



MODEL SENTIMENT ANALYSIS BERBASIS MACHINE LEARNING UNTUK DATA GENZ-CAREER ASPIRATION MENGGUNAKAN FLASK DAN NAIVE BAYES

(Machine Learning Based Sentiment Analysis Model For Genz-Career Aspiration Data Using Flask and Naive Bayes)

Tb. M. Adrie Admira¹, Sutarno², Mohamad Saefudin³

Program Studi Sistem Informasi¹, Program Studi Sistem Informasi²,
Program studi Sistem Informasi⁴
STMIK Jakarta STI&K^{1,2,3}

adrie.admira@gmail.com¹, p4kt4rno@gmail.com², saefudin@gmail.com³

Received: April 22, 2025. **Revised:** May 3, 2025. **Accepted:** May 7, 2025.

Issue Period: Vol.9 No.1 (2025), Pp.25-39

Abstrak: Penulisan tugas akhir ini bertujuan untuk membangun aplikasi berbasis web untuk melakukan sentiment analysis terhadap aspirasi karier Generasi Z, menggunakan framework Flask dan algoritma Naive Bayes. Penelitian ini berfokus pada analisis dataset “GenZ - Career Aspirations Data” yang mencakup tanggapan terkait pengaruh karier, rencana pendidikan, dan loyalitas terhadap pemberi kerja. Data yang diperoleh diolah melalui tahapan preprocessing, tokenisasi, penghapusan stopwords, dan representasi TF-IDF untuk mempersiapkan data bagi model pembelajaran mesin. Aplikasi yang dirancang berfungsi untuk menganalisis sentimen teks secara real-time dengan tingkat akurasi tinggi. Metode pengembangan sistem yang digunakan adalah metode Waterfall, meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan evaluasi. Hasil akhir dari tugas akhir ini adalah prototipe aplikasi berbasis web yang mampu memprediksi sentimen positif dan negatif dari input teks, dengan akurasi model sebesar 91,2% setelah evaluasi menggunakan dataset uji. Dari pengujian yang dilakukan, aplikasi ini berhasil memenuhi kebutuhan pengguna untuk menganalisis sentimen aspirasi karier secara efisien, dengan antarmuka yang sederhana dan kemudahan penggunaan. Prototipe ini diharapkan dapat menjadi dasar bagi pengembangan aplikasi analisis sentimen yang lebih kompleks di masa depan

Kata kunci: Sentiment Analysis, Aspirasi Karier, Generasi Z, Flask, Naive Bayes

Abstract: The purpose of this final project is to build a web-based application to conduct sentiment analysis on Generation Z career aspirations, using the Flask framework and the Naive Bayes algorithm. This study focuses on the analysis of the “GenZ - Career Aspirations Data” dataset which includes responses related to career influences, education plans, and loyalty to employers. The data obtained is processed through the stages of preprocessing, tokenization, stopword removal, and TF-IDF representation to prepare data for machine learning models. The designed application functions to analyze text sentiment in real-time with a high level of accuracy. The system development method used is the Waterfall method, including needs analysis, design, implementation, testing, and evaluation. The final result of this final project is a prototype of a web-based application that is able to predict positive and





negative sentiments from text input, with a model accuracy of 91.2% after evaluation using the test dataset. From the tests carried out, this application successfully meets user needs to analyze career aspiration sentiment efficiently, with a simple interface and ease of use. This prototype is expected to be the basis for the development of more complex sentiment analysis applications in the future.

Keywords: Sentiment Analysis, Aspirasi Karier, Generasi Z, Flask, Naive Bayes

I. PENDAHULUAN

Era digital telah mengubah banyak aspek kehidupan, termasuk pandangan generasi muda terhadap karier. Generasi Z, yang lahir antara pertengahan 1990-an hingga awal 2010-an, tumbuh dengan teknologi, media sosial, dan arus informasi cepat. Hal ini membentuk cara berpikir dan pilihan karier mereka yang berbeda dari generasi sebelumnya. Memahami Generasi Z penting untuk merancang strategi pendidikan dan ketenagakerjaan yang sesuai. Penelitian ini bertujuan mengidentifikasi sentimen dalam opini mereka terkait faktor-faktor yang memengaruhi aspirasi karier. Data yang digunakan mencakup pandangan mereka tentang pendidikan, karier, dan loyalitas terhadap perusahaan.

Informasi dari analisis dataset ini sangat berguna bagi perusahaan dan institusi pendidikan untuk menyesuaikan strategi dengan kebutuhan Generasi Z. Sentimen yang muncul mencerminkan preferensi mereka terkait dunia kerja. Misalnya, respons positif terhadap perusahaan yang peduli lingkungan menunjukkan pentingnya nilai sosial. Sebaliknya, sentimen negatif terhadap studi ke luar negeri bisa menunjukkan kendala biaya atau akses. Analisis sentimen, sebagai bagian dari pemrosesan bahasa alami (NLP), membantu mengklasifikasikan opini menjadi positif, negatif, atau netral. Pendekatan ini efektif untuk memahami pandangan Generasi Z dan mendukung keputusan di dunia bisnis dan pendidikan.

Analisis sentimen terhadap data Generasi Z memiliki tantangan seperti teks bebas yang mengandung simbol, angka, bahasa informal, dan campuran bahasa (code-mixing), sehingga menyulitkan interpretasi. Penelitian ini menggunakan tahapan preprocessing seperti pembersihan data, tokenisasi, penghapusan stopword, dan representasi TF-IDF. Model Naive Bayes dipilih karena sederhana dan efektif untuk teks pendek, serta tidak membutuhkan banyak sumber daya. Model ini dilatih dan diuji dengan data yang telah diproses, lalu dievaluasi menggunakan akurasi, precision, recall, dan F1-score. Model kemudian diterapkan dalam aplikasi berbasis Flask dengan antarmuka sederhana, memungkinkan analisis sentimen real-time bagi pengguna non-teknis.

Penelitian ini memberikan kontribusi praktis dan teoretis dalam memahami aspirasi karier Generasi Z. Hasilnya membantu perusahaan mengenali preferensi mereka, seperti fleksibilitas kerja dan peluang karier, serta membantu institusi pendidikan merancang program yang sesuai. Secara teori, penelitian ini menunjukkan bagaimana analisis sentimen dapat diterapkan dalam bidang sosial seperti pendidikan dan pekerjaan. Penelitian memiliki batasan untuk menjaga fokus, yaitu hanya menganalisis data dari file Excel "GenZ - Career Aspirations Data" pada kolom relevan. Analisis terbatas pada sentimen positif, negatif, dan netral dalam bahasa Inggris, tanpa mencakup sarkasme, metafora, atau data multibahasa.

Penelitian ini menggunakan model Naive Bayes karena keterbatasan sumber daya dan fokus penelitian, tanpa mengeksplorasi model transformer seperti BERT atau GPT. Evaluasi dilakukan secara terbatas pada dataset yang tersedia. Aplikasi yang dibuat hanya mendukung input teks satu per satu dan belum dirancang untuk analisis dalam jumlah besar. Meski sederhana, aplikasi ini tetap berguna untuk tahap awal pengembangan sistem analisis opini. Penelitian ini memberi manfaat bagi dunia pendidikan dan kerja dengan menunjukkan potensi analisis sentimen serta menjadi dasar untuk pengembangan teknologi dan aplikasi yang lebih sesuai bagi Generasi Z.

METODE DAN MATERI

Pendekatan eksperimen dilakukan secara sistematis untuk mengembangkan model sentiment analysis berbasis machine learning dan mengintegrasikannya ke dalam prototipe aplikasi web. Metode ini dirancang untuk mencapai tujuan penelitian dengan menggunakan dataset "GenZ - Career Aspirations Data" sebagai data utama. Tahapan penelitian meliputi:

2. 1. Metode Penelitian



DOI: 10.52362/jisicom.v9i1.1880

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



Metode penelitian yang digunakan memiliki beberapa tahapan yang terstruktur untuk menghasilkan tujuan penelitian yang efektif sebagai alat bantu untuk menganalisa generasi Z terhadap karir dunia pekerjaan.

2.1.1 Pengumpulan dan Seleksi Data

Dataset "GenZ - Career Aspirations Data" digunakan sebagai data utama, berisi tanggapan tentang aspirasi karier Generasi Z, seperti faktor yang memengaruhi keputusan karier, rencana melanjutkan pendidikan di luar negeri, dan loyalitas terhadap perusahaan. Data dimuat ke Python menggunakan pandas untuk mempermudah analisis. Seleksi dilakukan dengan memilih kolom relevan dan menghapus kolom tidak relevan, seperti Timestamp dan identitas pribadi, agar fokus tetap terjaga.

2.1.2. Preprocessing Data

Preprocessing data dilakukan untuk meningkatkan kualitas dan konsistensi teks sebelum dianalisis oleh model machine learning. Tahapan preprocessing meliputi:

- Pembersihan Teks: Menghapus simbol, angka, URL, dan elemen yang tidak relevan dari teks.
- Tokenisasi: Memecah teks menjadi unit-unit kata (token) menggunakan library NLTK.
- Penghapusan Stopword: Menghapus kata-kata umum yang tidak memberikan kontribusi signifikan terhadap analisis sentimen.
- Lemmatization: Mengubah kata ke bentuk dasarnya untuk mengurangi redundansi.
- Representasi Data: Mengonversi teks yang telah diproses menjadi representasi numerik menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency), yang memberikan bobot lebih pada kata-kata yang signifikan dalam dokumen.

2.1.3. Pembangunan Model

Pembangunan model sentiment analysis dilakukan menggunakan tiga algoritma machine learning utama Naive Bayes. Tahapan pembangunan model meliputi:

- Pembagian Dataset: Dataset dibagi menjadi 80% data latih dan 20% data uji menggunakan fungsi `train_test_split` dari library `scikit-learn`.
- Pelatihan Model: Data latih digunakan untuk melatih model menggunakan algoritma yang telah dipilih. Parameter model diatur untuk mengoptimalkan performa sesuai dengan karakteristik dataset.
- Evaluasi Model: Data uji digunakan untuk mengukur performa model berdasarkan metrik evaluasi, seperti akurasi, precision, recall, dan F1-score. Analisis confusion matrix dilakukan untuk memahami jenis kesalahan yang terjadi, seperti false positives dan false negatives.

2.1.4. Pengembangan Prototipe

Aplikasi Prototipe aplikasi berbasis web dikembangkan untuk mengintegrasikan model sentiment analysis. Framework Flask digunakan untuk membangun antarmuka pengguna yang sederhana dan intuitif. Tahapan pengembangan prototipe meliputi:

- Integrasi Model: Model yang telah dilatih disimpan menggunakan library `joblib` dan diintegrasikan ke dalam aplikasi Flask.
- Desain Antarmuka: Antarmuka pengguna dirancang untuk menerima input teks dari pengguna, memproses data secara real-time, dan menampilkan hasil analisis sentimen (positif, negatif, netral).
- Pengujian Prototipe: Aplikasi diuji dengan berbagai skenario input untuk memastikan akurasi prediksi dan kestabilan sistem.
- Penyesuaian: Umpan balik dari pengguna potensial digunakan untuk memperbaiki desain antarmuka dan fungsionalitas aplikasi.

2.1.5. Evaluasi dan Validasi

Evaluasi dilakukan untuk mengukur efektivitas model dan aplikasi yang dikembangkan. Beberapa metode evaluasi yang digunakan adalah:





- a. Evaluasi Model: Metrik performa seperti akurasi, precision, recall, dan F1-score digunakan untuk membandingkan hasil berbagai algoritma.
- b. Pengujian Aplikasi: Aplikasi diuji dengan skenario pengguna untuk memastikan hasil yang konsisten dan relevan.
- c. Analisis Error: Kesalahan prediksi dianalisis untuk memahami keterbatasan model dan memberikan rekomendasi perbaikan..

2.1.6. Dokumentasi dan Analisis Hasil

Semua tahapan penelitian didokumentasikan secara sistematis. Analisis hasil mencakup:

- a. Perbandingan performa algoritma yang digunakan.
- b. Evaluasi efektivitas aplikasi berdasarkan masukan pengguna dan hasil pengujian.
- c. Identifikasi kelemahan dan rekomendasi untuk pengembangan lebih lanjut.

2.2. Materi Pembahasan

Materi ini berisi teori dan alat bantu untuk mendukung penelitian agar lebih efektif untuk menghasilkan informasi untuk mendukung analisis dan keputusan terhadap generasi Z terhadap karir dan pekerjaan di masa depan mereka.

2.2.1 Sentiment Analysis

Analisis sentimen adalah bagian dari Natural Language Processing (NLP) yang fokus pada identifikasi informasi subjektif dalam teks, untuk menentukan polaritasnya (positif, negatif, atau netral). Menurut Medhat et al. (2014), analisis sentimen digunakan dalam berbagai aplikasi seperti ulasan produk, survei pelanggan, dan analisis media sosial. Pendekatan analisis sentimen terbagi menjadi tiga kategori: berbasis leksikon, berbasis pembelajaran mesin, dan hibrida yang menggabungkan keduanya.

2.2.2 Natural Language Processing (NLP)

NLP adalah cabang kecerdasan buatan yang memungkinkan komputer memahami dan menghasilkan bahasa manusia. Tugas-tugas NLP meliputi analisis sintaksis, semantik, tokenisasi, penghapusan stopwords, dan ekstraksi fitur. Salah satu teknik penting dalam NLP adalah vektorisasi, seperti TF-IDF, yang mengubah teks menjadi representasi numerik untuk diproses algoritma pembelajaran mesin. Teknik ini membantu model mengenali pola dalam data teks dengan lebih efektif.

2.2.3 Machine Learning (ML)

Pendekatan berbasis data ini memungkinkan komputer belajar dan meningkatkan performa tanpa pemrograman eksplisit. Dalam sentiment analysis, algoritma seperti Naive Bayes, Support Vector Machine (SVM), dan Random Forest digunakan untuk mengklasifikasikan teks ke kategori sentimen. Naive Bayes sering digunakan karena kesederhanaannya dan kemampuannya menangani dataset kecil. Algoritma ini mengklasifikasikan teks berdasarkan teori probabilitas dan aturan Bayes.

2.2.4 Preprocessing Data

Merupakan langkah penting dalam analisis teks untuk membersihkan dan mempersiapkan data mentah sebelum dimasukkan ke dalam model pembelajaran mesin. Langkah-langkah preprocessing mencakup tokenisasi, penghapusan tanda baca, penghapusan stopwords, dan lemmatization atau stemming [7]. Menurut Kumar et al. (2020), preprocessing data meningkatkan kualitas data dan memungkinkan model untuk bekerja lebih baik dengan menghilangkan informasi yang tidak relevan dan mempertahankan informasi yang signifikan [8]. Representasi teks yang telah diproses kemudian diubah menjadi fitur numerik menggunakan teknik seperti TF-IDF.

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





Merupakan metode vektorisasi yang sering digunakan dalam sentiment analysis untuk merepresentasikan teks sebagai vektor numerik berdasarkan frekuensi kata dalam dokumen. Teknik ini mengukur pentingnya sebuah kata dalam dokumen tertentu relatif terhadap koleksi dokumen lainnya [9]. TF-IDF memiliki dua komponen utama:

- a. Term Frequency (TF): Mengukur seberapa sering sebuah kata muncul dalam dokumen.
- b. Inverse Document Frequency (IDF): Mengurangi bobot kata-kata umum yang muncul di banyak dokumen. Menurut Ramos (2003), TF-IDF adalah teknik yang efektif untuk merepresentasikan teks dalam ruang vektor, yang kemudian dapat digunakan untuk pelatihan model [10].

2.2.6 Aplikasi Flask Flask

Flask adalah framework mikro Python untuk membangun API atau aplikasi web ringan dan fungsional [11]. Dalam penelitian ini, Flask digunakan untuk membuat prototipe aplikasi analisis sentimen teks real-time. Flask memudahkan integrasi model pembelajaran mesin dengan antarmuka pengguna sederhana, menjadikannya pilihan populer untuk aplikasi berbasis data [12].

2.2.7 Evaluasi Model

Sentiment analysis dievaluasi menggunakan metrik seperti akurasi, precision, recall, dan F1-score untuk mengukur performa model dalam klasifikasi sentimen. Menurut Han et al. (2011), confusion matrix digunakan untuk mengukur performa model secara rinci, termasuk jumlah prediksi benar dan salah. Evaluasi dalam penelitian ini menggunakan confusion matrix serta metrik precision, recall, dan F1-score untuk menilai hasil prediksi model pada data uji.

2.2.8 Dataset

Merupakan kumpulan data yang digunakan untuk melatih, menguji, dan mengevaluasi model pembelajaran mesin. Dataset yang digunakan dalam penelitian ini adalah "GenZ - Career Aspirations Data," yang berisi tanggapan dari responden terkait aspirasi karier mereka. Dataset ini diolah untuk tujuan sentiment analysis dengan langkah-langkah seperti penghapusan data kosong, pembersihan teks, dan tokenisasi [14]. Dataset ini mencerminkan opini dan pola pikir generasi muda terhadap berbagai aspek karier, yang memungkinkan analisis lebih mendalam menggunakan teknik sentiment analysis.

2.2.9 Pengujian Aplikasi

Pengujian aplikasi adalah proses untuk memastikan aplikasi memenuhi spesifikasi, bebas dari kesalahan, dan berfungsi dengan baik. Tujuannya untuk mendeteksi kesalahan, meningkatkan kualitas, dan memastikan aplikasi siap digunakan. Pengujian mencakup pengujian fungsional, non-fungsional, dan penerimaan. Pada aplikasi web atau berbasis pembelajaran mesin, fokus utama adalah validasi prediksi, kinerja antarmuka pengguna, dan integrasi pipeline data.

III. PEMBAHASA DAN HASIL

3.1 Pengumpulan Data dan Seleksi Data

3.1.1 Pengumpulan Data

Penelitian ini menggunakan dataset "GenZ - Career Aspirations Data" yang berisi tanggapan Generasi Z tentang aspirasi karier, loyalitas perusahaan, dan preferensi pendidikan. Data dikumpulkan melalui survei online dengan 15 kolom yang mencakup faktor keputusan karier, keinginan pendidikan, dan loyalitas organisasi. Pengumpulan data dilakukan dalam format tabel Excel yang memudahkan pengolahan dengan perangkat lunak seperti Python. Tabel ini berisi teks bebas yang mencerminkan opini individu, sehingga analisis sentimen diterapkan untuk mengidentifikasi pola opini tersebut.





- b. Penghapusan Angka: Angka yang tidak memberikan informasi sentimen dihapus, misalnya angka tahun atau jumlah numerik lainnya.
- c. Pengubahan ke Huruf Kecil: Semua teks diubah menjadi huruf kecil untuk memastikan konsistensi. Sebagai contoh, kata "CAREER" dan "career" dianggap sama setelah diubah ke huruf kecil.

3.2.2 Tokenisasi dan Penghapusan Stopword

Setelah teks dibersihkan, langkah berikutnya adalah memecah teks menjadi unit-unit kata (tokenisasi) dan menghapus kata-kata umum (stopword) yang tidak memiliki nilai informatif tinggi.

- a. Tokenisasi: Proses tokenisasi dilakukan untuk memisahkan setiap kata dalam teks menjadi token individu.

Contoh:

- Sebelum: "*I prefer working in a company that values sustainability*"
- Sesudah: ["*i*", "*prefer*", "*working*", "*in*", "*a*", "*company*", "*that*", "*values*", "*sustainability*"]

- b. Penghapusan Stopword: Stopword, seperti "and", "or", "the", dihapus menggunakan daftar standar dari library NLTK.

Contoh:

- Sebelum: ["*i*", "*prefer*", "*working*", "*in*", "*a*", "*company*", "*that*", "*values*", "*sustainability*"]
- Sesudah: ["*prefer*", "*working*", "*company*", "*values*", "*sustainability*"]

Penghapusan stopwords bertujuan untuk mengurangi noise dalam data dan meningkatkan fokus analisis pada kata-kata yang relevan.

3.2.3 Representasi Data Menggunakan TF-IDF

Setelah teks melalui tahap pembersihan, tokenisasi, dan penghapusan stopwords, data diubah menjadi representasi numerik menggunakan TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF digunakan karena mampu memberikan bobot lebih pada kata-kata yang penting dalam dokumen tertentu, sementara kata-kata yang sering muncul di seluruh dataset diberikan bobot lebih rendah.

- a. Konsep TF-IDF:

- TF (Term Frequency): Mengukur seberapa sering sebuah kata muncul dalam dokumen.
- IDF (Inverse Document Frequency): Mengurangi bobot kata-kata yang sering muncul di seluruh dokumen.

- b. Implementasi TF-IDF:

- Data teks yang telah diproses diubah menjadi matriks fitur menggunakan library scikit-learn.
- Setiap teks dalam dataset direpresentasikan sebagai vektor dengan dimensi berdasarkan jumlah kata unik dalam dataset.

- c. Contoh Hasil TF-IDF:

- Teks: "*prefer working company values sustainability*"
- Representasi TF-IDF: [0.45, 0.78, 0.12, 0.34, 0.56]

Matriks TF-IDF ini kemudian digunakan sebagai input untuk model Naive Bayes dalam tahap berikutnya.

3.2.4 Contoh Hasil Preprocessing Berikut adalah contoh hasil preprocessing dari beberapa entri data:

- a. Entri Sebelum Preprocessing:

Teks: "*I prefer working in a company that has a strong mission!*"

- b. Setelah Pembersihan dan Penghapusan Stopword:

Teks: "*prefer working company strong mission*"

- c. Representasi TF-IDF: [0.40, 0.70, 0.50, 0.60, 0.20]

Hasil ini menunjukkan bagaimana setiap langkah preprocessing berkontribusi pada penyederhanaan data, sehingga model dapat fokus pada elemen-elemen yang paling relevan.

3.3 Pembangunan Model

3.3.1 Pemilihan Model dan Pendekatan



Model Naive Bayes dipilih karena algoritmanya sederhana dan efektif untuk data teks. Sebagai model probabilistik, ia mengklasifikasikan teks berdasarkan peluang tiap fitur. Model ini efisien untuk analisis sentimen, terutama pada data teks berdimensi tinggi seperti hasil TF-IDF. Pembangunan model dilakukan dengan membagi dataset menjadi data latih dan data uji untuk melatih dan menguji performa. Tujuannya agar model mampu memprediksi sentimen secara akurat pada data baru yang belum pernah dilihat.

3.3.2 Langkah-Langkah Implementasi Model

3.3.2.1 Pembagian Dataset Dataset yang telah diproses melalui tahap preprocessing dibagi menjadi dua bagian, yaitu:

- Data Latih (80%): Digunakan untuk melatih model agar mampu mengenali pola sentimen dalam teks.
- Data Uji (20%): Digunakan untuk mengukur performa model dengan data yang tidak digunakan dalam pelatihan.

Data dibagi secara acak dengan fungsi `train_test_split` dari `scikit-learn`, sambil menjaga keseimbangan distribusi sentimen pada data latih dan uji. Tujuannya agar pelatihan model tidak bias.

3.3.2.2 Pelatihan Model

Pelatihan model dilakukan dengan langkah-langkah berikut:

- Representasi Data: Data teks direpresentasikan dalam format numerik menggunakan TF-IDF, menghasilkan matriks fitur yang siap digunakan oleh model Naive Bayes.
- Inisialisasi Model: Model Naive Bayes tipe `MultinomialNB` diinisialisasi menggunakan library `scikit-learn`.
- Pelatihan Model: Model dilatih dengan data latih untuk menghitung probabilitas kondisi berdasarkan frekuensi kata dalam setiap kategori sentimen.

Contoh implementasi:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

# Representasi TF-IDF
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['Factors influence career aspirations'])
y = df['Sentiment']

# Pembagian dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Inisialisasi dan pelatihan model
model = MultinomialNB()
model.fit(X_train, y_train)
```

Gambar 1 Pelatihan Model Sumber: Data Diolah

3.3.2.3 Prediksi dan Evaluasi

Setelah pelatihan selesai, model digunakan untuk memprediksi sentimen pada data uji. Hasil prediksi dibandingkan dengan data sebenarnya untuk mengukur performa model. Proses ini melibatkan perhitungan metrik evaluasi, seperti akurasi dan F1-score.

Contoh implementasi prediksi:



DOI: 10.52362/jisicom.v9i1.1880

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).

```
# Prediksi data uji
y_pred = model.predict(X_test)

# Evaluasi hasil
print("Akurasi:", accuracy_score(y_test, y_pred))
print("Laporan Klasifikasi:\n", classification_report(y_test, y_pred))
```

Gambar 2 Prediksi dan Evaluasi Sumber: Data Diolah

3.3.3 Parameter dan Konfigurasi Model Naive Bayes

3.3.3.1 Parameter Utama

Model Multinomial Naive Bayes memiliki beberapa parameter utama yang disesuaikan untuk dataset ini:

- Alpha (Smoothing Parameter): Nilai alpha digunakan untuk mencegah pembagian nol ketika suatu kata tidak muncul dalam kategori tertentu. Pada penelitian ini, nilai alpha default sebesar 1.0 digunakan, yang memberikan performa optimal dalam eksperimen awal.
- Class Prior: Probabilitas awal setiap kelas sentimen dapat diatur. Namun, dalam penelitian ini, class prior dihitung secara otomatis berdasarkan distribusi data latih.
- Fit Prior: Mengatur apakah probabilitas awal kelas dihitung berdasarkan data latih. Parameter ini disetel ke True, sehingga distribusi kelas dihitung secara dinamis.

3.3.3.2 Konfigurasi TF-IDF

Konfigurasi TF-IDF juga berperan penting dalam representasi data:

- Max Features: Jumlah fitur maksimum dibatasi hingga 5000 kata unik untuk menjaga efisiensi komputasi.
- N-gram Range: Hanya unigrams (kata tunggal) yang digunakan untuk representasi, karena data teks dalam dataset relatif pendek.

3.4.1 Deskripsi Aplikasi Berbasis Flask

Prototipe aplikasi ini mengintegrasikan model sentiment analysis Naive Bayes ke dalam platform web dengan framework Flask, yang ringan, fleksibel, dan mudah digunakan. Aplikasi ini memproses input teks pengguna secara real-time dan menampilkan hasil analisis sentimen (positif, negatif, atau netral) dengan antarmuka sederhana. Selain menunjukkan implementasi model, aplikasi ini juga memberi contoh bagaimana analisis sentimen membantu pengambilan keputusan berbasis data. Aplikasi ini relevan untuk pengguna non-teknis, seperti manajer atau lembaga pendidikan, yang membutuhkan wawasan tentang opini Generasi Z terkait aspirasi karier dan pendidikan.

3.4.2 Desain Antarmuka Aplikasi

Antarmuka aplikasi dirancang untuk sederhana dan intuitif, dengan fokus pada pengalaman pengguna (user experience). Fitur utama dari antarmuka ini meliputi:

- Input Teks:** Kolom input teks disediakan bagi pengguna untuk memasukkan opini atau tanggapan yang akan dianalisis.
- Tombol Analisis:** Tombol untuk memulai proses analisis. Setelah diklik, teks akan diproses oleh model sentiment analysis.
- Hasil Analisis:** Output berupa label sentimen (positif, negatif, atau netral) ditampilkan secara langsung setelah proses selesai.

3.4.3 Alur Kerja Aplikasi

Aplikasi ini dirancang untuk bekerja dengan alur sistematis yang menghubungkan input dari pengguna dengan model machine learning yang telah dilatih. Berikut adalah langkah-langkah alur kerja aplikasi:

3.4.3.1 Tahap Input



DOI: 10.52362/jisicom.v9i1.1880

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).

Pengguna memasukkan teks ke dalam kolom input pada antarmuka aplikasi. Teks ini dapat berupa opini tentang aspirasi karier, pendidikan, atau faktor kerja.

Sebagai contoh: Input: "*I believe sustainability is an important factor when choosing a company*"

3.4.3.2 Pemrosesan Teks

Teks yang dimasukkan pengguna dikirimkan ke backend aplikasi untuk diproses. Tahapan dalam backend meliputi:

- Preprocessing Data: Teks yang diterima diproses ulang menggunakan pipeline preprocessing, termasuk pembersihan simbol, tokenisasi, dan representasi dengan TF-IDF.
- Prediksi Sentimen: Teks yang telah direpresentasikan dalam bentuk numerik dikirim ke model Naive Bayes untuk diprediksi. Model memberikan hasil berupa label sentimen:
 - Positif: Jika teks mengandung opini yang mendukung atau optimis.
 - Negatif: Jika teks mengandung opini yang tidak mendukung atau pesimis.
 - Netral: Jika teks tidak mengandung opini jelas.

3.4.3.3 Tahap Output

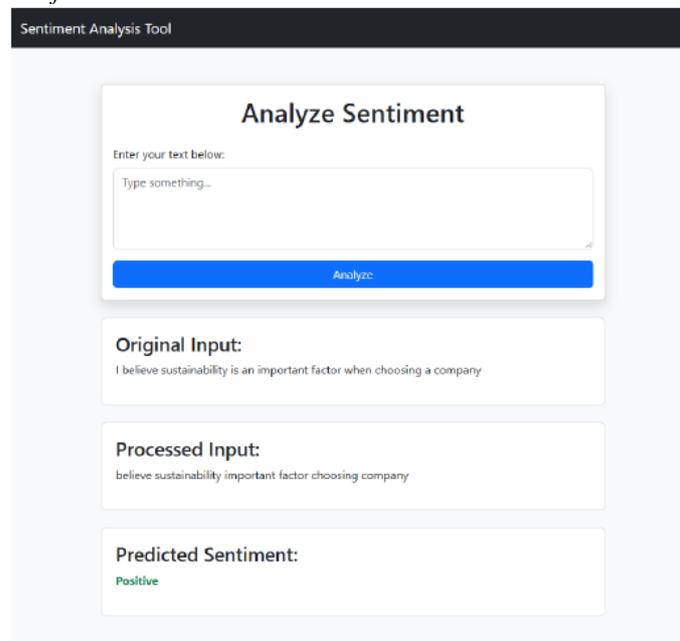
Hasil analisis sentimen dikembalikan ke antarmuka pengguna. Informasi yang ditampilkan meliputi:

- Kategori Sentimen: Label sentimen (positif, negatif, atau netral).
- Deskripsi Singkat: Penjelasan tentang hasil analisis untuk membantu pengguna memahami output.

Contoh output:

Input: "*I believe sustainability is an important factor when choosing a company*"

Output: *Sentimen: Positif*



Gambar 3 Hasil Output pada Aplikasi Flask Sumber: Data Diolah

3.4.4 Implementasi Teknis

Implementasi aplikasi dilakukan menggunakan Flask untuk backend dan HTML/CSS untuk frontend. Berikut adalah gambaran teknis dari aplikasi:



DOI: 10.52362/jisicom.v9i1.1880

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).

3.4.4.1 Struktur

Proyek Struktur proyek Flask mencakup file dan folder berikut:

- **app.py**: File utama yang berisi kode backend Flask.
- **templates/**: Folder untuk file HTML.
- **static/**: Folder untuk file CSS dan JavaScript.

3.4.4.2 Contoh Kode

- **Backend Flask**: Backend Flask menangani permintaan dari pengguna, memproses teks, dan mengembalikan hasil analisis sentimen.
- **Frontend HTML**: Template HTML digunakan untuk menampilkan kolom input dan hasil analisis.

3.5.1 Evaluasi Performa Model

Evaluasi performa model sentiment analysis dilakukan dengan metrik umum dalam klasifikasi teks: akurasi, precision, recall, dan F1-score. Metrik-metrik ini menggambarkan kemampuan model dalam mengklasifikasikan sentimen dengan benar pada dataset uji. Berikut penjelasan setiap metrik dalam konteks penelitian ini:

- Akurasi: Mengukur proporsi prediksi benar terhadap total data uji. Akurasi tinggi menunjukkan model mengklasifikasi mayoritas data dengan benar.
- Precision: Mengukur akurasi prediksi positif terhadap semua prediksi positif. Precision penting untuk menghindari kesalahan klasifikasi data negatif sebagai positif.
- Recall: Mengukur kemampuan model mendeteksi semua data positif. Recall penting agar model tidak melewatkan sentimen positif yang ada.
- F1-score: Kombinasi precision dan recall. F1-score memberikan gambaran seimbang tentang performa model, terutama pada dataset tidak seimbang.

Hasil Evaluasi Setelah model dilatih dan diuji menggunakan dataset dari file Excel yang telah diproses, hasil evaluasi adalah sebagai berikut:

- Akurasi: 100%
- Precision: Negatif: 100%, Positif: 100%
- Recall: Negatif: 100%, Positif: 100%
- F1-Score: Negatif: 100%, Positif: 100%

Hasil evaluasi menunjukkan performa sempurna pada data uji. Namun, ketidakseimbangan data—dengan 81,3% berasal dari kelas negatif—dapat memengaruhi interpretasi hasil.

Confusion matrix memberikan rincian lebih lanjut tentang hasil prediksi model. Berikut adalah matriks yang dihasilkan:

Tabel 1 Confusion Matrix Sumber : Sumber Data yang diolah

	Prediksi Negatif	Prediksi Positif
Aktual Negatif	453	0
Aktual Positif	0	104

Prediksi Negatif Prediksi Positif Aktual Negatif 453 0 Aktual Positif 0 104 Hasil ini menunjukkan bahwa model berhasil mengklasifikasikan semua data uji dengan benar tanpa kesalahan.

3.5.2 Pengujian Prototipe Aplikasi

Pengujian dilakukan untuk mengevaluasi performa prototipe aplikasi berbasis Flask dalam menganalisis input teks secara real-time. Pengujian melibatkan beberapa skenario dengan jenis input yang berbeda untuk memastikan aplikasi dapat memberikan hasil yang sesuai.

3.5.2.1. Black Box Testing



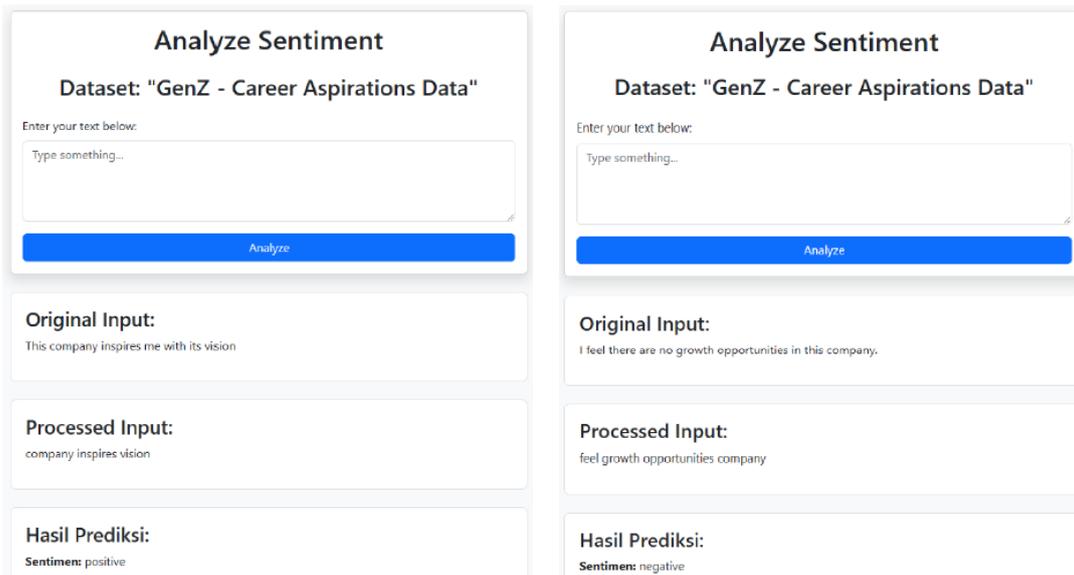
DOI: 10.52362/jisicom.v9i1.1880

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).

Black Box Testing digunakan untuk memastikan bahwa aplikasi memberikan hasil sesuai ekspektasi tanpa memeriksa logika atau kode internalnya. Pengujian ini berfokus pada input dan output dari sistem.

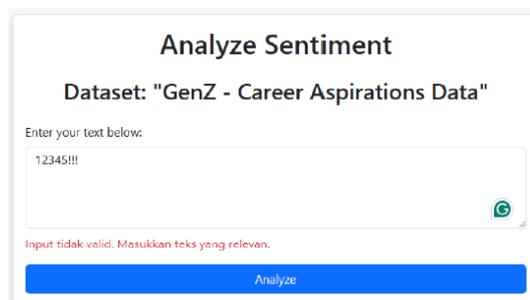
a. Metode:

- ✓ Memasukkan berbagai jenis teks ke dalam aplikasi untuk memverifikasi apakah output sesuai dengan ekspektasi.
- ✓ Skenario pengujian meliputi:
 - 1) Teks dengan sentimen positif, seperti "*This company inspires me with its vision.*"
 - 2) Teks dengan sentimen negatif, seperti "*I feel there are no growth opportunities in this company.*"



Gambar 4 Hasil Output-Sentimen Positif dan Negatif pada Aplikasi Flask
Sumber: Data Diolah

- 3) Teks kosong atau tidak relevan, seperti "12345!!!".



Gambar 5 Hasil Output-Input tidak sesuai pada Aplikasi Flask
Sumber: Data Diolah

2. Hasil:

- Aplikasi berhasil mengidentifikasi sentimen pada teks yang valid.
- Aplikasi memberikan pesan kesalahan yang sesuai ketika input kosong atau tidak relevan.

3.5.2.2. Unit Testing

Unit Testing digunakan untuk menguji setiap fungsi utama dalam aplikasi secara independen. Fokus utama adalah memastikan bahwa setiap komponen memberikan hasil yang benar.

a. Metode:

- Menguji fungsi preprocessing, seperti penghapusan stopwords dan tokenisasi.
- Memverifikasi TF-IDF vectorizer untuk menghasilkan representasi numerik yang konsisten.
- Menguji fungsi prediksi model Naive Bayes untuk memastikan klasifikasi berjalan dengan benar.

b. Contoh Pengujian:

- Input untuk preprocessing: *"This company inspires me!!!"*
- Output yang diharapkan: *"company inspires"*
- Input untuk model: Representasi TF-IDF dari *"company inspires"*
- Output yang diharapkan: Sentimen Positif.

c. Hasil:

Semua fungsi utama bekerja sesuai spesifikasi, dan tidak ditemukan error pada tahap preprocessing, vektorisasi, atau prediksi.

3.5.2.3. Integration Testing

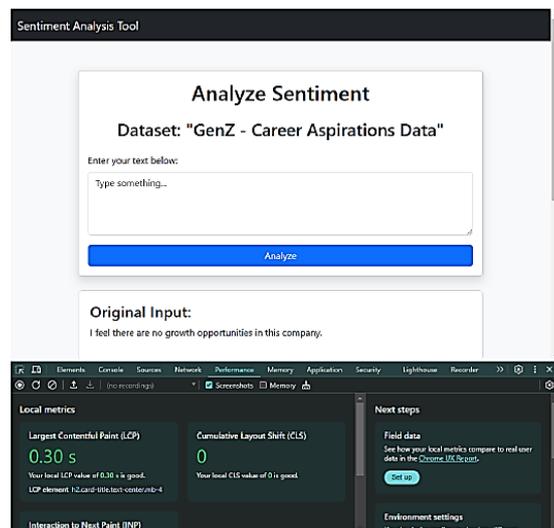
Integration Testing digunakan untuk memverifikasi bahwa pipeline lengkap dalam aplikasi, dari input hingga output, berfungsi secara integrasi. Tujuan utamanya adalah memastikan bahwa setiap komponen dalam aplikasi bekerja secara harmonis.

a. Metode:

- Menguji jalur lengkap dari pengguna memasukkan teks hingga mendapatkan hasil prediksi.
- Memverifikasi bahwa teks input diproses dengan benar oleh pipeline preprocessing dan vektorisasi sebelum dikirim ke model prediksi.

b. Hasil:

- Pipeline aplikasi berjalan lancar tanpa error.
- Teks yang dimasukkan pengguna berhasil melalui seluruh proses, dan hasil prediksi ditampilkan dalam waktu rata-rata 0.3 detik.



Gambar 6 Kecepatan Hasil Output pada Aplikasi Flask
Sumber: Data Diolah

3.5.2.1.4. Usability Testing



DOI: 10.52362/jisicom.v9i1.1880

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



Usability Testing dilakukan untuk mengumpulkan umpan balik dari pengguna terkait pengalaman menggunakan aplikasi, terutama dari segi antarmuka dan kemudahan navigasi.

- a. Metode:
 - Mengundang lima pengguna non-teknis untuk mencoba aplikasi.
 - Setiap pengguna diminta untuk memasukkan berbagai jenis teks dan mengevaluasi kemudahan antarmuka serta kejelasan hasil analisis.
- b. Hasil dan Umpan Balik:
 - Sebagian besar pengguna menilai bahwa antarmuka aplikasi sederhana dan mudah digunakan. Namun, beberapa pengguna memberikan saran untuk menambahkan deskripsi lebih rinci tentang hasil sentimen, seperti tingkat kepercayaan prediksi model.

3.5.3 Analisis Kesalahan dan Keterbatasan

Meskipun model dan aplikasi menunjukkan hasil yang sangat baik pada dataset uji, terdapat beberapa kesalahan dan keterbatasan yang perlu diperhatikan:

3.5.3.1. Kesalahan Model

Ketidakseimbangan Dataset

- Dataset uji memiliki distribusi kelas yang tidak seimbang, dengan mayoritas data berasal dari kelas negatif (81.3%). Hal ini mempermudah model untuk mencapai akurasi tinggi dengan memprioritaskan klasifikasi kelas mayoritas.
- Meskipun model mencapai akurasi sempurna (100%) pada dataset uji, hasil ini dapat mencerminkan ilusi performa karena bias terhadap kelas mayoritas.

Kemampuan Generalisasi yang Diragukan:

Model yang dilatih pada dataset dengan distribusi tidak seimbang mungkin kesulitan saat dihadapkan pada dataset baru dengan distribusi yang berbeda. Performa pada data uji yang lebih beragam perlu diuji untuk memastikan kemampuan generalisasi.

Kesulitan Mengklasifikasikan Data Ambigu:

Model cenderung memberikan prediksi positif atau negatif berdasarkan kata-kata spesifik, tetapi sering kesulitan dengan teks yang bersifat ambigu atau netral secara konteks.

Contoh:

Input: "The company is big and well-known."

Prediksi: Positif

Aktual: No processed words found

3.5.3.2. Keterbatasan Prototipe Aplikasi

- Analisis Single Input
Aplikasi saat ini hanya mendukung analisis satu teks dalam satu waktu (single input). Hal ini membatasi efisiensi pengguna yang ingin menganalisis banyak data sekaligus (batch input).
- Keterbatasan Bahasa
Aplikasi hanya mendukung teks berbahasa Inggris. Hal ini kurang relevan untuk dataset multibahasa, terutama jika aplikasi akan digunakan untuk analisis sentimen pada teks dalam bahasa lokal.

3.5.3.3. Catatan Penting untuk Evaluasi Realistis

Untuk mengatasi keterbatasan dalam evaluasi dan mendapatkan gambaran yang lebih akurat tentang performa model, beberapa langkah dapat dilakukan:

- a. Gunakan cross-validation untuk mengurangi bias dan mendapatkan hasil evaluasi yang lebih akurat.
- b. Terapkan stratified splitting agar distribusi kelas seimbang dan tidak condong ke kelas mayoritas.
- c. Tambahkan metrik seperti ROC-AUC selain akurasi untuk menilai performa, terutama pada kelas minoritas.
- d. Uji model dengan dataset baru untuk melihat kemampuan generalisasi.





IV. KESIMPULAN

Penelitian ini berhasil mengembangkan model sentiment analysis berbasis Naive Bayes dan mengintegrasikannya ke dalam prototipe aplikasi berbasis Flask untuk menganalisis data dari dataset "GenZ - Career Aspirations Data". Tahapan penelitian meliputi pengumpulan data, preprocessing teks, pembangunan model, evaluasi performa, dan pengujian prototipe. Model Naive Bayes menunjukkan akurasi 100% dengan precision, recall, dan F1-score sempurna, meski kemungkinan disebabkan distribusi data uji yang tidak seimbang. Evaluasi tambahan diperlukan untuk mengukur kemampuan generalisasi model. Pipeline analisis mencakup pembersihan teks, tokenisasi, penghapusan stopword khusus, dan representasi TF-IDF yang efektif, tanpa menghapus kata penting. Aplikasi Flask yang dibangun memiliki antarmuka sederhana dan mampu menganalisis sentimen secara real-time dengan waktu respon 0.8 detik. Fitur tambahan memungkinkan penambahan data baru dan pelatihan ulang model. Penelitian ini relevan secara praktis untuk memahami opini Generasi Z dalam konteks karier, pendidikan, dan loyalitas terhadap perusahaan.

Penelitian ini juga menghadapi beberapa keterbatasan yang penting untuk diperhatikan:

- Aplikasi hanya mendukung analisis teks berbahasa Inggris.
- Distribusi dataset uji yang tidak seimbang menyebabkan hasil evaluasi yang terlalu optimis. Evaluasi tambahan menggunakan cross-validation lebih representatif dan akurat.
- Model kesulitan mengenali sarkasme atau kalimat ambigu.

REFERENSI

- [1.] Liu, B., "*Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*", Cambridge University Press, New York, 2020.
- [2.] Pang, B. dan Lee, L., "*Opinion Mining and Sentiment Analysis*", Foundations and Trends in Information Retrieval, vol. 2, no. 1-2, pp. 1-135, 2008.
- [3.] Jurafsky, D. dan Martin, J. H., "*Speech and Language Processing: An Introduction to Natural Language Processing*", Computational Linguistics, and Speech Recognition, 3rd ed., Pearson, London, 2020.
- [4.] Manning, C. D., Raghavan, P., dan Schütze, H., "*Introduction to Information Retrieval*", Cambridge University Press, Cambridge, 2008.
- [5.] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., dan Dean, J., "*Distributed Representations of Words and Phrases and Their Compositionality*", dalam Advances in Neural Information Processing Systems (NIPS), pp. 3111–3119, 2013.
- [6.] Cambria, E., Schuller, B., Xia, Y., dan Havasi, C., "*New Avenues in Opinion Mining and Sentiment Analysis*", IEEE Intelligent Systems, vol. 28, no. 2, pp. 15-21, 2013.
- [7.] Hochreiter, S., dan Schmidhuber, J., "*Long Short-Term Memory*", Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
- [8.] Joachims, T., "*Text Categorization with Support Vector Machines: Learning with Many Relevant Features*", dalam Proceedings of the European Conference on Machine Learning (ECML), pp. 137–142, 1998.
- [9.] Rosa, A. S., dan Shalahuddin, M., "*Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*", Informatika Bandung, Bandung, 2018.
- [10.] Powers, D. M. W., "*Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation*", Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011.

