup Musik BTS Menggunakan Algoritma Support Vector Machine", JAIC, vol. 7, no. 1, pp. 34–41, Jul. 2023.

[13] E. Harieby, H. Hoiriyah, and M. Walid, "Twitter Text Mining Mengenai Isu Vaksinasi Covid-19 Menggunakan Metode Term Frequency, Inverse Document Frequency (TF-IDF)," *JATI J. Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 532–537, Aug. 2022, doi: 10.36040/jati.v6i2.5129.

[14] Hermanto, A. Y. Kuntoro, T. Asra, E. B. Pratama, L. Effendi, and R. Ocanitra, "Gojek and Grab User Sentiment Analysis on Google Play Using Naive Bayes Algorithm And Support Vector Machine Based Smote Technique," *J. Phys. Conf. Ser.*, vol. 1641, no. 1, p. 012102, Nov. 2020, doi: 10.1088/1742-6596/1641/1/012102.

[15] R. N. Ikhsani and F. F. Abdulloh, "Optimasi SVM dan Decision Tree Menggunakan SMOTE Untuk Mengklasifikasi Sentimen Masyarakat Mengenai Pinjaman Online," *J. MEDIA Inform. BUDIDARMA*, vol. 7, no. 4, p. 1667, Oct. 2023, doi: 10.30865/mib.v7i4.6809.

[16] R. M. R. W. P. K. Atmaja and W. Yustanti, "Analisis Sentimen Customer Review Aplikasi Ruang Guru Dengan Metode BERT (Bidirectional Encoder Representations from Transformers)," *J. Emerg. Inf. Syst. Bus. Intell. JEISBI*, vol. 2, no. 3, Jul. 2021, Accessed: Nov. 14, 2024. [Online]. Available: https://ejournal.unesa.ac.id/index.php/JEISBI/article/view/41567

[17] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," vol. 5, 2021.

[18] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," Aug. 13, 2020, *arXiv*: arXiv:2008.05756. Accessed: Nov. 14, 2024. [Online]. Available: http://arxiv.org/abs/2008.05756

[19] J. Setyanto and T. B. Sasongko, "Sentiment Analysis of Sirekap Application Users Using the Support Vector Machine Algorithm," *J. Appl. Inform. Comput.*, vol. 8, no. 1, pp. 71–76, Jul. 2024, doi: 10.30871/jaic.v8i1.7772.

[20] S. Patil and V. Lokesha, "Live Twitter Sentiment Analysis Using Streamlit Framework," *SSRN Electron. J.*, 2022, doi: 10.2139/ssrn.4119949.

[21] A. Nasrulloh and G. Nahumarury, "Development of a Web-Based Automatic Sentiment Analysis Application using Support Vector Machine (SVM) Model".