ICo SCIENCE: Journal Of Intelligent Computing And Advanced Data Science *Origina Paper*
VOL 1 Issue 2 (2026)

Deteksi dan Penyaringan Otomatis Komentar Toxic pada TikTok Menggunakan TF-IDF dan Support Vector Machine

Dewi Purnama Sari[1], Dicky Ramadhan Hutasuhut[2], Nadya Balqis Nasution[3], Risya Amalia Putri Nasution[4]

Universitas Satya Terra Bhinneka, Medan, Indonesia
Department of Informatics, Universitas Satya Terra Bhinneka, Medan, Indonesia

ABSTRACT

The rapid growth of social media, particularly TikTok, has increased user interaction through its comment feature. However, the large number of comments has also led to the emergence of toxic comments containing hate speech, insults, and cyberbullying. Manual moderation is considered ineffective due to the massive volume of comments and the complexity of informal language commonly used on social media. This study aims to implement a text mining approach using the Support Vector Machine (SVM) algorithm to detect toxic comments and develop an automatic filtering mechanism for TikTok comments. The dataset was collected by scraping comments directly from several TikTok videos. The preprocessing stage included case folding, cleaning, tokenization, stopword removal, and stemming. Feature extraction was performed using TF-IDF, followed by text classification using the SVM algorithm. Model performance was evaluated using accuracy, precision, recall, and F1-score. The experimental results show that the SVM model achieved an accuracy of 95.28%, with a precision of 1.00, recall of 0.67, and F1-score of 0.80 for the toxic class. Furthermore, the developed auto-filtering system successfully filters toxic comments automatically, making the content moderation process faster and more efficient. The proposed approach demonstrates that combining TF-IDF feature extraction with the SVM algorithm can effectively support automated content moderation and help reduce the spread of harmful comments on social media platforms.

ARTICLE INFO

Keywords:

Text Mining, Support Vector Machine, Toxic Comment, Auto Filtering, TikTok.

Histori:
Received: 03 July 2026
Accepted: 08 July 2026
Published: 29 July 2026

*Corresponding Author at Department of Informatics, Universitas Satya Terra Bhinneka, Medan, Indonesia.
E-mail address: dewisunarsih1704@gmail.com (author #1),dhickyhts@gmail.com (author #2), nadyabalqis501@gmail.com (author# 3) risyaamaliaa02@gmail.com (author # 4)

1. INTRODUCTION

Perkembangan teknologi informasi dan internet telah mengubah cara masyarakat dalam berkomunikasi, memperoleh informasi, serta berinteraksi di berbagai platform digital. Salah satu dampak yang paling signifikan adalah meningkatnya penggunaan media sosial sebagai sarana komunikasi, hiburan, pendidikan, hingga penyebaran informasi. Berbagai platform media sosial terus berkembang dengan menyediakan fitur yang memungkinkan pengguna untuk membuat, membagikan, dan memberikan tanggapan terhadap suatu konten secara cepat. Salah satu platform yang mengalami pertumbuhan pengguna yang sangat pesat adalah TikTok. Selain digunakan sebagai media hiburan, TikTok juga dimanfaatkan sebagai sarana promosi, edukasi, pemasaran digital, hingga penyebaran informasi kepada masyarakat. Fitur komentar yang tersedia pada setiap video memungkinkan pengguna berinteraksi secara langsung dengan kreator maupun pengguna lainnya sehingga menciptakan komunikasi dua arah yang aktif.

Meningkatnya jumlah pengguna TikTok menyebabkan volume komentar yang dihasilkan setiap hari juga terus bertambah. Di samping komentar yang bersifat positif, banyak ditemukan komentar yang mengandung unsur kebencian (hate speech), penghinaan, ujaran kasar, cyberbullying, maupun bentuk komunikasi negatif lainnya yang dikenal sebagai toxic comment. Komentar toxic merupakan bentuk komunikasi yang mengandung unsur negatif dan berpotensi menimbulkan konflik di media sosial sehingga dapat memberikan dampak psikologis bagi korban, menurunkan kualitas interaksi digital, serta menciptakan lingkungan media sosial yang tidak sehat (Sidiq et al., 2020). Karakteristik komentar pada TikTok yang umumnya menggunakan bahasa tidak baku, singkatan, bahasa gaul, emoji, serta berbagai variasi penulisan membuat proses identifikasi komentar toxic menjadi lebih kompleks dibandingkan dengan dokumen teks formal (Berindikasi et al., 2025). Selain itu, penyebaran komentar yang mengandung ujaran kebencian dapat berlangsung sangat cepat apabila tidak segera dimoderasi sehingga berpotensi memengaruhi pengguna lain (Sentimen et al., n.d.).

Saat ini proses moderasi komentar pada sebagian besar media sosial masih dilakukan melalui pelaporan pengguna (user reporting) atau pemeriksaan secara manual oleh moderator. Pendekatan tersebut dinilai kurang efektif karena membutuhkan waktu yang relatif lama, terutama ketika harus menangani ribuan bahkan jutaan komentar yang terus bertambah setiap hari. Moderasi manual juga sangat bergantung pada kemampuan manusia dalam melakukan penilaian sehingga berpotensi menimbulkan inkonsistensi hasil klasifikasi. Oleh karena itu, diperlukan suatu sistem yang mampu mendeteksi sekaligus menyaring komentar toxic secara otomatis agar proses moderasi dapat dilakukan secara lebih cepat, efisien, dan konsisten.

Salah satu pendekatan yang banyak digunakan untuk menyelesaikan permasalahan tersebut adalah text mining. Text mining merupakan teknik pengolahan data teks tidak terstruktur menjadi informasi yang dapat dianalisis melalui serangkaian tahapan, seperti case folding, cleaning, tokenization, stopword removal, dan stemming (Hermawan & Jowensen, 2023). Setelah melalui proses preprocessing, data teks dapat direpresentasikan ke dalam bentuk numerik menggunakan metode Term Frequency–Inverse Document Frequency (*TF-IDF*) sehingga dapat digunakan sebagai masukan bagi algoritma klasifikasi.Pemanfaatan text

mining pada media sosial telah banyak diterapkan untuk analisis sentimen, deteksi spam, klasifikasi opini pengguna, hingga deteksi komentar toxic.Dalam penelitian ini digunakan algoritma Support Vector Machine (SVM) sebagai metode klasifikasi. SVM dipilih karena memiliki kemampuan yang baik dalam menangani data berdimensi tinggi, khususnya data teks yang dihasilkan dari media sosial. Selain itu, SVM mampu membentuk hyperplane optimal yang memisahkan dua kelas secara efektif sehingga menghasilkan tingkat akurasi yang tinggi pada berbagai penelitian klasifikasi teks (Handayanto et al., 2019)(Arifin et al., 2021)(Romadoni et al., 2020).

Beberapa penelitian juga menunjukkan bahwa kombinasi metode TF-IDF dengan algoritma SVM mampu meningkatkan performa klasifikasi dokumen karena dapat merepresentasikan bobot kata secara lebih baik sebelum dilakukan proses pembelajaran model (Sakun et al., 2026)(Krisnandi et al., 2023).Beberapa penelitian terdahulu telah mengembangkan sistem deteksi komentar toxic menggunakan berbagai algoritma klasifikasi. Sidiq dkk. menerapkan algoritma Naïve Bayes pada komentar Facebook dan memperoleh hasil klasifikasi yang cukup baik (Sidiq et al., 2020). Romadina, Juwita, dan Pandunata menggunakan Naïve Bayes Classifier untuk menganalisis komentar pada YouTube, sedangkan Jasno dkk. menerapkan algoritma Decision Tree dalam mengidentifikasi komentar yang berpotensi mengandung unsur toxic pada TikTok (Berindikasi et al., 2025)(Sentimen et al., n.d.)(Jasno et al., 2025).

Hasil penelitian tersebut menunjukkan bahwa metode klasifikasi berbasis machine learning mampu membantu proses identifikasi komentar toxic secara otomatis. Meskipun demikian, sebagian besar penelitian sebelumnya masih berfokus pada proses klasifikasi komentar tanpa mengintegrasikan mekanisme penyaringan komentar secara otomatis pada sistem yang dikembangkan. Selain itu, penelitian yang mengombinasikan teknik text mining menggunakan algoritma Support Vector Machine (SVM) dengan fitur auto filtering berbasis data hasil scraping komentar TikTok masih relatif terbatas. Kondisi tersebut menunjukkan adanya research gap yang perlu dikaji lebih lanjut sehingga diperlukan penelitian yang tidak hanya menghasilkan model klasifikasi dengan performa yang baik, tetapi juga mampu mengimplementasikan hasil klasifikasi tersebut ke dalam mekanisme penyaringan komentar secara otomatis.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan menerapkan metode text mining menggunakan algoritma *Support Vector Machine* (*SVM*) untuk mendeteksi komentar toxic pada platform TikTok. Penelitian ini juga mengevaluasi performa model menggunakan metrik accuracy, precision, recall, dan F1-score, serta mengimplementasikan fitur auto filtering yang mampu menyaring komentar toxic secara otomatis. Hasil penelitian diharapkan dapat membantu proses moderasi komentar menjadi lebih cepat, efektif, dan efisien, serta memberikan kontribusi terhadap pengembangan penelitian di bidang text mining, machine learning, dan moderasi konten media sosial.

Kebaruan penelitian ini terletak pada penerapan metode TF-IDF dan Support Vector Machine (SVM) yang diintegrasikan dengan mekanisme *auto filtering* pada komentar TikTok berbahasa Indonesia. Berbeda dengan penelitian sebelumnya yang umumnya hanya berfokus pada proses klasifikasi komentar,

penelitian ini memanfaatkan hasil klasifikasi sebagai dasar penyaringan komentar secara otomatis sehingga dapat mendukung proses moderasi konten pada media sosial.

2. LITERATURE REVIEW

2.1. Text Mining

Text mining dalah suatu bidang khusus dari data mining. Text mining dapat diartikan sebagai suatu proses untuk menggali informasi dimana seorang user (pengguna) berinteraksi dengan kumpulan dokumen menggunakan tool analisis yang merupakan komponen-komponen dalam data mining. Dalam text mining berbeda dengan dengan data mining dimana data mining yang digunakan adalah structured (tersusun) data sementara dalam text mining umumnya data yang ditemui adalah semi-structured (sebagian terstruktur) atau unstructured (tidak terstruktur).

Text mining dalam prakeknya mencari pola-pola tertentu, mengasosiasikan satu bagian teks dengan lain berdasar aturan-aturan tertentu, kata-kata yang dapat diwakilkan sehingga dapat dilakukan analisa keterhubungan antar satu dengan lain, dalam kumpulan dokumen yang sangat banyak. Dokumen yang ada bisa bersifat statis, yaitu dokumen yang tidak akan di perbarui lagi ataupun dinamis yaitu dokumen yang akan selalu diperbarui dalam rentang waktu tertentu(Area, 2024).

2.2. Toxic Comment

Toxic comment merupakan komentar yang mengandung unsur negatif seperti ujaran kebencian (*hate speech*), penghinaan, kata-kata kasar, ancaman, diskriminasi, maupun tindakan cyberbullying. Komentar jenis ini dapat menimbulkan dampak negatif terhadap individu maupun lingkungan media sosial karena dapat memicu konflik dan mengurangi kenyamanan pengguna dalam berinteraksi.

Menurut (Sidiq et al., 2020), komentar *toxic* merupakan bentuk komunikasi yang mengandung ekspresi negatif yang dapat memengaruhi kualitas interaksi antar pengguna media sosial. Keberadaan komentar toxic menjadi salah satu permasalahan yang sering ditemukan pada berbagai platform digital karena tingginya kebebasan pengguna dalam menyampaikan pendapat.

Selain itu, (Romadina et al., 2024) menjelaskan bahwa komentar *toxic* dapat mengandung kata-kata yang bersifat menyerang, merendahkan, atau menyinggung pihak tertentu sehingga berpotensi menimbulkan dampak sosial maupun psikologis.Oleh karena itu, diperlukan sistem yang mampu mendeteksi komentar *toxic* secara otomatis agar penyebaran komentar negatif dapat diminimalkan. Dalam penelitian ini, *toxic comment* diklasifikasikan menjadi dua kategori, yaitu *toxic* dan non-*toxic*. Klasifikasi tersebut dilakukan berdasarkan pola kata yang terdapat pada komentar pengguna menggunakan metode text mining dan algoritma *Support Vector Machine* (SVM).

2.3. Tiktok sebagai media social

TikTok merupakan platform media sosial berbasis video pendek yang memungkinkan pengguna membuat, mengunggah, dan membagikan konten serta berinteraksi melalui fitur komentar. kolom komentar pada

TikTok menjadi salah satu sarana utama bagi pengguna untuk memberikan tanggapan terhadap suatu konten, baik berupa komentar positif maupun negatif.

Komentar pada TikTok memiliki karakteristik bahasa yang beragam, seperti penggunaan bahasa tidak baku, singkatan, bahasa gaul (*slang*), dan variasi penulisan kata sehingga proses identifikasi komentar *toxic* menjadi lebih kompleks apabila dilakukan secara manual (Jasno et al., 2025). Oleh karena itu, diperlukan sistem otomatis yang mampu mendeteksi dan menyaring komentar *toxic* guna mendukung proses moderasi konten secara lebih cepat dan efektif.

2.4. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan salah satu algoritma machine learning yang digunakan untuk proses klasifikasi data. SVM bekerja dengan mencari hyperplane optimal yang mampu memisahkan data ke dalam kelas yang berbeda dengan margin maksimum. Menurut (Handayanto et al., 2019), SVM memiliki kemampuan yang baik dalam mengolah data berdimensi tinggi dan menghasilkan model klasifikasi yang akurat. Oleh karena itu, algoritma ini banyak digunakan dalam berbagai penelitian *text mining* dan klasifikasi teks.(Arifin et al., 2021) menyatakan bahwa SVM mampu mengenali pola penting pada data teks sehingga efektif digunakan dalam proses klasifikasi dokumen. Selain itu, penelitian (Krisnandi et al., 2023) menunjukkan bahwa kombinasi metode TF-IDF dan SVM mampu menghasilkan performa yang baik dalam mendeteksi komentar yang mengandung unsur toksisitas dan agresi.

Dalam penelitian ini, algoritma SVM digunakan untuk mengklasifikasikan komentar TikTok ke dalam kategori *toxic* dan non-*toxic* berdasarkan hasil ekstraksi fitur dari proses *text mining*. Pemilihan algoritma SVM dilakukan karena memiliki kemampuan klasifikasi yang baik, stabil, serta cocok digunakan untuk pengolahan data teks dalam jumlah besar.

2.5. TF IDF

TF-IDF (*Term Frequency–Inverse Document Frequency*) adalah metode statistik yang digunakan dalam pemrosesan bahasa alami dan pengambilan informasi untuk mengevaluasi seberapa penting suatu kata bagi sebuah dokumen dalam kaitannya dengan kumpulan dokumen yang lebih besar. TF-IDF menggabungkan dua komponen(Rahmadhani et al., 2020):

1. Frekuensi Istilah (TF): Mengukur seberapa sering suatu kata muncul dalam sebuah dokumen. Frekuensi yang lebih tinggi menunjukkan kepentingan yang lebih besar. Jika suatu istilah sering muncul dalam sebuah dokumen, kemungkinan besar istilah tersebut relevan dengan isi dokumen.
2. Frekuensi Dokumen Terbalik (*Inverse Document Frequency*/IDF): Mengurangi bobot kata-kata umum di berbagai dokumen sekaligus meningkatkan bobot kata-kata langka. Jika suatu istilah muncul di lebih sedikit dokumen, kemungkinan besar istilah tersebut bermakna dan spesifik. Keseimbangan ini memungkinkan TF-IDF untuk menyoroti istilah-istilah yang sering muncul dalam dokumen tertentu dan juga khas di seluruh dokumen teks, menjadikannya alat yang berguna untuk tugas-tugas seperti pemeringkatan pencarian, klasifikasi teks, dan ekstraksi kata kunci.

2.6. Auto Filtering

Auto filtering merupakan mekanisme penyaringan konten secara otomatis berdasarkan aturan (*rule-based*) maupun hasil prediksi model *machine learning*. Pada platform media sosial, *auto filtering* digunakan untuk membantu proses moderasi dengan mengidentifikasi dan membatasi konten yang melanggar pedoman komunitas, seperti ujaran kebencian (*hate speech*), spam, maupun komentar *toxic*. Penerapan *auto filtering* memungkinkan proses moderasi dilakukan secara lebih cepat, konsisten, dan efisien dibandingkan moderasi manual, terutama pada platform yang memiliki volume komentar sangat besar.

Filtering adalah langkah yang diambil untuk mengekstrak kata-kata kunci yang terdapat dalam judul dokumen data buku. Pada tahap ini, dilakukan eliminasi kata-kata yang tidak memiliki nilai deskriptif, termasuk kata penghubung, kata depan, dan kata ganti seperti "di", "yang", "dan", "ke", dan "dari". Tujuan dari proses filtering adalah untuk mengurangi jumlah kata dalam judul dokumen, memungkinkan proses pengolahan lebih akurat dengan memfokuskan pada kata-kata yang memiliki signifikansi di dalam judul dokumen(Rahmadhani et al., 2020).

Pada penelitian ini, fitur *auto filtering* diimplementasikan dengan memanfaatkan hasil klasifikasi algoritma *Support Vector Machine* (SVM). Setelah komentar melalui tahapan *preprocessing* dan pembobotan TF-IDF, model SVM melakukan prediksi terhadap setiap komentar. Komentar yang diklasifikasikan sebagai *toxic* secara otomatis disaring sehingga tidak ditampilkan, sedangkan komentar yang termasuk kategori *non-toxic* tetap dapat dipublikasikan. Dengan demikian, mekanisme *auto filtering* diharapkan dapat mendukung proses moderasi komentar pada TikTok secara lebih cepat, efektif, dan konsisten.

2.7. Penelitian Terdahulu

1. Penelitian berjudul "Sentimen Analisis Komentar Toxic pada Grup Facebook Game Online Menggunakan Klasifikasi Naive Bayes". Penelitian ini bertujuan mengidentifikasi komentar toxic pada grup Facebook menggunakan algoritma Naive Bayes. Hasil penelitian menunjukkan bahwa algoritma tersebut mampu mengklasifikasikan komentar toxic dengan tingkat akurasi yang baik. Perbedaan dengan penelitian yang dilakukan adalah penggunaan algoritma Support Vector Machine (SVM) serta objek penelitian berupa komentar pada platform TikTok(Sentimen et al., n.d.).

2. Penelitian pada jurnal JUITA menerapkan algoritma Support Vector Machine (SVM) untuk klasifikasi data teks. Hasil penelitian menunjukkan bahwa SVM memiliki performa yang baik dalam mengelompokkan data teks dan mampu menghasilkan tingkat akurasi yang tinggi. Penelitian ini menjadi dasar penggunaan algoritma SVM dalam mendeteksi komentar toxic pada TikTok (Handayanto et al., 2019).

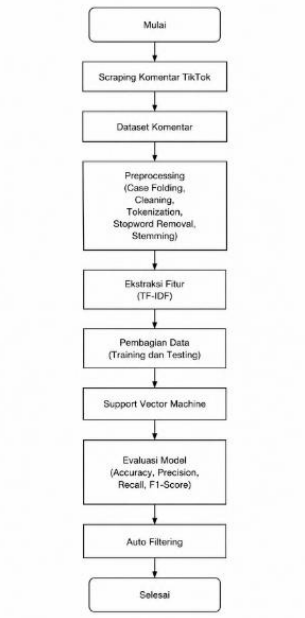
Meskipun demikian, penelitian yang mengintegrasikan metode TF-IDF, algoritma SVM, dan fitur *auto filtering* menggunakan data komentar TikTok hasil *scraping* masih relatif terbatas. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem deteksi komentar *toxic* yang tidak hanya melakukan klasifikasi, tetapi juga menerapkan mekanisme *auto filtering* untuk mendukung proses moderasi komentar secara otomatis.

Berdasarkan beberapa penelitian terdahulu, algoritma Support Vector Machine menunjukkan performa yang baik dalam klasifikasi teks. Namun, sebagian besar penelitian masih berfokus pada proses klasifikasi tanpa mengimplementasikan hasil prediksi sebagai mekanisme penyaringan komentar secara otomatis. Oleh karena itu, penelitian ini mengembangkan penerapan SVM sebagai dasar implementasi auto-filtering pada komentar TikTok.

3. METHOD, DATA, AND ANALYSIS

Penelitian ini menggunakan pendekatan text mining untuk mendeteksi komentar toxic pada platform TikTok menggunakan algoritma Support Vector Machine (SVM). Tahapan penelitian dimulai dari proses pengumpulan data komentar melalui teknik scraping, dilanjutkan dengan preprocessing data yang meliputi case folding, cleaning, tokenization, stopword removal, dan stemming.

Selanjutnya dilakukan ekstraksi fitur menggunakan metode Term Frequency–Inverse Document Frequency (TF-IDF), kemudian proses klasifikasi menggunakan algoritma SVM. Performa model dievaluasi menggunakan confusion matrix dengan metrik accuracy, precision, recall, dan F1-score. Tahap akhir penelitian adalah implementasi fitur auto filtering untuk menyaring komentar yang teridentifikasi sebagai toxic. Alur penelitian ditunjukkan pada Gambar 1.



Gambar 1 . Flowchart Tahapan Penelitian

Source: Processed by the Authors (2026)

3.1. Pengumpulan data

Tahap pertama dalam penelitian ini adalah pengumpulan data yang bertujuan memperoleh dataset komentar TikTok sebagai bahan penelitian. Data dikumpulkan melalui proses *scraping* menggunakan aplikasi TikTok *Comments Scraper* (Fitrah & Anggreini, n.d.). Informasi mengenai sumber dan karakteristik dataset yang digunakan dalam penelitian ini disajikan pada Tabel 1.

Tabel 1. Deskripsi Dataset

Komponen	Keterangan
Sumber Data	Komentar pada beberapa video TikTok
Metode Pengambilan Data	*Scraping* menggunakan TikTok Comments Scraper
Bentuk Data	Data teks komentar
Format Penyimpanan	CSV (*.csv*)
Tahapan Selanjutnya	*Preprocessing*, ekstraksi fitur TF-IDF, klasifikasi SVM, evaluasi model, dan *auto filtering*

Source: Processed by the Authors (2026)

3.2. *Preprocessing Data*

Tahap preprocessing bertujuan membersihkan dan menyiapkan data sebelum dilakukan proses klasifikasi. Data komentar yang diperoleh masih mengandung berbagai karakter yang tidak diperlukan sehingga perlu dilakukan beberapa tahapan pengolahan (Hasan et al., 2025).

Tahapan preprocessing yang dilakukan meliputi:

a. Case Folding

Case folding merupakan proses mengubah seluruh huruf menjadi huruf kecil (lowercase) sehingga tidak terdapat perbedaan antara huruf kapital dan huruf kecil.

b. Cleaning

Tahap cleaning dilakukan untuk menghapus URL, emoji, angka, tanda baca, karakter khusus, dan spasi berlebih sehingga hanya menyisakan teks yang diperlukan.

c. Tokenization

Tokenization merupakan proses memisahkan kalimat menjadi kumpulan kata (token) sehingga setiap kata dapat diproses secara individual.

d. Stopword Removal

Tahap ini bertujuan menghilangkan kata-kata umum yang tidak memiliki pengaruh besar terhadap proses klasifikasi, seperti "dan", "yang", "di", "ke", dan kata umum lainnya.

e. Stemming

Stemming dilakukan untuk mengubah kata menjadi bentuk dasar menggunakan pustaka Sastrawi sehingga kata yang memiliki makna sama dapat direpresentasikan dalam bentuk yang seragam.

Tabel 2.Tahapan Preprocessing

Tahapan	Fungsi	
Case Folding	Mengubah seluruh huruf menjadi huruf kecil	
Cleaning	Menghapus URL, emoji, angka, tanda baca, dan karakter khusus	
Tokenization	Memisahkan kalimat menjadi token kata	
Stopword Removal	Menghapus kata yang tidak memiliki makna penting	
Stemming	Mengubah kata menjadi bentuk dasar	

Source: Processed by the Authors (2026).

3.3. Ekstraksi Fitur Menggunakan TF-IDF

Setelah melalui tahapan *preprocessing*, data komentar diubah menjadi bentuk numerik menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF). Proses ini diperlukan karena algoritma *Support Vector Machine* (SVM) tidak dapat mengolah data teks secara langsung. Metode TF-IDF memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam suatu dokumen (*Term Frequency*) dan tingkat kepentingannya terhadap keseluruhan kumpulan dokumen (*Inverse Document Frequency*). Dengan demikian, kata-kata yang lebih representatif dalam membedakan komentar *toxic* dan *non-toxic* akan memiliki bobot yang lebih tinggi dibandingkan kata-kata yang sering muncul tetapi kurang memiliki nilai informasi.

Hasil ekstraksi fitur berupa matriks TF-IDF yang merepresentasikan setiap komentar dalam bentuk vektor numerik. Representasi tersebut kemudian digunakan sebagai data masukan pada proses pelatihan dan pengujian algoritma *Support Vector Machine* (SVM). Pada penelitian ini, proses ekstraksi fitur dilakukan menggunakan pustaka *Scikit-learn* pada bahasa pemrograman Python melalui kelas TfidfVectorizer, sehingga proses pembentukan fitur dapat dilakukan secara otomatis dan efisien. Penggunaan metode TF-IDF diharapkan mampu meningkatkan kualitas representasi data teks sehingga menghasilkan performa klasifikasi yang lebih baik.

3.4. Klasifikasi Menggunakan Support Vector Machine

Tahap berikutnya adalah proses klasifikasi menggunakan algoritma Support Vector Machine (SVM). Sebelum proses klasifikasi dilakukan, dataset dibagi menjadi data latih (*training data*) dan data uji (*testing data*) dengan perbandingan 80:20 menggunakan fungsi train_test_split() dari pustaka Scikit-learn. Data latih digunakan untuk membangun model klasifikasi, sedangkan data uji digunakan untuk mengukur kemampuan model dalam mengklasifikasikan data yang belum pernah dipelajari sebelumnya. Pembagian data tersebut bertujuan untuk memperoleh hasil evaluasi yang lebih objektif terhadap performa model.

Model Support Vector Machine (SVM) kemudian dilatih menggunakan data latih untuk mempelajari pola yang membedakan komentar *toxic* dan *non-toxic*. Algoritma SVM bekerja dengan mencari *hyperplane* optimal yang mampu memisahkan kedua kelas dengan margin terbesar sehingga menghasilkan proses klasifikasi yang lebih akurat. Setelah proses pelatihan selesai, model digunakan untuk memprediksi kelas pada data uji. Hasil prediksi tersebut selanjutnya dibandingkan dengan label sebenarnya sebagai dasar untuk mengevaluasi performa model menggunakan *confusion matrix*, *accuracy*, *precision*, *recall*, dan *F1-score* (Algoritma et al., 2024).

3.5. Evaluasi Model

Tahap evaluasi model bertujuan untuk mengukur tingkat keberhasilan algoritma *Support Vector Machine* (SVM) dalam mengklasifikasikan komentar ke dalam kategori *toxic* dan *non-toxic*. Evaluasi dilakukan setelah model selesai dilatih menggunakan data latih (*training data*) dan diuji menggunakan data uji (*testing data*). Hasil evaluasi digunakan untuk mengetahui kemampuan model dalam memberikan prediksi yang akurat serta mengidentifikasi kesalahan klasifikasi yang masih terjadi.Pada penelitian ini, proses evaluasi menggunakan *Confusion Matrix* sebagai dasar perhitungan performa model.*Confusion Matrix* merupakan metode evaluasi

yang menyajikan perbandingan antara hasil prediksi model dengan label sebenarnya sehingga dapat diketahui jumlah prediksi yang benar maupun yang salah. Melalui *Confusion Matrix*, diperoleh empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat komponen tersebut menjadi dasar dalam menghitung nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* (Helmiyah et al., 2025).

Metrik *Accuracy* digunakan untuk mengukur tingkat ketepatan model secara keseluruhan. Nilai *accuracy* menunjukkan persentase jumlah prediksi yang benar dibandingkan dengan seluruh data yang diuji. Semakin tinggi nilai *accuracy*, semakin baik kemampuan model dalam melakukan klasifikasi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Metrik *Precision* digunakan untuk mengukur ketepatan model dalam mengidentifikasi komentar yang diprediksi sebagai *toxic*. Nilai *precision* yang tinggi menunjukkan bahwa sebagian besar komentar yang diklasifikasikan sebagai *toxic* memang benar termasuk kategori tersebut. Metrik ini penting untuk meminimalkan kesalahan klasifikasi berupa komentar *non-toxic* yang salah ditandai sebagai *toxic* (*False Positive*).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Metrik *Recall* digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh komentar yang benar-benar termasuk kategori *toxic*. Nilai *recall* yang tinggi menunjukkan bahwa model mampu menemukan sebagian besar komentar *toxic* sehingga risiko komentar berbahaya lolos dari proses penyaringan dapat diminimalkan.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Metrik *F1-Score* merupakan rata-rata harmonik antara *precision* dan *recall*. Metrik ini digunakan untuk memberikan penilaian yang lebih seimbang terhadap performa model, terutama ketika diperlukan keseimbangan antara ketepatan prediksi dan kemampuan model dalam mendeteksi seluruh komentar *toxic*. Oleh karena itu, *F1-Score* menjadi salah satu indikator penting dalam mengevaluasi kualitas model klasifikasi.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

Keterangan:

TP (*True Positive*) adalah jumlah komentar *toxic* yang berhasil diprediksi sebagai *toxic*.

TN (*True Negative*) adalah jumlah komentar *non-toxic* yang berhasil diprediksi sebagai *non-toxic*.

FP (*False Positive*) adalah jumlah komentar *non-toxic* yang diprediksi sebagai *toxic*.

FN (*False Negative*) adalah jumlah komentar *toxic* yang diprediksi sebagai *non-toxic*.

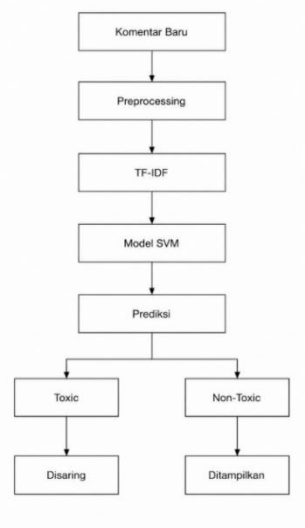
Penggunaan keempat metrik evaluasi tersebut memberikan gambaran yang lebih komprehensif mengenai performa model. *Accuracy* menunjukkan tingkat ketepatan prediksi secara keseluruhan, *Precision*

mengukur keakuratan prediksi pada kelas *toxic*, *Recall* menunjukkan kemampuan model dalam mendeteksi komentar *toxic*, sedangkan *F1-Score* memberikan ukuran keseimbangan antara *Precision* dan *Recall*. Dengan demikian, hasil evaluasi tidak hanya menunjukkan tingkat akurasi model, tetapi juga menggambarkan kemampuan model dalam mendeteksi komentar *toxic* secara efektif sehingga layak diterapkan sebagai dasar implementasi fitur *auto filtering*.

3.6. Implementasi Auto Filtering

Tahap terakhir dalam penelitian ini adalah implementasi fitur *auto filtering* berdasarkan hasil klasifikasi yang diperoleh dari algoritma *Support Vector Machine* (SVM). Setelah model selesai melakukan proses prediksi terhadap data komentar, setiap komentar akan diberikan label toxic atau non-toxic sesuai dengan hasil klasifikasi.Pada tahap ini, sistem melakukan proses penyaringan (*filtering*) secara otomatis berdasarkan label yang dihasilkan oleh model. Komentar yang diprediksi sebagai toxic akan ditandai dan dimasukkan ke dalam kategori komentar yang disaring sehingga tidak ditampilkan kepada pengguna. Sebaliknya, komentar yang diprediksi sebagai non-toxic tetap ditampilkan karena dianggap tidak mengandung unsur ujaran kebencian, penghinaan, maupun kata-kata yang bersifat negatif (Area, 2024).

Implementasi *auto filtering* dilakukan menggunakan bahasa pemrograman Python pada lingkungan Google Colab sebagai simulasi proses moderasi komentar. Hasil prediksi model disimpan ke dalam dataset dengan menambahkan kolom Prediksi yang berisi label *toxic* atau *non-toxic*. Selanjutnya dilakukan proses penyaringan menggunakan kondisi tertentu sehingga hanya komentar dengan label non-toxic yang ditampilkan sebagai komentar yang layak dipublikasikan.Proses implementasi tersebut menunjukkan bahwa model tidak hanya mampu mengklasifikasikan komentar, tetapi juga dapat diterapkan sebagai mekanisme moderasi otomatis. Dengan adanya fitur *auto filtering*, proses penyaringan komentar dapat dilakukan lebih cepat dibandingkan moderasi manual sehingga membantu mengurangi penyebaran komentar negatif pada platform media sosial(Erama, 2025).



Gambar 2. Alur Implementasi Auto Filtering

Source: Processed by the Authors (2026)

4. RESULT AND DISCUSSION

Pada penelitian ini dilakukan pengujian model Support Vector Machine (SVM) untuk mendeteksi komentar toxic pada TikTok. Proses pengujian dilakukan setelah data melalui tahapan preprocessing dan pembobotan TF-IDF. Selanjutnya, hasil klasifikasi dianalisis berdasarkan nilai accuracy, precision, recall, dan F1-score, kemudian diimplementasikan ke dalam fitur auto filtering.

4.1. Hasil Pengumpulan Data

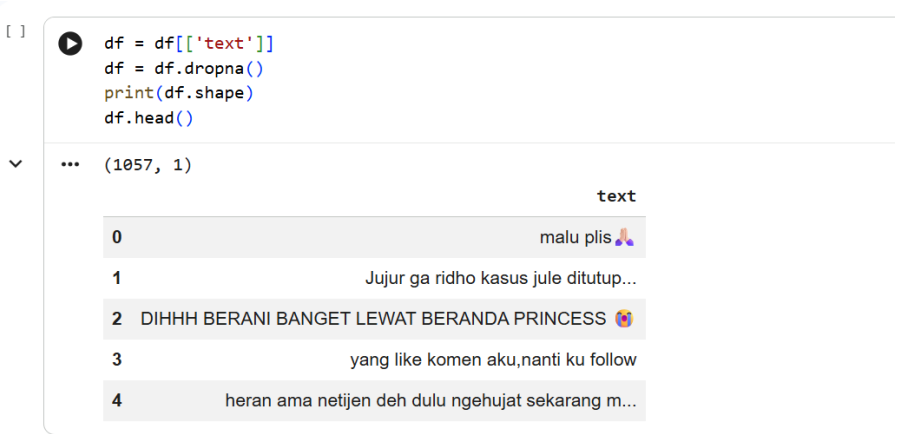
Data penelitian diperoleh melalui proses *scraping* komentar dari tiga video TikTok menggunakan aplikasi TikTok *Comments Scraper*. Hasil *scraping* diekspor dalam format CSV, kemudian seluruh komentar digabungkan (*merge*) menjadi satu dataset untuk memudahkan proses pelabelan, *preprocessing*, pelatihan model, dan evaluasi.

Meskipun dataset memuat beberapa atribut, seperti waktu komentar dan jumlah *like*, penelitian ini hanya menggunakan atribut komentar sebagai data utama dalam proses klasifikasi. Proses *scraping* ditunjukkan pada Gambar 3.

Tabel 3. Jumlah Data Hasil Scraping

Video	Jumlah Komentar
Video 1	500
Video 2	500
Video 3	57
Total	1.057

Source: Processed by the Authors (2026)

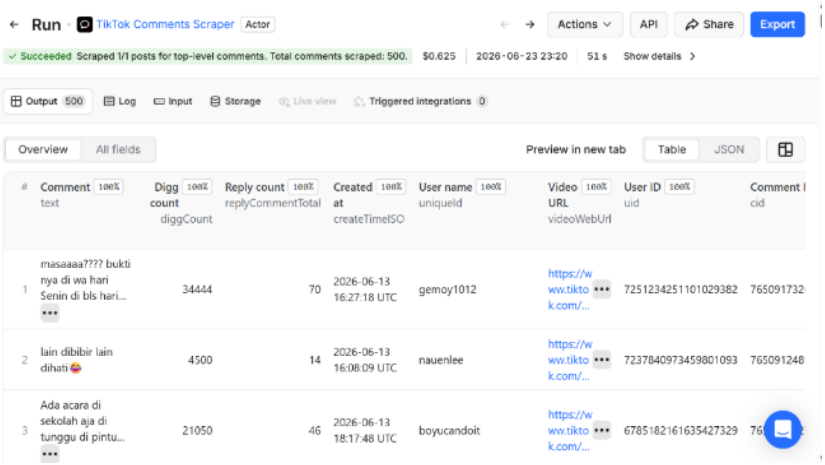


Gambar 3. Hasil Scraping Komentar TikTok Menggunakan TikTok Comments Scraper

Source: Processed by the Authors using Tiktok Comments Scraper (2026).

Berdasarkan Tabel 3, proses scraping menghasilkan total 1.057 komentar yang berasal dari tiga video TikTok. Seluruh komentar kemudian digabungkan menjadi satu dataset dan digunakan sebagai data penelitian.Pengumpulan data dilakukan menggunakan aplikasi Tiktok Comments Scarper sehingga seluruh komentar dapat diperoleh secara otomatis dan disimpan dalam format csv. Data yang diperoleh dari masing

masing video kemudian digabungkan menjadi satu dataset untuk mempermudah proses pengolahan dan analisis. Penggabungan dataset dilakukan agar jumlah data yang digunakan dalam penelitian menjadi lebih banyak dan mampu merepresentasikan berbagai karakteristik komentar pada platform Tiktok. Setelah proses scraping selesai, data komentar diekspor dalam format CSV kemudian diimpor ke Google Colab menggunakan pustaka Pandas. Pada penelitian ini hanya kolom text yang digunakan sebagai data utama karena berisi isi komentar pengguna. Selain itu dilakukan penghapusan data kosong (*missing value*) menggunakan fungsi dropna() sehingga diperoleh 1.057 komentar yang siap digunakan pada proses pelabelan dan *preprocessing*. Tampilan dataset hasil impor ditunjukkan pada Gambar 4.



Gambar 4. Tampilan Dataset Hasil Scraping pada Google Colab

Source: Processed by the Authors using Google Colab (2026)

4.2. Hasil Pelabelan Data

Sebelum data digunakan dalam proses klasifikasi, seluruh komentar yang telah diperoleh melalui proses *scraping* diberikan label secara manual menjadi dua kategori, yaitu toxic dan non-toxic. Proses pelabelan dilakukan berdasarkan isi komentar dengan memperhatikan adanya unsur ujaran kebencian (*hate speech*), penghinaan, kata-kata kasar, diskriminasi, maupun bentuk *cyberbullying*.

Komentar yang mengandung unsur-unsur tersebut diberi label toxic, sedangkan komentar yang bersifat positif, netral, atau tidak mengandung unsur negatif diberi label non-toxic.Pelabelan manual dilakukan untuk menghasilkan data yang sesuai sebagai data latih (*training data*) pada algoritma *Support Vector Machine* (SVM). Kualitas pelabelan sangat berpengaruh terhadap performa model karena label tersebut menjadi acuan utama dalam proses pembelajaran (*training*).

Proses pelabelan dilakukan secara manual oleh empat orang penulis sebagai anotator berdasarkan pedoman pelabelan yang telah ditentukan. Komentar yang mengandung ujaran kebencian, penghinaan, kata-kata kasar, diskriminasi, maupun cyberbullying diberi label toxic, sedangkan komentar yang tidak mengandung unsur tersebut diberi label non-toxic. Apabila terdapat perbedaan pendapat dalam proses pelabelan, dilakukan diskusi hingga diperoleh kesepakatan bersama sehingga label yang digunakan bersifat konsisten.

Distribusi data hasil pelabelan ditunjukkan pada Tabel 5.

Tabel 4. Distribusi Data Hasil Pelabelan

Label	Jumlah Komentar
Non-Toxic	909
Toxic	148
Total	1057

Source: Processed by the Authors (2026)

Berdasarkan Tabel 4, dataset terdiri atas 909 komentar non-toxic dan 148 komentar toxic, sehingga distribusi kelas bersifat tidak seimbang (*imbalanced dataset*). Kondisi ini dapat memengaruhi kemampuan model dalam mengenali kelas minoritas, sehingga evaluasi penelitian tidak hanya menggunakan nilai akurasi, tetapi juga precision, recall, dan F1-score untuk memberikan gambaran performa model secara lebih menyeluruh.

```
label
Non-Toxic    909
Toxic        148
Name: count, dtype: int64
                                          clean_text      label
0                                          malu plis      Toxic
1                   jujur ga ridho kasus jule tutup      Toxic
2         dihhh berani banget lewat beranda princess      Toxic
3                yang like komen aku nanti ku follow  Non-Toxic
4  heran ama netijen deh dulu ngehujat sekarang m...  Non-Toxic
5              yg like komen gw gw follow ga ngilang      Toxic
6        yang like komen gw gw follow nga ngilang ya      Toxic
7                     yang like gw follow seriusss  Non-Toxic
8                yang like gue follow grcep no tipu  Non-Toxic
9           yang like komen gue gue folow ga ngilang      Toxic
```

Gambar 5. Dataset Setelah Proses Pelabelan

Source: Processed by the Authors using Google Colab (2026)

Setelah proses pelabelan selesai, diperoleh dua kategori data, yaitu 909 komentar non-toxic dan 148 komentar toxic. Proses pelabelan dilakukan secara manual berdasarkan isi komentar dengan mengacu pada keberadaan unsur ujaran kebencian, penghinaan, kata-kata kasar, maupun bentuk *cyberbullying*. Dataset hasil pelabelan kemudian digunakan sebagai data masukan pada tahap *preprocessing* sebelum dilakukan ekstraksi fitur menggunakan metode TF-IDF.Contoh dataset yang telah diberi label ditampilkan pada Gambar 5.

4.3. Prepocessing Data

Pada tahap ini dilakukan proses *preprocessing* untuk membersihkan data komentar hasil *scraping* agar siap digunakan pada proses klasifikasi. Tahapan *preprocessing* bertujuan mengurangi *noise* pada data teks sehingga dapat meningkatkan kualitas fitur yang dihasilkan. Tahapan yang dilakukan meliputi case folding, cleaning, dan stemming.

a. Case Folding dilakukan dengan mengubah seluruh huruf menjadi huruf kecil (*lowercase*) agar penulisan kata menjadi seragam.
b. Cleaning bertujuan menghapus karakter yang tidak diperlukan seperti tanda baca, emoji, URL, angka, maupun simbol sehingga hanya menyisakan kata-kata yang relevan. Penghapusan emoji dilakukan karena penelitian ini berfokus pada analisis teks menggunakan TF-IDF. Emoji dihapus untuk mengurangi *noise* pada data sehingga proses ekstraksi fitur menjadi lebih optimal. Meskipun demikian, emoji dapat mengandung informasi sentimen sehingga dapat dipertimbangkan pada penelitian selanjutnya.
c. Stemming dilakukan untuk mengubah kata berimbuhan menjadi kata dasar menggunakan algoritma stemming bahasa Indonesia sehingga kata yang memiliki makna sama dapat direpresentasikan dalam bentuk yang seragam.

Hasil proses *preprocessing* ditunjukkan pada Gambar 6.

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()

X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)

print(X_train_tfidf.shape)
```

```
(845, 1250)
```

Gambar 6. Hasil Preprocessing Data Komentar TikTok

Source: Processed by the Authors using Google Colab (2026)

Berdasarkan Gambar 6 terlihat bahwa setiap komentar mengalami perubahan secara bertahap. Sebagai contoh, komentar "DIHHH BERANI BANGET LEWAT BERANDA PRINCESS" setelah melalui proses *case folding* berubah menjadi huruf kecil seluruhnya, kemudian pada tahap *cleaning* emoji dihapus sehingga hanya tersisa teks, dan pada tahap *stemming* kata-kata diubah ke bentuk dasar tanpa mengubah makna utama komentar. Hasil *preprocessing* inilah yang kemudian digunakan sebagai masukan pada proses ekstraksi fitur menggunakan metode TF-IDF.

4.4. Ekstraksi Fitur TF IDF

Setelah data komentar selesai melalui proses *preprocessing*, tahap selanjutnya adalah ekstraksi fitur menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Metode ini digunakan untuk mengubah data teks menjadi data numerik sehingga dapat diproses oleh algoritma *Support Vector Machine* (SVM). TF-IDF memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam suatu dokumen dan tingkat keunikannya terhadap seluruh dokumen.

Semakin penting suatu kata dalam membedakan dokumen, maka semakin besar bobot TF-IDF yang diperoleh. Pada penelitian ini, proses ekstraksi fitur dilakukan menggunakan kelas TfidfVectorizer dari pustaka *Scikit-learn*. Data latih (*training data*) digunakan untuk membangun kosakata (*vocabulary*) sekaligus menghitung bobot TF-IDF melalui fungsi fit_transform(), sedangkan data uji (*testing data*) hanya dilakukan proses transform() menggunakan kosakata yang telah dibentuk sebelumnya. Cara ini bertujuan untuk menghindari terjadinya *data leakage* sehingga proses pengujian model menjadi lebih objektif. Hasil ekstraksi fitur ditunjukkan pada Gambar 7.

```
#case folding
df_hasil = pd.DataFrame()

df_hasil['Text Asli'] = df['text']

df_hasil['Case Folding'] = df['text'].str.lower()

df_hasil['Cleaning'] = df_hasil['Case Folding'].apply(cleaning)

df_hasil['Stemming'] = df_hasil['Cleaning'].apply(stemmer.stem)

df_hasil.head(10)
```

	Text Asli	Case Folding	Cleaning	Stemming
0	malu plis🙏	malu plis🙏	malu plis	malu plis
1	Jujur ga ridho kasus jule ditutup...	jujur ga ridho kasus jule ditutup...	jujur ga ridho kasus jule ditutup	jujur ga ridho kasus jule tutup
2	DIHHH BERANI BANGET LEWAT BERANDA PRINCESS 😭	dihhh berani banget lewat beranda princess 😭	dihhh berani banget lewat beranda princess	dihhh berani banget lewat beranda princess
3	yang like komen aku,nanti ku follow	yang like komen aku,nanti ku follow	yang like komen aku nanti ku follow	yang like komen aku nanti ku follow
4	heran ama netijen deh dulu ngehujat sekarang m...	heran ama netijen deh dulu ngehujat sekarang m...	heran ama netijen deh dulu ngehujat sekarang m...	heran ama netijen deh dulu ngehujat sekarang m...
5	yg like komen gw ,gw follow\n(ga ngilang)	yg like komen gw ,gw follow\n(ga ngilang)	yg like komen gw gw follow ga ngilang	yg like komen gw gw follow ga ngilang
6	yang like komen gw. gw follow\n(nga ngilang ya)	yang like komen gw. gw follow\n(nga ngilang ya)	yang like komen gw gw follow nga ngilang ya	yang like komen gw gw follow nga ngilang ya
7	yang like gw follow,\nseriusss	yang like gw follow,\nseriusss	yang like gw follow seriusss	yang like gw follow seriusss
8	yang like gue follow grcep no tipu"	yang like gue follow grcep no tipu"	yang like gue follow grcep no tipu	yang like gue follow grcep no tipu
9	yang like komen gue,gue folow\n(ga ngilang)	yang like komen gue,gue folow\n(ga ngilang)	yang like komen gue gue folow ga ngilang	yang like komen gue gue folow ga ngilang

Gambar 7.Proses Ekstraksi Fitur Menggunakan TF-IDF

Source: Processed by the Authors using Google Colab (2026)

Berdasarkan Gambar 7 diperoleh ukuran matriks TF-IDF sebesar (845, 1250). Nilai tersebut menunjukkan bahwa terdapat 845 data latih yang berhasil dikonversi menjadi 1.250 fitur kata (*terms*). Setiap fitur merepresentasikan bobot TF-IDF dari suatu kata pada komentar sehingga seluruh komentar telah berubah dari bentuk teks menjadi vektor numerik. Representasi numerik inilah yang kemudian digunakan sebagai masukan pada proses pelatihan model menggunakan algoritma Support Vector Machine.

4.5. Pelatihan Model Support Vector Machine (SVM)

Sebelum proses pelatihan model dilakukan, dataset dibagi menjadi data latih (*training data*) dan data uji (*testing data*) dengan perbandingan 80:20 menggunakan fungsi train_test_split() dari pustaka Scikit-learn. Pembagian data dilakukan dengan menerapkan *stratified sampling* (stratify=y) agar proporsi kelas **toxic** dan **non-toxic** tetap terjaga pada data latih maupun data uji. Selain itu, digunakan nilai random_state = 42 untuk memastikan proses pembagian data dapat direproduksi.Pada penelitian ini, data dibagi menjadi data latih dan data uji dengan rasio 80:20. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk mengevaluasi kemampuan model dalam mengklasifikasikan komentar yang belum pernah dipelajari sebelumnya.

Proses ekstraksi fitur dilakukan menggunakan metode *Term Frequency-Inverse Document Frequency* (*TF-IDF*) melalui kelas *TfidfVectorizer*() pada pustaka Scikit-learn dengan konfigurasi bawaan (*default*). Selanjutnya,

proses klasifikasi menggunakan algoritma *Support Vector Machine* (*SVM*) melalui kelas SVC() dengan *kernel linear* dan *random_state* = 42, sedangkan parameter lainnya menggunakan konfigurasi bawaan dari Scikit-learn.

Setelah data komentar berhasil diubah menjadi representasi numerik menggunakan metode TF-IDF, tahap selanjutnya adalah pelatihan model menggunakan algoritma *Support Vector Machine* (SVM). Pada penelitian ini digunakan implementasi SVM dari pustaka Scikit-learn melalui kelas SVC dengan kernel linear. Pemilihan kernel linear dilakukan karena sesuai untuk klasifikasi data teks yang memiliki dimensi fitur tinggi serta mampu menghasilkan performa klasifikasi yang baik.Proses pelatihan dilakukan menggunakan data latih (*training data*) yang berjumlah 845 komentar. Setelah model selesai dilatih, dilakukan proses pengujian menggunakan data uji (*testing data*) untuk mengetahui kemampuan model dalam mengklasifikasikan komentar yang belum pernah digunakan pada tahap pelatihan. Setiap komentar pada data uji diprediksi ke dalam kategori *toxic* atau *non-toxic* berdasarkan pola yang telah dipelajari oleh model. Hasil prediksi tersebut kemudian dibandingkan dengan label sebenarnya untuk mengetahui tingkat ketepatan klasifikasi. Proses ini menjadi dasar dalam evaluasi performa model menggunakan *confusion matrix* dan metrik *accuracy*, *precision*, *recall*, serta *F1-score*, sehingga dapat diketahui sejauh mana algoritma Support Vector Machine (SVM) mampu mendeteksi komentar *toxic* secara akurat.

Pada tahap ini model mempelajari pola dari setiap fitur TF-IDF untuk membedakan komentar yang termasuk kategori toxic dan non-toxic. Selama proses pelatihan, algoritma SVM akan mencari *hyperplane* terbaik yang mampu memisahkan kedua kelas tersebut dengan margin maksimum sehingga diharapkan model dapat melakukan klasifikasi secara optimal terhadap data yang belum pernah dilihat sebelumnya.Implementasi proses pelatihan model ditunjukkan pada Gambar 8.

```
# Training ulang SVM
from sklearn.svm import SVC

svm_model_baru = SVC(
    kernel='linear',
    random_state=42
)

svm_model_baru.fit(X_train_tfidf, y_train)

print("Training selesai")
```

```
Training selesai
```

Gambar 8. Proses Pelatihan Model Support Vector Machine

Source: Processed by the Authors using Google Colab (2026).

Berdasarkan Gambar 8, proses pelatihan model berhasil dijalankan tanpa mengalami kesalahan yang ditunjukkan dengan munculnya pesan "Training selesai". Hal ini menunjukkan bahwa model Support Vector Machine telah berhasil mempelajari pola dari data latih dan siap digunakan pada tahap berikutnya, yaitu

melakukan prediksi terhadap data uji (*testing data*) serta mengevaluasi performa model menggunakan beberapa metrik evaluasi.

4.6 Evaluasi Model

Setelah model *Support Vector Machine* selesai dilatih menggunakan data latih, tahap berikutnya adalah melakukan evaluasi terhadap performa model menggunakan data uji. Evaluasi dilakukan untuk mengetahui kemampuan model dalam mengklasifikasikan komentar ke dalam kategori *toxic* dan *non-toxic*.

Nilai *Accuracy* digunakan untuk mengukur tingkat ketepatan prediksi model secara keseluruhan, sedangkan *Precision* menunjukkan ketepatan model dalam mengidentifikasi komentar yang diprediksi sebagai *toxic*. *Recall* digunakan untuk mengukur kemampuan model dalam menemukan seluruh komentar yang benar-benar termasuk kategori *toxic*, sementara *F1-Score* merupakan rata-rata harmonik antara *Precision* dan *Recall* yang memberikan penilaian lebih seimbang terhadap performa model. Penggunaan keempat metrik evaluasi tersebut diharapkan mampu memberikan hasil analisis yang lebih objektif sehingga kualitas model Support Vector Machine dapat diketahui secara menyeluruh sebelum diterapkan pada fitur *auto filtering*.

4.5.1. Accuracy

Nilai *accuracy* digunakan untuk mengukur tingkat ketepatan model secara keseluruhan dalam mengklasifikasikan komentar. Metrik ini menunjukkan persentase jumlah prediksi yang benar dibandingkan dengan seluruh data uji. Hasil perhitungan *accuracy* ditunjukkan pada Gambar 9.

```
#menghitung jumlah akurasi
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("Accuracy (%):", accuracy * 100)
```

```
Accuracy: 0.9528301886792453
Accuracy (%): 95.28301886792453
```

Gambar 9. Hasil Perhitungan Accuracy

Source: Processed by the Authors using Google Colab (2026)

Berdasarkan hasil pengujian, model memperoleh nilai Accuracy sebesar 95,28%. Nilai tersebut menunjukkan bahwa dari seluruh data uji, sekitar 95% komentar berhasil diklasifikasikan dengan benar oleh model Support Vector Machine. Tingginya nilai *accuracy* menunjukkan bahwa kombinasi tahapan *preprocessing*, ekstraksi fitur menggunakan TF-IDF, dan algoritma SVM mampu menghasilkan model klasifikasi yang memiliki performa yang baik dalam mendeteksi komentar *toxic* pada platform TikTok.

4.5.2. Classification Report

Selain menggunakan nilai *accuracy*, penelitian ini juga mengevaluasi performa model menggunakan *Classification Report*. Evaluasi ini memberikan informasi yang lebih rinci mengenai nilai *Precision*, *Recall*, *F1 Score* , dan jumlah data (*support*) pada masing masing kelas. Melalui metrik **Precision**, **Recall**, dan **F1-Score**, dapat diketahui kemampuan model dalam mengidentifikasi komentar toxic secara lebih rinci. Evaluasi ini penting karena pada permasalahan klasifikasi komentar toxic, model tidak hanya dituntut memiliki nilai accuracy yang tinggi, tetapi juga harus mampu meminimalkan kesalahan klasifikasi, terutama komentar toxic yang tidak berhasil terdeteksi.

Hasil *Classification Report* kemudian dianalisis untuk mengetahui keseimbangan performa model pada setiap kelas. Nilai *Precision* yang tinggi menunjukkan bahwa sebagian besar komentar yang diprediksi sebagai *toxic* memang benar termasuk kategori tersebut. Sementara itu, nilai *Recall* menggambarkan kemampuan model dalam mendeteksi seluruh komentar *toxic* yang terdapat pada data uji. Nilai *F1-Score* digunakan sebagai indikator keseimbangan antara *Precision* dan *Recall*, sehingga memberikan gambaran yang lebih komprehensif mengenai kualitas model. Berdasarkan hasil evaluasi tersebut, dapat diketahui bahwa algoritma *Support Vector Machine* (SVM) mampu memberikan performa klasifikasi yang baik dan layak digunakan sebagai dasar implementasi fitur *auto filtering*.

```
#Classification Report
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
                precision    recall  f1-score   support

             0       0.95      1.00      0.97       182
             1       1.00      0.67      0.80        30

      accuracy                           0.95       212
     macro avg       0.97      0.83      0.89       212
  weighted avg       0.96      0.95      0.95       212
```

Gambar 10. Hasil Classification Report

Source: Processed by the Authors using Google Colab (2026)

Berdasarkan hasil *Classification Report*, model memperoleh nilai weighted average Precision sebesar 0,96, Recall sebesar 0,95, dan F1-Score sebesar 0,95. Pada kelas non-toxic, model memperoleh nilai *precision* sebesar 0,95, *recall* sebesar 1,00, dan *F1-Score* sebesar 0,97, yang menunjukkan bahwa hampir seluruh komentar non-toxic berhasil dikenali dengan baik. Sementara itu, pada kelas toxic, model memperoleh nilai *precision* sebesar 1,00, *recall* sebesar 0,67, dan *F1-Score* sebesar 0,80. Nilai *recall* yang lebih rendah menunjukkan bahwa masih terdapat beberapa komentar *toxic* yang belum berhasil dideteksi oleh model.

Nilai recall pada kelas toxic yang masih sebesar 0,67 menunjukkan bahwa masih terdapat komentar toxic yang belum berhasil dikenali oleh model. Hal ini kemungkinan disebabkan oleh penggunaan bahasa

gaul, singkatan, maupun kata-kata yang bersifat ambigu sehingga model mengalami kesulitan dalam mengenali pola komentar tersebut.

4.5.3. Confusion Matrix

Untuk mengetahui jumlah prediksi yang benar maupun yang salah pada masing-masing kelas, dilakukan evaluasi menggunakan *Confusion Matrix*. Metode ini menyajikan perbandingan antara hasil prediksi model dengan label sebenarnya sehingga dapat diketahui jumlah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat komponen tersebut memberikan informasi mengenai kemampuan model dalam mengidentifikasi komentar *toxic* dan *non-toxic*, sekaligus menunjukkan jenis kesalahan klasifikasi yang masih terjadi selama proses pengujian.

Dengan demikian, Confusion Matrix tidak hanya digunakan untuk menghitung nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score,* tetapi juga membantu menganalisis performa model secara lebih mendalam.Melalui hasil *Confusion Matrix*, dapat diketahui apakah model lebih sering melakukan kesalahan dalam mengklasifikasikan komentar *toxic* sebagai *non-toxic* (*False Negative*) atau sebaliknya mengklasifikasikan komentar *non-toxic* sebagai *toxic* (*False Positive*). Informasi tersebut sangat penting karena dapat menjadi dasar untuk mengevaluasi kelebihan dan keterbatasan model yang dikembangkan. Analisis terhadap Confusion Matrix juga memberikan gambaran mengenai efektivitas algoritma *Support Vector Machine* (SVM) dalam membedakan kedua kelas berdasarkan pola yang telah dipelajari selama proses pelatihan. Hasil *Confusion Matrix* ditunjukkan pada Gambar 11 dan Gambar 12.

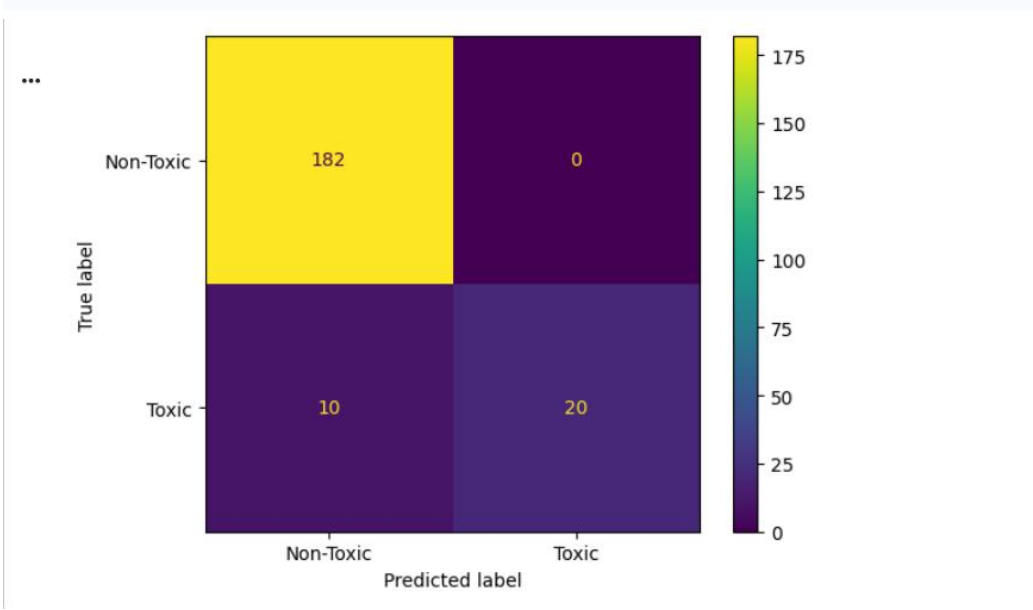
```
#confusion matrix
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)

print(cm)
```

```
[[182   0]
 [ 10  20]]
```

Gambar 11. Hasil Confusion Matrix



Gambar 12.Visualisasi Confusion Matrix

Source: Processed by the Authors using Google Colab (2026)

Hasil *Confusion Matrix* pada Gambar 11 dan Gambar 12 menunjukkan bahwa terdapat 182 komentar non-toxic yang berhasil diklasifikasikan dengan benar sebagai non-toxic (True Negative) dan 20 komentar toxic yang berhasil dikenali dengan benar sebagai toxic (True Positive). Selain itu, terdapat 10 komentar toxic yang diprediksi sebagai non-toxic (False Negative), sedangkan tidak terdapat komentar non-toxic yang salah diprediksi sebagai toxic (False Positive).

Hasil tersebut menunjukkan bahwa model *Support Vector Machine* (SVM) memiliki kemampuan yang sangat baik dalam membedakan komentar non-toxic, karena seluruh komentar non-toxic pada data uji berhasil diklasifikasikan dengan benar tanpa menghasilkan *False Positive*. Namun, masih terdapat beberapa komentar toxic yang tidak berhasil terdeteksi sehingga diklasifikasikan sebagai non-toxic. Kondisi ini menunjukkan bahwa meskipun performa model sudah sangat baik, masih terdapat peluang untuk meningkatkan kemampuan model dalam mengenali seluruh komentar toxic agar risiko komentar berbahaya yang lolos dari proses penyaringan dapat semakin diminimalkan.

Berdasarkan hasil confusion matrix, terdapat 10 komentar toxic yang diprediksi sebagai non-toxic (*false negative*). Kesalahan klasifikasi tersebut menunjukkan bahwa model masih memiliki keterbatasan dalam mendeteksi komentar toxic yang disampaikan secara tidak langsung atau menggunakan variasi bahasa tertentu. Oleh karena itu, masih terdapat kemungkinan komentar toxic lolos dari proses penyaringan.

4.6. Implementasi Auto Filtering

Setelah model Support Vector Machine (SVM) berhasil dilatih dan dievaluasi, tahap selanjutnya adalah implementasi fitur Auto Filtering. Fitur ini bertujuan untuk melakukan penyaringan komentar secara otomatis berdasarkan hasil prediksi model sehingga komentar yang mengandung unsur toxic, seperti ujaran kebencian, penghinaan, maupun kata-kata kasar, tidak ditampilkan kepada pengguna. Implementasi fitur ini

merupakan bentuk penerapan langsung dari model klasifikasi yang telah dikembangkan, sehingga proses moderasi komentar tidak lagi bergantung sepenuhnya pada pemeriksaan secara manual. Dengan adanya mekanisme ini, proses penyaringan komentar dapat dilakukan secara lebih cepat, konsisten, dan efisien, terutama ketika harus menangani komentar dalam jumlah yang besar.

Pada penelitian ini dibuat sebuah fungsi auto_filter() yang menerima masukan berupa sebuah komentar dari pengguna. Sebelum dilakukan klasifikasi, komentar terlebih dahulu melalui tahapan preprocessing menggunakan fungsi cleaning() agar format teks sesuai dengan data yang digunakan pada proses pelatihan model. Tahapan tersebut meliputi proses case folding, penghapusan URL, emoji, tanda baca, angka, karakter khusus, serta spasi berlebih sehingga menghasilkan teks yang lebih bersih dan siap diproses. Selanjutnya komentar diubah ke dalam bentuk vektor numerik menggunakan TF-IDF Vectorizer, sehingga setiap kata memiliki bobot yang dapat diproses oleh algoritma Support Vector Machine (SVM). Vektor hasil ekstraksi fitur kemudian digunakan sebagai masukan bagi model untuk menentukan apakah komentar termasuk kategori toxic atau non-toxic.

Selain memanfaatkan model klasifikasi Support Vector Machine (SVM), sistem juga menerapkan mekanisme penyaringan berdasarkan daftar kata-kata toxic (toxic words) yang telah ditentukan sebelumnya. Pemeriksaan ini dilakukan sebelum proses prediksi menggunakan model. Apabila komentar mengandung kata yang termasuk dalam daftar tersebut, sistem akan langsung memberikan status "KOMENTAR DIBLOKIR" tanpa melalui proses klasifikasi SVM. Pendekatan ini bertujuan untuk mempercepat proses penyaringan terhadap komentar yang secara eksplisit mengandung kata-kata kasar, penghinaan, atau ujaran kebencian, sehingga dapat mengurangi beban komputasi pada model klasifikasi sekaligus mempercepat proses moderasi.

Sebaliknya, apabila komentar tidak mengandung kata yang terdapat dalam daftar toxic words, maka sistem akan melanjutkan proses klasifikasi menggunakan model Support Vector Machine (SVM). Model akan memprediksi kelas komentar berdasarkan pola yang telah dipelajari selama proses pelatihan. Jika hasil prediksi menunjukkan bahwa komentar termasuk kategori toxic, sistem akan memberikan keluaran "KOMENTAR DIBLOKIR", sedangkan apabila hasil prediksi menunjukkan kategori non-toxic, sistem akan menampilkan keluaran "KOMENTAR DITAMPILKAN". Hasil implementasi ini menunjukkan bahwa mekanisme Auto Filtering mampu mengintegrasikan proses preprocessing, ekstraksi fitur TF-IDF, dan klasifikasi menggunakan SVM ke dalam satu alur kerja yang berjalan secara otomatis. Dengan demikian, sistem yang dikembangkan tidak hanya mampu mendeteksi komentar toxic, tetapi juga dapat langsung menerapkan hasil klasifikasi tersebut sebagai dasar penyaringan komentar pada platform media sosial.

Implementasi auto-filtering pada penelitian ini masih berupa simulasi menggunakan Google Colab dan belum terintegrasi secara langsung dengan platform TikTok. Oleh karena itu, hasil implementasi menunjukkan potensi penerapan sistem, namun masih memerlukan pengembangan lebih lanjut agar dapat digunakan pada lingkungan nyata.

```
def auto_filter(comment):

    comment_clean = cleaning(comment)

    # cek kata toxic terlebih dahulu
    for word in toxic_words:
        if word in comment_clean:
            return "KOMENTAR DIBLOKIR"

    # jika tidak ada, gunakan SVM
    comment_tfidf = tfidf.transform([comment_clean])

    prediksi = svm_model_baru.predict(comment_tfidf)[0]

    if prediksi == 1:
        return "KOMENTAR DIBLOKIR"
    else:
        return "KOMENTAR DITAMPILKAN"
```

```
print(auto_filter("dasar goblok kamu"))
print(auto_filter("anjing banget"))
print(auto_filter("video ini bagus"))
```

```
KOMENTAR DIBLOKIR
KOMENTAR DIBLOKIR
KOMENTAR DITAMPILKAN
```

Gambar 13. Implementasi Auto Filtering.

Source: Processed by the Authors using Google Colab (2026)

Penelitian ini menggunakan data komentar yang bersifat publik pada platform TikTok dan hanya dimanfaatkan untuk kepentingan akademik. Seluruh proses analisis dilakukan tanpa menampilkan identitas pengguna sehingga privasi pengguna tetap terjaga. Data yang diperoleh hanya digunakan untuk proses penelitian dan tidak dimanfaatkan untuk tujuan lain di luar kepentingan ilmiah.

5. CONCLUSION AND SUGGESTION

Penelitian ini berhasil menerapkan metode text mining menggunakan algoritma Support Vector Machine (SVM) untuk mendeteksi komentar toxic pada platform TikTok serta mengimplementasikan fitur auto filtering sebagai mekanisme penyaringan komentar secara otomatis. Tahapan penelitian meliputi proses pengumpulan data melalui scraping, preprocessing data, ekstraksi fitur menggunakan TF-IDF, proses klasifikasi menggunakan algoritma Support Vector Machine (SVM), serta evaluasi performa model menggunakan Accuracy, Precision, Recall, dan F1-Score. Selain itu, penelitian ini masih menggunakan satu kali pembagian data (*train-test split*) dengan rasio 80:20 dan belum menerapkan metode *Stratified K-Fold Cross Validation*. Oleh karena itu, penelitian selanjutnya disarankan menggunakan teknik validasi tersebut agar hasil evaluasi model menjadi lebih stabil dan representatif.

Berdasarkan hasil pengujian, model memperoleh nilai accuracy sebesar 95,28%, precision sebesar 1,00, recall sebesar 0,67, dan F1-score sebesar 0,80 pada kelas toxic. Hasil tersebut menunjukkan bahwa algoritma SVM mampu memberikan performa klasifikasi yang baik dalam membedakan komentar toxic dan

non-toxic.Implementasi fitur auto filtering juga berhasil diterapkan dengan memanfaatkan hasil klasifikasi model sebagai dasar penyaringan komentar. Sistem mampu mengidentifikasi komentar yang mengandung unsur toxic dan secara otomatis memberikan status "KOMENTAR DIBLOKIR", sedangkan komentar yang termasuk kategori non-toxic tetap ditampilkan kepada pengguna.

Penerapan mekanisme ini memberikan manfaat praktis dalam membantu proses moderasi komentar menjadi lebih cepat, efisien, dan konsisten dibandingkan moderasi secara manual. Selain itu, penelitian ini memberikan kontribusi terhadap pengembangan penerapan text mining, machine learning, dan moderasi konten media sosial, khususnya dalam memanfaatkan algoritma SVM untuk mendukung proses penyaringan komentar secara otomatis.

Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. Dataset yang digunakan berasal dari proses scraping komentar pada beberapa video TikTok sehingga belum sepenuhnya merepresentasikan keberagaman komentar yang terdapat pada platform tersebut. Selain itu, model masih menghasilkan sejumlah False Negative, yaitu komentar toxic yang diprediksi sebagai non-toxic, sehingga masih terdapat kemungkinan komentar yang bersifat negatif tidak terdeteksi oleh sistem. Keterbatasan tersebut dipengaruhi oleh karakteristik bahasa pada media sosial yang sangat beragam, seperti penggunaan bahasa gaul, singkatan, kesalahan penulisan, serta variasi kata yang terus berkembang. Berdasarkan keterbatasan tersebut, penelitian selanjutnya disarankan menggunakan dataset yang lebih besar dan lebih beragam agar model mampu mempelajari lebih banyak variasi komentar. Meskipun model memperoleh nilai accuracy sebesar 95,28%, nilai recall pada kelas toxic sebesar 0,67 menunjukkan bahwa kemampuan model dalam mendeteksi seluruh komentar toxic masih perlu ditingkatkan agar risiko komentar toxic yang lolos dari proses penyaringan dapat diminimalkan.

Penelitian berikutnya juga dapat membandingkan performa Support Vector Machine (SVM) dengan algoritma lain, seperti Random Forest, Long Short-Term Memory (LSTM), atau model berbasis Transformer (BERT), untuk memperoleh hasil klasifikasi yang lebih optimal. Selain itu, fitur auto filtering dapat dikembangkan agar terintegrasi langsung dengan platform media sosial sehingga proses moderasi komentar dapat dilakukan secara real-time dan memberikan perlindungan yang lebih efektif terhadap penyebaran komentar toxic.

Declaration of AI and AI-assisted technologies in the writing process

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) to improve grammar, language, manuscript organization, and the clarity of academic writing. After using this tool, the authors carefully reviewed, edited, and verified all content to ensure its accuracy and suitability for publication. The AI tool was not used to generate, analyze, or interpret the research data. The authors take full responsibility for the content and conclusions presented in this publication.

REFERENCE

Algoritma, I., Vector, S., Untuk, M., Status, K., & Balita, S. P. (2024). *G-Tech : Jurnal Teknologi Terapan*. *8*(3), 2070–2079.

Area, U. M. (2024). *SARKASME MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) SKRIPSI Oleh : ZULNANDI ZIDAN LUBIS FAKULTAS TEKNIK SARKASME MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) SKRIPSI Diajukan sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana di Fakultas Teknik Universitas Medan Area*.

Arifin, N., Enri, U., & Sulistiyowati, N. (2021). *PENERAPAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DENGAN TF-IDF N-GRAM UNTUK TEXT CLASSIFICATION*. *6*(2).

Berindikasi, P. K., Syawal, M. A., & Pradhana, B. A. (2025). *IMPLEMENTASI METODE K-MEANS UNTUK Implementation Of The K-Means Method For Grouping Words Indicative Of Cyberbullying In*. *9*(2), 330–347. https://doi.org/10.52362/jisicom.v9i2.2163

Erama, R. (2025). *Pemanfaatan Platform Cloud Google Colab Untuk Scraping Komentar Tiktok Pada Konten Gorontalo sebagai Dasar Analisis Respons Warganet*. *1*, 124–134.

Fitrah, N. K., & Anggreini, N. L. (n.d.). *PENDIDIKAN MILITER BAGI PELAJAR MENGGUNAKAN*. *7*(3), 862–873.

Handayanto, A., Latifa, K., Saputro, N. D., & Waliyansyah, R. R. (2019). *Analisis dan Penerapan Algoritma Support Vector Machine (SVM) dalam Data Mining untuk Menunjang Strategi Promosi (Analysis and Application of Algorithm Support Vector Machine (SVM) in Data Mining to Support Promotional Strategies)*. *7*(November), 71–79.

Hasan, F. N., Informatika, T., Muhammadiyah, U., & Hamka, P. (2025). *ANALISIS SENTIMEN TANGGAPAN PENGGUNA APLIKASI BALE BY BTN MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)*. *4*(4), 315–326.

Helmiyah, S., Pramestiawan, R., & Lampung, R. (2025). *Analisis Komparatif Algoritma Machine Learning dengan Metrik Akurasi , Presisi , Recall , dan F1-Score pada Dataset Kacang Kering*. *6*(3), 152–159.

Hermawan, A., & Jowensen, I. (2023). *Implementasi Text-Mining untuk Analisis Sentimen pada Twitter dengan Algoritma Support Vector Machine*. *12*(1), 129–137.

Jasno, B. A., Fathoni, A. A., Putra, D. D., Zidane, M., Amsury, F., Supendar, H., Teknologi, P. S., Bina, U., Informatika, S., Sentimen, A., & Toxic, K. (2025). *ANALISIS SENTIMEN KOMENTAR BERPOTENSI TOXIC PADA MEDIA SOSIAL*. *16*(c), 193–201.

Krisnandi, D., Ambarwati, R. N., Asih, A. Y., Ardiansyah, A., & Pardede, H. F. (2023). *Analisis Komentar Cyberbullying Terhadap Kata yang Mengandung Toksisitas dan Agresi Menggunakan Bag of Words dan TF-IDF dengan Klasifikasi SVM*. *6*(2), 36–41.

Rahmadhani, S., Hakim, L., Wibowo, G. H., & Kristanto, S. P. (2020). *SISTEM REKOMENDASI PENELUSURAN BUKU BERBASIS CONTENT-BASED FILTERING DENGAN PEMBOBOTAN TF-RF*. 491–500.

Romadina, V. N., Juwita, O., & Pandunata, P. (2024). *Analisis Komentar Toxic Terhadap Informasi COVID-19 pada YouTube Kementerian Kesehatan Menggunakan Metode Naïve Bayes Classifier*. *9*(1), 92–99.

Romadoni, F., Umaidah, Y., & Sari, B. N. (2020). *Text Mining Untuk Analisis Sentimen Pelanggan Terhadap*

Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine. *09*, 247–253.

Sakun, K., Rudianto, C., Studi, P., Informasi, S., Informasi, F. T., Kristen, U., & Wacana, S. (2026). *Analisis Sentimen Komentar pada Kanal Youtube Team RRQ Menggunakan Algoritma Support Vector Machine*. *5*(2), 1321–1329.

Sentimen, K., Terhadap, M., Pns, G., Resign, Y., Lingkungan, K., & Classifier, B. (n.d.). *BAYES CLASSIFIER*.

Sidiq, R. P., Dermawan, B. A., & Umaidah, Y. (2020). *Sentimen Analisis Komentar Toxic pada Grup Facebook Game Online Menggunakan Klasifikasi Naïve Bayes*. *5*(3), 356–363.