Vol. 10, No. 1, April 2025, Page. 143-155
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Analisis Performa Naive Bayes dan SVM terhadap Sentimen Teks Media Sosial dengan Word2Vec dan SMOTE

Juliandri Saputra[1], Lily Maryani[2], Rahmaddeni*[3], Denok Wulandari[4], Wisnu Eka[5]

[1,2,3,5]Prodi Teknik Informatika, Universitas Sains dan Teknologi Indonesia Pekanbaru
[4]Prodi Teknik Komputer, Institut Az Zuhra Pekanbaru
Email: [1]2310031802145@sar.ac.id, [2]2310031802146@sar.ac.id, [3]rahmaddeni@usti.ac.id,
[4]denokwulandari18@gmail.com, [5]2310031802062@sar.ac.id

Abstrak

Penelitian ini membandingkan performa algoritma Naive Bayes dan Support Vector Machine (SVM) dalam klasifikasi sentimen teks dari media sosial. Dataset berisi 736 unggahan dari Facebook, Instagram, dan Twitter yang telah dilabeli sebagai positif, netral, atau negatif. Proses prapemrosesan mencakup pembersihan teks, normalisasi, tokenisasi, penghapusan kata umum, dan *stemming*. Fitur diekstraksi menggunakan Word2Vec, sedangkan ketidakseimbangan kelas diatasi dengan Synthetic Minority Oversampling Technique (SMOTE). Model dilatih dan dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score, serta divalidasi melalui *K-Fold Cross-Validation*. Hasil menunjukkan bahwa algoritma SVM mencapai akurasi 88,85% dan F1-score 88,86%, lebih unggul dibandingkan Naive Bayes dengan akurasi 72,64% dan F1-score 72,26%. SVM juga menunjukkan konsistensi dalam memprediksi sentimen netral, yang menjadi kelemahan Naive Bayes. Temuan ini memperkuat posisi SVM sebagai algoritma yang lebih efektif untuk analisis sentimen teks media sosial.

Kata kunci: analisis sentimen, media sosial, naive bayes, support vector machine (SVM), Word2Vec, SMOTE.

Abstract

This paper presents a comparative analysis of the Naive Bayes and Support Vector Machine (SVM) algorithms for sentiment classification of social media text. The dataset comprises 736 labeled posts collected from Facebook, Instagram, and Twitter, categorized into positive, neutral, and negative sentiments. Preprocessing steps include text cleaning, normalization, tokenization, stop-word removal, and stemming. Feature extraction is performed using Word2Vec, and class imbalance is addressed using the Synthetic Minority Oversampling Technique (SMOTE). The models are evaluated using accuracy, precision, recall, and F1-score metrics, and validated through 5-fold cross-validation. Experimental results indicate that SVM outperforms Naive Bayes, achieving an accuracy of 88.85% and an F1-score of 88.86%, compared to Naive Bayes with 72.64% accuracy and 72.26% F1-score. Moreover, SVM demonstrates greater consistency in predicting neutral sentiments. These results highlight the superior effectiveness of SVM in handling sentiment classification tasks on unstructured social media data.

Keywords: *sentiment analysis, social media, Naive Bayes, support vector machine (SVM), Word2Vec, SMOTE.*.

1. PENDAHULUAN

Media sosial kala ini telah jadi platform utama untuk warga dalam menyatakan suatu opini, komentar, pemikiran serta ulasan dari bermacam topik. Kemajuan teknologi data telah meningkatkan pemakaian media sosial secara signifikan. Di situs media sosial seperti Facebook, Instagram dan Twitter, pengguna dapat menuangkan pemikiran dan ilham mereka tentang bermacam topik serta berbagi data seperti konten buatan pengguna. Aktivitas pengguna di media sosial menghasilkan data dalam jumlah besar yang memiliki potensi tinggi untuk dianalisis secara komputasional. Data ini dapat dimanfaatkan untuk memahami opini dan sentimen masyarakat terhadap berbagai topik atau isu publik[1][2].

Analisis sentimen adalah proses mengungkap sikap, perasaan, atau pendapat orang terhadap suatu subjek. Entitas ini dapat berupa orang, aktivitas, atau topik. Sikap dan pendapat ini biasanya dikategorikan sebagai positif, negatif atau netral. Dengan lebih dari 7.000 makalah yang diterbitkan, penelitian tentang emosi adalah salah satu bidang penelitian paling populer dalam ilmu komputer[3].

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Analisis sentimen adalah bagian dari *Natural Language Processing* (NLP) dan digunakan untuk mengidentifikasi konten dalam kumpulan data dalam bentuk opini atau pandangan tekstual (sentimen) tentang topik atau peristiwa positif, negatif atau netral[4].

Ada banyak algoritma pembelajaran mesin yang tersedia untuk melakukan analisis sentimen, seperti Menggunakan Naive Bayes Multinomial, Gradient Boosting, Random Forest, Support Vector Machine (SVM), Adaboost, dan Ensemble Voting untuk memeriksa konsistensi akurasi[5]. Namun, teknik pembelajaran mesin yang paling umum digunakan untuk sentimen analisis adalah algoritma Naive Bayes dan Support Vector Machine (SVM)[1]. Sementara SVM memiliki keunggulan dalam mencapai akurasi tinggi melalui teknik margin maksimum, Naive Bayes dikenal karena kemudahan implementasi dan efisiensinya dalam menangani data teks[4][6]. Namun, karakteristik data yang digunakan dapat memengaruhi kinerja kedua teknik ini.

Analisis sentimen dari teks media sosial merupakan hal yang menantang dan memakan waktu karena sifat data yang tersedia tidak terstruktur dan beragamnya varian bahasa yang mencakup emotikon dan akronim. Tantangan utama dalam analisis teks di media sosial terletak pada sifat data yang tidak terstruktur, penggunaan bahasa informal yang bervariasi, serta keberadaan emotikon, singkatan, dan slang yang memperumit proses pemrosesan bahasa alami[7][8][2]. Selain itu, ketidakseimbangan kelas dalam dataset sering menyebabkan model pembelajaran mesin cenderung bias terhadap kelas mayoritas, sehingga memerlukan teknik khusus untuk menanganinya seperti SMOTE[9]. Oleh karena itu, diperlukan metode yang andal untuk memastikan apakah suatu teks positif, negatif, atau netral. Pada sentimen analisis, metode Naive Bayes dan Support Vector Machine (SVM) dikenal luas karena popularitasnya. Klasifikasi Naive Bayes mengandalkan teorema Bayes dan mengasumsikan bahwa setiap karakteristik bersifat independen, sehingga menghasilkan model probabilistik dan statistik yang berkurang. Sifat kualitas yang berbentuk bebas mengurangi keakuratan, tetapi proses pembuatannya cepat dan tidak menimbulkan kesulitan, jadi ini merupakan nilai tambah[6]. SVM, di sisi lain, terbukti memberikan hasil yang akurat baik pada data linier maupun non-linier, terutama saat digunakan untuk klasifikasi teks dari media sosial[3][8]. Faktanya, kelebihan dan kekurangan masing-masing algoritme berbeda-beda, bergantung pada properti data dan kebutuhan analisis.

Berbagai penelitian telah dilakukan untuk menganalisis sentimen publik dengan menggunakan informasi data dari berbagai topik permasalahan, misalnya penelitian tentang sentimen terhadap ulasan game[10], media pembelajaran[11], instansi pemerintah[7][12], kebijakan pemerintah[13][14], penggunaan mobil listrik[8], prediksi kelulusan[15][16], penyakit[17][18], fashion[19], bahkan hingga analisis sentimen komentar di playstore[2][20]. Sebuah penelitian yang membandingkan kinerja metode Support Vector Machine (SVM) dengan algoritma Naive Bayes dalam analisis sentimen komentar TikTok tentang barang perawatan kulit menunjukkan keunggulan algoritma SVM. Untuk mengatasi permasalahan data, penelitian ini menggunakan Synthetic Minority Over-sampling Technique (SMOTE). Berdasarkan hasil evaluasi, SVM mengungguli Naive Bayes yang mencapai akurasi 47,65% dan 59,43%. Selain itu, kinerja SVM lebih baik dalam hal skor F1, dengan nilai 60,37% dibandingkan 54,74%. Namun, Naive Bayes mengungguli SVM dengan selisih kecil dalam hal presisi, dengan nilai 67,96% dibandingkan dengan 62,76% untuk SVM[3].

Penelitian ini bertujuan untuk menganalisis dan mengevaluasi performa algoritma Naive Bayes dan Support Vector Machine (SVM) dalam klasifikasi sentimen teks media sosial. Berbeda dengan penelitian sebelumnya yang hanya membandingkan performa algoritma berdasarkan akurasi umum, penelitian ini mengintegrasikan pendekatan optimasi berbasis Word2Vec untuk pembobotan fitur dan SMOTE untuk penyeimbangan kelas. Kombinasi ini memungkinkan model belajar representasi kata yang lebih kontekstual sekaligus mengurangi bias pada kelas mayoritas. Selain itu, penelitian ini juga secara khusus mengevaluasi kemampuan model dalam memprediksi kelas netral, yang sering diabaikan dalam studi analisis sentimen, sehingga memberikan kontribusi baru dalam mengevaluasi performa algoritma secara lebih menyeluruh dan mendalam.

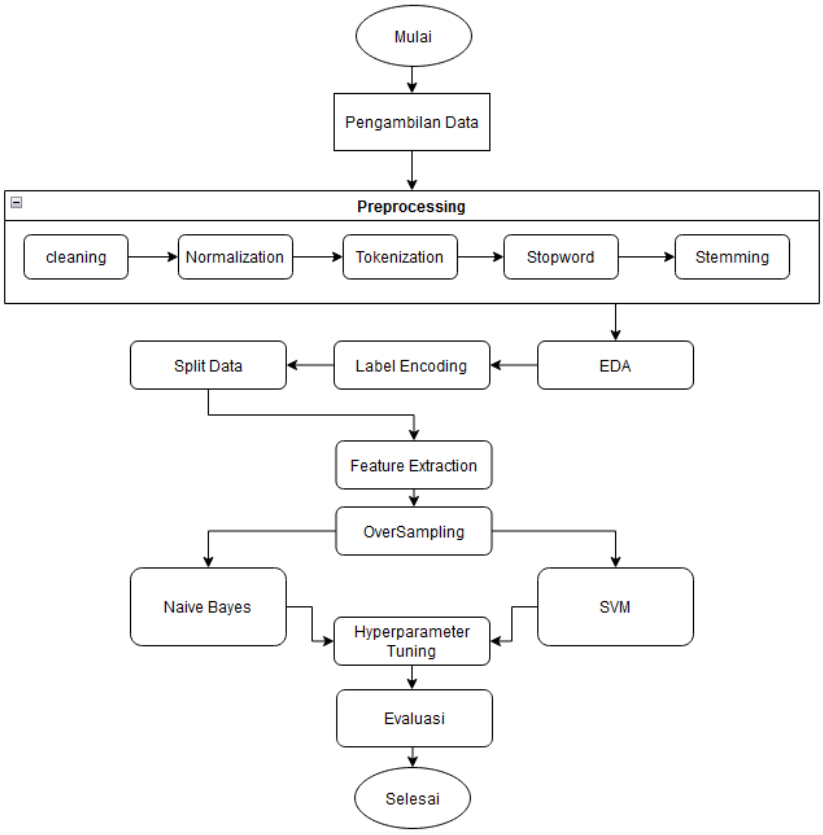
2. METODE PENELITIAN

Analisis masalah, arsitektur, dan rencana strategi pemecahan masalah adalah contoh metodologi penelitian[3][4]. Diagram alir digunakan untuk menggambarkan proses penelitian, dan setiap langkah proses dijelaskan dalam kaitannya dengan penelitian yang dilakukan. Permasalahan yang timbul dan

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

diperbaiki dalam penelitian ini diuraikan dengan menggunakan analisis masalah. Desainnya menguraikan solusi terhadap masalah tersebut dan harus dijelaskan secara lengkap dalam bentuk diagram. Contohnya termasuk diagram desain perangkat keras dan diagram pengiriman data, yang menunjukkan bagaimana data mentah diubah menjadi data lengkap.

Ada beberapa tahapan proses dalam penelitian ini. tahap awal yaitu pengumpulan data. Kaggle menyediakan data yang digunakan dalam penelitian ini, yang diunduh. Setelah pengumpulan data, langkah prapemrosesan akan dilakukan dengan tujuan mengubah data yang belum diproses ke dalam format yang dapat dimengerti. Setelah prapemrosesan, teknik Naive Bayes dan Support Vector Machine (SVM) akan digunakan untuk memproses dan mengklasifikasikan data. Setelah prosedur klasifikasi, tahap evaluasi akan menyusul, di mana metrik seperti akurasi, presisi, perolehan, dan waktu pemrosesan akan diukur. Gambar 1 di bawah mengilustrasikan langkah-langkah yang terlibat dalam tahapan ini.



Gambar 1. Alur tahapan penelitian

Berdasarkan gambar 1 diatas, maka alur tahapan penelitian diatas dapat dijelaskan sebagai berikut.

2.1. Pengambilan Data

Data yang digunakan pada penelitian ini adalah data yang bersifat open *dataset* bersumber dari Kaggle. Proses pengambilan data yang dilakukan adalah dengan cara mengunduh langsung dari situs Kaggle[21]. *Dataset* dari Kaggle dipilih, karena menyediakan data unggahan dari media sosial populer seperti Facebook, Instagram, dan Twitter yang representatif untuk analisis sentimen dengan menggunakan bahasa Inggris. Selain itu, *dataset* ini mencerminkan keragaman pola sentimen yang sangat cocok untuk dilakukan pengujian algoritma pembelajaran mesin. *Dataset* inilah yang selanjutnya akan digunakan untuk dilakuan perbandingan performa menggunakan algoritma Naïve Bayes dan Support Vector Machine (SVM).

2.2. *Preprocessing*

Langkah pertama yang penting dalam pemrosesan data teks adalah pra-pemrosesan, yaitu proses mengubah data mentah menjadi data yang siap digunakan untuk pemodelan dan analisis lanjutan seperti

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

melakukan perbandingan performa terhadap algoritma. Tahap pra-pemrosesan ini akan diawali dengan melalui berbagai prosedur.

2.2.1. *Cleaning*

Pada titik ini, konten disaring untuk menghilangkan elemen asing termasuk URL, sebutan, hashtag, simbol khusus, angka, dan tanda baca yang berlebihan. Tujuan dari langkah ini adalah untuk jaminan bahwa hanya data relevan yang tersisa untuk prosedur berikut.

2.2.2. *Normalization*

Semua huruf teks kini telah diubah menjadi huruf kecil. Misalnya, untuk menghilangkan perbedaan yang tidak perlu, kata "Hebat" dan "hebat" diubah ke dalam bentuk yang sama.

2.2.3. *Tokenization*

Langkah ini diperlukan untuk membagi teks menjadi komponen-komponen terpisah yang dikenal sebagai token, yang sering kali merupakan kata-kata berbeda. Ungkapan "Saya suka produk ini" diubah menjadi "Saya", "suka", "ini", dan "produk", misalnya.

2.2.4. *Stop Words Removal*

Pada tahapan ini, dilakukan penghapusan kata-kata umum yang sering muncul tetapi tidak membawa makna signifikan dalam analisis, seperti "and," "the," atau "is.".

2.2.5. *Stemming*

Stemming digunakan pada saat ini untuk menghilangkan kata-kata sesuai dengan kriteria dasar, misalnya, "berjalan" diubah menjadi "berjalan". Untuk meminimalkan perbedaan antar kata yang memiliki akar kata yang sama, setiap kata diubah ke bentuk dasarnya.

2.3. *Exploratory Data Analysis* (EDA)

Verifikasi identitas data yang dibersihkan dilakukan pada saat ini. Prosedur ini membantu dalam melihat kesalahan yang nyata, memahami pola data, menemukan *outlier* atau kejadian ganjil, dan mengidentifikasi korelasi yang menarik antar variabel.

2.4. *Label Encoding*

Dataset yang digunakan dalam penelitian ini sudah dilengkapi dengan label sentimen bawaan dari pembuat dataset, yang terbagi ke dalam tiga kategori: positif, netral, dan negatif. Label ini digunakan langsung sebagai target kelas dalam proses pelatihan model yang digunakan tanpa perlu dilakukan pelabelan ulang. Keberadaan label yang telah tersedia memungkinkan model dilatih dan divalidasi menggunakan pendekatan supervised learning secara optimal.

2.5. *Feature Extraction* (Word2Vec)

Pada tahap ini, dilakukan proses pemetaan setiap kata dalam teks ke dalam vector menggunakan teknik Word2Vec. Word2vec merepresentasikan kata ke dalam vektor yang dapat membawa makna semantik dari kata tersebut[22]. Word2Vec digunakan karena mampu mengubah kata menjadi angka dengan cara yang menangkap makna dan hubungan antar kata. Teknik ini lebih unggul dibandingkan metode sederhana seperti CountVectorizer atau TF-IDF yang hanya menghitung jumlah kata tanpa mempertimbangkan konteks. Word2Vec sangat cocok untuk data teks media sosial yang penuh variasi.

2.6. *Data Splitting*

Pada tahap ini, data akan dipisahkan menjadi dua kategori, yakni data latih sebagai data pelatihan model. dan model ini akan dievaluasi pada data baru menggunakan data uji. Delapan puluh persen data akan digunakan untuk pelatihan, dan dua puluh persen akan digunakan untuk pengujian..

2.7. *Synthetic Minority Oversampling Technique* (SMOTE)

Pada tahapan ini, dilakukan proses penyeimbangan kelas dengan menggunakan metode *Synthetic Minority Oversampling Technique* (SMOTE)[23]. Metode SMOTE bekerja dengan cara membuat data baru berdasarkan data yang ada untuk kelas yang jumlahnya lebih sedikit. Teknik ini dipilih karena

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

dapat membantu menyeimbangkan jumlah data dengan membuat data baru yang mirip dengan data yang lebih sedikit, bukan hanya menggandakan data yang ada seperti cara biasa. Ini mencegah masalah seperti model yang terlalu fokus pada data yang sedikit dan menghindari penghilangan informasi penting seperti yang terjadi pada cara mengurangi data dari kelompok mayoritas. Dibandingkan dengan metode lain seperti *Generative Adversarial Networks* (GAN) yang lebih rumit dan membutuhkan waktu lama, SMOTE lebih mudah dan cepat diterapkan. Dengan begitu, program pembelajaran mesin bisa lebih mudah mengenali pola dari data yang jarang muncul tanpa mengurangi akurasi pada data yang lebih banyak.

2.8. Pemodelan

Setelah melakukan penyeimbangan kelas, selanjutnya dilakukan tahapan pemodelan, pada penelitian ini digunakan dua model algoritma, yaitu algoritma Naive Bayes dan Support Vector Machine (SVM).

2.8.1. Naive Bayes

Teorema probabilitas Bayes menjadi dasar kategorisasi pembelajaran mesin yang dikenal sebagai Naive Bayes. Setiap fitur dalam kumpulan data diasumsikan independen agar algoritma ini dapat berfungsi. Dengan menilai setiap probabilitas dalam kelas tertentu, Naive Bayes tetap dapat menghasilkan prediksi yang akurat meskipun asumsinya sederhana. Algoritma ini menggunakan prinsip probabilitas untuk menentukan klasifikasi dengan cara memperkirakan kemungkinan setiap kelas berdasarkan informasi yang ada, kemudian memilih kelas dengan probabilitas tertinggi sebagai hasil prediksi akhir. Kategorisasi teks, sentimen analisis, dan aplikasi lain dimana asumsi independensi fitur dapat berhasil sering kali menggunakan teknik ini[23].

$$P(C|X) = \frac{P(X|C) - P(C)}{P(X)} \quad (1)$$

Dimana :
$P(C|X)$ = Probabilitas posterior, yaitu probabilitas kelas *C* diberikan data *X*
$P(X|C)$ = Probabilitas likelihood, yaitu probabilitas data *X* diberikan kelas *C*
$P(C)$ = Probabilitas prior, yaitu probabilitas awal kelas *C* sebelum melihat data
$P(X)$ = Probabilitas evidence, yaitu probabilitas data *X* secara keseluruhan

Jika fitur $X = \{x_1, x_2, \dots, x_n\}$ diasumsikan independen, maka:

$$P(C|X) \propto P(C) \,.\, \prod_{k=1}^{n} P\,(\,xi \mid C\,) \quad (2)$$

Jika fitur kontinu, probabilitas $P(xi|C)$ dihitung menggunakan fungsi distribusi probabilitas, seperti distribusi Gaussian:

$$P(\,x_i \mid C\,) = \frac{1}{\sqrt{2\pi\sigma_C^2}} exp\left(-\frac{(x_i - \mu c)}{2\sigma_C^2}\right) \quad (3)$$

Dimana :
μ_c = rata-rata (mean) fitur x_i untuk kelas *C*
σ_C^2 = variansi fitur x_i untuk kelas *C*

2.8.2 Support Vector Machine (SVM).

Pendekatan pembelajaran mesin yang disebut Support Vector Machine (SVM) membagi data menjadi beberapa wilayah yang sesuai dengan setiap kelas menggunakan hyperplanes. Pendekatan ini banyak digunakan untuk tugas klasifikasi teks dan sering digunakan untuk kumpulan data besar, terutama yang bersumber dari internet. Membangun hyperplane dengan batas simetris dan menghindari terlalu dekat dengan salah satu kelas adalah ide dasar SVM. Untuk menjamin batas ideal antar kelas,

prosedur ini dilakukan dengan mengukur margin dan mencari titik maksimal. Tujuan utama metode SVM adalah menentukan hyperplane yang optimal untuk digunakan sebagai pemisah kelas[4].

$$f(x) = w^T x + b = 0 \quad (4)$$

Dimana :
w = vektor bobot (normal ke hyperplane)
x = vektor data (fitur)
b = bias (intersep dari hyperplane)
$f(x) =$ fungsi keputusan untuk menentukan kelas

2.9. *Hyperparameter tuning* menggunakan GridSearchCV

Proses pengaturan *hyperparameter* bertujuan untuk meningkatkan kinerja model pembelajaran mesin dengan menemukan nilai *hyperparameter* yang optimal. metode ini akan mencoba setiap kombinasi nilai satu per satu sampai menemukan yang nilai terbaik. Untuk mencapai performa model terbaik pada penelitian ini, digunakan GridSearchCV untuk melakukan proses tuning *hyperparameter* pada algoritma Naive Bayes dan Support Vector Machine (SVM)[24].

2.10. Evaluasi Model

Tujuan tahap evaluasi adalah untuk menentukan seberapa baik model memprediksi sentimen. Berbagai metrik seperti akurasi, presisi, recall, dan F1-Score juga menggunakan *Confusion Matrix*, digunakan untuk mengevaluasi prediksi kinerja.*Cross-validation* juga dilakukan pada tahap evaluasi ini untuk memperjelas performa model dan memberikan gambaran yang lebih lengkap. Pendekatan *K-Fold Cross-validation* dengan parameter cv=5 akan digunakan dalam penelitian ini, artinya data pelatihan akan dibagi menjadi lima bagian (fold). Empat bagian model akan dilatih, dan sisanya akan diuji. Proses ini akan dilakukan sebanyak lima kali, dengan setiap titik data berfungsi sebagai data uji satu kali.

Tabel 1. *Confusion Matrix*

Aktual	**Prediksi**		
	Positive	Neutral	Negative
Positive	TP_{pos}	$FN_{pos,neu}$	$FN_{pos,neg}$
Neutral	$FN_{neu,pos}$	TP_{neu}	$FN_{neu,neg}$
Negative	$FN_{neg,pos}$	$FN_{neg,neu}$	TP_{neg}

Keterangan :
TP_{pos} = jumlah data yang benar-benar Positive dan diprediksi Positive
TP_{neu} = jumlah data yang benar-benar Neutral dan diprediksi Neutral
TP_{neg} = jumlah data yang benar-benar Negative dan diprediksi Negative
$FN_{pos,neu}$ = data yang benar-benar Positive, tetapi diprediksi Neutral
$FN_{pos,neg}$ = data yang benar-benar Positive, tetapi diprediksi Negative
$FN_{neu,pos}$ = data yang benar-benar Neutral, tetapi diprediksi Positive
$FN_{neu,neg}$ = data yang benar-benar Neutral, tetapi diprediksi Negative
$FN_{neg,pos}$ = data yang benar-benar Negative, tetapi diprediksi Positive
$FN_{neg,neu}$ = data yang benar-benar Negative, tetapi diprediksi Neutral

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan yang telah dilakukan akan disajikan lebih lanjut pada bab hasil dan pembahasan ini, berdasarkan uraian yang telah diberikan pada tahapan penelitian di atas. Penelitian ini menggunakan Visual Studio Code sebagai alat bantu, dengan Python dan berbagai library pendukung lainnya untuk mendukung proses analisis data.

Vol. 10, No. 1, April 2025, Page. 143-155
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

3.1 *Dataset*

Dataset diambil melalui hasil unduh langsung dari situs kaggle[21]. Data yang tersedia merupakan kalimat teks yang berasal dari tiga media sosial ternama, yaitu Facebook, Instagram dan Twitter. Terdapat data kalimat sebanyak 736 baris data teks dalam bahasa inggris, hal ini dapat dilihat pada gambar 2 dibawah.

	Unnamed: 0.1	Unnamed: 0	Text	Sentiment	Timestamp	User	Platform	Hashtags	Retweets	Likes	Country	Year	Month	Day	Hour
0	0	0	Enjoying a beautiful day at the park! ...	Positive	2023-01-15 12:30:00	User123	Twitter	#Nature #Park	15.0	30.0	USA	2023	1	15	12
1	1	1	Traffic was terrible this morning. ...	Negative	2023-01-15 08:45:00	CommuterX	Twitter	#Traffic #Morning	5.0	10.0	Canada	2023	1	15	8
2	2	2	Just finished an amazing workout! 💪 ...	Positive	2023-01-15 15:45:00	FitnessFan	Instagram	#Fitness #Workout	20.0	40.0	USA	2023	1	15	15
3	3	3	Excited about the upcoming weekend getaway! ...	Positive	2023-01-15 18:20:00	AdventureX	Facebook	#Travel #Adventure	8.0	15.0	UK	2023	1	15	18
4	4	4	Trying out a new recipe for dinner tonight. ...	Neutral	2023-01-15 19:55:00	ChefCook	Instagram	#Cooking #Food	12.0	25.0	Australia	2023	1	15	19

Gambar 2. *Dataset* teks media sosial

Gambar 2 menampilkan data mentah dari dataset media sosial yang digunakan dalam penelitian ini. Setiap baris mewakili unggahan asli dari platform seperti Facebook, Instagram, dan Twitter, yang berisi teks serta label sentimen awal (positif, netral, atau negatif) yang disediakan oleh pembuat dataset. Data ini selanjutnya akan diproses melalui tahapan pra-pemrosesan sebelum digunakan untuk pelatihan model.

3.2 *Pre-processing*

Cleaning, *normalization*, *tokenization*, *stopwords*, dan *label encoding* adalah langkah lanjutan dalam proses ini. Data teks akan dibersihkan menjadi data yang siap untuk digunakan pada langkah selanjutnya. Tabel 2 di bawah menunjukkan hasil dari langkah prapemrosesan ini.

Tabel 2. Hasil proses *Pre-Processing*

Teks Awal	Hasil Preprocessing	Sentiment
Just finished an amazing workout! 💪	finish amazing workout	positive
Just adopted a cute furry friend! 🐾	adopt a cute furri friend	positive
Celebrating a milestone at work! 🎉	celebr mileston work	positive

3.3 *Exploratory Data Analysis* (EDA)

Setelah data dibersihkan, selanjutnya dilakukan proses identifikasi pada data, tujuan utama dari proses ini adalah untuk melihat serta memastikan bahwa data teks yang akan diproses sudah sesuai untuk digunakan dalam analisis lanjutan. Dari proses ini didapatkanlah beberapa informasi mengenai data yang selanjutnya akan digunakan untuk analisis lebih lanjut kedalam model. Proses ini dapat dilihat pada gambar 3 dibawah.

```
RangeIndex: 732 entries, 0 to 731
Data columns (total 16 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0.1  732 non-null    int64
 1   Unnamed: 0    732 non-null    int64
 2   Text          732 non-null    object
 3   Sentiment     732 non-null    object
 4   Timestamp     732 non-null    object
 5   User          732 non-null    object
 6   Platform      732 non-null    object
 7   Hashtags      732 non-null    object
 8   Retweets      732 non-null    float64
 9   Likes         732 non-null    float64
 10  Country       732 non-null    object
 11  Year          732 non-null    int64
 12  Month         732 non-null    int64
 13  Day           732 non-null    int64
 14  Hour          732 non-null    int64
 15  Clean_Text    732 non-null    object
dtypes: float64(2), int64(6), object(8)
memory usage: 91.6+ KB
None
```

Gambar 3. Hasil dari *Exploratory Data Analysis* (EDA)

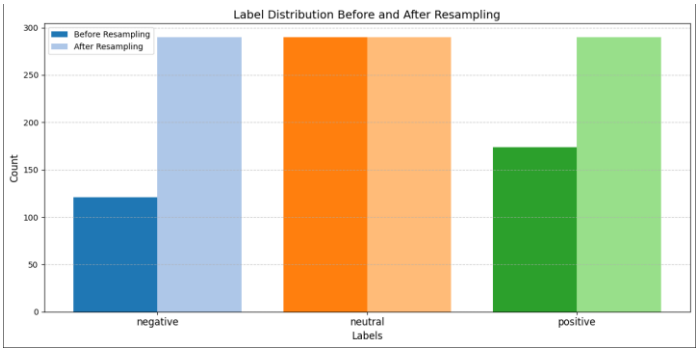
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Dari gambar 3 dapat diambil informasi :

a. *Dataset* memiliki 732 baris dan 16 kolom.
b. Terdapat 3 tipe data yang digunakan dalam setiap kolom, yaitu int46, float46 dan Object.
c. Tidak ada nilai yang hilang didalam *dataset.*
d. Ukuran *dataset* menggunakan memori yang relatif kecil yaitu 91.6 KB.

3.4 Pembobotan fitur dan optimasi keseimbangan data

Selanjutnya dilakukan pembobotan fitur dengan teknik Word2vec untuk menangkap informasi penting yang ada didalam teks. Hasil dari teknik ini kemudian diseimbangkan dengan teknik SMOTE atau Random *Oversampling*, jika SMOTE gagal dilakukan, program akan secara otomatis menggunakan teknik *Random Oversampling* untuk menangani ketidakseimbangan kelas yang ada didalam *dataset*. Perbedaan keseimbangan data antara sebelum dan sesudah dilakukan optimasi dapat kita lihat pada gambar 4 dibawah ini.



Gambar 4. Sebelum dan sesudah dilakukan optimasi

Terlihat pada gambar 4 bahwa setelah dilakukan SMOTE, jumlah data pada kelas minoritas meningkat sehingga distribusi kelas menjadi lebih seimbang. Langkah ini penting untuk mengurangi bias model terhadap kelas mayoritas saat proses pelatihan dilakukan.

3.5 Pemodelan dan *Hyperparameter Tuning*

Setelah tahapan *preprocessing* dan optimasi sampling selesai dilakukan, tahap selanjutnya adalah tahapan pemodelan, dalam penelitian ini, kami melakukan perbandingan performa terhadap dua model algoritma yaitu *Naive Bayes dan Support Vector Machine (SVM).*

```
# Optimized SVM with GridSearchCV
param_grid = {
    'C': [0.1, 1, 10, 100],
    'kernel': ['linear', 'rbf'],
    'gamma': [1, 0.1, 0.01, 0.001]
}
grid = GridSearchCV(SVC(random_state=42), param_grid, refit=True, verbose=2, cv=5)
grid.fit(X_train_w2v_smote, y_train_smote)

best_svm = grid.best_estimator_
y_pred_svm = best_svm.predict(X_test_w2v)
accuracy_svm = accuracy_score(y_test, y_pred_svm)
precision_svm = precision_score(y_test, y_pred_svm, average='weighted')
recall_svm = recall_score(y_test, y_pred_svm, average='weighted')
f1_svm = f1_score(y_test, y_pred_svm, average='weighted')
classification_rep_svm = classification_report(y_test, y_pred_svm)
```

Terlihat pada potongan kode diatas, bahwa model Naive Bayes menggunakan GaussianNB serta dioptimalkan menggunakan var_smoothing dengan gridsearchCV.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

```
# Optimized Naive Bayes (GaussianNB) with var_smoothing
param_grid_nb = {'var_smoothing': np.logspace(-9, -1, 10)}
grid_nb = GridSearchCV(GaussianNB(), param_grid_nb, cv=5, scoring='accuracy', verbose=2)
grid_nb.fit(X_train_w2v_smote, y_train_smote)

best_nb = grid_nb.best_estimator_
y_pred_nb = best_nb.predict(X_test_w2v)
accuracy_nb = accuracy_score(y_test, y_pred_nb)
precision_nb = precision_score(y_test, y_pred_nb, average='weighted')
recall_nb = recall_score(y_test, y_pred_nb, average='weighted')
f1_nb = f1_score(y_test, y_pred_nb, average='weighted')
classification_rep_nb = classification_report(y_test, y_pred_nb)
```

Sementara pada model SVM menggunakan Support Vector Classifier (SVC) dan dioptimalkan menggunakan parameter kernel (linear dan rbf), nilai regulasi(c) dan gamma melalui GridsearchCV.

```
[CV] END .................var_smoothing=0.012915496650148827; total time=   0.0s
[CV] END .................var_smoothing=0.012915496650148827; total time=   0.0s
[CV] END ..................................var_smoothing=0.1; total time=   0.0s
[CV] END ..................................var_smoothing=0.1; total time=   0.0s
[CV] END ..................................var_smoothing=0.1; total time=   0.0s
[CV] END ..................................var_smoothing=0.1; total time=   0.0s
[CV] END ..................................var_smoothing=0.1; total time=   0.0s

Optimized Naive Bayes Results:
Best Parameters: {'var_smoothing': 0.012915496650148827}
```

Gambar 5. Hasil optimasi model Naive Bayes

Dari gambar 5 dapat dilihat bahwa parameter terbaik yang ditemukan oleh GridSearchCV untuk Naive Bayes adalah var_smoothing dengan nilai 0.01291.

```
[CV] END ..................C=100, gamma=0.001, kernel=linear; total time=   0.0s
[CV] END ..................C=100, gamma=0.001, kernel=linear; total time=   0.0s
[CV] END ......................C=100, gamma=0.001, kernel=rbf; total time=   0.0s
[CV] END ......................C=100, gamma=0.001, kernel=rbf; total time=   0.0s
[CV] END ......................C=100, gamma=0.001, kernel=rbf; total time=   0.0s
[CV] END ......................C=100, gamma=0.001, kernel=rbf; total time=   0.0s
[CV] END ......................C=100, gamma=0.001, kernel=rbf; total time=   0.0s
Support Vector Machine Results:
Best Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
```

Gambar 6. Hasil optimasi model SVM

Sementara pada gambar 6 dapat dilihat bahwa parameter terbaik yang ditemukan oleh GridSearchCV untuk SVM adalah c=10, gamma=1, kernel=rbf.

3.6 Perbandingan sebelum dan sesudah penggunaan SMOTE

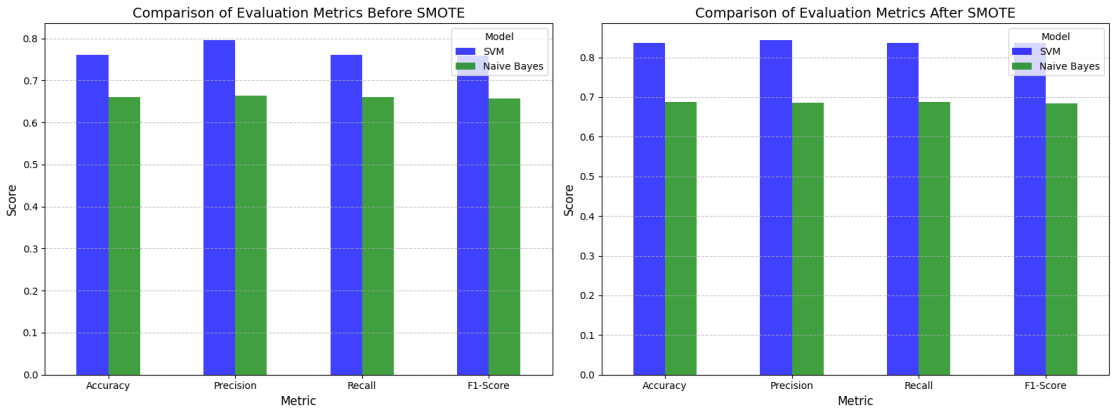
Setelah melakukan percobaan perbandingan antara sebelum dan sesudah menggunakan metode *Synthetic Minority Oversampling Technique* (SMOTE), didapatkan perbandingan nilai metrik seperti yang ditampilkan pada Tabel 3.

Tabel 3. Hasil perbandingan dengan dan tanpa menggunakan SMOTE

	Metric	Tanpa SMOTE		Dengan SMOTE	
		SVM	Naive Bayes	SVM	Naive Bayes
1	Accuracy	0.761905	0.659864	0.836735	0.687075
2	Precision	0.796037	0.664149	0.843248	0.685726
3	Recall	0.761905	0.659864	0.836735	0.687075
4	F1-Score	0.759988	0.657527	0.836070	0.683394

Dari Tabel 3 dapat dilihat bahwa distribusi data sangat mempengaruhi performa algoritma[9], karena algoritma memerlukan data yang representatif untuk semua kelas. Ketidakseimbangan data dapat

menyebabkan bias pada kelas mayoritas, sehingga performa pada kelas minoritas menurun. Teknik seperti SMOTE membantu memperbaiki ketidakseimbangan ini, memberikan algoritma kesempatan yang lebih adil untuk mempelajari pola dari semua kelas.

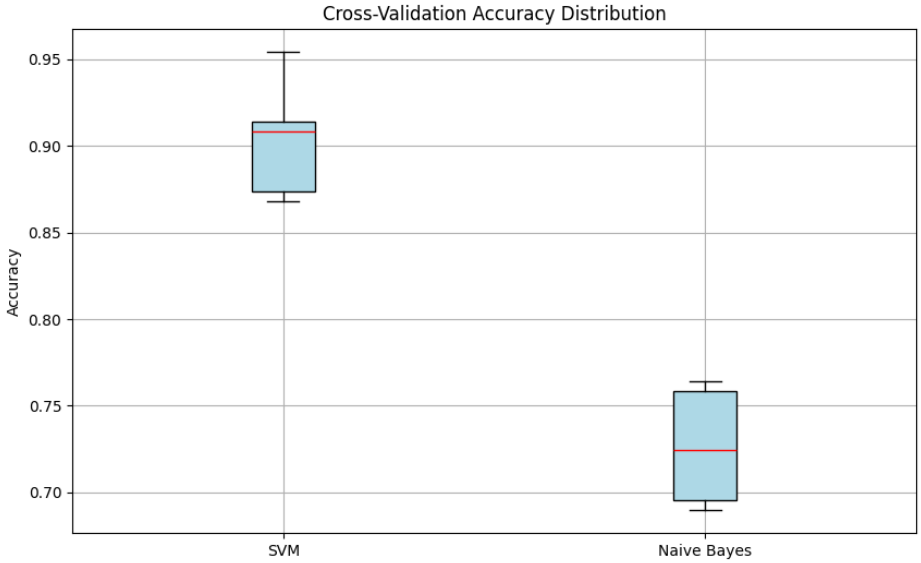


Gambar 7. Visualisasi perbandingan sebelum dan sesudah penggunaan SMOTE

Hasilnya, performa model, terutama dalam hal metrik seperti *precision, recall, dan F1-score*, akan meningkat secara signifikan seperti yang terlihat pada Gambar 7.

3.7 Evaluasi Model

Evaluasi terhadap kedua model dilakukan melalui dua tahapan, yaitu dengan mengukur akurasi dengan *Cross-validation* yang divisualisasikan kedalam bentuk boxplot, kemudian barulah dilakukan evaluasi menggunakan *Confusion Matrix*. *Cross-validation* dilakukan terlebih dahulu untuk memastikan bahwa model memiliki akurasi yang baik sebelum dilanjutkan ke evaluasi menggunakan *Confusion Matrix*.



Gambar 8. Boxplot *Cross-Validation*

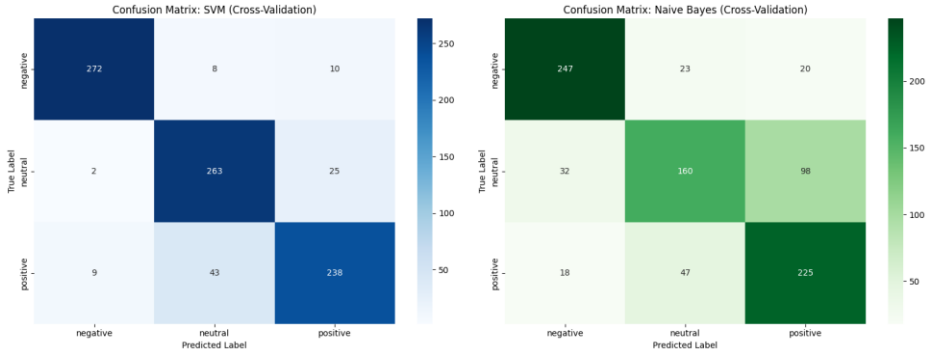
Dari Gambar 8 dapat dilihat bahwa model SVM memiliki median akurasi yang tinggi dengan hasil nilai 0.90 dibandingkan Naive Bayes yang mendapatkan hasil nilai 0.75. Hal ini menunjukkan bahwa performa SVM lebih stabil dari Naive Bayes selama proses *Cross-Validation*.

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Tabel 4. Hasil skor evaluasi metrik

	Metric	SVM	Naive Bayes
1	Accuracy	0.888506	0.726437
2	Precision	0.890168	0.727760
3	Recall	0.888506	0.726437
4	F1-Score	0.888574	0.722617

Dari Tabel 4. diatas dapat dilihat lebih spesifik mengenai evaluasi model berdasarkan 4 metrik utama, Berdasarkan seluruh metrik evaluasi yang terlihat pada tabel 3, performa SVM secara signifikan lebih baik dibandingkan Naive Bayes dalam klasifikasi *dataset* ini.



Gambar 9. *Confusion Matrix*

Berdasarkan Gambar 9, terlihat bahwa algoritma SVM mampu memberikan hasil yang lebih baik dalam mengelompokkan data ke dalam kategori sentimen positif, netral, dan negatif. SVM juga lebih tepat dalam mengenali sentimen netral, yang sering kali sulit dibedakan karena tidak terlalu jelas arahnya. Sementara itu, algoritma Naive Bayes tampak lebih sering keliru saat memprediksi sentimen netral. Hal ini bisa terjadi karena cara kerja Naive Bayes lebih sederhana dan tidak mempertimbangkan hubungan antar kata secara mendalam.

Penggunaan Word2Vec dalam penelitian ini membantu model mengenali arti kata berdasarkan konteks di sekitarnya, sehingga model bisa memahami kalimat dengan lebih baik. Selain itu, teknik SMOTE yang digunakan untuk menyeimbangkan jumlah data pada setiap kategori sentimen terbukti membantu meningkatkan hasil prediksi, terutama untuk kategori yang awalnya jumlah datanya sedikit. Secara keseluruhan, kombinasi SVM, Word2Vec, dan SMOTE memberikan hasil yang lebih baik dibandingkan dengan metode Naive Bayes dalam tugas mengklasifikasikan sentimen dari teks media sosial.

4. KESIMPULAN

SVM terbukti lebih unggul dibandingkan Naive Bayes dalam akurasi dan konsistensi prediksi sentimen, terutama untuk kelas netral. Penggunaan Word2Vec mampu menangkap makna semantik antar kata, sementara SMOTE efektif dalam mengatasi ketidakseimbangan kelas. Dengan kombinasi ini, SVM menunjukkan performa yang lebih stabil dan akurat. Penelitian lanjutan disarankan menggunakan pelabelan manual multi-annotator untuk meningkatkan kualitas ground truth dan mengeksplorasi pendekatan deep learning untuk dataset yang lebih besar..

DAFTAR PUSTAKA

[1] H. Sujadi, "Analisis Sentimen Pengguna Media Sosial Twitter Terhadap Wabah Covid-19 Dengan Metode Naive Bayes Classifier Dan Support Vector Machine," *INFOTECH J.*, vol. 8, no. 1, pp. 22–27, 2022, doi: 10.31949/infotech.v8i1.1883.

[2] R. I. Syah, H. Hoiriyah, and M. Walid, "Analisis Sentimen Pengguna Media Sosial Terhadap Aplikasi M-Health Peduli Lindungi Dengan Metode Lexicon Based Dan Naïve Bayes," *Indones. J. Bus. Intell.*, vol. 6, no. 1, 2023, doi: 10.21927/ijubi.v6i1.3275.

[3] T. Setiawan, S. Liem, and D. M. R. Pribadi, "Perbandingan Algoritma SVM dan Naïve Bayes dalam Analisis Sentimen Komentar Tiktok pada Produk Skincare," *Appl. Inf. Technol. Comput. Sci.*, vol. 3, no. 2, pp. 28–32, 2024, [Online]. Available: https://jurnal.politap.ac.id/index.php/aicoms

[4] M. I. Fikri, T. S. Sabrila, and Y. Azhar, "Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter," *Smatika J.*, vol. 10, no. 02, pp. 71–76, 2020, doi: 10.32664/smatika.v10i02.455.

[5] A. Fauzi and A. H. Yunial, "Analisis Sentimen Pada Media Sosial Menggunakan Perbandingan Algoritma Data Mining," *J. Edukasi dan Penelit. Inform.*, vol. 10, no. 2, p. 277, 2024, doi: 10.26418/jp.v10i2.76024.

[6] Gishella Septania Al-Husna, Dian Asmarajati, Iman Ahmad Ihsannuddin, and Rina Mahmudati, "Perbandingan Metode Naïve Bayes Dan Support Vector Machine Untuk Analisis Sentimen Pada Ulasan Pengguna Aplikasi Linkedin," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 3, no. 2, pp. 139–144, 2024, doi: 10.55123/storage.v3i2.3602.

[7] U. Khaira, R. Aryani, and R. W. Hardian, "Komparasi Algoritma Naïve Bayes Dan Support Vector Machine (SVM) Pada Analisis Sentimen Kebijakan Kemdikbudristek Mengenai Kuota Internet Selama Covid-19," *J. Process.*, vol. 18, no. 2, pp. 183–191, 2023, doi: 10.33998/processor.2023.18.2.897.

[8] W. Ningsih, B. Alfianda, R. Rahmaddeni, and D. Wulandari, "Perbandingan Algoritma SVM dan Naïve Bayes dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 556–562, 2024, doi: 10.57152/malcom.v4i2.1253.

[9] R. Ridwan, E. H. Hermaliani, and M. Ernawati, "Penerapan: Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada Klasifikasi Ujaran Kebencian," *Comput. Sci.*, vol. 4, no. 1, pp. 80–88, 2024, [Online]. Available: https://jurnal.bsi.ac.id/index.php/co-science/article/view/2990

[10] M. Safrudin, M. Martanto, and U. Hayati, "Perbandingan Kinerja Naïve Bayes Dan Support Vector Machine Untuk Klasifikasi Sentimen Ulasan Game Genshin Impact," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3182–3188, 2024, doi: 10.36040/jati.v8i3.8415.

[11] E. Apriani, F. Oktavianalisti, L. D. H. Monasari, I. Winarni, and I. F. Hanif, "Analisis Sentimen Penggunaan TikTok Sebagai Media Pembelajaran Menggunakan Algoritma Naïve Bayes Classifier," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 3, pp. 1160–1168, 2024, doi: 10.57152/malcom.v4i3.1482.

[12] D. Darwis, E. S. Pratiwi, and A. F. O. Pasaribu, "Penerapan Algoritma Svm Untuk Analisis Sentimen Pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia," *Edutic - Sci. J. Informatics Educ.*, vol. 7, no. 1, pp. 1–11, 2020, doi: 10.21107/edutic.v7i1.8779.

[13] M. Rahman Fauzan, H. Oktafia Lingga Wijaya, and J. Karman, "Analisis Sentimen Masyarakat Terhadap Kenaikan Harga Bbm Di Media Sosial Twitter Menggunakan Metode Support Vector Machine," *Semin. Ris. Mahasiswa-Computer Electr. (SERIMA-CE*, vol. 1, no. 1, p. 82, 2023.

[14] Yuyun, Nurul Hidayah, and Supriadi Sahibu, "Algoritma Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 820–826, 2021, doi: 10.29207/resti.v5i4.3146.

[15] A. Putri *et al.*, "Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 20–26, 2023, doi: 10.57152/malcom.v3i1.610.

[16] M. Dennis, R. Rahmaddeni, F. Zoromi, and M. K. Anam, "Penerapan Algoritma Naïve Bayes Untuk Pengelompokkan Predikat Peserta Uji Kemahiran Berbahasa Indonesia," *J. Media Inform. Budidarma*, vol. 6, no. 2, p. 1183, 2022, doi: 10.30865/mib.v6i2.3956.

[17] D. N. Herisnan, E. Dadynata, and L. Efrizoni, "Komparasi Algoritma Decision Tree, SVM, Naive Bayes Dalam Prediksi Penyakit Liver," *J. Jar. Sist. Inf. Robot.*, vol. 8, no. 1, pp. 104–109, 2024, [Online]. Available: http://ojsamik.amikmitragama.ac.id

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

[18] I. N. Rizki, D. Prayoga, M. L. Puspita, and M. Q. Huda, "Implementasi Exploratory Data Analysis Untuk Analisis Dan Visualisasi Data Penderita Stroke Kalimantan Selatan Menggunakan Platform Tableau," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 1, 2024, doi: 10.23960/jitet.v12i1.3856.

[19] R. Safitri, I. Ali, and N. Rahaningsih, "Analisis Sentimen Terhadap Tren Fashion Di Media Sosial Dengan Metode Support Vector Machine (Svm)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 2, pp. 1746–1754, 2024, doi: 10.36040/jati.v8i2.9045.

[20] N. Fitriyah, B. Warsito, and D. A. I. Maruddani, "Analisis Sentimen Gojek Pada Media Sosial Twitter Dengan Klasifikasi Support Vector Machine (Svm," *J. Gaussian*, vol. 9, no. 3, pp. 376–390, 2020, doi: 10.14710/j.gauss.v9i3.28932.

[21] K. Parmar, "Social Media Sentiments Analysis Dataset." Accessed: Jan. 24, 2025. [Online]. Available: https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset

[22] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *J. Tekno Kompak*, vol. 14, no. 2, p. 74, 2020, doi: 10.33365/jtk.v14i2.732.

[23] M. R. Hunafa and A. Hermawan, "KLIK: Kajian Ilmiah Informatika dan Komputer Perbandingan Algoritma Naïve Bayes dan K-Nearest Neighbor Pada Imbalace Class Dataset Penyakit Diabetes," *Media Online*, vol. 4, no. 3, pp. 1551–1561, 2023, doi: 10.30865/klik.v4i3.1486.

[24] I. Muhamad Malik Matin, "Hyperparameter Tuning Menggunakan GridsearchCV pada Random Forest untuk Deteksi Malware," *Multinetics*, vol. 9, no. 1, pp. 43–50, 2023, doi: 10.32722/multinetics.v9i1.5578.