JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XIII No 2, Desember 2024
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

ANALISIS SENTIMEN APLIKASI TIKTOK MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE* (SVM) DAN *RANDOM FOREST*

Riyanto Tangke[1)], Deiby Tineke Salaki[2)], Wisard Widsli Kalengkongan[3)], Eliasta Ketaren[4)]
Sistem Informasi
Univeristas Sam Ratulangi
JL. Kampus Unsrat Bahu, Kleak, Malalayang, Kota Manado, Sulawesi Utara
email: riyantotangke106@student.unsrat.ac.id[1)], deibyts.mat@unsrat.ac.id[2)],
wisard.kalengkongan@unsrat.ac.id[3)], eliasketaren@unsrat.ac.id[4)]

Abstrak
Pandemi Covid-19 mengakibatkan peningkatan penggunaan media sosial, termasuk TikTok, yang mendapatkan berbagai ulasan dari pengguna. Penelitian ini bertujuan untuk menganalisis sentimen ulasan TikTok menggunakan algoritma Support Vector Machine (SVM) dan Random Forest. Data diambil dari Google Play Store dan diproses dengan TF-IDF untuk pembobotan kata serta SMOTE untuk menangani ketidakseimbangan data. Hasil penelitian menunjukkan bahwa algoritma Random Forest mencapai akurasi 93% dan SVM 90%. Meskipun SVM menunjukkan kinerja yang baik pada data dengan margin kelas yang jelas, Random Forest lebih stabil dalam menangani variasi data dan lebih tahan terhadap overfitting. Oleh karena itu, Random Forest lebih cocok untuk analisis sentimen ulasan TikTok pada dataset yang besar dengan ketersediaan sumber daya komputasi yang memadai. Penelitian ini memberikan wawasan berharga bagi pengembang aplikasi dan pemangku kepentingan untuk meningkatkan kualitas aplikasi berdasarkan ulasan dari pengguna.
Kata Kunci: Tiktok, Analisis Sentimen, Support Vector Machine, Random Forest

1. Pendahuluan

Pandemi Covid-19 pada tahun 2020 menyebabkan perubahan besar di masyarakat. Perubahan ini didorong oleh IPTEK yang berkembang dengan sangat cepat dan canggih, serta meningkatkan penggunaan internet yang mempercepat penyebaran informasi melalui media sosial. Salah satu media sosial yang saat ini banyak mendapat perhatian di Indonesia adalah TikTok [1]. TikTok, yang diluncurkan secara resmi pada September 2016 oleh Zhang Yiming dari Cina, dapat diunduh secara gratis melalui Google Play Store. Menurut data dari situs web World Population Review, Indonesia adalah negara dengan pengguna aktif TikTok terbesar kedua di dunia setelah Amerika Serikat, dengan jumlah pengguna mencapai 99 juta.

TikTok adalah aplikasi yang sangat populer dan digunakan oleh orang-orang dari berbagai usia, termasuk anak-anak, remaja, dan dewasa. Aplikasi ini menyediakan fitur video dan musik dengan durasi hingga tiga menit, sehingga pengguna dapat menonton dan membagikan konten yang sesuai dengan minat mereka [2]. Aplikasi TikTok telah menawarkan banyak kenyamanan dan kemudahan bagi para pengguna, namun demikian, tidak semua pengguna merasa puas dengan kemudahan yang diberikan. Pada Google Play Store, pengguna memberikan rating untuk aplikasi dari skala 1 hingga 5. Namun, seringkali terjadi bahwa rating yang diberikan tidak selaras dengan isi ulasannya, sehingga rating saja tidak cukup untuk menilai kualitas aplikasi secara menyeluruh. Ulasan yang berbentuk kalimat memberikan informasi yang lebih jelas tentang bagaimana pengguna merespons aplikasi tersebut. Oleh karena itu, ulasan-ulasan ini memiliki potensi besar untuk mempengaruhi orang-orang yang sedang mencari aplikasi TikTok. Oleh karena itu, untuk membantu pihak pengembang aplikasi dalam memperoleh informasi mengenai kelebihan dan kekurangan dari aplikasi TikTok, perlu dilakukan analisis sentimen yang dapat mengklasifikasikan data ulasan dari pengguna berdasarkan rating dan isi ulasan.

Analisis sentimen bertujuan untuk mengelompokkan ulasan secara otomatis menjadi opini positif atau negatif [3]. Sentimen positif atau negatif dari masyarakat terhadap aplikasi TikTok dapat diidentifikasi secara otomatis menggunakan analisis sentimen dengan metode klasifikasi teks. Penelitian ini akan menggunakan metode Support Vector Machine (SVM) dan Random Forest untuk menentukan metode mana yang lebih efektif dalam konteks ini.

2. Landasan Teori

Analisis Sentimen

Analisis sentimen adalah suatu metode yang dipakai untuk secara otomatis menentukan sentimen atau opini yang terkandung dalam suatu teks. Metode ini biasanya digunakan dalam bidang linguistik dan ilmu komputer untuk mengetahui apakah sentimen yang terkandung dalam teks adalah positif atau negatif. Penggunaan analisis sentimen sering diaplikasikan untuk mengetahui apakah sebuah ulasan yang diposting online mengenai suatu produk, seperti film, buku, atau jasa, adalah positif atau negatif. Analisis sentimen juga telah menjadi alat yang

umum digunakan dalam analisis media sosial, khususnya oleh perusahaan, pemasar, dan analis politik. Cara kerja analisis sentimen melibatkan ekstraksi informasi dari kata-kata positif dan negatif dalam teks, serta dari konteks dan struktur linguistik yang digunakan dalam teks tersebut [4].

***Term Frequency-Inverse Document Frequency* (TF-IDF)**

Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah suatu proses untuk melakukan transformasi data dari data teks ke dalam data numerik untuk dilakukan pembobotan pada tiap kata atau fitur. TF-IDF ini adalah sebuah ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dokumen. TF adalah frekuensi kemunculan kata pada di tiap dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam tiap dokumen tersebut. DF adalah frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. IDF adalah *inverse* dari nilai DF. Hasil dari pembobotan kata menggunakan TF-IDF ini adalah hasil perkalian dari TF dikalikan dengan IDF [5]. Adapun rumus pembobotan TF-IDF sebagai berikut [6]:

$$TFIDF_{ab} = TF_{ab} \times IDF_a = \frac{freq_{ab}}{maxfreq_{ab}} \times \left(1 + log\ \frac{N}{df_a}\right) \quad (2.1)$$

Keterangan:

$TFIDFab$: bobot dari *term a* pada dokumen *b*
$TFab$: frekuensi *term a* pada dokumen *b*
$IDFa$: frekuensi dokumen invers *term a*
$freqab$: banyaknya kemunculan *term a* dalam dokumen *b*
$max\ freqab$: banyaknya *term a* pada dokumen *b*
N : banyaknya seluruh dokumen
dfa : banyaknya dokumen yang mengandung *term a*

***Synthetic Monitory Oversampling Technique* (SMOTE)**

Synthetic Minority Oversampling Technique (SMOTE) adalah sebuah metode yang digunakan untuk menangani permasalahan ketidakseimbangan kelas data pada dataset. Ketidakseimbangan data terjadi ketika jumlah objek pada suatu kelas data lebih banyak daripada kelas data yang lainnya. Kelas yang memiliki jumlah objek yang lebih sedikit disebut sebagai kelas minoritas, sedangkan kelas yang lain disebut kelas mayoritas. Ketidakseimbangan data dapat mempengaruhi hasil pembuatan model karena data cenderung diliputi oleh kelas mayoritas, sedangkan kelas minoritas diabaikan oleh algoritma yang tidak mempertimbangkan ketidakseimbangan tersebut. SMOTE mengatasi masalah ketidakseimbangan data dengan membuat sampel sintetis pada kelas minoritas dengan cara menggabungkan data yang ada dan menambahkan sampel baru di antara data yang sudah ada [7].

Pada metode SMOTE data kelas minor akan dilakukan penambahan dengan membangkitkan data buatan agar setara dengan data kelas mayor. Data buatan atau sintesis tersebut dibuat berdasarkan k-tetangga terdekat (*k-nearest neighbor*). Pembangkitan data buatan yang berskala numerik berbeda dengan kategorik. Data numerik diukur jarak kedekatannya dengan jarak Euclidean sedangkan data kategorik dengan nilai modus yaitu kategori yang paling sering muncul [8]. Jarak *Euclidean* dapat dicari menggunakan persamaan sebagai berikut [9]:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (2.2)$$

Keterangan :

d = Jarak
$x1$ = Koordinat latitude 1
$x2$ = Koordinat latitude 2
$y1$ = Koordinat Longitude 1
$y2$ = Koordinat Longitude 2

Pembangkitan data berskala numerik dilakukan berdasarkan persamaan berikut:

$$X_{baru} = x + (x^* - x) \times rand[0,1] \quad (2.3)$$

Keterangan :

$Xbaru$: data buatan hasil replikasi
x : data yang akan direplikasi
x^* : data yang memiliki jarak terdekat dari data yang akan direplikasi
$rand[0,1]$: bilangan acak antara 0 sampai 1

***Support Vector Machine* (SVM)**

Support Vector Machine (SVM) adalah metode pembelajaran yang digunakan untuk menganalisis data dan mengenali pola yang digunakan untuk mengklasifikasi. SVM berfungsi dengan membangun *hyperplane* dengan jangkauan Euclidean maksimum ke kasus latihan terdekat [10]. Data pada suatu dataset diberikan variabel xi , sedangkan untuk kelas pada dataset diberikan variabel yi . Metode SVM membagi dataset menjadi 2 kelas. Kelas pertama yang dipisah oleh *hyperplane* bernilai 1, sedangkan kelas lainnya bernilai -1. Maka persamaan yang didapatkan seperti Persamaan 4 dan Persamaan 5.

$$Xi.\ W + b \geq 1\ untuk\ Yi = 1 \quad (2.4)$$

$$Xi.\ W + b \leq -1\ untuk\ Yi = -1 \quad (2.5)$$

Keterangan :

Xi = data ke -i

W = niai bobot *support vector* yang tegak lurus dengan hyperplane

b = niai bias

$Yi = kelas\ data\ ke - i$

Bobot *vector* (*w)* adalah garis vektor yang tegak lurus antara titik pusat kordinat dengan garis *hyperplane*. Bias merupakan kordinat garis *relative* terhadap titik kordinat. Persamaan 6 merupakan persamaan untuk menghitung nilai *b*, sedangkan persamaan 7 merupakan persamaan untuk mencari nilai *w.*

$$b = -\frac{1}{2}(w.\ x^{+} + w.\ x^{-}) \quad (2.6)$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (2.7)$$

Keterangan :

b = nilai bias

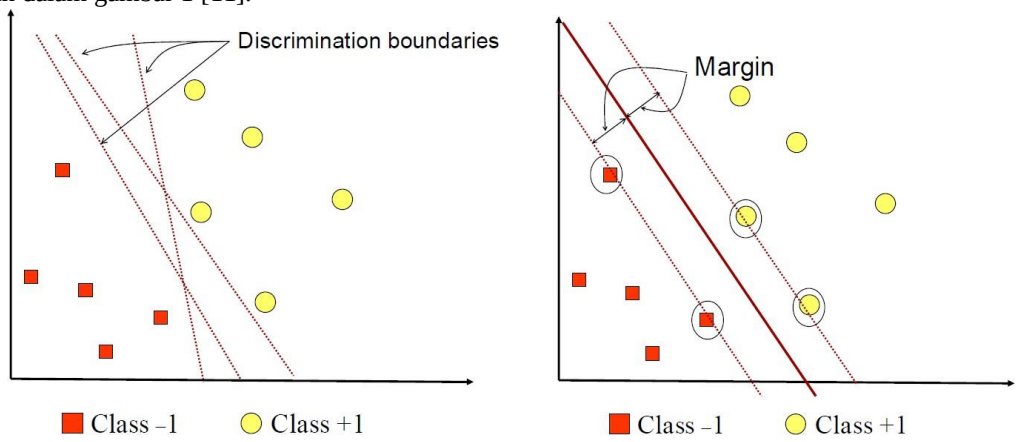
$w.\ x^{+}$ = nilai bobot untuk kelas data positif

$w.\ x^{-}$ = nilai bobot untuk kelas data negatif

w = bobot vektor

αi = nilai bobot data ke-i

SVM dapat disederhanakan sebagai usaha untuk menemukan *hyperplane* optimal yang memisahkan dua kelas dalam ruang input. Pada Gambar 1a, terdapat pola-pola yang termasuk dalam dua kelas: positif (+1) dan negatif (-1). Pola-pola yang termasuk dalam kelas negatif ditunjukkan dengan kotak, sedangkan yang dalam kelas positif ditunjukkan dengan lingkaran. Proses pembelajaran dalam masalah klasifikasi diartikan sebagai mencari garis (*hyperplane*) yang bisa memisahkan kedua kelompok ini. Berbagai opsi garis pemisah (*discrimination boundaries*) ditampilkan dalam gambar 1 [11].

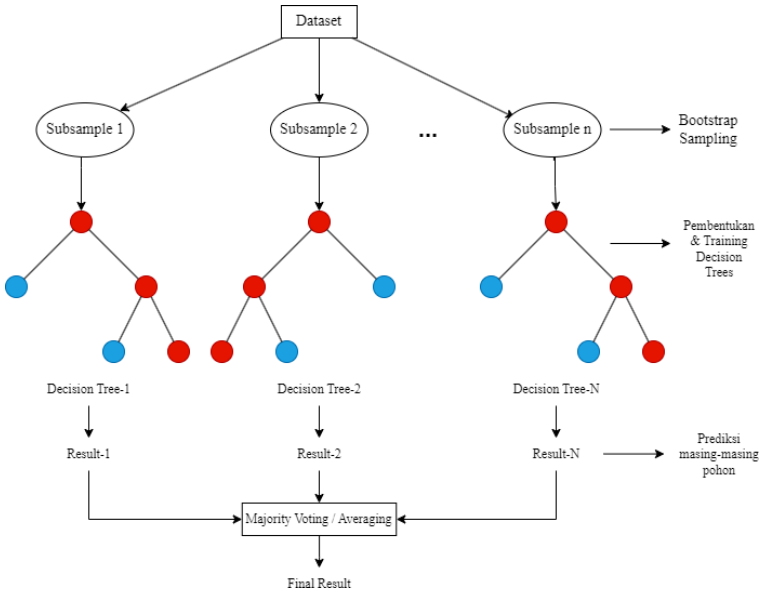


Gambar 1. Ilustrasi SVM dengan Regresi Linear

Random Forest

Random forest adalah sebuah metode yang dikembangkan dari metode *Classification and Regression Tree* (CART), yang juga merupakan metode atau algoritma dari teknik pohon keputusan. Yang membedakan metode random forest dari metode CART adalah *random forest* menerapkan metode *bootstrap aggregating* (bagging) dan juga seleksi fitur random atau bisa disebut random feature selection. *Random forest* adalah kombinasi dari masing masing teknik pohon keputusan yang ada, lalu kemudian digabung dan dikombinasikan ke dalam suatu model [12]. Ada tiga poin utama dalam metode Random Forest, tiga poin utama tersebut yaitu (1) melakukan bootstrap sampling untuk membangun pohon prediksi; (2) masing-masing pohon keputusan memprediksi dengan prediktor acak; (3) kemudian *random forest* melakukan prediksi dengan mengkombinasikan hasil dari tiap tiap pohon keputusan dengan cara majority vote untuk klasifikasi atau rata rata untuk regresi [13].

Random forest adalah algoritma *machine learning* yang menggunakan konsep *ensemble learning* untuk memecahkan masalah klasifikasi atau regresi. Konsep ensemble learning pada *random forest* didasarkan pada gabungan beberapa pohon keputusan atau *decision tree*. Setiap *tree* pada *random forest* akan mengeluarkan prediksi kelas, dan prediksi dengan vote terbanyak menjadi kandidat prediksi pada model. Semakin banyak jumlah *tree* yang digunakan, maka akan semakin meningkatkan akurasi dan mencegah terjadinya *overfitting* pada model. Algoritma *random forest* dikembangkan oleh Leo Breiman dan Adele Cutler sebagai solusi untuk mengatasi masalah *overfitting* pada *decision tree*. Dalam praktiknya, *random forest* adalah salah satu algoritma machine learning yang paling sering digunakan karena kemampuannya dalam memecahkan masalah yang kompleks dan meningkatkan kinerja model. Ilustrasi algoritma *random forest* dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi Algoritma *Random Forest*

Confusion Matrix

Confusion matrix adalah matriks 2x2 yang digunakan untuk merepresentasikan hasil dari klasifikasi biner pada suatu dataset. Ada beberapa rumus umum yang dapat digunakan untuk menghitung performa klasifikasi, dan nilai akurasi, presisi, dan recall dapat dinyatakan dalam bentuk persentase [14].

Accuracy, adalah proporsi dari jumlah prediksi yang benar dalam klasifikasi. Rumus perhitungan akurasi dapat ditemukan dalam persamaan berikut.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2.8)$$

Precision, merupakan rasio dari jumlah dokumen teks yang relevan dan terkendali terhadap total dokumen teks yang terpilih oleh sistem. Rumus precision dapat ditemukan dalam persamaan berikut.

$$Precision = \frac{TP}{TP+FP} \quad (2.9)$$

Recall, adalah proporsi dari jumlah dokumen teks yang relevan yang berhasil teridentifikasi di antara semua dokumen teks relevan dalam suatu koleksi. Rumus perhitungan recall dapat dilihat pada persamaan berikut.

$$Recall = \frac{TP}{TP+FN} \quad (2.10)$$

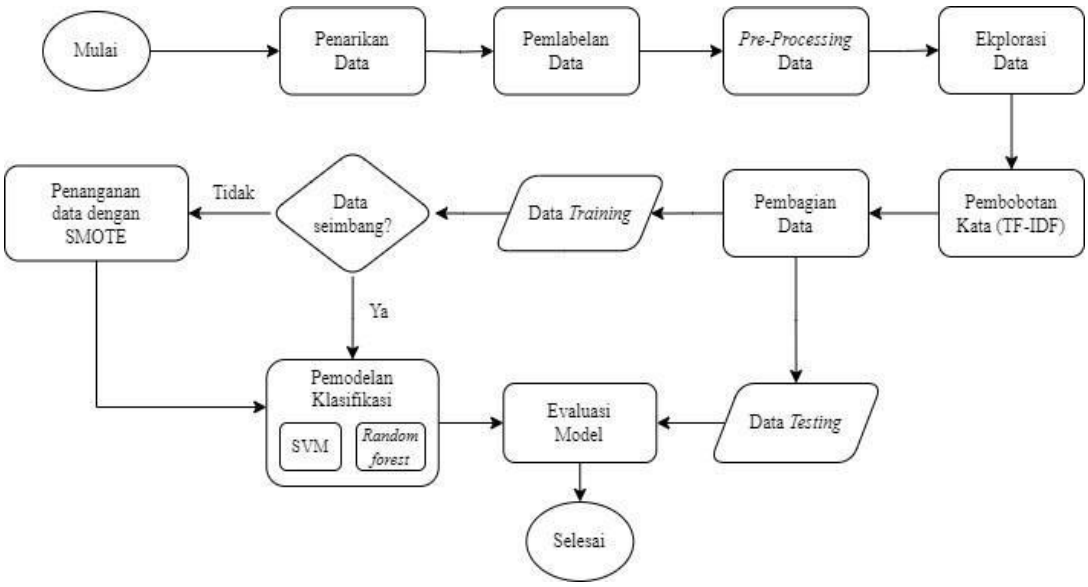
3. Metode Penelitian

Waktu dan Tempat Penelitian

Penelitian ini berlangsung dari April hingga Desember 2023. Proses penelitian dimulai dengan studi literatur, penyusunan proposal, pengumpulan data, dan pengolahan data. Data dikumpulkan secara online, sedangkan pengolahan data dilakukan dari rumah.

Analisis Data

Diagram alur analisis data pada penelitian ini dapat dilihat pada Gambar 3



Gambar 3. Diagram Alur Analisis Data

Analisis data dilakukan dengan menggunakan Google Colab dan bahasa pemrograman Python. Penjelasan prosedur analisis data adalah sebagai berikut:

Penarikan Data

Data ditarik dengan cara *scrapping* menggunakan Gooogle Colab dengan bahasa pemrograman Python dengan mengambil data ulasan masyarakat terhadap aplikasi TikTok di Google Play Store dengan total data sebanyak 5000. Ulasan yang diambil adalah ulasan yang paling relevan terhadap aplikasi dari 1 Juni sampai 23 November 2023.

Pelabelan Data

Selama proses pelabelan data, data dikategorikan ke dalam dua kelas, yaitu kelas negatif dan kelas positif. Pada penelitian ini, pelabelan dilakukan secara otomatis menggunakan library transformers.

***Pre-processing* Data**

Cleaning data, proses ini melibatkan penghapusan semua karakter lain selain huruf, seperti tanda baca, simbol, angka, dan emotikon. Selain itu, semua huruf akan diubah menjadi huruf kecil.

Tokenizing, bertujuan untuk membagi kalimat menjadi token atau bagian-bagian kata.
Stopwords removal, penghapusan kata tak bermakna dilakukan dengan menghilangkan kata-kata yang memberikan dampak minimal pada proses klasifikasi. Ini termasuk kata ganti orang, kata seruan, kata penghubung, dan kata lain yang terdapat dalam daftar kata tak bermakna.
Stemming, berfungsi untuk mengubah berbagai bentuk kata dalam teks menjadi bentuk aslinya. Dengan menggunakan stemming, kata-kata yang bervariasi seperti akhiran dapat dikelompokkan ke bentuk dasar yang sama. Dalam penelitian ini, modul Sastrawi digunakan sebagai referensi untuk mendapatkan bentuk kata dasarnya.

Eksplorasi Data

Eksplorasi data dilakukan untuk mengelompokkan data sesuai dengan kategori positif dan negatif serta untuk pembentukan awan kata.

Pembobotan TF-IDF

Tujuan dari pembobotan TF-IDF adalah untuk menemukan kata-kata yang paling relevan atau yang paling sering muncul dalam sebuah dokumen, dan membedakan kata-kata tersebut dengan kata-kata yang kurang relevan. Tiap kata unik dalam dokumen tersebut diberi bobot kata yang sesuai dengan pembobotan TF-IDF. Dalam penelitian ini, proses pembobotan TF-IDF menggunakan library *Scikit-learn.*

Pembagian Data dan Penanganan Ketidakseimbangan Data

Data akan dipecah menjadi dua kelompok, yakni data pelatihan (*training)* dan data uji (*testing)* dengan perbandingan 80:20. Keseimbangan data *training* akan diperiksa, dan apabila ditemukan ketidakseimbangan, metode *Synthetic Minority Oversampling Technique* (SMOTE) akan diterapkan untuk mengatasi permasalahan tersebut.

Pemodelan Klasifikasi

Proses pengembangan model klasifikasi melibatkan penggunaan metode SVM dan Random Forest. Model SVM menggunakan fungsi *linear kernel* dan dibangun dengan menggunakan pustaka *Support Vector Classification* (SVC) dari *toolkit Scikit-Learn*. Di sisi lain, model *Random Forest* dibuat dengan memanfaatkan pustaka *RandomForestClassifier* dari *Scikit-Learn.*

Evaluasi Model

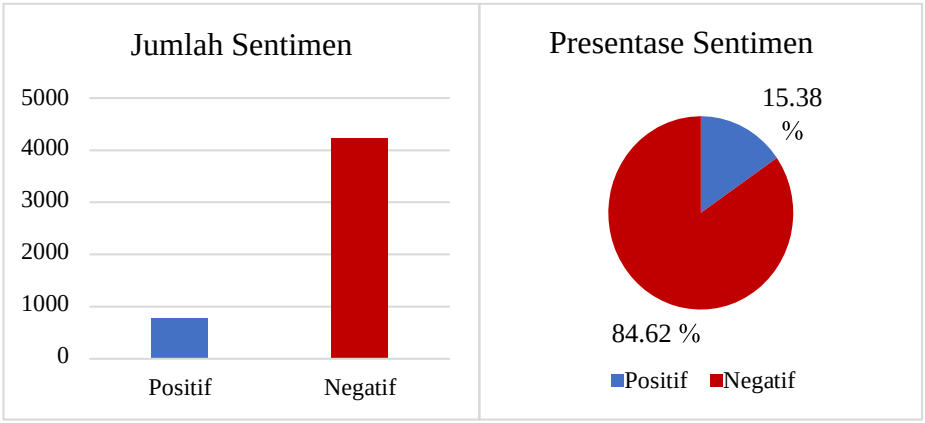
Evaluasi dilakukan untuk menilai kinerja dari model klasifikasi menggunakan SVM dan *random forest*. Data yang akan dipakai untuk mengevaluasi model klasifikasi adalah data *testing*. Evaluasi model akan dilihat berdasarkan perhitungan dari *confusion matrix*.

4. Hasil Penelitian

Eksplorasi Data

Ekplorasi Data Menurut Label

Data yang telah didapatkan sebanyak 5000 ulasan diberikan label positif dan negatif. Data yang berasal dari kelas negatif dan positif secara berurutan memiliki jumlah ulasan sebanyak 4231 dan 769. Dalam kategori sentimen negatif, ulasan memiliki persentase terbesar, yakni 84.62%, sementara sentimen positif sebesar 15.38%. Gambaran visual terkait jumlah dan persentase sentimen dapat dilihat melalui Gambar 4.



Gambar 4. Jumlah dan Presentase Sentimen

Pembentukan Awan Kata

Awan kata dibuat dengan tujuan memberikan representasi visual yang lebih mudah dipahami tentang kata-kata yang sering muncul dalam data. Dalam awan kata, ukuran kata disesuaikan terhadap frekuensi kemunculan kata tersebut dalam data. Sebagai contoh, kata yang terlihat lebih besar dalam awan kata menunjukkan frekuensi kemunculannya yang lebih tinggi dalam data [15]. Awan kata dari sentimen positif dapat dilihat pada Gambar 5.

Gambar 5. Awan Kata Sentimen Positif

Kata-kata dalam sentimen positif yang paling sering muncul adalah "tiktok" (627 kali), "aplikasi" (533 kali), dan "bagus" (347 kali). Visualisasi awan kata menunjukkan dominasi kata-kata ini, mengindikasikan bahwa pengguna sangat menikmati menggunakan aplikasi TikTok. Awan kata dari sentimen negatif dapat diperlihatkan pada Gambar 6.

Gambar 6. Awan Kata Sentimen Negatif

Tiga kata yang paling sering muncul dalam sentimen negatif adalah "tiktok" (4250 kali), "buka" (2839 kali), dan "update" (1591 kali). Visualisasi awan kata menegaskan dominasi kata-kata ini, mencerminkan keluhan utama pengguna. Kata "buka" yang sering muncul menunjukkan masalah dalam membuka atau menggunakan aplikasi, sementara tingginya frekuensi kata "update" menunjukkan ketidakpuasan pengguna setelah melakukan pembaruan aplikasi TikTok, yang mungkin menyebabkan masalah atau ketidaknyamanan.

***Term Frequency-Inverse Document Frequency* (TF-IDF)**

Matriks TF-IDF pada ulasan menghasilkan struktur berdimensi yang memetakan dokumen sebagai baris dan kata unik dalam dataset sebagai kolom. Dalam konteks dataset ini, hasil pembobotan menghasilkan matriks berdimensi (5000 x 8624), yang mencerminkan adanya 5000 baris dokumen dan sebanyak 8624 kata unik dalam dataset. Pembobotan TF-IDF ini akan berperan penting dalam proses pemodelan klasifikasi, yang akan dilakukan menggunakan metode SVM dan *random forest*.

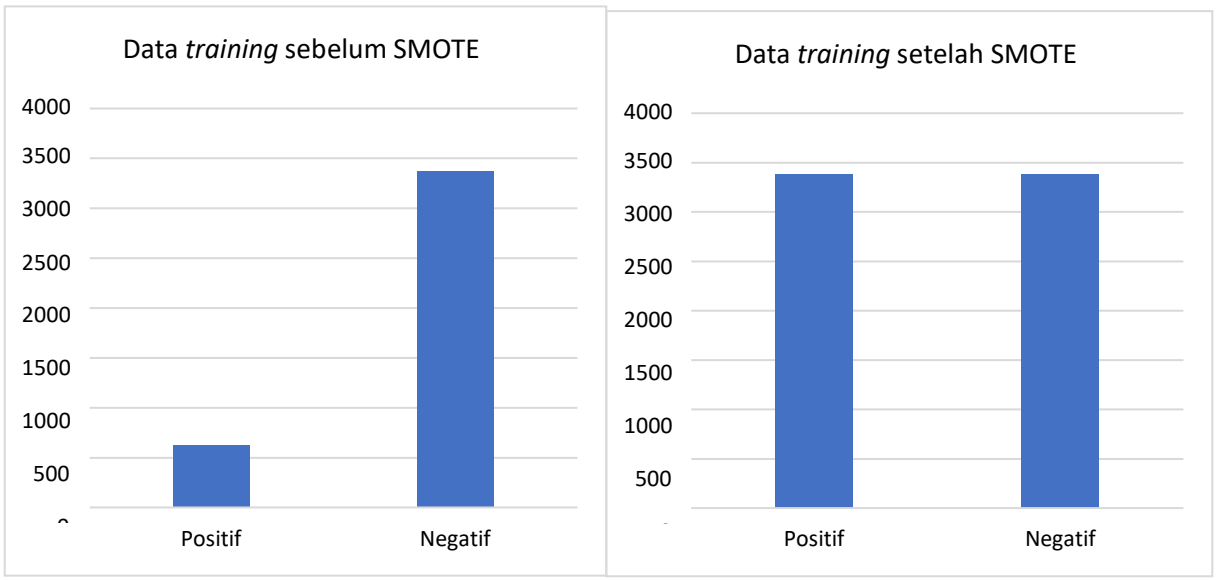
***Synthetic Minority Oversampling Technique* (SMOTE)**

Pengecekan ketidakseimbangan dalam dataset menjadi suatu langkah yang penting karena dapat berdampak pada hasil klasifikasi [16]. Untuk mengatasi ketidakseimbangan tersebut, digunakan metode *Synthetic Minority Oversampling Technique* (SMOTE) yang melakukan penambahan sampel (*oversampling)* pada kelas minoritas [17]. Proses *oversampling* untuk kelas minoritas dilakukan pada data *training*. Data awal dibagi menjadi dua kelompok, yaitu data *training* dan data *testing*, dengan perbandingan 80:20 untuk jumlah data *training* dan data *testing*.

Tabel 1. Hasil Pembagian Data *Training* dan *Testing*

Sentimen	80% data *training*	Total data *Training*	20% Data *testing*	Total data *testing*
Positif	625	4000	144	1000
Negatif	3375		856	

Tabel 1 menunjukkan ketidakseimbangan antara data *training* positif dan negatif dengan kelompok data *training* yang menjadi kelas minoritas adalah data dengan sentimen positif dengan total 625 sentimen. Oleh karena itu, dilakukan *oversampling* menggunakan SMOTE. Proses oversampling ini menghasilkan tambahan 2750 data pada kelas positif, sehingga total data menjadi 6750. Gambar 7 menunjukkan bahwa jumlah data pada setiap kelas sekarang seimbang, masing-masing dengan 3375 data.



Gambar 7. Data *Training* sebelum dan setelah SMOTE

Implementasi Model SVM

Metode *Support Vector Machine* (SVM) digunakan sebagai salah satu algoritma klasifikasi untuk menerapkan analisis sentimen dalam penelitian ini. Library Scikit-learn, yang merupakan salah satu library paling populer untuk pembelajaran mesin di Python, digunakan untuk mengimplementasikan SVM. Adapun hasil model SVM ditampilkan pada Tabel 2.

Tabel 2. Hasil Model SVM

Classification Report:	precision	recall	f1-score	support
Negatif	0.94	0.94	0.94	856
Positif	0.66	0.65	0.66	144
accuracy			0.90	1000
macro avg	0.80	0.80	0.80	1000
weighted avg	0.90	0.90	0.90	1000

Hasil klasifikasi dari model yang digunakan menunjukkan sejumlah metrik evaluasi yang memberikan gambaran komprehensif terkait kinerja model dalam mengklasifikasikan dua kelas sentimen, yaitu "Negatif" dan "Positif".

Implementasi Model *Random Forest*

Selain metode SVM, library Scikit-learn digunakan untuk mengimplementasikan Random Forest. Pertama, modul RandomForestClassifier dan GridSearchCV diimpor untuk membuat model dan melakukan pencarian parameter terbaik. Model Random Forest dibuat dengan menentukan random_state untuk hasil yang konsisten. Grid parameter yang diuji mencakup jumlah pohon (n_estimators) dengan nilai 50, 100, dan 200. GridSearchCV digunakan untuk mencari parameter terbaik berdasarkan grid parameter dengan cross-validation sebanyak 5 kali (cv=5) dan metrik akurasi sebagai scoring. Rincian lengkap hasil implementasi model *Random Forest* dapat ditemukan dalam Tabel 3.

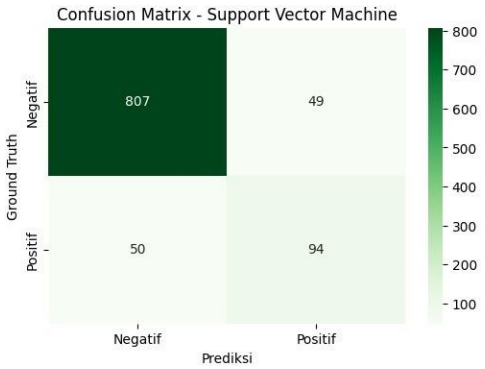
Tabel 3. Hasil Model *Random Forest*

Classification Report:	precision	recall	f1-score	support
Negatif	0.94	0.98	0.96	856
Positif	0.83	0.63	0.72	144
accuracy			0.93	1000
macro avg	0.89	0.81	0.84	1000
weighted avg	0.93	0.93	0.92	1000

Hasil klasifikasi menunjukkan performa model yang baik dalam memprediksi sentimen pada dataset yang digunakan.

Evaluasi Model

Pada tahapan ini, evaluasi dilakukan terhadap model klasifikasi menggunakan metode SVM dan *Random Forest*. Evaluasi dilakukan dengan memanfaatkan *confusion matrix* untuk memberikan gambaran yang lebih rinci tentang performa model dalam memprediksi sentimen. Masing-masing algoritma klasifikasi memiliki tingkat kesalahan terhadap proses klasifikasi dan prediksi, yang tercermin melalui hasil *confusion matrix*. Berikut adalah hasil evaluasi menggunakan *confusion matrix* dari SVM :.



Gambar 8. *Confusion Matrix* SVM

Berdasarkan gambar 6, terdapat 807 dokumen "Negatif" yang diprediksi benar sebagai "Negatif" (*true negatives)*, 94 dokomen "Positif" yang diprediksi benar sebagai "Positif" (*true positives)*, 50 dokumen yang seharusnya "Positif" tetapi diprediksi sebagai "Negatif" (*false negatives)*, serta 49 dokumen yang seharusnya "Negatif" tetapi diprediksi sebagai "Positif" (*false positives)*. Menggunakan nilai-nilai tersebut, dapat dihitung metrik evaluasi sebagai berikut :

Akurasi *(Accuracy):* Rasio dokumen yang diprediksi dengan benar (TN + TP) dengan jumlah total dokumen.

$$Accuracy = \frac{807 + 94}{94 + 49 + 807 + 50} = 0.901$$

Presisi *(Precision)*: Rasio dokumen positif yang diprediksi dengan benar (TP) dengan total dokumen yang diprediksi sebagai positif (TP + FP).

$$Precision = \frac{94}{94 + 49} = 0.657$$

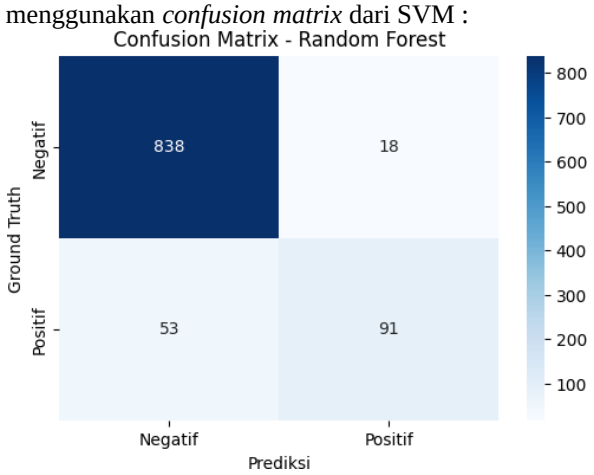
Recall (Sensitivity): Rasio dokumen positif yang diprediksi dengan benar (TP) dengan total dokumen positif sebenarnya (TP + FN).

$$Recall = \frac{94}{94 + 50} = 0.653$$

F1-*Score*: Ukuran gabungan dari presisi dan *recall*, menggunakan rumus:

$$F1 - score = 2 \times \frac{(0.657 \times 0.653)}{(0.657 + 0.653)} = 0.655$$

Berikut adalah hasil evaluasi menggunakan *confusion matrix* dari SVM :



Gambar 9. *Confusion Matrix Random Forest*

Berdasarkan gambar 7, terdapat 838 dokumen "Negatif" yang diprediksi benar sebagai "Negatif" (*true negatives)*, 91 dokomen "Positif" yang diprediksi benar sebagai "Positif" (*true positives)*, 53 dokumen yang seharusnya "Positif" tetapi diprediksi sebagai "Negatif" (*false negatives)*, serta 18 dokumen yang seharusnya "Negatif" tetapi diprediksi sebagai "Positif" (*false positives)*. Menggunakan nilai-nilai di atas, dapat dihitung metrik evaluasi sebagai berikut :

Akurasi *(Accuracy):* Rasio dokumen yang diprediksi dengan benar (TN + TP) dengan jumlah total dokumen.

$$Accuracy = \frac{831 + 91}{89 + 17 + 839 + 55} = 0.929$$

Presisi *(Precision)*: Rasio dokumen positif yang diprediksi dengan benar (TP) dengan total dokumen yang diprediksi sebagai positif (TP + FP).

$$Precision = \frac{91}{91 + 18} = 0.835$$

Recall (Sensitivity): Rasio dokumen positif yang diprediksi dengan benar (TP) dengan total dokumen positif sebenarnya (TP + FN).

$$Recall = \frac{91}{91 + 53} = 0.632$$

F1-*Score*: Ukuran gabungan dari presisi dan *recall*, menggunakan rumus:

$$F1 - score = 2 \times \frac{(0.835 \times 0.632)}{(0.835 + 0.632)} = 0.719$$

5. Kesimpulan

Berdasarkan penelitian, *Random Forest* menunjukkan tingkat akurasi sebesar 93%, sedangkan SVM mencapai 90%. Keduanya menggunakan TF-IDF untuk pembobotan kata dan SMOTE untuk mengatasi ketidakseimbangan data. SVM efektif pada data dengan margin kelas yang jelas namun kurang optimal pada dataset yang tidak seimbang, sementara *Random Forest* lebih stabil dalam menangani variasi data dan lebih tahan terhadap overfitting, meskipun memerlukan lebih banyak sumber daya komputasi. Dalam konteks analisis sentimen ulasan TikTok, penelitian ini menegaskan bahwa *Random Forest* menawarkan kinerja yang lebih konsisten dan akurat dibandingkan SVM, terutama pada dataset yang lebih besar. Oleh karena itu, untuk aplikasi dalam analisis sentimen ulasan TikTok, *Random Forest* menjadi pilihan yang lebih cocok, terutama jika tersedia sumber daya komputasi yang memadai.

6. Daftar Pustaka

[1] I. Carolin, G. D. Victoria, S. Dina dan M. Nastain, "Pengaruh Penggunaan New Media Tiktok Terhadap Pembentukan Konsep Diri Generasi Muda Indonesia 2022," Jurnal Ilmu Komunikasi dan Media Sosial, vol. 2, pp. 35-40, 2023.

[2] R. Rasdin, Y. Mulyanti dan K. Kurniawan, "Fenomena Tik Tok sebagai Media Komunikasi Edukasi," Riksa Bahasa XV, 2021.

[3] D. A. Kharisma dan Z. M. Nawawi, "Pengaruh Aplikasi Tik Tok Shop Terhadap Minat Berwirausaha Mahasiswa (Studi Kasus Mahasiswa Manajemen FEBI UINSU)," Jurnal Ilmiah Manajemen, Bisnis dan Kewirausahaan, vol. 3, no. 1, pp. 22-23, 2023.

[4] M. Taboada, "Sentiment Analysis: An Overview from Linguistics," Annu. Rev. Linguist. , vol. 2, no. 1, pp. 325- 347, 2016.

[5] J. A. Septian, T. M. Fahrudin dan A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TFIDF Dan K-Nearest Neighbor," Journal of Intelligent Systems and Computation, vol. 1, no. 1, pp. 43-49, 2019.

[6] D. Yogisth, T. N. Manjunath dan R. S. Hegadi, "Variants Of Term Frequency And Inverse Document Frequency Of Vector Space Model For Effective Document Ranking In Information Retrieval," IJTEE, vol. 8, no. 7, pp. 414-421, 2019.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall dan W. P. Kegelmaeyer, "SMOTE : Syntethic Minority Over-sampling Technique," Journal of Artificial Intellegence Research, vol. 16, pp. 321-357, 2002.

[8] R. A. Barro, I. D. Silvianti dan F. M. Afendi, "Penerapan Synthetic Minority Oversampling Technique (SMOTE) Terhadap Data Tidak Seimbang pada Pembuatan Model Komposisi Jamu," Xplore, vol. 1, no. 1, 2013.

[9] N. F. Umma, B. Warsito dan D. A. I. Maruddin, "Klasifikasi Status Kemiskinan Rumah Tangga dengan Algoritma C5.0 di Kabupaten Pemalang," Jurnal Gaussian, vol. 10, no. 2, pp. 221-299, 2021.

[10] A. S. H. Basari, B. Hussin, I. G. P. Ananta dan J. Zeniarja, "Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization," Procedia Engineering, pp. 453-462, 2013.

[11] A. S. Nugroho, A. B. Witarto dan D. Handoko, Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika, 2003.

[12] M. R. Adrian, M. P. Putra dan N. A. Rakhmawati, "Perbandingan Metode Klafisikasi Random Forest dan SVM Pada Analisis Sentimen PSBB," Jurnal Informatika UPGRIS, pp. 36-40, 2021.

[13] A. Primaya dan B. N. Sari, "Random Forest Algorithm for Prediction of Precipitation," IJAIDM, vol. 1, no. 1, pp. 27-31, 2018.

[14] L. A. Andika, "Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier," Indonesian Journal of Applied Statistics, vol. 2, no. 1, 2019.

[15] M. G. Pradana, "Penggunaan Fitur Wordcloud dan Document Term Matrix dalam Text Mining," Jurnal Ilmiah Informatika, vol. 8, no. 1, 2020.

[16] O. D. Amelia, A. M. Soleh dan S. Rahardiantoro, "Pemodelan support vector machine data tidak seimbang keberhasilan studi mahasiswa magister IPB," Xplore, vol. 5, no. 1, pp. 122-130, 2018.

[17] R. A. Nurdian, M. Ridwan dan A. Yusuf, "Komparasi Metode SMOTEdan ADASYNdalam Meningkatkan Performa Klasifikasi Herregistrasi Mahasiswa Baru," Jurnal Teknik Informatika dan Sistem Informasi, vol. 8, no. 1, pp. 24-32, 2022.