

PENERAPAN *MULTINOMIAL NAIVE BAYES* DALAM ANALISIS SENTIMEN KOMENTAR *YOUTUBE* TERKAIT MOBIL LISTRIK *AYLA EV*

Achmad Khoiri¹, Wahyu Pramusinto^{2*}

^{1,2}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: ¹2011501182@student.budiluhur.ac.id, ^{2*}wahyu.pramusinto@budiluhur.ac.id
(* : corresponding author)

Abstrak- Analisis sentimen *YouTube* merupakan salah satu proses analisis data pada kolom komentar video *YouTube* untuk menentukan dan mengategorikan sentimen atau perspektif yang diluapkan dalam sebuah teks. Proses analisis ini akan menghasilkan dua kategori sentimen, yaitu sentimen positif dan negatif. Analisis ini dilakukan terhadap salah satu video *YouTube* yang membahas kemunculan Kendaraan Listrik atau *Electric Vehicle (EV) Ayla* era teknologi 4.0 sebagai solusi transportasi yang hemat energi dan ramah lingkungan, berpotensi mengurangi ketergantungan pada minyak bumi. Di Indonesia, pemerintah aktif mempromosikan penggunaan kendaraan listrik melalui berbagai kebijakan, termasuk Peraturan Presiden (PP) No. 55 Tahun 2019. Namun, adopsi *Electric Vehicle (EV)* di Indonesia menuai beragam tanggapan, baik pro maupun kontra. Pendukung *Electric Vehicle (EV)* memandang kendaraan listrik sebagai langkah penting untuk mengurangi ketergantungan pada minyak bumi dan menurunkan emisi karbon. Di sisi lain, tantangan seperti tingginya biaya, terbatasnya infrastruktur pengisian daya, dan kekhawatiran tentang ketersediaan bahan baku baterai menjadi alasan kontra dari sebagian masyarakat. Tujuan dari penelitian ini adalah menganalisis sentimen masyarakat terhadap kendaraan listrik menggunakan algoritma *Multinomial Naive Bayes* pada data komentar *YouTube*. Sebanyak 700 data dikumpulkan dengan bahasa pemrograman *python* yang terkoneksi *API YouTube*, lalu diolah melalui *text preprocessing* data berkurang menjadi 659 data, dilakukan *labeling* menggunakan *InSet lexicon* dengan sentimen positif (299) serta sentimen negatif (234), dan dibagi menjadi data latih dan data uji dengan perbandingan yang berbeda, yaitu 60:40, 70:30, 80:20. Berdasarkan hasil evaluasi kinerja model melalui tiga perbandingan rasio yang berbeda, didapatkan bahwa perbandingan (60:40) mendapatkan akurasi terbaik sebesar 77.57%, presisi 86.18%, *recall* 77.37%, dan *f1-score* 81.54%.

Kata Kunci: Analisis Sentimen, *YouTube*, *Text Mining*, *Multinomial Naive Bayes*, *Electric Vehicle (EV)*

APPLICATION OF *MULTINOMIAL NAIVE BAYES* IN SENTIMENT ANALYSIS OF *YOUTUBE COMMENTS* RELATED TO *AYLA EV ELECTRIC CAR*

Abstract- *YouTube sentiment analysis is a process of analysing data in YouTube video comment sections to determine and categorise the sentiment or perspective expressed in a text. This analysis process will produce two categories of sentiment, namely positive and negative sentiment. This analysis was conducted on one YouTube video that discusses the emergence of Electric Vehicle (EV) Ayla in the era of technology 4.0 as an energy-efficient and environmentally friendly transportation solution, potentially reducing dependence on petroleum. In Indonesia, the government is actively promoting the use of electric vehicles through various policies, including Presidential Regulation (PP) No. 55 of 2019. However, the adoption of Electric Vehicles (EVs) in Indonesia has been met with mixed responses, both pro and con. Electric Vehicle (EV) proponents see electric vehicles as an important step towards reducing dependence on petroleum and lowering carbon emissions. On the other hand, challenges such as the high cost, limited charging infrastructure, and concerns about the availability of battery raw materials are the reasons for the cons. The purpose of this study is to analyse public sentiment towards electric vehicles using Multinomial Naive Bayes algorithm on YouTube comment data. A total of 700 data were collected with the python programming language connected to the YouTube API, then processed through text preprocessing the data was reduced to 659 data, labelled using InSet lexicon with positive sentiment (299) and negative sentiment (234), and divided into training data and test data with different comparisons, namely 60:40, 70:30, 80:20. Based on the results of evaluating the performance of the model through three different ratio comparisons, it is found that the ratio (60:40) gets the best accuracy of 77.57%, precision 86.18%, recall 77.37%, and f1-score 81.54%.*

Keywords: *Sentiment Analysis, YouTube, Text Mining, Multinomial Naive Bayes, Electric Vehicles (EV)*

1. PENDAHULUAN

Perkembangan zaman telah mendorong kemajuan teknologi yang begitu pesat. Salah satunya adalah kemudahan penyebaran informasi yang dapat dijangkau oleh siapapun, dimanapun, dan kapanpun. Penggunaan

gadget atau gawai yang terkoneksi internet membuat masyarakat lebih mudah mendapatkan informasi hingga berinteraksi di jejaring media sosial, seperti *YouTube*. Salah satu *channel YouTube* yang memberikan informasi mengenai kemajuan teknologi di Indonesia, yaitu *LugNutz Auto Junkie*. *Channel* ini memberikan informasi mengenai kemajuan teknologi mobil listrik *Ayla EV* di Indonesia dengan jumlah komentar sebanyak 700 data komentar *YouTube* pada video yang berjudul “Akhirnya Jepang Bikin Mobil Listrik Murah! *Daihatsu Ayla EV* Siap Bikin Pusing *Hyundai & Wuling*”.

YouTube merupakan salah satu media sosial populer gratis yang dapat melakukan berbagai media, mulai dari media video hingga suara. Pada Oktober 2023, *user platform YouTube* di Indonesia mencapai 139 juta *user*. Platform *Youtube* menyediakan berbagai konten video yang tersebar luas, bahkan tiap konten mengandung berbagai komentar positif hingga negatif [1]. Hal ini membuat, komentar pada platform *YouTube* sering dijadikan sebagai dataset dalam penelitian topik yang sedang tren hingga analisis sentimen pada *text mining*.

Penelitian terdahulu yang dengan topik “Analisis Sentimen Pada *Twitter* Terhadap Mobil Listrik Menggunakan Algoritma *Naive Bayes*”, penelitian ini melakukan *data scrapping* dari *Twitter* dan melakukan pemodelan algoritma *Naive Bayes* yang menghasilkan akurasi sebesar 87.43% [2]. Kemudian, penelitian terdahulu oleh Junaedy dkk pada 2022 yang membahas topik “Komparasi Algoritma *Support Vector Machine* Dan *Naive Bayes* Pada Analisis Sentimen Formula-E Jakarta”. Penelitian ini dilakukan untuk menghasilkan kategori sentimen dan membandingkan hasil akurasi dari tiap algoritma, maka dari itu didapatkan akurasi tertinggi sebesar 70.57% dengan menggunakan algoritma *Naive Bayes*, sedangkan penerapan algoritma *Support Vector Machine* menghasilkan akurasi sebesar 65.3% [3].

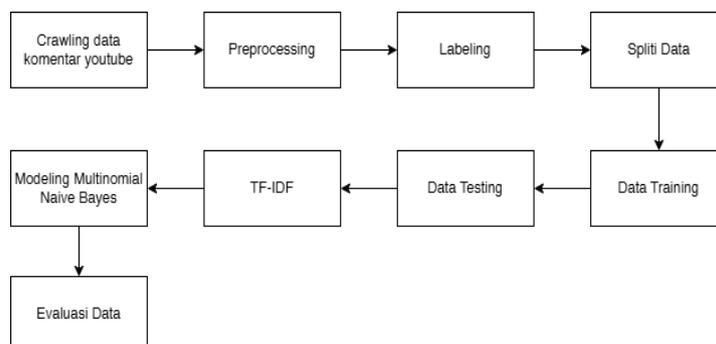
Dalam melakukan penelitian ini, peneliti akan melakukan analisis sentimen komentar *YouTube* dengan penerapan algoritma *Multinomial Naive Bayes*. Algoritma *Multinomial Naive Bayes* sangat cocok dalam memperkirakan peluang dimasa depan berdasarkan kejadian lampau [4]. Penelitian ini menggunakan dataset yang ada pada komentar *YouTube* di salah satu media yang mengunggah sebuah konten terkait mobil listrik buatan Jepang yang murah dan ramah lingkungan. Dataset diambil dengan menggunakan bahasa pemrograman *python* yang telah terkoneksi *API YouTube*. Penelitian ini dilakukan untuk menentukan kategori sentimen masyarakat pada kolom komentar *YouTube* menjadi kategori sentiment negatif atau positif dengan memanfaatkan metode *InSet Lexicon* saat proses *labeling*.

Dalam menghasilkan *labeling* positif dan negatif diambil dari kamus *InSet Lexicon* yang mana terdapat deretan daftar kata *labeling* positif serta negatif dalam kata tersebut memiliki bobot nilai per kata. Pada kamus *lexicon* terdapat 3610 kata positif dan 6610 kata negatif. Nilai dalam kamus ini berada diantara rentang skor dari -5 sampai +5, *lexicon* akan melakukan perhitungan bobot pada data komentar [5].

Perbedaan penelitian ini dengan penelitian sebelumnya terdapat pada platform dan metode *labeling* yang digunakan. Penelitian ini menganalisis sentimen komentar *YouTube* tentang kemunculan mobil listrik menggunakan algoritma *Multinomial Naive Bayes* dan metode *InSet Lexicon* untuk pelabelan otomatis. Sebaliknya, penelitian sebelumnya oleh [2] menggunakan data dari *Twitter* dengan algoritma *Naive Bayes* tanpa fokus pada metode *labeling*, sementara penelitian [3] membandingkan algoritma *Support Vector Machine* dan *Naive Bayes* tanpa penerapan metode *lexicon*.

2. METODE PENELITIAN

Dalam implementasi algoritma *Multinomial Naive Bayes* untuk membangun sistem analisis sentimen dilakukan beberapa proses penting sebagai bagian dari desain atau rancangan utama Pada gambar 1. ditampilkan sampel mengenai alur sistem penelitian, sebagai berikut.



Gambar 1. Alur Sistem Penelitian

Pada alur pertama merupakan *crawling* data atau mengumpulkan data dengan bahasa pemrograman *python* yang terkoneksi *API YouTube*, data didapatkan berdasarkan *ID YouTube* dari salah satu konten *YouTube*. Setelah itu, hasil *crawling data* disimpan kedalam *database* yang telah dibuat. Data yang telah disimpan dalam *database* diolah melalui proses *text preprocessing* untuk menghasilkan dataset yang *clean* agar data dapat diolah dengan menggunakan pemodelan algoritma *Multinomial Naive Bayes*. Setelah proses tersebut selesai, data yang *clean* kembali disimpan kedalam *database*. Selanjutnya, dataset yang *clean* dilakukan proses pelabelan dengan menggunakan *Inset lexicon* untuk mendapatkan dua kategori sentimen, yaitu positif dan negatif. Setelah proses *labeling* selesai, dataset kembali disimpan kedalam *database* dan kemudian dilakukan pembagian data. Data latih dikumpulkan dengan melakukan proses *text preprocessing* dan dilanjutkan dengan representasi numerik menggunakan teknik *Term Frequency Inverse Document Frequency (TF-IDF)* untuk perhitungan pembobotan kata, lalu data dilatih berdasarkan label data termasuk perhitungan probabilitas prior. Sementara itu, data uji dipakai setelah dilatih dalam menilai prediksi terhadap label data, setelah dilakukan prediksi antara data uji dan data latih akan dihasilkan data akurasi dari kedua data tersebut dengan menerapkan *Confusion Matrix Method*. Klasifikasi data sentimen yang dihasilkan akan divisualisasikan dalam bentuk grafik dan *wordcloude*.

2.1 Crawling Data

Proses *crawling data* merupakan tahapan utama dalam mengumpulkan data dengan bahasa pemrograman *python* menerapkan *API YouTube* yang telah terkoneksi, data didapatkan berdasarkan *ID YouTube* dari salah satu konten *YouTube*. Data yang terkumpul kemudian disimpan dengan terstruktur kedalam *database*.

2.2 Text Preprocessing

Pada dasarnya *text mining* berhubungan dengan semua fitur khusus dan umumnya tidak terstruktur, sehingga dilakukan *preprocessing*[1]. Maka dari itu, tahapan *preprocessing* sangat penting dilakukan untuk mengolah data teks asli yang mungkin kurang sesuai, baik dari segi tanda baca, karakter khusus, simbol, angka, bahkan kata atau istilah baru yang ada menjadi sebuah teks yang sesuai, sehingga teks hasil *preprocessing* menjadi lebih bersih, normal serta menyesuaikan standar agar lebih akurat dan mudah untuk dilakukan analisis lebih lanjut. Di bawah ini merupakan beberapa tahapan dalam proses teks *preprocessing* yang dilakukan dalam keperluan penelitian, diantaranya:

- Case Folding*, proses perubahan bentuk kata dari *uppercase* menjadi *lowercase* agar dataset menjadi setara secara keseluruhan.
- Cleaning*, proses ini terdiri dari penghapusan *URL* (link), selain huruf abjad dasar A hingga Z, serta 7 karakter atau simbol khusus seperti, (@), (&), bahkan *hashtag* (#), dan lainnya.
- Tokenizing*, proses merubah kalimat menjadi kata-kata individu atau token [6].
- Normalisasi, proses perubahan informasi teks agar menjadi bentuk yang menyesuaikan standar.
- Stemming*, proses reduksi kata menjadi kata dasar saja, menggunakan aturan kamus kata dasar [3].
- Menghilangkan *Stopword*, proses ini dapat dilakukan secara otomatis dengan menerapkan *library* yang ada pada bahasa pemrograman *python* yaitu *library nltk.corpus stopwords* [7], [8].

2.3 Labeling

Tahap *labeling* dapat diproses secara *labeling* otomatis juga manual, namun dalam penelitian ini pelabelan hanya dilakukan secara otomatis. Pelabelan otomatis dilakukan dengan metode *InSet lexicon*, dengan memanfaatkan kamus *InSet Lexicon* yang mana terdapat deretan daftar kata *labeling* positif serta negatif. Pada kamus *lexicon* terdapat 3610 kata positif dan 6610 kata negatif. Kamus *lexicon* memiliki nilai dengan rentang skor dari -5 sampai +5, *lexicon* akan melakukan perhitungan bobot kata pada data komentar [9]. Kata-kata bernilai positif memiliki skor antara +1 hingga +5, sementara kata-kata bernilai negatif memiliki skor antara -1 hingga -5. *InSet Lexicon* kemudian menghitung bobot keseluruhan dari data komentar berdasarkan perhitungan skor untuk menentukan apakah kalimat dilabeli sebagai sentimen positif atau sentimen negatif.

2.4 Split Data

Pada proses pembagian data, dataset berlabel positif dan negatif akan dipecah sebagai *data train* dan *data test*. Dalam proses ini dijalankan tiga kali percobaan dengan tiga rasio perbandingan yang berbeda, diantaranya *data train* sebesar 80% dengan *data test* sebesar 20%, *data train* sebesar 70% dengan *data test* sebesar 30%, dan *data train* sebesar 60% dengan *data test* sebesar 40%.

2.5 TF-IDF

Dalam tahapan ini terdapat dua proses utama, yaitu mencari nilai *TF* dan mencari nilai *IDF* [2]. Pencarian nilai *TF* dilakukan dengan mengukur kemunculan kata yang sering. Sedangkan, pencarian nilai *IDF* dengan mengukur pentingnya kata [10]. Perhitungan nilai *TF-IDF* ini dilakukan dengan beberapa tahapan sebagai berikut [7].

a. Nilai Term Frequency atau TF

$$TF_{(w,d)} = \frac{K_{w,d}}{J_d} \quad (1)$$

b. Nilai Inverse Document Frequency atau IDF

$$IDF_{(w,D)} = \log\left(\frac{K}{|\{d \in D: w \in d\}|}\right) \quad (2)$$

c. Nilai *TF-IDF*

$$TF - IDF_{(w,d,D)} = TF_{(w,d)} \times IDF_{(w,D)} \quad (3)$$

Keterangan :

w : Kata
 d : Dokumen
K_{w,d} : Banyaknya kemunculan kata-w didokumen d
J_d : Total kata dalam dokumen d
 D : Total dokumen
 K : Keseluruhan jumlah dokumen dalam kumpulan
 $|\{d \in D: w \in d\}|$: Total dokumen pada kata-w

2.6 Modelling Multinomial Naïve Bayes

Salah satu algoritma yang berfungsi melakukan klasifikasi variabel tertentu dengan *probability* dan *statistics method* adalah algoritma *Multinomial Naive Bayes*. Algoritma ini berasumsi bahwa proses klasifikasi dilakukan secara mandiri oleh semua variabel hal ini biasa disebut *class conditional independence* yang didasarkan pada *teorema bayes* tentang *conditional probability* [11]. Penelitian akan diawali dengan proses *data crawling* untuk pengambilan data sebagai dataset dalam proses selanjutnya, kemudian dibuat daftar kata (wordlist) yang berfungsi mengumpulkan semua kata unik yang ada pada data latih. Setelah semua ini dilakukan, akan dilakukan perhitungan prior atau peluang awal kemunculan suatu kategori terhadap data latih. Kemudian, dilakukan perhitungan probabilitas kondisional dengan *smoothing laplace* menggunakan nilai ($\alpha=1$) untuk menghindari probabilitas yang bernilai 0. Setelah mendapatkan probabilitas kondisional, dapat dilanjutkan perhitungan skor tiap kelas atau class dengan perbandingan dibawah ini, sebagai berikut [1].

$$Score(c) = \log P(c) + \sum (\log P(w|c) \times tfidf(w, d)) \quad w \in d \quad (4)$$

Keterangan:

$\log P(w|c)$: Probabilitas pada kata w yang muncul pada class c
 $tfidf(w, d)$: Nilai *TF-IDF* yang berasal dari kata w dalam data d

2.7 Confusion Matrix

Tahapan terakhir pada algoritma *multinomial naive bayes* dalam membuat evaluasi dapat digunakan sebuah metode klasifikasi yang disebut dengan *Confusion Matrix Method*. Metode ini diawali dengan penentuan dan perhitungan jumlah *TruePositive*, *FalsePositive*, *TrueNegative*, dan *FalseNegative* yang terdapat disetiap baris kolom komentar, setelah itu dilakukan perhitungan untuk menentukan nilai akurasi, presisi, *recall*, dan *f1-score*. Dibawah ini merupakan perhitungan pada metode *Confusion Matrix* sebagai berikut [12].

a. Rumus perhitungan nilai akurasi (Ac) dari evaluasi model

$$Ac = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)$$

b. Rumus perhitungan dalam mendapatkan nilai presisi (Pr) dari evaluasi model

$$Pr = \frac{TP}{TP + FP} \quad (6)$$

c. Rumus perhitungan dalam mendapatkan nilai *recall* (Re) dari evaluasi model

$$Re = \frac{TP}{TP + FN} \quad (7)$$

d. Rumus perhitungan dalam mendapatkan nilai *f1-score* dari evaluasi model

$$f1 - score = 2 \times \frac{Re \times Pr}{Re + Pr} \quad (8)$$

Keterangan:

TP : *TruePositive* (Score positif yang diprediksi benar)

FP : *FalsePositive* (Score positif yang diprediksi salah)

TN : *TrueNegative* (Score negatif yang diprediksi benar)

FN : *FalseNegative* (Score negatif yang diprediksi salah)

3. HASIL DAN PEMBAHASAN

Dalam memastikan replikasi dan seberapa akurat hasil penelitian yang dilakukan, perlu dilakukan uji coba terhadap sistem agar didapatkan kesesuaian seperti yang telah dirancang dan diharapkan. Maka dari itu, diberikan penjabaran hasil dari analisis sentimen, sebagai berikut.

3.1 Crawling Data

Tahapan *crawling* data dilakukan dengan menggunakan bahasa pemrograman *python* yang terkoneksi *API YouTube*. Seluruh dataset yang berisikan informasi berupa *id*, *published*, *user*, dan *comment* yang didapatkan akan disimpan dalam basis data atau database yang kemudian diproses pada tahap selanjutnya. Dalam penelitian ini komentar yang diambil berdasarkan salah satu konten video yang bersumber dari saluran *LugNutz Auto Junkie* dengan judul video “Akhirnya Jepang Bikin Mobil Listrik Murah! *Daihatsu Ayla EV* Siap Bikin Pusing *Hyundai & Wuling*”, yang dapat di akses dengan *url YouTube* <https://www.youtube.com/watch?v=gOVgHyQP764> dengan jumlah data komentar sebanyak 700. Hasil dari *crawling* data dapat dilihat pada tabel 1.

Tabel 1. *Crawling* Data

Komentar YouTube	Username	Published
kereenn top dah, Insya Allah beli di tahun depan Aamiin ☐ Daihatsu mah kalau buat mobil konsep ga ada yang jadi! 😊😊😊	@asrulytb	20 September 2022
	@simplyme5123	19 Agustus 2022

3.2 Text Preprocessing

Dalam penelitian ini terdapat enam tahapan *preprocessing* lebih lanjut yang dilakukan untuk menghasilkan dataset yang clean atau bersih dari noise juga membuat dataset lebih terstruktur[13]. Keenam tahapan ini meliputi *case folding*, *cleaning*, normalisasi, *tokenizing*, *stemming*, dan menghilangkan *stopword*.

3.2.1 Case Folding

Dalam tahap ini, dilakukan perubahan keseluruhan dataset menjadi huruf kecil (lowercase) seperti pada tabel 2. terdapat kata "Daihatsu" menjadi "daihatsu" dan lain sebagainya.

Tabel 2. *Case Folding*

Sebelum	Sesudah
Daihatsu buat mobil konsep ga ada yang jadi 😊	daihatsu buat mobil konsep ga ada yang jadi 😊

3.2.2 Cleaning

Dalam tahap ini, dilakukan *cleaning* untuk menghapus segala informasi pada teks yang tidak sesuai agar data lebih konsisten. Pada tabel 3. dilakukan penghapusan segala karakter atau simbol yang tidak sesuai, seperti *user mention* hingga *hashtag*.

Tabel 3. *Cleaning*

Sebelum	Sesudah
daihatsu buat mobil konsep ga ada yang jadi 😊	daihatsu buat mobil konsep ga ada yang jadi

3.2.3 Tokenizing

Dalam tahap ini, dilakukan perubahan kalimat menjadi per kata seperti pada tabel 4. Hal ini dilakukan agar tiap karakter yang berbeda dapat terbaca dalam proses.

Tabel 4. *Tokenizing*

Sebelum	Sesudah
daihatsu buat mobil konsep ga ada yang jadi	[`daihatsu`, `buat`, `mobil`, `konsep`, `ga`, `ada`, `yang`, `jadi`]

3.2.4 Normalisasi

Dalam tahap ini, dilakukan penyesuaian kata yang berstandar KBBI. Selain itu, dilakukan representasi dan transmudasi kata untuk mengurangi variasi kata tak sesuai. Seperti pada tabel 5. terdapat kata "aja" menjadi "saja", "ga" menjadi "tidak", "tp" menjadi "tapi", dan sebagainya.

Tabel 5. Normalisasi

Sebelum	Sesudah
daihatsu buat mobil konsep ga ada yang jadi	daihatsu buat mobil konsep tidak ada yang jadi

3.2.5 Stemming

Dalam tahap ini, dilakukan proses menghilangkan awalan atau akhiran pada sebuah kata menjadi kata dasarnya saja. Hal ini dilakukan untuk menghindari kesamaan makna kata. Seperti pada tabel 6. terdapat kata "mengingat" menjadi "ingat" yang memiliki arti yang sama.

Tabel 6. *Stemming*

Sebelum	Sesudah
belum tahu harganya kenapa judulnya murah	belum tahu harga kenapa judul murah

3.2.6 Menghilangkan *Stopword*

Dalam hal ini, dilakukan penghilangan *stopword* seperti pada tabel 7. untuk menghapus kata umum yang tidak memiliki nilai lebih. Hal ini dilakukan dengan memanfaatkan daftar kata *stopword* (Stopword List).

Tabel 7. Menghilangkan *Stopword*

Sebelum	Sesudah
daihatsu buat mobil konsep tidak ada yang jadi	daihatsu mobil konsep

3.3 Labelling

Pada tahap ini, proses pelabelan berfungsi untuk mengklasifikasikan hasil berupa dua kategori sentimen, yaitu sentimen positif dan sentimen negatif. Dalam penelitian ini pelabelan otomatis dengan menggunakan *InSet Lexicon* menghasilkan dua kategori sentimen, yaitu 425 komentar positif dan 234 komentar negatif dari keseluruhan jumlah 659 data yang diproses (data bersih).

3.4 Data Split

Dalam proses splitting data atau pembagian data, dua kategori dataset yang telah dihasilkan akan dipecah menjadi data latih dan data uji. Percobaan dilakukan sebanyak tiga kali dengan tiga rasio perbandingan yang berbeda. Dalam hal ini akan didapatkan akurasi tertinggi dari tiga percobaan pada rasio yang berbeda.

3.5 TF-IDF

Dalam proses prediksi, dilakukan perhitungan nilai tiap bobot dengan metode *TF-IDF* berdasarkan data latih. Langkah pertama yang dilakukan merupakan perhitungan nilai *TF* pada tabel 8. dengan membagi frekuensi data dengan jumlah keseluruhan kata pada dokumen yang ada sebagai berikut.

Tabel 8. Nilai TF

Teks	Kata	TF
"musnah mobil bensin ganti mobil listrik"	musnah	1/6
	mobil	2/6
	bensin	1/6
	ganti	1/6
	listrik	1/6

Langkah kedua yang dilakukan merupakan perhitungan nilai *IDF* pada tabel 9. dengan menghitung *log* dari hasil pembagian total dokumen dan jumlah dokumen kata tertentu sebagai berikut.

Tabel 9. Nilai IDF

Teks	Kata	IDF
"mobil bensin ganti mobil listrik"	musnah	$\log(6/1) = 0.77815$
	mobil	$\log(6/3) = 0.30103$
	bensin	$\log(6/1) = 0.77815$
	ganti	$\log(6/1) = 0.77815$
	listrik	$\log(6/3) = 0.3010$

Langkah terakhir adalah perhitungan nilai *TF-IDF* pada tabel 10. yang didapatkan dari hasil kali dari nilai *TF* dan *IDF*.

Tabel 10. Nilai TF-IDF

Teks	Kata	TF-IDF
"musnah mobil bensin ganti mobil listrik"	musnah	0.12969
	mobil	0.10034
	bensin	0.12969
	ganti	0.12969
	listrik	0.05017

Dibawah ini merupakan tabel kamus kata *data training* pada tabel 11. yang didapatkan dari perhitungan frekuensinya dengan menggunakan metode *TF-IDF* sebagai berikut.

Tabel 11. Kamus Kata

Kamus Kata	Jumlah
mobil listrik keren simple produk moga beli amin musnah bensin ganti suka bentuk bodi ekterior ayla mantap informasi buruk berani banget bilang bensin murah harga kelas tau juta alasan mahal jarak tempuh jakarta karawang lancar daya baterai	37

3.6 Modeling Multinomial Naïve Bayes

Setelah menghitung bobot kata (TF-IDF) dari data latih, dilakukan perhitungan pada tabel 12. untuk mencari prior atau peluang awal kemunculan pada sentimen kategori positif dan negatif.

Tabel 12. Nilai Prior

Prior	Hasil
Positive	$3/6 = 0.5$
Negative	$3/6 = 0.5$

Kemudian pada tabel 13. dilakukan proses perhitungan dalam mencari nilai probabilitas kondisional dengan *smoothing laplace*.

Tabel 13. Probabilitas Kondisional

Kata	Kata	Kelas	Nilai
"Mantap Ayla"	Mantap	Positif	$\frac{0.07782 + 1}{(1.98633) + 1.37} = 0.0276$
		Negatif	$\frac{0 + 1}{(1.9525) + 1.37} = 0.0256$

$$\begin{array}{l}
 \text{Positif} \quad \frac{0.04771 + 1}{(1.98633) + 1.37} = 0.0268 \\
 \text{Ayla} \\
 \text{Negatif} \quad \frac{0.05301 + 1}{(1.9525) + 1.37} = 0.027
 \end{array}$$

Dengan mencari nilai probabilitas kondisional dilakukan perhitungan pada tabel 14. untuk menentukan *score* positif dan negatif dari dokumen yang diuji.

Tabel 14. Score Positive dan Negative

Score (positif)	=	$\frac{\log(0,5) + (\log(0,02765) \times 0,11116) + (\log(0,02823) \times 0,06816) + (\log(0,02565) \times 0) + (\log(0,02565) \times 0) + (\log(0,02687) \times 0,06816) + (\log(0,02565) \times 0) + (\log(0,02565) \times 0)}{(1.98633) + 1.37}$	=	-0,6869
Score (negatif)	=	$\frac{\log(0,5) + (\log(0,02567) \times 0,11116) + (\log(0,02567) \times 0,06816) + (\log(0,02567) \times 0) + (\log(0,02567) \times 0) + (\log(0,02703) \times 0,06816) + (\log(0,02567) \times 0) + (\log(0,02567) \times 0)}{(1.9525) + 1.37}$	=	-0,6931

Hasil dari perhitungan *score* terhadap *class* diatas dapat diketahui bahwa peluang pada dokumen kelas positif menghasilkan *score* -0,686918357 lebih besar daripada kelas negatif yang menghasilkan *score* -0,693134377. Maka, dapat disimpulkan bahwa sampel *data test* dikategorikan sentimen positif.

3.7 Evaluasi

Dalam pengujian metode terdapat tahapan yang tidak boleh dilewatkan saat membangun sistem. Hal ini dilakukan agar dapat ditampilkan hasil evaluasi akan tingkat keakuratan sistem sesuai yang diharapkan atau tidak. Pengujian ini, dilakukan sebanyak tiga kali dengan tiga rasio perbandingan yang berbeda, diantaranya *data train* sebesar 80% dengan *data test* sebesar 20%, *data train* sebesar 70% dengan *data test* sebesar 30%, dan *data train* sebesar 60% dengan *data test* sebesar 40%. Oleh karena itu, berdasarkan ketiga perbandingan ini, akan dihasilkan evaluasi berupa *accuracy*, *presisi*, *recall*, dan *f1-score* dengan menerapkan metode *Multinomial Naive Bayes*. Pada tabel 15. terdapat perhitungan dengan metode *Confusion Matrix* sebagai berikut.

Tabel 15. Confussion Matrix

Rasio Data Split	Confussion Matrix				Score			
	TP	TN	FP	FN	Akurasi	Presisi	Recall	F1-Score
60%:40%	106	60	17	31	77.57%	86.18%	77.37%	81.54%
70%:30%	74	45	24	17	74.38%	75.51%	81.32%	78.31%
80%:20%	48	29	16	14	71.96%	75%	77.42%	76.29%

Berdasarkan perhitungan metode *Confusion Matrix* pada tabel 15. didapatkan akurasi tertinggi sebesar 77.57% pada pengujian dengan rasio perbandingan *data train* sebesar 60% dengan *data test* sebesar 40%. Hal ini dikarenakan adanya penambahan *data training* yang dapat mengenali pola, hubungan, dan variasi dalam data dengan lebih baik, sehingga menjadi lebih terlatih dan mampu membuat prediksi yang lebih akurat pada data uji.

4. KESIMPULAN

Dalam analisis 700 komentar *YouTube* mengenai sentimen masyarakat terhadap kendaraan mobil listrik *Ayla EV*, beberapa langkah penting dilakukan. Proses ini meliputi pengambilan data (*crawling*) menggunakan *API YouTube*, *preprocessing* teks, pelabelan otomatis dengan *Inset Lexicon*, serta pembagian data untuk validasi model algoritma *Multinomial Naive Bayes* menggunakan *Confusion Matrix Method*. Hasil evaluasi menunjukkan akurasi sebesar 77.57% untuk rasio 60%:40%, 74.38% untuk rasio 70%:30%, dan 71.96% untuk rasio 80%:20%. Perolehan akurasi terbesar terdapat pada uji coba perbandingan rasio 60%:40% dengan nilai presisi sebesar 86.18%, *recall* sebesar 77.37%, dan *f1-score* sebesar 81.54%. Penelitian ini juga mengembangkan antarmuka pengguna grafis (GUI) berbasis web dengan *PHP* (Laravel) untuk menampilkan hasil analisis sentimen, memberikan akses dan pemahaman yang jelas bagi pengguna dan peneliti.

DAFTAR PUSTAKA

- [1] R. A. Firsttama, A. A. Arifiyanti, and D. S. Y. Kartika, “Analisis Sentimen Komentar Youtube Konferensi Tingkat Tinggi G20 Menggunakan Metode Naive Bayes,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 2, pp. 282–285, 2024, doi: 10.47233/jteksis.v6i2.1263.
- [2] P. G. Aryanti and I. Santoso, “Analisis Sentimen Pada Twitter Terhadap Mobil Listrik Menggunakan Algoritma Naive Bayes.” [Online]. Available: <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive>
- [3] F. Z. Junaedy *et al.*, “Komparasi Algoritma Support Vector Machine Dan Naive Bayes Pada Analisis Sentimen Formula-E Jakarta Tahun 2022,” *Jurnal IKRAITH-Informatika*, vol 7, no. 2, pp. 157-164, 2023. [Online]. Available: <https://youtu.be/mhAEiHuVFxY>
- [4] S. Hasanah, I. Purwasih, and I. Santoso, “Analisis Sentimen Terhadap Masyarakat Adanya Uang Kertas Baru Menggunakan Algoritma K-Nearest Neighbor (KNN).” [Online]. Available: <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive>
- [5] Y. N. Prasetya and D. Winarso, “Penerapan Lexicon Based Untuk Analisis Sentimen Pada Twiter Terhadap Isu Covid-19,” *Jurnal Fasilkom*, vol. 11, no. 2, pp. 97–103, 2021, Accessed: Aug. 12, 2024. [Online]. Available: <https://ejournal.umri.ac.id/index.php/JIK/article/view/2772/1566>
- [6] S. Alfarizi and E. Fitriani, “Analisis Sentimen Kendaraan Listrik Menggunakan Algoritma Naive Bayes dengan Seleksi Fitur Information Gain dan Particle Swarm Optimization,” *Indonesian Journal on Software Engineering (IJSE)*, vol. 9, no. 1, pp. 19–27, 2023, [Online]. Available: <http://ejournal.bsi.ac.id/ejournal/index.php/ijse>
- [7] I. Verawati and S. N. Jaelani, “Jurnal Media Informatika Budidarma Analisis Sentimen Pengguna Twitter Terhadap Bus Listrik Menggunakan Naive Bayes,” 2024, doi: 10.30865/mib.v8i2.7030.
- [8] F. Rahutomo, A. Retno, and T. H. Ririd, “Evaluasi Daftar Stopword Bahasa Indonesia,” vol. 6, no. 1, pp. 41–48, 2019, doi: 10.25126/jtiik.201861226.
- [9] Y. N. Prasetya and D. Winarso, “Penerapan Lexicon Based Untuk Analisis Sentimen Pada Twiter Terhadap Isu Covid-19,” *Jurnal Fasilkom*, vol. 11, no. 2, pp. 97-103, 2021.
- [10] F. Syofiani, S. Alam, and M. I. Sulisty S, “Analisis Sentimen Penilaian Masyarakat Terhadap Childfree Berdasarkan Komentar di Youtube Menggunakan Algoritma Naive Bayes,” *Jurnal Teknologi Informatika dan Komputer*, vol. 9, no. 2, pp. 688–703, 2023, doi: 10.37012/jtik.v9i2.1661.
- [11] A. Wahid and G. Saputri, “Analisis Sentimen Komentar Youtube Tentang Relawan Patwal Ambulance Menggunakan Algoritma Naive Bayes dan Decision Tree,” *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 2, p. 319, 2022, doi: 10.30865/json.v4i2.4941.
- [12] M. J. Palepa, N. Pratiwi, and R. Q. Rohmansa, “Analisis Sentimen Masyarakat Tentang Pengaruh Politik Identitas Pada Pemilu 2024 Terhadap Toleransi Beragama Menggunakan Metode K-Nearest Neighbor,” *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 1, pp. 389–401, 2024, doi: 10.29100/jupi.v9i1.4957.
- [13] R. Asrianto and M. Herwinanda, “Analisis Sentimen Kenaikan Harga Kebutuhan Pokok Dimedia Sosial Youtube Menggunakan Algoritma Support Vector Machine,” *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 3, pp. 431–440, 2022, doi: 10.37859/coscitech.v3i3.4368.