

Klasifikasi Sentimen Ulasan Aplikasi Gojek Berbasis *Decision Tree* Dengan Optimasi *Grid Search*

Nabilah Mufti^{a,1,*}, Aurahaqqi Aprianoputri^{b,2}, Rendra Gustriansyah^{c,3}

^{a,b} Teknik Informatika, Fakultas Ilmu Komputer dan Sains, Universitas Indo Global Mandiri, Indonesia

¹ 2022110016@students.uigm.ac.id*; ² 2022110087@students.uigm.ac.id; ³ rendra@uigm.ac.id

* Penulis Korespondensi

ABSTRAK

Penelitian ini bertujuan untuk mengklasifikasikan sentimen ulasan pengguna aplikasi Gojek secara otomatis menggunakan algoritma *Decision Tree* dengan optimasi *hyperparameter GridSearchCV* dan teknik penyeimbangan data SMOTE. Ulasan pengguna dari *Google Play Store* mencerminkan persepsi terhadap layanan, sehingga diperlukan metode analisis yang efisien. Dataset terdiri dari 44.950 ulasan yang diperoleh dari *Kaggle* dan diproses melalui tahapan tokenisasi, *stopword removal*, *stemming*, dan representasi numerik menggunakan TF-IDF. Model *Decision Tree* awal menghasilkan akurasi 87,59%. Setelah penerapan *GridSearchCV* dan SMOTE, akurasi model pada data seimbang menjadi 83,5% dengan *F1-score* 0,84. Namun, performa terhadap kelas minoritas (netral) masih rendah. Sebagai pembandingan, model *Random Forest* menunjukkan hasil lebih baik dengan akurasi 86,5% dan *F1-score* 0,86. Hasil penelitian menunjukkan bahwa kombinasi TF-IDF dan *Decision Tree* efektif untuk klasifikasi sentimen mayoritas, tetapi kurang optimal pada kelas minoritas. Penggunaan model yang lebih kompleks seperti *Random Forest* disarankan untuk hasil klasifikasi yang lebih merata.

Riwayat Artikel

Diterima 10 Juni 2025

Diperbaiki 24 Juni 2025

Diterbitkan 30 Oktober 2025

Kata Kunci

Decision Tree

GridSearchCV

Sentiment

Random Forest

TF-IDF



This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

1. Pendahuluan

Di zaman digital yang terus berkembang dengan cepat, aplikasi transportasi *online* menjadi solusi utama untuk mobilitas masyarakat. Salah satu aplikasi yang sangat dikenal di Indonesia adalah Gojek. Aplikasi ini tidak hanya menawarkan layanan transportasi, tetapi juga meliputi pemesanan makanan, pengiriman barang, pembayaran secara digital, hingga jasa kesehatan [1] Seiring dengan bertambahnya jumlah pengguna, jumlah ulasan dan pendapat pengguna mengenai layanan Gojek juga semakin meningkat, khususnya di *platform* seperti *Google Play Store*. Ulasan-ulasan ini menggambarkan beragam pengalaman dan tingkat kepuasan pengguna yang dapat dianalisis untuk menilai dan meningkatkan kualitas layanan dari aplikasi tersebut.

Pada masa digital saat ini, jumlah ulasan dari pengguna mengenai layanan *online*, seperti transportasi dan perdagangan elektronik, mengalami peningkatan yang signifikan. Proses analisis manual terhadap ulasan-ulasan tersebut menjadi kurang efisien dan memakan waktu. Oleh karena itu, diperlukan metode otomatis yang mampu dengan tepat dan efisien mengidentifikasi serta mengklasifikasikan sentimen dalam ulasan [2]

Analisis sentimen, sebagai cabang dari pemrosesan bahasa alami (*NLP*), telah berkembang pesat dengan penerapan teknik pembelajaran mesin dan pembelajaran mendalam. Pendekatan ini memungkinkan pemahaman opini publik secara sistematis dan dapat diandalkan [3] Studi terbaru juga menyoroti pentingnya analisis sentimen dalam berbagai domain, termasuk media sosial dan ulasan produk, dengan menekankan tantangan seperti adaptasi domain dan keragaman *linguistic*[4].

Dalam penelitian ini, algoritma yang digunakan adalah *Decision Tree*, salah satu metode pembelajaran terawasi yang umum digunakan untuk klasifikasi. *Decision Tree* membangun struktur pohon keputusan dari data pelatihan, di mana setiap simpul mewakili keputusan berdasarkan nilai dari fitur tertentu. Keunggulan utama dari *Decision Tree* adalah kemampuannya menghasilkan

model yang mudah dipahami serta kemampuannya menangani data kategorikal dan numerik. Penelitian sebelumnya menunjukkan bahwa *Decision Tree* dapat mencapai akurasi yang kompetitif dalam analisis sentimen, meskipun performanya dapat dipengaruhi oleh panjang teks dan kompleksitas data [5]

Keberhasilan dalam analisis sentimen bergantung pada langkah-langkah pra-pemrosesan teks yang tepat dan pemilihan fitur yang akurat. Dalam studi ini, dilakukan aktivitas seperti pemecahan kalimat (*tokenisasi*), penghilangan kata umum (*stopword*), dan ekstraksi fitur menggunakan metode *Term Frequency-Inverse Document Frequency (TF-IDF)*. TF-IDF berfungsi untuk mengubah teks ulasan menjadi format numerik yang menunjukkan seberapa penting kata-kata tertentu dalam dokumen, yang kemudian digunakan sebagai masukan untuk model klasifikasi [4]. Gabungan metode pra-pemrosesan dan pemodelan dengan *Decision Tree* diharapkan dapat memberikan hasil klasifikasi yang terbaik dalam membagi ulasan menjadi sentimen positif, negatif, atau netral.

Penelitian yang dilakukan oleh Ananda, Syahri, dan Hasan pada tahun 2024 menerapkan algoritma *Naive Bayes* untuk mengategorikan 300 ulasan pengguna Gojek dan Grab di Google Play Store. Temuan tersebut menunjukkan bahwa Gojek mendapat 49% ulasan positif dan 51% negatif, sedangkan Grab memperoleh 67% ulasan positif dan 33% negatif. *Naive Bayes* terbukti efektif dalam merefleksikan pandangan pengguna terhadap kedua aplikasi tersebut [6].

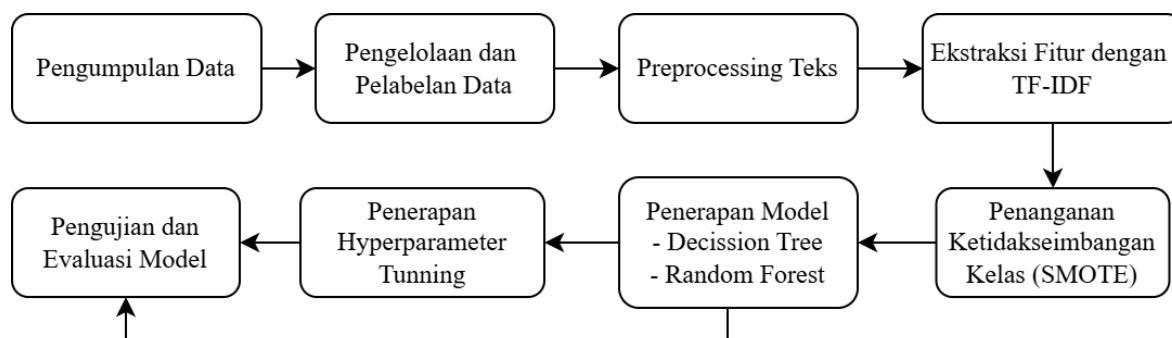
Studi yang dilakukan oleh Azwanti dan Putria pada tahun 2024 mengevaluasi kepuasan pelanggan di *Sdtechnology Computer* menggunakan metode *Decision Tree* berdasarkan data survei. Hasil analisis menunjukkan bahwa kualitas produk dan layanan pelanggan menjadi faktor utama yang mempengaruhi kepuasan. Algoritma *Decision Tree* diperlihatkan mampu menemukan pola penting dalam data yang diperoleh dari survei [7].

Penelitian oleh Cahyaningtyas, Nataliani, dan Widiyari pada tahun 2021 memanfaatkan *Decision Tree* dengan pendekatan *SMOTE* untuk mengklasifikasikan sentimen *rating* aplikasi Shopee. Dengan penerapan *SMOTE*, model ini berhasil mencapai akurasi sebesar 99,91%, AUC 0,999, recall 99,88%, dan precision 99,98%. Ini menunjukkan bahwa kombinasi *Decision Tree* dan *SMOTE* sangat efektif dalam menangani masalah ketidakseimbangan data sentimen [8].

Berdasarkan celah dari penelitian sebelumnya, penelitian ini berfokus pada penggunaan algoritma *Decision Tree* yang dikombinasikan dengan teknik preprocessing teks dan pemilihan fitur berbasis TF-IDF. Dengan pendekatan ini, diharapkan hasil analisis dapat lebih akurat dalam menggambarkan persepsi pengguna terhadap berbagai aspek layanan dalam aplikasi Gojek, serta memberikan kontribusi bagi pengembangan kualitas layanan berbasis data.

2. Metode

Langkah-langkah penelitian dijelaskan dalam Gambar 1. sebagai petunjuk untuk setiap tahap yang akan dilaksanakan.



Gambar 1 Langkah-Langkah Penelitian

2.1 Pengumpulan data

Data yang digunakan dalam penelitian ini diambil dari situs Kaggle (www.kaggle.com) dengan nama dataset “GojekAppReviewV4. 0. 0-V4. 9. 3_Cleaned. csv”. Dataset tersebut memuat 225. 044 ulasan dari pengguna aplikasi Gojek yang telah diproses melalui tahap pembersihan. Setiap entri dalam dataset tersebut terdiri dari teks ulasan serta label sentimen yang dikelompokkan menjadi tiga kategori, yaitu positif, netral, dan negatif. Dataset ini dijadikan sebagai dasar untuk pengembangan model klasifikasi sentimen menggunakan algoritma Decision Tree.

Adapun variabel-variabel yang tersedia dalam dataset tersebut antara lain: (1) *userName* yang menunjukkan nama pengguna yang memberikan ulasan, (2) *content* berupa teks ulasan yang diberikan, (3) *score* yang merupakan skor rating dari pengguna (dalam rentang nilai 1 hingga 5), (4) *at* yang menunjukkan tanggal ulasan diberikan, serta (5) *appVersion* yang mencatat versi aplikasi Gojek yang digunakan saat ulasan dibuat.

2.2 Pengelolaan Data

Setelah informasi berhasil dikumpulkan, tahap selanjutnya adalah mengatur data dan memberikan *label*. Pengaturan data bertujuan untuk memastikan bahwa kumpulan data dalam keadaan siap untuk dianalisis lebih lanjut. Di dalam penelitian ini, seluruh tahapan pengaturan data dilakukan menggunakan bahasa pemrograman *Python* yang digunakan dalam *platform Google Colaboratory*. Beberapa tahapan penting yang dilakukan meliputi pemeriksaan konsistensi data untuk memastikan setiap entri memiliki struktur yang seragam dan tidak terdapat nilai kosong atau atribut yang tidak relevan, pembersihan data lanjutan seperti menghapus karakter khusus dan simbol yang tidak diperlukan serta memperbaiki teks ulasan yang rusak, seleksi data dengan memfokuskan pada kolom-kolom penting yaitu teks ulasan dan label sentimen, serta standarisasi format data untuk menyeragamkan tipe data seperti string dan numerik sehingga seluruh dataset berada dalam format yang konsisten [9].

Prosedur pelabelan data atau *labeling* dilaksanakan untuk menentukan jenis sentimen pada masing-masing teks ulasan, sehingga dapat digunakan dalam metode pembelajaran terawasi. Dalam studi ini, jenis sentimen dibagi menjadi tiga kategori, yaitu positif, netral, dan negatif. Proses pelabelan dilakukan dengan menggunakan nilai skor atau penilaian dari pengguna sebagai dasar untuk mengklasifikasikan sentimen. Selain itu, jika ditemukan entri yang belum memiliki label yang sesuai maka dilakukan pelabelan ulang secara manual dengan bantuan fitur visualisasi dan manipulasi data pada *Python* seperti menggunakan *library pandas* dan *matplotlib* yang memudahkan proses identifikasi dan pengelompokan data secara efisien [10]

2.3 Preprocessing Teks

Preprocessing bertujuan untuk menyiapkan data teks mentah agar siap untuk diproses oleh algoritma pembelajaran mesin. Tahapan dalam proses ini meliputi beberapa langkah penting, seperti tokenisasi yang membagi teks menjadi kata-kata terpisah, penghapusan *stopword* untuk menghapus kata-kata umum yang tidak memiliki makna analitis seperti konjungsi dan artikel, *stemming* yang mengubah kata-kata menjadi bentuk dasarnya, serta *case folding* yang mengkonversi semua huruf menjadi huruf kecil. Tahapan *preprocessing* ini mengacu pada praktik pengolahan bahasa alami sebagaimana dijelaskan oleh Rifaldi et al. (2023) [11]

2.4 Ekstraksi Fitur dengan TF-IDF

Setelah tahap pemrosesan awal, data teks diubah menjadi bentuk angka dengan memanfaatkan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Metode ini menilai seberapa penting suatu kata dalam sebuah dokumen dengan mempertimbangkan frekuensinya, sehingga model dapat lebih konsentrasi pada kata-kata yang signifikan dalam proses pengklasifikasian [12]. TF-IDF merupakan metode untuk menilai seberapa penting sebuah kata dalam suatu dokumen relatif terhadap kumpulan dokumen lainnya atau korpus. Metode ini menekankan kata-kata penting dan mengurangi bobot kata-kata umum yang sering muncul di seluruh dokumen [13]. Pendekatan ini

banyak digunakan dalam pemrosesan bahasa alami dan klasifikasi teks karena kesederhanaannya serta kemampuannya dalam merepresentasikan informasi penting dari dokumen.

Secara matematis, TF-IDF dihitung dengan Persamaan 1 dan 2.

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (1)$$

$$TF(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \quad (2)$$

dimana $f_{t,d}$ merujuk pada jumlah kali kata t muncul dalam dokumen d , dan dimana N merupakan total keseluruhan dokumen dan n_t adalah banyaknya dokumen yang memuat kata t sesuai ditunjukkan pada Persamaan 3.

$$IDF(t) = \log\left(\frac{N}{n_t}\right) \quad (3)$$

2.5 Penanganan Ketidakseimbangan Kelas (SMOTE)

Teknik SMOTE (Synthetic Minority Over sampling Technique) digunakan untuk mengatasi masalah ketidakseimbangan kelas dalam dataset, di mana kelas minoritas memiliki jumlah data yang lebih sedikit. SMOTE Borderline berfokus pada titik data yang sulit untuk dipisahkan. Metode ini lebih ampuh dibandingkan SMOTE standar dalam menangani data yang terletak di area batas keputusan. Hal ini disebabkan karena metode ini menekankan pada sampel yang paling sulit untuk diklasifikasikan. Rumus yang digunakan untuk menciptakan titik data sintetis dengan SMOTE pada Persamaan 4.

$$X_{new} = X_i + \lambda (X_k - X_i) \quad (4)$$

SMOTE menciptakan data baru yang bersifat sintetis (X_{new}) dengan cara menginterpolasi antara data dari kelas yang kurang (X_i) dan tetangganya yang paling dekat (X_k) dengan menggunakan nilai acak (λ). Metode ini menekankan pada data yang berada di sekitar batas keputusan antara kelas, memberikan dukungan kepada model untuk mempelajari data yang sulit dipisahkan sekaligus meningkatkan kinerja pada dataset yang tidak seimbang [14], [15].

2.6 Penerapan Metode Decision Tree

Decision Tree (Pohon keputusan) adalah teknik klasifikasi yang didasarkan pada bentuk pohon, di mana data dipisahkan menjadi cabang-cabang berdasarkan ciri-ciri yang paling penting. Algoritma ini berfungsi dengan memilih atribut yang memberikan peningkatan informasi tertinggi sebagai fondasi untuk membuat simpul pada pohon keputusan [16]. Keunggulan utama algoritma *Decision Tree* meliputi kemudahan interpretasi dan visualisasi yang jelas, serta efisiensi dalam menangani dataset berukuran besar dan data dengan hubungan *non-linear*. Selain itu, algoritma ini mampu memproses data baik dalam bentuk numerik maupun kategorikal. Dalam konteks analisis sentimen, *Decision Tree* terbukti efektif dalam mengenali pola-pola kata yang berkaitan dengan polaritas ulasan pengguna, sehingga mendukung proses klasifikasi sentimen menjadi kategori positif, netral, dan negatif [17]. Dalam penelitian ini, dilakukan tuning terhadap *hyperparameter* algoritma *Decision Tree* untuk meningkatkan akurasi model. Beberapa parameter yang disesuaikan meliputi:

- *max_depth*, untuk membatasi kedalaman maksimal pohon dan menghindari *overfitting*.
- *min_samples_split*, menentukan jumlah sampel paling sedikit yang diperlukan untuk memecah *node*.
- *min_samples_leaf*, menentukan jumlah minimum sampel dalam *node* daun.
- *max_features*, untuk membatasi jumlah fitur yang digunakan saat pemisahan.
- *criterion*, menggunakan metode *Gini Impurity* atau *Entropy* dalam mengukur kualitas split.

Proses penyetelan dilakukan dengan cara *grid search* atau *random search* untuk mencari kombinasi parameter yang memberikan kinerja optimal berdasarkan metrik seperti akurasi, presisi, pengembalian, dan nilai F1. Formula-formula utama yang digunakan selama proses pembentukan dan penyetelan pohon ditunjukkan pada Persamaan 5 sampai Persamaan 7.

1) *Split Function*

a. *Gini Index*

$$Gini(t) = 1 - \sum_{i=1}^c p_i^2 \quad (5)$$

dimana p_i : probabilitas kelas ke-i di node t. semakin kecil nilai Gini, semakin murni *node*-nya

b. *Entropy*

Mengukur tingkat ketidakpastian atau keacakan data:

$$Entropy(t) = - \sum_{i=1}^c p_i \log_2(p_i) \quad (6)$$

nilai *entropy* digunakan untuk menghitung *Information Gain*.

c. *Information Gain* (jika memakai *entropy*)

Digunakan untuk memilih atribut terbaik berdasarkan pengurangan *entropy*:

$$IG(D, A) = Entropy(D) - \sum_{i=1}^c \frac{|D_v|}{|D|} \cdot Entropy(D_v) \quad (7)$$

dengan D_v adalah subset dari D berdasarkan nilai atribut A .

2) *Fungsi Pemilihan Fitur Terbaik*

Fitur terbaik A^* ditunjukkan pada Persamaan 8.

$$A^* = \arg_{A \in \text{fitur}}^{max} IG(D, A) \quad (8)$$

3) *Fungsi Pemangkasan dan Batasan Model (Hyperparameter)*

- *Max Depth Rule*:

Stop jika $depth > max_depth$

- *Min Samples Split Rule*:

Split jika $n_{sample} \geq min_samples_leaf$

- *Min Samples Leaf Rule*

Node valid jika $n_{sample} \geq min_sample_leaf$

- *Max Features*:

$$\text{Fitur yang digunakan} = \begin{cases} \sqrt{n} & \text{jika } max_features = 'sqrt' \\ \log_2(n) & \text{jika } max_features = 'log_2' \\ k & \text{jika } max_features = k \end{cases}$$

4) *Tujuan Optimasi Hyperparameter*

Mencari kombinasi parameter (θ) ditunjukkan pada Persamaan 9.

$$\theta^* = \arg_{\theta \in \Theta}^{max} Accuracy(M_\theta) \quad (9)$$

dengan Θ : ruang parameter dan M_θ : model dengan kombinasi parameter θ

2.7 Komparasi Model: *Random Forest*

Untuk menilai kinerja algoritma Decision Tree dengan lebih mendalam, penelitian ini juga menggunakan Random Forest, yang merupakan metode pembelajaran ensemble yang menciptakan beberapa pohon keputusan dan mengintegrasikan hasil klasifikasinya lewat pemungutan suara mayoritas. Random Forest diterapkan dengan pengaturan default serta versi yang telah dioptimalkan dengan menggunakan `class_weight='balanced'` untuk menangani perbedaan distribusi label. Metode ini dipilih karena kemampuannya dalam menurunkan varians dari model tunggal seperti Decision Tree, sekaligus memberikan kinerja yang lebih konsisten pada kelas yang lebih kecil. [18].

2.8 Pengujian dan Evaluasi Model (*Confusion Matrix*)

Confusion Matrix adalah teknik evaluasi yang sering digunakan dalam analisis data dan pembelajaran mesin untuk menilai kinerja model klasifikasi. Secara dasar, *Confusion Matrix* membandingkan nilai asli dari data uji dengan nilai prediksi yang dihasilkan oleh model. Proses ini memberikan informasi rinci mengenai kemampuan model dalam mengklasifikasikan data ke dalam kategori yang benar, dengan menganalisis empat hasil prediksi yang mungkin, yaitu: *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)*. Berdasarkan analisis ini, *Confusion Matrix* menjadi alat penting untuk mengevaluasi kelebihan dan kekurangan dari model klasifikasi yang sedang diuji [19], [20]. Perhitungan matrik evaluasi untuk pengujian ditunjukkan pada Tabel 1.

Tabel 1 Confusion Matrix

<i>Confusion Matrix</i>	<i>Predicted Positive</i>	<i>Predicted Negative</i>
<i>Actual Positive</i>	TP	FN
<i>Actual Negative</i>	FP	TN

Perhitungan masing- masing komponen ditunjukkan pada persamaan 10 hingga persamaan 13.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (10)$$

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

$$recall = \frac{TP}{TP + FN} \quad (12)$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \quad (13)$$

3. Hasil dan Pembahasan

3.1 Pengelolaan Data

Setelah data dikumpulkan, proses pengelolaan dilakukan terhadap lebih dari 225.044 baris ulasan dari pengguna aplikasi Gojek. Langkah pertama yang dilakukan adalah menghapus data yang memiliki entri duplikat, sehingga jumlah data menyusut menjadi 225.043 baris unik. Selanjutnya, dilakukan pembersihan terhadap baris yang tidak memiliki isi ulasan (kosong pada kolom *content*), yang menghasilkan 224.928 data bersih yang valid. Struktur data akhir terdiri dari 224.928 baris dan 9 kolom, yang mencakup informasi penting seperti *userName*, *content*, *score*, *at*, *appVersion*, serta beberapa kolom tidak bernama (*Unnamed*) yang merupakan sisa dari proses ekspor data dan sebagian besar berisi nilai kosong. Semua kolom masih bertipe teks (*object*), yang sesuai untuk keperluan analisis berbasis teks seperti analisis sentimen atau pengelompokan opini. Data pelabelan pada penelitian ini ditunjukkan pada Tabel 2.

Tabel 2 Pelabelan Data

Ulasan	Rating	Label
akun gopay saya di blok	1	Negatif
lambat banget sekarang ini bosku aplikasi gojek tidak seperti sebelumnya	3	Netral
kenapa dari kemarin saya membuka aplikasi gojek malah tertutup sendiri kemudian saya kasih bintang 2 dulu, kalau sudah baik saya kasih bintang 7	4	Positif
mantap	5	Positif

Label sentimen ditentukan berdasarkan nilai rating sebagai berikut:
rating ≥ 4 dianggap positif (1), rating = 3 dianggap netral, lainnya negatif (0)

3.2 Preprocessing Teks

Proses *preprocessing* meliputi tahapan *case folding*, tokenisasi, *stopword removal*, dan *stemming*. Langkah-langkah ini berhasil mengurangi *noise* dan memfokuskan analisis hanya pada kata-kata bermakna. Hasil *preprocessing* menunjukkan peningkatan efisiensi dan akurasi pada tahap transformasi fitur.

Data Mentah

kenapa dari kemarin saya membuka aplikasi gojek malah tertutup sendiri kemudian saya kasih bintang 2 dulu, kalau sudah baik saya kasih bintang 7

Case Folding

kenapa dari kemarin saya membuka aplikasi gojek malah tertutup sendiri kemudian saya kasih bintang 2 dulu, kalau sudah baik saya kasih bintang 7

Tokenisasi

['kenapa', 'dari', 'kemarin', 'saya', 'membuka', 'aplikasi', 'gojek', 'malah', 'tertutup', 'sendiri', 'kemudian', 'saya', 'kasih', 'bintang', '2', 'dulu', 'kalau', 'sudah', 'baik', 'saya', 'kasih', 'bintang', '7']

Stopword Removal

['kenapa', 'kemarin', 'saya', 'membuka', 'aplikasi', 'gojek', 'malah', 'tertutup', 'sendiri', 'kemudian', 'saya', 'kasih', 'bintang', '2', 'dulu', 'kalau', 'sudah', 'baik', 'saya', 'kasih', 'bintang', '7']

Stemming

['kenapa', 'kemarin', 'saya', 'membuka', 'aplikasi', 'gojek', 'malah', 'tertutup', 'sendiri', 'kemudian', 'saya', 'kasih', 'bintang', '2', 'dulu', 'kalau', 'sudah', 'baik', 'saya', 'kasih', 'bintang', '7']

3.3 Ekstraksi Fitur dengan TF-IDF

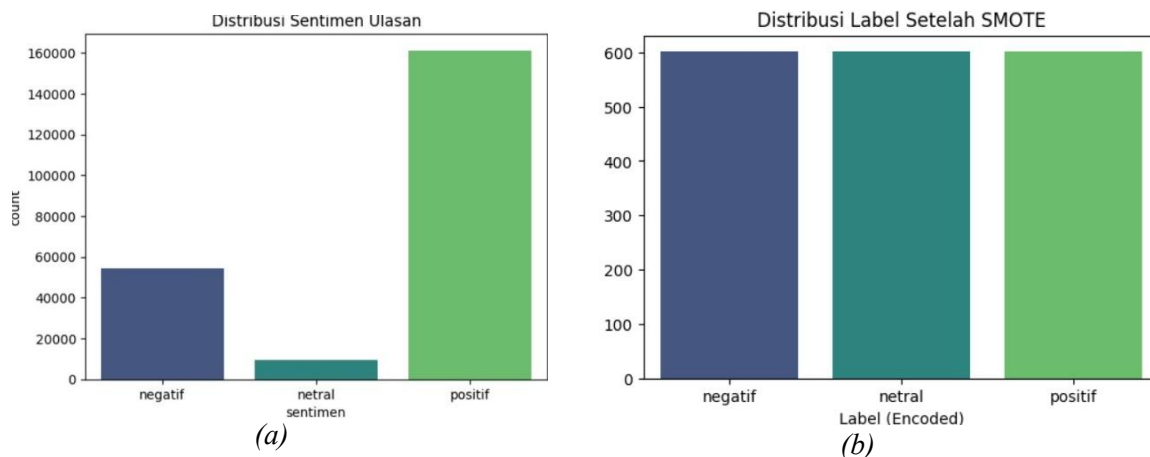
Setelah tahap *preprocessing*, ulasan tersebut diubah menjadi format angka dengan menggunakan metode TF-IDF (Frekuensi Kata-Inverse Frekuensi Dokumen). Prosedur ini menghasilkan matriks fitur yang nantinya akan diterapkan dalam model klasifikasi.

Hasil dari ekstraksi fitur dengan metode TF-IDF menghasilkan matriks yang memiliki banyak dimensi (1000, 1000), yang berarti terdapat 1000 dokumen dengan masing-masing direpresentasikan oleh 1000 fitur kata unik. Setiap nilai dalam matriks ini menunjukkan skor TF-IDF, yang merepresentasikan tingkat kepentingan sebuah kata terhadap suatu dokumen relatif terhadap seluruh dokumen. Matriks ini digunakan sebagai input bagi model *Decision Tree* untuk proses pelatihan dan prediksi klasifikasi teks.

3.4 Penanganan Ketidakseimbangan Kelas (SMOTE)

Distribusi awal data sentimen pengguna aplikasi Gojek menunjukkan ketidakseimbangan kelas yang signifikan, sebagaimana terlihat pada Gambar 3.1 (a). Kelas sentimen positif mendominasi dataset dengan jumlah lebih dari 160.000 ulasan, sementara kelas negatif berada di sekitar 55.000 dan kelas netral hanya sekitar 12.000. Ketimpangan ini berdampak pada performa model klasifikasi, terutama pada rendahnya presisi dan recall untuk kelas minoritas (netral), yang terbukti pada hasil pengujian awal model *Decision Tree*.

Untuk mengatasi masalah ini, diterapkan teknik Synthetic Minority Over-sampling Technique (SMOTE) setelah proses ekstraksi fitur TF-IDF. Hasil SMOTE terlihat pada Gambar 3.1 (b), di mana seluruh kelas positif, negatif, dan netral telah diseimbangkan menjadi sekitar 600 sampel untuk masing-masing label. Ini menciptakan kondisi yang lebih adil dalam proses pelatihan model karena semua kelas diwakili dalam jumlah yang setara. Distribusi Label sebelum dan setelah SMOTE ditunjukkan pada Gambar 2.



Gambar 2 Distribusi Label (a) Sebelum SMOTE, (b) Setelah SMOTE

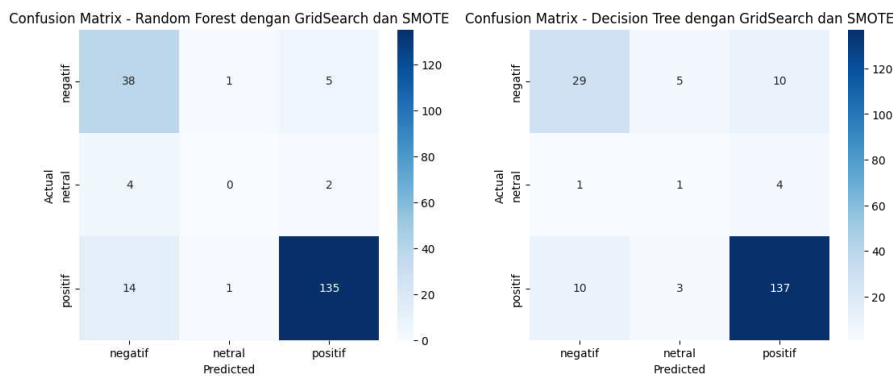
3.5 Pengujian dan Evaluasi Model Setelah SMOTE dan GridSearch

Setelah dilakukan proses penyeimbangan data menggunakan SMOTE dan optimasi *hyperparameter* dengan *GridSearchCV*, model *Decision Tree* menunjukkan akurasi sebesar 83,5% dengan nilai *Precision* 0,84, *Recall* 0,835, dan *F1-score* 0,84. Pada kelas positif, model menunjukkan performa tinggi (*Precision* 0,91, *Recall* 0,91, *F1-score* 0,91), menandakan kemampuannya yang sangat baik dalam mengenali ulasan bernada positif. Untuk kelas negatif, model menunjukkan hasil moderat (*Precision* 0,72, *Recall* 0,66, *F1-score* 0,69). Namun, pada kelas netral, kinerjanya masih sangat rendah, dengan *Precision* 0,11, *Recall* 0,17, dan *F1-score* 0,13, hanya mampu mengenali 1 dari 6 data aktual dengan benar. Ini menunjukkan bahwa walaupun SMOTE telah diterapkan, *Decision Tree* tetap kesulitan dalam menggeneralisasi kelas minoritas yang sangat terbatas. Hasil *Grid Search* ditunjukkan pada Tabel 3.

Tabel 3 Hasil Grid Search

Kategori	Parameter / Matrix	Nilai
Parameter Terbaik	<i>criterion</i>	gini
	<i>max_depth</i>	10
	<i>min_sample_leaf</i>	2
	<i>min_sample_split</i>	15
Skor Terbaik	<i>F1-Score (weighted) Cross-Validation</i>	0.7974

Sebagai pembandingan, model *Random Forest* dengan konfigurasi serupa (SMOTE + *GridSearch*) menunjukkan performa yang lebih tinggi, dengan akurasi 86,5%, *Precision* 0,86, *Recall* 0,865, dan *F1-score* 0,86. Pada kelas positif, model mempertahankan performa unggul (*Precision* 0,95, *Recall* 0,90, *F1-score* 0,92). Untuk kelas negatif, terjadi peningkatan signifikan dibanding *Decision Tree*, dengan *Recall* sebesar 0,86. Namun demikian, performa pada kelas netral tetap nihil (*Precision* 0,00, *Recall* 0,00, *F1-score* 0,00), menunjukkan bahwa meskipun algoritma lebih kuat, masih dibutuhkan pendekatan khusus untuk menangani jumlah sampel minoritas yang sangat kecil. Untuk meningkatkan performa model, dilakukan tuning *hyperparameter* menggunakan teknik *GridSearchCV*. Hasil Pengujian dan Evaluasi Model (*Confusion Matrix*) ditunjukkan pada Gambar 3 dan Tabel 3.



Gambar 3. Confusion Matrix: (a) Decision Tree + Grid Search dan SMOTE, (b) Random Forest + Grid Search dan SMOTE

Tabel 3. Pengujian dan Evaluasi Model (*Confusion Matrix*)

<i>Matrix</i>	<i>Decision Tree + Grid Search</i>	<i>Random Forest + Grid Search</i>
<i>Accuracy</i>	0.835	0.865
<i>Precision</i>	0,8432	0,8623
<i>Recall</i>	0,835	0,865
<i>F1-Score</i>	0,8438	0,8606

Hasil tersebut menunjukkan bahwa *Random Forest* lebih *robust* dalam menangani data yang telah diseimbangkan, terutama untuk kelas mayoritas dan kelas dengan distribusi menengah seperti negatif. Namun, kelas netral tetap menjadi tantangan, memperkuat kesimpulan bahwa strategi balancing harus dikombinasikan dengan pendekatan lain seperti data augmentation, model *hybrid*, atau *threshold adjustment* jika tujuan klasifikasi mencakup representasi adil semua kelas.

3.6 Analisis Perbandingan

Tabel 4 menyajikan perbandingan kinerja antara Decision Tree dan Random Forest. Berdasarkan Tabel 4, dapat disimpulkan bahwa *Random Forest* merupakan model yang lebih efektif untuk klasifikasi sentimen ulasan aplikasi Gojek pada dataset ini. Keunggulannya dalam menggabungkan banyak pohon keputusan menghasilkan model yang lebih akurat dan tahan terhadap *noise* atau *outlier* dibandingkan dengan *Decision Tree* tunggal.

Tabel 4. Perbandingan Antara *Decision Tree* dan *Random Forest*

Model	Akurasi	<i>F1-score</i>	Kelebihan Utama
Decision Tree	85,5%	84,1%	Interpretasi sederhana, cepat dilatih
Random Forest	86,5%	86,1%	Akurasi lebih tinggi, stabil, minim overfitting

4. Kesimpulan

Penelitian ini berhasil menerapkan algoritma *Decision Tree* yang dioptimasi menggunakan *GridSearchCV*, serta pendekatan penyeimbangan data dengan SMOTE, untuk klasifikasi sentimen ulasan pengguna aplikasi Gojek. Hasil menunjukkan bahwa model *Decision Tree* mampu mengklasifikasikan sentimen positif dengan baik (*F1-score* 0,91), dan cukup stabil pada sentimen negatif. Namun, kinerja terhadap kelas *netral* masih sangat rendah (*F1-score* 0,13), meskipun data telah diseimbangkan, yang menunjukkan bahwa pendekatan ini belum cukup efektif untuk menangani kelas minoritas secara optimal.

Sebagai pembanding, model *Random Forest* menunjukkan performa yang lebih baik secara keseluruhan (Akurasi 86,5% dan *F1-score* 0,86), meskipun masih menghadapi tantangan dalam klasifikasi sentimen netral. Hal ini menegaskan bahwa ketidakseimbangan kelas dan kompleksitas pola dalam data memerlukan strategi yang lebih komprehensif, baik dari sisi pemilihan model maupun teknik penyeimbangan data.

Dengan demikian, dapat disimpulkan bahwa kombinasi TF-IDF dan *Decision Tree* dapat digunakan untuk memahami sentimen ulasan secara otomatis, khususnya untuk kelas mayoritas. Namun, untuk mendapatkan hasil yang lebih merata pada semua kelas, terutama kelas minoritas, diperlukan model yang lebih kompleks seperti *Random Forest* dan penerapan teknik *balancing* yang lebih lanjut. Penelitian ini memberikan kontribusi awal dalam eksplorasi klasifikasi sentimen aplikasi lokal di Indonesia serta membuka peluang untuk pengembangan sistem evaluasi layanan berbasis data secara lebih adil dan akurat.

Deklarasi

Kontribusi Penulis. Semua penulis berkontribusi secara bersama-sama dengan kontributor utama dalam artikel ini. Semua penulis membaca dan menyetujui versi akhir dari artikel yang diajukan. Pernyataan Pendanaan. Tidak ada penulis yang menerima dana atau hibah dari lembaga atau badan pendanaan untuk penelitian ini.

Konflik Kepentingan. Penulis menyatakan tidak ada konflik kepentingan.

Informasi Tambahan. Tidak ada informasi tambahan dalam artikel ini.

Daftar Pustaka

- [1] Gojek, "Tentang Gojek." Accessed: May 26, 2025. [Online]. Available: <https://www.gojek.com/en-id/about>
- [2] Nikhil Sanjay Suryawanshi, "Sentiment Analysis With Machine Learning And Deep Learning: A Survey Of Techniques And Applications," *International Journal Of Science And Research Archive*, Vol. 12, No. 2, Pp. 005–015, Jul. 2024, Doi: 10.30574/Ijsra.2024.12.2.1205.
- [3] P. S. Ghatora, S. E. Hosseini, S. Pervez, M. J. Iqbal, And N. Shaukat, "Sentiment Analysis Of Product Reviews Using Machine Learning And Pre-Trained Llm," *Big Data And Cognitive Computing*, Vol. 8, No. 12, P. 199, Dec. 2024, Doi: 10.3390/Bdce8120199.
- [4] W. Zhang, X. Li, Y. Deng, L. Bing, And W. Lam, "A Survey On Aspect-Based Sentiment Analysis: Tasks, Methods, And Challenges," Vol. 2, P. 002, Mar. 2022, Doi: <https://doi.org/10.48550/Arxiv.2203.01054>.
- [5] A. Mabrouk, R. P. D. Redondo, And M. Kayed, "Deep Learning-Based Sentiment Classification: A Comparative Survey," *Ieee Access*, Vol. 8, Pp. 85616–85638, 2020, Doi: 10.1109/Access.2020.2992013.
- [6] Hermanto, A. Y. Kuntoro, T. Asra, E. B. Pratama, L. Effendi, And R. Ocanitra, "Gojek And Grab User Sentiment Analysis On Google Play Using Naive Bayes Algorithm And Support Vector Machine Based Smote Technique," *J Phys Conf Ser*, Vol. 1641, No. 1, P. 012102, Nov. 2020, Doi: 10.1088/1742-6596/1641/1/012102.
- [7] N. Azwanti And N. E. Putra, "Analisis Kepuasan Customer Pada Sdtechnology Computer Dengan Algoritma Decision Tree," *Jurnal Desain Dan Analisis Teknologi*, Vol. 3, No. 2, Pp. 002–012, 2024.
- [8] C. Cahyaningtyas, Y. Nataliani, And I. R. Widiyari, "Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis Smote," *Aiti*, Vol. 18, No. 2, Pp. 173–184, Nov. 2021, Doi: 10.24246/Aiti.V18i2.173-184.
- [9] T. M. Lab, "Data Cleaning And Preprocessing With Python," Medium.
- [10] W. McKinney, *Python For Data Analysis*, 3rd Edition, 3rd Ed. O'reilly Media, Inc., 2022.
- [11] D. Rifaldi, Abdul Fadlil, And Herman, "Teknik Preprocessing Pada Text Mining Menggunakan Data Tweet 'Mental Health,'" *Decode: Jurnal Pendidikan Teknologi Informasi*, Vol. 3, No. 2, Pp. 161–171, Apr. 2023, Doi: 10.51454/Decode.V3i2.131.

-
- [12] M. Nurjannah, H. Hamdani, And I. F. Astuti, “Penerapan Algoritma Term Frequency-Inverse Document Frequency (Tf-Idf) Untuk Text Mining,” *Jurnal Ilmiah Ilmu Komputer, Informatika Mulawarman*, Vol. 8, No. 3, Pp. 001–004, 2020, Doi: [Http://Dx.Doi.Org/10.30872/Jim.V8i3.113](http://dx.doi.org/10.30872/Jim.V8i3.113).
- [13] P. R. Albin, R. Rudiman, And N. A. Verdikha, “Metode Pembobotan Tf-Idf Untuk Klasifikasi Teks Quick Count Pemilihan Wakil Presiden Indonesia 2024 Pada X Twitter Dengan Metode Svm,” *Jurnal Teknologi Informasi : Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, Vol. 18, No. 2, Pp. 127–138, 2024.
- [14] L. Ja’far And L. B. Handoko, “Optimasi Analisis Sentimen Lowongan Kerja Di Twitter Dengan Xgboost-Vader Dan Evaluasi Smote Borderline,” *Jurnal Pendidikan Dan Teknologi Indonesia*, Vol. 5, No. 4, Pp. 999–1010, Apr. 2025, Doi: [10.52436/1.Jpti.741](https://doi.org/10.52436/1.Jpti.741).
- [15] R. Aryanti, T. Misriati, And R. Hidayat, “Klik: Kajian Ilmiah Informatika Dan Komputer Klasifikasi Risiko Kesehatan Ibu Hamil Menggunakan Random Oversampling Untuk Mengatasi Ketidakseimbangan Data,” *Media Online*, Vol. 3, No. 5, Pp. 409–416, 2023, [Online]. Available: [Https://Djournals.Com/Klik](https://djournals.com/klik)
- [16] I. H. Witten, E. Frank, And M. A. Hall, *Data Mining: Practical Machine Learning Tools And Techniques*, 11th Ed. Elsevier, 2011. Doi: [10.1016/C2009-0-19715-5](https://doi.org/10.1016/C2009-0-19715-5).
- [17] C. Cahyaningtyas, Y. Nataliani, And I. R. Widiyari, “Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis Smote,” *Aiti: Jurnal Teknologi Informasi*, Vol. 18, No. Agustus, Pp. 173–184, 2021.
- [18] K. Karfindo, R. Turaina, And R. Saputra, “Optimalisasi Klasifikasi Umpan Balik Mahasiswa Terhadap Layanan Kampus Dengan Sinergi Random Forest Dan Smote,” *Jurnal Nasional Komputasi Dan Teknologi Informasi (Jnkti)*, Vol. 6, No. 6, Pp. 820–827, Jan. 2024, Doi: [10.32672/Jnkti.V6i6.7269](https://doi.org/10.32672/jnkti.V6i6.7269).
- [19] Sebastian. Raschka And Vahid. Mirajalili, *Python Machine Learning : Machine Learning And Deep Learning With Python, Scikit-Learn, And Tensorflow*, Second Edition. Packt Publishing, 2018.
- [20] R. Gustriansyah, N. Suhandi, S. Puspasari, And A. Sanmorino, “Machine Learning Method To Predict The Toddlers’ Nutritional Status,” *Jurnal Infotel*, Vol. 16, No. 1, Jan. 2024, Doi: [10.20895/Infotel.V15i4.988](https://doi.org/10.20895/infotel.V15i4.988).