

# Perbandingan Algoritma SVM, Random Forest, dan Naive Bayes Terhadap Kasus Scam di Media Sosial Twitter

Rizky Herdian Saputra, Ryan Randy Suryono\*

<sup>1</sup>Fakultas Teknik dan Ilmu Komputer, Sistem Informasi, Universitas Teknokrat Indonesia, Bandar Lampung, Indonesia

<sup>2</sup>Fakultas Teknik dan Ilmu Komputer, Magister Ilmu Komputer, Universitas Teknokrat Indonesia, Bandar Lampung, Indonesia

Email: <sup>1</sup>rizky\_herdian\_saputra@teknokrat.ac.id, <sup>2</sup>\*ryan@teknokrat.ac.id

Email Penulis Korespondensi : ryan@teknokrat.ac.id

Submitted: 29/04/2025; Accepted: 01/09/2025; Published: 02/09/2025

**Abstrak**—Pertumbuhan teknologi informasi dan komunikasi yang cepat memberikan pengaruh besar terhadap tingkat kejahatan di dunia maya. Internet yang sebelumnya digunakan untuk mempercepat proses pertukaran informasi, juga disalahgunakan oleh pihak-pihak tertentu yang tidak bertanggung jawab. Sebagian dari bentuk kejahatan yang marak terjadi adalah *scam*, yaitu penipuan yang dilakukan dengan tujuan untuk memperoleh keuntungan secara tidak sah dengan mengeksploitasi korban melalui berbagai modus. Tujuan dari penelitian ini adalah untuk mengevaluasi dan menilai perbandingan kinerja tiga algoritma, yaitu, *Support Vector Machine* (SVM), *Random Forest*, dan *Naive Bayes* dalam menganalisis persepsi publik tentang kasus scam di media sosial Twitter. Dataset terdiri dari 9.132 *tweet*, yang kemudian melalui tahapan *preprocessing* seperti *cleaning*, *case folding* dan *normalisasi kata*, sehingga menyisakan 8.879 *tweet* untuk dianalisis. Selanjutnya, diterapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE), dengan pembagian dataset dilakukan dengan 80% untuk *training* dan 20% untuk *testing*. Hasil memperlihatkan sebelum penerapan SMOTE, algoritma SVM mempunyai tingkat akurasi tertinggi sebesar 82%, di ikuti dengan *Random Forest* sebesar 79%, dan *Naive Bayes* sebesar 74%. Setelah penerapan SMOTE, akurasi meningkat signifikan dengan SVM sebesar 88%, *Random Forest* 84%, dan *Naive Bayes* 76%. Hal ini menunjukkan bahwa dalam analisis sentimen pada kasus *Scam*, metode SVM memperoleh nilai akurasi lebih besar dari *Random Forest*, dan *Naive Bayes*.

**Kata Kunci:** Analisis Sentimen; Scam; Sosial Media; Naive Bayes; Random Forest; SMOTE; Support Vector Machine (SVM)

**Abstract**—The rapid growth of information and communication technology has a significant impact on the level of cybercrime. The internet, which was originally used to expedite the exchange of information, is also misused by irresponsible parties. One of the prevalent forms of crime is scams, which are fraudulent activities aimed at gaining unlawful profits by exploiting victims through various tactics. The purpose of this research is to evaluate and compare the performance of three algorithms: Support Vector Machine (SVM), Random Forest, and Naive Bayes in analyzing public sentiment regarding scam cases on social media Twitter. The dataset consists of 9,132 tweets, which undergo preprocessing stages such as cleaning, case folding, and word normalization, leaving 8,879 tweets for analysis. Then, the Synthetic Minority Over-sampling Technique (SMOTE) is applied, with the dataset divided into 80% for training and 20% for testing. The results show that before applying SMOTE, the SVM algorithm achieved the highest accuracy at 82%, followed by Random Forest at 79%, and Naive Bayes at 74%. After applying SMOTE, accuracy significantly increased, with SVM reaching 88%, Random Forest at 84%, and Naive Bayes at 76%. This demonstrates that in sentiment analysis of scam cases, the SVM method achieves higher accuracy than both Random Forest and Naive Bayes.

**Keywords:** Sentiment Analysis; Scam; Social Media; Naive Bayes; Random Forest; SMOTE; Support Vector Machine (SVM)

## 1. PENDAHULUAN

Dalam era digital media sosial sudah menjadi salah satu platform utama untuk komunikasi yang terus berkembang, interaksi sosial, berbagi informasi, serta berbisnis. Dengan kemudahan yang ditawarkan, media sosial memungkinkan penggunaannya untuk berinteraksi secara cepat dan efektif dengan orang lain di seluruh dunia [1]. Namun, seiring dengan pesatnya perkembangan media sosial dalam kehidupan sehari-hari, platform ini juga menghadirkan berbagai tantangan baru terkait dengan keamanan dan privasi data pribadi. Salah satu tantangan terbesar adalah maraknya kasus *scam* atau penipuan yang dapat merugikan penggunaannya secara finansial dan emosional. *Scam* di media sosial sering kali memanfaatkan berbagai metode manipulasi psikologis untuk menipu pengguna agar mengambil tindakan yang merugikan. Misalnya, penipuan melalui phishing, di mana pelaku mencoba mengelabui pengguna untuk mengungkapkan informasi pribadi atau login ke akun yang terancam dicuri. Ada juga penipuan yang menawarkan produk atau layanan palsu, yang berakhir dengan kerugian finansial bagi korban. Selain itu, *scam* dapat berupa tautan berbahaya yang mengarahkan pengguna ke situs web berisi *malware* yang dapat mencuri data sensitif atau merusak perangkat mereka. Fenomena ini semakin diperburuk dengan pesatnya penyebaran informasi yang tidak terkontrol dan kemampuan algoritma untuk menyebarkan pesan-pesan tersebut dengan cepat kepada banyak orang [2].

Twitter menjadi salah satu media sosial populer yang dimanfaatkan masyarakat untuk mengungkapkan opini berhubungan dengan berbagai isu, termasuk penipuan online atau *scam*. Beragam pandangan yang muncul, mulai dari kekhawatiran hingga solusi yang diusulkan, mencerminkan dinamika persepsi publik terhadap ancaman *scam* di media sosial. Dengan kemampuannya untuk menyebarkan informasi dengan cepat, Twitter menjadi sumber data yang sangat relevan untuk menganalisis opini publik mengenai masalah ini [3]. Analisis sentimen terhadap komentar masyarakat di platform ini dapat memberikan gambaran yang lebih lengkap mengenai bagaimana publik menilai fenomena *scam*, terutama terkait dengan transparansi, keadilan, dan dampak penipuan terhadap pengguna media sosial. Untuk mendapatkan hasil yang akurat dari analisis sentimen publik, penting untuk menggunakan metodologi yang tepat *Support Vector Machine* (SVM), *Random Forest* dan, *Naive Bayes* adalah tiga metode yang biasa dipakai dalam

analisis sentiment. Penelitian ini membandingkan efisiensi dari masing-masing algoritma dengan menggunakan ketiga metode tersebut.

Sejumlah penelitian telah dilakukan untuk membandingkan performa algoritma klasifikasi seperti *Naive Bayes*, *Support Vector Machine (SVM)*, dan *Random Forest* pada berbagai kasus. Adi dan Rani (2024) membandingkan ketiga algoritma dalam klasifikasi Halo BCA di Goggle Play Store *Random Forest* memberikan akurasi optimal sebesar 91,28%, presisi 0,91%, recall sebesar 0,91%, dan F1 -score sebesar 0,91%. *Support Vector Machine* berada di posisi kedua dengan akurasi sebesar 87,55%, sedangkan *Naive Bayes* memiliki akurasi sebesar 81,73 %. [4]. Penelitian oleh Nalatissifa dkk. (2021) menunjukkan bahwa *Random Forest* memberikan hasil yang unggul dalam memprediksi ketidakhadiran di tempat kerja, dengan akurasi mencapai akurasi 99,38%, presisi 99,42%, dan recall 99,39%, berdasarkan dataset *Absenteeism at Work* dari UCI Machine Learning Repository [5].

Sitorus dkk. (2021) membandingkan SVM dan *Random Forest* pada klasifikasi data untuk deteksi malware di Android, dengan hasil bahwa *Random Forest* unggul secara signifikan dengan akurasi 98,99% serta nilai presisi, recall, dan F1-score untuk masing-masing dengan nilai 99%, sedangkan SVM mencatat akurasi 96,23% dan skor evaluasi 97% [6]. Dalam analisis sentimen terhadap kebijakan kenaikan harga BBM Indonesia pada 2022, Samantri dan Afiyati (2024) menemukan bahwa SVM maupun *Random Forest* menunjukkan hasil yang cukup baik, walaupun SVM menunjukkan kinerja yang lebih baik dengan tingkat keakuratan 77% dibandingkan *Random Forest* sebesar 76% [7]. Cahyani dan Abdillah meneliti performa ketiga algoritma dalam analisis sentimen pengguna media sosial X, dan hasilnya membuktikan bahwa SVM memiliki akurasi terbaik sebesar 87%, lalu *Random Forest* sebesar 85%, dan *Naive Bayes* sebesar 73% [8].

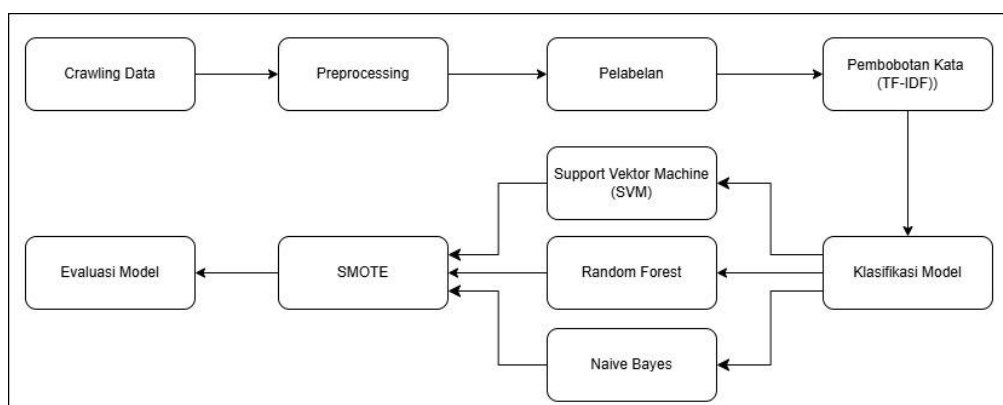
Berbagai studi sebelumnya telah berkontribusi pada pengembangan metode analisis sentimen, khususnya melalui penerapan algoritma seperti *Support Vector Machine*, *Random Forest*, serta *Naive Bayes*. [9]. Namun demikian, sebagian besar penelitian tersebut masih terbatas pada penggunaan satu jenis algoritma atau pendekatan *preprocessing* tertentu, tanpa mempertimbangkan teknik tambahan seperti *stemming* dan SMOTE (*Synthetic Minority Over-sampling Technique*) yang berpotensi mengoptimalkan performa model klasifikasi [10]. Selain itu, kajian yang secara spesifik menyoroti persepsi masyarakat terhadap isu *scam*, terutama di platform media sosial X, masih sangat terbatas sehingga membuka peluang untuk penelitian lanjutan.

Analisis sentimen terhadap isu *scam* memiliki tantangan tersendiri karena karakteristik data media sosial yang bersifat dinamis, tidak terstruktur, serta mengandung variasi bahasa yang kompleks. Oleh karena itu, tujuan dari penelitian ini adalah untuk mengevaluasi dan menganalisis perbandingan kinerja algoritma SVM, *Random Forest*, dan *Naive Bayes* dalam mengklasifikasikan sentimen terkait isu *scam* di media sosial Twitter. Studi ini tidak hanya berfokus pada identifikasi algoritma dengan akurasi terbaik, tetapi juga bertujuan memberikan rekomendasi praktis bagi industri dan pemangku kebijakan dalam merespons fenomena *scam* secara lebih adaptif. Selain itu, diharapkan penelitian ini dapat memperkaya literatur ilmiah mengenai penggunaan *machine learning* pada konteks analisis sentimen.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini dilakukan secara sistematis untuk menganalisis sentimen publik terhadap isu *scam* di media sosial Twitter. Data dikumpulkan melalui teknik *crawling* dengan kata kunci tertentu yang relevan, menghasilkan 9.132 *tweet*. Data kemudian diproses melalui tahapan *preprocessing* yaitu *cleansing*, *case folding*, *tokenizing*, *stopword removal* *normalization*, dan *stemming* sebelum dilakukan pelabelan sentimen dikategorikan menjadi tiga kelompok, yaitu positif, negatif, dan netral. Untuk mengatasi ketidakseimbangan kelas, digunakan teknik SMOTE, dilanjutkan dengan representasi data menggunakan TF-IDF dan dibagi menjadi data training dan data testing. Tiga algoritma klasifikasi diterapkan, yaitu SVM, *Random Forest*, dan *Naive Bayes* dengan evaluasi kinerja menggunakan *akurasi*, *presisi*, *recall* dan *F1-score*. Seluruh alur penelitian divisualisasikan pada Gambar 1.



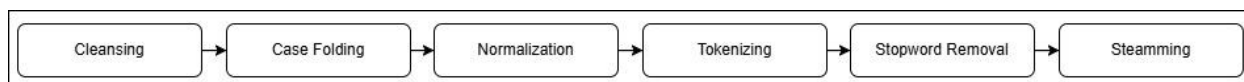
Gambar 1. Tahapan Penelitian

## 2.2 Pengumpulan Data

Penelitian ini melibatkan tahap pengumpulan data dari platform media sosial Twitter melalui metode crawling yang dilakukan secara otomatis menggunakan *Application Programming Interface* (API) [11]. Proses ekstraksi data dilakukan dengan bantuan *library Harvest* dengan pemanfaatan bahasa pemrograman Python yang dioperasikan dalam Google *Colaboratory* untuk mendukung efisiensi dan fleksibilitas selama proses pengambilan data [12].

## 2.3 Preprocessing Data

Teknik *preprocessing* berfungsi sebagai proses awal dalam pemrosesan data berbasis teks yang dimaksudkan untuk meningkatkan kualitas data sebelum dianalisis lebih lanjut [13]. Tahap ini dilakukan agar model analisis sentimen dapat bekerja lebih efektif dengan data yang lebih bersih dan terstruktur. Proses preprocessing data ditampilkan pada Gambar 2.



**Gambar 2.** Alur Preprocessing

Pada Gambar 2 menunjukkan tahapan preprocessing data yang terdiri dari cleansing, case folding, normalization, tokenizing, stopword removal, dan stemming. Penjelasan dari masing-masing tahapan adalah sebagai berikut.

### 2.3.1 Cleansing

*Cleansing* merupakan tahap awal dalam proses *preprocessing* yang bertujuan untuk menghilangkan komponen dalam teks yang dianggap tidak memiliki nilai signifikan, seperti tanda baca, angka, simbol, emoji, hastag, dan tautan [14].

### 2.3.2 Case Folding

*Case folding* adalah langkah dalam *preprocessing* teks yang berfungsi untuk menyeragamkan bentuk huruf dengan cara mengubah seluruh karakter alfabet dalam dokumen menjadi huruf kecil (*lowercase*) [15]. Tujuannya adalah untuk menghindari perbedaan interpretasi antara kata yang sama namun ditulis dengan kapitalisasi berbeda, sehingga meningkatkan konsistensi data dalam analisis teks.

### 2.3.3 Normalization

*Normalization* adalah proses standarisasi bentuk kata dalam teks agar sesuai dengan kaidah bahasa yang baku. Tahapan ini biasanya digunakan untuk menyamakan kata tidak baku, singkatan, atau variasi penulisan menjadi bentuk yang benar [16]. Contohnya, kata seperti “gk”, “ga”, atau “nggak” akan dinormalisasi menjadi “tidak”. Tujuan *normalization* adalah agar analisis teks menjadi lebih konsisten dan akurat.

### 2.3.4 Tokenizing

*Tokenizing* adalah proses memecah teks menjadi unit-unit kecil yang disebut *token* yang umumnya berupa kata. Tahapan ini bertujuan untuk mempermudah analisis teks dengan mengubah kalimat atau paragraf menjadi deretan kata yang terpisah. *Tokenizing* menjadi langkah awal penting dalam pemrosesan bahasa alami sebelum dilakukan analisis lebih lanjut [17].

### 2.3.5 Stopword Removal

*Stopword Removal* adalah proses menghapus kata-kata umum yang tidak memiliki dampak yang signifikan terhadap hasil analisis, seperti "dan", "yang", "di", serta "ke". Proses ini membantu mengurangi elemen-elemen yang tidak penting dalam data yang membantu analisis lebih fokus pada kata-kata yang memiliki makna penting [18].

### 2.3.6 Stemming

*Stemming* adalah teknik yang digunakan untuk mengembalikan kata turunan ke bentuk kata asal. Tahapan ini dilakukan untuk menyederhanakan berbagai variasi kata agar dianalisis sebagai satu entitas yang sama [19]. Misalnya, kata "penipuan", "menipu", dan "ditipu" semuanya akan dikembalikan ke bentuk dasar "tipu". Tujuan *stemming* adalah untuk mengurangi kompleksitas teks dan memperkuat konsistensi data dalam analisis.

## 2.4 Pelabelan Data

Setelah melalui tahap *pre-processing*, jumlah *tweet* yang dapat digunakan untuk analisis berkurang dari 9.132 menjadi 8.879 *tweet*. Langkah berikutnya adalah pelabelan sentimen, yang dimaksudkan untuk mengelompokkan setiap *tweet* ke dalam kelompok sentimen yang telah ditentukan dengan metode berbasis kamus. Setiap *tweet* diberi label sebagai sentimen positif, negatif, atau netral dengan menerapkan metode *lexicon-based*, yaitu teknik yang secara otomatis mencocokkan kata-kata dalam teks dengan daftar kosakata sentimen, dengan referensi dari kamus *lexicon* positif dan negatif yang diperoleh melalui *GitHub*. Metode *lexicon-based* bekerja dengan mengidentifikasi frasa, ekspresi, atau kata-kata yang sering muncul dalam percakapan, dialog, unggahan media sosial, ulasan, serta berbagai bentuk teks

lainnya. Dengan mencocokkan teks dalam *tweet* dengan daftar kata positif dan negatif, metode ini menentukan polaritas sentimen secara sistematis [20]. Penilaian terhadap polaritas sentimen proses ini dilakukan dengan mengukur perbedaan antara banyaknya kata positif dan negatif di setiap *tweet*. Jika skor hasil perhitungan lebih besar dari nol, *tweet* diklasifikasikan sebagai positif. Sebaliknya, apabila skor lebih kecil dari nol, *tweet* dikategorikan sebagai negatif. Sementara itu, skor bernilai nol menunjukkan sentimen netral

## 2.5 Pembobotan Kata TF-IDF

Teknik *Term Frequency–Inverse Document Frequency* (TF-IDF) diterapkan sebagai metode untuk memberi nilai pada setiap kata pada teks sesuai dengan tingkat kepentingannya [21]. TF-IDF berperan untuk menemukan kata-kata yang paling signifikan dalam menggambarkan sentimen, dengan mengombinasikan seberapa sering suatu kata terdapat pada satu *tweet* (TF) dan seberapa jarang kata tersebut ditemukan di *tweet* (IDF). Kata-kata yang muncul dengan konsisten dalam *tweet* tertentu, tetapi jarang muncul di *tweet* lainnya, akan mengandung bobot yang lebih besar dan dalam menentukan polaritas sentimen [22]. Dengan demikian, TF-IDF meningkatkan akurasi model klasifikasi sentimen dengan menekankan kata-kata yang bermakna serta mengurangi efek kata-kata umum yang tidak signifikan. Berikut merupakan formula yang digunakan untuk menghitung nilai TF-IDF (*Term Frequency–Inverse Document Frequency*). Formula TF-IDF ditampilkan pada persamaan 1 berikut.

$$\text{TF-IDF}(t,d)=\left(\frac{\text{Jumlah Kemunculan kata } t \text{ dalam dokumen } d}{\text{Jumlah total kata dalam dokumen } d}\right) \times \log\left(\frac{\text{Jumlah Total Dokumen}}{\text{Jumlah Dokumen yang Mengandung Kata } t}\right) \quad (1)$$

## 2.6 Klasifikasi Model

Dalam penelitian kali ini, pengembangan dilakukan melalui tiga algoritma klasifikasi model utama, yakni algoritma *Support Vector Machine (SVM)*, *Random Forest*, dan *Naive Bayes*. Model dilatih melalui memanfaatkan 80% dari total data sebagai data latih, sedangkan 20% sisanya dipakai sebagai data uji. Penjelasan mengenai masing-masing algoritma yang dimanfaatkan dalam penelitian ini dijelaskan sebagai berikut:

### 2.6.1 Support Vektor Machine (SVM)

*Support Vector Machine (SVM)* merupakan satu metode algoritma dalam teknik pembelajaran mesin yang digunakan untuk menyelesaikan permasalahan klasifikasi maupun *regresi*. Algoritma ini beroperasi melalui cara menemukan hyperplane atau batas pemisah optimal yang mampu memisahkan data berdasarkan kelasnya yang berbeda dengan margin terbesar. Titik data yang paling mendekati Hyperplane ditentukan oleh support vectors. dan sangat memengaruhi penentuan posisi hyperplane. [23]. Salah satu keunggulan SVM adalah kemampuannya mengolah data berdimensi tinggi serta data yang tidak terpisah secara linear dengan memanfaatkan metode kernel. Rumus perhitungannya adalah sebagai berikut pada persamaan 2

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

Vektor  $x$  merupakan representasi fitur dari data input, yang biasanya diperoleh melalui proses ekstraksi. Sementara itu,  $b$  adalah vektor bobot yang ditentukan selama proses pelatihan, dan  $b$  merupakan nilai bias yang berfungsi untuk menggeser posisi hyperplane.

### 2.6.2 Random Forest

*Random Forest* merupakan model *machine learning* berbasis *ensemble* yang digunakan dalam pemecahan masalah klasifikasi maupun *regresi*. Model ini menghasilkan keputusan akhir dengan mengkonstruksi banyak pohon keputusan pada data pelatihan, kemudian menyatukan hasil prediksi dari pohon-pohon tersebut guna meningkatkan akurasi dan kestabilan [24]. Pada klasifikasi, *Random Forest* akan mengambil suara terbanyak (*majority voting*) dari hasil pohon-pohon tersebut untuk menentukan kelas suatu data. Rumus perhitungannya adalah sebagai berikut pada persamaan 3.

$$f(x) = \text{Average}(f_1(x), f_2(x), \dots, f_n(x)) \quad (3)$$

Dalam persamaan tersebut,  $f(x)$  merupakan prediksi akhir dari model *Random Forest* untuk suatu input data  $x$  sedangkan  $(f_1(x), f_2(x), \dots, f_n(x))$  merupakan prediksi yang dihasilkan oleh setiap pohon keputusan dari  $n$  pohon yang telah dilatih. Dengan merata-ratakan seluruh hasil prediksi tersebut, *Random Forest* dapat menghasilkan estimasi yang lebih akurat dan tahan terhadap noise atau data ekstrem.

### 2.6.3 Naive Bayes

*Naive Bayes* adalah algoritma klasifikasi berbasis probabilistik yang didasarkan pada *Teorema Bayes* dengan asumsi independensi antar fitur. Artinya, setiap fitur atau atribut dalam data dianggap saling tidak bergantung satu sama lain terhadap kelas target [25]. Meskipun asumsi ini sering kali tidak sepenuhnya akurat di dunia nyata, pendekatan ini terbukti sangat efektif dalam berbagai aplikasi, terutama dalam klasifikasi teks seperti email spam detection, analisis sentimen, dan pengkategorian dokumen. Algoritma ini bekerja dengan menghitung probabilitas dari setiap kelas yang mungkin diberikan nilai-nilai fitur yang diamati, kemudian memilih kelas dengan probabilitas terbesar sebagai hasil akhir prediksi [26]. Rumus perhitungannya adalah sebagai berikut pada persamaan 4.

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \tag{4}$$

$P(H|X)$  adalah kemungkinan posterior atau peluang data  $X$  berada pada kelas  $H$  setelah melihat bukti.  $P(X|H)$  adalah *likelihood* atau kemungkinan fitur  $X$  muncul jika berasal dari kelas  $H$ ,  $P(H)$  adalah prior atau peluang awal kelas tersebut sebelum melihat data, dan  $P(X)$  adalah evidence atau total probabilitas dari data  $X$ . Karena  $P(X)$  sama untuk semua kelas, fokus utama klasifikasi ada pada hasil perkalian  $P(X|H) \cdot P(H)$ , dan kelas dengan nilai terbesar dipilih sebagai hasil prediksi.

### 2.7 SMOTE

SMOTE (*Synthetic Minority Over-sampling Technique*) berperan penting untuk mengatasi masalah keseimbangan kelas yang tidak merata, yaitu kondisi ketika jumlah data dalam salah satu kategori sentimen seperti positif, negatif, atau netral lebih sedikit dibandingkan kelas lainnya. Ketidakseimbangan ini bisa mengakibatkan bias dalam pelatihan model, di mana algoritma menjadi lebih efektif dalam mengenali kelas mayoritas dan kurang responsif terhadap kelas minoritas [27]. Teknik ini menciptakan sampel baru melalui proses *interpolasi*, yaitu dengan menghitung nilai-nilai baru di antara titik data minoritas dan tetangga terdekatnya dalam ruang fitur. Dengan pendekatan ini, SMOTE mampu menambah variasi data yang relevan dan tetap mencerminkan karakteristik asli kelas *minoritas*, sehingga memperkaya distribusi data tanpa mengorbankan keaslian pola yang ada.

### 2.8 Evaluasi Model

Evaluasi model merupakan langkah untuk mengukur seberapa baik kinerja model dalam memprediksi data baru dengan mengukur keakuratannya. Dalam klasifikasi, evaluasi dilakukan memanfaatkan metrik seperti *akurasi*, *presisi*, *recall*, serta *F1-score*. Tujuan utamanya adalah proses untuk menjamin bahwa mode mampu mengenali pola secara efektif dan dapat digunakan secara andal pada data yang tidak pernah diproses sebelumnya [28]. Rumus perhitungannya adalah sebagai berikut disajikan pada persamaan 5, 6, 7, dan 8.

$$Accuracy = \frac{TP+PN}{TP+TN+FP+FN} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi+Recall} \tag{8}$$

*True Positive* (TP) adalah total data yang sesungguhnya berada dalam kelas positif yang diprediksi secara tepat oleh model. *False Positive* (FP) menggambarkan kesalahan model ketika memprediksi data sebagai positif, padahal sebenarnya data tersebut negatif. *True Negative* (TN) mengacu pada jumlah data yang sebenarnya berasal dari kelas negatif yang diprediksi dengan benar oleh model. Di sisi lain, *False Negative* (FN) menggambarkan kesalahan di mana model memprediksi data sebagai negatif, meskipun kenyataannya data tersebut merupakan bagian dari kelas positif.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengumpulan Data

Proses evaluasi model dalam penelitian ini dimulai dengan pengumpulan data dari platform media sosial Twitter menggunakan kata kunci “*scam*,” “*penipuan*,” dan “*investasi bodong*” untuk mendapatkan *tweet* yang relevan. Sebanyak 9.132 *tweet* berhasil dikumpulkan dengan menggunakan bahasa pemrograman *Python*, memanfaatkan library *Tweet Harvest* untuk ekstraksi data dan *pandas* untuk pengolahan dataset. Dataset yang digunakan sebagai sampel penelitian ditampilkan pada Tabel 1.

**Tabel 1.** Hasil Pengumpulan Data *Tweet*

No	<i>Tweet</i>
1	@Tan_Mar3M kl beneran terjadi spt itu itu artinya SCAM dan perekonomian negara bisa langsung terhenti alias BANGKRUT. dimana letak hebatnya ? <a href="https://t.co/pf5fGr5Zau">https://t.co/pf5fGr5Zau</a>
9132	@SobatHAPE Akibat sms spam gk dianggap trus sekarang ganti jadi penipuan berkedok giveaway.

Dari Tabel 1 diperoleh gambaran bahwa tweet-tweet yang dikumpulkan tidak hanya memuat kata kunci seperti “*scam*” atau “*penipuan*,” tetapi juga mengandung opini yang beragam. Variasi bahasa dan struktur tweet ini menjadi dasar penting dalam proses analisis sentimen, yang selanjutnya akan diproses melalui tahapan pemrosesan data, pelabelan, dan klasifikasi.

### 3.2 Preprocessing Data

Setelah memperoleh data yang valid, proses berikutnya adalah tahap *preprocessing* atau pemrosesan data, yang sangat penting dalam mengelola dan menganalisis data berbasis teks, khususnya untuk mendukung analisis sentimen terkait topik *scam*. Proses ini mencakup beberapa tahapan penting seperti *cleansing*, *case folding*, *normalisasi kata*, *tokenizing*, *stopword removal*, dan *stemming*. Teknik-teknik tersebut digunakan untuk membersihkan, mengatur, dan mengkonsolidasikan data, termasuk memperbaiki nilai yang hilang atau salah serta mengurangi *noise* yang dapat memengaruhi akurasi model. Tujuan utama *preprocessing* adalah meningkatkan kualitas data sebelum masuk ke proses klasifikasi sentimen, sehingga hasil analisis menjadi lebih akurat dan relevan. Setelah dilakukan serangkaian tahap pembersihan dan penyaringan data, jumlah data yang siap digunakan untuk analisis sentimen terhadap berkurang menjadi 8879 *tweet*.

#### 3.2.1 Cleansing

Tahap *cleansing* dalam analisis sentimen dilakukan untuk membersihkan data teks dari komponen-komponen yang tidak berpengaruh secara signifikan terhadap hasil analisis. Hasil dari proses ini dapat dilihat pada Tabel 2.

Tabel 2. Hasil *Cleansing*

<i>Tweet</i>	<i>Cleansing</i>
@Tan_Mar3M kl beneran terjadi spt itu itu artinya SCAM dan perekonomian negara bisa langsung terhenti alias BANGKRUT. dimana letak hebatnya ? <a href="https://t.co/pf5fGr5Zau">https://t.co/pf5fGr5Zau</a>	kl beneran terjadi spt itu itu artinya SCAM dan perekonomian negara bisa langsung terhenti alias BANGKRUT dimana letak hebatnya
@SobatHAPE Akibat sms spam gk dianggap trus sekarang ganti jadi penipuan berkedok giveaway.	Akibat sms spam gk dianggap trus sekarang ganti jadi penipuan berkedok giveaway

Tabel 2 menyajikan hasil dari proses *cleansing*. Pada *tweet* pertama, elemen-elemen seperti *mention*, tautan *URL*, dan tanda baca dihapus, namun makna utama *tweet* tetap terjaga. Begitu juga pada *tweet* kedua, *mention* dihilangkan tanpa mengurangi informasi penting mengenai modus penipuan yang disampaikan.

#### 3.2.2 Case folding

*Case folding* merupakan langkah untuk mengubah seluruh huruf dalam teks menjadi huruf kecil untuk menghilangkan perbedaan penulisan akibat kapitalisasi. Langkah ini bertujuan menyederhanakan analisis teks serta meningkatkan keakuratan dalam menganalisis sentimen menggunakan kata-kata yang sebenarnya identik meskipun ditulis dalam format huruf yang berbeda. Hasil dari proses ini Dapat dilihat pada Tabel 3.

Tabel 3. Hasil *Case Folding*

<i>Cleansing</i>	<i>Case Folding</i>
kl beneran terjadi spt itu itu artinya SCAM dan perekonomian negara bisa langsung terhenti alias BANGKRUT dimana letak hebatnya	kl beneran terjadi spt itu itu artinya scam dan perekonomian negara bisa langsung terhenti alias bangkrut dimana letak hebatnya
Akibat sms spam gk dianggap trus sekarang ganti jadi penipuan berkedok giveaway	akibat sms spam gk dianggap trus sekarang ganti jadi penipuan berkedok giveaway

Tabel 3 menunjukkan hasil dari proses *case folding*. Pada *tweet* pertama, kata "SCAM" dan "BANGKRUT" yang awalnya menggunakan huruf kapital diubah menjadi "scam" dan "bangkrut" dalam huruf kecil. Begitu juga pada *tweet* kedua, tidak ada perubahan signifikan selain pengubahan huruf kapital menjadi huruf kecil di seluruh kata.

#### 3.2.3 Normalization

*Normalization* merupakan tahapan penting untuk mengubah teks tidak terstruktur dan bersifat informal menjadi bentuk standar. Hal ini sangat penting saat memproses data dari media sosial, yang umumnya mengandung ejaan tidak baku, singkatan, dan gaya bahasa yang beragam. Hasil dari tahapan *normalization* dapat dilihat pada Tabel 4.

Tabel 4. Hasil *Normalization*

<i>Case Folding</i>	<i>Normalization</i>
kl beneran terjadi spt itu itu artinya scam dan perekonomian negara bisa langsung terhenti alias bangkrut dimana letak hebatnya	kalau benaran terjadi seperti itu itu artinya scam dan perekonomian negara bisa langsung terhenti alias bangkrut dimana letak hebatnya
akibat sms spam gk dianggap trus sekarang ganti jadi penipuan berkedok giveaway	akibat sms spam tidak dianggap terus sekarang ganti jadi penipuan berkedok giveaway

Tabel 4 menunjukkan hasil dari tahap *normalization*. Pada *tweet* pertama, frasa "kl beneran" diubah menjadi "kalau benaran," dan kata "spt" menjadi "seperti." Begitu juga pada *tweet* kedua, singkatan "gk" diubah menjadi "tidak," dan "trus" menjadi "terus".

### 3.2.4 Tokenizing

*Tokenizing* merupakan proses yang bertujuan memecah teks menjadi unit-unit yang lebih kecil, seperti kata atau frasa. Langkah ini memfasilitasi analisis dan pemrosesan data teks dengan memberikan struktur yang dapat diinterpretasikan oleh sistem dengan membagi teks menjadi token. Hasil dari proses tokenisasi disajikan dalam Tabel 5.

**Tabel 5.** Hasil *Tokenizing*

<i>Normalization</i>	<i>Tokenizing</i>
kalau benaran terjadi seperti itu itu artinya scam dan perekonomian negara bisa langsung terhenti alias bangkrut dimana letak hebatnya akibat sms spam tidak dianggap terus sekarang ganti jadi penipuan berkedok giveaway	['kalau', 'benaran', 'terjadi', 'seperti', 'itu', 'itu', 'artinya', 'scam', 'dan', 'perekonomian', 'negara', 'bisa', 'langsung', 'terhenti', 'alias', 'bangkrut', 'dimana', 'letak', 'hebatnya'] ['akibat', 'sms', 'spam', 'tidak', 'dianggap', 'terus', 'sekarang', 'ganti', 'jadi', 'penipuan', 'berkedok', 'giveaway']

Tabel 5 menyajikan hasil dari proses tokenisasi. Contohnya, kalimat "kalau benaran terjadi seperti itu itu artinya scam..." dipecah menjadi daftar kata seperti ['kalau', 'benaran', 'terjadi', 'seperti', 'itu', ...]. Demikian juga pada tweet kedua, teks dibagi menjadi token-token individual seperti ['akibat', 'sms', 'spam', 'tidak', 'dianggap', ...].

### 3.2.5 Stopword Removal

*Stopword removal* adalah proses untuk menghapus kata-kata umum yang tidak memberikan kontribusi signifikan terhadap pemahaman konteks dalam analisis teks, seperti "dan", "yang", atau "di". Tahapan ini bertujuan untuk memfokuskan pemrosesan hanya pada kata-kata yang bermakna dan relevan, sehingga meningkatkan efisiensi analisis dan mengurangi beban komputasi. Hasil dari tahapan *stopword removal* dapat dilihat pada Tabel 6.

**Tabel 6.** Hasil *Stopword Removal*

<i>Tokenizing</i>	<i>Stopword Removal</i>
['kalau', 'benaran', 'terjadi', 'seperti', 'itu', 'itu', 'artinya', 'scam', 'dan', 'perekonomian', 'negara', 'bisa', 'langsung', 'terhenti', 'alias', 'bangkrut', 'dimana', 'letak', 'hebatnya'] ['akibat', 'sms', 'spam', 'tidak', 'dianggap', 'terus', 'sekarang', 'ganti', 'jadi', 'penipuan', 'berkedok', 'giveaway']	['benaran', 'scam', 'perekonomian', 'negara', 'langsung', 'terhenti', 'alias', 'bangkrut', 'dimana', 'letak', 'hebatnya'] ['akibat', 'sms', 'spam', 'dianggap', 'ganti', 'penipuan', 'berkedok', 'giveaway']

Tabel 6 menunjukkan hasil *stopword removal* dimana, kata-kata umum seperti "kalau," "itu," dan "dan" dihapus, menyisakan kata-kata penting seperti "scam," "perekonomian," dan "bangkrut."

### 3.2.6 Steaming

*Stemming* dalam analisis sentimen merupakan proses untuk mengganti kata-kata yang memiliki makna serupa menjadi bentuk dasar kata. Teknik ini memungkinkan model untuk mengenali pola dengan lebih cepat, meningkatkan efisiensi dalam analisis, serta menghasilkan evaluasi sentimen yang lebih akurat. Dengan mengurangi kata ke bentuk dasarnya, sistem dapat lebih fokus pada elemen-elemen penting dalam teks. Hasil *stemming* disajikan pada Tabel 7.

**Tabel 7.** Hasil *Stemming*

<i>Stopword Removal</i>	<i>Stemming</i>
['benaran', 'scam', 'perekonomian', 'negara', 'langsung', 'terhenti', 'alias', 'bangkrut', 'dimana', 'letak', 'hebatnya'] ['akibat', 'sms', 'spam', 'dianggap', 'ganti', 'penipuan', 'berkedok', 'giveaway']	benar scam ekonomi negara langsung henti alias bangkrut mana letak hebat akibat sms spam anggap ganti tipu kedok giveaway

Tabel 7 menunjukkan hasil *stemming*. Pada contoh pertama, kata seperti "benaran," "perekonomian," dan "terhenti" diubah menjadi bentuk dasar "benar," "ekonomi," dan "henti" untuk menyederhanakan teks dan memperjelas makna.

## 3.3 Pelabelan Data

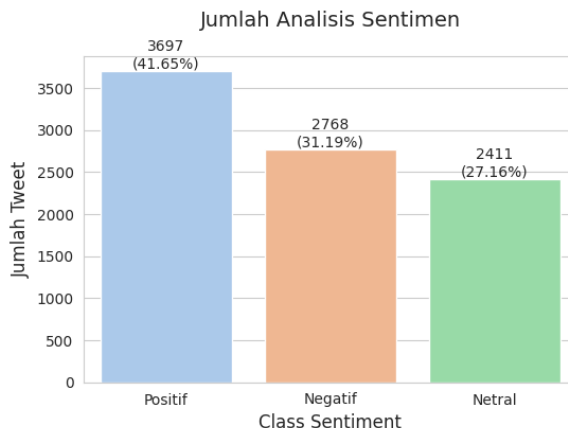
Pelabelan sentimen dilakukan dengan pendekatan berbasis leksikon, di mana sistem mengevaluasi kata-kata dalam komentar dengan mencocokkannya terhadap kamus sentimen InSet. Kamus ini telah dikelompokkan menjadi tiga kategori polaritas: positif, negatif, dan netral. Hasil pelabelan sentimen ditampilkan pada Tabel 8.

**Tabel 8.** Pelabelan Data

<i>Username</i>	<i>Stemming</i>	Score	Label
ranikunty9966	benar scam ekonomi negara langsung henti alias bangkrut mana letak hebat	-1	Negatif
micinchikiii	scam coba coba ya kali bantu saldo order suruh top up saldo embel tarik saldo	5	Positif

Username	Stemming	Score	Label
pholover0	kali scam info google next time percaya	0	Netral

Hasil pelabelan sentimen menunjukkan bahwa 3697 data tergolong positif, 2768 negatif, dan 2411 netral. Klasifikasi ini berdasarkan nilai polaritas, di mana skor di atas 0 dianggap positif, di bawah 0 negatif, dan 0 sebagai netral. Visualisasi distribusi sentimen disajikan pada Gambar 3 untuk memberikan gambaran yang lebih jelas.



Gambar 3. Visualisasi Hasil Pelabelan Data

Berdasarkan hasil analisis sentimen yang ditampilkan dalam Gambar 3, diketahui bahwa mayoritas *tweet* pengguna menunjukkan sentimen positif, dengan jumlah sebesar 41,65% dari total data yang dianalisis. Sentimen negatif berada di posisi kedua dengan 31,19%, sedangkan sentimen netral sebanyak 27,16%. Persentase ini menunjukkan bahwa tanggapan warganet terhadap topik yang dibahas didominasi oleh sentimen positif, meskipun masih terdapat proporsi signifikan dari sentimen negatif dan netral.

### 3.4 Visualisasi WordCloud

Visualisasi *WordCloud* dimanfaatkan untuk memperlihatkan kata-kata yang memiliki frekuensi kemunculan tertinggi dalam komentar Twitter terkait kasus *scam*. Pada tahap awal, sebelum dilakukan preprocessing, *WordCloud* masih dipenuhi oleh kata-kata yang bersifat umum dan tidak mengandung makna yang spesifik. Namun, setelah dilakukan tahapan penghapusan *stopwords*, proses *stemming*, serta *normalisasi*, hasil visualisasi menjadi lebih terfokus pada kata-kata yang merepresentasikan opini publik secara lebih akurat. Gambar 4 menyajikan hasil akhir dari tahapan ini.



Gambar 4. Visualisasi Wordcloud

Dari Gambar 4 kata-kata yang paling sering muncul adalah "modus tipu", "investasi bodong", dan "scam." Kata-kata ini menunjukkan fokus utama mengenai berbagai jenis penipuan yang banyak dibahas di media sosial Twitter.

### 3.5 Feature Extraction

Dalam tahap *Feature Extraction*, Dalam penelitian ini, pendekatan TF-IDF (*Term Frequency-Inverse Document Frequency*) untuk mengkonversi komentar terkait kasus *scam* menjadi bentuk numerik yang dapat dianalisis oleh algoritma *machine learning*. TF-IDF menilai seberapa relevan suatu kata dalam sebuah komentar berdasarkan kemunculannya di dokumen tersebut serta di seluruh korpus data. Kata-kata umum yang tidak memberikan banyak informasi akan diberi bobot lebih rendah, sementara kata-kata yang jarang muncul namun penting seperti istilah terkait

penipuan akan memiliki bobot yang lebih tinggi. Proses ini menghasilkan sebuah *sparse matrix*, di mana setiap kolom merepresentasikan kata unik dan setiap baris mewakili satu komentar. Tabel 9 menyajikan hasil dari TF-IDF.

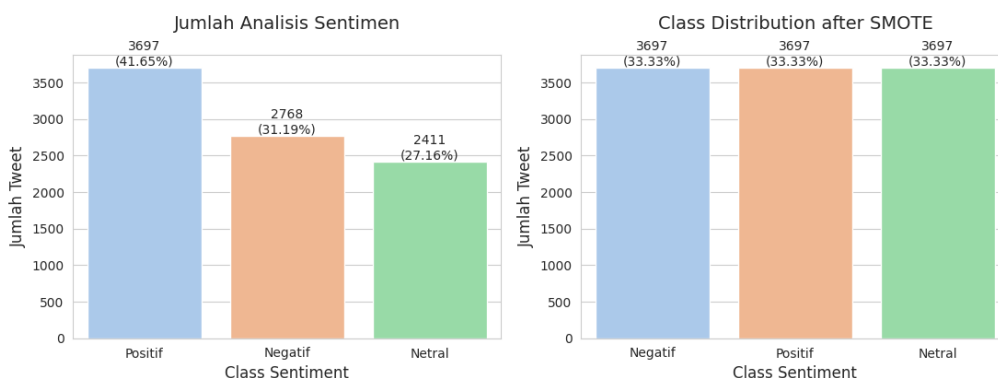
Tabel 9. Hasil TF-ID

	Term	TFIDF SUM
12563	ya	585.424302
10039	scam	499.438363
293	akun	496.129705
4836	investasi	484.726090
5384	kai	483.479900

Tabel 9 menunjukkan hasil dari perhitungan TF-IDF, di mana kata-kata seperti "ya," "scam," dan "akun" memiliki nilai TF-IDF tertinggi, mencerminkan pentingnya kata-kata tersebut dalam konteks komentar mengenai scam.

### 3.6 Optimasi SMOTE

SMOTE (*Synthetic Minority Over-sampling Technique*) diterapkan sebagai solusi untuk menangani ketimpangan jumlah data antar kelas dalam dataset. Gambar 5 menunjukkan visualisasi perbandingan antara kondisi data sebelum dan sesudah dilakukan proses oversampling dengan metode SMOTE.



Gambar 5. Visualisasi Sebelum SMOTE dan Sesudah SMOTE

Gambar 5 menampilkan perbandingan distribusi data sebelum dan sesudah penerapan metode SMOTE. Sebelum proses ini dilakukan, data didominasi oleh sentimen positif, sementara jumlah data untuk sentimen negatif dan netral relatif sedikit. Ketimpangan ini menyebabkan model cenderung mengklasifikasikan komentar sebagai positif, yang data pada kelas *minoritas* diperluas melalui penciptaan sampel *synthetic*, sehingga distribusi antar kelas menjadi lebih seimbang. Dengan demikian, model memiliki peluang yang lebih besar untuk mengenali sentimen negatif dan netral secara lebih tepat.

### 3.7 Hasil Pengujian Model

Bagian ini menyajikan hasil evaluasi terhadap performa model dalam mengklasifikasikan sentimen positif, negatif, dan netral pada data Twitter yang telah melalui proses prapemrosesan. Proses pengujian dilakukan dengan membagi dataset menjadi 80% untuk pelatihan dan 20% untuk pengujian. Model dievaluasi dengan menggunakan metrik seperti *akurasi*, *presisi*, *recall*, dan *F1-score*. Hasil yang diperoleh ditampilkan dalam dua tahap, yaitu sebelum dan sesudah penerapan metode SMOTE untuk penyeimbangan data..

#### 3.7.1 Hasil Sebelum Penerapan SMOTE:

Sebelum penerapan teknik SMOTE, terdapat perbedaan kinerja yang cukup signifikan di antara ketiga model. SVM mencatat hasil akurasi terbaik mencapai 82%, selanjutnya, model *Random Forest* mencatat akurasi sebesar 79%, sedangkan *Naive Bayes* memperoleh akurasi 74%. Tabel 10 di bawah ini menunjukkan hasil evaluasi untuk model SVM, *Random Forest*, dan *Naive Bayes*.

Tabel 10. Hasil Sebelum Penerapan SMOTE

Model	Sentimen	Precision	Recall	F1-Score
SVM	Positif	89%	89%	89%
	Negatif	83%	81%	82%
	Netral	71%	73%	72%
Random Forest	Positif	87%	86%	86%
	Negatif	78%	75%	77%

Model	Sentimen	Precision	Recall	F1-Score
Naive Bayes	Netral	69%	73%	71%
	Positif	76%	87%	81%
	Negatif	74%	73%	74%
	Netral	70%	56%	62%

Berdasarkan Tabel 10 SVM menunjukkan hasil terbaik di semua kategori sentimen, diikuti oleh Random Forest dan Naive Bayes memiliki kinerja yang lebih rendah, terutama pada sentimen Netral, dengan Recall yang sangat rendah pada kategori tersebut.

### 3.7.2 Hasil Setelah Penerapan SMOTE:

Setelah penerapan teknik SMOTE, kinerja ketiga model menunjukkan peningkatan yang signifikan pada berbagai metrik evaluasi. *Support Vector Machine* (SVM) unggul dengan precision tertinggi untuk sentimen positif 94%, negatif 91%, dan netral 81%, serta recall tinggi pada sentimen positif 86%, negatif 91%, dan netral 89%, menghasilkan *F1-score* terbaik pada masing-masing kategori 89%, 91%, dan 85%. Di sisi lain, *Random Forest* memiliki memiliki performa yang cukup baik dengan *precision* 90% untuk sentimen positif dan *F1-score* 81% untuk netral. *Naive Bayes* menunjukkan kinerja terendah, dengan *precision* tertinggi 84% untuk positif dan *F1-score* 80% pada sentimen negatif. Hasil dari model algoritma bisa dilihat dalam Tabel 11.

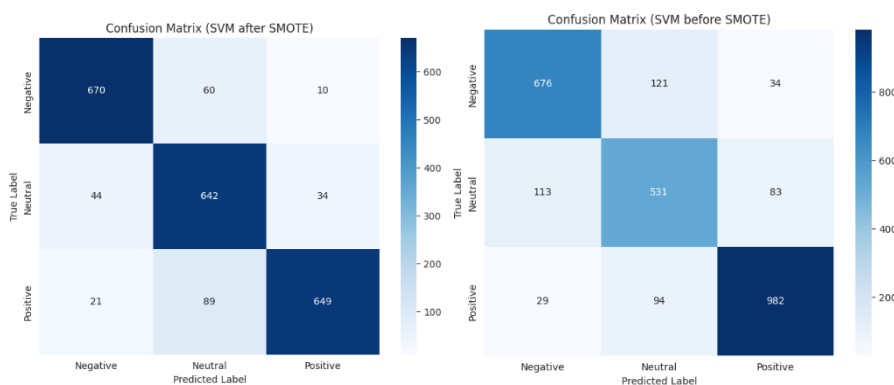
Tabel 11. Hasil Setelah Penerapan SMOTE

Model	Sentimen	Precision	Recall	F1-Score
SVM	Positif	94%	86%	89%
	Negatif	91%	91%	91%
	Netral	81%	89%	85%
Random Forest	Positif	90%	83%	86%
	Negatif	84%	85%	85%
	Netral	77%	84%	81%
Naive Bayes	Positif	84%	72%	78%
	Negatif	79%	80%	80%
	Netral	68%	77%	72%

Dari Tabel 11 diatas SVM masih menunjukkan model paling andal dalam klasifikasi sentimen setelah penyeimbangan data dengan SMOTE, karena konsistensinya dalam menghasilkan nilai precision, recall, dan F1-score yang tinggi pada ketiga kategori sentimen, diikuti oleh Random Forest yang menunjukkan performa stabil, serta Naive Bayes yang berada di posisi terakhir dengan hasil evaluasi yang relatif lebih rendah.

### 3.8 Confusion Matrix - Evaluasi Model

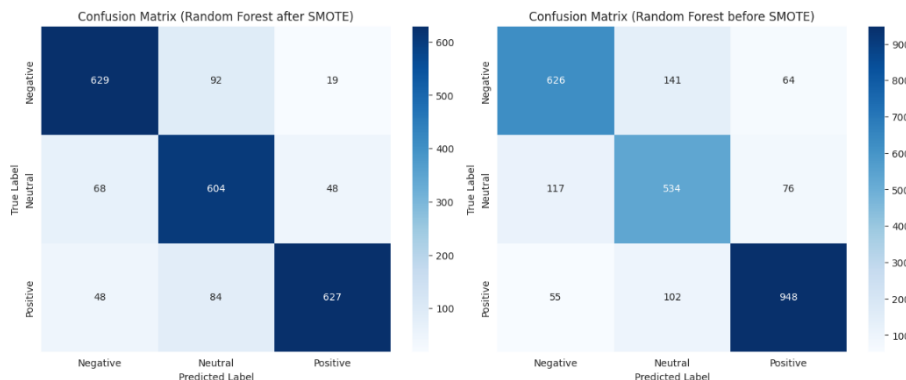
Dalam upaya menentukan algoritma yang paling optimal, peneliti melakukan serangkaian. Eksperimen ini berfokus pada perbandingan nilai-nilai *confusion matrix* yang dihasilkan oleh tiga algoritma yang diuji. Perbandingan ini bertujuan untuk menilai kemampuan masing-masing model dalam mengklasifikasikan data secara tepat. Visualisasi dari perbandingan hasil dapat dilihat pada Gambar 6.



Gambar 6. Perbandingan Confusion Matrix SVM before SMOTE dan after SMOTE

Gambar 6 menunjukkan perbandingan *confusion matrix* dari model *Support Vector Machine* (SVM) pada tahap sebelum dan setelah diterapkannya metode SMOTE. Sebelum SMOTE diterapkan, model menunjukkan kecenderungan bias terhadap kelas mayoritas, khususnya kelas positif, yang ditandai dengan jumlah prediksi benar (*True Positive*) yang sangat tinggi, yaitu 982. Sebaliknya, performa pada kelas negatif dan netral tampak lebih rendah, dengan cukup banyak kesalahan klasifikasi ke kelas lain. Hal ini menunjukkan adanya ketidakseimbangan data yang memengaruhi akurasi model dalam mengenali kelas *minoritas*. kelas negatif dan netral tampak lebih rendah, dengan

cukup banyak kesalahan klasifikasi ke kelas lain. Hal ini menunjukkan adanya ketidakseimbangan data yang memengaruhi akurasi model dalam mengenali kelas *minoritas*. Setelah SMOTE diterapkan, terlihat adanya peningkatan kinerja model secara signifikan. Distribusi prediksi menjadi lebih seimbang, dan kesalahan klasifikasi ke kelas positif menurun secara drastis. Jumlah data negatif yang sebelumnya diklasifikasikan sebagai positif turun dari 34 menjadi hanya 10, dan prediksi netral yang salah menjadi positif juga berkurang dari 83 menjadi 34.



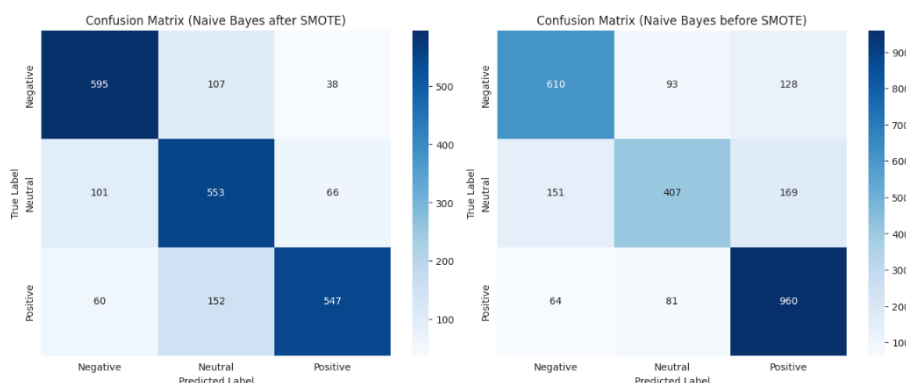
Gambar 7. Perbandingan *Confusion Matrix Random Forest* before SMOTE dan after SMOTE

Gambar 7 menampilkan *confusion matrix* dari model *Random Forest* pada kondisi sebelum dan sesudah penerapan SMOTE. Sebelum SMOTE model lebih akurat pada kelas positif dengan *true positive* sebanyak 948, namun banyak salah klasifikasi pada kelas negatif yaitu 141 ke netral dan 64 ke positif, serta pada kelas netral yaitu 117 ke negatif dan 76 ke positif. *True positive* untuk kelas negatif dan netral masing-masing sebesar 626 dan 534. Setelah SMOTE, prediksi menjadi lebih seimbang, dengan *true positive* untuk kelas negatif, netral, dan positif masing-masing sebesar 629, 604, dan 627. Selain itu, kesalahan klasifikasi negatif ke positif turun dari 64 menjadi 19.

Gambar 8 memperlihatkan hasil *confusion matrix Naive Bayes*, baik sebelum maupun setelah penerapan teknik SMOTE diterapkan. Sebelum SMOTE, model sangat akurat pada kelas positif dengan *true positive* sebesar 960, namun memiliki kinerja rendah pada kelas netral dan negatif, dengan *true positive* masing-masing hanya 407 dan 610. Terdapat kesalahan signifikan, seperti 151 data netral diklasifikasikan sebagai negatif dan 169 ke positif. Setelah SMOTE, performa model menjadi lebih seimbang dengan *true positive* untuk kelas negatif, netral, dan positif masing-masing sebesar 595, 553, dan 547. Selain itu, kesalahan klasifikasi pada kelas negatif ke positif turun dari 128 menjadi hanya 38. Hal ini menunjukkan bahwa SMOTE berhasil memperbaiki ketidakseimbangan data dan membuat model lebih adil dalam mengklasifikasikan ketiga kelas.

### 3.9 Visualisasi Perbandingan Algoritma

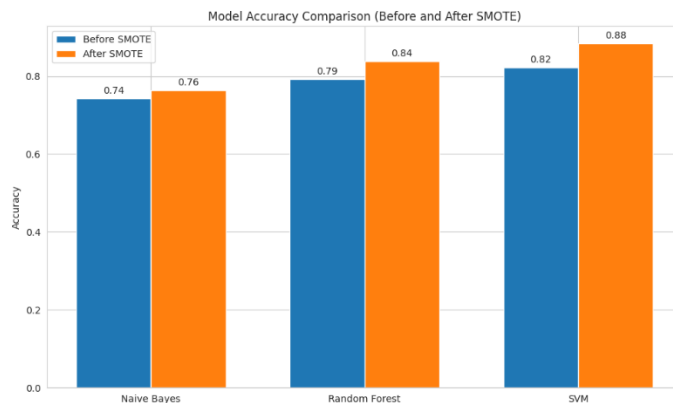
Visualisasi perbandingan algoritma memperlihatkan perbedaan akurasi model *Naive Bayes*, *Random Forest*, dan SVM sebelum dan setelah penggunaan teknik SMOTE (Synthetic Minority Over-sampling Technique). Dari grafik terlihat bahwa seluruh model mengalami peningkatan akurasi usai diterapkannya SMOTE. Model *Naive Bayes* mengalami kenaikan dari 74% menjadi 76%, *Random Forest* meningkat dari 79% menjadi 84%, dan SVM naik dari 82% menjadi 88%. Kenaikan yang paling menonjol terjadi pada *Random Forest* dan SVM. Hal ini menunjukkan SMOTE terbukti efektif dalam menangani ketidakseimbangan data dan meningkatkan performa model dalam melakukan klasifikasi sentimen secara lebih seimbang. Gambar visualisasi perbandingan dapat dilihat pada Gambar 8.



Gambar 8. Perbandingan *Confusion Matrix Naive Bayes* Confusion before SMOTE dan after SMOTE

Berdasarkan Gambar 8 peningkatan akurasi tertinggi ditunjukkan oleh model SVM, menjadikannya model yang paling efektif dalam menganalisis kasus scam berdasarkan data yang telah diseimbangkan menggunakan SMOTE. Kemampuan SVM dalam memisahkan kelas dengan margin yang optimal membuatnya unggul dalam

menangani data yang kompleks dan tidak seimbang. Efektivitas berikutnya ditunjukkan oleh Random Forest yang juga mengalami peningkatan signifikan, disusul oleh Naive Bayes yang meskipun peningkatannya lebih kecil.



Gambar 9. Visualisasi Perbandingan Algoritma

#### 4. KESIMPULAN

Penelitian ini menyajikan pemahaman yang lebih mendalam mengenai perbandingan kinerja model klasifikasi seperti *Support Vector Machine (SVM)*, *Random Forest*, dan *Naive Bayes* dalam analisis sentimen terhadap *tweet* yang berkaitan dengan topik *scam* atau penipuan di media sosial Twitter. Penelitian ini memperlihatkan bahwa pada tahap sebelum SMOTE, SVM menunjukkan memperoleh hasil terbaik melalui akurasi sebesar 82%, Random Forest memperoleh akurasi 79%, diikuti oleh Naive Bayes yang menghasilkan akurasi terendah, yakni 74%. Setelah penerapan SMOTE, seluruh model mengalami peningkatan kinerja, dengan SVM tetap menjadi yang terbaik dengan akurasi mencapai 88%, disusul *Random Forest* sebesar 84%, dan *Naive Bayes* meningkat menjadi 76%. Dalam penelitian ini, SVM terbukti sebagai algoritma terbaik, menunjukkan performa tertinggi baik pada kondisi sebelum maupun setelah penerapan SMOTE. Penelitian ini memiliki beberapa batasan yang dapat diperbaiki. Pertama, jumlah data yang digunakan dalam penelitian ini masih relatif terbatas dan cenderung didominasi oleh *tweet* dengan sentimen positif. Hal ini kemungkinan besar disebabkan oleh keterbatasan proses pelabelan yang digunakan pada tahap *preprocessing*. Beberapa kata atau frasa bermakna negatif mungkin tidak berhasil teridentifikasi secara akurat, sehingga menyebabkan bias dalam klasifikasi yang mengarah pada dominasi sentimen positif. Perbaikan terhadap kamus sentimen atau pendekatan berbasis *machine learning* dapat menjadi langkah strategis dalam meningkatkan kualitas analisis pada penelitian selanjutnya. Selain itu, sumber data yang sepenuhnya diambil dari platform media sosial Twitter dapat membatasi cakupan representasi opini publik. Untuk memperoleh hasil yang lebih menyeluruh, disarankan agar penelitian selanjutnya menggabungkan data dari berbagai platform media sosial dan menggunakan pendekatan lanjutan seperti *deep learning* guna menangkap dimensi emosional yang lebih kompleks dan beragam.

#### REFERENCES

- [1] W. Priatna, "Dampak Pengambilan Sampel Data untuk Optimalisasi Data tidak seimbang pada Klasifikasi Penipuan Transaksi E-Commerce," *Indones. J. Comput. Sci.*, vol. 13, no. 2, pp. 3070–3079, 2024, doi: 10.33022/ijcs.v13i2.3698.
- [2] R. Fahlapi, A. Y. Kuntoro, and T. Asra, "Perbandingan Algoritma Klasifikasi Analisis Sentimen Pengguna Aplikasi Getcontact Dalam Pencegahan Penipuan Online," *J-INTECH (Journal Inf. Technol.)*, no. 204, pp. 158–167, 2024, doi: <https://doi.org/10.32664/j-intech.v12i1.1262>.
- [3] A. T. Arsanto, A. Faizin, Z. N. Saadah, U. Y. Pasuruan, and C. V. Method, "Optimasi Algoritma Naive Bayes dengan Kombinasi SMOTETomek untuk Imbalance Class Fraud Detection," *Sist. J. Sist. Inf.*, vol. 13, pp. 2709–2721, 2024, doi: <https://doi.org/10.32520/stmsi.v13i6.4719>.
- [4] S. Adi *et al.*, "Analisis Sentimen Aplikasi Halo Bca Di Google Play Store Menggunakan Metode Naive Bayes , Support Vector Machine Dan Random Forest Pendahuluan," *HOAQ (High Educ. Organ. Arch. Qual. J. Teknol. Inf.)*, vol. 15, no. c, pp. 69–79, 2024, doi: <https://doi.org/10.52972/hoaq.vol15no2.p69-79>.
- [5] H. Nalattisifa, W. Gata, S. Diantika, and K. Nisa, "Perbandingan Kinerja Algoritma Klasifikasi Naive Bayes , Support Vector Machine ( SVM ), dan Random Forest untuk Prediksi Ketidakhadiran di Tempat Kerja," *J. Inform.*, vol. 5, no. 4, pp. 578–584, 2021, doi: <https://doi.org/10.32493/informatika.v5i4.7575>.
- [6] Y. W. Sitorus, P. Sukarno, and S. Mandala, "Analisis Deteksi Malware Android menggunakan metode Support Vector Machine & Random Forest," *Eng. J. Telkom Univ.*, vol. 8, no. 6, pp. 12500–12518, 2021, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/16864>
- [7] M. Samantri and Afyati, "Perbandingan Algoritma Support Vector Machine dan Random Forest untuk Analisis Sentimen Terhadap Kebijakan Pemerintah Indonesia Terkait Kenaikan Harga BBM Tahun 2022," *J. JTIK (Jurnal Teknol. Inf. Dan Komunikasi)*, vol. 8, no. 1, pp. 1–9, 2024, doi: <https://doi.org/10.35870/jtik.v8i1.1202>.
- [8] P. Cahyani and L. Abdillah, "Perbandingan Performa Algoritma Naive Bayes , SVM dan Random Forest : Studi Kasus Analisis Sentimen Pengguna Sosial Media X," *KALBISCIENTIA (Jurnal Sains dan Teknol.)*, vol. 11, no. 02, pp. 12–21, 2024, doi: <https://doi.org/10.53008/kalbiscientia.v11i02.3624>.



- [9] D. Irawan, E. B. Perkasa, D. Wahyuningsih, and E. Helmud, “Perbandingan Klasifikasi SMS Berbasis Support Vector Machine , Naive Bayes Classifier , Random Forest dan Bagging Classifier,” *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 10, pp. 432–437, 2021, doi: <https://doi.org/10.32736/sisfokom.v10i3.1302>.
- [10] P. I. Purnama and R. R. Suryono, “Analisis Sentimen Acara Clash of Champions dengan Algoritma Naive Bayes dan Support Vector Machine,” *Build. Informatics, Technol. Sci.*, vol. 6, no. 4, pp. 2277–2287, 2025, doi: <https://doi.org/10.47065/bits.v6i4.6575>.
- [11] D. N. Agustia, R. R. Suryono, U. T. Indonesia, L. Ratu, and K. B. Lampung, “Comparison Of Naive Bayes , Random Forest , And Logistic Regression Algorithms For Sentiment Analysis Online Gambling Komparasi Algoritma Naive Bayes , Random Forest , Dan Logistic Regresion Untuk Analisis,” *INOVTEK Polbeng*, vol. 10, no. 1, pp. 284–295, 2025, doi: <https://doi.org/10.35314/prk93630>.
- [12] F. A. Artanto, “Support Vector Machine Berbasis Particle Swarm Optimization Pada Analisis Sentimen Anggota KPPS,” *J. Fasilkom*, vol. 14, no. 1, pp. 75–79, 2024, doi: <https://doi.org/10.37859/jf.v14i1.6795>.
- [13] S. Syafrizal, M. Afdal, and R. Novita, “Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naive Bayes Classifier dan K-Nearest Neighbor,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 10–19, 2023, doi: <https://doi.org/10.57152/malcom.v4i1.983>.
- [14] A. Karimah, G. Dwilestari, and M. Mulyawan, “Analisis Sentimen Komentar Video Mobil Listrik Di Platform Youtube Dengan Metode Naive Bayes,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 1, pp. 767–737, 2024, doi: <https://doi.org/10.36040/jati.v8i1.8373>.
- [15] I. F. Rahman, A. N. Hasanah, and N. Heryana, “Analisis Sentimen Ulasan Pengguna Aplikasi Samsat Digiital Nasional (Signal) Dengan Menggunakan Metode Naive Bayes Classifier,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 963–969, 2024, doi: <http://dx.doi.org/10.23960/jitet.v12i2.4073>.
- [16] J. Ipmawati, S. Saifulloh, and K. Kusnawi, “Analisis Sentimen Tempat Wisata Berdasarkan Ulasan pada Google Maps Menggunakan Algoritma Support Vector Machine,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 247–256, 2024, doi: <https://doi.org/10.57152/malcom.v4i1.1066>.
- [17] A. T. Susilawati, N. A. Lestari, and P. A. Nina, “Analisis Sentimen Publik Pada Twitter Terhadap Boikot Produk Israel Menggunakan Metode Naive Bayes,” *Nian Tana Sikk. J. Ilm. Mhs.*, vol. 2, no. 1, pp. 26–35, 2024, doi: <https://doi.org/10.59603/niantanasikka.v2i1.240>.
- [18] A. Kevin, K., Enjeli, M., & Wijaya, “Analisis Sentimen Penggunaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes,” *J. Ilm. Comput. Sci.*, vol. 2, pp. 89–98, 2024, doi: <https://doi.org/10.58602/jics.v2i2.24>.
- [19] C. C. Muhamad Fajar Yudhistira Herjanto, “Analisis Sentimen Ulasan Pengguna Aplikasi Sirekap Pada Play Store Menggunakan Algoritma Random Forest Classifier,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 1204–1210, 2024, doi: <http://dx.doi.org/10.23960/jitet.v12i2.4192>.
- [20] E. Yuniar and N. Hendrastuty, “Perbandingan Metode Naive Bayes , Random Forest dan SVM Untuk Analisis Sentimen Pada Twitter Tentang Kenaikan Gaji Guru,” *Build. Informatics, Technol. Sci.*, vol. 6, no. 4, pp. 2480–2490, 2025, doi: <https://doi.org/10.47065/bits.v6i4.6970>.
- [21] A. N. Hasanah, B. N. Sari, U. S. Karawang, T. Timur, and J. Barat, “Analisis Sentimen Ulasan Pengguna Aplikasi Jasa Ojek Online Maxim Pada Google Play Dengan Metode Naive Bayes Classifier,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 1, pp. 90–96, 2024, doi: <http://dx.doi.org/10.23960/jitet.v12i1.3628>.
- [22] R. A. Saputra, “Analisis Sentimen Review Skincare Skintific Dengan Algoritma Support Vector Machine,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: <http://dx.doi.org/10.23960/jitet.v12i2.4095>.
- [23] S. Andean, “Review aplikasi kredivo menggunakan analisis sentimen dengan algoritma support vector machine 1),” *rabit*, vol. 9, no. 1, pp. 39–49, 2024, doi: <https://doi.org/10.36341/rabit.v9i1.4107>.
- [24] I. Septiana and D. Alita, “Perbandingan Random Forest dan SVM dalam Analisis Sentimen Quick Count Pemilu 2024,” *JPIT(Jurnal Pengemb. IT)*, vol. 9, no. 3, pp. 224–233, 2024, doi: <https://doi.org/10.30591/jpit.v9i3.6640>.
- [25] B. A. Maulana and M. J. Fahmi, “Sentiment Analysis of Pluang Applications With Naive Bayes and Support Vector Machine ( SVM ) Algorithm Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine ( SVM ),” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. April, pp. 375–384, 2024, doi: <https://doi.org/10.57152/malcom.v4i2.1206>.
- [26] W. Ningsih, B. Alfianda, and D. Wulandari, “Comparison of Naive Bayes and SVM Algorithms in Twitter Sentiment Analysis on Electric Car Use in Indonesia Perbandingan Algoritma SVM dan Naive Bayes dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. April, pp. 556–562, 2024, doi: <https://doi.org/10.57152/malcom.v4i2.1253>.
- [27] K. W. Chandra and H. Irsyad, “Efektifitas SMOTE dalam Mengatasi Imbalanced Class Algoritma K-Nearest Neighbors pada Analisis Sentimen terhadap Starlink,” *J. Ilmu Komput. Dan Inform.*, vol. 4, no. 1, pp. 31–42, 2024, doi: <https://doi.org/10.54082/jiki.132>.
- [28] Sasmita, R. N. J. S. Intam, D. F. Surianto, and Muhammad Fajar, “Analisis Sentimen Terhadap Kontroversi Putusan MK Mengenai Usia Capres-Cawapres Menggunakan Multi-Layer Perceptron Dengan Teknik SMOTE,” *Fakt. Exacta*, vol. 17, no. 2, pp. 188–198, 2024, doi: [10.30998/faktorexacta.v17i2.22442](https://doi.org/10.30998/faktorexacta.v17i2.22442).