

Optimasi Model Machine Learning Menggunakan Teknik SMOTE pada Analisis Sentimen Pengguna RedBus

Arman Ramadhani^{1*}, Riska Aryanti², Sarifah Agustiani³

^{1,2,3}Informatika, Universitas Bina Sarana Informatika, Indonesia

^{1*}15210200@bsi.ac.id, ²riska.rts@bsi.ac.id, ³sarifah.sgu@bsi.ac.id

Abstrak: Perkembangan teknologi digital semakin memudahkan masyarakat dalam memenuhi kebutuhan transportasi, salah satunya melalui aplikasi pemesanan tiket bus seperti *RedBus*. Aplikasi ini menghadirkan layanan pemesanan secara praktis, namun ulasan pengguna yang semakin banyak di *Google Play Store* bersifat tidak terstruktur sehingga memerlukan analisis lebih lanjut untuk menilai kualitas layanan secara objektif. Penelitian ini bertujuan untuk mengklasifikasikan sentimen kepuasan pengguna aplikasi *RedBus* dengan memanfaatkan algoritma *Naïve Bayes* dan *Random Forest*. Untuk mengatasi masalah ketidakseimbangan data, digunakan teknik *Synthetic Minority Over-sampling Technique (SMOTE)*. Data yang digunakan berjumlah 2.000 ulasan yang dikumpulkan melalui metode *web scraping*, kemudian diproses melalui tahapan preprocessing yang meliputi data *cleaning*, *cleansing*, *case folding*, *tokenization*, *stopword*, dan *stemming*. Selanjutnya, data diberi label kepuasan berdasarkan rating, lalu dikonversi menjadi fitur numerik dengan metode TF-IDF. Data dibagi menjadi 90% data latih dan 10% data uji agar dapat dievaluasi secara menyeluruh. Hasil pengujian menunjukkan bahwa algoritma *Naïve Bayes* menghasilkan akurasi 91%, *precision* 97%, *recall* 89%, dan *F1-score* 92%. Sementara itu, algoritma *Random Forest* memperoleh akurasi 90%, *precision* 94%, *recall* 90%, dan *F1-score* 92%. Keunggulan *Naïve Bayes* terlihat pada nilai *precision* yang tinggi, menunjukkan kemampuannya dalam meminimalkan kesalahan klasifikasi positif palsu. Kesimpulannya, penerapan *Naïve Bayes* dengan dukungan *SMOTE* dinilai lebih optimal dalam mengklasifikasikan sentimen ulasan, sehingga dapat menjadi masukan bagi pengembang *RedBus* dalam meningkatkan kualitas layanan dan kepuasan pengguna.

Kata Kunci: Analisis Sentimen; *RedBus*; *Machine Learning*; *Naïve Bayes*; *Random Forest*; *SMOTE*

Abstract: The development of digital technology has made it easier for people to meet their transportation needs, one of which is through bus ticket booking applications such as *RedBus*. This application provides practical booking services, yet the increasing number of user reviews on the *Google Play Store* is unstructured, requiring further analysis to objectively assess service quality. This study aims to classify user satisfaction sentiment in the *RedBus* application using the *Naïve Bayes* and *Random Forest* algorithms. To address the data imbalance issue, the *Synthetic Minority Over-sampling Technique (SMOTE)* was applied. A total of 2,000 reviews were collected through *web scraping* and then processed through several preprocessing stages, including data *cleaning*,

cleansing, case folding, tokenization, stopword removal, and stemming. The reviews were labeled based on ratings, transformed into numerical features using the TF-IDF method, and divided into 90% training data and 10% testing data for evaluation. The results show that the Naïve Bayes algorithm achieved an accuracy of 91%, precision of 97%, recall of 89%, and F1-score of 92%. Meanwhile, the Random Forest algorithm obtained an accuracy of 90%, precision of 94%, recall of 90%, and F1-score of 92%. The advantage of Naïve Bayes is reflected in its higher precision value, indicating its ability to minimize false positive classifications. In conclusion, the implementation of Naïve Bayes with SMOTE is considered more optimal in classifying review sentiments, providing useful insights for RedBus developers to improve service quality and enhance user satisfaction.

Keywords: Sentiment Analysis; RedBus; Machine Learning; Naïve Bayes; Random Forest; SMOTE

1. PENDAHULUAN

Saat ini, kemajuan teknologi informasi sangat penting dan bermanfaat di segala bidang, terutama dalam infrastruktur, mulai dari konstruksi hingga jalan, sehingga meningkatkan kesejahteraan dan kualitas hidup kota. Transportasi umum, juga dikenal sebagai transportasi massal, adalah cara bagi mereka yang tidak memiliki kendaraan sendiri untuk bepergian [1]. Transportasi umum terbagi menjadi dua kategori, yaitu transportasi konvensional (*offline*) dan *online*. Yang pertama digunakan oleh orang-orang tanpa perlu menggunakan aplikasi untuk memesan, sedangkan yang kedua digunakan oleh orang-orang dengan aplikasi untuk memesan [2].

Saat ini, masyarakat menuntut kemudahan yang lebih tinggi dalam mengakses layanan transportasi umum, khususnya terkait proses pemesanan tiket yang praktis dan cepat. Kehadiran aplikasi pemesanan tiket bus daring, seperti *RedBus*, menjadi solusi yang mampu menjawab kebutuhan tersebut dengan menyediakan sistem yang lebih efisien untuk melakukan pembelian tiket, memantau rute perjalanan, serta mengatur aktivitas perjalanan secara digital [3]. Sejak kehadirannya, *RedBus* sukses menarik perhatian luas dari masyarakat global. Jumlah unduhan yang telah melampaui 50 juta kali di *Google Play Store* menjadi bukti nyata tingginya minat serta kepercayaan pengguna terhadap aplikasi ini. Pencapaian tersebut menunjukkan bahwa *RedBus* kini dipandang sebagai salah satu platform terbaik dalam memenuhi kebutuhan perjalanan dengan transportasi umum, khususnya layanan bus.

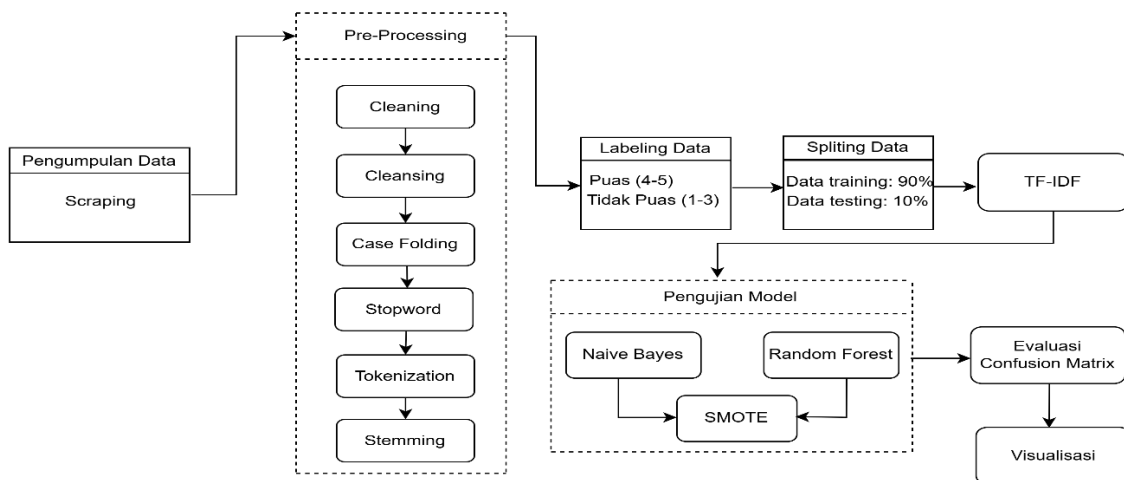
Sebagai platform pemesanan tiket bus secara *online*, *RedBus* menerima berbagai ulasan dari penggunaannya melalui *Google Play Store*. Ulasan dan penilaian tersebut tidak hanya mencerminkan tingkat kepuasan maupun ketidakpuasan pengguna, tetapi juga berperan penting sebagai masukan bagi pengembang dalam meningkatkan kualitas layanan. Namun demikian, jumlah ulasan yang sangat banyak serta formatnya yang tidak terstruktur menimbulkan tantangan, karena penafsiran secara manual menjadi kurang efektif. Oleh karena itu, diperlukan suatu pendekatan yang mampu mengolah serta memahami data ulasan secara komprehensif, salah satunya melalui penerapan analisis sentimen [4].

Menurut [5] Analisis sentimen merupakan sebuah metode yang digunakan untuk mengekstraksi serta mengolah data opini sehingga dapat dianalisis secara otomatis. Oleh karena itu, penerapan analisis sentimen pada aplikasi *RedBus* dapat digunakan untuk mengidentifikasi tingkat kepuasan pengguna terhadap fitur maupun layanan yang tersedia [6]. Tujuan dari penelitian ini untuk mengklasifikasi sentimen ulasan pengguna terhadap aplikasi *RedBus* berdasarkan rating [7]. Dalam penelitian ini diterapkan dua algoritma, yakni *Naïve Bayes* dan *Random Forest*, untuk melihat perbandingan kinerjanya dalam menghasilkan klasifikasi yang optimal. Penerapan teknik *SMOTE* turut dilakukan sebagai

strategi penyeimbang data, sehingga menjadikan penelitian ini berbeda dibandingkan dengan penelitian terdahulu, seperti pada penelitian [8] pendekatan yang digunakan terbatas pada satu algoritma, yakni *K-Nearest Neighbors* (KNN), tanpa melibatkan teknik *SMOTE* atau metode tambahan lainnya.

2. METODE PENELITIAN

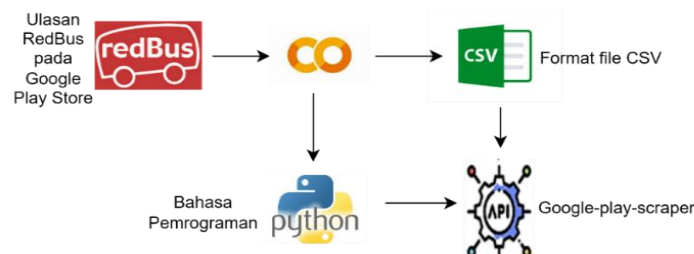
Penelitian ini menggunakan data ulasan aplikasi *RedBus* dari *Google Play Store* yang diperoleh melalui *scraping*. Data kemudian diberi label berdasarkan rating, diproses melalui tahap *pre-processing* (*cleaning, case folding, tokenization, stopword, dan stemming*), lalu dibobotkan dengan *TF-IDF*. Proses klasifikasi dilakukan menggunakan *Naive Bayes* dan *Random Forest* dengan bantuan *SMOTE* untuk menangani ketidakseimbangan data. Hasil kinerja model dievaluasi menggunakan *Confusion Matrix* dan divisualisasikan, dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Pengumpulan Data

Tahap pertama yaitu pengumpulan data yang dilakukan secara otomatis dengan memanfaatkan bahasa pemrograman *python* melalui teknik *web scraping*. Seluruh ulasan yang berhasil dikumpulkan kemudian disimpan dalam format *Comma Separated Values* (CSV) agar lebih mudah diolah dan dianalisis pada tahap berikutnya. Rentang waktu pengambilan data dimulai pada 31 Desember 2023 hingga 13 Juni 2025 dengan jumlah sebanyak 2000 data, seperti pada Gambar 2.



Gambar 2. Pengumpulan Data

Pre-Processing

Tahapan berikutnya adalah *Pre-Processing* yang terdiri dari *cleaning, cleansing, case folding, stopword, tokenization, dan stemming*. *Pre-Processing* data ini bertujuan untuk

membersihkan data dari noise dan kesalahan, serta mengubah data menjadi bentuk yang terstruktur dan siap untuk diproses. Tahapan *pre-processing* yaitu :

Cleaning, merupakan tahapan penting dalam pengolahan data yang bertujuan untuk memastikan kualitas data agar layak digunakan pada proses analisis selanjutnya. Proses ini mencakup identifikasi dan penanganan nilai yang hilang (*missing value*), penghapusan data yang tidak konsisten atau mengandung *noise* maupun outlier, pencarian serta eliminasi data duplikat, hingga penanganan data yang bersifat tidak lengkap. Dengan melakukan data cleaning, keakuratan hasil analisis dapat ditingkatkan karena data yang digunakan lebih bersih, terstruktur, dan representatif terhadap kondisi sebenarnya [9].

Cleansing, merupakan proses karakter yang tidak relevan seperti tanda baca, *emoticon*, *hashtag*, URL, simbol khusus, dan angka dihapus agar teks lebih bersih dan siap untuk dianalisis [10].

Case Folding merupakan proses mengubah seluruh karakter dalam teks menjadi huruf kecil (*lowercase*) [11].

Stopword adalah proses menghapus kata-kata umum yang tidak memiliki makna signifikan dalam teks [12].

Tokenization merupakan proses pembersihan teks kemudian diuraikan menjadi unit terkecil yang disebut token. Proses tokenisasi dilakukan agar kata maupun kalimat dapat dipisahkan secara sistematis, sehingga memudahkan analisis lebih lanjut menggunakan teknik *Natural Language Processing* (NLP), termasuk di dalamnya analisis sentimen [13].

Stemming adalah proses dimana kata-kata diolah dengan cara menghapus imbuhan seperti awalan atau akhiran sehingga diperoleh bentuk dasar atau 'akar kata'. Tujuannya adalah untuk menyamakan variasi morfologis dari kata-kata yang berbeda namun memiliki makna serupa, sehingga dapat diperlakukan sebagai satu bentuk kata yang sama dalam analisis [14].

Labeling Data

Tahap berikutnya adalah *labeling data*, dalam analisis sentimen, *labeling data* dilakukan untuk mengelompokkan ulasan ke dalam kategori tertentu, misalnya positif dan negatif. Proses ini memudahkan algoritma pembelajaran mesin mengenali pola sesuai kelas yang sudah ditentukan. Pada penelitian ini, ulasan dengan rating 1–3 negatif, sedangkan rating 4–5 positif [15].

Splitting Data

Selanjutnya *splitting data*, yaitu data yang sudah diberi label sebelumnya akan dilakukan pembagian data secara acak, pembagian pada penelitian ini adalah 90% untuk data latih dan 10% untuk data uji.

TF-IDF

Pembobotan dengan TF-IDF. TF-IDF merupakan salah satu metode pembobotan yang paling banyak digunakan dalam pemrosesan teks, karena dikenal memiliki tingkat akurasi yang baik serta kemampuan recall yang tinggi. Teknik ini menggabungkan konsep *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Dalam hal ini, jumlah kemunculan suatu kata dalam sebuah dokumen disebut sebagai frekuensi [16]. Rumus dari TF-IDF yaitu:

a. *Term Frequency* (TF):

TF (term, document) adalah frekuensi kemunculan term dalam dokumen tertentu.

b. *Inverse Document Frequency* (IDF):

$$IDF(\text{term}) = \frac{\text{Total dokumen}}{\text{Jumlah dokumen yang mengandung term}}$$

yaitu:

1. Total dokumen adalah jumlah total dokumen dalam koneksi.

2. Jumlah dokumen yang mengandung term adalah jumlah dokumen di dalam koleksi yang mengandung term tersebut.
- c. *TF-IDF (Term Frequency – Inverse Document Frequency)*:
 $TF-IDF (term, document) = TF (term, document) \times IDF (term)$
Gabungan dari TF dan IDF untuk menghitung bobot atau nilai penting suatu *term* dalam sebuah dokumen dalam konteks keseluruhan koleksi dokumen.

Pengujian Model

Pengujian model, pada penelitian ini menggunakan dua model algoritma yaitu *naïve bayes* dan *random forest* serta penggunaan *SMOTE* untuk menangani data yang tidak seimbang dan meningkatkan akurasi.

Evaluasi Confusion Matrix

Confusion matrix merupakan salah satu metode evaluasi yang digunakan untuk menilai kinerja model prediksi. Melalui metode ini, dapat dilakukan perbandingan antara nilai aktual dengan nilai hasil prediksi yang dihasilkan oleh model [17]. *Confusion Matrix* berfungsi untuk menggambarkan kinerja suatu algoritma serta digunakan sebagai dasar dalam menghitung performa model dalam melakukan prediksi terhadap data aktual. Terdapat sejumlah parameter evaluasi pada *confusion matrix* yang berhubungan langsung dengan pengukuran kinerja model klasifikasi:

a. Accuracy

Parameter penilaian untuk mengubah akurasi dan tingkat kepercayaan model selama proses klasifikasi, dan ini adalah rumusnya:

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

b. Precision

Merupakan parameter penilaian yang menghitung nilai rata-rata *precision* dari data hasil klasifikasi yaitu jumlah data yang benar antara nilai sebenarnya dengan hasil prediksi model yang dapat dihitung dengan rumus seperti pada persamaan berikut:

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

c. Recall

Parameter penilaian yang didapat dari jumlah data verifikasi melalui rumus pada persamaan, serta data yang keluar dari hasil klasifikasi. Berikut adalah rumusnya:

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

d. F1-Score

Matrix evaluasi *F1-Score* digunakan untuk menurunkan tingkat partisipasi angka kerja dalam model klasifikasi, dan ini rumusnya:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Visualisasi

Visualisasi dalam analisis sentimen berperan penting untuk menyajikan hasil pengolahan data teks agar lebih mudah dipahami. Visualisasi pada penelitian ini ditampilkan dalam bentuk *wordcloud* dan juga diagram batang. *wordcloud* merupakan representasi visual dari sekumpulan kata, di mana ukuran setiap kata ditampilkan secara proporsional sesuai dengan frekuensi kemunculannya dalam suatu teks atau dataset. Hasil analisis melalui visualisasi *wordcloud* diharapkan mampu memberikan gambaran yang lebih jelas mengenai kata-kata atau topik yang paling menonjol pada masing-masing

kategori sentimen. Dengan cara ini, dapat diketahui ciri khas ulasan yang berkaitan dengan fitur layanan maupun aspek di luar layanan [18], begitupun juga dengan diagram batang yang memberikan informasi kalimat terbanyak yang sering muncul pada masing-masing kelas sentimen.

3. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil penelitian, termasuk proses pre-processing, penerapan strategi *SMOTE*, dan evaluasi kinerja model. Metode yang digunakan bertujuan untuk membandingkan kinerja algoritma *Naive Bayes* dan *Random Forest* dalam mengklasifikasikan kepuasan pengguna berdasarkan ulasan di *Google Play Store*. Pembahasan berfokus pada nilai akurasi, *precision*, *recall*, dan *f1-score* untuk mengevaluasi kemampuan model untuk mengidentifikasi sentimen positif dan negatif serta memberikan gambaran objektif tentang tingkat kepuasan pengguna aplikasi *RedBus*.

Pre-Processing Data

Pada tahap *pre-processing*, dataset yang telah diperoleh melalui proses scraping dipersiapkan terlebih dahulu. Proses *pre-processing* ini meliputi beberapa tahapan, antara lain:

a. Cleaning

Pada tahap ini, ulasan pengguna yang bersifat duplikat akan dihapus untuk mencegah berkurangnya tingkat akurasi model. Selain itu, data kosong yang tidak mengandung informasi relevan juga dihilangkan karena tidak dapat diproses lebih lanjut. Dataset pada penelitian ini yang awalnya berjumlah 2000 ulasan mengalami perubahan menjadi 1617 setelah melakukan *cleaning* data.

b. Cleansing

Hasil tahapan *cleansing* pada penelitian ini dapat dilihat pada tabel 1.

Tabel 1. Hasil *Cleansing*

Sebelum <i>Pre-Processing</i>	Sesudah <i>Pre-Processing</i>
ERROR MULU SETIAP PILIH KURSI 🗑️🗑️	ERROR MULU SETIAP PILIH KURSI
saya selalu terbantu dengan layanan ini, keren RedBus 😊😊😊	saya selalu terbantu dengan layanan ini keren RedBus
aplikasi jos pokoknya ketik langsung jadi 😊😊	aplikasi jos pokoknya ketik langsung jadi
refund dana susah amat !!!!	refund dana susah amat

c. Case Folding

Hasil tahapan *case folding* pada penelitian ini dapat dilihat pada tabel 2.

Tabel 2. Hasil *Case Folding*

Sebelum <i>Pre-Processing</i>	Sesudah <i>Pre-Processing</i>
ERROR MULU SETIAP PILIH KURSI	error mulu setiap pilih kursi
saya selalu terbantu dengan layanan ini keren RedBus	saya selalu terbantu dengan layanan ini keren redbus

aplikasi jos pokoknya ketik langsung
jadi

aplikasi jos pokoknya ketik
langsung jadi

refund dana susah amat

refund dana susah amat

d. *Tokenization*

Hasil tahapan *tokenization* pada penelitian ini dapat dilihat pada tabel 3.

Tabel 3. Hasil *Tokenization*

Sebelum <i>Pre-Processing</i>	Sesudah <i>Pre-Processing</i>
error mulu setiap pilih kursi	["error", "mulu", "setiap", "pilih", "kursi"]
saya selalu terbantu dengan layanan ini keren redbus	["saya", "selalu", "terbantu", "dengan", "layanan", "ini", "keren", "redbus"]
aplikasi jos pokoknya ketik langsung jadi	["aplikasi", "jos", "pokoknya", "ketik", "langsung", "jadi"]
refund dana susah amat	["refund", "dana", "susah", "amat"]

e. *Stopword*

Hasil tahapan *stopword* pada penelitian ini dapat dilihat pada tabel 4.

Tabel 4. Hasil *Stopword*

Sebelum <i>Pre-Processing</i>	Sesudah <i>Pre-Processing</i>
["error", "mulu", "setiap", "pilih", "kursi"]	["error", "mulu", "pilih", "kursi"]
["saya", "selalu", "terbantu", "dengan", "layanan", "ini", "keren", "redbus"]	["selalu", "terbantu", "layanan", "keren", "redbus"]
["aplikasi", "jos", "pokoknya", "ketik", "langsung", "jadi"]	["aplikasi", "jos", "ketik", "langsung", "jadi"]
["refund", "dana", "susah", "amat"]	["refund", "dana", "susah"]

f. *Stemming*

Hasil tahapan *stemming* pada penelitian ini dapat dilihat pada tabel 5.

Tabel 5. Hasil *Stemming*

Sebelum <i>Pre-Processing</i>	Sesudah <i>Pre-Processing</i>
["error", "mulu", "pilih", "kursi"]	["error", "mulu", "pilih", "kursi"]
["selalu", "terbantu", "layanan", "keren", "redbus"]	["selalu", "bantu", "layan", "keren", "redbus"]

["aplikasi", "jos", "ketik", "langsung",
"jadi"]

["aplikasi", "jos", "ketik",
"langsung", "jadi"]

["refund", "dana", "susah"]

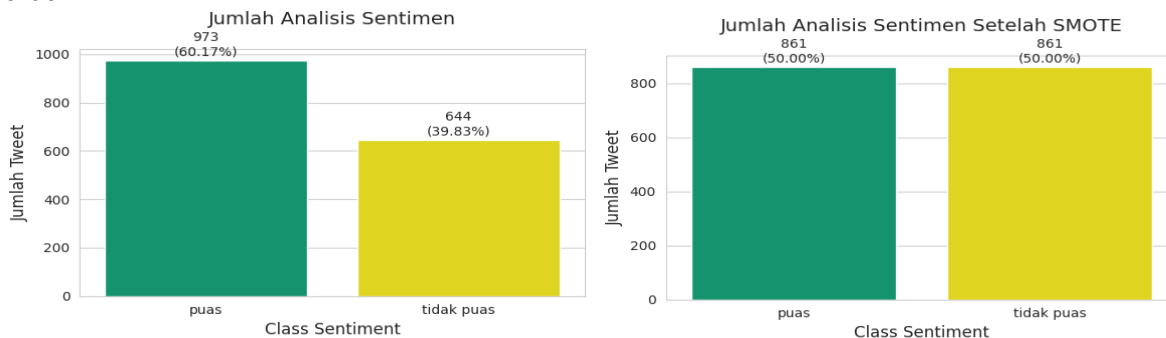
["refund", "dana", "susah"]

Labeling Data

Pada tahap labeling data, proses klasifikasi difokuskan pada dua kategori sentimen berdasarkan nilai rating. Rating dengan nilai 4–5 dikategorikan sebagai sentimen puas, sedangkan rating dengan nilai 1–3 dikategorikan sebagai sentimen tidak puas, seperti pada penelitian [15].

SMOTE

Penerapan metode *SMOTE* mampu mengatasi permasalahan ketidakseimbangan pada dataset sehingga dapat meningkatkan kualitas hasil klasifikasi [19]. Perbandingan dataset yang telah diberi label sebelum dan setelah penggunaan *SMOTE* dapat dilihat pada gambar 6 dan 7.



Gambar 3. Distribusi Sentimen sebelum dan sesudah *SMOTE*

Jika dilihat pada gambar 3. menunjukkan dari jumlah 1617 dataset pada *RedBus* berupa 973 ulasan puas dan 644 tidak puas. Kemudian setelah dilakukan penggunaan *SMOTE* untuk menangani ketidakseimbangan data menghasilkan 861 ulasan puas dan 861 ulasan tidak puas.

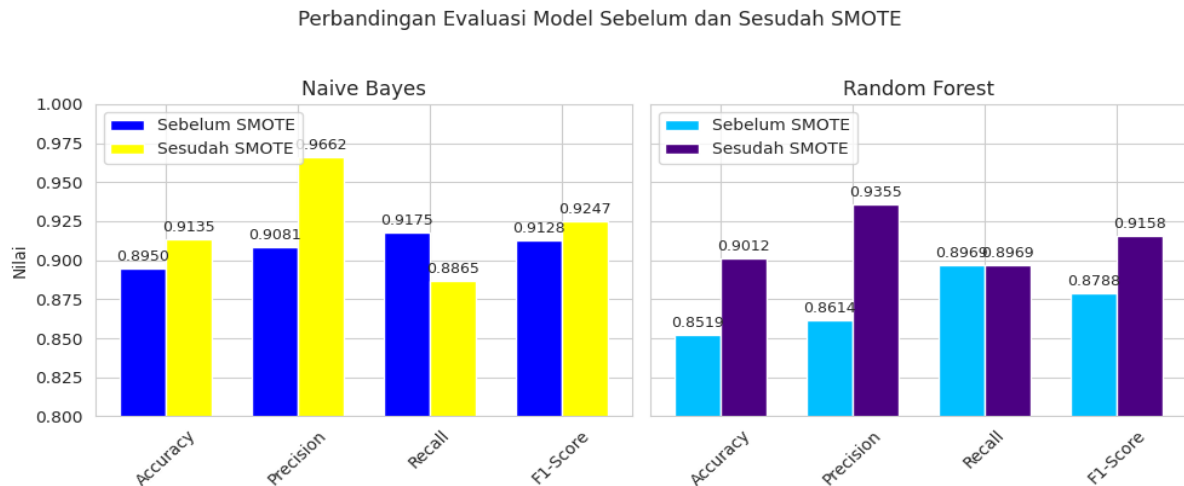
Pengujian Model

Pada tahap ini, dilakukan pengujian dengan menggunakan dua algoritma klasifikasi, yaitu *Naïve Bayes* dan *Random Forest*. Pengujian dilakukan dalam dua kondisi berbeda, yakni sebelum dilakukan penyeimbangan data dengan metode *SMOTE* dan sesudah penerapannya. Hal ini bertujuan untuk mengetahui perbedaan performa yang dihasilkan oleh kedua algoritma dalam menganalisis sentimen kepuasan pengguna aplikasi *RedBus*. Hasil dari pengujian ini memberikan gambaran yang jelas mengenai seberapa baik kedua algoritma mampu beradaptasi terhadap distribusi data yang seimbang maupun tidak seimbang. hasil pengujian akan dijelaskan dan dapat dilihat pada tabel 6 dan Gambar 4.

Tabel 6. Hasil *Naïve Bayes* dan *Random Forest* sebelum dan sesudah *SMOTE*

Algoritma	Accuracy	Precision	Recall	F1-Score
<i>Naïve Bayes</i>	90%	91%	92%	91%
<i>Naïve Bayes + SMOTE</i>	91%	97%	89%	92%
<i>Random Forest</i>	85%	86%	90%	88%

Random Forest + SMOTE 90% 94% 90% 92%

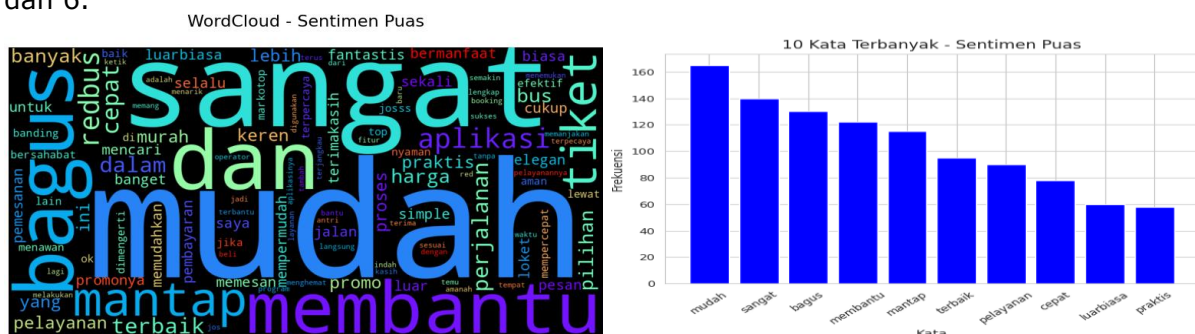


Gambar 4. Grafik perbandingan algoritma *Naïve Bayes* dan *Random Forest*

Berdasarkan pada Tabel 6 dan Gambar 4, terlihat bahwa penggunaan metode *SMOTE* memberikan dampak positif terhadap peningkatan kinerja kedua algoritma yang digunakan, yaitu *Naïve Bayes* dan *Random Forest*. Pada algoritma *Naïve Bayes*, akurasi yang semula 90% meningkat menjadi 91%, *precision* naik dari 91% menjadi 97%, *recall* turun dari 92% menjadi 89%, dan *F1-Score* dari 91% menjadi 92%. Sementara itu, pada algoritma *Random Forest* juga terlihat adanya perbaikan performa, di mana akurasi bertambah dari 85% menjadi 90%, *precision* meningkat dari 86% menjadi 94%, *recall* tetap pada angka 90%, dan *F1-Score* mengalami kenaikan dari 88% menjadi 92%. Temuan ini menunjukkan bahwa penerapan *SMOTE* berhasil memperbaiki ketidakseimbangan data, sehingga mampu menghasilkan klasifikasi yang lebih optimal dalam analisis sentimen kepuasan pengguna aplikasi *RedBus*.

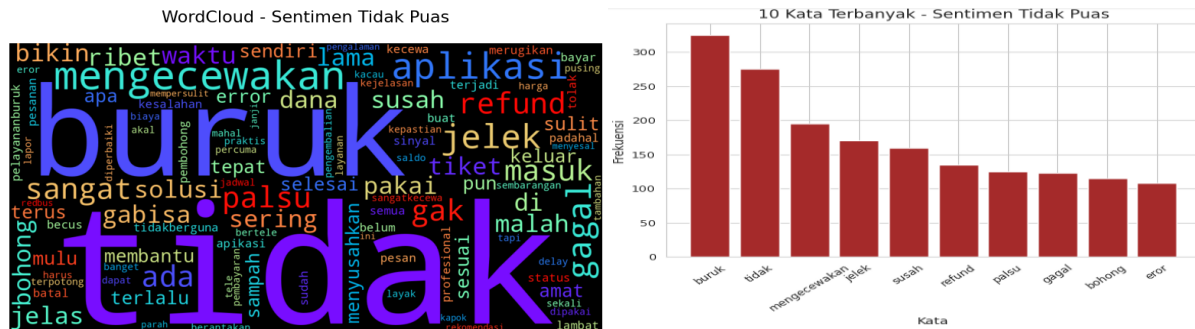
Visualisasi

Pada visualisasi ini akan ditampilkan dalam bentuk wordcloud dan diagram batang. Berdasarkan hasil analisis sentimen yang telah dilakukan menampilkan 10 kata yang sering muncul pada ulasan aplikasi *RedBus* pada masing-masing class sentiment yang ditampilkan dalam bentuk wordcloud dan diagram batang. Dapat dilihat pada gambar 5. dan 6.



Gambar 5. Visualisasi sentimen puas

Pada gambar 5. ditampilkan visualisasi dalam bentuk wordcloud dan digaram batang pada sentimen puas, dan terdapat 10 kata yang sering muncul pada ulasan puas, yaitu mudah, sangat, bagus, membantu, mantap, terbaik, pelayanan, cepat, luar biasa, dan praktis. Kalimat yang paling banyak diantara 10 kata tersebut adalah mudah.



Gambar 6. Visualisasi sentimen tidak puas

Pada gambar 6. ditampilkan visualisasi dalam bentuk wordcloud dan digaram batang pada sentimen tidak puas, dan terdapat 10 kata yang sering muncul pada ulasan puas, yaitu buruk, tidak, mengecewakan, jelek, susah, refund, palsu, gagal, luar, bohong, dan error. Kalimat yang paling banyak diantara 10 kata tersebut adalah buruk.

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa mayoritas pengguna aplikasi RedBus memberikan ulasan positif, terutama terkait kemudahan pemesanan tiket, kepercayaan terhadap layanan, dan tampilan aplikasi yang sederhana. Sebaliknya, ulasan negatif banyak muncul akibat kendala teknis seperti masalah *refund* dan *error* pada pemilihan kursi. Penerapan teknik *SMOTE* terbukti mampu mengoptimalkan performa model *Machine Learning* dalam analisis sentimen. Algoritma *Naïve Bayes* mencapai akurasi 91%, *precision* 97%, *recall* 89%, dan *F1-Score* 92%, sedangkan *Random Forest* memperoleh akurasi 90%, *precision* 94%, *recall* 90%, dan *F1-Score* 92%. Dibandingkan sebelum penerapan *SMOTE*, hasil ini menunjukkan peningkatan efektivitas klasifikasi, terutama pada algoritma *Naïve Bayes*. Visualisasi berupa *wordcloud* dan diagram batang turut memperkuat temuan dengan menampilkan kata yang paling sering muncul sebagai representasi tingkat kepuasan dan keluhan pengguna.

5. REFERENCES

- [1] H. Haryanda, M. F. Nasution, D. Hutabarat, A. Razzaq, and A. Syahputra, "Implementasi Metode Bubble Sort pada Aplikasi Pencarian Rute Berdasarkan Jarak Tempuh Transportasi Umum," *Blend Sains J. Tek.*, vol. 1, no. 3, pp. 213–219, 2023, doi: 10.56211/blendsains.v1i3.183.
- [2] D. Nugraha and D. Gustian, "Analisis Sentimen Penggunaan Aplikasi Transportasi Online Pada Ulasan Google Play Store Menggunakan Algoritma Svm (Support Vector Machine)," *SISMATIK (Seminar Nas. Sist. Inf. dan Manaj. Inform.)*, vol. 1, no. 1, pp. 326–335, 2023.
- [3] M. Shodikin, S. N. Saputra, and A. Prasetyo, "Aplikasi Pembelian Tiket Bus BUSSIN App di Wilayah Kediri," vol. 4, pp. 683–687, 2025.
- [4] M. H. Wicaksono, M. D. Purbolaksono, and S. Al Faraby, "Perbandingan Algoritma Machine Learning untuk Analisis Sentimen Berbasis Aspek pada Review Female

- Daily," *eProceedings Eng.*, vol. 10, no. 3, pp. 3591–3600, 2023.
- [5] S. A. Azzahra and A. Wibowo, "Analisis Ulasan Wisatawan," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, p. 737, 2020, doi: 10.25126/jtiik.202071907.
- [6] D. Nugraha and D. Gustian, "Analisis Sentimen Penggunaan Aplikasi Transportasi Online Pada Ulasan Google Play Store dengan Metode Naive Bayes Classifier," *J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 5, no. 1, pp. 326–335, 2024.
- [7] F. Fatmawati, B. Irawan, and A. Bahtiar, "Analisis Sentimen Pengguna Aplikasi Shejek Berdasarkan Ulasan Di Google Play Store Menggunakan Metode Naive Bayes," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 3, pp. 2976–2981, 2024, doi: 10.36040/jati.v8i3.9607.
- [8] J. Ilmiah *et al.*, "Analisis Sentimen Access by Bus Kota se-Indonesia Menggunakan Metode K-Nearest Neighbors," vol. 3, 2025.
- [9] T. Gori, A. Sunyoto, and H. Al Fatta, "Preprocessing Data dan Klasifikasi untuk Prediksi Kinerja Akademik Siswa," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 1, pp. 215–224, 2024, doi: 10.25126/jtiik.20241118074.
- [10] A. Yoga Pratama *et al.*, "Analisis Sentimen Media Sosial Twitter Dengan Algoritma K-Nearest Neighbor Dan Seleksi Fitur Chi-Square (Kasus Omnibus Law Cipta Kerja)," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 5, no. 2, pp. 897–910, 2021.
- [11] N. Fibriyanti Arminda, N. Sulistiyowati, and T. Nur Padilah, "Implementasi Algoritma Multinomial Naive Bayes Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Brimo," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 3, pp. 1817–1822, 2023, doi: 10.36040/jati.v7i3.7012.
- [12] N. L. P. C. Savitri, R. A. Rahman, R. Venyutzky, and N. A. Rakhmawati, "Analisis Klasifikasi Sentimen Terhadap Sekolah Daring pada Twitter Menggunakan Supervised Machine Learning," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 47–58, 2021, doi: 10.28932/jutisi.v7i1.3216.
- [13] D. Meisa Azzahra, M. Hafid Totohendarto, and S. Alam, "Analisis Sentimen Ulasan Produk Serum Wajah Pada Beauty Brand Somethinc Menggunakan Metode Naive Bayes Classifier," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 3, pp. 1604–1611, 2023, doi: 10.36040/jati.v7i3.6929.
- [14] N. Cahyono and Anggista Oktavia Praneswara, "Analisis Sentimen Ulasan Aplikasi TikTok Shop Seller Center di Google Playstore Menggunakan Algoritma Naive Bayes," *Indones. J. Comput. Sci.*, vol. 12, no. 6, pp. 3925–3940, 2023, doi: 10.33022/ijcs.v12i6.3473.
- [15] H. Firda *et al.*, "Perbandingan Pelabelan Rating - based dan Inset Lexicon - based dalam Analisis Sentimen Menggunakan SVM (Studi Kasus : Ulasan Aplikasi GoBiz di Google Play Store) Comparison of Rating - based and Inset Lexicon - based Labeling in Sentiment Analysis usin," vol. 14, pp. 516–528, 2025.
- [16] H. C. Husada and A. S. Paramita, "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Teknika*, vol. 10, no. 1, pp. 18–26, 2021, doi: 10.34148/teknika.v10i1.311.
- [17] A. A. Muttaqin, S. Alam, M. A. Komara, T. Informatika, and A. Hermawan, "ANALISIS SENTIMEN ISU KECURANGAN PEMILU 2024 BERDASARKAN OPINI PADA MEDIA SOSIAL TWITTER MENGGUNAKAN METODE CRISP-DM DENGAN ALGORITMA NAIVE BAYES CLASSIFIER," vol. 8, no. 5, pp. 8764–8772, 2024.

- [18] J. A. Wibowo, V. C. Mawardi, and T. Sutrisno, "Visualisasi Word Cloud Hasil Analisis Sentimen Berbasis Fitur Layanan Aplikasi Gojek Dengan Support Vector Machine," *J. Serina Sains, Tek. dan Kedokt.*, vol. 2, no. 1, pp. 61–70, 2024, doi: 10.24912/jsstk.v2i1.32058.
- [19] Kurnia, I. Purnamasari, and D. D. Saputra, "Analisis Sentimen Dengan Metode Naïve Bayes, SMOTE Dan Adaboost Pada Twitter Bank BTN," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 7, no. 2, pp. 235–242, 2023, doi: 10.35870/jtik.v7i3.707.