

Perbandingan Akurasi Naive Bayes dengan SMOTE dan SVM dengan Hyperparameter Tuning untuk Analisis Sentimen Pemilihan Presiden Indonesia

Eka Belandini¹, Kamila Fajar Pertiwi²

^{1,2} Informatika, Universitas Jenderal Soedirman, Indonesia
¹H1D022002, ²H1D022035

Email: ¹eka.belandini@mhs.unsoed.ac.id, ²kamila.pertiwi@mhs.unsoed.ac.id

Received : Nov 26, 2024; Revised : Nov 26, 2024; Accepted : Nov 26, 2024; Published : Nov 26, 2024

Abstrak

Pemilihan presiden 2024 telah menghasilkan banyak tanggapan beragam dari masyarakat Indonesia. Reaksi tersebut mencerminkan berbagai pandangan dan harapan terhadap pemimpin masa depan Indonesia. Penelitian ini menginvestigasi efektivitas algoritma Naive Bayes dengan Synthetic Minority Over-sampling Technique (SMOTE) dibandingkan dengan algoritma Support Vector Machine (SVM) dengan Hyperparameter Tuning dalam menganalisa sentimen mengenai pemilihan presiden Indonesia di Twitter. Tujuan utamanya adalah untuk menentukan metode mana yang menghasilkan akurasi yang lebih tinggi dalam klasifikasi sentimen. Metodologi yang digunakan adalah *preprocessing* dataset Twitter, menerapkan SMOTE untuk menangani ketidakseimbangan kelas pada Naive Bayes, dan menggunakan Grid Search untuk Hyperparameter Tuning pada SVM. Hasil penelitian menunjukkan bahwa algoritma Naive Bayes dengan SMOTE mencapai akurasi yang lebih tinggi yaitu 91,69%, dibandingkan dengan SVM dengan kernel linear (90,26%) dan SVM dengan kernel RBF (85,77%) setelah dilakukan Hyperparameter Tuning. Hal ini menunjukkan bahwa penanganan ketidakseimbangan kelas dengan SMOTE secara signifikan meningkatkan kinerja algoritma Naive Bayes dalam tugas analisis sentimen dibandingkan SVM, bahkan dengan pengoptimalan.

Keywords : *Class Imbalance, Hyperparameter Tuning, Naive Bayes, Sentiment Analysis, SMOTE, SVM.*

This work is an open access article and licensed under a Creative Commons Attribution-Non Commercial 4.0 International License



1. LATAR BELAKANG

Saat ini, perkembangan teknologi informasi dan komunikasi berlangsung dengan pesat. Di era dinamis seperti sekarang, komunikasi telah menjadi bagian integral dari kehidupan manusia. Internet memainkan peran penting dalam meningkatkan efisiensi dan fleksibilitas teknologi komunikasi, memungkinkan manusia untuk dengan mudah mengakses informasi yang dibutuhkan. Akibatnya, lingkungan baru terbentuk di mana manusia berevolusi menjadi pencari informasi yang handal. Informasi yang tersedia juga semakin beragam, baik dalam konten maupun tautan yang menyertainya. Hal ini mendorong banyak orang untuk terus mencari informasi yang mereka perlukan secara tak henti di internet [1]. Saat ini dengan kepopuleran media sosial telah menjadi alat yang ampuh untuk mempengaruhi orang dan berbagi opini dengan publik. Sebagai contoh, Twitter merupakan salah satu media sosial yang sering digunakan oleh para politisi untuk kampanye politik di Indonesia [2].

Twitter merupakan media sosial yang berisi kumpulan pesan tentang opini, dan ungkapan emosi. Manfaat Twitter tidak hanya sebagai media informasi dan komunikasi, tetapi juga sebagai media untuk mengekspresikan opini publik. Di Indonesia, khususnya pada momen pemilu, Twitter biasanya digunakan untuk kegiatan politik, yaitu kampanye, mengkritik kebijakan tertentu, dan para tim sukses dari berbagai partai pun ramai-ramai terjun ke media sosial untuk meningkatkan popularitas kandidatnya. Masyarakat di media sosial, khususnya Twitter, bebas memberikan pendapatnya mengenai tokoh-tokoh publik. Opini ini tidak hanya berupa tanggapan positif tetapi juga negatif [3].

Memasuki tahun 2024, Indonesia masuk ke tahun politik, dimana para peserta pemilihan umum mulai mempersiapkan diri untuk menghadapi pesta demokrasi pemilihan umum presiden yang akan berlangsung pada tahun 2024. Berbagai lembaga telah melakukan survei lapangan untuk mengetahui elektabilitas terhadap seorang figur bakal calon presiden. Tiga lembaga survei atau lembaga politik telah melakukan survei untuk menentukan elektabilitas 3 besar calon presiden 2024. Hasil survei Lingkaran Survei Indonesia dan Indo Barometer menempatkan 3 figur bakal calon presiden tertinggi, figur tersebut adalah Ganjar Pranowo, Anies Baswedan, dan Prabowo Subianto. Masyarakat baik yang mendukung ataupun tidak, banyak membicarakan pada media sosial. Salah satu media sosial yang digunakan untuk beropini figur tersebut adalah media sosial Twitter. Pada media sosial Twitter, pengguna dapat dapat mem-posting, membaca, dan mengomentari apa saja yang dituliskan oleh pengguna, sehingga pengguna dapat membaca sebuah informasi kemudian memberikan sebuah sentimen atau opini dalam bentuk positif atau negatif terhadap figur bakal calon presiden 2024 [4]. Analisis sentimen merupakan teknik untuk menganalisis opini, sentimen, rating, dan emosi terhadap entitas seperti produk, layanan, peristiwa, atau atribut lainnya [5]. Pemikiran dasar dari teknik analisis sentimen adalah untuk mengelompokkan teks, kalimat, atau dokumen kemudian menentukan teks, kalimat, atau dokumen tersebut termasuk ke dalam sentimen atau opini yang positif, negatif, atau netral [6]. Hal ini dapat digunakan untuk menganalisis tweet feed untuk mendapatkan sentimen opini tokoh publik untuk menemukan tren respon positif atau negatif. Analisis sentimen adalah proses memahami, mengekstraksi, dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam kalimat opini. Kemudian mengklasifikasikan berbagai polaritas teks pada kalimat atau dokumen tersebut dalam dua kelas kemungkinan, baik positif atau negatif.

Penelitian mengenai analisis sentimen telah dilakukan berdasarkan penelitian sebelumnya. Penelitian yang dilakukan oleh Bayu, *dkk.* Hasil penelitian menunjukkan bahwa penggunaan FastText dalam ekstraksi fitur dengan menggunakan TF-IDF dalam analisis sentimen terhadap judul berita terkait Pilpres 2024 dengan metode Support Vector Machine (SVM). Model SVM dengan parameter terbaik (RBF, 10, 1) pada rasio data uji 90:10 mencapai akurasi 93% dan F1-Score 93%, memfasilitasi analisis sentimen judul berita dengan baik [7]. Penelitian selanjutnya dilakukan oleh Lisy, *dkk.* Hasil penelitian menunjukkan pendekatan terbaik pada rasio 80:20 dengan akurasi sebesar 88.94%, Recall 93.08%, F1-Score 90.43%, dan akurasi 90.75% pada menganalisis informasi terkait opini publik mengenai Pilpres Indonesia 2024 menggunakan algoritma SVM dengan ekstraksi fitur Word2Vec [8].

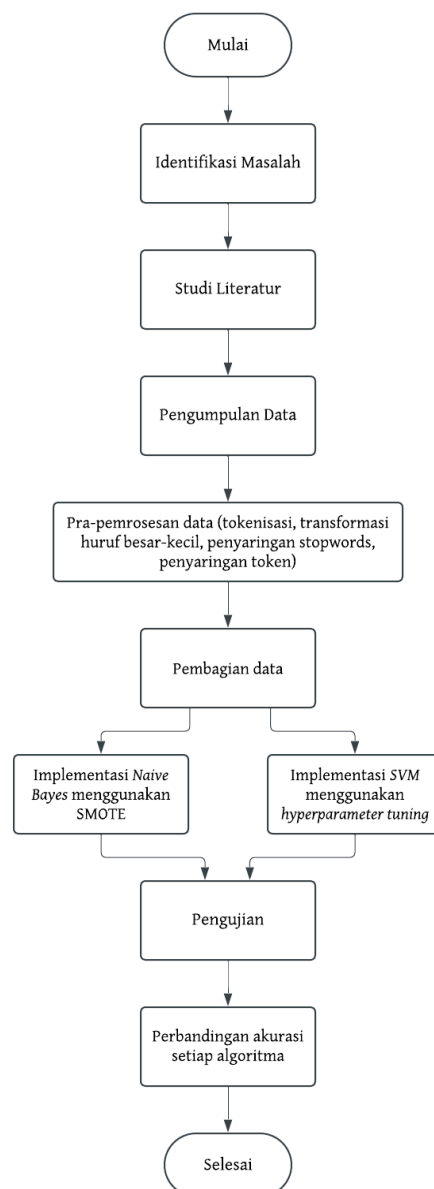
Penelitian mengenai analisis sentimen telah dilakukan berdasarkan penelitian sebelumnya. Penelitian yang dilakukan oleh Bayu, *dkk.* Hasil penelitian menunjukkan bahwa penggunaan FastText dalam ekstraksi fitur dapat menggunakan TF-IDF dalam analisis sentimen terhadap judul berita terkait Pilpres 2024 dengan metode Support Vector Machine (SVM). Model SVM dengan parameter terbaik (RBF, 10, 1) pada rasio data uji 90:10 mencapai akurasi 93% dan F1-Score 93%, memfasilitasi analisis sentimen judul berita dengan baik [7]. Penelitian selanjutnya dilakukan oleh Lisy, *dkk.* Hasil penelitian menunjukkan pendekatan terbaik pada rasio 80:20 dengan akurasi sebesar 88.94%, Recall 93.08%, F1-Score 90.43%, dan akurasi 90.75% pada menganalisis informasi terkait opini publik mengenai Pilpres Indonesia 2024 menggunakan algoritma SVM dengan ekstraksi fitur Word2Vec [8].

Dengan analisis sentimen, sebuah metode untuk mendukung klasifikasi. Kami menggunakan algoritma Naive Bayes dan Support Vector Machine (SVM). Kami memilih untuk menggunakan algoritma Naive Bayes dan meningkatkan akurasi dengan metode SMOTE. Sedangkan untuk

algoritma SVM, kami meningkatkan akurasi dengan metode tuning hyperparameter, yaitu grid search, dan mencoba dua kernel, yaitu linear dan RBF dalam penelitian kami. Dari 1336 tweet dalam dataset, hanya 267 tweet yang diuji dalam perbandingan antara algoritma Naive Bayes dan SVM. Naive Bayes merupakan algoritma yang sederhana dan cepat, efektif dalam klasifikasi teks dan sering digunakan dalam analisis sentimen, namun memiliki keterbatasan dalam menangani data yang tidak seimbang. Dengan metode SMOTE, masalah data yang tidak seimbang dapat diatasi melalui pembangkitan sampel sintetis di kelas minoritas, sehingga meningkatkan performa Naive Bayes. Untuk SVM, yang dikenal menghasilkan performa dengan margin keputusan yang lebih optimal, kami menerapkan hyperparameter tuning grid search untuk mencari kombinasi parameter terbaik dan mencoba dua kombinasi kernel yang berbeda untuk menentukan kernel yang paling akurat menurut data kami. Kombinasi ini diharapkan dapat memberikan hasil analisis sentimen yang lebih akurat dan dapat diandalkan

2. METODE

Metode yang digunakan dalam penelitian ini ditunjukkan pada gambar 1:



Gambar 1. Tahapan Penelitian

2.1. Identifikasi Masalah

Permasalahan dalam penelitian ini adalah bagaimana cara meningkatkan nilai akurasi dengan menggunakan Naive Bayes dan Support Vector Machine (SVM) pada dataset pemilihan presiden yang diperoleh dari Kaggle.

2.2. Studi Literatur

Informasi yang relevan dikumpulkan berdasarkan sumber-sumber yang berasal dari buku, artikel ilmiah, jurnal, dan tesis. Terdapat sekitar 28 literatur yang digunakan dalam penelitian ini. Informasi terkait penelitian meliputi Pemilihan Presiden 2024, analisis sentimen, algoritma Naive Bayes, algoritma SVM, dan metode pengujian dengan menggunakan precision, recall, dan akurasi.

2.3. Pengumpulan Data

Dalam penelitian ini, data yang digunakan adalah dataset open-source yang diperoleh dari platform Kaggle [9]. Dataset ini terdiri dari 1336 tweet dari masyarakat Indonesia yang berkaitan dengan calon presiden dan Pemilu 2024.

2.4. Pra-pemrosesan Data

Untuk menghasilkan data yang bersih sesuai kebutuhan penelitian, maka dilakukan beberapa tahapan, yaitu:

- a. Labelling : Labelling memberikan pelabelan positif, negatif, atau netral pada data yang diekstrak dari sentimen Twitter.
- b. Tokenisasi : Memecah teks menjadi unit yang lebih kecil seperti kata-kata atau token
- c. Transform Case : Mengubah semua teks menjadi huruf kecil untuk konsistensi pemrosesan
- d. Filter Stopwords (Dictionary) : Menghapus kata-kata yang tidak punya arti tertentu dalam analisis teks
- e. Filter Tokens (by Length): Memfilter token-token berdasarkan panjangnya dan mengeliminasi token-token yang terlalu pendek atau terlalu panjang yang mungkin tidak relevan.

2.5. Pembagian Data

Pada tahap ini, data akan dipisahkan menjadi dua bagian, yaitu data training dan data testing. Data pelatihan digunakan untuk melatih dan mengembangkan model. Kumpulan data training biasanya digunakan untuk mengestimasi parameter yang berbeda atau untuk membandingkan performa model yang berbeda. Data pengujian digunakan setelah proses pelatihan selesai. Data pelatihan dan pengujian dibandingkan untuk memeriksa apakah model akhir yang digunakan bekerja dengan benar [10].

2.6. Implementasi

Penelitian ini menggunakan metode Naive Bayes yang cocok untuk analisis sentimen karena sederhana dan efektif dalam menangani data probabilistik, kemudian penggunaan SMOTE juga digunakan untuk meningkatkan kinerja metode Naive Bayes. Selain itu, penelitian ini mengeksplorasi penggunaan Support Vector Machine (SVM) dengan fungsi kernel Linear dan Radial Basis Function (RBF) yang kemudian dioptimasi parameternya

dengan menggunakan tuning Hyperparameter melalui grid search untuk mencapai akurasi yang lebih tinggi.

2.6.1. Naive Bayes

Metode Naive Bayes adalah metode klasifikasi untuk analisis sentimen penambangan teks. Metode ini dapat diklasifikasikan dan memiliki kemampuan akurasi data dan komputasi [10]. Secara umum, teorema Bayes dapat dilambangkan dengan persamaan berikut.

$$P(A|B) = \frac{P(A|B)P(A)}{P(B)} \quad (1)$$

Probabilitas bersyarat dari A yang diberikan B dinyatakan dengan $P(A|B)$. $P(B|A)$ adalah probabilitas bersyarat dari B yang diberikan A. $P(A)$ adalah probabilitas kejadian A. $P(B)$ adalah probabilitas kejadian B [11].

2.6.2. Support Vector Machine (SVM)

Support vector machine (SVM) adalah jenis model vektor berdasarkan pengklasifikasi yang memerlukan konversi teks ke vektor sebelum dapat digunakan untuk klasifikasi [12]. Konsep klasifikasi dengan Support Vector Machine adalah mencari hyperplane terbaik yang berfungsi sebagai pemisah dua kelas data. Support Vector Machine dapat bekerja pada dataset berdimensi tinggi dengan menggunakan trik kernel. Support Vector Machine hanya menggunakan beberapa titik data terpilih yang memberikan kontribusi (support vector) untuk membentuk sebuah model yang akan digunakan dalam proses klasifikasi. Persamaan SVM adalah sebagai berikut.

$$f(x) = w \cdot x + b \quad (2)$$

$$f(x) = \sum_{i=1}^m a_i y_i K(x, x_i) + b \quad (3)$$

Parameter hyperplane yang dicari (Garis tegak lurus antara garis hyperplane dan titik vektor pendukung) dinyatakan dengan $w \cdot x$ adalah titik data input Support Vector Machine. a_i adalah nilai bobot dari setiap titik data. $K(x, x_i)$ adalah fungsi kernel. b adalah parameter hyperplane yaitu dicari (nilai bias) [12]. Pada algoritma SVM, terdapat 2 kasus dalam memisahkan kelas dengan Hyperplane, yaitu kelas yang dapat dipisahkan secara sempurna yang disebut dengan linear SVM, dan kelas yang tidak dapat dipisahkan secara sempurna yang disebut dengan non linear SVM. SVM non-linear merupakan solusi dari permasalahan SVM linear dengan melakukan fungsi kernel pada ruang fitur berdimensi tinggi [15]. Pada penelitian ini, fungsi kernel yang digunakan adalah Linear dan Radial Basis Function (RBF). Definisi SVM linear dan nonlinear dapat dilihat pada Tabel 1. Parameter hyperplane yang dicari (Garis tegak lurus antara garis hyperplane dengan titik support vector) dinyatakan dengan $w \cdot x$ adalah titik data input Support Vector Machine. a_i adalah titik data input Support Vector Machine. nilai bobot dari setiap titik data. $K(x, x_i)$ adalah fungsi kernel. b adalah parameter hyperplane yaitu dicari (nilai bias) [13]. Pada algoritma SVM, terdapat 2 kasus dalam memisahkan kelas dengan Hyperplane, yaitu kelas yang dapat dipisahkan secara sempurna yang disebut dengan linear SVM, dan kelas yang tidak dapat dipisahkan secara sempurna yang disebut dengan non linear SVM. SVM non-linear merupakan solusi

dari permasalahan SVM linear dengan melakukan fungsi kernel pada ruang fitur berdimensi tinggi [14]. Pada penelitian ini, fungsi kernel yang digunakan adalah Linear dan Radial Basis Function (RBF). Definisi SVM linear dan nonlinear dapat dilihat pada Tabel 1.

Tabel 1. SVM Linear dan Non-Linear

SVM	Jenis Kernel	Rumus
Linear	Linear	$K(x, y) = x \cdot y$
Non-Linear	RBF	$K(X_1 - X_2) = \exp\left(-\frac{\ X_1 - X_2\ ^2}{2\sigma^2}\right)$

2.6.3 Metode SMOTE

Metode pengambilan sampel berlebih SMOTE (Synthetic Minority Over-Sampling Technique) digunakan untuk menangani masalah ketidakseimbangan kelas. Teknik ini mensintesis sampel baru dari kelas minoritas untuk menyeimbangkan dataset dengan membuat instance baru dari kelas minoritas dengan membentuk kombinasi cembung dari instance yang berdekatan. Kemudian, tingkat oversampling yang diperlukan dipilih secara acak. Algoritma SMOTE secara umum lebih akurat daripada pendekatan oversampling biasa [15].

2.6.4. Metode Hyperparameter Turning

Hyperparameter tuning melibatkan penentuan nilai optimal untuk parameter yang mengendalikan proses pembelajaran model pembelajaran mesin, yang ditetapkan sebelum pelatihan. Grid Search mempartisi rentang hiperparameter ke dalam node grid untuk kombinasi parameter. Dengan nilai parameter M dan N untuk C dan γ masing-masing, kombinasi M x N dieksplorasi melalui pelatihan SVM. Akan tetapi, M dan N yang lebih besar menyebabkan terhadap tantangan komputasi karena ekspansi kombinatorial. Setiap eksplorasi simpul jaringan meningkatkan waktu, dan pemilihan jarak langkah berdampak pada akurasi [16].

2.7. Pengujian

Setelah mengimplementasikan algoritma, langkah selanjutnya adalah mengevaluasi kinerjanya. Tahap pengujian melibatkan penggunaan set data uji untuk mengukur seberapa baik setiap model menggeneralisasi data baru yang belum pernah dilihat sebelumnya. Hal ini dicapai dengan membandingkan label prediksi yang dihasilkan oleh model dengan label aktual dalam set pengujian. Metrik kinerja utama yang digunakan dalam penelitian ini adalah presisi, recall, dan akurasi:

- Presisi: Mengukur jumlah prediksi positif yang benar dibagi dengan jumlah total prediksi positif yang dibuat oleh model.

$$\text{Presisi} = \frac{TP}{FP+TP} \quad (4)$$

- Recall: Mengukur jumlah prediksi positif yang benar dibagi dengan jumlah total aktual contoh positif dalam data.
-

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

- Akurasi: Proporsi hasil yang benar (baik positif maupun negatif) di antara total jumlah kasus yang diperiksa.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Deskripsi:

TP: True Positives, jumlah kasus positif yang diprediksi dengan benar sebagai TN positif: Negatif Sejati, jumlah kasus yang diprediksi dengan benar sebagai negatif

FN: False Negatives, jumlah kasus positif yang secara tidak tepat diprediksi sebagai negatif. FP: Positif Palsu, jumlah kasus negatif yang salah diprediksi sebagai positif.

2.8. Perbandingan Akurasi Setiap Algoritma

Tahap terakhir dari penelitian ini adalah membandingkan kinerja pengklasifikasi Naive Bayes (ditingkatkan dengan SMOTE) dan pengklasifikasi SVM (dioptimalkan melalui hyperparameter tuning) menggunakan metrik yang diperoleh selama pengujian. Perbandingan ini membantu untuk menentukan algoritma mana yang bekerja lebih baik dalam hal akurasi keseluruhan dan kemampuan untuk mengidentifikasi sentimen dengan benar dalam set data. Perbandingan ini melibatkan analisis akurasi kedua model untuk mengidentifikasi model mana yang memiliki tingkat akurasi yang lebih tinggi. Dengan menganalisis akurasi yang diperoleh dari kedua algoritma setelah modifikasi, kita dapat menyimpulkan algoritma lebih cocok untuk analisis sentimen dalam konteks dataset pemilihan presiden 2024.

3. HASIL

Penelitian ini mengimplementasikan dan menguji algoritma Naive Bayes dengan teknik SMOTE dan algoritma Support Vector Machine (SVM) dengan hyperparameter tuning, dimana *tool* yang dipilih untuk membantu proses klasifikasi adalah RapidMiner. Dataset dibagi menjadi set training dan testing dengan perbandingan 80:20. Performa setiap model dievaluasi dengan menggunakan metrik akurasi.

3.1. Pengumpulan Data

Pengumpulan data dimulai dengan pencarian dataset melalui Kaggle. Setelah dataset diperoleh sebelum memulai preprocessing, pelabelan data dilakukan untuk menentukan polaritas setiap objek data. Dataset terdiri dari 1336 tweet. Pengumpulan data sebelum pra-pemrosesan ditunjukkan pada Tabel 2.

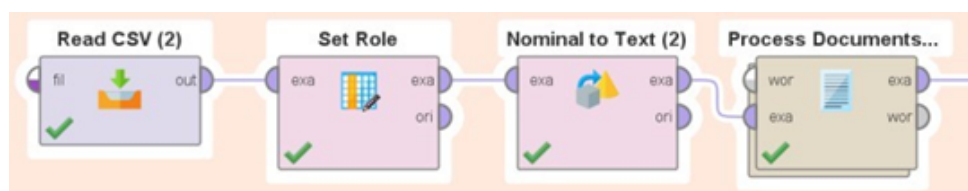
Tabel 2. Hasil Pengumpulan Data

No	Tanggal	Teks	Nama Pengguna	Panjang Teks	Sentimen
1	Rabu 13 Desember 23:47:11	@Hasbil_Lbs @aniesbaswedan gampang sih bikin video kya gt... konsepnya gt gampang	DzulfiqorParisi	254	Negatif

	+0000 2023	banget, sebarin d medsos... rame deh netizen. tp yg proteksi ingat adalah orang pintar pasti minumnya tolak dungu... #AniesMuhaimin2024			
2	Rabu 13 Desembe r 23:46:34 +0000 2023	Lagi, lagi dan terus Tokoh Harapan yang dirindukan di setiap Jengkal Tanah air Indonesia disambut dengan meriah dan penuh antusias. #AMINAJaDulu #AminPalingSiapUntukIndon esia #AniesMuhaimin2024	sumadiseloguno	222	Positif
...					
1336	Senin 18 Desembe r 00:11:04 +0000 2023	Pak Anies itu sebetulnya kami minta sebagai Cawapres, tapi beliau mengatakan beliau sudah angkat sumpah 5 tahun, Beliau harus setia pada sumpahnya kepada rakyat dan sebagainya ~ PS *Silahkan rakyat menilai #AMINAJaDulu #AniesMuhaimin2024	Mdy_Asmara170 1	297	Positif

3.2. Pemrosesan Data

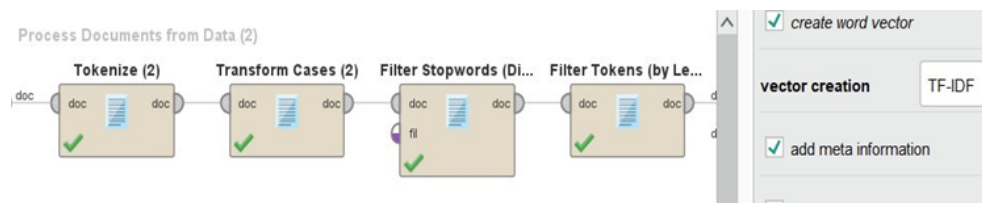
Gambar 2 menunjukkan proses pra-pemrosesan data yang terdiri dari beberapa langkah. Hal pertama yang harus dilakukan adalah mengimpor dataset. Selanjutnya, pada operator Set Role, peran dari kolom label sentimen ditetapkan sebagai label target. Kemudian tipe data dari kolom yang berisi data nominal diubah menjadi teks.



Gambar 2. Pemrosesan Awal Data

Kemudian langkah selanjutnya adalah memproses teks dari kolom data ke dalam format yang siap untuk analisis teks. Pada Process Documents, langkah-langkah pemrosesan

teks dan transformasi teks menjadi vektor kata menggunakan TF-IDF dilakukan untuk mempersiapkan data teks secara efektif untuk analisis lebih lanjut.



Gambar 3. Memproses Dokumen dari Data

Pra-pemrosesan lanjutan meliputi Tokenize, Transform Case, Filter Stopwords, dan Filter Token. Langkah selanjutnya adalah mengeksekusi dokumen keluaran dari setiap tahap pra-pemrosesan lanjutan. Tabel 3 menunjukkan hasil pra-pemrosesan lanjutan untuk satu kalimat/dokumen.

Tabel 3. Tahap Preprocessing Lanjutan

No	Tahap Pemrosesan Awal	Dokumen
1	Kalimat Asli	SAKING TAKUTNYA SAMA KEKUATAN AMIN https://t.co/GAuRjX3s6F SO AFRAID OF THE POWER OF AMIN https://t.co/GAuRjX3s6F
2	Tokenize	['SAKING', 'TAKUTNYA', 'SAMA', 'KEKUATAN', 'AMIN', ' https://t.co/GAuRjX3s6F '] ['SO', 'AFRAID', 'OF', 'THE', 'POWER', 'OF', 'AMIN', ' https://t.co/GAuRjX3s6F ']
3	Ubah Kasus	['saking', 'takutnya', 'sama', 'kekuatan', 'amin', ' https://t.co/gaurjx3s6f '] ['so', 'afraid', 'of', 'the', 'power', 'of', 'amin', ' https://t.co/gaurjx3s6f ']
4	Saring Kata	['saking', 'takutnya', 'kekuatan', 'amin', ' https://t.co/gaurjx3s6f '] ['so', 'afraid', 'power', 'amin', ' https://t.co/gaurjx3s6f ']
5	Token Filter	['saking', 'takutnya', 'kekuatan', 'amin'] ['so', 'afraid', 'power', 'amin']

3.3. Pembagian Data

Pada tahap selanjutnya, dilakukan split data dengan parameter 0.8 dan 0.2. Jumlah data yang terdapat pada dataset adalah 1336 data. Dataset tersebut dibagi menjadi dua bagian, 1069 data untuk data training dan 267 data untuk data testing.

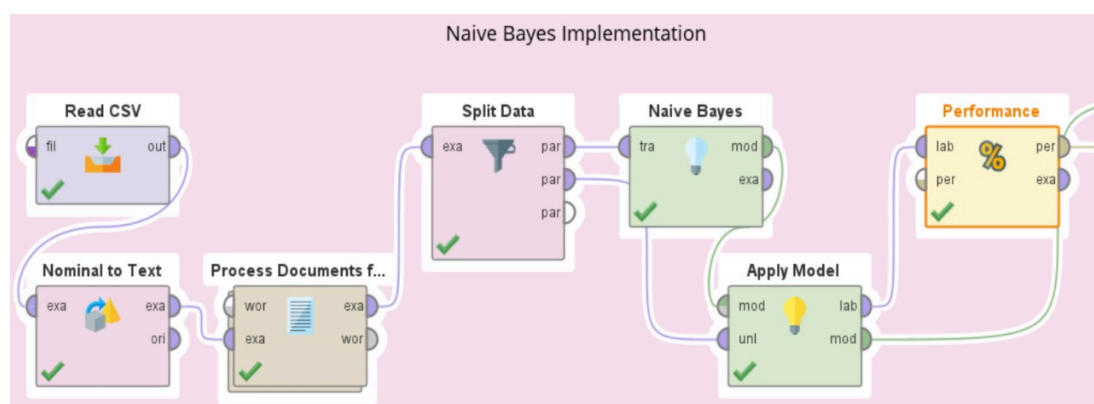
3.4. Pemodelan Klasifikasi

Klasifikasi adalah teknik analisis data yang digunakan untuk memprediksi kategori atau kelas dari data baru berdasarkan informasi yang dipelajari dari data yang sudah ada. Dalam penelitian ini, dua metode klasifikasi yang digunakan adalah Naive Bayes dan

Support Vector Machine (SVM). Kedua metode tersebut diterapkan pada dataset yang telah diproses sebelumnya dan hasilnya dievaluasi menggunakan metrik akurasi.

3.4.1. Implementasi Naive Bayes

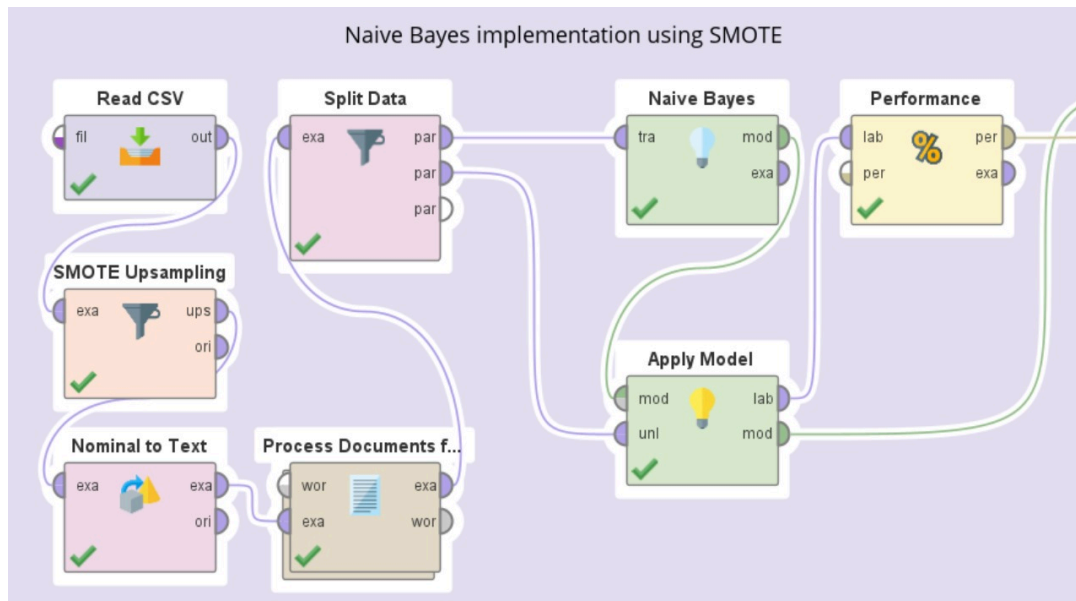
Hal pertama yang harus dilakukan adalah mengimplementasikan Algoritma Klasifikasi Naive Bayes dengan menggunakan aplikasi RapidMiner. Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai-nilai atribut secara kondisional independen satu sama lain jika diberikan nilai output [17]. Proses pertama yang dilakukan adalah melakukan split data untuk membagi dataset menjadi data training dan data testing dengan rasio 0.8 dan 0.2. Selanjutnya, data training masuk ke dalam model Naive Bayes, dan data testing masuk ke dalam Apply Model untuk menghubungkannya dengan model Naive Bayes. Terakhir, operator kinerja digunakan untuk menguji kinerja algoritma.



Gambar 4. Proses Algoritma Naive Bayes Proses Algoritma Naive Bayes

3.4.2. Implementasi Naive Bayes menggunakan SMOTE

Setelah mengimplementasikan Naive Bayes sebelumnya, hasil akurasi yang didapat belum maksimal, kemudian kita akan mencoba meningkatkan akurasi dengan menggunakan metode SMOTE. Dalam proses pemodelan, Naive Bayes digunakan dengan SMOTE sebagai operator untuk mengatasi ketidakseimbangan data. Teknik ini sangat penting untuk mengurangi dampak dari distribusi kelas yang miring yang biasa ditemui dalam tugas analisis sentimen [18]. Berikut ini adalah proses implementasi SMOTE pada Naive Bayes dengan menggunakan dataset yang sama dan data split yang sama.



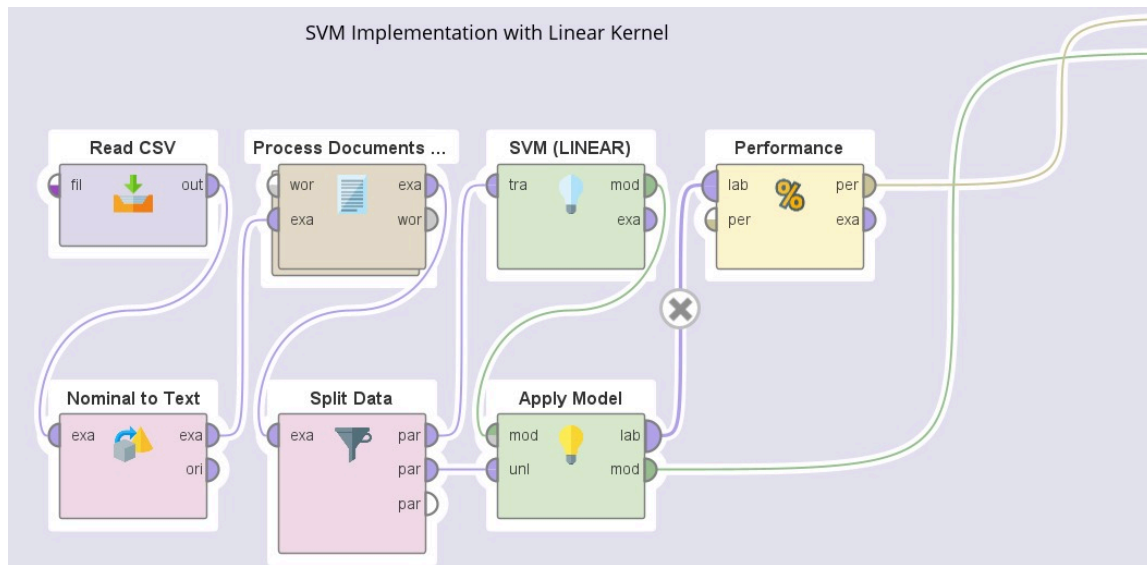
Gambar 5. Proses Algoritma Naive Bayes Proses Algoritma Naive Bayes dengan metode SMOTE

3.4.3 Implementasi SVM

Tahapan proses selanjutnya adalah data yang akan diupload ke dalam *tool* diolah dengan algoritma Support Vector Machine (SVM) untuk mendapatkan hasil awal dari algoritma ini [19]. Pada penelitian ini digunakan dua jenis kernel sebagai pembanding untuk mendapatkan hasil yang terbaik terutama dalam akurasi. Algoritma klasifikasi SVM Linear menggunakan kernel Linear dan SVM Non-Linear menggunakan kernel RBF diimplementasikan. Pemodelan klasifikasi algoritma SVM dilakukan dengan menggunakan aplikasi RapidMiner.

3.4.3.1. Implementasi SVM dengan Kernel Linier

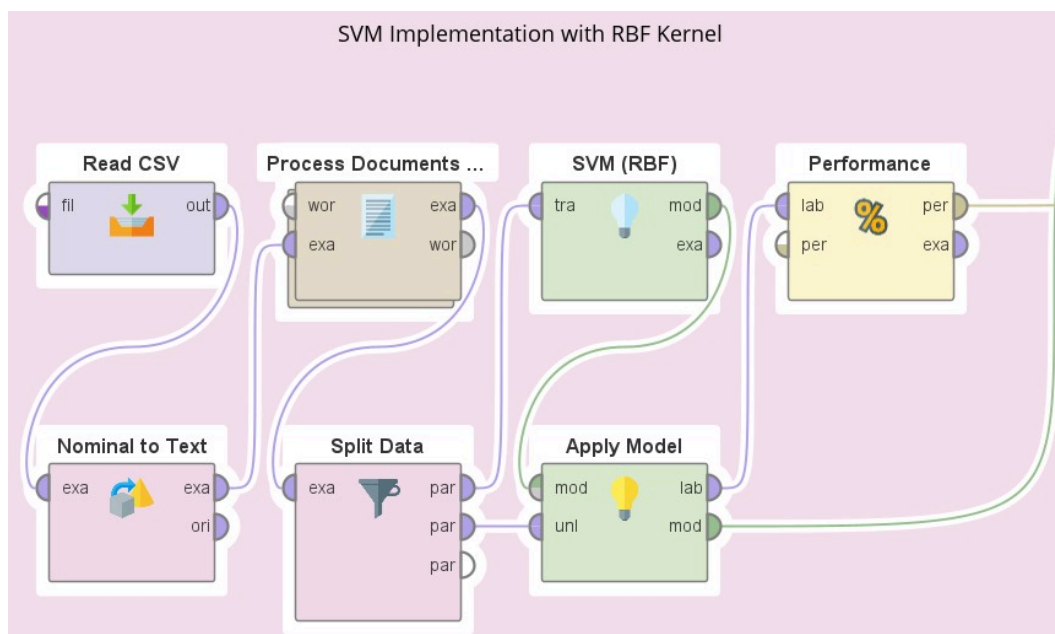
Kernel linear yang juga dikenal sebagai soft margin akan mencoba menemukan hyperplane yang berupa garis lurus tetapi dapat mentoleransi satu atau lebih kesalahan klasifikasi data. Besarnya toleransi *misclassification* yang diberikan sangat mempengaruhi akurasi hyperplane. Dalam Sklearn toleransi tersebut disebut dengan C . Semakin besar nilai C , maka toleransi *misclassification* semakin sedikit dan margin semakin sempit [20]. Implementasi SVM linier menggunakan kernel linier menggunakan satu parameter yaitu C . Pada penelitian ini, nilai C yang digunakan adalah 100.



Gambar 6. Implementasi SVM dengan Kernel Linier Implementasi SVM dengan Kernel Linier

3.4.3.2. Implementasi SVM dengan Kernel RBF

Radial Basis Function (RBF) adalah fungsi kernel yang nilainya bergantung pada jarak dari titik asal atau titik tertentu [21]. Terdapat dua parameter yang perlu diperhatikan, yaitu C dan γ (gamma). Parameter C bertujuan untuk mengetahui seberapa besar kesalahan yang harus dihindari dalam mengklasifikasikan data training, semakin besar nilai C , maka semakin rendah kesalahan klasifikasi data training. Parameter γ (gamma) menentukan seberapa jauh pengaruh satu sampel data pelatihan. Ini berarti semakin kecil nilai γ (gamma), semakin jauh jarak dari titik-titik data yang akan dihitung [20]. Dalam penelitian ini, nilai C yang digunakan adalah 100 dan nilai γ (gamma) yang digunakan adalah 0,01.



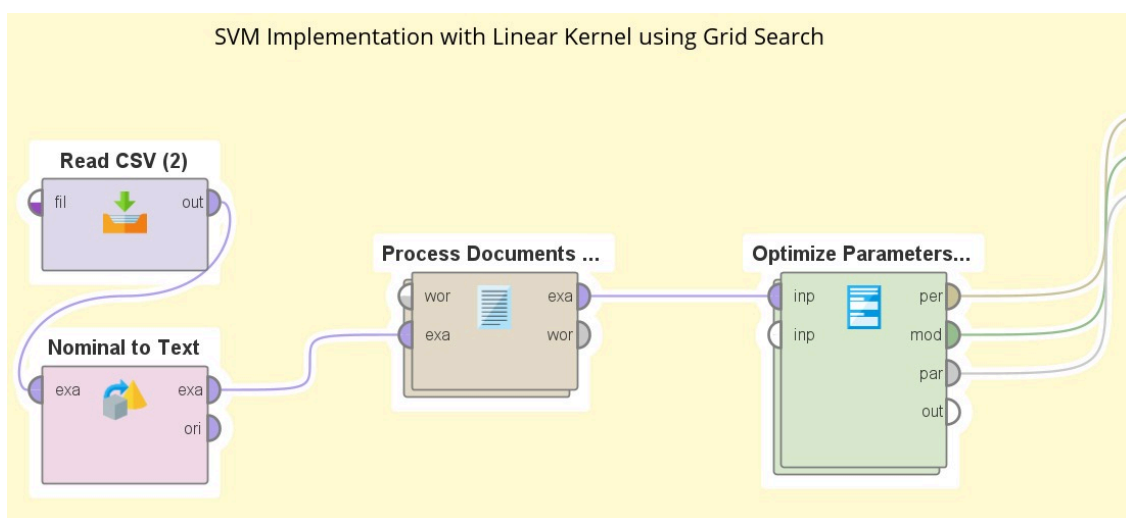
Gambar 7. Implementasi SVM dengan Kernel RBF Implementasi SVM dengan Kernel RBF

3.4.4. Implementasi SVM menggunakan Hyperparameter Tuning dengan Grid Search

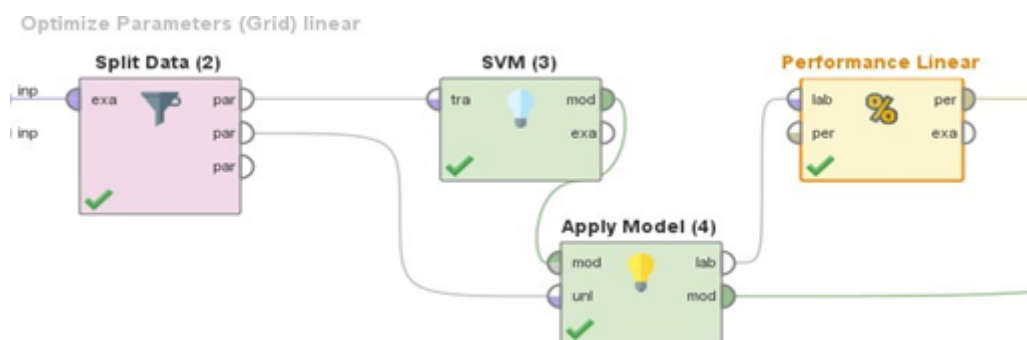
Pada tahap ini dilakukan modifikasi dengan Hyperparameter Tuning dengan metode Grid Search pada SVM Linear menggunakan kernel linear dan SVM Non Linear menggunakan kernel RBF. Grid Search adalah cara brute force untuk mencari hyperparameter, dengan batas bawah dan batas atas yang telah ditentukan dengan langkah-langkah tertentu. Grid Search bekerja berdasarkan hasil kali kartesius dari kumpulan nilai yang berbeda, mengevaluasi setiap konfigurasi, dan mengembalikan kombinasi dengan performa terbaik [22].

3.4.4.1. Implementasi SVM dengan Kernel Linier menggunakan Grid Search

Modifikasi dilakukan dengan menambahkan operator 'Optimize Parameters (Grid)'. Pada tahap ini ditambahkan parameter yang akan dioptimasi, yaitu nilai C dengan nilai min = 0.1, max = 100, steps = 10, scale = logaritmik (legacy), dan tipe kernel diatur menjadi linear. Di dalam operator Optimize Parameters (Grid), dilakukan pemisahan data, pelatihan model SVM, penerapan model, dan evaluasi kinerja. Model dilatih dengan kombinasi hiperparameter yang sedang diuji [23].



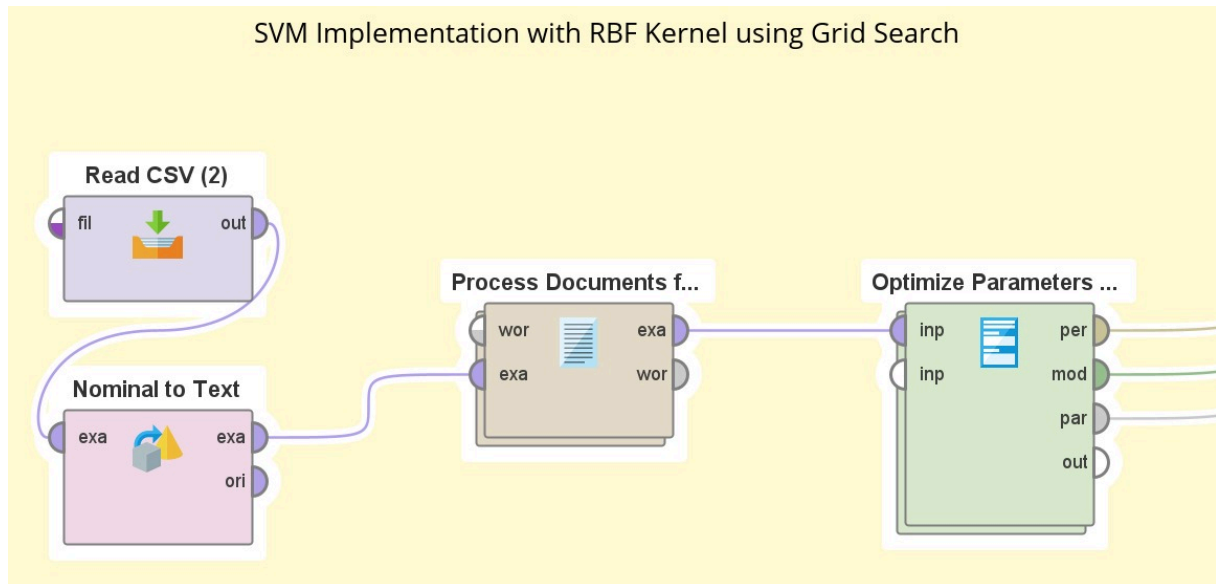
Gambar 8. Implementasi SVM dengan Kernel Linier menggunakan Grid Search



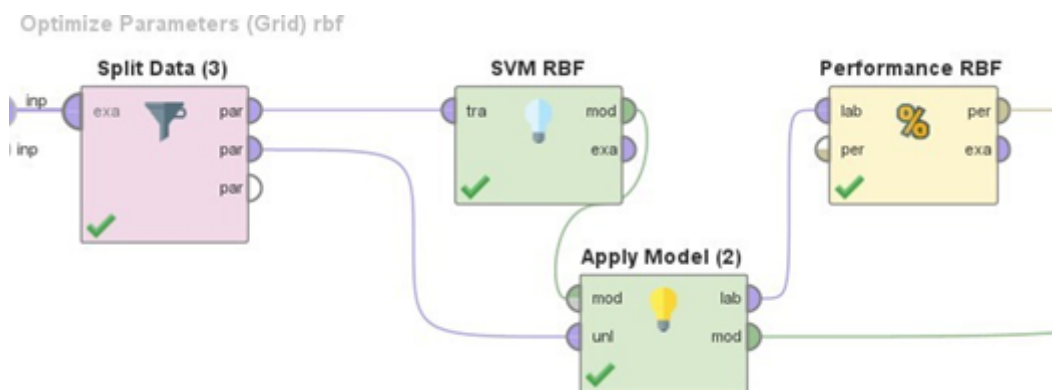
Gambar 9. Mengoptimalkan Parameter (Grid) pada SVM dengan Proses Kernel Linier

3.4.4.2. Implementasi SVM dengan Kernel RBF menggunakan Grid Search

Modifikasi pada tahap ini sama seperti pada implementasi SVM dengan kernel linear menggunakan Grid Search. Namun, parameter yang akan dioptimalkan adalah C dengan $\min = 0.1$, $\max = 100$, $\text{steps} = 10$, $\text{scale} = \text{logaritmik (legacy)}$, γ (gamma) dengan $\min = 0.0001$, $\max = 0.01$, $\text{steps} = 50$, dan $\text{scale} = \text{logaritmik (legacy)}$ serta tipe kernel diatur ke RBF.



Gambar 10. Implementasi SVM dengan Kernel RBF Implementasi SVM dengan Kernel RBF menggunakan Grid Search



Gambar 11. Mengoptimalkan Parameter (Grid) pada SVM dengan Proses Kernel RBF

3.5. Pengujian

Pada tahap pengujian, model yang telah dilatih dan dievaluasi dengan menggunakan data uji untuk menilai kinerjanya dalam mengklasifikasikan data yang belum pernah diproses sebelumnya. Pengujian ini dilakukan untuk memastikan bahwa model memiliki generalisasi yang baik dan dapat memberikan hasil yang akurat pada data yang baru. Performa model dievaluasi dengan menggunakan metrik seperti akurasi, presisi, dan recall yang disajikan dalam bentuk confusion matrix.

3.5.1. Naive Bayes

Hasil perhitungan akurasi dataset dengan menggunakan metode Naive Bayes didapatkan nilai akurasi sebesar 83.15%. Penulis melabeli data sebagai sentimen positif dan sentimen negatif secara manual [24].

Tabel 4. Confussion matrix Naive Bayes

Akurasi: 83,15%

	true Negative	true Positive	true Neutral	class Precision
pred. Negative	28	11	3	66.67%
pred. Positive	3	181	21	88.29%
pred. Neutral	1	6	13	65.00%
class recall	87.50%	91.41%	35.14%	

Dari tabel confusion matrix di atas, model ini memiliki presisi 66.67% untuk negatif, 88.29% untuk positif, 65.00% untuk netral, dan recall 87.50% untuk negatif, 91.41% untuk positif, dan 35.14% untuk netral. Model ini efektif dalam mengklasifikasikan sentimen positif tetapi kurang akurat dalam mengklasifikasikan sentimen netral.

3.5.2. Naive Bayes menggunakan SMOTE

Tabel 5 menunjukkan confusion matrix dari Naive Bayes dengan SMOTE. Integrasi SMOTE dengan Naive Bayes secara signifikan meningkatkan kinerja klasifikasi, yang dibuktikan dengan peningkatan akurasi yang signifikan menjadi 91,69%, dibandingkan dengan model tanpa SMOTE.

Tabel 5. Confussion matrix Naive Bayes menggunakan SMOTE

Akurasi: 91,69%

	true Negative	true Positive	true Neutral	class Precision
pred. Negative	198	4	5	95.65%
pred. Positive	0	184	17	91.54%
pred. Neutral	0	10	15	60.00%
class recall	100.00%	92.93%	40.54%	

Dari tabel confusion matrix di atas, model ini memiliki presisi 95.65% untuk negatif, 91.54% untuk positif, dan 60.00% untuk netral, serta recall 100.00% untuk negatif, 92.93% untuk positif, dan 40.54% untuk netral. Hal ini membuktikan bahwa Naive Bayes tanpa SMOTE dapat menangani data yang tidak seimbang secara efektif dan secara akurat mengklasifikasikan contoh positif, menekankan kemampuan SMOTE dalam meningkatkan kinerja klasifikasi untuk tugas-tugas analisis sentimen [25].

3.5.3. Support Vector Machine

Algoritma SVM bekerja dengan menemukan hyperplane optimal yang memisahkan kelas-kelas data dalam ruang fitur. Optimalisasi hyperplane dapat ditentukan dengan mengukur jarak antara hyperplane dengan pola terdekat dalam setiap kategori [26]. Berikut adalah hasil pengujian SVM dengan kernel Linear dan kernel RBF.

3.5.3.1. SVM dengan Kernel Linier

Pengujian SVM dengan kernel Linear menghasilkan akurasi sebesar 85.77%. Matriks kebingungan untuk SVM dengan kernel linear menunjukkan distribusi prediksi pada Tabel 6.

Tabel 6. Confusion Matrix SVM dengan Kernel Linier

Akurasi: 85,77%

	true Negative	true Positive	true Neutral	class Precision
pred. Negative	21	0	0	100.00%
pred. Positive	10	196	25	84.85%
pred. Neutral	1	2	12	80.00%
class recall	65.62%	98.99%	32.43%	

Model tersebut menunjukkan presisi sebesar 100.00% untuk negatif, 84.85% untuk positif, dan 80.00% untuk netral, serta recall sebesar 65.62% untuk negatif, 98.99% untuk positif, dan 32.43% untuk netral. Hasil ini menunjukkan bahwa model ini sangat baik dalam mengidentifikasi data positif, tetapi mengalami kesulitan dalam mengklasifikasikan data negatif dan netral, terutama dalam hal recall untuk kelas netral. Meskipun demikian, presisi yang tinggi untuk kelas negatif menunjukkan bahwa ketika model memprediksi negatif, prediksinya sangat akurat.

3.5.3.2. SVM dengan Kernel RBF

Hasil pengujian SVM dengan kernel Radial Basis Function (RBF) menunjukkan akurasi keseluruhan sebesar 83,52% seperti yang ditunjukkan pada Tabel 7.

Tabel 7. Confusion Matrix SVM dengan kernel RBF

Akurasi: 83,52%

	true Negative	true Positive	true Neutral	class Precision
pred. Negative	24	0	1	96.00%
pred. Positive	8	197	34	82.43%
pred. Neutral	0	1	2	66.67%

class recall	75.00%	99.49%	5.41%	
---------------------	--------	--------	-------	--

Dapat dilihat bahwa model memiliki presisi sebesar 96.00% untuk negatif, 82.43% untuk positif, dan 66.67% untuk netral, serta recall sebesar 75.00% untuk negatif, 99.49% untuk positif, dan 5.41% untuk netral, hasil ini menunjukkan bahwa model sangat efektif dalam mengklasifikasikan sentimen positif namun kurang berhasil dalam mengklasifikasikan sentimen netral. Meskipun demikian, presisi yang tinggi untuk kelas negatif menunjukkan kemampuan model untuk mengidentifikasi data negatif dengan baik.

3.5.4. Support Vector Machine menggunakan Penyetelan Hyperparameter

Penggunaan Support Vector Machine (SVM) dengan tuning hyperparameter melalui Grid Search dapat meningkatkan akurasi model secara signifikan. Pada bagian ini, akan dibahas pengujian SVM dengan dua jenis kernel, yaitu kernel Linear dan kernel Radial Basis Function (RBF), serta hasil tuning hyperparameter menggunakan Grid Search.

3.5.4.1. SVM dengan Kernel Linier menggunakan Pencarian Grid

Hasil pengujian SVM dengan kernel Linear yang telah dioptimasi menggunakan Grid Search menunjukkan akurasi sebesar 90.26%.

Tabel 8. Confusion Matrix SVM dengan kernel Linear menggunakan Grid Search

Akurasi: 90,26%

	true Negative	true Positive	true Neutral	class Precision
pred. Negative	21	0	0	100.00%
pred. Positive	10	196	25	84.85%
pred. Neutral	1	2	12	80.00%
class recall	65.62%	98.99%	32.43%	

Dapat dilihat bahwa model memiliki presisi 100.00% untuk negatif, 89.45% untuk positif, dan 85.71% untuk netral, serta recall 87.50% untuk negatif, 98.48% untuk positif, dan 48.65% untuk netral, hasil ini menunjukkan bahwa optimasi Grid Search meningkatkan performa model secara signifikan. Dari sebelumnya akurasi SVM tanpa optimasi hanya sebesar 85.77%.

Tabel 9. Mengoptimalkan Parameter (Grid) Linear

Iterasi	SVM.kernel_type	SVM.C	accuracy
1	linear	0.100	0.805
2	linear	0.686	0.828

3	linear	1.616	0.865
4	linear	3.092	0.884
5	linear	5.432	0.869
6	linear	9.145	0.861
7	linear	15.043	0.891
8	linear	24.377	0.869
9	linear	39.197	0.858
10	linear	62.707	0.843
11	linear	100	0.903

Tabel 9 menunjukkan iterasi tuning hyperparameter untuk SVM dengan kernel Linear. Pada tahap ini, dilakukan 11 kali iterasi mulai dari $C = 0.1$ hingga $C = 100$. Pada setiap iterasi terjadi peningkatan dan penurunan nilai akurasi. Peningkatan akurasi terbesar terjadi pada iterasi ke-2 ke iterasi ke-3, yaitu sebesar 4.47%. Sedangkan penurunan iterasi terbesar terjadi pada iterasi ke-7 ke iterasi ke-8, yaitu sebesar 2,47%. Dengan demikian, rentang persen kenaikan nilai tersebut adalah dari -2.47% hingga +4.47%. Dapat dilihat bahwa nilai C yang optimal terdapat pada iterasi ke-11 dengan nilai 100 dan akurasi sebesar 90,3%.

3.5.4.2. SVM dengan Kernel RBF menggunakan Grid Search

Hasil pengujian SVM dengan kernel RBF yang telah dioptimasi menggunakan Grid Search menunjukkan akurasi sebesar 85.77%.

Tabel 10. Confusion Matrix SVM dengan kernel RBF

Akurasi: 85.77%

	true Negative	true Positive	true Neutral	class Precision
pred. Negative	25	0	0	100.00%
pred. Positive	7	198	31	84.85%
pred. Neutral	0	0	6	80.00%
class recall	78.12%	100.00%	16.22%	

Confusion matrix menunjukkan precision sebesar 100.00% untuk negatif, 83.90% untuk positif, dan 100.00% untuk netral, serta recall sebesar 78.12% untuk negatif, 100.00% untuk positif, dan 16.22% untuk netral, model ini mampu mengklasifikasikan sentimen positif dan negatif dengan baik namun kurang berhasil dalam mengklasifikasikan sentimen netral. Meskipun memiliki akurasi yang lebih rendah dibandingkan dengan model SVM linier setelah dioptimasi dengan Grid Search, model ini masih memberikan hasil yang cukup baik.

Tabel 11. Mengoptimalkan Parameter (Grid) RBF

Iterasi	SVM.C	SVM.kernel_type	SVM.gamma	Akurasi
1	0.100	rbf	0.0001	0.742
2	0.686	rbf	0.0001	0.779
3	1.616	rbf	0.0001	0.757
4	3.092	rbf	0.0001	0.794
5	5.432	rbf	0.0001	0.772
6	9.145	rbf	0.0001	0.768
7	15.034	rbf	0.0001	0.757
8	24.377	rbf	0.0001	0.757
9	39.197	rbf	0.0001	0.722
10	62.707	rbf	0.0001	0.734
...
561	100	rbf	0.010	0.801

Tabel 11 menunjukkan iterasi tuning hyperparameter untuk SVM dengan kernel RBF. Pada SVM dengan kernel RBF menggunakan Grid Search, dilakukan iterasi sebanyak 561 kali mulai dari $C = 0.1$ hingga $C = 100$ dan $\gamma(\text{gamma}) = 0.0001$ hingga $\gamma(\text{gamma}) = 0.01$. Terlihat bahwa nilai C dan nilai γ (gamma) yang optimal terdapat pada iterasi ke-561 dengan nilai 100 untuk C dan 0,01 untuk γ (gamma). Dengan demikian, diperoleh akurasi sebesar 80,1%.

4. DISKUSI

Penelitian ini membandingkan hasil pengujian algoritma SVM dan algoritma Naive Bayes. Pada evaluasi penelitian ini, dilakukan pencarian pada sistem yang dibangun untuk menemukan parameter terbaik dari kedua kernel SVM [27], kemudian hasil dari masing-masing kernel akan dibandingkan satu sama lain.

4.1 Perbandingan Algoritma

Penelitian ini mengevaluasi kedua algoritma tersebut dengan berbagai pengaturan, termasuk menerapkan SMOTE untuk menangani ketidakseimbangan kelas dalam Naive Bayes dan menggunakan Hyperparameter Tuning untuk SVM untuk mengoptimalkan parameter kernel. Melalui analisis metrik kinerja, penelitian ini bertujuan

Pada pengujian algoritma Naive Bayes tanpa menggunakan SMOTE dan Naive Bayes dengan menggunakan SMOTE, akurasi terbesar didapatkan pada Naive Bayes dengan menggunakan SMOTE

yaitu sebesar 91.69%. Pada pengujian algoritma SVM dengan kernel linear dan SVM dengan kernel RBF tanpa menggunakan Hyperparameter Tuning, akurasi terbesar didapatkan pada SVM dengan kernel linear yaitu sebesar 85.77%. Sedangkan pada pengujian algoritma SVM dengan kernel linear dan SVM dengan kernel RBF menggunakan Hyperparameter Tuning melalui Grid Search, akurasi terbesar didapatkan pada SVM dengan kernel linear yaitu sebesar 90.26%. Setelah dilakukan pengujian SVM menggunakan Hyperparameter Tuning dengan metode Grid Search, dapat dilihat bahwa pengaturan nilai min, max, dan step pada C dan γ (gamma) sangat mempengaruhi peningkatan akurasi. Kemudian Dari semua pengujian yang dilakukan, hasil dari algoritma Naive Bayes dengan menggunakan metode SMOTE memiliki nilai akurasi yang paling tinggi dengan nilai 91.69%.

untuk mengidentifikasi algoritma dan konfigurasi yang mencapai akurasi tertinggi.

4.1.1 Perbandingan Akurasi Naive Bayes dan Naive Bayes menggunakan SMOTE

Bagian ini berfokus pada perbandingan akurasi algoritma Naive Bayes ketika diterapkan secara langsung pada dataset dan ketika dikombinasikan dengan Synthetic Minority Over-sampling Technique (SMOTE).

Tabel 12. Perbandingan Akurasi dalam Naive Bayes

Algoritma	Akurasi
Naive Bayes	83.15%
Naive Bayes menggunakan SMOTE	91.69%

Hasilnya menunjukkan peningkatan akurasi yang signifikan ketika SMOTE diterapkan. Secara khusus, Naive Bayes tanpa SMOTE mencapai akurasi 83,15%, sedangkan Naive Bayes dengan SMOTE mencapai akurasi yang jauh lebih tinggi, yaitu 91,69%. Perbandingan ini menyoroti keefektifan SMOTE dalam meningkatkan performa algoritma Naive Bayes dengan memastikan representasi kelas yang lebih seimbang.

4.1.2 Perbandingan Akurasi pada SVM dan SVM menggunakan Hyperparameter Tuning

Bagian ini berfokus pada perbandingan akurasi SVM dengan algoritma kernel Linear dan RBF ketika diterapkan secara langsung pada dataset dan ketika dikombinasikan dengan Hyperparameter Tuning menggunakan Grid Search.

Tabel 13. Perbandingan Akurasi dalam SVM

	SVM	Tipe Kernel	Akurasi
Tanpa menggunakan Hyperparameter Tuning	Linear	Linear	85.77%
	Non-Linear	RBF	83.52%
Menggunakan Hyperparameter Tuning	Linear	Linear	90.26%
	Non-Linear	RBF	85.77%

Hasil penelitian menunjukkan bahwa penerapan tuning hyperparameter melalui Grid Search secara signifikan meningkatkan akurasi algoritma SVM, terutama untuk kernel Linear. Sebelum dilakukan tuning hyperparameter, SVM dengan kernel Linear mencapai akurasi 85.77%, yang meningkat menjadi 90.26% setelah dilakukan tuning. Demikian pula, SVM dengan kernel RBF menunjukkan peningkatan akurasi dari 83,52% menjadi 85,77% setelah tuning.

4.1.3 Perbandingan Akurasi Algoritma Naive Bayes yang Dimodifikasi dan SVM yang Dimodifikasi

Bagian ini berfokus pada perbandingan akurasi algoritma Naive Bayes menggunakan SMOTE dengan akurasi algoritma SVM menggunakan tuning hyperparameter.

Tabel 14. Perbandingan Akurasi antara Naive Bayes Termodifikasi dan SVM Termodifikasi

Algoritma	Accuracy
Naive Bayes menggunakan SMOTE	91.69%
SVM menggunakan Hyperparameter tuning (Linear)	90.26%
SVM menggunakan Hyperparameter tuning (RBF)	85.77%

Perbandingan tersebut menunjukkan bahwa algoritma Naive Bayes menggunakan SMOTE mencapai akurasi tertinggi sebesar 91,69%, melampaui SVM dengan kernel Linear (90,26%) dan SVM dengan kernel RBF (85,77%) setelah penyetelan hyperparameter. Hal ini menunjukkan bahwa untuk tugas analisis sentimen khusus ini, kombinasi Naive Bayes dengan SMOTE lebih efektif daripada SVM, bahkan ketika SVM dioptimalkan dengan tuning hyperparameter. Hal ini menunjukkan bahwa algoritma Naive Bayes mendapat manfaat yang signifikan dari penanganan ketidakseimbangan kelas melalui SMOTE, yang menghasilkan kinerja keseluruhan yang lebih unggul dibandingkan dengan algoritma SVM.

4.2 Penelitian Sebelumnya

Rian dkk. [28] melakukan penelitian tentang analisis sentimen terhadap edukasi e-sports dengan menggunakan algoritma Naive Bayes dan Support Vector Machine. Masalah yang diidentifikasi adalah mengukur opini terhadap pendidikan e-sports dan memisahkan sentimen positif dan negatif. Hasil penelitian menunjukkan bahwa algoritma Naive Bayes dengan SMOTE mencapai akurasi yang lebih tinggi daripada Support Vector Machine, dengan nilai akurasi 70.32% dan nilai AUC 0.954. Hal ini menunjukkan potensi Naive Bayes dalam memprediksi pencapaian e-sports untuk kurikulum pendidikan pembelajaran siswa dengan lebih baik. Yerik dkk. [29] membahas tentang analisis sentimen publik terhadap isu over-tourism pada konten pariwisata yang bertanggung jawab dengan menggunakan Naive Bayes Classifier. Penelitian ini bertujuan untuk memahami pandangan masyarakat tentang dampak negatif dari pariwisata yang berlebihan. Hasil penelitian menunjukkan bahwa Naive Bayes Classifier menunjukkan akurasi yang tinggi yaitu 84,82% dengan augmentasi SMOTE memberikan performa yang unggul dalam mengklasifikasikan sentimen terkait isu pariwisata berlebihan pada konten pariwisata bertanggung jawab. Sama halnya dengan penelitian sebelumnya, pada penelitian kami, setelah melakukan pengujian terhadap Naive Bayes menggunakan SMOTE dan SVM menggunakan Hyperparameter Tuning, algoritma yang mendapatkan akurasi tertinggi adalah Naive Bayes menggunakan SMOTE dengan akurasi sebesar 91,69%.

5. KESIMPULAN

Penelitian ini menunjukkan implementasi Synthetic Minority Over-sampling Technique (SMOTE) pada algoritma Naive Bayes Classifier dan implementasi Hyperparameter Tuning dengan

metode Grid Search pada Support Vector Machine (SVM) yang diaplikasikan untuk menganalisa hasil akurasi dari setiap pengujian. Hasil penelitian menunjukkan bahwa algoritma Naive Bayes, ketika disempurnakan dengan SMOTE, mengungguli SVM dengan Hyperparameter Tuning dalam tugas analisis sentimen pada data Twitter mengenai pemilihan presiden Indonesia. Pada awalnya, algoritma Naive Bayes dikenal dengan kesederhanaan dan efisiensinya dalam tugas klasifikasi, terutama dengan data teks, karena algoritma ini mengasumsikan independensi fitur dan menghitung probabilitas untuk setiap kelas. Namun, algoritma ini dapat kesulitan dengan set data yang tidak seimbang. Di sisi lain, SVM adalah metode klasifikasi yang kuat yang berusaha menemukan hyperplane optimal yang memisahkan kelas-kelas yang berbeda, sehingga efektif untuk data berdimensi tinggi tetapi sensitif terhadap pengaturan parameter dan data yang tidak seimbang.

Dengan mengatasi masalah ketidakseimbangan kelas, SMOTE secara signifikan meningkatkan akurasi algoritma Naive Bayes, mencapai 91,69% dibandingkan dengan akurasi 90,26% dari SVM dengan kernel linier dan akurasi 85,77% dari SVM dengan metode Hyperparameter Tuning dengan metode Grid Search yang diimplementasikan dengan kernel RBF. Temuan ini menunjukkan bahwa untuk tugas analisis sentimen yang serupa, algoritma Naive Bayes dengan SMOTE merupakan pendekatan yang lebih efektif daripada SVM, bahkan dengan teknik penyetelan hyperparameter yang canggih.

REFERENSI

- [1] O. Manullang, C. Prianto, and N. H. Harani, "Analisis Sentimen Untuk Memprediksi Hasil Pemilihan Calon Presiden Menggunakan Lexicon Based Dan Random Forest," *J. Ilm. Inform.* vol. 11, no. 02, pp. 159-169, 2023, doi: 10.33884/jif.v11i02.7987.
- [2] A. F. Hidayatullah, S. Cahyaningtyas, dan A. M. Hakim, "Analisis Sentimen pada Twitter menggunakan Neural Network: Indonesian Presidential Election 2019 Dataset," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012001, 2021, doi: 10.1088/1757-899x/1077/1/012001.
- [3] N. Hayatin, G. I. Marthasari, dan L. Nuraini, "Optimasi Analisis Sentimen Pemilihan Presiden Indonesia menggunakan Naïve Bayes dan Particle Swarm Optimization," *J. Online Inform.*, vol. 5, no. 1, hal. 81-88, 2020, doi: 10.15575/join.v5i1.558.
- [4] B. B. Pranata and Susanti, "Support Vector Machine untuk Analisis Sentimen Bakal Calon Presiden Republik Indonesia 2024," *Indones. J. Comput. Sains*, vol. 12, no. 3, pp. 1335-1349, 2023, doi: 10.33022/ijcs.v12i3.3231.
- [5] Y. Findawati, U. Indahyanti, Y. Rahmawati, dan R. Puspitasari, "Analisis Sentimen Calon Presiden Potensial 2024: Sebuah Studi Berbasis Twitter," *Acad. Open*, vol. 8, no. 1, pp. 1-17, 2023, doi: 10.21070/acopen.8.2023.7138.
- [6] G. Sanjaya and K. M. Lhaksana, "Analisis Sentimen Komentar YouTube tentang Terpilihnya Menteri Kabinet Indonesia Maju Menggunakan Lexicon Based Gupta," *e-Proceeding Eng.*, vol. 7, no. 3, pp. 9698-9710, 2020.
- [7] B. M. Iqbal, K. M. Lhaksana, and E. B. Setiawan, "Analisis Sentimen Pemilihan Presiden 2024 pada Media Berita Menggunakan Support Vector Machine," *J. Comput. Sist. Informatika*, vol. 4, no. 2, pp. 397-404, 2023, doi: 10.47065/josyc.v4i2.3051.
- [8] L. Damayanti dan K. M. Lhaksana, "Analisis Sentimen Pemilihan Presiden Indonesia 2024 di Twitter," *Sinkron*, vol. 8, no. 2, hlm. 938-946, 2024, doi: 10.33395/sinkron.v8i2.13379.
- [10] Samsir dkk., "Algoritma Naives Bayes untuk Analisis Sentimen Twitter," *J. Phys. Conf. Ser.* vol. 1933, no. 1, 2021, doi: 10.1088/1742-6596/1933/1/012019.
- [11] H. H. A. Santoso, E. H. Rachmawanto, A. Nugraha, A. A. Nugroho, D. R. I. M. Setiadi, and R. S. Basuki, "Klasifikasi hoax dan analisis sentimen berita berbahasa Indonesia menggunakan optimasi Naive Bayes," *Telkomnika (Telecommunication Comput. Elektron. Control)*, vol. 18, no. 2, pp. 799- 806, 2020, doi: 10.12928/TELKOMNIKA.V18I2.14744.
- [12] A. Guswandri, R. P. Cahyono, S. I. Akutansi, and T. Komputer, "Penerapan Sentimen Analis

- Menggunakan Metode Naïve Bayes Dan Svm," *Ilmudata.org*, vol. 2, no. 12, pp. 2022-2023, 2022.
- [13] L. Luthfiana, J. C. Young, and A. Rusli, "Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi," *Ultim. J. Tek. Inform.* vol. 12, no. 2, pp. 125- 126, 2020, doi: 10.31937/ti.v12i2.1828.
- [14] C. B. Sonjaya, A. Fitri, N. Masruriyah, and D. Sulistya, "Perbandingan Performa Algoritma Klasifikasi Untuk Mendeteksi Penyakit Jantung," *Inf. Syst. J.*, vol. 5, no. 2, pp. 166-175, 2022.
- [15] Hermanto, A. Y. Kuntoro, T. Asra, E. B. Pratama, L. Effendi, dan R. Ocanitra, "Analisis Sentimen Pengguna Gojek dan Grab di Google Play Menggunakan Algoritma Naive Bayes dan Teknik Smote Berbasis Support Vector Machine," *J. Phys. Conf. Ser.* vol. 1641, no. 1, 2020, doi: 10.1088/1742- 6596/1641/1/012102.
- [16] A. H. Ali dan M. Z. Abdullah, "Optimasi grid paralel dari hyperparameter SVM untuk klasifikasi data besar menggunakan spark radoop," *Karbala Int. J. Mod. Sci.*, vol. 6, no. 1, 2020, doi: 10.33640/2405-609X.1270.
- [17] D. M. Alvina Felicia Watratan, Arwini Puspita. B, "Implementasi Algoritma Naive Bayes Untuk Memprediksi Tingkat Penyebaran Covid-19 Di Indonesia," *J. Appl. Comput. Sci. Technol.*, vol. 1, no. 1, pp. 7-14, 2020, doi: 10.55606/jurritek.v1i1.127.
- [18] Y. A. Singgalen, "Evaluasi Kinerja Model Klasifikasi Sentimen: Sebuah Perbandingan Kajian NBC, SVM, dan DT dengan SMOTE," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 5, pp. 2539-2548, 2024, doi: 10.30865/klik.v4i5.1827.
- [19] Dhina Nur Fitriana dan Yuliant Sibaroni, "Analisis Sentimen pada Postingan Twitter KAI Menggunakan Multiclass Support Vector Machine (SVM)," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 5, pp. 846-853, 2020, doi: 10.29207/resti.v4i5.2231.
- [20] A. F. Rochim, K. Widyaningrum, and D. Eridani, "Perbandingan Fungsi Kernel antara Linear, Radial Basis dan Polinomial pada Metode Support Vector Machine untuk Analisis Sentimen COVID-19," *Minist. Res. Technol.*, pp. 224-228, 2021.
- [21] A. Nurkholis, D. Alita, dan A. Munandar, "Perbandingan Kernel Support Vector Machine Multi- Class pada Analisis Sentimen PPKM di Twitter," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 2, hal. 227-233, 2022, doi: 10.29207/resti.v6i2.3906.
- [24] T. A. Amini and K. Setiawan, "Penerapan Algoritma Naive Bayes pada Analisis Sentimen Twitter Calon Wakil Presiden 2024 Gibran Rakabuming Raka menggunakan Rapidminer," *Int. J. Softw. Eng. Comput. Sci.*, vol. 4, no. 1, pp. 234-246, 2024, doi: 10.35870/ijsecs.v4i1.2236.
- [25] Y. A. Singgalen, "Analisis Jejaring Sosial dan Klasifikasi Sentimen Konten Produk Extended Reality," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 4, pp. 2197-2208, 2024, doi: 10.30865/klik.v4i4.1710.
- [26] M. Daffa Dhiyaulhaq dan P. Harry Gunawan, "Analisis Sentimen Terhadap Proyek Kereta Api Cepat Jakarta-Bandung Menggunakan Metode SVM," *J. Media Inform. Budidarma*, vol. 7, no. 4, pp. 2128- 2136, 2023, doi: 10.30865/mib.v7i4.6855.
- [27] Y. Farida, N. Ulinnuha, S. K. Sari, dan L. N. Desinaini, "Membandingkan Support Vector Machine dan Metode Naïve Bayes dengan Pemilihan Fitur Filter Berbasis Korelasi Cepat pada Mendeteksi Penyakit Parkinson," *Lontar Komput. J. Ilm. Teknol. Inf.* vol. 14, no. 2, pp. 80-90, 2023, doi: 10.24843/lkjiti.2023.v14.i02.p02.
- [28] R. Ardianto, T. Rivanie, Y. Alkhalifi, F. S. Nugraha, dan W. Gata, "ANALISIS SENTIMEN TENTANG E-SPORTS UNTUK KURIKULUM PENDIDIKAN MENGGUNAKAN NAIVE BAYES DAN DUKUNGAN MESIN VEKTOR," *J. Comput. Sci. Inf.*, vol. 2, no. analisis sentimen, hal. 13, 2020.
- [29] Y. Afrianto Singgalen, "Klasifikasi Sentimen Isu Pariwisata Berlebih di Konten Pariwisata menggunakan Naïve Bayes Classifier," *J. Comput. Syst. Informatika*, vol. 5, no. 2, pp. 275- 285, 2024, doi: 10.47065/josyc.v5i2.4904.
-

