

Comparison of Naive Bayes and Support Vector Machine Algorithms in Sentiment Analysis

DOI: <http://dx.doi.org/10.35889/jutisi.v14i2.2727>

Creative Commons License 4.0 (CC BY – NC) 

Tiara Sela^{1*}, Anisya Sonita²

Teknik Informatika, Universitas Muhammadiyah Bengkulu, Bengkulu, Indonesia

*e-mail *Corresponding Author*: shellasprn@gmail.com

Abstract

Twitter is a social media platform that quickly spreads public opinion. The Ronald Tannur case, widely discussed on this platform, triggered various public reactions. This study aims to compare the Naive Bayes and Support Vector Machine (SVM) algorithms in analyzing netizens' sentiment toward the case. The process includes data collection and text preprocessing, such as removing duplicates, cleaning, case folding, word normalization, stopword removal, tokenization, and stemming. The text is then transformed using TF-IDF. Sentiment classification is performed using both algorithms and evaluated through metrics like accuracy, precision, recall, F1-score, and AUC. The models are tested on three data split schemes: 90:10, 80:20, and 70:30. The results show that SVM with 90% training data provides the best performance, achieving 88.78% accuracy and 0.84 AUC, outperforming Naive Bayes, which only reached 71.29% accuracy and 0.77 AUC. This shows that SVM is more accurate in detecting sentiment on social media.

Keywords : *Sentiment Classification; Naive Bayes Algorithm; Support Vector Machine Algorithm; Twitter; Python*

Abstrak

Twitter merupakan media sosial yang cepat menyebarkan opini publik. Kasus Ronald Tannur, yang banyak dibicarakan di platform ini, memicu beragam reaksi dari masyarakat. Penelitian ini bertujuan untuk membandingkan algoritma *Naive Bayes* dan *Support Vector Machine* (SVM) dalam menganalisis sentimen warganet terhadap kasus tersebut. Proses analisis mencakup pengumpulan data, praproses teks seperti penghapusan duplikat, pembersihan, *case folding*, normalisasi kata, penghapusan *stopword*, tokenisasi, dan stemming, lalu data ditransformasi menggunakan TF-IDF. Klasifikasi sentimen dilakukan menggunakan kedua algoritma dan dievaluasi dengan metrik seperti akurasi, presisi, *recall*, *F1-score*, dan AUC. Pengujian dilakukan pada tiga skema pembagian data latih dan uji, yaitu 90:10, 80:20, dan 70:30. Hasil menunjukkan bahwa SVM dengan rasio data latih 90% memberikan hasil terbaik dengan akurasi 88,78% dan AUC 0,84, melampaui *Naive Bayes* yang hanya mencapai akurasi 71,29% dan AUC 0,77. Ini menunjukkan bahwa SVM lebih akurat dalam mengenali sentimen di media sosial.

Kata kunci: *Klasifikasi Sentimen; Algoritma Naive Bayes; Algoritma Support Vector Machine; Twitter; Python*

1. Pendahuluan

Seiring kemajuan teknologi, media sosial kini menjadi bagian penting dalam kehidupan sehari-hari. Melalui platform ini, orang dapat berbagi informasi, berdiskusi, dan menyampaikan pendapat mereka tentang berbagai peristiwa yang sedang terjadi[1]. Dalam beberapa tahun terakhir, peran media sosial telah berkembang pesat, tidak lagi sekadar menjadi sarana komunikasi, tetapi juga menjadi medium utama dalam membentuk opini publik serta mendistribusikan berbagai jenis informasi, baik yang bersifat faktual maupun bersifat spekulatif[2]. Di antara platform yang ada, *Twitter*, yang kini dikenal sebagai "X", memiliki peran yang signifikan karena kemampuannya dalam menyebarkan informasi secara langsung dan

membuka ruang bagi diskusi publik secara luas. Dengan karakteristiknya yang cepat dalam menyebarkan berita, Twitter banyak digunakan untuk mengevaluasi sentimen masyarakat terkait dengan isu-isu sosial, politik, hukum, dan ekonomi[3].

Kasus Ronald Tannur menjadi perhatian luas di platform *Twitter*, memunculkan beragam reaksi dari warganet. Tanggapan yang muncul bervariasi, mulai dari dukungan, kritik, hingga sikap netral. Banyaknya komentar yang tersebar menunjukkan bahwa kasus ini memiliki nilai penting untuk diteliti lebih lanjut melalui pendekatan analisis sentimen. Metode ini bertujuan untuk menggali persepsi atau pandangan publik terhadap suatu isu dengan cara mengelompokkan opini menjadi tiga kategori utama, yakni positif, negatif, dan netral [4]. Pendekatan ini berguna dalam memahami cara masyarakat berpikir dan bagaimana isu tertentu diterima serta dibahas di media sosial.

Pada penelitian ini, diterapkan dua algoritma pembelajaran mesin, yakni *Support Vector Machine* serta *Naive Bayes*, dalam rangka membangun model klasifikasi sentimen terkait dengan kasus Ronald Tannur. Masing-masing algoritma tersebut memiliki karakteristik dan pendekatan analisis yang berbeda. *SVM* dikenal unggul dalam menangani data berdimensi tinggi dan mampu membentuk model klasifikasi yang akurat, khususnya saat dihadapkan pada pola teks yang kompleks dan beragam [5]. Sementara itu, *Naive Bayes* merupakan metode klasifikasi yang didasarkan pada pendekatan *probabilistik*. Meskipun memiliki struktur yang relatif sederhana, algoritma ini terbukti efektif dan cepat dalam memproses data berukuran besar[6]. Oleh karena itu, membandingkan kedua algoritma ini dalam konteks analisis sentimen di *Twitter* sangat penting untuk memilih metode yang lebih optimal dalam mengklasifikasikan *opini public*.

Penelitian ini bertujuan untuk membandingkan efektivitas algoritma *Naive Bayes* dan *Support Vector Machine* dalam menganalisis sentimen warganet di *Twitter* terkait kasus Ronald Tannur. Untuk mengetahui algoritma mana yang lebih baik, digunakan beberapa ukuran penilaian seperti *akurasi*, *presisi*, *recall*, *F1-score*, dan *AUC*. Hasil penelitian ini diharapkan dapat memberikan gambaran algoritma mana yang lebih efisien dalam mengklasifikasikan opini masyarakat, serta dapat menjadi dasar dalam pengembangan sistem pemantauan sentimen publik di media sosial.

2. Tinjauan Pustaka

Penelitian terdahulu telah banyak mengkaji penerapan algoritma klasifikasi dalam analisis sentimen, khususnya terkait isu-isu sosial. Salah satu studi mengkaji kasus kejahatan *carding* dengan menggunakan algoritma *Naive Bayes* dan *SVM*. Hasil penelitian menunjukkan bahwa kedua algoritma mampu mengklasifikasikan data dengan akurasi tinggi, namun *SVM* lebih unggul dengan akurasi mencapai 99%, sedangkan *Naive Bayes* memperoleh 98% [7].

Studi lain melakukan analisis sentimen terhadap kasus korupsi bantuan sosial (*bansos*) beras selama masa pandemi *Covid-19* dengan memanfaatkan data dari *Twitter*. Dalam penelitian ini, algoritma *Naive Bayes* dan *SVM* diterapkan untuk mengukur respons publik. Sentimen negatif mendominasi dengan persentase 58,79%, mencerminkan ketidakpercayaan masyarakat terhadap proses distribusi bantuan. Dari sisi performa, *SVM* memperoleh akurasi 66,67%, lebih tinggi dibandingkan *Naive Bayes* yang mencatat 60,61% [8].

Penelitian selanjutnya mengangkat kasus pembunuhan dua remaja di Cirebon dan menggunakan pendekatan analisis sentimen dengan algoritma *Naive Bayes* dan *SVM*. Teknik *SMOTE* digunakan untuk menangani ketidakseimbangan data. Berdasarkan hasil pengujian, *SVM* kembali menunjukkan performa yang lebih baik dengan akurasi 97%, sementara *Naive Bayes* mencapai 84,86%. Mayoritas opini publik dikategorikan sebagai sentimen positif terhadap penanganan kasus tersebut [9].

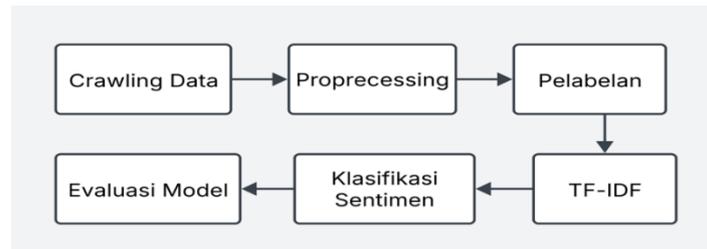
Sementara itu, analisis terhadap isu kekerasan dalam rumah tangga di *Twitter* juga telah dilakukan dengan menggunakan algoritma *Multinomial Naive Bayes* dan *SVM*. Penelitian ini menggunakan dua metode pelabelan, yaitu manual dan *Vader Lexicon*, serta beberapa rasio data latih dan uji. Pada pelabelan manual, akurasi tertinggi diperoleh pada rasio 80:20 dengan hasil 70% untuk *Naive Bayes* dan 73% untuk *SVM*. Sedangkan pada pelabelan *Vader Lexicon*, akurasi terbaik ditemukan pada rasio 60:40, yaitu 68% untuk *Naive Bayes* dan 76% untuk *SVM* [10].

Berdasarkan penelitian terdahulu yang telah dijelaskan penelitian ini memiliki perbedaan konsep yang jelas dibandingkan penelitian terdahulu. Dari sisi metode, penelitian ini membandingkan algoritma *SVM* dan *Naive Bayes* dalam analisis sentimen terhadap kasus sosial aktual, yakni kasus Ronald Tannur. Tidak hanya mengandalkan akurasi, penelitian ini juga menggunakan metrik evaluasi yang lebih lengkap seperti *precision*, *recall*, *F1-score*, dan *AUC*

untuk mengukur performa model secara menyeluruh. Selain itu, penelitian ini menguji tiga variasi rasio data (90:10, 80:20, dan 70:30) dan menggunakan dataset yang lebih besar, yakni 2.046 tweet. Kebaruan dalam penelitian ini terlihat dari kombinasi evaluasi yang lebih lengkap, penggunaan data yang sesuai dengan isu terkini, serta analisis pengaruh perbandingan rasio data latih dan uji, yang masih jarang dibahas dalam penelitian sebelumnya.

3. Metodologi

Tahapan penelitian ini meliputi pengambilan data, *preprocessing*, pelabelan, pemodelan, dan evaluasi, seperti yang digambarkan dibawah ini



Gambar 1. Langkah-langkah dalam Penelitian

3.1 Crawling data

Crawling atau Pengumpulan data dilakukan menggunakan *Google Colab* yang terhubung ke *Twitter API*, dengan menyalin *link* setiap tweet untuk memastikan data relevan dan akurat.

3.2 Preprocessing

Tahapan ini dilakukan untuk menyiapkan serta membersihkan data sebelum dianalisis dalam penelitian. Proses yang diterapkan mencakup penghapusan data duplikat, pembersihan data, konversi huruf menjadi bentuk seragam (*case folding*), normalisasi kata, penghilangan kata yang tidak penting (*stopwords*), pemisahan kata (*tokenisasi*), dan pengembalian kata ke bentuk dasar (*stemming*).

3.3 Pelabelan

Tahapan pelabelan bertujuan untuk mengidentifikasi data tweet berdasarkan nadanya, apakah termasuk positif, netral, atau negatif[11].

3.4 Transformasi Data (TF-IDF)

Tahap transformasi data dalam penelitian ini menggunakan metode *Term Frequency-Inverse Document Frequency*, yang berfungsi untuk mengubah teks yang telah diproses menjadi bentuk vektor numerik. Proses ini memungkinkan teks untuk diproses lebih lanjut oleh algoritma machine learning. Bobot kata dihitung berdasarkan seberapa sering kata tersebut muncul dalam satu dokumen dan seberapa jarang kata itu muncul di seluruh kumpulan dokumen. Bobot *TF-IDF* dihitung berdasarkan persamaan:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

Dengan:

$$TF(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \text{ dan } IDF(t) = \log\left(\frac{N}{1+df(t)}\right) + 1 \quad (2)$$

di mana:

- $f_{t,d}$ adalah jumlah kemunculan kata t dalam dokumen d ,
- $\sum_k f_{k,d}$ adalah jumlah total kata dalam dokumen d ,
- N adalah jumlah total dokumen,
- $df(t)$ adalah jumlah dokumen yang mengandung kata t .

Proses transformasi dilakukan secara terpisah untuk setiap kategori sentimen (positif, negatif, dan netral). Data yang telah ditransformasi kemudian disimpan dalam bentuk matriks *TF-IDF*, yang selanjutnya digunakan sebagai input dalam proses klasifikasi dengan metode *Naive Bayes* serta *SVM*.

3.5 Klasifikasi Sentimen

Tahap klasifikasi sentimen dalam penelitian ini bertujuan untuk membandingkan kinerja dua algoritma pembelajaran mesin, yaitu *Multinomial Naive Bayes* dan *Support Vector Machine*, dalam mengelompokkan tweet ke dalam tiga kategori sentimen: positif, negatif, dan netral. Kedua algoritma diterapkan pada data yang telah melalui tahap pra-proses, transformasi fitur menggunakan metode *TF-IDF*, serta penyeimbangan kelas menggunakan teknik *SMOTE*. Algoritma MNB dipilih karena kemampuannya yang baik dalam menangani data teks, terutama dalam menganalisis frekuensi kemunculan kata untuk klasifikasi.

Multinomial Naive Bayes adalah varian dari *Naive Bayes* yang umum digunakan untuk klasifikasi teks. Algoritma ini menggunakan pendekatan teori peluang (probabilitas) dan menganggap bahwa setiap kata dalam dokumen bersifat independen atau berdiri sendiri [12]. MNB menghitung kemungkinan sebuah teks termasuk ke dalam salah satu kategori sentimen dengan rumus:

$$P(c|d) = \frac{P(c) \prod_{i=1}^n p(w_i|c)}{P(d)} \quad (3)$$

Keterangan:

- $P(c|d)$: probabilitas teks d termasuk dalam kelas c
- $P(c)$: probabilitas awal kelas c
- $P(w_i|c)$: probabilitas kata ke- i muncul dalam kelas tersebut
- $P(d)$: probabilitas dari teks d secara keseluruhan (umumnya tidak berpengaruh dalam perbandingan antar kelas)

Sementara itu, *SVM* adalah algoritma klasifikasi yang bekerja dengan mencari garis pemisah terbaik (*hyperplane*) yang dapat memisahkan data ke dalam kelas berbeda secara optimal. Tujuannya adalah memaksimalkan jarak antara data dari masing-masing kelas dengan rumus:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{dengan syarat} \quad y_i(w \cdot x_i + b) \geq 1 \quad (4)$$

Keterangan:

- w : vektor bobot, yang menentukan arah garis pemisah
- b : bias (intersep), menentukan posisi garis pemisah
- x_i : vektor fitur dari data ke- i
- y_i : label kelas dari data ke- i biasanya berupa nilai diskret seperti $-1, 0$, atau $+1$
- $w \cdot x_i + b$: : fungsi keputusan untuk menentukan posisi data terhadap *hyperplane*

Dengan cara ini, *SVM* berusaha membentuk garis pemisah sebaik mungkin agar data dari tiap kategori sentimen bisa dipisahkan dengan jelas, sehingga hasil klasifikasinya lebih tepat. Model ini digunakan setelah data teks diubah ke dalam bentuk angka menggunakan *TF-IDF* dan jumlah data di setiap kelas diseimbangkan dengan teknik *SMOTE*.

3.6 Evaluasi model

Tujuan dari evaluasi kinerja model dalam penelitian ini adalah untuk mengukur seberapa efektif algoritma *Naive Bayes* dan *SVM* dalam melakukan klasifikasi sentimen. Penilaian dilakukan dengan cara membandingkan hasil kinerja kedua model menggunakan berbagai metrik evaluasi, seperti *akurasi*, *presisi*, *recall*, *F1-score*, dan *Area Under Curve*. Untuk memperoleh hasil yang objektif dan mengukur konsistensi performa model, dilakukan pengujian dengan tiga skenario proporsi data, yaitu menggunakan 70% data sebagai pelatihan dan 30% sebagai pengujian, 80% pelatihan dan 20% pengujian, serta 90% pelatihan dan 10% pengujian. Variasi rasio ini bertujuan untuk mengidentifikasi dampak proporsi data terhadap *akurasi klasifikasi*. Dalam proses evaluasi, *confusion matrix* digunakan sebagai dasar perhitungan metrik utama, sedangkan *AUC* dihitung dengan pendekatan *macro average* agar dapat menangani skenario klasifikasi *multi-kelas* secara seimbang.

4. Hasil dan Pembahasan

4.1 Crawling data

Crawling atau pengumpulan data adalah langkah penting untuk mengumpulkan informasi yang sesuai dengan topik penelitian. Dalam penelitian ini, sebanyak 2.188 data berhasil

dikumpulkan dari 20 tweet yang berkaitan dengan kasus Ronald Tannur di media sosial *Twitter*. Proses pengumpulan dilakukan dengan memanfaatkan token akses dari *Twitter API* yang terintegrasi melalui *Google Colab*, sehingga memungkinkan penarikan data secara otomatis dari setiap tautan tweet yang telah ditentukan. Hasil dari proses *crawling* tersebut kemudian disajikan dalam bentuk tabel, yang menunjukkan keseluruhan data yang telah diperoleh dari *Twitter*.

Tabel 1. Rincian jumlah data pada 20 postingan
Jumlah komentar yang sudah terseleksi

Postingan 1	119
Postingan 2	139
Postingan 3	106
Postingan 4	49
Postingan 5	37
Postingan 6	139
Postingan 7	129
Postingan 8	134
Postingan 9	108
Postingan 10	118
Postingan 11	129
Postingan 12	125
Postingan 13	84
Postingan 14	126
Postingan 15	120
Postingan 16	110
Postingan 17	90
Postingan 18	132
Postingan 19	145
Postingan 20	49

4.2 Preprocessing

Sebelum digunakan untuk implementasi pemodelan, data harus dibersihkan terlebih dahulu untuk menghilangkan noise dengan melakukan beberapa langkah *preprocessing*

4.2.1 Duplicate Data Removal

Penghapusan data duplikat dilakukan untuk memastikan setiap entri dalam dataset bersifat unik, sehingga hasil analisis sentimen lebih akurat dan tidak dipengaruhi oleh informasi berulang. Proses ini menggunakan metode deduplikasi berbasis *Pandas* dengan fungsi *drop_duplicates()*. Hasil eksekusi menunjukkan bahwa jumlah data sebelum penghapusan duplikat adalah 2.187, dan setelah proses tersebut tersisa 2.093 entri. Dengan demikian, sebanyak 94 data duplikat berhasil dihapus, yang meningkatkan kualitas dataset untuk analisis lebih lanjut.

4.2.2 Cleansing

Pembersihan menghapus *mention*, *URL*, hashtag, angka, tanda baca, dan spasi berlebih untuk memastikan teks lebih relevan. Berikut adalah contoh dari hasil yang diperoleh setelah proses *Cleansing*.

Tabel 2. Contoh Hasil *Cleansing* Data

Sebelum <i>Cleansing</i>	Setelah <i>Cleansing</i>
@Heraloebss itu tanganya emg udah item / lebam atau karna gelap aja?	itu tanganya emg udah item lebam atau karna gelap aja
@Heraloebss USUT TUNTAS! jangan sampe ada kalimat CCTV rusak	USUT TUNTAS jangan sampe ada kalimat CCTV rusak
https://t.co/qs6fZYuamF	

4.2.3 Case Folding

Langkah ini bertujuan untuk mengonversi seluruh teks ke dalam huruf kecil guna menjaga konsistensi format penulisan serta memastikan keseragaman dalam proses analisis[13]. Contoh berikut menunjukkan hasil proses konversi huruf.

Tabel 3. Hasil Penerapan *Case Folding*

Sebelum <i>Case Folding</i>	Setelah <i>Case Folding</i>
itu tanganya emg udah item lebam atau karna gelap aja USUT TUNTAS jangan sampe ada kalimat CCTV rusak	itu tanganya emg udah item lebam atau karna gelap aja usut tuntas jangan sampe ada kalimat cctv rusak

4.2.4 Word Normalization

Mengubah kata tidak baku ke dalam bentuk baku untuk menjaga konsistensi dalam analisis teks. Berikut adalah contoh hasil dari penerapan *Word Normalization*.

Tabel 4. Contoh Teks Setelah Proses *Word Normalization*

Sebelum <i>Word Normalization</i>	Setelah <i>Word Normalization</i>
itu tanganya emg udah item lebam atau karna gelap aja usut tuntas jangan sampe ada kalimat cctv rusak	itu tanganya emang sudah hitam lebam atau karena gelap saja usut tuntas jangan sampai ada kalimat cctv rusak

4.2.5 Stopword Removal

Menghilangkan kata-kata yang tidak memberikan kontribusi signifikan untuk meningkatkan relevansi dalam analisis teks. Berikut adalah contoh hasil dari penerapan *Stopword Removal*.

Tabel 5. Hasil Penerapan *Stopword Removal*

Sebelum <i>Stopword Removal</i>	Setelah <i>Stopword Removal</i>
itu tanganya emang sudah hitam lebam atau karena gelap saja usut tuntas jangan sampai ada kalimat cctv rusak	tanganya emang hitam lebam gelap usut tuntas jangan ada kalimat cctv rusak

4.2.6 Tokenizing

Proses pemotongan kalimat menjadi unit-unit terkecil, yaitu kata-kata, untuk mempermudah analisis lebih lanjut dalam pemrosesan teks[14]. Berikut adalah contoh hasil dari penerapan *Tokenizing*.

Tabel 6. Hasil Penerapan *Tokenizing*

Sebelum <i>Tokenizing</i>	Setelah <i>Tokenizing</i>
tanganya emang hitam lebam gelap usut tuntas jangan ada kalimat cctv rusak	['tanganya', 'emang', 'hitam', 'lebam', 'gelap'] ['usut', 'tuntas', 'jangan', 'ada', 'kalimat', 'cctv', 'rusak']

4.2.7 Stemming

Proses pengembalian kata ke bentuk dasarnya dengan menghilangkan imbuhan, sehingga kata seperti "penelitian" menjadi "teliti" atau "pembelajaran" menjadi "ajar". Berikut adalah contoh hasil dari penerapan *Stemming*.

Tabel 7. Hasil Pengembalian Kata ke Bentuk Dasar

Sebelum <i>Stemming</i>	Setelah <i>Stemming</i>
[tanganya] , 'emang', 'hitam', 'lebam', 'gelap'] ['usut', 'tuntas', 'jangan', 'ada', 'kalimat', 'cctv', 'rusak']	tangan emang hitam lebam gelap usut tuntas jangan ada kalimat cctv rusak

4.3 Pelabelan

Proses pelabelan dilakukan dengan menggunakan *Python* untuk menentukan sentimen setiap tweet. Tweet dengan sentimen positif berisi pujian atau emosi positif seperti senang, puas,

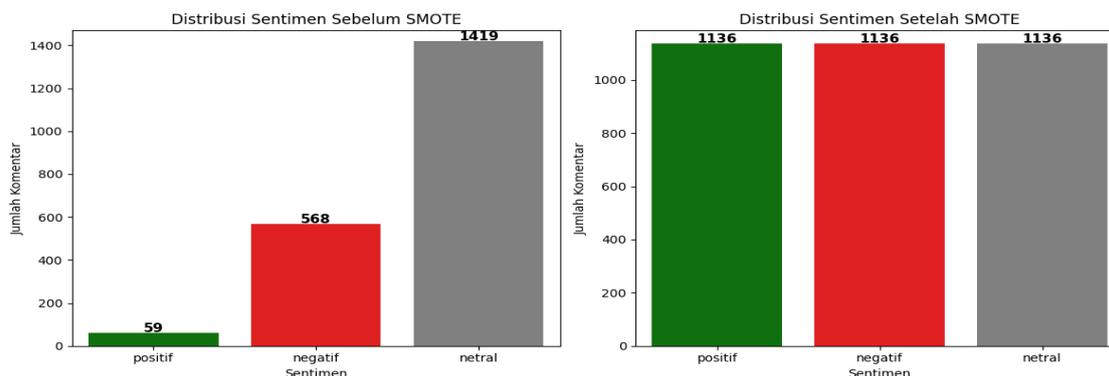
atau bahagia, sedangkan sentimen negatif mencakup keluhan, hinaan, kritik, dan emosi negatif seperti marah, kesal, dan kecewa. Selain itu, tweet dengan sentimen netral tidak mengandung ekspresi emosional yang kuat, baik positif maupun negatif. Kategori-kategori ini ditampilkan dalam tabel berikut.

Tabel 8. Hasil Pelabelan Sentimen

Tweet	Sentimen
alhamdulillah masih ada orang berhati nurani baik	Positif
tangan memang hitam lebam gelap	Netral
Lanjut kawan bakar hakim	Negatif

Berdasarkan hasil pelabelan terhadap 2.046 tweet terkait kasus Ronald Tannur, anak anggota DPR yang diduga terlibat dalam penganiayaan hingga menyebabkan kematian Devina Kharisma, diperoleh distribusi sentimen sebagai berikut: 59 tweet (2,88%) positif, 1.419 tweet (69,34%) netral, dan 568 tweet (27,74%) negatif. Meskipun sentimen netral mendominasi, banyak di antaranya mengandung kritik atau sindiran yang secara makna mendekati sentimen negatif. Beberapa komentar juga menyoroti kekhawatiran terhadap keadilan hukum, menunjukkan betapa beragam dan tidak sederhananya pandangan publik terhadap kasus ini.

Untuk mengatasi ketimpangan jumlah data antar kelas sentimen yang dapat memengaruhi kinerja model klasifikasi, diterapkan metode *Synthetic Minority Over-sampling Techniqu*. Metode ini menyeimbangkan data dengan menambahkan sampel sintesis pada kelas positif dan negatif, sehingga seluruh kelas positif, negatif, dan netral memiliki jumlah yang sama, yaitu 1.136 tweet. Dengan distribusi yang seimbang, model diharapkan mampu mengenali pola sentimen secara lebih akurat dan tidak bias terhadap kelas dominan. Gambar perbandingan distribusi sebelum dan sesudah *SMOTE* ditampilkan untuk memperjelas perubahan proporsi data.



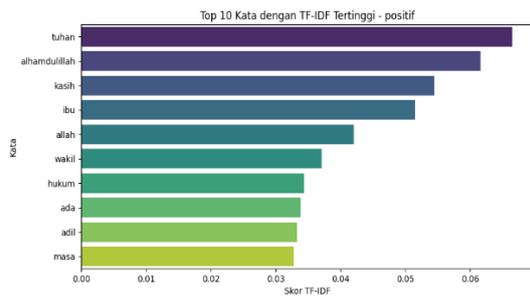
Gambar 2. Grafik Distribusi Sentimen Komentar Sebelum dan Sesudah *SMOTE*

4.4 Transformasi Data dan Visualisasi (TF-IDF)

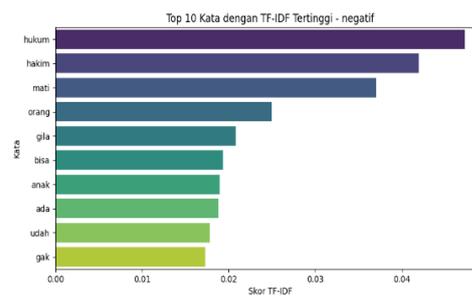
Setelah teks dikelompokkan menjadi sentimen positif, negatif, dan netral, langkah berikutnya adalah mengonversinya ke dalam bentuk angka menggunakan metode *TF-IDF*. Metode ini mengukur seberapa penting suatu kata dalam sebuah dokumen berdasarkan frekuensinya dalam kategori tertentu dan distribusinya di seluruh dataset. Dengan pendekatan ini, kata-kata yang paling berpengaruh dalam menentukan sentimen dapat diidentifikasi lebih tepat.

Untuk memahami lebih dalam distribusi kata dalam setiap kategori sentimen, dilakukan visualisasi menggunakan Diagram Batang. Grafik ini menampilkan 10 kata dengan nilai *TF-IDF* tertinggi pada masing-masing kategori, sehingga memberikan gambaran yang lebih jelas mengenai kata-kata dominan dalam menentukan sentimen suatu teks. Visualisasi ini membantu dalam menginterpretasikan pola bahasa yang muncul pada setiap sentimen, serta memberikan wawasan tambahan bagi analisis lebih lanjut.

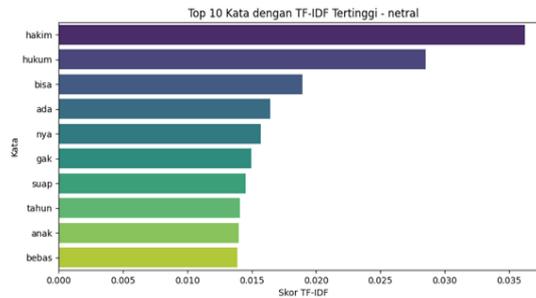
Berikut adalah *Bar Chart* yang menunjukkan kata-kata dengan nilai *TF-IDF* tertinggi dalam kategori positif, negatif, dan netral.



Gambar 3. Grafik Sentimen Positif



Gambar 4. Grafik Sentimen Negatif



Gambar 5. Grafik Sentimen Netral

4.5 Klasifikasi Sentimen

Pada tahap klasifikasi sentimen, algoritma *Multinomial Naive Bayes* digunakan untuk menghitung probabilitas suatu dokumen termasuk ke dalam salah satu kelas sentimen berdasarkan rumus:

$$P(c|d) = \frac{P(c) \prod_{i=1}^n p(w_i | c)}{P(d)} \tag{5}$$

Dalam implementasinya, nilai $P(c)$ diperoleh dari proporsi jumlah data latih pada masing-masing kelas, sedangkan $P(w_i | c)$ dihitung berdasarkan frekuensi kemunculan kata w_i dalam kelas tertentu, ditambah *smoothing* untuk menghindari probabilitas nol. Sebagai contoh, pada dokumen uji “*Lanjut kawan bakar hakim*”, perhitungan probabilitas menghasilkan nilai tertinggi pada kelas negatif sebesar $1,82 \times 10^{-12}$, dibandingkan dengan kelas netral $3,42 \times 10^{-19}$, dan positif $1,68 \times 10^{-17}$ sehingga dokumen diklasifikasikan sebagai sentimen negatif. Evaluasi performa model *Naive Bayes* menunjukkan bahwa hasil terbaik diperoleh pada *rasio* data 90:10 dengan *akurasi* 71,29%, *presisi* 0,79, *recall* 0,71, *F1-score* 0,73, dan *AUC* sebesar 0,77. Dengan penurunan performa saat *rasio* diturunkan ke 80:20 dan sedikit peningkatan kembali pada 70:30, menunjukkan bahwa *Naive Bayes* sensitif terhadap ukuran data pelatihan dan bekerja optimal dengan jumlah data latih yang besar.

Sebaliknya, algoritma *Support Vector Machine (SVM)* bekerja dengan mencari *hyperplane* optimal yang memisahkan data ke dalam kelas-kelas berbeda secara maksimal, berdasarkan rumus:

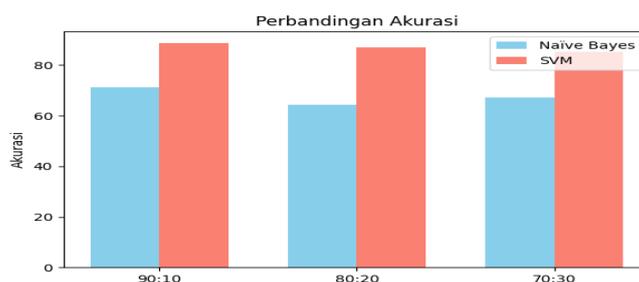
$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{dengan syarat} \quad y_i(w \cdot x_i + b) \geq 1 \tag{6}$$

Dalam penerapannya, data teks terlebih dahulu ditransformasi menjadi vektor numerik menggunakan *TF-IDF*, kemudian dilakukan penyeimbangan data menggunakan teknik *SMOTE*. Setelah model dilatih, proses klasifikasi pada data uji dilakukan dengan menghitung skor keputusan menggunakan fungsi $w \cdot x_i + b$, di mana nilai tertinggi menjadi penentu kelas akhir. Sebagai contoh, pada dokumen uji “*Lanjut kawan bakar hakim*”, model menghasilkan *confidence score* sebesar 0,87 untuk kelas negatif, 0,10 untuk netral, dan 0,03 untuk positif, sehingga dokumen diklasifikasikan sebagai sentimen negatif. Evaluasi performa model menunjukkan bahwa hasil terbaik diperoleh pada *rasio* data 90:10 dengan *akurasi* 88,78%, *presisi* 0,89, *recall* 0,76, *F1-score* 0,82, dan *AUC* 0,84. Tidak seperti *Naive Bayes*, performa *SVM* tetap tinggi dan stabil pada *rasio* 80:20 maupun 70:30, yang menunjukkan bahwa *SVM* tetap mampu menghasilkan hasil klasifikasi yang baik dan konsisten meskipun jumlah data pelatihan dikurangi

4.6 Evaluasi Model

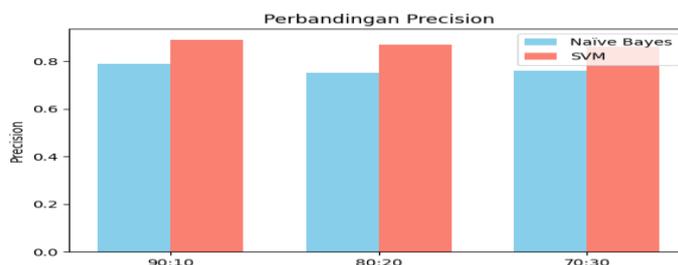
Evaluasi model dalam penelitian ini dimaksudkan untuk mengidentifikasi metode klasifikasi yang paling optimal dalam mengolah sentimen terhadap kasus Ronald Tannur di platform *Twitter*. Dua metode pembelajaran mesin yang dibandingkan dalam studi ini adalah *Naive Bayes* dan *SVM*. Penilaian performa kedua model dilakukan dengan mengacu pada lima indikator utama, yakni *akurasi*, *presisi*, *recall*, *F1-score*, dan *AUC*. Evaluasi ini juga bertujuan untuk mengukur kemampuan masing-masing algoritma dalam mendeteksi serta mengenali pola data secara akurat, sehingga dapat menghindari risiko *overfitting* maupun *underfitting*.

Untuk memperoleh pemahaman yang menyeluruh tentang pengaruh perbandingan data latih dan uji terhadap hasil klasifikasi, dilakukan pengujian pada tiga skenario proporsi data: 70:30, 80:20, dan 90:10. Tujuannya adalah untuk mengetahui sejauh mana variasi rasio data tersebut mempengaruhi efektivitas masing-masing algoritma dalam melakukan klasifikasi sentimen.



Gambar 6. Grafik Akurasi

Berdasarkan gambar di atas, diketahui bahwa SVM memiliki akurasi tertinggi pada setiap rasio data yang digunakan



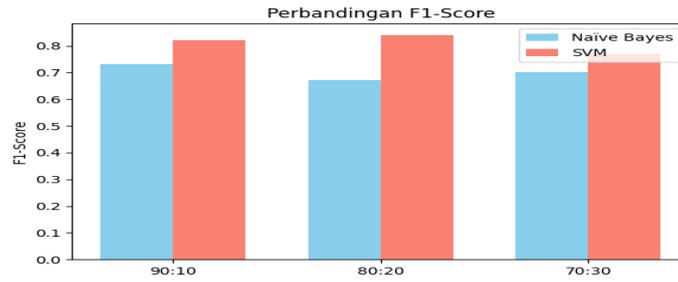
Gambar 7. Grafik Precision

Gambar 7 menunjukkan bahwa SVM tetap memiliki nilai presisi tertinggi di setiap rasio, meskipun selisihnya tidak terlalu besar.



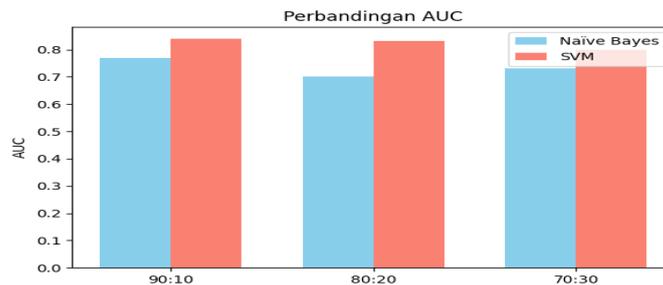
Gambar 8. Grafik Recall

Pada Gambar 8, terlihat bahwa nilai recall SVM selalu lebih tinggi dibandingkan Naive Bayes di setiap rasio.



Gambar 9. Grafik F1-Score

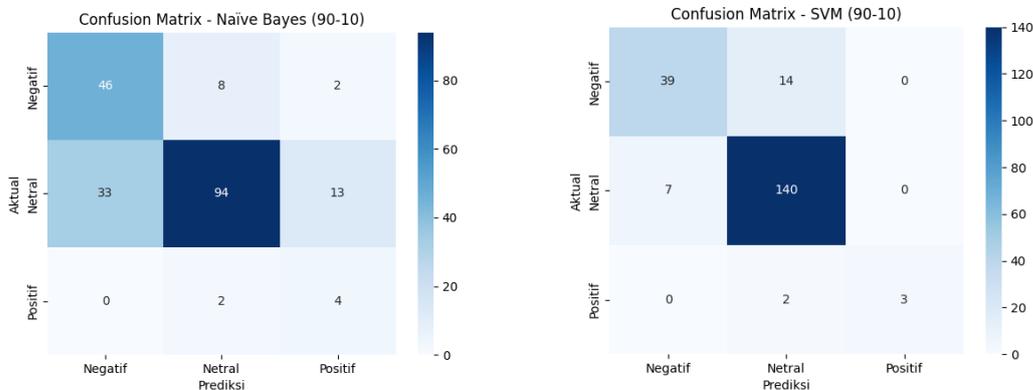
Berdasarkan gambar di atas, dapat diketahui bahwa nilai F1-score tertinggi pada setiap rasio diperoleh oleh SVM.



Gambar 10. Grafik AUC

Gambar 10 diatas menunjukkan bahwa nilai auc pada algoritma svm lebih baik dibandingkan naive bayes pada setiap rasio walaupun perbedaannya tidak terlalu jauh.

Algoritma SVM menunjukkan performa yang konsisten dan unggul di seluruh *rasio* data yang diuji. Pada *rasio* data 90:10, SVM mencatatkan akurasi tertinggi sebesar 88,78%, dengan *presisi* rata-rata 0,89, *recall* 0,76, *F1-score* 0,82, dan *AUC* sebesar 0,84. Kinerja ini melampaui hasil *Naive Bayes*, yang pada *rasio* data yang sama hanya menghasilkan akurasi 71,29%, dengan *presisi* 0,79, *recall* 0,71, *F1-score* 0,73, dan *AUC* 0,77. Selain itu, pada rasio 80:20, SVM tetap unggul dengan akurasi 87,07%, *presisi* 0,87, *recall* 0,79, *F1-score* 0,84, dan *AUC* 0,83, dibandingkan dengan *Naive Bayes* yang mencatatkan akurasi 64,36%, *presisi* 0,75, *recall* 0,64, *F1-score* 0,67, dan *AUC* 0,70. Sementara itu, pada rasio 70:30, performa SVM tetap stabil dengan akurasi 85,34%, *presisi* 0,86, *recall* 0,71, *F1-score* 0,77, dan *AUC* 0,80, jauh mengungguli *Naive Bayes* yang hanya memperoleh akurasi 67,33%, *presisi* 0,76, *recall* 0,67, *F1-score* 0,70, dan *AUC* 0,73. Selain akurasi, SVM juga lebih unggul dalam *precision*, *recall*, *F1-score*, dan *AUC*, menunjukkan kemampuannya dalam menangkap pola sentimen secara lebih akurat dibandingkan *Naive Bayes*.



Gambar 11. Confusion matrix Naive Bayes dan SVM dengan rasio 90:10.

Untuk memahami lebih lanjut bagaimana masing-masing model menangani klasifikasi sentimen, ditampilkan *confusion matrix* dari model terbaik pada setiap algoritma, yaitu *Naive Bayes* dan *SVM* dengan rasio 90:10.

Hasil analisis *confusion matrix* memperlihatkan bahwa algoritma *SVM* mampu mengklasifikasikan sentimen dengan lebih akurat dibandingkan *Naive Bayes*. *SVM* berhasil mengklasifikasikan 140 data netral dengan tepat, hanya 7 data yang salah prediksi sebagai negatif. Untuk kelas negatif, 39 dari 53 data diprediksi benar, dan pada kelas positif, 3 dari 5 data terklasifikasi dengan akurat. Tingkat kesalahan pada *SVM* tergolong rendah, terutama pada kelas netral yang merupakan kategori terbanyak dalam dataset. Sebaliknya, *Naive Bayes* cukup baik dalam memprediksi kelas negatif (46 dari 56 data), tetapi banyak melakukan kesalahan pada kelas netral, dengan 33 data salah diprediksi sebagai negatif dan 13 sebagai positif.

Hasil dari penelitian ini mengindikasikan bahwa metode *SVM* secara konsisten memberikan hasil klasifikasi yang lebih baik dibandingkan dengan *Naive Bayes* dalam melakukan analisis sentimen di media sosial, khususnya pada kasus-kasus yang melibatkan opini publik seperti kasus Ronald Tannur. Hasil ini sejalan dengan penelitian sebelumnya oleh [15] dan [9], yang juga menemukan bahwa *SVM* unggul dalam menangani data teks dengan distribusi kelas yang tidak seimbang. Namun, berbeda dengan studi-studi tersebut, Penelitian ini memberikan kontribusi melalui penggunaan metrik *AUC* sebagai alat bantu penilaian tambahan untuk melihat seberapa baik model membedakan tiap kelas sentimen. Selain itu, dengan menguji model menggunakan tiga variasi pembagian data (90:10, 80:20, dan 70:30), penelitian ini memberikan gambaran yang lebih menyeluruh terhadap kestabilan model dibandingkan penelitian seperti [13] dan [16] yang hanya menggunakan satu *rasio* pembagian data. Dengan demikian, hasil temuan ini memperkuat bukti bahwa *SVM* merupakan pendekatan yang lebih andal dalam klasifikasi sentimen, sekaligus memperkaya kajian terkait pemanfaatan pembelajaran mesin dalam analisis media sosial.

Berdasarkan keseluruhan hasil evaluasi, dapat disimpulkan bahwa model *SVM* dengan *rasio* pembagian data 90:10 menjadi pilihan terbaik dalam analisis sentimen untuk kasus ini. Selain mencatatkan akurasi tertinggi, model ini juga menunjukkan kestabilan dan kinerja yang seimbang pada seluruh metrik evaluasi. Oleh karena itu, *SVM* sangat disarankan sebagai algoritma yang tepat dan terpercaya untuk digunakan dalam klasifikasi sentimen berbasis teks, terutama di media sosial seperti *Twitter*.

5. Simpulan

Berdasarkan evaluasi kinerja dua metode untuk klasifikasi sentimen, yakni *Naive Bayes* dan *Support Vector Machine*, dalam konteks analisis sentimen terhadap kasus Ronald Tannur di *Twitter*, dapat disimpulkan bahwa *SVM* memberikan hasil yang lebih memuaskan. Pada seluruh metrik evaluasi, seperti akurasi, presisi, recall, F1-score, dan *AUC*, terutama pada rasio pembagian data 90:10 dengan akurasi mencapai 88,78%. Sementara *Naive Bayes*, meskipun lebih efisien dalam pemrosesan, menunjukkan kinerja yang lebih rendah, dengan akurasi hanya 71,29% pada rasio yang sama. Keunggulan *SVM* dalam klasifikasi sentimen ini dapat dikaitkan dengan kemampuannya untuk menangani hubungan kompleks antar fitur dalam data teks melalui transformasi *TF-IDF* dan pemisahan kelas sentimen yang lebih efektif menggunakan *hyperplane* di ruang berdimensi tinggi. Meskipun demikian, *Naive Bayes* tetap memiliki peran penting dalam situasi yang membutuhkan efisiensi komputasi lebih cepat. Dengan temuan ini, *SVM* direkomendasikan sebagai algoritma yang lebih andal untuk analisis sentimen berbasis teks, terutama dalam konteks media sosial. Ke depan, pengembangan lebih lanjut dapat dilakukan dengan mengkombinasikan *SVM* dengan teknik pra-proses lanjutan atau metode *ensemble* untuk meningkatkan akurasi dan efisiensi, serta mengadaptasi model ini pada berbagai jenis analisis sentimen dalam konteks sosial yang lebih luas.

Daftar Referensi

- [1] S. Rabbani, D. Safitri, N. Rahmadhani, A. A. F. Sani, and M. K. Anam, "Perbandingan Evaluasi Kernel SVM untuk Klasifikasi Sentimen dalam Analisis Kenaikan Harga BBM," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 153–160, 2023, doi: 10.57152/malcom.v3i2.897.
- [2] D. Oktavia, Y. R. Ramadahan, and Minarto, "Analisis Sentimen Terhadap Penerapan Sistem E-Tilang Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine

- (SVM),” *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 407–417, 2023, doi: 10.30865/klik.v4i1.1040.
- [3] M. A. A., M. Alamsyah, and M. F. Arif, “Analisis Sentimen Twitter Tentang Pinjaman Online di Indonesia Menggunakan Metode Random Forest,” *Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 13, no. 2, pp. 1410–1421, 2024, doi: 10.35889/jutisi.v13i2.2215.
- [4] A. Asyer and M. A. I. Pakereng, “Analisis Sentimen Tweet Pengguna Twitter Terkait Diabetes Menggunakan Metode Naive Bayes,” *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 12, no. 2, pp. 627–636, 2023, doi: 10.35889/jutisi.v12i2.1234.
- [5] A. F. Setyaningsih, D. Septiyani, and S. R. Widiyarsi, “Implementasi Algoritma Naive Bayes untuk Analisis Sentimen Masyarakat pada Twitter mengenai Kepopuleran Produk Skincare di Indonesia,” *Jurnal Teknologi Informatika dan Komputer MH. Thamrin*, vol. 9, no. 1, pp. 224–235, 2023, doi: 10.37012/jtik.v9i1.1409.
- [6] R. Rachman and R. N. Handayani, “Klasifikasi Algoritma Naive Bayes Dalam Memprediksi Tingkat Kelancaran Pembayaran Sewa Teras UMKM,” *Jurnal Informatika*, vol. 8, no. 2, pp. 111–122, 2021, doi: 10.31294/ji.v8i2.10494.
- [7] Putri Ramayanti and Tata Sutabri, “Perbandingan Algoritma Naive Bayes Dan Svm Untuk Analisis Penyalahgunaan Kejahatan Carding,” *Jurnal Informatika Teknologi dan Sains*, vol. 5, no. 1, pp. 18–24, 2023, doi: 10.51401/jinteks.v5i1.2231.
- [8] R. I. Agustin, “Komparasi Algoritma Naive Bayes Dan Svm Untuk Analisis Sentimen Twitter Korupsi Bansos Beras Masa Pandemi,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, pp. 912–918, 2024, doi: 10.23960/jitet.v12i2.4020.
- [9] S. D. Fitri, “Jurnal Sistem dan Teknologi Informasi Indonesia Perbandingan Metode Naive Bayes dan Support Vector Machine Pada Kasus Pembunuhan Vina Cirebon Berdasarkan Data X Comparison of Naive Bayes and Support Vector Machine Methods in the Vina Cirebon Murder Case B,” *Jurnal Sistem dan Teknologi Informasi Indonesia*, vol. 10, no. 1, pp. 39–49, 2025, doi: 10.32528/justindo.v10i1.2550.
- [10] U. R. Debora, I. P. A. E. Pratama, and G. M. A. Sasmita, “Sentiment Analysis of Domestic Violence Issues on Twitter Using Multinomial Naive Bayes and Support Vector Machine,” *JITTER: Jurnal Ilmiah Teknologi dan Komputer*, vol. 4, no. 3, p. 1992, 2024, doi: 10.24843/jtrti.2023.v04.i03.p06.
- [11] Septi Putri, Yohanes Agung Apriyanto, and Andri Wijaya, “Analisis Sentimen Ulasan Aplikasi DeepL Pada Google Play Dengan Metode Support Vector Machine (Svm),” *Jurnal Sistem Informasi (JUSIN)*, vol. 4, no. 2, pp. 59–66, 2023, doi: 10.32546/jusin.v4i2.2368.
- [12] C. Muhamad, S. Ramdani, A. N. Rachman, and R. Setiawan, “Comparison of the Multinomial Naive Bayes Algorithm and Decision Tree with the Application of AdaBoost in Sentiment Analysis Reviews PeduliLindungi Application,” *International Journal of Information System & Technology Akreditasi*, vol. 6, no. 158, pp. 419–430, 2022, doi: 10.30645/ijistech.v6i4.257.
- [13] J. C. Moris Dachi Sunneng Sandino Berutu2, “Klasifikasi Sentimen Opini Pada Aplikasi Gitar Tuner,” *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 13, no. 2, pp. 1444–1459, 2024, doi: 10.35889/jutisi.v13i2.2041.
- [14] A. Muzaki and A. Witanti, “Sentiment Analysis of the Community in the Twitter To the 2020 Election in Pandemic Covid-19 By Method Naive Bayes Classifier,” *Jurnal Teknik Informatika (Jutif)*, vol. 2, no. 2, pp. 101–107, 2021, doi: 10.20884/1.jutif.2021.2.2.51.
- [15] B. A. Maulana, M. J. Fahmi, A. M. Imran, and N. Hidayati, “Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM),” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 2, pp. 375–384, 2024, doi: 10.57152/malcom.v4i2.1206.
- [16] F. Matheos Sarimole and K. Kudrat, “Analisis Sentimen Terhadap Aplikasi Satu Sehat Pada Twitter Menggunakan Algoritma Naive Bayes Dan Support Vector Machine,” *Jurnal Sains dan Teknologi*, vol. 5, no. 3, pp. 783–790, 2024, doi: 10.55338/saintek.v5i3.2702.