



# Penerapan Support Vector Machine Dengan Smote Untuk Klasifikasi Sentimen Pada Data Ulasan Aplikasi Trading View

Muhammad Badri, Elin Haerani\*, Fadhilah Syafria, Okfalisa Okfalisa, Lola Oktavia

Fakultas Sains dan Teknologi, Program Studi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

Email: <sup>1</sup>11950113425@students.uin-suska.ac.id, <sup>2,\*</sup>elin.haerani@uin-suska.ac.id, <sup>3</sup>fadhilah.syafria@uin-suska.ac.id, <sup>4</sup>okfalisa@gmail.com, <sup>5</sup>lola.oktavia@uin-suska.ac.id.

Email Penulis Korespondensi: elin.haerani@uin-suska.ac.id

**Abstrak**—Di era digital, tanggapan pengguna terhadap aplikasi mobile berperan sebagai pemberi informasi yang sangat berharga bagi para developer dalam mengevaluasi performa aplikasi. Salah satu aplikasi populer di bidang keuangan dan investasi adalah TradingView, yang banyak digunakan untuk analisis teknikal oleh para trader. Tanggapan pengguna terhadap aplikasi ini mencerminkan berbagai sentimen pengguna, baik positif, negatif, maupun netral. Namun, banyaknya jumlah ulasan dan bentuk teks yang tidak terstruktur mengakibatkan proses analisis manual dinilai kurang efisien serta memiliki potensi bias subjektif yang tinggi. Dengan demikian, dibutuhkan penggunaan metode klasifikasi otomatis yang mampu mengolah data teks dengan tingkat keakuratan yang cukup baik. Study ini dilakukan dengan tujuan untuk mengimplementasikan teknik “Support Vector Machine (SVM)” dalam mengklasifikasikan tanggapan pemakai aplikasi TradingView. Guna mengatasi permasalahan ketidakseimbangan distribusi data antar kelas sentimen, juga memanfaatkan pendekatan “Synthetic Minority Over-sampling Technique (SMOTE)” studi ini menggunakan pemanfaatan 10.000 tanggapan yang didapat dengan teknik web scraping pada Google Playstore. Alur Study terdiri dari tahap prapemrosesan teks, penerapan teknik “Term Frequency Inverse Document Frequency (TF-IDF)” untuk ekstraksi fitur, penyeimbangan data, proses pelatihan menggunakan SVM, serta evaluasi hasil model. Hasil evaluasi diperoleh bahwa penerapan algoritma SVM dengan bantuan Synthetic Minority Oversampling Technique (SMOTE) memberikan tingkat akurasi sebesar  $\pm 85,56\%$  pada proporsi data (70:30, 80:20, 90:10). Pada setiap skenario, nilai F1-score tertinggi dicapai pada kelas sentimen positif, sementara performa pada kelas minoritas (negatif dan netral) menunjukkan peningkatan setelah dilakukan penyeimbangan data dengan SMOTE dengan rata-rata peningkatan nilai F1-score pada kelas negatif sebesar 1,67% dan 10,67% pada kelas netral sedangkan tanpa penerapan teknik SMOTE nilai F1-score negatif rata-rata sebesar  $\pm 57\%$  dan netral tidak terdeteksi 0,00%. Lebih lanjut, hasil validasi menggunakan K-Fold Cross Validation menghasilkan akurasi rata-rata 89,20%, sedangkan setelah ditambahkan SMOTE akurasi meningkat menjadi 95,10%. Kenaikan ini konsisten pada proporsi data (70:30, 80:20, 90:10) dengan rata-rata peningkatan 5,44%. Temuan ini mengesakan bahwa integrasi SVM dengan SMOTE tidak hanya mampu meningkatkan kemampuan klasifikasi pada data yang tidak seimbang, tetapi juga menjaga stabilitas performa model. Sehingga, penelitian ini turut berperan dalam kemajuan sistem klasifikasi sentimen berbasis otomatis, khususnya pada ulasan aplikasi finansial berbasis mobile, serta dapat menjadi rujukan bagi penelitian lanjutan di bidang analisis ulasan pengguna pada aplikasi serupa.

**Kata Kunci:** Sentimen Analisis; Support Vector Machine; Synthetic Minority Over-sampling Technique; Tanggapan Aplikasi; TradingView; Klasifikasi Sentimen.

**Abstract**—In the digital era, user feedback on mobile applications serves as highly valuable information for developers to evaluate app performance. One popular application in the field of finance and investment is TradingView, widely used for technical analysis by traders. User feedback on this application reflects various user sentiments, including positive, negative, and neutral. However, the large volume of reviews and the unstructured nature of text data make manual analysis inefficient and prone to high subjective bias. Therefore, the use of automatic classification methods capable of processing text data with reasonable accuracy is required.

This study aims to implement the “Support Vector Machine (SVM)” technique to classify user feedback on the TradingView application. To address the issue of imbalanced sentiment class distribution, the study also employs the “Synthetic Minority Over-sampling Technique (SMOTE)”. The study utilizes 10,000 reviews obtained via web scraping from the Google Play Store. The study workflow consists of text preprocessing, feature extraction using “Term Frequency-Inverse Document Frequency (TF-IDF)”, data balancing, SVM model training, and model evaluation. The evaluation results show that the application of SVM with SMOTE achieves an accuracy of approximately  $\pm 85.56\%$  across data splits (70:30, 80:20, 90:10). In each scenario, the highest F1-score was achieved for the positive sentiment class, while the performance of minority classes (negative and neutral) improved after data balancing with SMOTE, with an average F1-score increase of 1.67% for the negative class and 10.67% for the neutral class. Without SMOTE, the average negative F1-score was  $\pm 57\%$ , and the neutral class was undetected (0.00%). Furthermore, validation using K-Fold Cross Validation yielded an average accuracy of 89.20%, which increased to 95.10% after applying SMOTE. This improvement was consistent across all data proportions (70:30, 80:20, 90:10), with an average increase of 5.44%. These findings confirm that integrating SVM with SMOTE not only enhances classification performance on imbalanced data but also maintains model stability. Therefore, this study contributes to the advancement of automated sentiment classification systems, particularly for financial mobile app reviews, and can serve as a reference for future research in user review analysis on similar applications.

**Keywords:** Sentiment Analysis; Support Vector Machine; Synthetic Minority Over-sampling Technique; Application Reviews; TradingView; Sentiment Classification.

## 1. PENDAHULUAN

Pada era ketika teknologi berkembang dengan sangat pesat, aplikasi mobile telah memainkan peran penting dalam menunjang aktivitas sehari-hari masyarakat, termasuk pada bidang keuangan dan investasi. Salah satu aplikasi yang banyak dimanfaatkan oleh para trader adalah TradingView. Aplikasi ini menawarkan berbagai fitur analisis teknikal yang lengkap, seperti grafik interaktif, indikator teknikal, dan berbagai alat bantu analisis lainnya. Popularitas yang tinggi terhadap aplikasi tersebut terlihat dari banyaknya unduhan pengguna serta ribuan tanggapan aplikasi pengguna di Google



Playstore, yang menggambarkan berbagai bentuk kepuasan, keluhan, serta pengalaman mereka dalam menggunakan layanan tersebut. Tanggapan-tanggapan tersebut memiliki nilai informasi yang signifikan untuk mengevaluasi kualitas aplikasi dan mengidentifikasi aspek-aspek yang perlu diperbaiki.

Namun demikian, tanggapan aplikasi pengguna di platform digital umumnya berbentuk teks bebas dan tidak terstruktur. Jumlahnya yang sangat besar membuat proses analisis secara manual menjadi tidak efektif, memakan banyak waktu yang menyebabkan ketidakefisienan, serta rentan terhadap subjektivitas analisis. Selain itu, volume data yang terus bertambah dari waktu ke waktu menjadikan proses pemantauan secara terus-menerus semakin sulit dilakukan tanpa dukungan teknologi otomatis. Dikarenakan kondisi yang disebabkan oleh permasalahan tersebut maka dibutuhkan pendekatan otomatis yang mampu mengolah teks berskala besar dengan konsistensi dan akurasi yang tinggi untuk mengatasi permasalahan tersebut, melalui teknik text mining dan machine learning yang mampu mengklasifikasikan sentimen pengguna secara cepat, tepat, dan objektif. Sentimen tersebut biasanya digolongkan ke dalam tiga kategori utama, yaitu tanggapan positif, negatif, dan netral. Informasi ini bisa dimanfaatkan dan digunakan sebagai pedoman oleh para developer aplikasi untuk mengupgrade kualitas layanan, memperbaiki fitur tertentu, dan memberikan pengalaman pengguna yang lebih baik. Dengan adanya kemampuan klasifikasi otomatis ini, proses pengambilan keputusan dapat dilakukan secara lebih terarah dan berbasis data nyata dari pengguna secara real time.

Dalam ranah analisis sentimen, "Support Vector Machine (SVM)" merupakan salah satu algoritma yang banyak di terapkan karena kemampuannya guna mengklasifikasikan data berdimensi tinggi serta performanya yang stabil pada berbagai jenis data. Berbagai penelitian telah menunjukkan efektivitas SVM dalam analisis sentimen. Misalnya, penelitian yang dilakukan oleh B. Maulana[1] menunjukkan bahwa SVM mampu menghasilkan akurasi sebesar 99,50% dan mengungguli metode lain dalam pengolahan data teks. Penelitian lainnya oleh A. Muhammadin[2] menghasilkan akurasi sebesar 83,3%, dan memperkuat temuan bahwa SVM cenderung lebih optimal dibandingkan metode alternatif seperti Naive Bayes. Prinsip kerja SVM yang mengandalkan pencarian hyperplane terbaik untuk memisahkan kelas membuat algoritma ini memiliki keunggulan pada data yang kompleks dan tidak linear[3]. Selain keunggulannya, SVM juga dikenal memiliki tingkat generalisasi yang baik ketika diterapkan pada data berukuran besar, sehingga menjadikannya salah satu metode pilihan dalam berbagai riset terkait klasifikasi teks.

Meskipun demikian, tantangan utama dalam analisis sentimen adalah masalah perbedaan proporsi kelas (imbalanced data). Kondisi ini sering ditemukan ketika jumlah ulasan dalam satu kategori sentimen jauh lebih besar dibandingkan kategori lainnya. Misalnya, ulasan positif biasanya lebih cenderung daripada ulasan negatif atau netral. Perbedaan proporsi seperti ini dapat menurunkan performa model, terutama pada kelas minoritas, sehingga menghasilkan nilai recall atau precision yang sangat rendah[4]. Temuan M. Ibnu Choldun[5] menunjukkan fenomena serupa ketika percobaan dilakukan menggunakan data yang tidak seimbang. Dalam eksperimen tanpa SMOTE, akurasi model mencapai 93,9%; namun metrik lainnya, seperti presisi, recall, dan F1-score bernilai nol pada kelas minoritas karena model hanya mengenali kelas mayoritas. Setelah penerapan SMOTE untuk menyeimbangkan jumlah data pada tiap kelas, akurasi memang menurun menjadi 90,0%, tetapi nilai presisi meningkat menjadi 15,5%, recall menjadi 14,5%, dan F1-score menjadi 15,0%. Hal ini menegaskan bahwa penyeimbangan data sangat penting agar model tidak bias terhadap kelas tertentu[6]. Kondisi ini menunjukkan bahwa akurasi yang tinggi bukanlah suatu yang dapat menjadi jaminan bahwa suatu model atau teknik yang di terapkan bisa bekerja secara adil, sehingga diperlukan pendekatan yang mampu meningkatkan kinerja pada seluruh kelas secara seimbang tanpa condong ke dalam kelas mayoritas.

Untuk mengatasi permasalahan tersebut, "Synthetic Minority Over-sampling Technique (SMOTE)" menjadi salah satu upaya yang banyak diterapkan. SMOTE bekerja dengan membuat sampel sintetis baru yang menyerupai data minoritas berdasarkan analisis terhadap tetangga terdekat ( $k$ -nearest neighbors). Dengan cara ini, pembagian data menjadi lebih adil karenanya model dapat mempelajari pola dari seluruh kelas secara lebih adil[7]. Penggunaan SMOTE tampak mengoptimalkan efektivitas model pada metrik recall dan F1-score, terutama pada kelas minoritas yang sebelumnya kurang terwakili. Selain itu, SMOTE juga membantu mengurangi risiko model tidak mampu melakukan generalisasi yang sering terjadi pada metode oversampling konvensional, karena data sintetis yang dihasilkan tidak berupa duplikasi, melainkan variasi baru yang tetap relevan dengan karakteristik data asli.

Berdasarkan latar belakang tersebut, studi ini difokuskan pada pengujian performa algoritma SVM dengan menggunakan kernel linear untuk mengklasifikasikan sentimen pada ulasan pengguna aplikasi TradingView. study ini juga mengaplikasikan SMOTE untuk menangani perbedaan proporsi data sehingga model dapat menghasilkan kinerja yang lebih representatif pada ketiga kelas sentimen, yaitu positif, negatif, dan netral[8]. Pemilihan SVM kernel linear didasarkan pada karakteristik data teks yang cenderung memiliki dimensi tinggi, sehingga kernel ini dianggap paling sesuai dan efisien untuk pemrosesan data dalam jumlah besar.

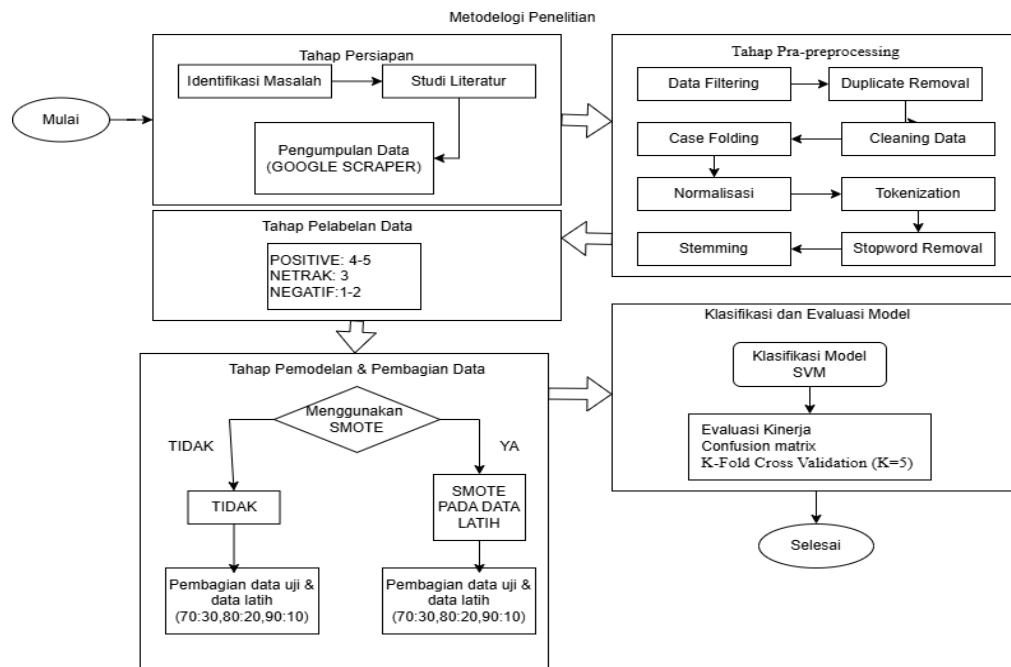
Studi ini diantisipasi dapat menghasilkan dampak dalam bentuk analisis komprehensif mengenai pengaruh penerapan SMOTE terhadap efektifitas teknik SVM dalam klasifikasi sentimen. Selain itu, hasil study ini diantisipasi dapat memfasilitasi para pengelola aplikasi TradingView dalam memahami persepsi pengguna secara lebih mendalam, sehingga mereka dapat meningkatkan kualitas layanan dan pengalaman pengguna secara keseluruhan. Temuan ini juga diantisipasi dapat menjadi sumbangan bagi penelitian selanjutnya khususnya dalam pengembangan model analisis sentimen yang lebih efektif dan adaptif pada data ulasan aplikasi mobile. Diharapkan juga studi ini berpotensi memberikan pengetahuan terhadap metodologis baru yang dapat dijadikan acuan dalam pengolahan data tidak seimbang serta mengoptimalkan akurasi metode pada berbagai konteks klasifikasi serupa di masa mendatang. Selain itu, penelitian ini diharapkan mampu memperkuat pemahaman mengenai hubungan antara kualitas data dan performa model, sehingga



meningkatkan adaptabilitas model terhadap variasi ekspresi linguistik pengguna. Penelitian ini juga diharapkan mampu memberikan pemahaman yang lebih mendalam dan komprehensif.

## 2. METODOLOGI PENELITIAN

Penelitian ini mengadopsi metode, yang melibatkan pengumpulan data secara sistematis kemudian dianalisis dengan pendekatan statistik atau komputasi[1]. Dalam studi ini, penggunaan SVM difokuskan pada evaluasi akurasi klasifikasi sentimen pengguna aplikasi TradingView dari platform Google Play Store, sedangkan SMOTE diterapkan untuk menyeimbangkan keseluruhan data pada setiap kelas sentimen. Alur study ini ditunjukkan pada Gambar 1.



Gambar 1. Metodologi

### 2.1. Identifikasi Masalah

Langkah pertama dalam penelitian adalah menganalisis permasalahan, yang kemudian dijadikan dasar untuk merancang tujuan, serta arah penelitian secara terstruktur. Permasalahan yang diangkat berkaitan dengan penerapan SVM dapat dalam klasifikasi sentimen pada tanggapan pengguna aplikasi TradingView di Google Playstore serta penanganan ketimpangan distribusi data antar kelas sentimen. Upaya tersebut dilakukan melalui penerapan SMOTE untuk mengoptimalkan performa klasifikasi model SVM[9].

### 2.2. Studi Literatur

Tahapan ini bertujuan untuk meninjau serta memperdalam pemahaman mengenai teori, metode, dan hasil riset terdahulu yang relevan. Hasil kajian ini menjadi dasar pijakan teoritis dan metodologis dalam merancang serta mengimplementasikan penelitian yang sedang dilakukan.

### 2.3. Data Collection

Pengumpulan data dilaksanakan dengan pendekatan otomatis melalui teknik web scraping. Teknik ini merupakan teknik untuk memperoleh informasi secara terstruktur dari sebuah situs web dengan memanfaatkan perangkat lunak khusus, data dikumpulkan melalui halaman tanggapan pengguna aplikasi TradingView yang terdapat pada Google Playstore. Proses ekstraksi ulasan dilakukan dengan memanfaatkan bahasa pemrograman Python, yang mendukung berbagai pustaka khusus untuk scraping sehingga memudahkan pengambilan data secara efisien[10].

Penggunaan Python dipilih karena bahasa ini memiliki ekosistem pustaka yang kaya, sintaks yang sederhana, serta dukungan luas untuk data science dan machine learning[11].

### 2.4. Preprocessing

Sebelum data ulasan dapat digunakan dalam proses klasifikasi sentimen, Untuk memastikan data dapat diolah dengan baik oleh algoritma machine learning, diperlukan tahap pra-pemrosesan (preprocessing) yang bertujuan untuk melakukan pembersihan serta penyesuaian data sehingga memiliki format dan struktur yang sesuai[10]. Prosedur ini menjadi upaya yang sangat penting karena kualitas data mentah sangat memengaruhi performa model, sehingga setiap tahapan seperti



normalisasi teks, penghapusan karakter yang tidak relevan, serta standarisasi bentuk kata harus dilakukan secara sistematis. Adapun tahapannya meliputi:

- a. Data Filtering: Data filtering dalam konteks tabel adalah proses menyaring informasi dari kumpulan data dengan tujuan mengekstraksi hanya baris atau kolom yang memenuhi kriteria tertentu[12].
- b. Duplicate Removal: Proses penghapusan data yang bersifat duplikat.
- c. Cleaning Data: Pembersihan kata yang memuat komponen yang tidak diperlukan untuk proses analisis, seperti simbol, angka, tanda baca, emoji, maupun tautan (URL)[13].
- d. Case Folding: Mengonversi seluruh karakter pada teks ulasan ke bentuk huruf kecil guna menyamakan representasi kata[14].
- e. Normalisasi: Merubah kata-kata yang salah eja maupun ditulis dalam bentuk singkatan disesuaikan ke bentuk bakunya[13], guna memastikan konsistensi makna dalam analisis teks dan akan disesuaikan kamuskatabaku yang ada di kaggle.
- f. Tokenization: Memisah teks ulasan menjadi potongan kata yang biasanya disebut token[13].
- g. Stop Words Removal: Menghilangkan kata umum yang ada di dalam bahasa Indonesia yang memiliki frekuensi kemunculan tinggi seperti kata sambung maupun kata yang mana kurang memiliki sokongan dalam sentimen analisis[9], menggunakan library nltk.corpus stopwords.
- h. Stemming : Merubah kata menjadi bentuk dasar atau kata dasar[15], menggunakan pustaka sastrawi.

## 2.5. Pelabelan Data

Setiap entri data ulasan dikategorikan ke dalam kelas tertentu berdasarkan kandungan sentimennya. Dalam penelitian ini, sentimen dikelompokkan menjadi tiga kategori, yaitu:

- a. Positif, untuk ulasan yang mengandung kepuasan, atau tanggapan yang membangun terhadap aplikasi.
- b. Negatif, untuk ulasan yang mengandung keluhan, kekecewaan, atau kritik terhadap performa dan fitur aplikasi.
- c. Netral, untuk ulasan yang tidak menunjukkan kecenderungan emosi positif maupun negatif secara dominan.

Pelabelan sentimen dalam penelitian ini merujuk pada nilai *rating* pengguna[16], di mana nilai 4 dan 5 dianggap sebagai indikator sentimen 'Positif', nilai 3 sebagai 'Netral', serta nilai 1 dan 2 menunjukkan sentimen 'Negatif'.

## 2.6. Pembobotan Kata TF-IDF

“Term Frequency-Inverse Document Frequency (TF-IDF)” diterapkan sebagai teknik memberikan bobot pada setiap kata yang berfungsi mengonversi informasi berbasis teks dalam bentuk representasi numerik, sehingga memungkinkan pemrosesan lebih lanjut menggunakan algoritma pembelajaran mesin[17]. Pendekatan TF-IDF dimanfaatkan sebagai pengukur tingkat kepentingan kata di dokumen, dengan membandingkannya pada semua dokumen lain dalam korpus[18].

## 2.7. Modeling Data dan Implementasi SMOTE

Membagi data ke dalam set pelatihan dan pengujian merupakan tahapan krusial untuk mengevaluasi performa model secara optimal[19]. Pada penelitian ini data dibagi ke dalam tiga skenario, yaitu (70:30, 80:20, 90:10) tanpa implementasi SMOTE dan yang menggunakan implementasi SMOTE.

Didalam algoritma SVM memiliki kelemahan dalam mengatasi ketidak seimbangan pada dataset[4]. Namun, untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode SMOTE sebagai teknik oversampling yang menghasilkan data sintetik pada kelas minoritas, sehingga dapat menyeimbangkan distribusi data dalam proses pelatihan model[18]. Dengan demikian, distribusi antar kelas dalam data latih menjadi lebih seimbang tanpa menghilangkan data asli.

Penting untuk dicatat bahwa proses oversampling dengan SMOTE hanya diterapkan pada data latih, bukan pada keseluruhan dataset. Hal ini dilakukan untuk menjaga validitas proses evaluasi[20], karena data uji sebaiknya mencerminkan distribusi alami dari data asli. Evaluasi pada data uji yang tidak diubah memungkinkan pengukuran performa model secara lebih realistis dan mendekati kondisi penerapan di dunia nyata.

## 2.8. Support Vector Machine

SVM adalah teknik pengelompokan yang ampuh untuk mengatasi data berdimensi tinggi serta pola yang bersifat non-linear dalam machine learning[3]. SVM mengelompokkan data dengan mencari hyperplanes optimal sebagai batas keputusan yang mengoptimalkan margin tiap kelas pada ruang fitur[9], rumusnya sebagai berikut:

$$f(x) = w^t x + b \quad (1)$$

Keterangan:

x : vektor fitur (hasil dari TF-IDF, misalnya)

w : vektor bobot (weight vector)

b : bias (intercept)

f(x): berperan sebagai fungsi keputusan yang memetakan data ke salah satu kelas. Nilai positif dari f(x) menunjukkan bahwa data termasuk ke dalam kelas +1, sedangkan nilai negatif mengindikasikan bahwa data berada pada kelas -1



## 2.9. Pengujian

Pengujian diterapkan guna mengevaluasi hasil teknik klasifikasi sentimen yang dilatih Dengan pendekatan SVM sebagai teknik klasifikasi utama. Setelah melawati tahap preprocessing, mengubah teks sebagai vektor dengan TF-IDF, pembagian data, dan penyeimbangan kelas menggunakan SMOTE, maka data akan siap digunakan untuk proses pelatihan dan pengujian. Adapun metrik evaluasi yang digunakan dalam pengujian antara lain:

### a. Confusion Matrix

Confusion matrix, digunakan untuk menilai hasil suatu teknik klasifikasi, memberikan gambaran apakah model tersebut berkinerja baik atau sebaliknya. Nilai TruePositive(TP), TrueNegative(TN) mencerminkan jumlah dugaan yang benar yang diidentifikasi oleh teknik yang digunakan, sementara FalsePositive(FP), False Negative(FN) menampilkan jumlah salah prediksi [21]. Selain itu, confusion matrix juga menjadi dasar dalam menentukan nilai akurasi, error rate, precision, dan recall untuk menilai tingkat keberhasilan klasifikasi suatu model[22]. Dengan demikian, keberadaan confusion matrix tidak hanya berguna sebagai instrumen untuk membantu dalam melihat distribusi prediksi, tetapi juga menjadi komponen penting dalam penghitungan metrik evaluasi yang berperan langsung dalam menilai efektivitas model klasifikasi. Berikut rumus untuk menghasilkan nilai sebagai berikut:

1. Accuracy Proporsi jumlah prediksi yang benar terhadap total data.

$$Accuracy = \frac{TN+TP}{FN+FP+TN+TP} \times 100 \quad (2)$$

2. Recall Mengukur seberapa lengkap model mengenali semua data dari suatu kelas.

$$rec = \frac{TP}{FN+TP} \times 100 \quad (3)$$

3. Precision Mengukur seberapa tepat model saat memprediksi suatu kelas.

$$pre = \frac{TP}{FP+TP} \times 100 \quad (4)$$

4. F1 - Score hasil Perataan harmonik antara nilai Precision/recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

### b. K-Fold Cross Validation

Selain evaluasi model terhadap data uji, penelitian ini juga menerapkan validasi silang menggunakan metode "Stratified K-Fold Cross Validation" yang menggunakan jumlah lipatan sebanyak (K=5). cross - validation adalah metode evaluasi dalam supervised learning, biasanya di terapkan untuk menaksir tingkat kesalahan pada data uji maupun data latih[21]. Tujuan dari validasi silang adalah untuk memastikan model tidak overfitting dan memiliki performa yang stabil terhadap berbagai subset data. Teknik Stratified K-Fold mengalokasikan sampel latih ke dalam K subset (fold) dengan tetap mempertahankan proporsi label di setiap fold[6]. Model kemudian dilatih dan diuji sebanyak K kali, di mana setiap fold bergantian menjadi data uji, sementara sisanya menjadi data latih. Evaluasi selama validasi silang dilakukan dengan menghitung Skor Akurasi pada Setiap Fold, Rata-rata Akurasi dan Standard Deviasi, validasi ini dilakukan hanya pada data latih untuk menghindari kebocoran data dan menjaga objektivitas evaluasi model.

## 3. HASIL DAN PEMBAHASAN

### 3.1. Hasil Pengumpulan Data

Proses pengumpulan sample, informasi diambil meliputi Review ID, Username, Rating, Teks Ulasan, dan Tanggal. Secara keseluruhan, jumlah data yang berhasil dikumpulkan sebanyak 10.000 ulasan, mencakup berbagai variasi penilaian dan isi ulasan yang dapat digunakan untuk keperluan klasifikasi sentimen. Hasil dari ulasan yang berhasil di scrapping tersaji pada Tabel 1.

**Tabel 1.** Hasil Ulasan

Review Id	Username	Rating	Review Text	Date
52e5686d	Agung	5	Ok	2025-06-02 19:13:08
bbf8808c	shela safitri	5	bagus lumayan buat pemula kaya saya 😊	2025-06-02 18:55:41
b3282fe1	Rosadi Didik	5	Bagus	2025-06-02 18:55:23
12478b0c	OmJhon 1357	5	bagus	2025-06-02 17:11:58
		5	bagus banget makasih trading view pengen bange...	2025-06-02 16:53:05

### 3.2. Hasil Preprocessing

Pada tahap ini Sebelum data ulasan dapat digunakan dalam proses klasifikasi sentimen, diperlukan tahap pra-pemrosesan untuk membersihkan dan menyiapkan data agar sesuai dengan kebutuhan algoritma machine learning. Tahapan pra-



pemrosesan dalam penelitian ini meliputi beberapa Langkah yaitu: data filtering, duplicate removal, cleaning data, case folding, normalisasi, tokenization, stop word removal, dan stemming, dibawah ini adalah proses Text Preprocessing yang di tulis secara berurutan:

- a. Data filtering, yaitu mengambil kolom yang akan digunakan dalam proses klasifikasi, daftar yang akan dipakai tersaji pada Tabel 2.

**Tabel 2.** Hasil Data Filtering

Username	Rating	Review Text	Date
Agung	5	Ok	2025-06-02 19:13:08
shela safitri	5	bagus lumayan buat pemula kaya saya 😊	2025-06-02 18:55:41
Rosadi Didik	5	Bagus	2025-06-02 18:55:23
OmJhon 1357	5	bagus	2025-06-02 17:11:58
	5	bagus banget makasih trading view pengen banget upgrade	2025-06-02 16:53:05

- b. Duplicate removal adalah penghapusan data yang bersifat duplikat, dari proses duplicate removal yang awalnya data berjumlah 10.000 ulasan menjadi 5.797 ulasan yang akan digunakan dalam proses klasifikasi.
- c. Cleaning Data: penghapusan komponen yang tidak berguna terhadap klasifikasi sentimen berupa simbol, angka, tanda baca, emoji, maupun tautan (URL). Hasil Cleaning tersaji pada Tabel 3.
- d. Case Folding: Langkah selanjutnya adalah mengonversi seluruh karakter pada teks ulasan ke bentuk huruf kecil guna menyamakan representasi kata. Hasil casefolding tersaji pada Tabel 4.

**Tabel 3.** Hasil Cleaning

Review Text	Cleaned review
Ok	Ok
bagus lumayan buat pemula kaya saya 😊	bagus lumayan buat pemula kaya saya
Bagus	Bagus
bagusbanget makasih trading view pengen banget upgrade sejauh ini terbaik	bagusbanget makasih trading view pengen banget upgrade sejauh ini terbaik

**Tabel 4.** Hasil Case Folding

Normalisasi	Case folding
Ok	ok
bagus lumayan buat pemula kaya saya	bagus lumayan buat pemula kaya saya
Bagus	bagus
bagusbanget makasih trading view pengen banget upgrade sejauh ini terbaik	bagusbanget makasih trading view pengen banget upgrade sejauh ini terbaik

- e. Normalisasi: Mencakup proses normalisasi, di mana kata-kata yang salah eja maupun ditulis dalam bentuk singkatan disesuaikan ke bentuk bakunya. Hasil Normalisasi tersaji pada Tabel 5.

**Tabel 5.** Hasil Normalisasi

Case folding	Normalisasi
ok	ok
bagus lumayan buat pemula kaya saya	bagus lumayan buat pemula kayak saya
bagus	bagus
bagusbanget makasih trading view pengen banget upgrade sejauh ini terbaik	bagusbanget terima akasih trading view pengen banget upgrade sejauh ini terbaik

- f. Tokenization: Memisah teks ulasan menjadi potongan kata yang biasanya disebut token. Hasil Tokenization tersaji pada Tabel 6.



**Tabel 6.** Hasil Tokenization

Normalisasi	Tokenization
ok	[ok]
bagus lumayan buat pemula kayak saya	[bagus, lumayan, buat, pemula, kayak, saya]
bagus	[bagus]
bagusbanget terima kasih trading view	[bagus, banget, terima, kasih, trading, view, pingin, banget, upgrade]
sejauh ini terbaik.	[sejauh, ini, terbaik]

g. Stop Words Removal: Menghilangkan kata umum yang ada di dalam bahasa Indonesia yang memiliki frekuensi kemunculan tinggi seperti kata sambung maupun kata depan yang kurang memiliki kontribusi signifikan dalam analisis sentimen. Hasil stopword removal tersaji pada Tabel 7.

**Tabel 7.** Hasil Stopword Removal

Tokenization	Stopword removal
[ok]	[ok]
[bagus, lumayan, buat, pemula, kayak, saya]	[bagus, lumayan, pemula, kayak, saya]
[bagus]	[bagus]
[bagus,banget, terima, kasih, trading, view, pingin, banget, upgrade]	[bagus,banget, terima, kasih, trading, view, pingin, banget, upgrade]
[sejauh, ini, terbaik]	[terbaik]

h. Stemming : kata akan diubah menjadi bentuk dasar atau kata dasar. Hasil Stemming tersaji pada Tabel 8.

**Tabel 8.** Hasil Stemming

Stopword removal	Stemming Data
[ok]	ok
[bagus, lumayan, pemula, kayak, saya]	bagus lumayan mula kayak saya
[bagus]	bagus
[bagus,banget, terima, kasih, trading, view, pingin, banget, upgrade]	bagusbanget terima kasih trading view pingin banget upgrade
[terbaik]	baik

### 3.3. Pelabelan Data

Dalam study ini merujuk pada nilai *rating* pengguna, di mana nilai 4 dan 5 dianggap sebagai indikator sentimen 'Positif', nilai 3 sebagai 'Netral', serta nilai 1 dan 2 menunjukkan sentimen 'Negatif'. Contoh data dapat dilihat dalam Tabel 9.

**Tabel 9.** Hasil Pelabelan

Stemming Data	Rating	Label
bagus banget terima kasih trading view pingin banget upgrade	5	POSITIF
pakai alarm indikator	4	POSITIF
indikator gratis batas	3	NETRAL
fitur ya pakai versi gratis versi langgan depan lihat chart indikator	2	NEGATIF
ampas login broker susah banget kayak diecb	1	NEGATIF

### 3.4. Pembobotan Kata TF-IDF

Dalam penelitian ini, digunakan “*library TfidfVectorizer*” dari modul “*sklearn.feature\_extraction.text*” untuk mengubah data teks didalam kolom hasil pra-pemrosesan beralihke representasi bilangan .Jumlah attribute dibatasi hingga 5000. berarti hanya 5000 kata dengan nilai TF-IDF tertinggi yang akan dipertimbangkan dalam proses transformasi[10]. Sementara itu, kolom *sentiment* dijadikan sebagai variabel target dalam pemodelan.

Hasil rata-rata bobot TF-IDF menunjukkan adanya kata-kata dominan yang sering digunakan pengguna saat memberikan ulasan pada aplikasi. Hasil pembobotan TF-IDF tersaji pada Tabel 10.

**Tabel 10.** Sepuluh Kata dengan Rata-rata TF-IDF Tertinggi

Kata	bantu	bagus	aplikasi	mantap	trading	good	mudah	ya	keren
TF-IDF	0,062588	0,055884	0,039716	0,34041	0,027027	0,024911	0,022032	0,021014	0,020427

### 3.5. Modeling Data dan Implementasi SMOTE

Pada penelitian ini, data dibagi ke dalam 3 skenario pengalokasian: 70:30 , 80:20 , 90:10.



Namun, berdasarkan pengamatan terhadap distribusi label awal, diketahui bahwa data memiliki ketidakseimbangan kelas yang cukup signifikan. Dari total 5.797 data ulasan, mayoritas berasal dari kelas "Positif" dengan 4.979 data, sementara kelas "Negatif" dan "Netral" masing-masing hanya berjumlah 638 dan 180 data. Distribusi ini dapat mempengaruhi performa model dalam mengenali kelas minoritas.

Oleh karena itu, untuk mengatasi permasalahan ketidak seimbangan kelas (class imbalance), SMOTE diterapkan dalam penelitian ini sebagai pendekatan untuk menyeimbangkan distribusi kelas pada data pelatihan. Pendekatan ini bertujuan mengurangi kecenderungan model untuk memihak pada kelas mayoritas serta meningkatkan kemampuannya dalam mengidentifikasi semua kelas secara seimbang[19]. Sehingga terdapat 6 skenario pembagian yaitu : 70:30, 80:20, 90:10. tanpa implementasi SMOTE dan yang menggunakan implementasi smote. jumlah penyeimbangan data melalui metode SMOTE dapat di lihat di Tabel 11.

**Tabel 11.** Modeling Data dan Implementasi Smote

Data	Asli	70:30		80:20		90:10				
		SVM no Smote		SVM + Smote		SVM no Smote		SVM + Smote		
		training	testing	training	testing	training	testing	training	testing	
Positif	4979	3485	1494	3485	3983	996	3983	4481	498	4481
Negatif	638	446	192	3485	510	128	3983	574	64	4481
Netral	180	126	54	3485	144	36	3983	162	18	4481
Jumlah	5797									

### 3.6. Klasifikasi dan Evaluasi model

Sesudah sample ulasan beralihke representasi representasi bilangan memakai TF-IDF dan diseimbangkan dengan metode SMOTE, tahap berikutnya adalah pelatihan dan pengujian model klasifikasi. Dalam study ini, menggunakan algoritma SVM dengan kernel linear[13], yang terbukti efektif untuk menyelesaikan permasalahan klasifikasi pada data teks yang memiliki dimensi fitur tinggi. Strategi ini digunakan karena mampu memberikan hasil pemisahan kelas yang stabil dan konsisten meskipun struktur data bersifat kompleks. Lebih lanjut, penggunaan kernel linear memungkinkan proses pembelajaran lebih efisien dan tetap mempertahankan kinerja teknik secara optimal pada dataset berukuran besar.

#### 3.6.1. Evaluasi Model

Pemeriksaan kinerja model dengan memanfaatkan sejumlah matrix yang lazim digunakan dalam permasalahan klasifikasi, seperti akurasi , precision , recall , dan f1-score juga, "confusion matrix" turut digunakan untuk menggambarkan distribusi hasil prediksi terhadap label kelas sebenarnya, sehingga mempermudah analisis kesalahan klasifikasi.

Guna menguji kinerja model SVM dalam klasifikasi sentimen melalui tiga proporsi pembagian data pelatihan dan pengujian (70:30, 80:20, 90:10), yang masing-masing dievaluasi dengan metrik akurasi, precision, recall, dan f1-score. Untuk memperoleh hasil yang andal dan generalizable, evaluasi juga dilengkapi dengan confusion matrix dan validasi silang menggunakan metode K-Fold Cross Validation.

##### a. Rasio 70:30

Pada pembagian data 70:30 , model SVM tanpa SMOTE memperoleh akurasi sebesar 89,94%. Meskipun akurasi terlihat tinggi, distribusi recall antar kelas menunjukkan ketidakseimbangan: kelas Positif mencapai recall 98%, sedangkan kelas Negatif hanya 52% dan kelas Netral gagal dikenali 0%. Hal ini menandakan model dominan cenderung terhadap mayoritas dan kurang sensitif terhadap minoritas.

Penerapan SMOTE pada rasio yang sama menurunkan akurasi keseluruhan menjadi 85,46%, namun recall untuk kelas minoritas meningkat: Negatif 61% dan Netral 7%. Recall kelas Positif tetap tinggi di 91%. Ini menunjukkan SMOTE berhasil memperbaiki representasi kelas minoritas sehingga model mulai mampu mendeteksi pola dari data yang sebelumnya jarang muncul.

Hasil K-Fold Cross Validation model SVM tanpa SMOTE memiliki akurasi rata-rata 89,20% ( $\pm 0,54\%$ ), sedangkan model SVM dengan SMOTE mencapai akurasi 95,52% ( $\pm 0,41\%$ ). Confusion matrix memperlihatkan bahwa penerapan SMOTE mampu mengurangi kesalahan klasifikasi pada kelas minoritas secara signifikan. Misalnya, jumlah false negative pada kelas kedua turun dari 36 menjadi 11, dan false positive dari 90 menjadi 9. Meskipun terjadi sedikit peningkatan pada false negative kelas ketiga, distribusi prediksi menjadi lebih seimbang, sehingga model lebih mampu mengenali semua kelas secara akurat dan meningkatkan kemampuan generalisasi.

##### b. Rasio 80:20

Pada pembagian 80:20, pola serupa terlihat. Model Non-SMOTE memperoleh akurasi 89,31%, dengan recall kelas Positif 98%, Negatif 48%, dan Netral 0%. Model tetap bias terhadap kelas mayoritas, sedangkan kelas minoritas kurang terdeteksi.

Model SVM dengan SMOTE pada rasio ini memperoleh akurasi sedikit lebih rendah (85,69%), tetapi recall kelas minoritas meningkat: Negatif 64% dan Netral 8%. Recall kelas Positif tetap tinggi di 91%. Hal ini menegaskan bahwa SMOTE efektif meningkatkan kemampuan deteksi kelas minoritas tanpa mengorbankan terlalu banyak performa pada kelas mayoritas.



Hasil K-Fold Cross Validation tanpa SMOTE mencapai akurasi rata-rata 89,89% ( $\pm 0,35\%$ ), sedangkan model SMOTE meningkat menjadi 95,18% ( $\pm 0,18\%$ ). Confusion matrix menunjukkan bahwa kelas minoritas mengalami pengurangan kesalahan prediksi yang signifikan: false negative kelas kedua turun dari 45 menjadi 2, dan false positive dari 99 menjadi 89. Sementara kelas mayoritas tetap diprediksi dengan baik, hal ini menegaskan bahwa SMOTE efektif dalam menyeimbangkan distribusi kelas dan meningkatkan keandalan model.

c. Rasio 90:10

Dengan rasio 90:10 Non-SMOTE tetap menunjukkan bias terhadap kelas mayoritas: akurasi 89,14%, recall Positif 97%, Negatif 52%, dan Netral 0%. Kinerja model pada kelas minoritas tetap rendah.

Model SMOTE menurunkan akurasi menjadi 85,52%, namun meningkatkan recall Negatif menjadi 64% dan Netral hingga 28%. Recall kelas Positif 90%. Hal ini menunjukkan bahwa SMOTE semakin efektif untuk rasio yang lebih tinggi dalam mendeteksi kelas minoritas yang jumlahnya relatif kecil.

Hasil K-Fold Cross Validation model tanpa SMOTE memiliki akurasi 89,88% ( $\pm 0,42\%$ ), sedangkan model SMOTE meningkat menjadi 94,59% ( $\pm 0,61\%$ ). Confusion matrix menunjukkan bahwa penerapan SMOTE mengurangi kesalahan prediksi pada kelas minoritas: false negative kelas kedua tetap tinggi 51 ke 64 tetapi false positive menurun, sedangkan kelas mayoritas tetap memiliki prediksi yang baik. Hal ini menunjukkan bahwa meskipun data mayoritas semakin dominan, SMOTE membantu model mempertahankan kemampuan mengenali kelas minoritas secara lebih seimbang, sehingga performa keseluruhan lebih stabil dan dapat digeneralisasikan.

Rekapitulasi hasil evaluasi model pada setiap rasio data dirangkum dalam Tabel 12 untuk memperjelas perbedaan kinerja antara model SVM tanpa SMOTE dan model SVM dengan SMOTE.

**Tabel 12.** Hasil Evaluasi Model SVM Berdasarkan Rasio Data

Rasio	Model	Akurasi	Precision Neg	Precision Net	Precision Pos	Recall Neg	Recall Net	Recall Pos	FI Neg	FI Ne	FI Pos
70:30	SVM-SMOTE	89,94%	69%	0%	92%	52%	0%	98%	59%	0%	95%
	SVM+SMOTE	85,46%	57%	4%	95%	61%	7%	91%	59%	5%	93%
80:20	SVM-SMOTE	89,31%	65%	0%	92%	48%	0%	98%	55%	0%	95%
	SVM+SMOTE	85,69%	54%	6%	95%	64%	8%	91%	59%	7%	93%
90:10	SVM-SMOTE	89,14%	63%	0%	92%	52%	0%	97%	57%	0%	94%
	SVM+SMOTE	85,52%	53%	15%	96%	64%	28%	90%	58%	20%	93%

Selain pengujian berdasarkan rasio data, performa model juga divalidasi menggunakan metode K-Fold Cross Validation. Ringkasan hasil pengujian tersebut ditampilkan pada Tabel 13.

**Tabel 13.** K-Fold Cross Validation (K=5)

Rasio data	Model	Akurasi	Deviasi $\pm$	Confusion Matrix Hasil K-Fold
70:30	SVM no SMOTE	89,20%	0,54%	[[192, 2, 252],[36, 0, 90],[58, 0, 3427]]
	SVM+SMOTE	95,52%	0,41%	[[3383,58, 44],[11,3465,9],[169,177,3139]]
80:20	SVM no SMOTE	89,89%	0,35%	[[248,1,261],[,45,0,99],[62,1,3920]]
	SVM+SMOTE	95,18%	0,18%	[[3850,85,48],[2,3892,89],[196,156,3631]]
90:10	SVM no SMOTE	89,88%	0,42%	[[285,0,289],[51,0,111],[77,0,4404]]
	SVM+SMOTE	94,59%	0,61%	[[4345,76,60],[64,4300,117],[216,194,4071]]

**3.6.2. Pembahasan**

Dari Hasil evaluasi model pada berbagai skema proporsi data (70:30, 80:20, 90:10) serta validasi silang K-Fold memberikan gambaran yang cukup jelas mengenai dampak ketidakseimbangan data terhadap performa klasifikasi sentimen. Secara umum, model SVM tanpa penerapan SMOTE cenderung bias terhadap kelas mayoritas Positif. Hal ini terlihat dari tingginya nilai recall pada kelas Positif di atas 97%, namun dengan kegagalan total dalam mengenali kelas Netral recall = 0%. Kondisi ini menunjukkan bahwa model lebih banyak “bermain aman” dengan memprediksi kelas yang paling dominan dalam data, sehingga performa akurasi terlihat tinggi tetapi tidak adil bagi kelas minoritas.

Penerapan SMOTE menunjukkan perubahan yang signifikan, khususnya pada kemampuan model mendeteksi kelas minoritas. Walaupun akurasi keseluruhan mengalami sedikit penurunan pada skema hold-out misalnya turun dari 89,94% menjadi 85,46% pada rasio 70:30, recall pada kelas Negatif dan Netral meningkat secara konsisten. Artinya, SMOTE berhasil menambah variasi sampel sintesis sehingga distribusi data menjadi adil. Demikian, Model dapat mengkaji lebih akurat dalam memahami pola minoritas, meskipun harus “mengorbankan” sebagian akurasi global.

Analisis lebih lanjut melalui K-Fold Cross Validation menunjukkan bahwa implementasi SMOTE secara konsisten menguatkan performa SVM, terutama dalam hal generalisasi. Model SVM dengan SMOTE mencapai akurasi rata-rata lebih dari 95% pada semua proporsi data, dengan deviasi yang relatif kecil ( $\pm 0,18\%$   $\pm 0,61\%$ ), menunjukkan kestabilan performa antar fold. Hal ini mengindikasikan bahwa ketika oversampling dilakukan hanya pada data training tiap fold, model tidak sekadar menghafal data sintesis, melainkan mampu membangun representasi yang lebih seimbang untuk setiap kelas. Confusion matrix K-Fold menegaskan perbaikan pada kelas minoritas. Pada rasio 70:30, jumlah false negative dan false positive pada kelas minoritas menurun drastis dibanding model tanpa SMOTE. Tren serupa tampak pada proporsi 80:20 dan 90:10, di mana SMOTE membantu model mengenali kelas minoritas lebih baik tanpa menurunkan akurasi kelas mayoritas secara signifikan. Dengan kata lain, SMOTE tidak hanya menguatkan akurasi



keseluruhan, tetapi juga memperbaiki kemampuan model dalam mendeteksi sampel yang sebelumnya sulit diklasifikasikan, terutama opini negatif atau netral pada analisis sentimen.

#### 4. KESIMPULAN

Berdasarkan hasil studi, dapat dirumuskan bahwa model SVM tanpa SMOTE meskipun memiliki akurasi tinggi (sekitar 89–90%) tetap menunjukkan bias terhadap kelas mayoritas, dengan recall kelas Positif sangat tinggi (97–98%) dan kelas Netral hampir tidak terdeteksi, sementara penerapan SMOTE sedikit menurunkan akurasi hold-out menjadi sekitar 85–86% tetapi secara signifikan meningkatkan kemampuan model dalam mendeteksi kelas minoritas, terutama kelas Negatif dan Netral. Validasi silang K-Fold memperlihatkan keunggulan SMOTE secara konsisten, di mana SVM+SMOTE mencapai rata-rata akurasi 94–95% dengan deviasi rendah ( $\pm 0,18\%$ – $0,61\%$ ), menunjukkan kestabilan dan kemampuan generalisasi yang lebih baik, serta distribusi prediksi yang lebih seimbang antar kelas. Temuan ini menegaskan bahwa penggunaan SMOTE efektif dalam memperbaiki sensitivitas terhadap kelas minoritas tanpa menurunkan performa kelas mayoritas secara signifikan, sehingga kombinasi SVM dan SMOTE layak diterapkan pada dataset ulasan yang tidak sepadan. Untuk penelitian selanjutnya, disarankan eksplorasi metode representasi fitur berbasis semantic embedding, seperti Word2Vec atau BERT, dan penggunaan model non-linier untuk meningkatkan kemampuan mendeteksi kelas Netral secara lebih akurat.

#### REFERENCES

- [1] 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 Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 375–384, Feb. 2024, doi: 10.57152/malcom.v4i2.1206.
- [2] A. Muhammadin and I. A. Sobari, "Analisis Sentimen Pada Ulasan Aplikasi Kredito Dengan Algoritma SVM Dan NBC," *Reputasi: Jurnal Rekayasa Perangkat Lunak*, vol. 2, no. 2, pp. 85–91, 2021, doi: 10.31294/reputasi.v2i2.785.
- [3] H. Eldo, Ayuliana, D. Suryadi, G. Chrisnawati, and L. Judijanto, "Penggunaan Algoritma Support Vector Machine (SVM) Untuk Deteksi Penipuan pada Transaksi Online," *J. Minfo Polgan*, vol. 13, no. 2, pp. 1627–1632, Oct. 2024, doi: 10.33395/jmp.v13i2.14186.
- [4] A. A. Nurrahman, M. Mauladi, and A. Rahman, "Analisis Sentimen Masyarakat terhadap Kenaikan Harga Bahan Bakar Minyak Menggunakan Support Vector Machine dan SMOTE," *sudo J. Tek. Inform.*, vol. 4, no. 2, pp. 50–56, 2025, doi: 10.56211/sudo.v4i2.908.
- [5] Muhammad Ibnu Choldun Rachmatullah, Sari Armia, and Mubassiran, "Menerapkan Smote pada Klasifikasi Data Penyakit Stroke," *Improv. Ilm. Manaj. Tek. Inform.*, vol. 17, no. 1, pp. 9–12, 2025, [Online]. Available: <https://ejurnal.ulbi.ac.id/index.php/improve/article/view/4307>
- [6] M. I. Putri and I. Kharisudin, "Analisis Sentimen Pengguna Aplikasi Marketplace Tokopedia Pada Situs Google Play Menggunakan Metode Support Vector Machine (SVM), Naive Bayes, dan Logistic Regression," *Prism. Pros. Semin. Nas. Mat.*, vol. 5, pp. 759–766, 2022, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/>
- [7] E. Eskiyaturrofikoh and R. R. Suryono, "Analisis Sentimen Aplikasi X pada Google Play Store Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM)," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 9, no. 3, pp. 1408–1419, Aug. 2024, doi: 10.29100/jupi.v9i3.5392.
- [8] A. Muliandi, Y. Chrisnanto, and H. Ashaury, "Optimalisasi Klasifikasi Support Vector Machine dengan SMOTE: Studi Kasus Ulasan Pengguna Aplikasi Alfagift," *J. Pekommas*, vol. 9, no. 2, pp. 249–258, Dec. 2024, doi: 10.56873/jpkm.v9i2.5583.
- [9] V. Alessandro Riyanto and D. Budi Santoso, "Penerapan Model Support Vector Machine Pada Klasifikasi Sentimen Ulasan Aplikasi Lazada," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 9, no. 1, pp. 178–184, 2024, doi: 10.30645/jurasik.v9i1.725.
- [10] T. Sugihartono, R. Rian, and C. Putra, "Penerapan Metode Support Vector Machine dalam Classifikasi Ulasan Pengguna Aplikasi Mobile JKN," *SKANIKASistem Komput. dan Tek. Inform.*, vol. 7, no. 2, pp. 144–153, Jul. 2024, doi: 10.36080/skanika.v7i2.3193.
- [11] A. Chapagain, *Hands-on Web Scraping with Python: Extract Quality Data from the Web Using Effective Python Techniques*. Birmingham, UK: Packt Publishing, 2023.
- [12] G. Radiana and A. Nugroho, "Analisis sentimen berbasis aspek pada ulasan aplikasi KAI Access menggunakan metode Support Vector Machine," *J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, pp. 1–10, 2023, doi: 10.37792/jukanti.v6i1.836.
- [13] A. Mudya Yolanda and R. Tri Mulya, "Implementasi Metode Support Vector Machine untuk Analisis Sentimen pada Ulasan Aplikasi Sayurbox di Google Play Store," *VARIANSI J. Stat. Its Appl. Teach. Res.*, vol. 6, no. 2, pp. 76–83, 2024, doi: 10.35580/variansiunm258.
- [14] I. S. K. Idris, Y. A. Mustofa, and Irvan, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *JJEEJambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 32–35, Jan. 2023, doi: 10.37905/jjee.v5i1.16830.
- [15] V. Fitriyana, L. Hakim, D. C. R. Novitasari, and A. H. Asyhar, "Analisis Sentimen Ulasan Aplikasi Jamsostek Mobile Menggunakan Metode Support Vector Machine," *J. Buana Inform.*, vol. 14, no. 1, pp. 40–49, 2023, doi: 10.24002/jbi.v14i01.6909.
- [16] S. Delimasari and K. Kusrini, "Komparasi Algoritma Machine Learning Untuk Menganalisis Sentimen Ulasan Pada Aplikasi Digital Korlantas Polri," *G-Tech J. Teknol. Terap.*, vol. 8, no. 4, pp. 2411–2419, Oct. 2024, doi: 10.70609/gtech.v8i4.5089.
- [17] Tinaliah and T. Elizabeth, "Analisis Sentimen Ulasan Aplikasi PrimaKu Menggunakan Metode Support Vector Machine," *J. Tek. Inform. dan Sist. Inf.*, vol. 9, no. 4, pp. 3436–3442, 2022, doi: 10.35957/jatise.v9i4.3586.
- [18] R. Ridwan, E. H. Hermaliani, and M. Ernawati, "Penerapan: Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada Klasifikasi Ujuran Kebencian," *Comput. Sci.*, vol. 4, no. 1, pp. 80–88, 2024, doi: 10.31294/coscience.v4i1.2990.
- [19] S. P. Azzahra, Y. A. Apriyanto, and A. Wijaya, "Analisis Sentimen Ulasan Aplikasi DeepL Pada Google Play Dengan Metode Support Vector Machine (SVM)," *JUSINJurnal Sist. Inf.*, vol. 4, no. 2, pp. 59–66, Dec. 2023, doi: 10.32546/jusin.v4i2.2368.



- [20] E. M. O. N. Haryanto, A. K. A. Estetikha, and R. A. Setiawan, "Implementasi SMOTE Untuk Mengatasi Imbalanced Data Pada Sentimen Analisis Sentimen Hotel di Nusa Tenggara Barat Dengan Menggunakan Algoritma SVM," *J. Inf. Interaktif*, vol. 7, no. 1, pp. 16–20, 2022.
- [21] F. R. Lumbaraja, E. C. L. Gaol, D. A. Shofiana, and A. Junaidi, "Implementasi SMOTE dan Support Vector Machine Pada Klasifikasi Data Tidak Seimbang Metilasi Arginin," *J. Pepadun*, vol. 5, no. 1, pp. 27–37, 2024, doi: 10.23960/pepadun.v5i1.209.
- [22] S. OKFALISA, LESTARI HANDAYANI1, DINDA JUWITA P., MUHAMMAD AFFANDES, S. S .M. FAUZI, "Coronary Heart Disease Using Support Vector Machine," *J. Eng. Sci. Technol.*, vol. 16, no. 2, pp. 1370–1385, 2021.