IMPLEMENTASI METODE *ENSEMBLE MAJORITY VOTE* PADA ALGORITMA *NAIVE BAYES* DAN *RANDOM FOREST* UNTUK ANALISIS SENTIMEN TWITTER HARGA TIKET PESAWAT DOMESTIK

Dimas Triyana, Muhammad Muharrom Al Haromainy, Hendra Maulana
Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur
Jalan Raya Rungkut Madya No. 1, Gunung Anyar, Surabaya, Indonesia
20081010049@student.upnjatim.ac.id

ABSTRAK

Transportasi merupakan komponen penting dalam mendukung pertumbuhan ekonomi suatu negara. Industri penerbangan domestik di Indonesia, khususnya, mengalami pertumbuhan pesat setelah terdampak pandemi Covid-19. Tetapi kenaikan jumlah penumpang mengakibatkan perbedaan pendapat terkait harga tiket pesawat domestik yang semakin mahal, yang menjadi faktor krusial dalam keputusan pembelian konsumen. Analisis sentimen melalui media sosial seperti Twitter menjadi solusi untuk memahami opini masyarakat terhadap harga tiket pesawat domestik. Fokus Penelitian ini menggunakan algoritma *Naive Bayes* dan *Random Forest* kemudian diimplementasikan pada *Ensemble Majority Vote*. Pada analisis sentimen pada tweet terkait harga tiket penerbangan domestik. Skenario pertama *Naive Bayes* menggunakan data *training* dan *testing* dengan rasio 80:20, menghasilkan akurasi sebesar 70,31% setelah menerapkan *SMOTE* untuk menangani ketidak seimbangan pada *labelling* data. Skenario kedua, dengan rasio 70:30, mendapat akurasi 71,75%. Metode *Random Forest* juga digunakan sebagai algoritma perbandingan. Pada skenario pertama, *Random Forest* mencapai akurasi 84,91%, dan pada skenario kedua, 85,71%. Penggabungan kedua algoritma menggunakan *ensemble majority voting* menghasilkan akurasi 85,88% dan 85,87% untuk kedua skenario, menunjukkan peningkatan akurasi prediksi dibandingkan model individu algoritma. Dalam artian metode *ensemble majority vote* dinilai efektive dalam meningkatkan akurasi analisis sentimen terhadap harga tiket pesawat domestik. Arah opini masyarakat terkait harga tiket pesawat domestik dalam media sosial Twitter adalah negatif.

Kata kunci : Analisis Sentimen, *Naive Bayes*, *Random Forest*, *Ensemble Majority Vote*

1. PENDAHULUAN

Transportasi berperan krusial dalam mendukung kemajuan ekonomi masyarakat dan merupakan pilar utama dalam upaya pembangunan ekonomi suatu negara. Salah satu sektor yang terus berkembang pesat adalah industri penerbangan domestik. Industri penerbangan di Tanah Air memasuki fase pemulihan setelah terdampak Covid-19. Diprediksi bahwa Bisnis penerbangan akan secara bertahap pulih dan mengalami peningkatan permintaan yang signifikan [1].

Indonesia merupakan pasar terbesar dengan pertumbuhan tertinggi dalam industri penerbangan di ASEAN. Pada tahun 2030, pasarnya diperkirakan mencapai 390 juta penumpang dari dan di dalam Indonesia [2].

Meskipun transportasi pesawat menawarkan banyak keunggulan, harga tiket pesawat domestik menjadi faktor penting dalam keputusan pembelian konsumen. Dengan meningkatnya jumlah penumpang, opini tentang harga tiket menjadi bervariasi, dengan persepsi dan respon masyarakat yang tidak selalu mudah dipahami. Saat ini topik yang sedang ramai dibicarakan di media sosial adalah "Harga Tiket Pesawat Domestik". Analisis sentimen adalah metode yang digunakan untuk mengidentifikasi teks yang mengandung opini, pandangan, pendapat, penilaian, dan emosi seseorang terhadap suatu objek [3].

Tujuannya adalah untuk memberikan informasi kepada pemerintah dan pemangku kepentingan agar dapat mengambil keputusan yang lebih bijaksana dengan memahami opini positif dan negatif. Media sosial twitter digunakan karena memfasilitasi penyampaian opini publik di ruang digital. *Tweet* yang tersedia secara publik menunjukkan mereka dapat menunjukkan reaksi dan keterlibatan terhadap sebuah *tweet* melalui *retweet*, *like*, *mention*, atau balasan terhadap *tweet* pengguna lain [4].

Ada banyak algoritma yang dapat mendukung proses klasifikasi dalam analisis sentimen, di antaranya adalah *Naïve Bayes Classifier*, *Support Vector Machine*, *Decision Tree*, *Random Forest*, *K-Nearest Neighbor*, *Rule-based Classification*, *K-Means*, dan masih banyak lagi.

Penelitian ini berfokus pada algoritma *Naïve Bayes* dan *Random Forest*. *Naïve Bayes* dipilih karena akurasi tinggi pada data besar, sementara *Random Forest* menggunakan metode *ensemble* untuk menghasilkan kesimpulan kuat. Metode TF-IDF digunakan untuk pembobotan kata. Hasil kedua algoritma akan dibandingkan untuk menilai akurasi masing-masing dan efektivitas metode *ensemble majority vote* dalam meningkatkan akurasi analisis sentimen.

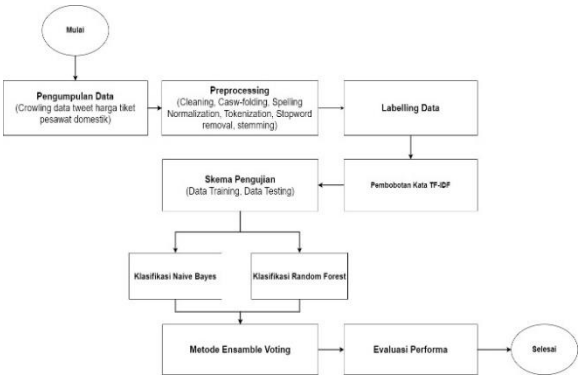
2. TINJAUAN PUSTAKA

Pada 2019, [5] melakukan penelitian analisis sentimen terhadap komentar aplikasi Tokopedia menggunakan *Naïve Bayes*. Mereka menggunakan dataset dari komentar pengguna di *Google Playstore*, mengklasifikasikannya dengan algoritma tersebut, dan mengukur akurasinya di *Rapidminer*. Pengumpulan data dilakukan dalam dua tahap: tahap pertama adalah mengumpulkan data yang akan digunakan sebagai data *training*, dan tahap kedua adalah mengumpulkan data yang akan digunakan sebagai data *testing*. Hasilnya menunjukkan akurasi yang baik, mencapai 97,13%, dengan nilai *precision* sebesar 1 dan *recall* sebesar 95,49%. AUC juga mencapai 0,980

Pada tahun 2020, [6] melakukan penelitian tentang analisis sentimen terhadap aplikasi Ruangguru. Menggunakan algoritma *Naive Bayes*, *Random Forest*, dan *Support Vector Machine* dalam penelitiannya. Mereka mengumpulkan dataset dari *Playstore*, melakukan *preprocessing*, dan mengaplikasikan model klasifikasi untuk menentukan sentimen positif atau negatif serta mengukur akurasi dan *AUC*. Dengan menerapkan metode *SMOTE* karena ketidakseimbangan kelas, *Random Forest* menghasilkan model terbaik dengan akurasi 97,16% dan *AUC* 0,996.

Selanjutnya pada 2022, perbandingan menggunakan *Random Forest* juga pernaah dilakukan [7]. Melalui pemilihan atribut terbaik berdasarkan fitur keuntungan informasi, data diproses dan dibagi menjadi data pelatihan dan pengujian. Hasil menunjukkan bahwa empat atribut terbaik memberikan klasifikasi yang signifikan. Algoritma *Random Forest* mencapai akurasi 100%, lebih unggul daripada *K-Nearest Neighbor* (K-NN) yang hanya mencapai 86,67%. Analisis menunjukkan bahwa *Random Forest* lebih efektif dalam mengklasifikasikan durasi studi mahasiswa.

3. METODE PENELITIAN



Gambar 1. Alur penelitian

Pada penelitian yang dilakukan, terdapat beberapa tahapan seperti yang terlihat pada Gambar 1. Tahap pertama adalah pengumpulan data, kemudian dilanjutkan dengan tahap *preprocessing* data, *labeling* data, ekstraksi fitur menggunakan TF-IDF, pembagian data menjadi data *training* dan *testing*, serta tahap klasifikasi menggunakan algoritma *Naïve Bayes* dan *Random Forest*, tahap penggabungan agar algoritma klasifikasi tersebut menjadi optimal dengan implementasi metode *Ensemble Majority Voting* dan diakhiri dengan tahap pengujian data.

3.1. Pengumpulan Data

Pada tahap awal, data dikumpulkan melalui *crawling* di Twitter untuk menganalisis *tren*, pola, dan pendapat dalam percakapan online. *Python* digunakan untuk proses *crawling* menggunakan alat *Tweet-harvest* yang mengakses data Twitter melalui API. Kata kunci seperti "penerbangan domestik," "maskapai domestik," "tiket penerbangan domestik," dan "tiket domestik mahal" digunakan untuk pencarian. Tweet yang dikumpulkan mencakup periode dari 1 Januari 2022 hingga Mei 2024, dengan fokus pada pemulihan ekonomi Indonesia pasca pandemi COVID-19, terutama isu harga tiket penerbangan domestik. Data yang terkumpul akan diproses melalui tahap *preprocessing* untuk analisis lebih lanjut.

3.2. Preprocessing

Preprocessing adalah langkah penting dalam analisis data yang bertujuan untuk membersihkan dan menyiapkan data mentah agar siap untuk analisis lebih lanjut. Ini melibatkan serangkaian langkah seperti: *Case folding*, cleaning, *tokenization*, *stopword removal*, *spelling normalization*, *stemming*.Tujuannya adalah menghasilkan data yang terstruktur dan konsisten untuk analisis sentimen atau topik tertentu. *Preprocessing* yang efektif dapat meningkatkan kualitas analisis dan akurasi hasil.

3.3. Labeling Data

Labelling dilakukan dengan mengkategorikan ulasan ke dalam dua klasifikasi utama, yaitu negatif dan positif, berdasarkan karakteristik atau atribut yang relevan. Proses pelabelan sentimen melibatkan definisi fungsi *polarity()* dan *subjectivity()*. *Polarity* memiliki rentang nilai dari -1 hingga 1; nilai kurang dari 0 mengindikasikan sentimen negatif, sedangkan nilai lebih dari 0 mengindikasikan sentimen positif. *Subjectivity* memiliki rentang nilai antara 0 hingga 1.

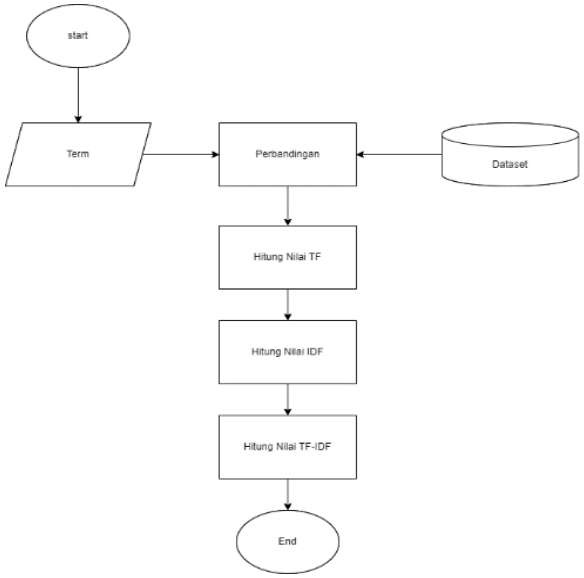
3.4. Pembobotan Kata

Pembobotan kata yang dilakukan mnggunakan metode TF-IDF, metode ini digunakan untuk menghitung bobot relatif dari setiap kata dalam setiap dokumen berdasarkan seberapa sering kata tersebut muncul dalam dokumen tersebut dengan metode *Term Frequency* (TF) dan seberapa umumnya kata tersebut di seluruh koleksi dokumen menggunakan *Inverse Document Frequency* (IDF).

Setelah melewati proses *preprocessing*, hasil sampel akan digunakan sebagai basis kamus kata-kata yang akan digunakan untuk menghitung bobot nilai

[8]. Dataset yang telah mengalami proses pelabelan pada *preprocessing* harus diubah menjadi bentuk angka [9].

Pada dasarnya pada gambar 2 alur *tf-idf* langkah pertama yang dilakukan adalah dilakukan perbandingan bahwa kata yang lebih sering muncul dalam satu dokumen dan lebih jarang di dokumen lain harus diberikan prioritas lebih tinggi karena mereka lebih berguna untuk klasifikasi.



Gambar 2. Alur *tf-idf*

Algoritma ini mengkomputasi nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) dari sebuah kata dalam dataset. Nilai TF kemudian dihitung untuk mengukur frekuensi kemunculan suatu kata dalam sebuah dataset atau dokumen. Ekspresi matematis dari *Term Frequency* (TF) terdapat pada Persamaan (1).

$$TF = \frac{Frekuensi\ term\ dalam\ satu\ dokumen}{Total\ kata\ dalam\ satu\ dokumen} \quad (1)$$

Setelah mendapat nilai TF, Kemudian hitung nilai IDF untuk menilai seberapa jarang kata tersebut muncul dalam seluruh dataset. Ekspresi matematis dari *Inverse Document Frequency* (IDF) terdapat pada Persamaan (2).

$$IDF=log\ \frac{Total\ Dokumen}{Frekuensi\ dokumen\ mengandung\ term} \quad (2)$$

Setelah mendapatkan nilai TF dan IDF Adapun perhitungan *Term Frequency-Inverse Document Frequency* (TF-IDF) sebagai berikut:

$TF{-}IDF(term,doc)=TF(term,doc)*IDF(term)$ (3)

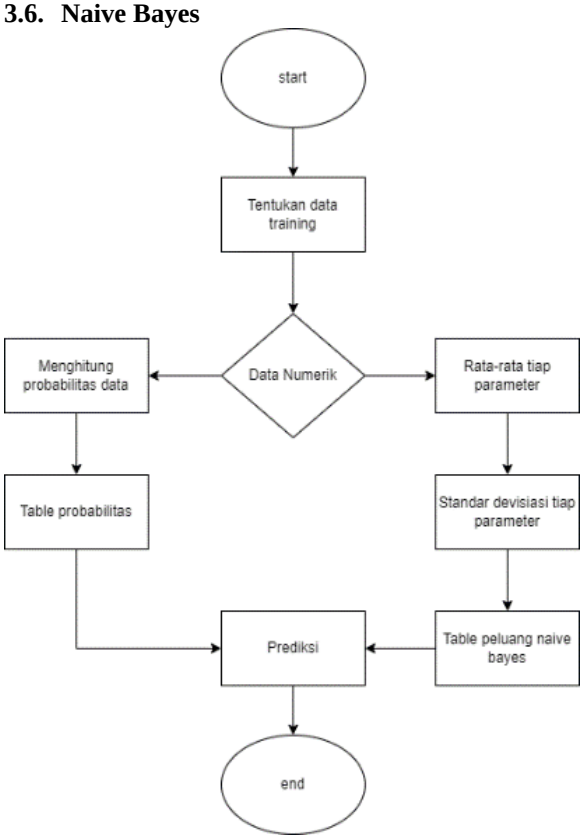
- *term* adalah kata yang dihitung nilai TF-IDF-nya.
- *doc* adalah dokumen yang dihitung nilai TF-IDF-nya.
- TF(*term*, *doc*) adalah nilai *Term Frequency* dari kata tersebut di dalam dokumen.
- IDF(*term*) adalah nilai *Inverse Document Frequency* dari kata tersebut dalam seluruh kumpulan dokumen.

Dengan mengkombinasikan nilai TF dan IDF, dapat dihitung nilai TF-IDF dari kata tersebut dalam dokumen [8]. Bobot suatu istilah semakin besar ketika istilah tersebut sering muncul dalam suatu dokumen, dan menurun ketika istilah tersebut muncul dalam banyak dokumen [10].

3.5. Skema Pengujian

Sebelum melakukan klasifikasi dan pengujian, penelitian ini membagi data menjadi dua bagian: data *train* dan data *test*. Pada skema pertama, data dibagi menjadi 80% data *train* dan 20% data *test*, sedangkan pada skema kedua, pembagian menjadi 70% data *train* dan 30% data *test*. Tujuannya adalah untuk menentukan perbandingan optimal antara data train dan data test yang dapat menghasilkan tingkat akurasi terbaik.

3.6. Naive Bayes



Gambar 3. Alur naive bayes

Algoritma yang digunakan *Naïve Bayes* yaitu *Multinominal Naïve Bayes*. Pada gambar 3 diagram alir *naive bayes* algoritma *Naive Bayes* menunjukkan bahwa untuk proses prediksi, *Naive Bayes* menggunakan dua jenis data: dataset sebagai data pelatihan dan data tes sebagai data yang akan diprediksi. Dataset digunakan sebagai data pelatihan untuk menentukan pola atau model prediksi. Sementara data uji digunakan sebagai data untuk

prediksi menggunakan model yang telah dilatih dari dataset pelatihan tersebut.

Metode ini menganggap bahwa pengaruh nilai atribut terhadap suatu kelas tidak bergantung pada nilai atribut lainnya. Asumsi ini dikenal sebagai independensi bersyarat terhadap kelas, yang membuat metode klasifikasi ini dianggap sederhana. Teorema dasar dari algoritma *Naive Bayes* dapat dirumuskan sebagai berikut [11].

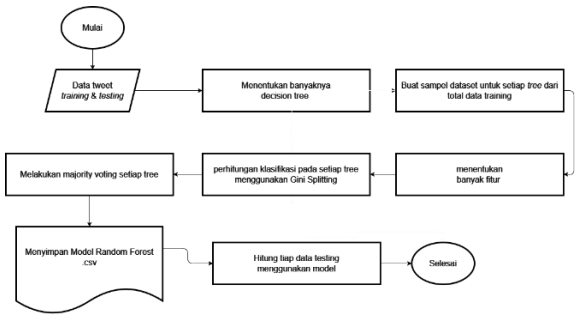
$$P(B|A) = \frac{P(B)P(A|B)}{P(A)} \quad (3)$$

Keterangan:

- *A*: Data dengan class yang belum diketahui
- B: Hipotesis data *A* yang merupakan suatu class spesifik
- P(B|A): Merupakan probabilitas dari dokumen(teks) yang dimiliki oleh kelas B
- P(B): Probabilitas hipotesis B (prior probabilitas)
- P(A|B): Probabilitas A berdasarkan kondisi pada hipotesis B
- P(A): Probabolitas A

3.7. Random Forest

Random forest adalah teknik penggabungan (*ensemble*) yang terdiri dari banyak pohon keputusan. Setiap pohon keputusan dibangun dengan menggunakan subset acak dari data, dan prediksi dari setiap pohon dikombinasikan untuk meningkatkan akurasi prediksi. Ketika proses prediksi selesai dilakukan oleh setiap pohon, hasilnya digabungkan dengan cara mengambil mayoritas suara dari semua pohon yang terlibat. Pendekatan ini memungkinkan *Random Forest* untuk mengatasi berbagai jenis masalah prediksi dengan memanfaatkan kekuatan dari kombinasi pohon keputusan yang berbeda-beda.

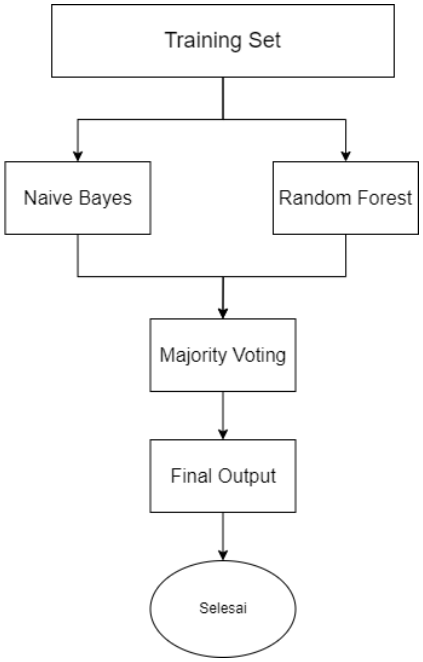


Gambar 4. Alur *random forest*

Dalam model klasifikasi *Random Forest*, langkah pertama adalah menentukan jumlah *tree* untuk setiap model. Kemudian, dataset di-*sampling* untuk setiap *tree* dengan metode *bootstrapping* atau diacak. *Tree-tree* dibuat tanpa pemangkasan dan dengan fitur-fitur yang dipilih secara acak. Setiap *tree* menghitung pemisahan menggunakan nilai gini-split terkecil pada data *training*. Proses ini diulang hingga *node-node internal homogen* terbentuk atau tidak ada atribut lagi yang bisa dipisahkan. Hasil prediksi dari setiap *tree* digabungkan melalui *majority voting* untuk menghasilkan kelas prediksi akhir. Tahap selanjutnya adalah menguji model dengan data testing untuk evaluasi klasifikasi *Random Forest*.

3.8. Ensamble Majority Voting

Ensemble Voting adalah teknik dalam *machine learning* di mana beberapa model atau algoritma yang berbeda digabungkan untuk membuat prediksi atau keputusan akhir. Ada dua kategori utama metode ensemble berbasis *majority*, yaitu *voting* dan *averaging*. Umumnya, *voting* digunakan untuk klasifikasi, sedangkan *averaging* digunakan untuk *regresi*.



Gambar 5. Alur *majority voting*

Metode kelasifikasi yang digunakan ialah implementasi metode *ensemble majority vote* yang menggabungkan algoritma *Naive Bayes* dipilih karena kemampuannya dalam mengelola data dalam skala besar dengan tingkat akurasi yang tinggi, serta *Random Forest* karena kapabilitasnya dalam menangani berbagai fitur secara efektif, mencegah *overfitting*, toleran terhadap data tidak seimbang dan *missing values*, serta memberikan perkiraan fitur yang penting dalam klasifikasi. Dua hasil algoritma tersebut akan dikombinasikan dengan metode *ensemble majority vote* dengan tujuan untuk optimalisasi dan mendapatkan hasil prediksi akhir yang lebih akurat dibanding dengan hasil klasifikasi individu.

Majority voting dilakukan dengan memilih prediksi yang paling sering muncul di antara prediksi model untuk setiap *tweet*. Jika terdapat seri (*tie*), maka dipilih prediksi dari model yang lebih akurat. Berikut digunakan perhitungan majority voting untuk setiap tweet T_i sebagai berikut:

$C(X) = mode\{h_1(X), h_2(X)\}$ (4)

Jika terdapat *tie*, disini memilih h_2 karena dalam kasus ini, bahwa model *random forest* menunjukkan akurasi lebih baik. jadi, jika terjadi *tie*, label dari prediksi dari *Random Forest* yang dipilih. Berikut perhitungannya:

$C(X) = h_2(X)$ (5)

Menentukaan hasil *majority voting* untuk setiap dokumen tweet dengan mempertimbangkan prediksi yang telah dihasilkan oleh model-model yang terlibat.

3.9. Confusion Matrix

Tahap terakhir merupakan tahap analisis evaluasi performa dari algoritma klasifikasi yang telah dilakukan. Dalam penelitian ini, alat evaluasi yang digunakan adalah confusion matrix. Untuk mengukur kinerja algoritma, dilakukan evaluasi dengan menggunakan accuracy, precision, recall, dan F1 score. Dengan menggunakan confusion matrix, nilai-nilai tersebut dapat dihitung untuk setiap skenario pengujian.

Accuracy adalah ukuran evaluasi yang menunjukkan seberapa tepat model memprediksi label data uji. Secara matematis, akurasi dijabarkan pada persamaan 6.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Precission adalah ukuran proporsi jumlah positif yang diprediksi dengan benar dibandingkan total positif yang diprediksi oleh model. Secara matematis, akurasi dijabarkan pada persamaan 7.

$$Precission = \frac{TP}{TP + FP} \quad (7)$$

Recall adalah ukuran persentase positif yang diprediksi dengan benar dari total sampel yang sebenarnya positif dalam dataset. Secara matematis, akurasi dijabarkan pada persamaan 8.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

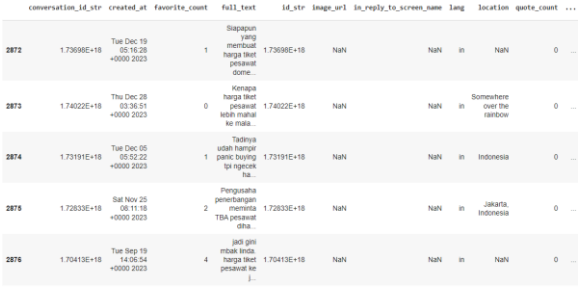
F1 score adalah pengukuran evaluasi yang menggabungkan presisi dan *recall* menjadi satu nilai tunggal untuk mengevaluasi performa model klasifikasi. Secara matematis, akurasi dijabarkan pada persamaan 9.

$$F1\ score = 2\ \frac{Recall \times Precision}{precision + recal} \quad (9)$$

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Proses awal dalam pengumpulan data dimulai dengan mencari kata kunci menggunakan Twitter API. Kata kunci yang digunakan adalah "penerbangan domestik", "pesawat domestik", "tiket pesawat domestik", dan "tiket domestik mahal". Data tweet diambil dari 1 Januari 2022 hingga Mei 2024, untuk melihat perkembangan isu harga tiket pesawat domestik Kondisi ini menjadi fokus utama penelitian untuk mengetahui sentimen masyarakat terhadap isu tersebut.



	conversation_id_str	created_at	favorite_count	full_text	id_str	image_url	in_reply_to_screen_name	lang	location	quote_count	...
2872	1.73698E+18	Tue Dec 19 05:16:28 +0000 2023	1	Siapapun yang membuat harga tiket pesawat dome...	1.73698E+18	NaN	NaN	in	NaN	0	...
2873	1.74022E+18	Thu Dec 28 03:36:51 +0000 2023	0	Kenapa harga tiket pesawat lebih mahal ke mala...	1.74022E+18	NaN	NaN	in	Somewhere over the rainbow	0	...
2874	1.73191E+18	Tue Dec 05 05:52:22 +0000 2023	1	Tadinya udah hampir panic buying tpi ngecek ha...	1.73191E+18	NaN	NaN	in	Indonesia	0	...
2875	1.72833E+18	Sat Nov 25 08:11:18 +0000 2023	2	Pengusaha penerbangan meminta TBA pesawat dihа...	1.72833E+18	NaN	NaN	in	Jakarta, Indonesia	0	...
2876	1.70413E+18	Tue Sep 19 14:06:54 +0000 2023	4	jadi gini mbak linda. harga tiket pesawat ke j...	1.70413E+18	NaN	NaN	in	NaN	0	...

Gambar 6. Hasil data crawling

Setelah proses *crawling*, berhasil terkumpul sebanyak 2877 tweet. Selanjutnya, dilakukan penghapusan duplikasi sehingga tersisa 2053 data sentimen yang akan menjadi fokus dalam penelitian ini.

4.2. Preprocessing Data

Preprocessing meliputi mengonversi huruf menjadi huruf kecil, memisahkan kata-kata, menghapus kata-kata tidak relevan, dan menormalisasi format kata-kata. Langkah ini mempersiapkan data untuk analisis lebih lanjut dengan lebih efisien.

Tabel 1. Hasil Data *Preprocessing*

full_text	full_text_tokens	tweet_normalized	tweet_tokens_stemmed	tweet_stemmed_sentence
kenapa harga tiket pesawat ke siborongborong medan itu mahal sekali ya sampai sejuta lebih melebihi tiket pergi ke bal	['kenapa', 'harga', 'tiket', 'pesawat', 'ke', 'siborongborong', 'medan', 'itu', 'mahal', 'sekali', 'ya', 'sampai', 'sejuta', 'lebih', 'melebihi', 'tiket', 'pergi', 'ke', 'bali']	['harga', 'tiket', 'pesawat', 'siborongborong', 'medan', 'mahal', 'sejuta', 'melebihi', 'tiket', 'pergi', 'bali']"	['harga', 'tiket', 'pesawat', 'siborongborong', 'medan', 'mahal', 'juta', 'lebih', 'tiket', 'pergi', 'bali']	harga tiket pesawat siborongborong medan mahal juta lebih tiket pergi bali
kenapa harga tiket pesawat lebih mahal ke malangs ih dibanding ke malaysia lipet lohhhh heuhhh	[kenapa', 'harga', 'tiket', 'pesawat', 'lebih', 'mahal', 'ke', 'malangs', 'ih', 'dibanding', 'ke', 'malaysia', 'lipet', 'lohhhh', 'heuhhh']	['harga', 'tiket', 'pesawat', 'mahal', 'malang', 'dibanding', 'malaysia', 'lipet']	['harga', 'tiket', 'pesawat', 'mahal', 'malangs', 'banding', 'malaysia', 'lipet', 'lohhhh', 'heuhhh']	harga tiket pesawat mahal malang banding malaysia

4.3. Case Folding

Pada tahap ini, semua huruf dalam teks tweet diubah menjadi *lowercase* untuk menghindari ambiguitas antara kata yang sama namun ditulis dalam huruf besar atau kecil.

```
Case Folding Result :

0       meledak lantai per lantai berurutan. tidak ada...
1       terungkap sudah status vaksin aku so... selama...
2       terpantau proses imigrasi jepang lumayan cepat...
3       efek pandemi harga f-35 lightning ii naik? dal...
4       new zealand : pandemi covid usai amerika : bid...
                          ...
2872    siapapun yang membuat harga tiket pesawat dome...
2873    kenapa harga tiket pesawat lebih mahal ke mala...
2874    tadinya udah hampir panic buying tpi ngecek ha...
2875    pengusaha penerbangan meminta tba pesawat diha...
2876    jadi gini mbak linda. harga tiket pesawat ke j...
```

Gambar 7. Hasil data case folding

4.4. Cleaning

Pada proses ini, data dibersihkan dari penggunaan URL, karakter khusus seperti *tab*, *newline*, backslash, serta karakter *non-ASCII* dari teks seperti contoh *tweet* berikut: "kenapa harga tiket pesawat ke siborong-borong medan itu mahal sekali ya @bobbynasution_ sampai sejuta lebih melebihi tiket pergi ke bali???" *Tweet* tersebut akan dibersihkan dari karakter khusus dan diterapkan pada kolom *full_text* dalam tabel 1.

4.5. Tokenizing

Proses tokenisasi adalah langkah untuk memisahkan teks menjadi token-token individu, yang merupakan unit analisis. Tokenisasi memecah teks menjadi unit-unit kecil berdasarkan spasi, memfasilitasi pemrosesan yang lebih mendetail dan identifikasi entitas atau elemen penting dalam teks.

4.6. Stopword Removal

Pada tahap ini, "stopwords" yang umumnya terdiri dari kata-kata seperti "dan", "atau", "di", "dari", dan sebagainya, dihapus dari teks setelah tokenization. Langkah ini penting karena kata-kata tersebut sering kali tidak memberikan kontribusi signifikan terhadap makna atau inti teks.

4.7. Normalisasi

Normalisasi sangat penting karena data dari Twitter cenderung tidak terstruktur dan sering kali mengandung variasi karena kesalahan penulisan, singkatan, atau bahasa informal. Penulis menggunakan file eksternal untuk memperkaya dan memastikan variasi kata atau frasa, sehingga tidak hanya terbatas pada model internal.

a. Stemming

Tahap terakhir dalam preprocessing data adalah stemming, di mana kata-kata dalam dataset disederhanakan menjadi bentuk dasarnya dengan menghapus awalan dan akhiran menggunakan StemmerFactory dari modul sastrawi. Pada tabel 1, tweet_tokens_stemmed mengubah teks tweet dari proses normalisasi ke bentuk dasar atau akarnya, memfasilitasi analisis lebih lanjut.

4.8. Labeling Data

Setelah preprocessing, dilakukan tahap labeling data. Langkah pertama adalah menerjemahkan tweet ke dalam bahasa Inggris menggunakan modul googletrans==4.0.0-rc1, untuk memperluas cakupan analisis. Labeling kemudian mengkategorikan opini menjadi dua klasifikasi: negatif dan positif, menggunakan modul textblob untuk analisis sentimen. Label positif diubah menjadi nilai 1 dan negatif menjadi nilai 0 untuk menyederhanakan representasi kelas sentimen, memudahkan proses pembobotan kata atau pemodelan klasifikasi, dan mendukung perhitungan statistik yang efisien.

	tweet_tokens_stemmed	tweet_stemmed_sentence	tweet_english	label
0	['ledak', 'lantai', 'lantai', 'ber urut', 'pes...	ledak lantai lantai ber urut pesawat tabrak ge...	explosive floor in sorting floor plane crash c...	0
1	['ungkap', 'status', 'vaksin', 'kenapa', 'joke...	ungkap status vaksin kenapa joker kemanamana p...	Reveal the vaccine status why Joker Kemanamana...	0
2	['pantau', 'proses', 'imigrasi', 'jepang', 'lu...	pantau proses imigrasi jepang lumayan cepat ce...	Monitor the Japanese immigration process is qu...	0
3	['efek', 'pandemi', 'harga', 'lightning', 'waw...	efek pandemi harga lightning wawancara defense...	Pandemic Effects Lightning Prices Interview De...	0
4	['new', 'zealand', 'pandemi', 'covid', 'amerik...	new zealand pandemi covid amerika biden umumin...	New Zealand Pandemi Covid American Biden Publi...	1

Gambar 8. Hasil *labeling* data

4.9. Pembobotan Kata

Ekstraksi fitur bertujuan untuk memberikan bobot kata-kata dalam dataset untuk menilai tingkat signifikansinya. Dalam tahap ini, ekstraksi fitur menggunakan metode TF-IDF yang membantu dalam mengevaluasi bobot dari term terkait sentimen yang terkandung di dalamnya.

a. Term Frequency (TF)

Proses ini menentukan frekuensi kemunculan sebuah kata dalam sebuah dokumen.

Pada gambar 9 pengukuran ini membantu dalam menentukan seberapa penting atau relevan suatu kata terhadap dokumen atau dataset yang sedang dianalisis. Semakin frekuensi kemunculan sebuah kata meningkat dalam sebuah teks, semakin tinggi juga nilai Term Frequency-nya.

term	TF
alas	0.06666666666666667
harga	0.13333333333333333
avtur	0.13333333333333333
mahal	0.06666666666666667
murah	0.13333333333333333
dex	0.06666666666666667
dexlite	0.06666666666666667
tiket	0.06666666666666667
pesawat	0.06666666666666667
internasional	0.06666666666666667
domestik	0.06666666666666667
kartel	0.06666666666666667

Gambar 9. Hasil *term frequency*

b. *Inverse Document Frequency* (IDF)
Inverse Document Frequency (IDF) mengukur seberapa umumnya sebuah kata muncul di seluruh koleksi dokumen dengan membandingkan jumlah total dokumen dengan jumlah dokumen yang mengandung kata tersebut. Kata-kata yang sering muncul di banyak dokumen dianggap kurang memberikan informasi unik tentang dokumen tertentu.

c. TF-IDF
Pada pembobotan kata dengan TF-IDF, kata-kata yang muncul lebih sering dalam satu dokumen dan lebih jarang di dokumen lain diberikan bobot yang lebih tinggi karena dianggap lebih informatif dan bermanfaat untuk proses klasifikasi.

term	TF	IDF	TF-IDF
alas	0.06666666666666667	4.491563201089784	0.29943754673931894
harga	0.13333333333333333	1.0976385787567082	0.14635181050089444
avtur	0.13333333333333333	4.919007215916723	0.6558676287888965
mahal	0.06666666666666667	1.1182882800472518	0.0745525520031501
murah	0.13333333333333333	1.8401600356522259	0.2453546714202967
dex	0.06666666666666667	6.933910236458988	0.4622606824305992
dexlite	0.06666666666666667	6.933910236458988	0.4622606824305992
tiket	0.06666666666666667	0.2227782989816655	0.014851886598777704
pesawat	0.06666666666666667	0.27141631404468	0.0180944209363119
internasional	0.06666666666666667	2.751860093817782	0.18345733958785215
domestik	0.06666666666666667	0.7622096390480732	0.05081397593653821
kartel	0.06666666666666667	4.736685659122769	0.31577904394151796

Gambar 10. Hasil *tf-idf*

4.10. Skema Pengujian

Persentase pembagian data yang digunakan pada skema pengujian ini ialah pada skema pengujian pertama, pembagian data dilakukan dengan persentase 80% untuk data *training* dan 20% untuk data *testing*. Sedangkan untuk skema pengujian kedua, digunakan persentase 70% untuk data *training* dan 30% untuk data *testing*. Pembagian dataset ini dilakukan setelah proses *preprocessing* dan *labeling* data, yang menghasilkan total data pada dataset yang siap dilakukan analisis sebanyak 2053 data. Bisa dilihat pada table 2 berikut.

Table 2. Skema Pengujian

Skema	Data Training	Data Testing	Total
80:20	1642	411	2053
70:30	1437	616	2053

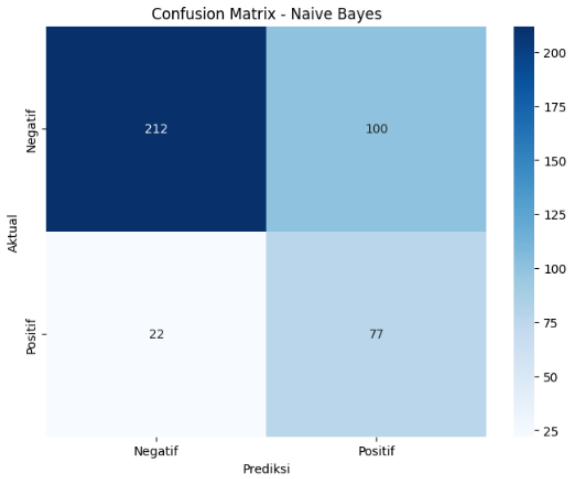
4.11. Naive Bayes

Pada tahap ini, peneliti memilih salah satu jenis algoritma Naive Bayes, yang terdiri dari beberapa varian utama seperti Multinomial Naive Bayes, Bernoulli Naive Bayes, dan Gaussian Naive Bayes. Peneliti memilih untuk menggunakan Multinomial Naive Bayes dalam analisisnya.

a. Skenario Pertama Klasifikasi Naive Bayes
Skenario pertama menggunakan data training dengan rasio 80:20, yaitu 1642 tweet yang telah dilabeli. Pada percobaan pertama, akurasi yang diperoleh adalah 70,31%. Prosesnya melibatkan perhitungan frekuensi kata per kelas dan probabilitas bersyarat dari setiap fitur terhadap label menggunakan Laplace Smoothing, serta perhitungan probabilitas prior berdasarkan jumlah sampel dalam setiap kelas. Prediksi dilakukan menggunakan model Naive Bayes. Penulis juga menggunakan *SMOTE* untuk menangani ketidakseimbangan kelas pada dataset, meningkatkan representasi kelas minoritas dalam data latihan sehingga model pembelajaran mesin dapat belajar dengan lebih baik tanpa terpengaruh dominasi kelas mayoritas.

	precision	recall	f1-score	support
0	0.91	0.68	0.78	312
1	0.44	0.78	0.56	99
accuracy			0.70	411
macro avg	0.67	0.73	0.67	411
weighted avg	0.79	0.70	0.72	411

Gambar 11. Hasil akurasi pertama klasifikasi naive bayes



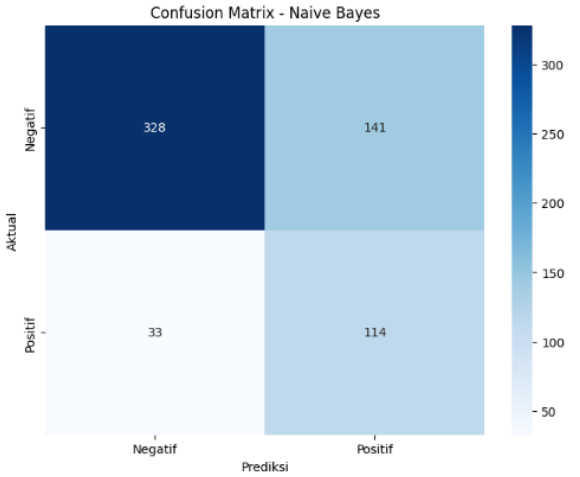
Gambar 12. Hasil *confusion matrix* skenario pertama *naive bayes*

Hasil dari confusion matrix untuk skenario 1, seperti yang terlihat pada Gambar 12, adalah sebagai berikut: 212 data negatif diprediksi benar sebagai negatif, 22 data positif salah diprediksi sebagai negatif, 100 data negatif salah diprediksi sebagai positif, dan 77 data positif diprediksi benar sebagai positif.

b. Skenario Kedua Klasifikasi Naive Bayes
Skenario kedua menggunakaan data training yaitu 70:30 adalah 1437. data tweet yang telah melalui tahap pelabelan data dalam proses pembangunan model. Pada percobaan kedua mendapatkan hasil akurasi 71,75%. Seperti yang terlihat di gambar 13.

	precision	recall	f1-score	support
0	0.91	0.70	0.79	469
1	0.45	0.78	0.57	147
accuracy			0.72	616
macro avg	0.68	0.74	0.68	616
weighted avg	0.80	0.72	0.74	616

Gambar 13 Hasil akurasi kedua klasifikasi *naive bayes*



Gambar 14 Hasil *confusion matrix* skenario kedua *naive bayes*

Berikut adalah hasil dari confusion matrix untuk skenario yang terdapat pada gambar 14: Terdapat 328 data aktual negatif yang diprediksi sebagai negatif, 33 data aktual positif yang salah diprediksi sebagai negatif, 141 data aktual negatif yang salah diprediksi sebagai positif, dan 114 data aktual positif yang diprediksi dengan benar sebagai positif.

4.12. Random Forest

Prosesnya dimulai dengan menentukan jumlah tree, kemudian membuat sampel dataset untuk setiap tree menggunakan bootstrapping. Setiap tree dibentuk tanpa pemangkasan dengan menentukan root node dan internal node berdasarkan kriteria gini-split terkecil. Setelah semua tree dibangun, hasil prediksi digabungkan untuk menentukan kelas prediksi akhir. Model ini kemudian diuji dengan data testing untuk mengevaluasi akurasinya.

a. Skenario Pertama Klasifikasi *Random Forest*

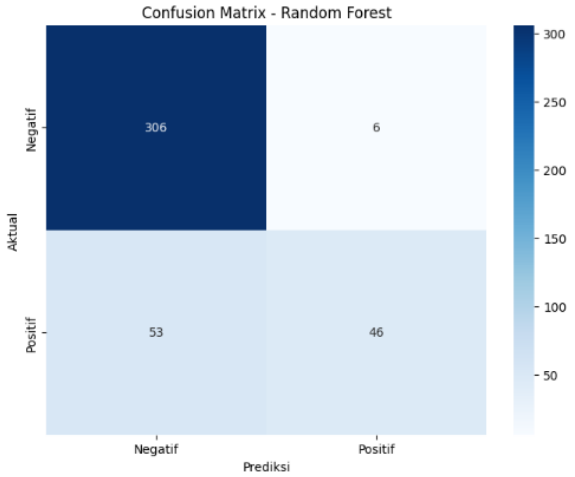
Skenario pertama menggunakaan data training yaitu 80:20 adalah 1642. data tweet yang telah melalui tahap pelabelan data dalam proses pembangunan model. Pada percobaan pertama mendapatkan hasil akurasi 84,91%. Gambar 15 menunjukan hasil akurasi pertama klasifikasi Random Forest.

```
Akurasi Random Forest: 0.8491484184914841
Classification Report Random Forest:
              precision    recall  f1-score   support

           0       0.84      0.99      0.91       312
           1       0.95      0.39      0.56        99

    accuracy                           0.85       411
   macro avg       0.89      0.69      0.73       411
weighted avg       0.87      0.85      0.82       411
```

Gambar 15. Hasil akurasi pertama klasifikasi *random forest*



Gambar 16. Hasil *confusion matrix* skenario pertama *random forest*

Berikut adalah hasil dari confusion matrix untuk skenario yang terlihat pada gambar 16: Terdapat 306 data aktual negatif yang diprediksi sebagai negatif, 53 data aktual positif yang salah diprediksi sebagai negatif, 6 data aktual negatif yang salah diprediksi sebagai positif, dan 46 data aktual positif yang diprediksi dengan benar sebagai positif.

b. Skenario kedua Klasifikasi *Random Forest*

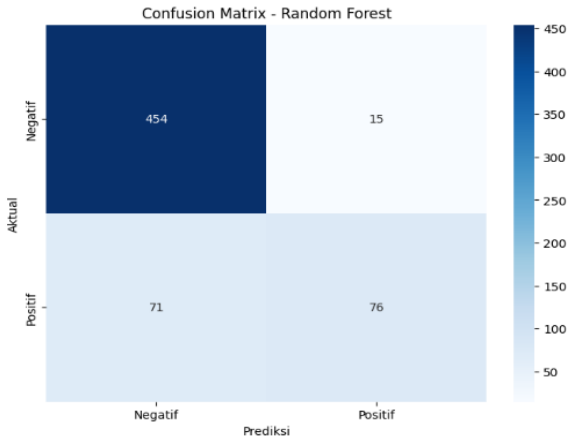
Skenario kedua menggunakaan data training yaitu 70:30 adalah 1437. data tweet yang telah melalui tahap pelabelan data dalam proses pembangunan model. Pada percobaan kedua mendapatkan hasil akurasi 85,71%. Bisa dilihat pada gambar 17.

```
Akurasi Random Forest: 0.8571428571428571
Classification Report Random Forest:
              precision    recall  f1-score   support

           0       0.86      0.97      0.91       469
           1       0.84      0.50      0.62       147

    accuracy                           0.86       616
   macro avg       0.85      0.73      0.77       616
weighted avg       0.86      0.86      0.84       616
```

Gambar 17. Hasil akurasi kedua klasifikasi *random forest*



Gambar 18. Hasil *confusion matrix* skenario kedua *random forest*

Berikut merupakan hasil dari confusion matrix skenario pada gambar 18: Terdapat 454 data aktual negatif yang diprediksi sebagai negatif, 71 data aktual positif yang salah diprediksi sebagai negatif, 15 data aktual negatif yang salah diprediksi sebagai positif, dan 76 data aktual positif yang diprediksi dengan benar sebagai positif.

4.13. *Ensamble Majority Voting*

Hasil algoritma *Naive bayes* dan *Random Forest* pada masing masing skenario tersebut akan dikombinasikan dengan metode *ensemble majority vote* dengan tujuan untuk optimalisasi dan mendapatkan hasil prediksi akhir yang lebih akurat dibanding dengan hasil klasifikasi individu.

a. Skenario Pertama *Ensamble Majority Voting*
Skenario pertama penggabungan dua algoritma tersebut menggunakaan data training yaitu 80:20 adalah 1642. data tweet yang telah melalui tahap pelabelan data dalam proses pembangunan model. Pada percobaan pertama mendapatkan hasil akurasi 85,88%. Bisa dilihat pada gambar 19.

```
Akurasi Ensemble Majority Vote: 0.8588807785888077

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.99      0.91       312
           1       0.92      0.45      0.61        99

    accuracy                           0.86       411
   macro avg       0.88      0.72      0.76       411
weighted avg       0.87      0.86      0.84       411
```

Gambar 19. Hasil akurasi skenario pertama *ensamble majority voting*

```
Confusion Matrix:
TP: 45, TN: 308, FP: 4, FN: 54
```

Gambar 20. Hasil *confusion matrix* skenario pertama *ensamble majority voting*

Berikut adalah hasil dari confusion matrix untuk skenario pertama yang terlihat pada gambar 20: Terdapat 308 data aktual negatif yang diprediksi dengan benar sebagai negatif (True Negative, TN), 54 data aktual positif yang salah diprediksi sebagai negatif (False Negative, FN), 4 data aktual negatif yang salah diprediksi sebagai positif (False Positive, FP), dan 45 data aktual positif yang diprediksi dengan benar sebagai positif (True Positive, TP).

b. Skenario Kedua *Ensamble Majority Voting*
Skenario kedua penggabungan dua algoritma tersebut menggunakaan data training yaitu 70:30 adalah 1437. data tweet yang telah melalui tahap pelabelan data dalam proses pembangunan model. Pada percobaan kedua mendapatkan hasil akurasi 85,87%. Bisa dilihat pada gambar 21.

```
Akurasi Ensemble Majority Vote: 0.8587662337662337

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.99      0.91       469
           1       0.93      0.44      0.60       147

    accuracy                           0.86       616
   macro avg       0.89      0.72      0.76       616
weighted avg       0.87      0.86      0.84       616
```

Gambar 21. Hasil *akurasi* skenario kedua *ensamble majority voting*

```
Confusion Matrix:
TP: 65, TN: 464, FP: 5, FN: 82
```

Gambar 22. Hasil *confusion matrix* skenario kedua *ensamble majority voting*

Berikut adalah hasil dari confusion matrix untuk skenario kedua yang terlihat pada gambar 22: Terdapat 464 data aktual negatif yang diprediksi dengan benar sebagai negatif (True Negative, TN), 82 data aktual positif yang salah diprediksi sebagai negatif (False Negative, FN), 5 data aktual negatif yang salah diprediksi sebagai positif (False Positive, FP), dan 65 data aktual positif yang diprediksi dengan benar sebagai positif (True Positive, TP).

Terakhir, hasil prediksi keseluruhan dicetak untuk memberikan gambaran akhir tentang klasifikasi yang dilakukan oleh ensemble model. Maka dihasilkan arah opini masyarakat terkait harga tiket pesawat domestik yaitu Negatif.

Gambar 23. Visualisasi kata majority voting

4.14. Hasil dan Analisa

Ensemble Majority Voting mengkombinasikan prediksi dari dua model klasifikasi yang berbeda, yaitu *Naive Bayes* dan *Random Forest*, untuk menghasilkan keputusan final. Dalam analisis ini, mengevaluasi performa ensemble menggunakan metrik akurasi, yang merupakan ukuran umum untuk mengevaluasi keberhasilan model klasifikasi.

Tabel 3. Perbandingan akurasi

Skema	Naive Baye	Random Forest	Ensamble Majority Voting
80:20	70,31%	84,91%	85,88%
70:30	71,75%	85,71%	85,87%

Setelah melakukan skenario pengujian didapatkan parameter terbaik dari setiap metode, yang

selanjutnya akan dilakukan analisis lebih lanjut menggunakan majority voting. Pada Table 3 pada masing-masing skenario didapatkan nilai akurasi yang baik kemudian pada perbandingan pada masing-masing algoritma secara individu antara naive bayes dan random forest didapatkan klasifikasi random forest memiliki nilai akurasi yang tinggi dan konsisten pada masing-masing skenario pengujian yaitu pada 84,91% untuk skenario pertama 80:20 dan 85,71% pada skenario kedua 70:30.

Dari hasil perbandingan akurasi tersebut, dapat dilihat bahwa selisih akurasi antara menggunakan majority voting dan model individu dapat dipengaruhi oleh beberapa faktor. *Majority voting* cenderung lebih efektif ketika model-model yang digabungkan memiliki kesalahan yang tidak berkorelasi, yang berarti mereka membuat kesalahan yang berbeda pada data yang sama. Dalam kasus ini, *Naive Bayes* dan *Random Forest* menunjukkan pola kesalahan yang berbeda, sehingga ketika digabungkan, kesalahan masing-masing model dapat saling menutupi, menghasilkan peningkatan akurasi keseluruhan.

Model dengan akurasi individu tertinggi, yaitu *Random Forest*, memberikan kontribusi yang lebih besar dalam *majority voting*. Meskipun setiap model memberikan satu suara, model yang lebih akurat cenderung memberikan prediksi yang lebih benar secara konsisten, sehingga suara dari model tersebut dominan dalam hasil akhir. Ini menjelaskan mengapa akurasi *majority voting* lebih mendekati akurasi *Random Forest* dengan selisih rata-rata hanya 0,56%, tetapi dengan peningkatan karena kontribusi *Naive Bayes*.

Selisih 0,56% dalam akurasi antara *Random Forest* dan *majority voting* dalam kedua skenario menunjukkan bahwa penambahan *Naive Bayes* dalam *ensemble* membantu dalam menangani beberapa kesalahan yang dibuat oleh *Random Forest* sendiri, sehingga memberikan peningkatan kinerja keseluruhan. Penggunaan *ensemble majority voting* yang menggabungkan prediksi dari *Naive Bayes* dan *Random Forest* terbukti efektif dalam meningkatkan akurasi analisis sentimen, menunjukkan konsistensi dalam peningkatan nilai akurasi yang diperoleh.

5. KESIMPULAN DAN SARAN

Berdasarkan penilaian evaluasi, perbandingan tingkat akurasi antara algoritma Naive Bayes, Random Forest secara individu mendapatkan nilai akurasi Random Forest hasil yang cukup baik dengan nilai akurasi sebesar 84,91%, 85,71% lebih baik dari Naive Bayes, setelah menggunakan implementasi metode ensemble majority vote pada algoritma dalam analisis sentimen terhadap harga tiket pesawat domestik pada media sosial menhasilkan nilai akurasi yang baik dengan nilai 85,88%, 85,87% pada masing masing skenario. Dalam artian arah opini masyarakat terkait harga tiket pesawat domestik dalam media sosial Twitter adalah negatif.

DAFTAR PUSTAKA

[1] A. Purwanto, "Tahun 2024, Industri Penerbangan Optimistis Pulih dan Bangkit," Kompas.id, Dec. 2023. [Online]. Available: https://www.kompas.id/baca/riset/2023/12/09/tahun-2024-industri-penerbangan-optimis-pulih-dan-bangkit

[3] Y. D. Pusparisa, "Penerbangan Indonesia: Pasar Terbesar, Konektivitas Minim," Kompas.id, Nov. 2023. [Online]. Available: https://www.kompas.id/baca/ekonomi/2023/11/02/indonesia-berpotensi-jadi-pemain-utama-penerbangan-di-asean

[3] P. Pasek, O. Mahawardana, G. A. Sasmita, P. Agus, and E. Pratama, "Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap 'Figure Pemimpin' Menggunakan Python," 2022.

[4] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi, "Twitter and Research: A Systematic Literature Review through Text Mining," *IEEE Access*, vol. 8, pp. 67698–67717, 2020, doi: 10.1109/ACCESS.2020.2983656.

[5] R. Apriani *et al.*, "ANALISIS SENTIMEN DENGAN NAÏVE BAYES TERHADAP KOMENTAR APLIKASI TOKOPEDIA," 2019.

[6] E. Fitri, Y. Yuliani, S. Rosyida, and W. Gata, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine," *TRANSFORMTIKA*, vol. 18, no. 1, pp. 71–80, 2020, [Online]. Available: www.nusamandiri.ac.id,

[7] I. K. Hasan, R. Resmawan, and J. Ibrahim, "Perbandingan K-Nearest Neighbor dan Random Forest dengan Seleksi Fitur Information Gain untuk Klasifikasi Lama Studi Mahasiswa," *Indonesian Journal of Applied Statistics*, vol. 5, no. 1, p. 58, May 2022, doi: 10.13057/ijas.v5i1.58056.

[8] I. B. Ketut and S. Arnawa, "Analisis Sentimen pada Media Sosial Terhadap Perkuliahan Hybrid Menggunakan Algoritma TF IDF dan K Nearest Neighbor," *JURNAL SISTEM DAN INFORMATIKA (JSI)*, pp. 40–46, 2022.

[9] H. Taufiqqurrahman, F. Tri Anggraeny, and M. Muharrom Al Haromainy, "PERBANDINGAN ALGORITMA NAÏVE BAYES DAN K-NEAREST NEIGHBOR PADA ANALISIS SENTIMEN ULASAN APLIKASI MYPERTAMINA," 2023.

[10] F. Amin, "Implementasi Search Engine (Mesin Pencari) Menggunakan Metode Vector Space Model," Dinamika Teknik, vol. V, no. 1, pp. 45-48, 2011.

[11] M. Alfi, R. Reynaldhi, and Y. Sibaroni, "Analisis Sentimen Review Film pada Twitter menggunakan Metode Klasifikasi Hybrid SVM, Naïve Bayes, dan Decision Tree," n.d.