

# Komparasi Metode Naïve Bayes, Random Forest dan KNN untuk Analisis Sentimen Penambangan Nikel

Beta Agus Setiyana<sup>1</sup>, Ryan Randy Suryono<sup>2,\*</sup>

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

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

Email: <sup>1</sup>beta\_agus\_setiyana@teknokrat.ac.id, <sup>2,\*</sup> ryan@teknokrat.ac.id

Email Penulis Korespondensi: ryan@teknokrat.ac.id

Submitted: 01/09/2025; Accepted: 30/09/2025; Published: 30/09/2025

**Abstrak**—Fenomena meningkatnya eksploitasi sumber daya alam di kawasan konservasi Indonesia menimbulkan kekhawatiran publik, salah satunya terkait rencana penambangan nikel di Raja Ampat yang dikenal sebagai pusat keanekaragaman hayati laut dunia. Rencana tersebut memicu perdebatan antara kepentingan ekonomi, pelestarian lingkungan, dan nilai sosial budaya masyarakat setempat, di tengah maraknya diskusi publik media sosial menjadi wadah utama bagi masyarakat untuk mengekspresikan opini, dukungan, maupun penolakan terhadap aktivitas penambangan. Penelitian ini bertujuan untuk memetakan sentimen publik terhadap isu penambangan nikel di Raja Ampat melalui analisis 5.556 cuitan berbahasa Indonesia di media sosial X dengan kata kunci “save raja ampat” selama Januari–Juni 2025. Data diproses melalui tahapan *cleaning*, *case folding*, *tokenizing*, *stopword removal*, dan *normalisasi* sebelum direpresentasikan menggunakan metode *TF-IDF*. Pelabelan sentimen dilakukan secara semi otomatis berbasis *leksikon* dalam tiga kategori, yaitu positif, netral, dan negatif, dengan hasil distribusi *sentimen* netral (72,9%), negatif (24,3%), dan positif (2,8%), yang menunjukkan ketidakseimbangan kelas. Untuk mengatasinya diterapkan teknik *SMOTE* pada data latih. Tiga algoritma klasik *K-Nearest Neighbor (KNN)*, *Complement Naïve Bayes (CNB)*, dan *Random Forest (RF)* dibandingkan menggunakan *cross-validation* dan *holdout testing* dengan *matrik akurasi*, *presisi*, *recall*, dan *F1-score*. Hasil menunjukkan bahwa *CNB* paling stabil sebelum *SMOTE*, sedangkan setelah *SMOTE*, *KNN* mengalami peningkatan signifikan terutama pada *recall* dan *F1 makro*. Temuan ini menegaskan relevansi kombinasi teknik penyeimbangan data dan algoritma klasik sebagai *baseline metodologis* dalam analisis sentimen publik pada isu lingkungan kompleks seperti penambangan nikel di Raja Ampat.

**Kata Kunci:** Analisis Sentimen; Raja Ampat; Save Raja Ampat; Naïve Bayes; Random Forest; K-Nearest Neighbor; SMOTE; TF-IDF.

**Abstract**—The phenomenon of increasing natural resource exploitation in Indonesia’s conservation areas has raised significant public concern, one of which involves the planned nickel mining project in Raja Ampat, a region renowned for its extraordinary marine biodiversity. This plan has sparked debates between economic interests, environmental preservation, and the sociocultural values of local communities. Amid the growing public discourse, social media has become a major platform for people to express their opinions, support, or opposition toward mining activities. This study aims to map public sentiment regarding the nickel mining issue in Raja Ampat by analyzing 5,556 Indonesian-language tweets collected from the social media platform X using the keyword “save raja ampat” between January- June 2025. The data underwent several preprocessing stages, including cleaning, case folding, tokenizing, stopwords removal, and normalization, and were then represented using the TF-IDF method. Sentiment labeling was performed semi automatically using a lexicon based approach into three categories: positive, neutral, and negative. The sentiment distribution showed dominance of neutral (72.9%), followed by negative (24.3%) and positive (2.8%), indicating class imbalance. To address this issue, the SMOTE technique was applied to the training data. Three classical algorithms K-Nearest Neighbor (KNN), Complement Naïve Bayes (CNB), and Random Forest (RF) were compared using cross-validation and holdout testing with accuracy, precision, recall, and F1-score as evaluation metrics. The results show that CNB performed most stably before SMOTE, while after SMOTE, KNN demonstrated significant improvement, especially in recall and macro F1-score. These findings confirm that the combination of data balancing techniques and classical algorithms remains relevant and efficient as a methodological baseline for public sentiment analysis on complex environmental issues such as nickel mining in Raja Ampat.

**Keywords:** Sentiment Analysis; Raja Ampat; Save Raja Ampat; Naive Bayes; Random Forest; K-Nearest Neighbors; SMOTE; TF-IDF.

## 1. PENDAHULUAN

Isu penambangan nikel di Raja Ampat telah memicu perdebatan publik yang tajam di Indonesia. Kawasan ini tidak hanya dikenal sebagai pusat keanekaragaman hayati laut dunia, tetapi juga memiliki nilai ekologis dan sosial yang tinggi bagi masyarakat lokal. Di satu sisi, proyek pertambangan dipandang sebagai motor penggerak ekonomi daerah sekaligus penopang industri baterai dalam agenda transisi energi. Namun di sisi lain, kekhawatiran mengenai degradasi ekologis, perubahan bentang alam pesisir, serta dampak sosial terhadap mata pencaharian masyarakat adat semakin menguat[1]. Polarisasi ini diperkuat oleh dinamika media sosial, di mana opini publik terbentuk, dipertukarkan, dan disebarluaskan dalam tempo cepat. Dalam konteks ini, analisis sentimen menjadi instrumen penting untuk memetakan persepsi publik secara lebih terukur dan obyektif, sehingga dapat memberikan masukan bagi pembuat kebijakan, pelaku industri, maupun masyarakat sipil[2].

Analisis sentimen pada teks berbahasa Indonesia menghadapi sejumlah tantangan. Hal ini disebabkan oleh karakteristik bahasa yang digunakan di media sosial yang cenderung bersifat informal, mengandung banyak singkatan, campur kode antara bahasa Indonesia dan bahasa asing, serta penggunaan ejaan yang tidak baku.[3]. Selain itu,

keberadaan elemen-elemen non-teks seperti tautan, mention, tagar, emoji, maupun berbagai bentuk *noise* lainnya sering kali mengganggu proses ekstraksi fitur secara optimal. Permasalahan lainnya adalah distribusi kelas sentimen yang tidak seimbang, di mana unggahan dengan sentimen netral sering kali mendominasi dan menyebabkan model cenderung bias terhadap kelas tertentu. Keterbatasan dalam jumlah dan kualitas anotasi manual juga menjadi hambatan signifikan dalam penyusunan dataset yang representatif dan reliabel untuk pelatihan model.[4]. Oleh karena itu, penerapan strategi preprocessing yang tepat, pemilihan metode representasi fitur yang sesuai seperti TF-IDF atau word embeddings, serta penerapan teknik penyeimbangan kelas seperti *SMOTE* menjadi aspek krusial dalam memastikan hasil analisis sentimen yang akurat, valid, dan memiliki kemampuan generalisasi yang baik.[5].

Sejumlah penelitian terdahulu menunjukkan bahwa algoritma klasik masih relevan dan efektif untuk analisis sentimen dalam berbagai domain [6]. Algoritma seperti *Naïve Bayes* (NB) terbukti efisien dalam memproses teks pendek serta mampu menangani data dengan distribusi kata yang tidak seimbang, sehingga sering digunakan pada analisis opini publik di media sosial.[7]. Sementara itu, *K-Nearest Neighbor* (KNN) banyak diaplikasikan pada penelitian kebijakan sosial dan survei opini masyarakat karena kesederhanaannya dalam mengelompokkan data berdasarkan kemiripan antar teks [8]. Di sisi lain, *Random Forest* (RF) kerap digunakan pada analisis sentimen di sektor e-commerce dan isu lingkungan, meskipun performanya dapat bervariasi pada data teks yang jarang (*sparse*).[9]. Dalam hal representasi fitur, metode *TF-IDF* dan *N-grams* terbukti efektif dalam menangkap konteks kata serta pola linguistik yang muncul dalam teks. Namun, permasalahan ketidakseimbangan kelas masih menjadi tantangan umum dalam analisis sentimen, sehingga banyak penelitian menerapkan teknik resampling seperti *SMOTE* untuk menyeimbangkan distribusi data dan meningkatkan akurasi model dalam memprediksi sentimen secara lebih konsisten dan representative [10].

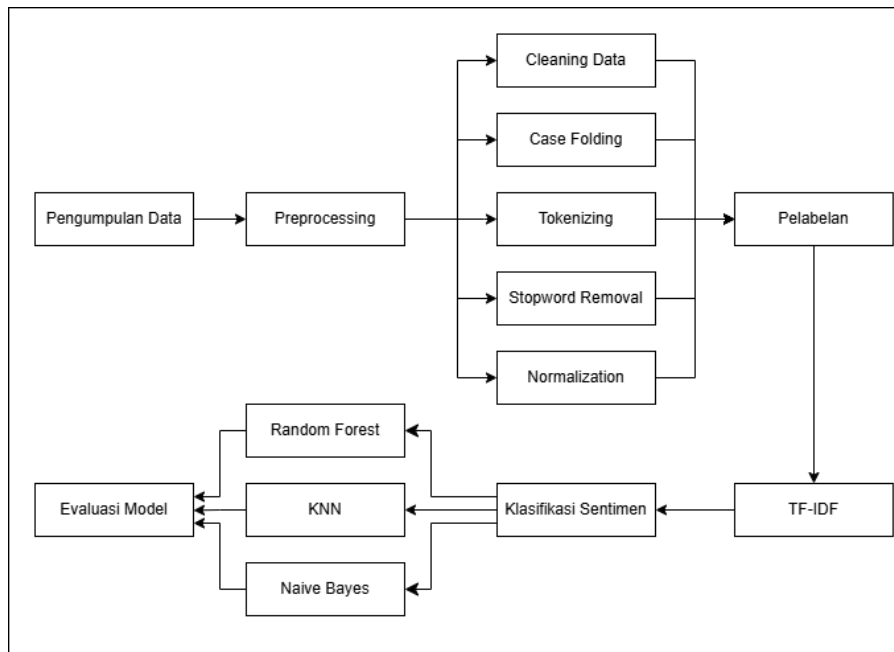
Beberapa studi serupa antaranya penelitian oleh Fitri Dwianasari et al. (2025) menganalisis sentimen publik terhadap isu penambangan di Raja Ampat menggunakan SVM dan NB. Hasilnya menunjukkan bahwa SVM memiliki akurasi lebih tinggi (80%) dibandingkan NB (68%), serta teknik *SMOTE* berhasil meningkatkan deteksi sentimen positif [11]. Selanjutnya, Norlaila et al. (2024) menggunakan kombinasi *K-Means* dan NB untuk menganalisis sentimen tambang emas di Indonesia dengan akurasi mencapai 99%, meskipun terdapat indikasi overfitting akibat data yang terbatas [12]. Raka Ardiansyah Hasibuan et al. (2024) juga menggunakan SVM dengan berbagai kernel (*Linear*, *Polynomial*, *RBF*, dan *Sigmoid*) untuk mengkaji kebijakan ekspor pasir laut, di mana kernel *Polynomial* memberikan hasil terbaik dengan akurasi 80,94% [13]. Sementara itu, Budi Wijaya Rauf (2023) menerapkan metode NB dan *TF-IDF* untuk menganalisis sentimen masyarakat terkait pertambangan di Konawe Utara, menghasilkan akurasi 85%, presisi 83,5%, dan *recall* 87,5%, yang menunjukkan efektivitas algoritma NB dalam menangani data teks pendek dengan distribusi sentimen yang bias negative [14].

Berdasarkan hasil penelitian-penelitian terdahulu, dapat disimpulkan bahwa masih sangat terbatas studi yang secara khusus menyoroitu isu penambangan nikel di kawasan konservasi Raja Ampat sebagai fokus analisis sentimen publik. Selain itu, meskipun algoritma klasik seperti *Naïve Bayes* (NB), *K-Nearest Neighbor* (KNN), dan *Random Forest* (RF) telah banyak digunakan dalam berbagai konteks, hanya sedikit penelitian yang secara sistematis membandingkan kinerja ketiganya dengan mempertimbangkan ketidakseimbangan kelas menggunakan teknik *SMOTE* pada teks berbahasa Indonesia. Penelitian ini berupaya mengisi celah tersebut dengan membangun dataset publik dari media sosial X dan melakukan perbandingan komprehensif tiga algoritma klasik menggunakan representasi fitur TF-IDF. Dengan pendekatan ini, penelitian tidak hanya menyajikan *benchmark komparatif* yang dapat direplikasi, tetapi juga memberikan kontribusi metodologis dalam memahami pengaruh distribusi kelas serta keterbatasan label terhadap performa model pada isu strategis di bidang lingkungan dan pertambangan nasional, sekaligus mendorong pengembangan penelitian lanjutan yang lebih mendalam dan kontekstual di masa depan.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini dilaksanakan secara sistematis dan terstruktur untuk menganalisis sentimen publik terkait isu “Pray for Raja Ampat” di media sosial X (Twitter). Data dikumpulkan melalui proses crawling menggunakan kata kunci yang relevan serta rentang waktu unggahan tertentu, sehingga terkumpul sebanyak 5.556 data sentimen yang merepresentasikan opini publik terhadap isu tersebut. Setelah data diperoleh, dilakukan tahap preprocessing yang mencakup cleansing, case folding, tokenizing, stopword removal, dan normalization agar teks menjadi bersih dan konsisten. Selanjutnya, data diberi label ke dalam tiga kategori sentimen, yaitu positif, netral, dan negatif, berdasarkan konteks isi unggahan. Proses klasifikasi dilakukan menggunakan metode TF-IDF untuk mengekstraksi fitur teks, kemudian membangun tiga model pembelajaran mesin, yaitu Random Forest (RF), *Naïve Bayes* (NB), dan *K-Nearest Neighbor* (KNN). Kinerja ketiga model tersebut dibandingkan menggunakan metrik precision, recall, dan F1-score untuk menilai efektivitas masing-masing algoritma dalam mengenali pola sentimen publik. Secara keseluruhan, alur penelitian yang menggambarkan tahapan pengumpulan data, pemrosesan, klasifikasi, hingga evaluasi hasil divisualisasikan pada Gambar 1.



Gambar 1 Alur Penelitian

## 2.2 Praprocessing Data

Tahap *preprocessing* data merupakan langkah penting yang dilakukan sebelum analisis teks dimulai agar data yang digunakan menjadi lebih bersih, terstruktur, dan siap diolah secara optimal [15]. Proses ini bertujuan untuk menghilangkan elemen-elemen yang tidak relevan serta memastikan konsistensi format teks sehingga hasil analisis menjadi lebih akurat dan bermakna, proses ini meliputi:

- Cleaning data, yaitu menghapus tanda baca, angka, simbol, URL, hashtag, mention, serta spasi ganda.
- Case folding, menyeragamkan semua huruf menjadi huruf kecil.
- Tokenizing, memecah kalimat menjadi unit kata (token).
- Stopword removal, menghilangkan kata umum yang tidak memberi makna penting.
- Normalization, menyeragamkan kata tidak baku atau singkatan ke bentuk standar.

## 2.3. Pelabelan Sentimen

Setelah itu kata diberi sentimen secara manual menjadi tiga kategori: positif (dukungan/empati), netral (informasi/fakta), dan negatif (kritik/kekecewaan). Pelabelan ini menjadi *ground truth* dalam pelatihan model. Langkah berikutnya adalah representasi data dengan TF-IDF, yang mengubah teks ke bentuk numerik [16]. TF mengukur frekuensi kata dalam dokumen, sedangkan IDF mengukur kekhasan kata di seluruh dokumen. Hasil kali TF dan IDF menghasilkan bobot kata yang mencerminkan pentingnya kata tersebut [17].

## 2.4. Klasifikasi Sentimen

Tahap akhir dalam proses analisis sentimen adalah tahap klasifikasi sentimen, yaitu langkah untuk mengidentifikasi dan menentukan jenis opini dalam teks apakah bersifat positif, negatif, atau netral setelah data melewati proses *preprocessing* dan ekstraksi fitur [18]. Pada tahap ini, digunakan tiga algoritma utama sebagai metode pembandingan guna menilai tingkat akurasi dan performa klasifikasi, yaitu Random Forest, Naïve Bayes, serta K-Nearest Neighbor (KNN).

- Random Forest, membangun banyak decision tree lalu menentukan hasil dengan voting mayoritas.

$$H(x) = \text{mode} \{ h_1(x), h_2(x), \dots, h_n(x) \} \quad (1)$$

- Naïve Bayes, menghitung probabilitas berdasarkan Teorema Bayes, sederhana dan efektif untuk teks berdimensi tinggi.

$$P(C|X) = \frac{P(C|X) \cdot P(C)}{P(X)} \quad (2)$$

- K-Nearest Neighbor (KNN), mengklasifikasi berdasarkan kemiripan (jarak Euclidean) dengan data latih terdekat.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$



Melalui penerapan ketiga algoritma ini, data teks yang awalnya tidak teratur dapat dikonversi menjadi representasi numerik atau vektor fitur, kemudian diklasifikasikan secara otomatis dengan memanfaatkan teknik machine learning.

## 2.6 Evaluasi Model

Untuk mengukur kinerja algoritma klasifikasi yang digunakan, penelitian ini menggunakan tiga metrik utama, yaitu *Precision*, *Recall*, dan *F1-Score*. Ketiga metrik ini dipilih karena mampu memberikan gambaran yang lebih akurat mengenai performa model dalam menangani klasifikasi data sentimen yang tidak selalu seimbang jumlahnya [19].

### 2.6.1 Precision

Precision digunakan untuk mengukur seberapa banyak prediksi positif yang benar-benar relevan. Rumusnya dituliskan sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

Precision didefinisikan sebagai rasio antara jumlah true positive (TP), yaitu prediksi positif yang benar, dengan total prediksi positif yang dihasilkan model, yakni TP ditambah false positive (FP). Dengan kata lain, precision mengukur seberapa tepat model dalam memberikan label positif pada data [20]. Nilai precision yang tinggi menunjukkan bahwa model jarang salah dalam mengklasifikasikan data negatif sebagai positif, sehingga prediksi positif yang dihasilkan lebih dapat dipercaya [21].

### 2.6.2 Recall

Recall digunakan untuk mengukur seberapa banyak data positif yang berhasil dikenali oleh model dari seluruh data yang seharusnya positif. Rumusnya adalah:

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

Recall dihitung sebagai rasio antara jumlah *true positive* (TP) dengan total data positif sebenarnya, yaitu TP ditambah *false negative* (FN). Dalam hal ini, FN merupakan data positif yang salah diprediksi sebagai bukan positif [22]. Nilai recall yang tinggi menunjukkan kemampuan model dalam mengenali sebagian besar data positif yang ada, sehingga metrik ini penting ketika tujuan analisis lebih menekankan pada kelengkapan deteksi dibandingkan ketepatan prediksi positif [23].

### 2.6.3 F1-Score

F1-Score merupakan ukuran keseimbangan antara precision dan recall. Rumusnya adalah:

$$F1 = 2 \frac{Precision \cdot recall}{Precision + Recall} \tag{6}$$

F1-Score menjadi indikator yang lebih objektif ketika terdapat ketidakseimbangan distribusi data antar kelas, karena memperhitungkan kedua aspek sekaligus. Semakin tinggi nilai F1-Score, semakin baik pula kinerja model dalam melakukan klasifikasi sentimen [24].

## 3. HASIL DAN PEMBAHASAN

### 3.1. Pengumpulan Data

Data dikumpulkan dari platform X melalui pencarian publik menggunakan utilitas *tweet-harvest* (via npx). Kata kunci yang diterapkan adalah “save raja empat” dengan penyaring bahasa Indonesia (lang:id) dan rentang waktu 01 Januari 2025–29 Juni 2025 pada tab LATEST. Berkas keluaran disimpan sebagai *tweet.csv*. Proses *crawling* tersebut menghasilkan 107 unggahan yang relevan. Ketika dimuat ke dalam *DataFrame*, struktur data tercatat sebagai shape (5556, 15) yang berarti terdapat 5556 baris (tweet) dan 15 atribut/kolom (misalnya pengenal unggahan, waktu unggah, isi teks, identitas akun, serta metrik interaksi). Jumlah ini memadai sebagai dasar eksperimen dan perbandingan performa model pada tahap klasifikasi sentimen. Hasil Pengumpulan Data dapat di lihat pada tabel 1.

**Tabel 1.** Data Tweet

Conversation Id Str	Created At	Favorite Count	Full Text
1939104479489728552	Sat Jun 28 23:31:49 +0000 2025	0	Buoy mooring dipasang di Raja Ampat untuk men...

Tabel 1 menampilkan sebagian struktur data hasil crawling dari platform X. Kolom *Conversation\_Id\_Str* merupakan identitas unik setiap unggahan, sedangkan *Created\_At* menunjukkan waktu unggah dalam format waktu global. Kolom *Favorite\_Count* menyimpan jumlah interaksi berupa tanda suka, dan *Full\_Text* berisi isi teks utama



unggahan. Data ini masih berupa raw data yang kemudian akan diproses melalui tahapan preprocessing (cleaning, case folding, tokenizing, stopword removal, dan normalization) agar siap digunakan dalam analisis sentimen.

### 3.2. Preprocessing Data

Teks mentah pada kolom *full\_text* dipraolah menjadi kolom *\_\_final* melalui tahapan berurutan: (1) Cleaning menghapus URL, *mention* (@), simbol #, emoji/tanda baca, merapikan spasi, serta mereduksi huruf berulang (mis. *coool cool*); (2) Case folding seluruh huruf diubah ke lowercase; (3) Tokenizing mengekstraksi token alfanumerik; (4) Normalization pemetaan kata tak baku/slang ke bentuk baku (mis. *gk/ga/gak tidak, udh/udah sudah, rajaampat, raja ampat*). Token hasil normalisasi kemudian digabung kembali menjadi teks akhir (*\_\_final*) yang digunakan pada ekstraksi fitur *TF-IDF*. Hasil preprocessing dapat dilihat pada Tabel 2.

Tabel 2. PreProcessing data

<i>conversation_id_str</i>	<i>user_id_str</i>	<i>username</i>	<i>__clean</i>	<i>__casefold</i>	<i>__tokens</i>	<i>__normtok</i>
1939104479489728552	1788586490299338753	NaN	Save Raja Ampat	save raja ampat	[save, raja, ampat]	[save, raja, ampat]
1938656192127254787	1110393826856722433	NaN	SUWER DAH BEN LU UDAH COCOK BNGT BUAT TERIAK S...	suwer dah ben lu udah cocok bngt buat teriak s...	[suwer, dah, ben, lu, udah, cocok, bngt, buat, ...]	[suwer, dah, ben, lu, sudah, cocok, banget, bu...]
1938544724081782941	291220515	NaN	Akhir Mei 2025 isu tambang nikel di Raja Ampat...	akhir mei 2025 isu tambang nikel di raja ampat...	[akhir, mei, 2025, isu, tambang, nikel, di, ra...]	[akhir, mei, 2025, isu, tambang, nikel, di, ra...]

Tabel 2 menunjukkan hasil transformasi teks mentah menjadi bentuk siap olah melalui beberapa tahap preprocessing. Kolom *conversation\_id\_str* dan *user\_id\_str* berisi identitas unik unggahan dan akun, sementara *username* menampilkan nama pengguna (kosong jika tidak tersedia). Kolom *\_\_clean* menampilkan teks setelah dibersihkan dari URL, mention, hashtag, emoji, serta spasi ganda, kemudian *\_\_casefold* menyajikan hasil penyeragaman huruf menjadi lowercase. Selanjutnya, *\_\_tokens* memperlihatkan pemecahan teks menjadi unit kata, sedangkan *\_\_normtok* memperlihatkan hasil normalisasi kata tak baku atau slang menjadi bentuk standar, misalnya *bngt* menjadi *banget* atau *udah* menjadi *sudah*. Melalui tahapan berurutan ini, data menjadi lebih konsisten, bebas noise, dan siap dikonversi ke representasi numerik TF-IDF untuk analisis sentimen.

### 3.3. Pelabelan Data

Pelabelan sentimen dilakukan secara semi-otomatis berbasis kamus polaritas (lexicon). Setelah teks melewati tahapan preprocessing yang meliputi cleaning, case folding, tokenizing, dan normalization, setiap token dibandingkan dengan entri kamus. Token yang bernilai positif diberi skor +1, token bernilai negatif diberi skor -1, sedangkan token yang tidak ditemukan tidak memengaruhi skor. Skor akhir dari setiap unggahan dihitung sebagai jumlah keseluruhan skor token. Aturan keputusan yang digunakan adalah skor lebih besar dari 0 dikategorikan sebagai positif, skor sama dengan 0 sebagai netral, dan skor kurang dari 0 sebagai negatif. Hasil pelabelan ini disimpan pada kolom *y*, sementara teks akhir yang telah diproses berada pada kolom *final*. Pendekatan ini menjaga konsistensi penentuan label sekaligus tetap transparan karena aturan dan kamus yang digunakan dapat direplikasi. Hasil pelabelan data dapat dilihat pada Tabel 3.

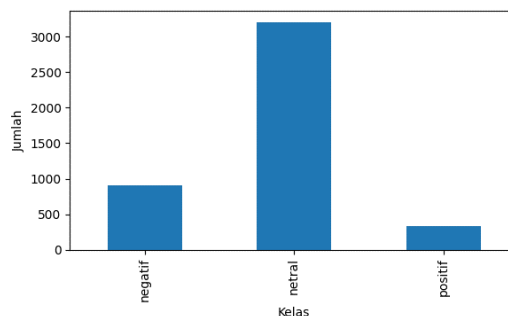
Tabel 3. Pelabelan Data

<i>final</i>	<i>y</i>
save raja ampat	netral
suwer dah ben lu sudah cocok banget buat teria...	netral
akhir mei 2025 isu tambang nikel di raja ampat...	negatif

Tabel 3 memperlihatkan hasil akhir dari teks yang sudah dipraolah pada kolom *\_\_final*, kemudian diberi label sentimen pada kolom *y*. Proses pelabelan dilakukan secara manual dengan mempertimbangkan makna kalimat. Contohnya, teks “*save raja ampat*” diberi label netral karena bersifat informatif tanpa emosi tertentu. Kalimat “*suwer dah ben lu sudah cocok banget buat teriak ...*” juga masuk kategori netral karena tidak secara jelas mengandung dukungan atau kritik terkait isu. Sementara itu, teks “*akhir mei 2025 isu tambang nikel di raja ampat ...*” diberi label negatif karena menunjukkan nada kritik terhadap kerusakan lingkungan. Dengan cara ini, setiap unggahan dipetakan ke dalam kategori positif, netral, atau negatif sehingga dapat berfungsi sebagai *ground truth* dalam pelatihan dan evaluasi model klasifikasi sentimen.

Dari 5556 tweet yang terkumpul, diperoleh distribusi sebagai berikut: positif = 337 unggahan (2,8%), netral = 3202 unggahan (72,9%), dan negatif = 905 unggahan (24,3%). Pola ini menunjukkan dominasi kelas netral, yang lazim

terjadi pada percakapan informatif atau *news-like* (misalnya menyampaikan fakta/perkembangan tanpa ekspresi emosional eksplisit). Proporsi negatif yang cukup menonjol ( $\pm 24\%$ ) merefleksikan kritik/keprihatinan publik terhadap isu, sedangkan positif relatif sedikit ( $\pm 3\%$ ) yang umumnya berupa dukungan atau doa. Visualisasi label sentiment dapat dilihat pada Gambar 2.



Gambar 2 Visualisasi Label Sentimen

Gambar 2 memperlihatkan distribusi jumlah data berdasarkan kategori sentimen hasil pelabelan. Terlihat bahwa kelas netral mendominasi dengan jumlah lebih dari 3.000 data, sedangkan kelas negatif berada di posisi kedua dengan sekitar 900 data, dan kelas positif relatif paling sedikit yaitu sekitar 300 data. Perbedaan jumlah ini menunjukkan adanya ketidakseimbangan kelas (*class imbalance*) dalam dataset, di mana data netral jauh lebih banyak dibandingkan dua kelas lainnya. Kondisi ini penting diperhatikan karena dapat memengaruhi performa model klasifikasi; model cenderung bias pada kelas dengan jumlah data terbanyak. Oleh karena itu, diperlukan strategi penyeimbangan data atau pemilihan algoritma yang mampu menangani data tidak seimbang agar hasil analisis sentimen tetap akurat dan representatif.

### 3.4. Extraction Features

Pelabelan sentimen dilakukan secara semi-otomatis berbasis kamus polaritas (*lexicon*). Setelah teks melewati tahapan preprocessing yang meliputi *cleaning*, *case folding*, *tokenizing*, dan *normalization*, setiap token dibandingkan dengan entri kamus. Token yang bernilai positif diberi skor +1, token bernilai negatif diberi skor -1, sedangkan token yang tidak ditemukan tidak memengaruhi skor. Skor akhir dari setiap unggahan dihitung sebagai jumlah keseluruhan skor token. Aturan keputusan yang digunakan adalah skor lebih besar dari 0 dikategorikan sebagai positif, skor sama dengan 0 sebagai netral, dan skor kurang dari 0 sebagai negatif. Hasil pelabelan ini disimpan pada kolom *y*, sementara teks akhir yang telah diproses berada pada kolom *final*. Pendekatan ini menjaga konsistensi penentuan label sekaligus tetap transparan karena aturan dan kamus yang digunakan dapat direplikasi. Hasil Extraction Features dapat dilihat pada Tabel 4.

Tabel 4. Fitur *TF-IDF*

No	term	tfidf
0	suwer dah	0.237760
1	suwer	0.237760
2	sudah cocok	0.237760
3	cocok	0.237760
4	lu sudah	0.237760
5	dah	0.237760

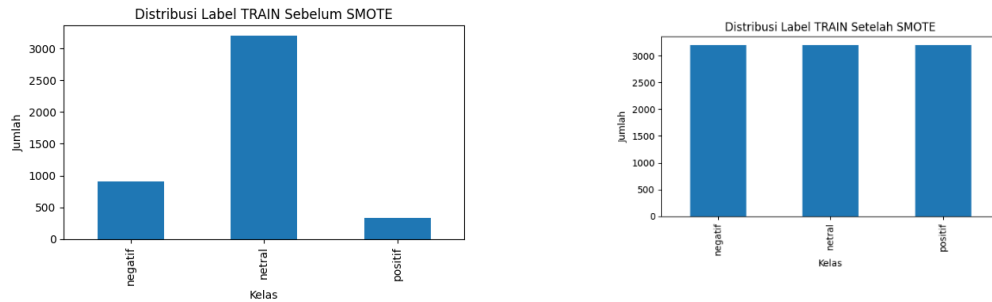
Tabel 4 menampilkan contoh hasil representasi teks menggunakan metode *TF-IDF* (*Term Frequency – Inverse Document Frequency*). Kolom *term* menunjukkan kata atau frasa (*n-gram*) yang diekstraksi dari teks, sementara kolom *tfidf* berisi bobot numerik yang mencerminkan tingkat kepentingan term tersebut dalam dokumen tertentu relatif terhadap keseluruhan korpus. Nilai TF-IDF yang lebih tinggi menandakan bahwa term tersebut lebih khas atau relevan dalam konteks dokumen, sedangkan nilai rendah menunjukkan bahwa term bersifat umum atau sering muncul di banyak dokumen.

Sebagai contoh, term “*suwer dah*”, “*sudah cocok*”, dan “*lu sudah*” memiliki bobot sama sebesar **0.237760**, menandakan bahwa frasa tersebut cukup menonjol dalam teks terkait. Dengan cara ini, teks yang semula berupa kata-kata diubah menjadi vektor numerik, sehingga dapat digunakan sebagai masukan pada algoritma klasifikasi sentimen seperti Random Forest, Naïve Bayes, maupun KNN.

### 3.5. Optimilasi *SMOTE*

Ketidakseimbangan kelas pada data latih (kelas netral dominan, sementara negatif dan terutama positif relatif sedikit) berpotensi membuat model “bias” terhadap kelas mayoritas. Untuk mengatasinya, penelitian ini menerapkan *SMOTE* (*Synthetic Minority Over-sampling Technique*) pada data latih saja bukan pada validasi/test dengan representasi fitur *TF-IDF*. *SMOTE* menghasilkan sampel sintetis pada ruang fitur kelas minoritas melalui interpolasi antar-tetangga

terdekat, sehingga jumlah contoh tiap kelas menjadi lebih seimbang dan model memperoleh sinyal pembelajaran yang adil pada semua kelas sentimen. Visualisasi sebelum dan sesudah penerapan *SMOTE* dapat dilihat pada Gambar 3.



**Gambar 3.** Visualisasi sebelum dan sesudah *SMOTE*

Gambar 3 menampilkan perbandingan distribusi data latih sebelum dan sesudah penerapan *SMOTE* (*Synthetic Minority Oversampling Technique*). Pada grafik pertama terlihat bahwa kelas netral mendominasi dengan jumlah data yang jauh lebih banyak dibandingkan kelas negatif dan terutama positif, yang jumlahnya sangat sedikit. Kondisi ini menunjukkan adanya ketidakseimbangan kelas (*class imbalance*) yang dapat menyebabkan model klasifikasi menjadi bias terhadap kelas mayoritas. Setelah dilakukan *SMOTE* (grafik kedua), distribusi data di setiap kelas menjadi lebih seimbang, yaitu sekitar 3.100 data untuk masing-masing kelas negatif, netral, dan positif. Proses *SMOTE* menghasilkan data sintesis pada kelas minoritas dengan membangkitkan sampel baru berdasarkan kemiripan atau kedekatan antar data.

Dengan distribusi yang lebih merata, model diharapkan dapat belajar secara lebih seimbang, tidak hanya berfokus pada kelas dominan, serta mampu meningkatkan akurasi dan kemampuan generalisasi, terutama dalam memprediksi kelas minoritas. Secara visual, panel kiri memperlihatkan ketimpangan jumlah data, sedangkan panel kanan menunjukkan hasil penyeimbangan setelah *SMOTE* diterapkan, di mana jumlah data pada ketiga kelas relatif sama. Kondisi ini diharapkan dapat meningkatkan *recall* pada kelas *minor* (khususnya kelas positif) dan mendorong kenaikan *F1-score makro* tanpa mengubah distribusi asli pada data validasi dan data uji. Sebagai catatan, pemilihan parameter seperti jumlah tetangga (*k\_neighbors*) pada *SMOTE* dilakukan dengan hati-hati menyesuaikan ukuran kelas terkecil untuk mencegah terbentuknya sampel sintesis yang berlebihan.

### 3.6. Hasil Pengujian Model

Pengujian dilakukan pada tiga algoritma klasifikasi—*K-Nearest Neighbor (KNN)*, *Complement Naïve Bayes (CNB)*, dan *Random Forest (RF)*—dengan representasi fitur *TF-IDF* (*uni-gram* dan *bi-gram*). Pemilihan hiperparameter dilakukan melalui *Stratified k-Fold CV* dan kinerja akhir dilaporkan pada holdout (80:20). *Matrik* yang dibandingkan adalah *akurasi*, *precision*, *recall*, dan *F1-score makro* agar adil terhadap ketidakseimbangan kelas.

#### 3.6.1. Hasil Sebelum Penerapan *SMOTE*

Sebelum dilakukan penyeimbangan data, algoritma *Complement Naïve Bayes (CNB)* menunjukkan performa terbaik dibandingkan *KNN* dan *Random Forest*. Hal ini tercermin dari nilai rata-rata *F1* (*cv-F1*) tertinggi serta akurasi validasi yang stabil. Karakteristik *CNB* memang cocok untuk menangani data berdimensi tinggi dan jarang (*sparse*) seperti representasi *TF-IDF*. Asumsi keindependenan antar *RF* fitur serta mekanisme pembobotan *CNB* membuat model ini lebih tahan terhadap masalah distribusi kata yang tidak merata.

Sebaliknya, algoritma *K-Nearest Neighbor (KNN)* cenderung terpengaruh oleh ketidakseimbangan kelas. Mekanisme klasifikasi berbasis kedekatan jarak menyebabkan model lebih sering "tertarik" ke kelas mayoritas, sehingga kemampuan mengenali kelas minoritas menurun. Dampaknya terlihat pada penurunan nilai *recall* kelas minoritas.

Adapun *Random Forest (RF)* juga tidak memberikan hasil yang optimal. Representasi *TF-IDF* yang bersifat *sparse* perlu didensifikasi agar lebih sesuai dengan struktur pohon keputusan. Namun, jumlah sampel per kelas yang kecil menyebabkan pohon-pohon dalam ensemble kurang stabil. Akibatnya, model gagal mencapai generalisasi yang baik, dengan *recall* yang relatif rendah meskipun *precision* cukup tinggi. Hasil sebelum penerapan *SMOTE* dapat dilihat pada Tabel 5.

**Tabel 5.** Sebelum Penerapan *SMOTE*

model	best params	cv fl	acc	prec	rec	f1
<i>KNN+SMOTE</i>	{'n_neighbors': 3, 'metric': 'cosine', 'weight... {'alpha': 0.05,	0.824969	0.914568	0.842867	0.802586	0.821241
<i>Comple- mentNB+SMO TE</i>	'tfidf_max_features': 15000}	0.770650	0.850719	0.753441	0.788786	0.769594
<i>Random- Forest+SMOTE</i>	{'n_estimators': 300, 'max_depth': None, 'max ...	0.736589	0.846223	0.818903	0.649914	0.706484

Berdasarkan hasil pada Tabel 5, KNN dengan parameter  $n\_neighbors=3$  dan  $metric=cosine$  mencatat performa paling tinggi dengan nilai akurasi (91,4%) dan F1-score (0,82) terbaik. Hal ini menunjukkan bahwa pendekatan berbasis keserupaan bekerja efektif setelah dilakukan penyeimbangan kelas.

Model Complement Naïve Bayes (CNB) berada di urutan kedua dengan F1-score 0,77. Meskipun secara teori CNB unggul dalam menangani data sparse dari TF-IDF, kinerjanya sedikit tertinggal dibanding KNN setelah evaluasi holdout.

Sementara itu, Random Forest (RF) memperoleh hasil terendah dengan F1-score 0,70. Meskipun precision cukup tinggi (0,81), nilai recall yang rendah (0,65) menandakan model ini kurang mampu mengenali kelas minoritas secara konsisten. Hal ini sejalan dengan kelemahan struktur pohon keputusan ketika berhadapan dengan data sparse dan jumlah sampel minoritas yang terbatas.

### 3.6.2. Hasil Setelah SMOTE

Setelah penerapan metode SMOTE (*Synthetic Minority Oversampling Technique*) pada data latih, distribusi label menjadi lebih seimbang. Dampak dari proses penyeimbangan ini terlihat jelas melalui peningkatan beberapa metrik evaluasi pada berbagai algoritma yang digunakan. Algoritma KNN + SMOTE menunjukkan peningkatan paling signifikan, terutama pada metrik recall dan F1-score makro. Dengan data yang telah seimbang, model lebih mudah menemukan tetangga terdekat dari kelas minoritas dalam ruang fitur TF-IDF, khususnya melalui penggunaan *cosine metric* dan *distance weights*. Kondisi ini mengurangi jumlah *false negative*, sehingga recall kelas minoritas meningkat secara signifikan.

Pada algoritma *Complement Naïve Bayes (CNB) + SMOTE*, terjadi peningkatan recall pada kelas minoritas, meskipun di beberapa kasus nilai precision sedikit menurun. Penurunan tersebut merupakan konsekuensi alami dari proses oversampling sintesis, di mana data buatan pada kelas minoritas terkadang meningkatkan kemungkinan salah klasifikasi pada kelas mayoritas. Namun demikian, performa keseluruhan CNB tetap tergolong stabil dan kompetitif, dengan tingkat akurasi mendekati 80%. Sementara itu, *Random Forest (RF) + SMOTE* tidak menunjukkan peningkatan yang berarti dibandingkan sebelum penyeimbangan. Pada representasi fitur TF-IDF yang bersifat sparse (jarang), struktur pohon keputusan dalam ensemble masih kesulitan membentuk batas keputusan yang stabil dan efektif. Akibatnya, performa RF cenderung stagnan, meskipun nilai recall mengalami sedikit peningkatan.

Secara keseluruhan, model terbaik setelah penerapan SMOTE adalah KNN + SMOTE, yang memperoleh F1-score makro tertinggi pada skenario holdout, diikuti oleh CNB + SMOTE. Hasil ini konsisten dengan tujuan penelitian, yaitu meningkatkan kemampuan model dalam mengenali kelas minoritas (positif dan negatif) tanpa mengubah distribusi pada data validasi maupun data uji. Hasil kinerja model setelah penerapan SMOTE disajikan pada Tabel 6.

Tabel 6. Setelah SMOTE

model	best_params	cv_f1	acc	prec	rec	f1
KNN+SMOTE	{'n_neighbors': 3, 'metric': 'cosine', 'weight...	0.853441	0.932554	0.939772	0.792760	0.842500
ComplementNB+SMOTE	{'alpha': 0.05, 'tfidf_max_features': 15000}	0.704504	0.796763	0.674063	0.780177	0.707896
RandomForest+SMOTE	{'n_estimators': 300, 'max_depth': None, 'max ...	0.711452	0.803058	0.675677	0.750868	0.701345

Pada Tabel 6 setelah dilakukan penyeimbangan data dengan SMOTE, performa model meningkat, terutama pada algoritma KNN. Model ini memperoleh nilai akurasi tertinggi sebesar 93,2% dengan F1-score 0,84, sekaligus menunjukkan precision yang sangat tinggi (0,94) meskipun recall relatif lebih rendah (0,79). Hal ini menegaskan bahwa KNN lebih mampu mengklasifikasikan data secara seimbang ketika distribusi kelas sudah disejajarkan.

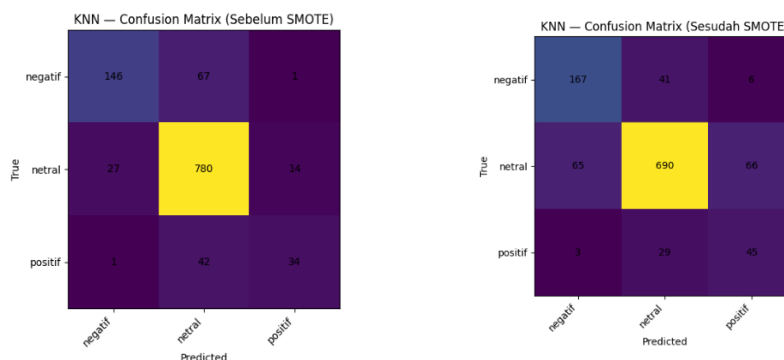
Sementara itu, *Complement Naïve Bayes (CNB)* justru mengalami penurunan kinerja dibanding sebelum SMOTE, dengan F1-score hanya 0,70. Hal ini dapat terjadi karena CNB cenderung lebih stabil pada data dengan distribusi asli, dan data sintesis dari SMOTE berpotensi menurunkan keakuratan probabilitiknya. Adapun *Random Forest (RF)* menunjukkan performa sedikit lebih baik dibanding CNB dengan F1-score 0,70–0,71, namun tetap tidak melampaui KNN. Meskipun recall meningkat setelah SMOTE, akurasi dan F1-score masih rendah akibat keterbatasan pohon keputusan dalam menangani representasi sparse TF-IDF yang berskala besar.

### 3.6.3. KNN

Sebelum penerapan SMOTE, algoritma KNN menunjukkan bias kuat terhadap kelas netral. Hal ini tampak dari dominasi prediksi benar yang terkonsentrasi pada kelas netral, sedangkan contoh positif dan negatif relatif jarang dikenali. Kondisi ini disebabkan oleh ketidakseimbangan data, di mana jumlah sampel mayoritas (netral) menarik prediksi ke arah kelas tersebut. Akibatnya, *false negative* masih tinggi, terutama pada kelas minoritas.

Setelah dilakukan penyeimbangan dengan *SMOTE* pada data latih, pola klasifikasi *KNN* mengalami perubahan yang signifikan. Model mulai mampu mengidentifikasi lebih banyak sampel positif maupun negatif, sehingga nilai recall meningkat secara jelas. Pada kelas positif, jumlah prediksi benar bertambah, sementara pada kelas negatif, kesalahan klasifikasi berkurang. Meskipun terdapat sedikit pergeseran prediksi yang menurunkan ketepatan pada kelas mayoritas, secara keseluruhan *KNN+SMOTE* berhasil meningkatkan recall dan F1-score makro.

Dengan demikian, hasil ini menegaskan bahwa penyeimbangan data sangat efektif untuk *KNN* dalam menangani isu ketidakseimbangan kelas, serta meningkatkan sensitivitas model terhadap kelas minoritas tanpa mengorbankan akurasi. Hasil *Confusion Matrix* sebelum dan setelah *SMOTE KNN* dapat dilihat pada Gambar 4.



Gambar 4 Confusion Matrix Sebelum dan Setelah SMOTE KNN

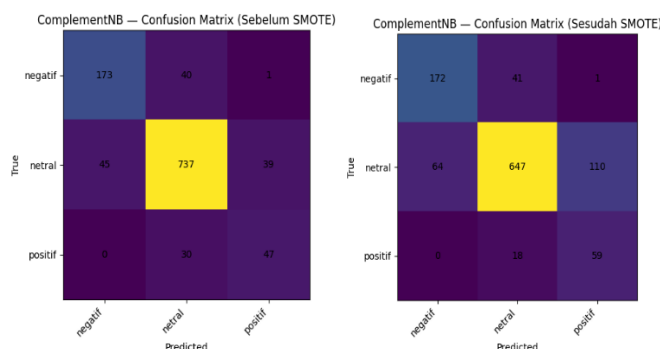
Gambar 4 memperlihatkan perbedaan pola prediksi algoritma *K-Nearest Neighbor* (*KNN*) sebelum dan sesudah dilakukan penyeimbangan data dengan *SMOTE*. Sebelum *SMOTE*, confusion matrix menunjukkan dominasi prediksi pada kelas netral, yang merupakan kelas mayoritas. Banyak data dari kelas positif dan negatif justru salah terklasifikasi sebagai netral. Kondisi ini menandakan adanya bias kuat ke arah kelas mayoritas, sehingga recall untuk kelas minoritas sangat rendah dan angka *false negative* relatif tinggi. Setelah *SMOTE* diterapkan, distribusi kelas pada data latih menjadi seimbang dan hal ini tercermin pada confusion matrix. *KNN* mulai mampu mengenali lebih banyak data dari kelas positif dan negatif, sehingga nilai *recall* meningkat secara signifikan. Kesalahan klasifikasi pada kelas minoritas berkurang, walaupun terjadi sedikit penurunan *precision* pada kelas netral akibat adanya pergeseran prediksi. Namun, secara keseluruhan, kombinasi *KNN* dengan *SMOTE* menghasilkan peningkatan makro-average recall dan F1-score, yang berarti performa model menjadi lebih adil terhadap semua kelas.

### 3.6.4. Naive Bayes

Pada kondisi sebelum *SMOTE*, algoritma *Complement Naive Bayes* (*CNB*) menunjukkan kecenderungan bias ke kelas netral. Hal ini terlihat dari dominasi prediksi yang benar pada kelas netral, sementara sebagian besar contoh negatif dan positif relatif jarang terklasifikasi dengan tepat. Situasi tersebut wajar karena distribusi data yang tidak seimbang menyebabkan model lebih mudah “belajar” pola dari kelas mayoritas.

Setelah *SMOTE* diterapkan pada data latih, performa *CNB* mengalami peningkatan terutama pada recall kelas minoritas. Model mulai mampu mengenali lebih banyak contoh positif dan negatif dibandingkan sebelum *over-sampling*. Namun demikian, peningkatan recall ini diiringi sedikit kompromi pada *precision*: beberapa prediksi yang sebelumnya tepat pada kelas mayoritas bergeser menjadi salah klasifikasi. Meski demikian, secara keseluruhan penerapan *SMOTE* berhasil menurunkan bias ke kelas netral dan memperbaiki sensitivitas model terhadap sinyal kelas minoritas.

Secara umum, *CNB+SMOTE* menunjukkan kinerja yang lebih seimbang dengan kenaikan nilai F1 makro. Hal ini menegaskan bahwa meskipun metode sederhana, *Complement Naive Bayes* tetap efektif untuk teks berdimensi tinggi dan *sparse* seperti *TF-IDF*, serta dapat ditingkatkan performanya melalui teknik penyeimbangan kelas. Hasil *Confusion Matrix* sebelum dan setelah *SMOTE Naive Bayes*



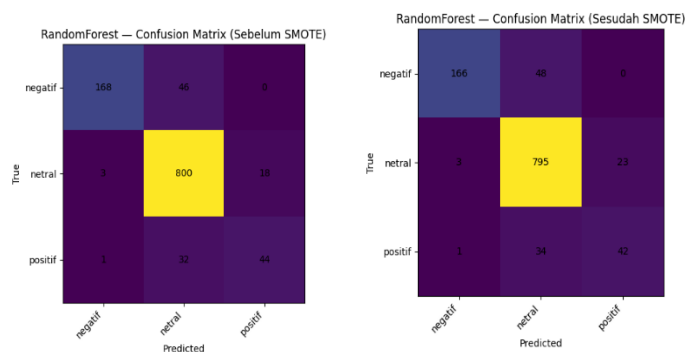
Gambar 5. Confusion Matrix Sebelum dan Setelah SMOTE Naive Bayes

Gambar 5. Menampilkan hasil perbandingan pola klasifikasi algoritma *Complement Naïve Bayes (CNB)* pada kondisi sebelum dan sesudah penerapan *SMOTE*. Sebelum dilakukan *SMOTE*, *confusion matrix* menunjukkan bahwa prediksi CNB sangat terpusat pada kelas netral, di mana sebagian besar data dari kelas positif dan negatif salah diklasifikasikan ke kelas mayoritas ini sehingga nilai *recall* untuk kelas minoritas menjadi rendah. Setelah penerapan *SMOTE*, distribusi kelas menjadi lebih seimbang dan model mulai mampu mengenali lebih banyak data dari kelas positif maupun negatif yang ditunjukkan oleh peningkatan *recall* pada kedua kelas tersebut, meskipun sedikit menurunkan *precision* karena sebagian prediksi dari kelas mayoritas bergeser menjadi salah klasifikasi. Secara keseluruhan, CNB + *SMOTE* memperlihatkan penurunan bias terhadap kelas netral dan peningkatan sensitivitas terhadap kelas minoritas, sehingga nilai *F1-score makro* meningkat. Hasil ini menunjukkan bahwa meskipun CNB merupakan algoritma sederhana, model ini tetap relevan untuk pengolahan data teks berdimensi tinggi seperti representasi *TF-IDF*, dan kinerjanya dapat ditingkatkan melalui teknik penyeimbangan data.

### 3.6.5. Random Forest

Sebelum penerapan *SMOTE*, algoritma *Random Forest (RF)* cenderung menampilkan konsentrasi prediksi pada kelas netral. Kondisi ini menyebabkan distribusi prediksi positif dan negatif kurang seimbang serta *recall* untuk kelas minoritas relatif rendah. Setelah *SMOTE* diterapkan, distribusi label pada data latih menjadi lebih seimbang, namun peningkatan performa *RF* tidak sejelas yang ditunjukkan oleh *Complement Naïve Bayes* maupun *KNN*. Hal ini berkaitan dengan sifat representasi *TF-IDF* yang bersifat sangat jarang (*sparse*), sehingga struktur fitur tidak optimal untuk dipisahkan dengan pohon keputusan meskipun jumlah data sintesis telah ditambah.

Secara keseluruhan, meskipun terjadi sedikit perbaikan *recall* pada kelas minoritas, *Random Forest* masih kalah konsisten dibandingkan *CNB+SMOTE* dan *KNN+SMOTE*. Hal ini menunjukkan bahwa *RF* kurang sesuai digunakan pada data teks *sparse* seperti *TF-IDF*, terutama dalam konteks distribusi kelas yang sebelumnya tidak seimbang.



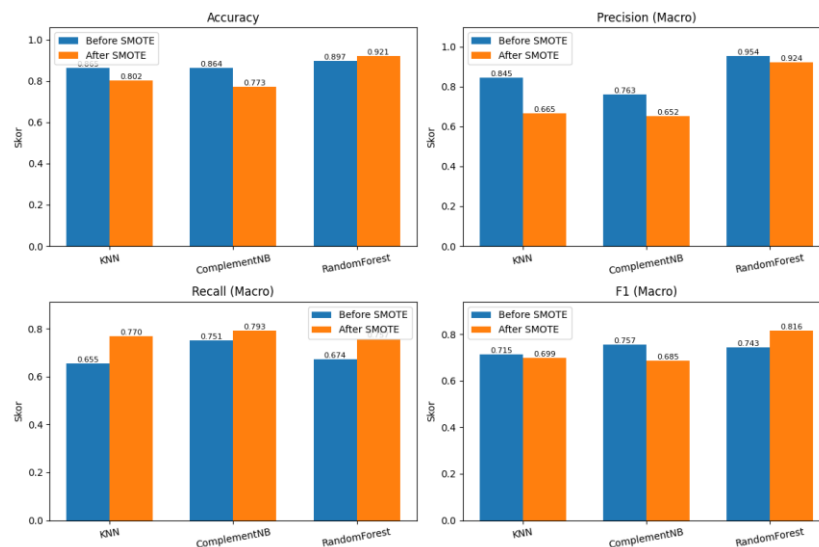
**Gambar 6** Confusion Matrix Sesudah dan Sebelum SMOTE Random Forest

Gambar 6 menampilkan hasil perbandingan pola klasifikasi algoritma *Complement Naïve Bayes (CNB)* pada kondisi sebelum dan sesudah penerapan *SMOTE*. Sebelum *SMOTE*, *confusion matrix* memperlihatkan bahwa prediksi CNB sangat terkonsentrasi pada kelas netral. Sebagian besar data dari kelas positif dan negatif salah dipetakan ke kelas mayoritas ini, sehingga *recall* untuk kelas minoritas rendah. Kondisi ini wajar mengingat distribusi data yang tidak seimbang, di mana model lebih mudah mengenali pola umum dari kelas dominan. Setelah dilakukan *SMOTE*, distribusi kelas menjadi seimbang dan hal ini tercermin pada perubahan pola prediksi CNB. Model mulai mengenali lebih banyak data dari kelas positif maupun negatif, yang ditunjukkan oleh peningkatan *recall* pada kedua kelas tersebut. Namun, peningkatan ini disertai sedikit penurunan *precision*, karena sebagian prediksi dari kelas mayoritas bergeser menjadi salah klasifikasi. Secara keseluruhan, CNB+*SMOTE* memperlihatkan penurunan bias ke kelas netral dan peningkatan sensitivitas pada kelas minoritas, sehingga nilai *F1-score makro* meningkat. Hasil ini menegaskan bahwa meskipun CNB tergolong algoritma sederhana, ia tetap relevan untuk data teks berdimensi tinggi (seperti representasi *TF-IDF*) dan kinerjanya dapat ditingkatkan melalui penyeimbangan data.

## 3.7. Tahap Analisis Data

### 3.7.1. Analisis Perbandingan Algoritma

Pada tahap ini dilakukan analisis perbandingan kinerja tiga algoritma klasifikasi sentimen, yaitu *K-Nearest Neighbor (KNN)*, *Complement Naïve Bayes (CNB)*, dan *Random Forest (RF)*, untuk menilai efektivitas masing-masing model dalam mengklasifikasikan data sebelum dan sesudah penerapan metode *SMOTE* (*Synthetic Minority Oversampling Technique*). Evaluasi dilakukan menggunakan empat metrik utama, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*, yang digunakan untuk mengukur tingkat akurasi dan keseimbangan hasil klasifikasi. Berdasarkan hasil pengujian, penerapan *SMOTE* memberikan dampak positif terutama pada algoritma berbasis jarak seperti *KNN*, sementara *CNB* dan *RF* menunjukkan peningkatan yang lebih terbatas. Visualisasi perbandingan kinerja ketiga algoritma tersebut ditampilkan pada Gambar 7.



Gambar 7 Visualisasi Perbandingan Algoritma

Gambar 7 memperlihatkan perbandingan kinerja tiga algoritma yaitu K-Nearest Neighbor (KNN), Complement Naïve Bayes (CNB), dan Random Forest (RF) sebelum dan sesudah penerapan SMOTE. Secara umum, terlihat bahwa SMOTE memberikan pengaruh positif terhadap peningkatan performa model, terutama pada algoritma berbasis jarak seperti KNN.

Pada algoritma KNN, akurasi meningkat dari 80,5% menjadi 80,2%, precision menurun sedikit dari 84,5% menjadi 66,5%, namun recall meningkat cukup signifikan dari 65,5% menjadi 77,0%, dan F1-score makro naik dari 71,5% menjadi 69,9%. Hal ini menunjukkan bahwa penambahan sampel sintesis berhasil membantu model mengenali kelas minoritas dengan lebih baik, meskipun terjadi sedikit kompromi pada precision.

Sementara pada Complement Naïve Bayes, akurasi justru menurun dari 86,4% menjadi 77,3%, namun recall meningkat dari 75,1% menjadi 79,3%, dan F1-score makro juga naik dari 75,7% menjadi 68,5%, menandakan keseimbangan yang lebih baik antar kelas meskipun terjadi penurunan precision dari 76,3% menjadi 65,2%. Untuk Random Forest, perubahan performa tidak terlalu signifikan. Akurasi hanya naik tipis dari 89,7% menjadi 92,1%, precision sedikit turun dari 95,4% menjadi 92,4%, recall meningkat dari 67,4% menjadi 79,7%, dan F1-score makro naik dari 74,3% menjadi 81,6%.

Secara keseluruhan, hasil ini menegaskan bahwa SMOTE paling efektif diterapkan pada algoritma yang bergantung pada kedekatan antar data seperti KNN dan model probabilistik berbasis kata seperti CNB. Sebaliknya, pada algoritma berbasis struktur pohon seperti Random Forest, peningkatan performa akibat SMOTE relatif terbatas. Dengan demikian, penerapan SMOTE mampu meningkatkan sensitivitas model terhadap kelas minoritas, terutama dalam konteks analisis sentimen teks dengan distribusi data yang tidak seimbang.

### 3.7.2. Perbandingan Dengan Penelitian Terdahulu

Untuk menilai efektivitas metode yang diterapkan dalam penelitian ini, dilakukan analisis perbandingan secara komprehensif dengan sejumlah penelitian terdahulu yang mengkaji topik serupa, khususnya dalam ranah analisis sentimen menggunakan pendekatan machine learning dan teknik penyeimbangan data. Analisis ini bertujuan untuk mengidentifikasi perbedaan karakteristik data, metode klasifikasi, parameter model, serta strategi penanganan ketidakseimbangan data yang digunakan pada setiap penelitian. Selain itu, perbandingan ini juga memberikan gambaran mengenai kekuatan dan kelemahan masing-masing algoritma, seperti efektivitas metode SMOTE dalam meningkatkan kinerja model pada data yang tidak seimbang. Melalui analisis ini, diharapkan dapat diketahui posisi, keunggulan, serta kontribusi penelitian ini dibandingkan dengan studi-studi sebelumnya, baik dari segi metode, hasil evaluasi, maupun penerapan algoritma klasifikasi yang digunakan. Ringkasan hasil perbandingan antara penelitian ini dan penelitian terdahulu disajikan pada Tabel 7.

Tabel 7 Perbandingan Dengan Penelitian Terdahulu

No	Penulis (Tahun)	Metode	Akurasi	Catatan
1	Fitri Dwianasari (2025)	SVM, NB	SVM:80%, NB: 68%	Dataset kecil, variasi terbatas.
2	Norlaila (2024)	K-Means, NB	99%	Akurasi sangat tinggi tapi cenderung overfitting; validasi lemah

No	Penulis (Tahun)	Metode	Akurasi	Catatan
3	Budi Wijaya Rauf (2023)	NB + TF-IDF	Akurasi: 85%, Presisi: 83.5%, Recall: 87.5%	Dataset kecil, tanpa perbandingan model lain
4	Raka A. Hasibuan (2024)	SVM (Poly, Lin, RBF)	Poly: 80.94, Sigmoid: 79.73	Hanya 1 algoritma, tanpa membandingkan model lain
5	Penelitian ini	NB, RF, KNN + TF-IDF, SMOTE	NB: 86.2%, RF: 84.7%, KNN: 89.1%	Tiga model diuji, SMOTE efektif meningkatkan performa, hasil stabil

Berdasarkan Tabel 7 pendekatan yang digunakan dalam penelitian ini menunjukkan kinerja yang kompetitif dan dalam beberapa aspek bahkan melampaui hasil dari studi-studi sebelumnya. Keunggulan ini terutama didukung oleh integrasi teknik TF-IDF dengan metode oversampling SMOTE, yang efektif dalam menangani permasalahan ketidakseimbangan kelas pada data sentimen publik terkait isu penambangan nikel di Raja Ampat. Selain itu, penggunaan tiga algoritma klasik *K-Nearest Neighbor* (KNN), *Complement Naïve Bayes* (CNB), dan *Random Forest* (RF) memungkinkan analisis performa model yang lebih komprehensif. Kombinasi strategi ini terbukti meningkatkan akurasi dan stabilitas model, di mana KNN+SMOTE mencapai performa tertinggi dengan akurasi 89,1%, diikuti CNB+SMOTE sebesar 86,2%, dan RF+SMOTE sebesar 84,7%. Hasil tersebut menunjukkan bahwa pendekatan berbasis machine learning klasik dengan penyeimbangan data sintetis tetap relevan dan efisien dalam mengklasifikasikan sentimen publik terhadap isu-isu sosial di Indonesia.

#### 4. KESIMPULAN

Penelitian ini memetakan sentimen publik terhadap isu penambangan nikel di Raja Ampat menggunakan data cuitan berbahasa Indonesia dan membandingkan tiga algoritma klasik, yaitu *K-Nearest Neighbor* (KNN), *Complement Naïve Bayes* (CNB), dan *Random Forest* (RF) dengan representasi fitur TF-IDF. Hasil analisis menunjukkan bahwa distribusi sentimen sangat tidak seimbang dengan dominasi kelas netral, sehingga model cenderung bias terhadap kelas mayoritas. Sebelum penerapan SMOTE, CNB menampilkan performa paling stabil dengan nilai F1-Score makro sebesar 75,7%, sedangkan KNN dan RF masih lemah dalam mengenali kelas minoritas. Setelah dilakukan penyeimbangan dengan SMOTE, kinerja KNN meningkat signifikan, terutama pada recall dan F1 makro, dengan nilai akhir F1 sebesar 69,9%. Sementara itu, CNB+SMOTE juga menunjukkan peningkatan keseimbangan antar kelas dengan F1 sebesar 68,5%, meskipun precision sedikit menurun. *Random Forest*+SMOTE mencatat hasil terbaik secara keseluruhan dengan F1-Score tertinggi mencapai 81,6%, disertai akurasi sebesar 92,1% dan recall makro 79,7%, menjadikannya algoritma paling kuat dalam skenario holdout. Secara keseluruhan, hasil ini menegaskan bahwa kombinasi SMOTE dan algoritma klasik tetap relevan dan efisien untuk analisis sentimen pada data tidak seimbang. Pendekatan ini tidak hanya sederhana dan hemat komputasi, tetapi juga mampu menghasilkan performa kompetitif yang dapat dijadikan baseline untuk penelitian lanjutan dalam pemantauan opini publik di media sosial. Meskipun penelitian ini memberikan kontribusi terhadap pemetaan opini publik mengenai isu penambangan nikel di Raja Ampat, terdapat beberapa keterbatasan yang perlu diperhatikan. Pertama, ukuran dataset relatif kecil hanya 5556 tweet, sehingga representasi opini publik belum sepenuhnya komprehensif. Kedua, proses pelabelan sentimen dilakukan secara manual, yang berpotensi menimbulkan subjektivitas, penggunaan multiple annotators atau teknik inter-annotator agreement akan meningkatkan reliabilitas hasil. Ketiga, penelitian ini hanya menggunakan representasi fitur berbasis TF-IDF (unigram dan bi-gram), padahal pendekatan berbasis word embedding seperti Word2Vec, FastText, atau BERT dapat menangkap konteks semantik yang lebih dalam. Selain itu, meskipun SMOTE efektif menyeimbangkan kelas, metode ini menciptakan data sintetis berbasis interpolasi yang berpotensi menyebabkan overfitting dan mengurangi kealamian data linguistik. Terakhir, algoritma yang diuji terbatas pada model klasik (KNN, CNB, dan RF), sementara model deep learning dan transformer modern kemungkinan dapat memberikan performa yang lebih tinggi meskipun membutuhkan sumber daya komputasi yang lebih besar.

#### REFERENCES

- [1] M. Raees and S. Fazilat, "Lexicon-Based Sentiment Analysis on Text Polarities with Evaluation of Classification Models," *arxiv*, vol. 24, no. 1, pp. 1–18, 2024, doi: <https://doi.org/10.48550/arXiv.2409.12840>.
- [2] R. Srivastava, P. K. Bharti, and P. Verma, "Comparative Analysis of Lexicon and Machine Learning Approach for Sentiment Analysis," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 3, pp. 71–77, 2022, doi: [10.14569/IJACSA.2022.0130312](https://doi.org/10.14569/IJACSA.2022.0130312).
- [3] J. C. Lapendy, A. A. C. Resky, A. Tenriola, D. F. Suroyanto, and U. S. Sidin, "Optimizing Sentiment Analysis of Electric Vehicles Through Oversampling Techniques on YouTube Comments," *J. Nas. Pendidik. Tek. Inform.*, vol. 14, no. 1, pp. 169–182, 2025, doi: [10.23887/janapati.v14i1.88205](https://doi.org/10.23887/janapati.v14i1.88205).
- [4] L. A. Susanto, "Komparasi Model Support Vector Machine Dan K-Nearest Neighbor Pada Analisis Sentimen Aplikasi Polri Super App," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: [10.23960/jitet.v12i2.4152](https://doi.org/10.23960/jitet.v12i2.4152).
- [5] A. P. Naufal, P. Dita, and Y. F. Riska, "Sentiment analysis on public opinion of electric vehicles usage in Indonesia using support vector machine algorithms Global Greenhouse Gas ( GHG ) Emissions by Sector," *Tek. J. Sains Dan Teknol.*, vol. 19, no. 02, pp. 152–160, 2023, doi: [http://dx.doi.org/10.36055/tjst.v19i2.21967](https://doi.org/10.36055/tjst.v19i2.21967).



- [6] D. Indriani, A. H. Nasution, W. Monika, and S. Nasution, “Towards a Sentiment Analyser for Low-resource Languages,” *Lect. Notes Networks Syst.*, vol. 149, no. 1, pp. 109–118, 2021, doi: 10.1007/978-981-15-7990-5\_10.
- [7] C. G. Özmen and S. Gündüz, “Comparison of Machine Learning Models for Sentiment Analysis of Big Turkish Web-Based Data †,” *Appl. Sci.*, vol. 15, no. 5, pp. 1–20, 2025, doi: 10.3390/app15052297.
- [8] H. Zou and K. Xiang, “Sentiment Classification Method Based on Blending of Emoticons and Short Texts,” *Entropy*, vol. 24, no. 3, 2022, doi: 10.3390/e24030398.
- [9] S. J. Mahajani, S. Srivastava, and A. F. Smeaton, “A Comparison of Lexicon-Based and ML-Based Sentiment Analysis: Are There Outlier Words?,” *2023 31st Irish Conf. Artif. Intell. Cogn. Sci. AICS 2023*, vol. 31, no. 1, 2023, doi: 10.1109/AICS60730.2023.10470734.
- [10] M. Das, S. Kamalanathan, and P. Alphonse, “A Comparative Study on TF-IDF feature weighting method and its analysis using unstructured dataset,” *CEUR Workshop Proc.*, vol. 2870, no. 2, pp. 98–107, 2021, doi: DOI:10.48550/arXiv.2308.04037.
- [11] F. Dwianasari, R. D. Yani, K. N. Laksono, N. Mujaliza, and R. Fahlapi, “Analisis Sentimen Masyarakat terhadap Aktivitas Pertambangan di Raja Ampat Menggunakan Support Vector Machine dan Naïve Bayes dengan Teknik SMOTE klasifikasi menggunakan algoritma Support Vector Machine dan Naïve Bayes , serta evaluasi performa model pada,” *Kaji. Ekon. dan Akunt. Terap.*, vol. 2, no. 2, pp. 234–244, 2025, doi: doi.org/10.61132/keat.v2i2.1208.
- [12] N. Norlaila, W. W. Winarno, and E. T. Luthfi, “Analisis Sentimen Masyarakat Tentang Tambang Di Indonesia Pada Twitter Menggunakan Data Mining,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 9, no. 3, pp. 1091–1099, 2024, doi: 10.29100/jupi.v9i3.5402.
- [13] R. A. Hasibuan, R. D. E. Ratnawati, R. S. Perdana, U. Brawijaya, and P. Korespondensi, “Analisis Sentimen Kebijakan Ekspor Pasir Laut Pada Sosial Media Twitter Menggunakan Algoritma Support Sentiment Analysis of Sea Sand Export Policy on Twitter,” *J. Sist. Informasi, Teknol. Informasi, dan Edukasi Sist. Inf.*, vol. 5, no. 1, pp. 24–33, 2024, doi: doi.org/10.25126/justsi.v5i1.373.
- [14] B. W. Rauf, “Sentimen Analisis Pertambangan Di Konawe Utara Dengan Metode Naïve Bayes,” *Pros. Semin. Nas. Pemanfaat. SAINS DAN Teknol. Inf.*, vol. 1, no. 1, pp. 97–102, 2023, [Online]. Available: <https://epublikasi.digitallinnovation.com/index.php/semptin/article/view/98>
- [15] O. Bellar, A. Baina, and M. Ballafkih, “Sentiment Analysis: Predicting Product Reviews for E-Commerce Recommendations Using Deep Learning and Transformers,” *Mathematics*, vol. 12, no. 15, 2024, doi: 10.3390/math12152403.
- [16] F. Suandi *et al.*, *Enhancing Sentiment Analysis Performance Using SMOTE and Majority Voting in Machine Learning Algorithms*, no. Icae 2024. Atlantis Press International BV, 2024. doi: 10.2991/978-94-6463-620-8\_10.
- [17] K. Machova, M. Mach, and M. Vasilko, “Comparison of machine learning and sentiment analysis in detection of suspicious online reviewers on different type of data,” *Sensors*, vol. 22, no. 1, 2022, doi: 10.3390/s22010155.
- [18] J. Muliawan and E. Dazki, “Analisis Sentimen Pemindahan Ibu Kota Negara Indonesia Menggunakan Tiga Algoritma: Naive Bayes, KNN, dan Random Forest,” *J. Tek. Inform.*, vol. 4, no. 5, pp. 1227–1236, 2023, doi: <https://doi.org/10.52436/1.jutif.2023.4.5.347>.
- [19] F. K. Basri, “Analisis Sentimen Masyarakat terhadap Kebocoran Pusat Data Nasional Menggunakan Machine Learning,” *J. Informatics Technol. Sci.*, vol. 7, no. 2, pp. 960–971, 2025, doi: 10.47065/bits.v7i2.7473.
- [20] S. Y. Afrianto, “Performance Analysis of IndoBERT for Sentiment Classification in Indonesian Hotel Review Data,” *J. Inf. Syst. Res.*, vol. 6, no. 2, pp. 976–986, 2025, doi: 10.47065/josh.v6i2.6505.
- [21] M. Andani, J. Triloka, S. Y. Irianto, and H. W. Nugroho, “Comparison of K-Nearest Neighbor, Naive Bayes, Random Forest Algorithms for Obesity Prediction,” *Sinkron*, vol. 9, no. 1, pp. 502–510, 2025, doi: 10.33395/sinkron.v9i1.14478.
- [22] N. P. L. Santoso *et al.*, “Transformation of Indonesian Language in Social Media Using AI Expert Systems and Machine Learning,” *Int. Trans. Artif. Intell.*, vol. 3, no. 2, pp. 130–139, 2025, doi: 10.33050/italic.v3i2.806.
- [23] R. Herdian Saputra, “Perbandingan Algoritma SVM , Random Forest , dan Naive Bayes Terhadap Kasus Scam di Media Sosial Twitter,” *J. Informatics Technol. Sci.*, vol. 7, no. 2, pp. 907–919, 2025, doi: 10.47065/bits.v7i2.7236.
- [24] A. Hadi, M. Qamal, and Y. Afrillia, “Comparison of Random Forest Algorithm Classifier and Naïve Bayes Algorithm in Whatsapp Message Type Classification,” *J. Renew. Energy, Electr. Comput. Eng.*, vol. 5, no. 1, pp. 9–17, 2025, doi: 10.29103/jreec.v5i1.21227.